

Tandy 4000

Technical Reference Manual



**Tandy 4000
Page Insertion Guide
Important Customer Note:**

A gray stripe has been printed along the right edge of the title page of each of the sections to facilitate your finding the beginning of the section. Also, a tabbed divider for each section has been provided for insertion at this point.

**Pages To Be Inserted At The End
of the Main Logic Section**

Note to Theory of Operation (Part of Appendix A)	1 Sheet
Timing Diagrams	Sheets 1 of 4 thru 4 of 4
Main Logic Schematic (Rev. A)	8000284 Sheets 1 of 8 thru 8 of 8
Main Logic Schematic (Rev. C) Part of Appendix A	8000284 Sheets 1 of 9 thru 9 of 9
Serial/Parallel Card	8000264 Sheets 1 of 3 thru 3 of 3

**Pages To Be Inserted At The End
of the Devices Section**

U49 Signal Generator Chip Spec.(Part of Appendix A)	1 Sheet
U53 Address Decode Chip Spec. (Part of Appendix A)	1 Sheet
80387 Numeric Processor Chip Spec.(Part of Appendix A)	18 Sheets Sheets 4-138 thru 4-172

**Foldout Page To Be Inserted At The End
of the 192 Watt Power Supply Section**

Schematic	AAl3265 Sheet 1 of 1
-----------	-------------------------

**Foldout Page To Be Inserted At The End
of the 192 Watt Power Supply Section**

Schematic 8790095
 Sheet 1 of 1

**Foldout Page To Be Inserted At The End
of the Keyboard Section**

Schematic 15459
 Sheet 1 of 1

**Foldout Pages To Be Inserted At The End
of the Disk Drive Section**

Exploded View Sheet 1 of 1

Schematic FC-52(MD)
 Sheet 1 of 1

TANDY COMPUTER PRODUCTS

**TANDY 4000
TECHNICAL REFERENCE MANUAL
Cat. No. 25-4108**

TANDY COMPUTER PRODUCTS

**Tandy 4000 Technical Reference Manual
Copyright 1987, Tandy Corporation
All Rights Reserved.**

Reproduction or use of any portion of this manual without written permission from Tandy Corporation and/or its licensor, is prohibited. While reasonable efforts have been taken in the preparation of this manual to assure its accuracy, Tandy Corporation assumes no liability resulting from any errors in or omissions from this manual, or from the use of the information obtained herein.

**Tandy 4000
Technical Reference Manual
Contents**

Sections

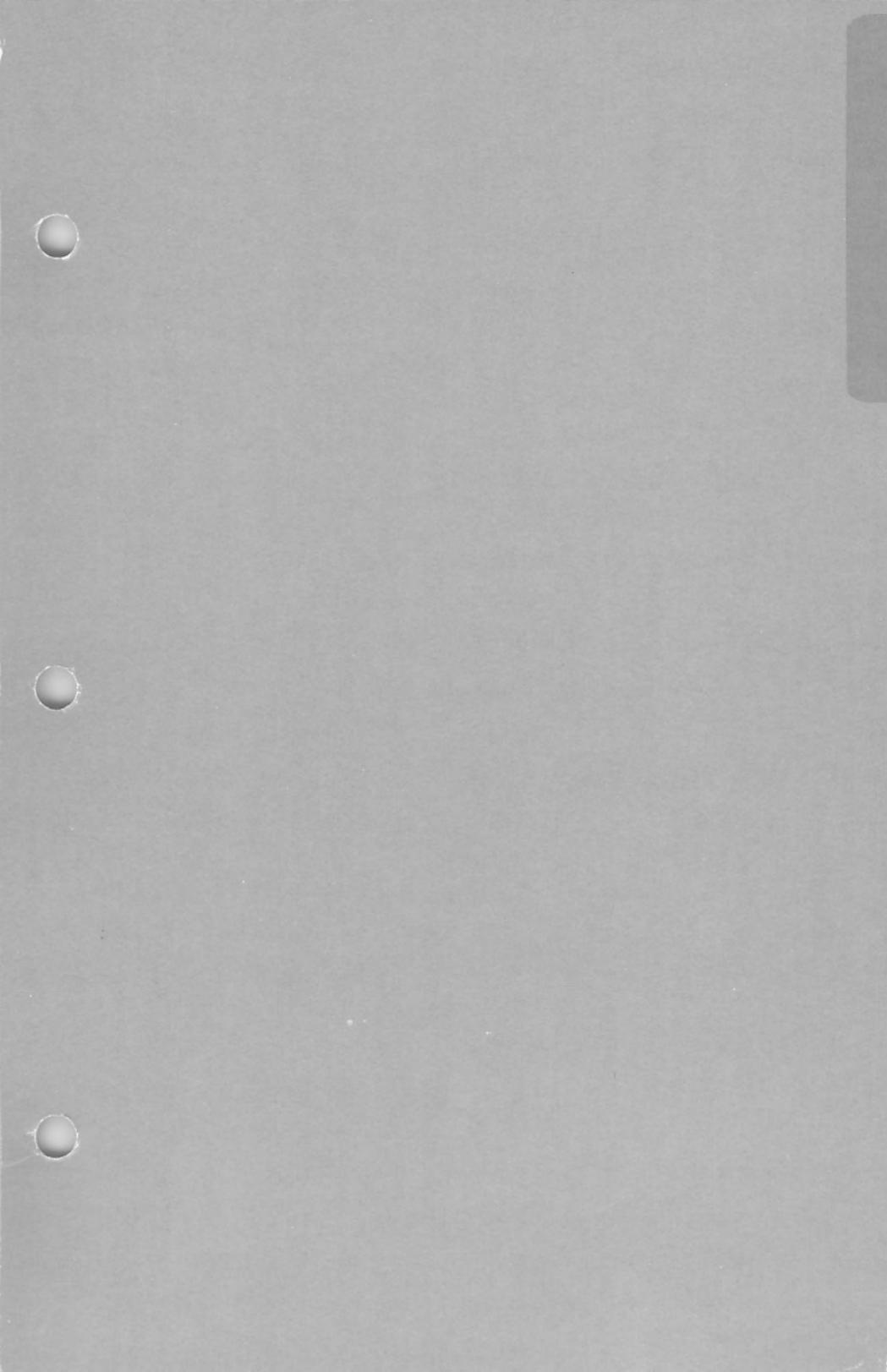
Main Logic Board
Devices
Power Supplies
Keyboard
Disk Drive
Options
Software

Important Customer Notes:

A gray stripe has been printed along the right edge of the title page of each of the sections to facilitate your finding the begining of the section. Also, a tabbed divider for each section has been provided for insertion at this point.

If you need additional technical reference information for your 80387 Math Co-processor or other update information, obtain the 700-4108 package from your local Radio Shack Store.







TANDY COMPUTER PRODUCTS

4000 Main Logic Board



Contents

Scope of Document.....	4
Introduction.....	5
Specifications and Features.....	7
Switch Settings And Jumper Pin Configurations.....	8
Functional Description.....	9
80386 Processor.....	9
Weitek Numeric Processor (Optional).....	10
Clock Generation.....	11
Command and Control Signals.....	11
Generation.....	11
82C301 Bus Controller.....	11
82A306 Control Buffer.....	18
Memory Control and Refresh.....	20
82C302 Memory Controller.....	20
Memory Configurations.....	20
Memory Expansion.....	31
Address Buffers.....	32
82A303 High Address Buffers.....	32
82A304 Low Address Buffers.....	34
Data Buffers And Conversion Logic.....	35
82A305 Data Buffers.....	35

I/O Decode.....	38
82C206 Integrated Peripherals Controller.....	38
DMA Controller.....	44
Interrupt Controller.....	64
Timer/Counter.....	76
Real-Time Clock.....	85
Floppy Disk Controller.....	93
80287 Numeric Processor (Optional).....	120
Serial/Parallel Interface.....	120
Keyboard Control.....	122
Interface Requirements.....	126
System I/O Bus Interface.....	126
PC Bus.....	127
AT Bus.....	131
Memory Expansion Bus.....	134

SCOPE OF DOCUMENT

This document provides detailed specifications on the Tandy® 4000.

The Tandy 4000 computer system is composed of four buses:

- **The 80386 Local Bus**--consists of the 80386 CPU, the 82C301 bus controller, and the optional Weitek Numeric Processor.
- **The 32-Bit Memory Bus**--consists of up to four banks of 32-bit access dynamic RAM (DRAM). Two banks are located on the main logic board which also has an expansion slot for an additional two banks. Each bank consists of either four 256Kb x 9 DRAM modules or four 1Mb x 9 DRAM modules with an access time of 100 ns.
- **The System I/O (AT) Bus**--contains the BIOS ROM devices, the floppy disk controller (FDC), and eight expansion slots (two IBM ® PC-compatible and six AT-compatible) for optional devices, and the 80287 co-processor.
- **The Peripheral Bus**--an extension of the system I/O bus; contains the direct memory access (DMA), interrupt, counter/timer, speaker, keyboard, and real-time clock (RTC) operations.

Each bus has different timing characteristics. For example, the 80386 bus operates at a clock rate of 16MHz with either no wait state or one wait state when accessing 32-bit RAM, and the AT bus operates at 8MHz with up to four wait states.



INTRODUCTION

Compatibility. The Tandy® 4000 is designed to be as compatible as possible with the existing Tandy 3000 and the IBM AT®, in terms of both hardware and software.

Most operating systems and application programs that run on the Tandy 3000 will run on the Tandy 4000. The Tandy 3000 BIOS ROMs will not run in the Tandy 4000, however, because of the extra programming information needed to configure the Tandy 4000.

Certain hardware functions from the Tandy 3000 are duplicated on the Tandy 4000. All I/O and memory maps, interrupts, DMA channels, the keyboard interface, and the AT bus have the same configurations and timing parameters. Additional circuitry supports CPU 32-bit data movements to and from the 8- and 16-bit I/O and memory devices.

Although 8- and 16-bit memory cards can be used on the AT bus, performance will be significantly decreased compared to the 32-bit memory options.

Modular Design. The Tandy 4000 is modular in design to allow maximum flexibility in system configuration. The computer consists of a main unit and a detachable keyboard with coiled cable. The main unit is supplied with one internal 3½-inch, 1.44Mb floppy disk drive. The standard types of monitors used with the Tandy 4000 are the monochrome, RGB, and EGA monitors. Because these units are modular, they can be placed on top of the main unit or at any convenient location.

Memory. The Tandy 4000 comes standard with 1 megabyte of 32-bit wide system RAM. This 32-bit wide System memory can be expanded to 2, 4, 8, 10 or 16 megabytes, the maximum RAM allowed by the system memory map.

Additional Features. Other features include a parallel printer port, a serial port, and a speaker for audio feedback.

Main Unit. The main unit is the heart of the Tandy 4000. It houses the main logic assembly, the system power supply, optional hard disk unit, and the floppy disk drive.

The main logic assembly is a large board mounted on the floor of the main unit and connected to the keyboard, power supply, and disk drive by a series of cables.

The power supply is a 192 watt switching regulator type, designed to provide adequate power capacity for a fully configured system using all the option slots.

The floppy disk drive uses 3½-inch, double-sided, double-density diskettes to read or write data. These are soft-sector diskettes. The disk drive assembly is installed in the main unit. The floppy disk stores approximately 1.44 megabytes (formatted) of data. All system programs, with the exception of the system startup sequence, are stored on disk.

Monitors. A monochrome, EGA, or RGB display can be used with the Tandy 4000. The appropriate display adapter must be installed in the main unit.

FEATURES

16MHz, 32-bit 80386 processor with on-chip memory management and protection.

Standard 1Mb, 0-wait state, 32-bit memory.

- Supports up to 8 megabytes of 32-bit wide memory on board
- Bus addressing supports PC (8-bit), AT (16-bit), and 32-bit data widths

Standard 1.44Mb, 3½-inch floppy disk drive.

- Selectable 1.44Mb or 720Kb format allows compatibility with the Tandy 3000, Tandy 1000, IBM AT, or IBM PC
- Channel for additional internal floppy disk drive of 3½-inch or 5½-inch type

Channel for user-selectable hard disk drive.

Six IBM AT-compatible slots for standard peripherals and additional memory (16-bit) expansion.

Two IBM PC-compatible slots for standard peripherals and additional memory (8 bit) expansion.

One 32-bit memory upgrade slot.

- 2Mb or 8Mb for a total of 16Mb
- 0 wait state (statistical)

True software compatibility with the IBM AT in the single-user MS-DOS® (real) mode or the multi-user (protected) mode. Can support Xenix or a UNIX based operating system and MS-DOS concurrently in the Protected Mode.

Built-in real-time clock with CMOS RAM and battery backup.

Serial/parallel interface adapter included.

Enhanced IBM AT-compatible keyboard.

Support for optional 80287 or Weitek 1167 math co-processor.

192 watt power supply.

PHYSICAL SPECIFICATIONS

Width	19 in.
Height	6 in.
Depth	18 in.
Weight	47 lbs.

SWITCH SETTINGS AND JUMPER PIN CONFIGURATIONS

Jumper	Function	As Shipped
E1-E2	Color/Monochrome Monitor installed = color not installed = monochrome	installed
E3-E4	Not Used	not installed
E5-E6 and E6-E7	Floppy Disk Controller Select E5-E6 installed = primary FDC E6-E7 installed = secondary FDC	installed not installed
E8-E9	80287 Option installed = option present not installed = option not present	not installed

Table 1. Switch Settings and Jumper Configurations.

FUNCTIONAL DESCRIPTION

The following sections of this document explain in detail the functions of the various devices found in the Tandy® 4000.

80386 PROCESSOR

General Overview

The 80386 processor is an advanced 32-bit microprocessor designed for applications needing very high performance. It is also optimized for multi-tasking operating systems.

The 32-bit registers and data paths support 32-bit addresses and data types. The processor addresses up to four gigabytes of physical memory and 64 terabytes of virtual memory. Even though the processor will access up to four gigabytes of memory the implementation used in the Tandy 4000 only supports up to 16 megabytes.

The integrated memory management and protection architecture includes address translation registers, advanced multi-tasking hardware, and a protection mechanism to support operating systems. The processor also allows the simultaneous running of multiple operating systems.

The 80386 consists of a central processing unit, a memory management unit, and a bus interface.

Central Processing Unit

The central processing unit consists of the execution unit and the instruction unit.

The execution unit contains the eight 32-bit general-purpose registers, which are used for both address calculation and data operations. In addition, it contains a 64-bit barrel shifter, which is used to speedshift, rotate, multiply, and divide data contained in the registers.

The instruction unit decodes the instruction opcodes and stores them in the decoded instruction queue for immediate use by the execution unit.

Memory Management Unit (MMU)

The memory management unit consists of a segmentation unit and a paging unit.

Memory is organized into one or more variable length segments, each up to 4 gigabytes in size. A given region of the linear address space, a segment, has certain attributes associated with it. These attributes include its location, size, type, and protection characteristics. Each task on the 80386 can have a maximum of 16,381 segments of up to 4 gigabytes each, thus, due to the memory management organization of the processor, it provides 64 terabytes of virtual memory to each task. The segmentation unit provides four levels of protection for isolating applications and the operating system from each other. Even though the processor will address up to four gigabytes of memory, the Tandy 4000 only supports up to 16 megabytes.

The 80386 has two modes of operation: Real Address Mode (Real Mode) and Protected Virtual Address Mode (Protected Mode).

In Real Mode, the 80386 operates as a very fast 8086 but with 32-bit extensions if desired. Real Mode is required primarily to set up the processor for Protected Mode operation.

Protected Mode provides access to the sophisticated memory management, paging, and privilege capabilities of the processor. Within Protected Mode, software can perform a task switch to enter into tasks designated as Virtual 8086 Mode tasks. Each such task behaves with 8086 semantics, thus allowing 8086 software to execute.

Bus Interface

Finally, to facilitate high-performance system hardware designs, the 80386 local bus offers address pipelining, dynamic data bus sizing, and direct byte enable signals for each byte of the data bus. These features are discussed in later sections of this manual.

WEITEK NUMERIC PROCESSOR WTL1167 (OPTIONAL)

The optional Weitek Numeric Processor Board connects to the 80386 local bus through a 121-pin PGA socket designated U15. This board provides a 32-bit data path to the CPU and performs 32- and 64-bit floating point operations at 4-5 times the speed of other available co-processors. Added modifications are needed for this option on the Rev. A-1 system board. Revision 3002-0044-00 of the Weitek board works with these modifications, however, Radio Shack® does not provide support for the Weitek board. Also note that the 80387 socket on the Weitek board is not supported by this revision of the system board.

CLOCK GENERATION

The 82C301 (U14) provides three major system clocks:

- The processor clock, CLK2
- The BCLK for the AT bus state machine of the 82C301
- The AT bus clock, SYSCLK

The BCLK is a clock internal to the 82C301.

The clock generation circuitry uses two external clocks, CLK2IN and ATCLK1, as selectable clock sources. These are supplied by TTL oscillator (Y3). CLK2IN is a 32MHz input signal and ATCLK1 is a 12MHz input signal.

The clock input to the 80386 processor is CLK2, which is a 32MHz signal. The clock signal provided to the AT bus is ATSCLK and is rated at 8MHz and may be slowed to 6MHz by the speed command in MS-DOS 3.20.3

COMMAND AND CONTROL SIGNALS

GENERATION

The command and control signals necessary for system operation are generated by two devices: the 82C301 bus controller (U14) and the 82C306 control buffer (U13).

82C301 Bus Controller

This 84-pin PLCC device produces and synchronizes the CPU and system clocks and performs interfacing between the 80386 and the rest of the system control lines.

Reset Control

RESET1* is produced at power up/down or when the reset button on the front panel is depressed. When the RESET1* is asserted, the bus controller asserts RESET3 and RESET4 for a system reset. RESET3 is routed to the 80386 CPU and the Weitek co-processor, and RESET4 is sent to the control buffer. For a warm restart not requiring an extensive reset, RESET2* can be asserted to generate RESET3 for resetting only the processor and co-processor. RESET2* is generated by pressing the Ctrl, Alt, and Del keys on the keyboard simultaneously. RESET3 is also asserted when a CPU shutdown is detected. This differentiation is provided so that some register states can be maintained through the reset if so desired.

Bus Arbitration

General

The bus controller performs the synchronization and control required for communication between the local processor bus, the memory subsystem, and the system I/O (AT) bus. It controls all bus activities and handles the HRQ1, HRQ2, and REFREQ signals by generating a HOLD request to the CPU and arbitrating among these requests in a non-preemptive manner.

Upon the CPU assertion of HLDA, the arbitration logic responds by asserting HLDA1 (for HRQ1), or HLDA2 (for HRQ2). The requesting DMA or master device has control of the bus until it de-asserts the Hold Request signal to terminate the HLDA cycle. During the HLDA cycle, the bus controller generates both SMCMD* and AC0 through AC3 to control the buffer enable and directions for the address and data buffers. HRQ2 is disabled in the Tandy 4000. Bus size conversions are not supported by the bus controller for these bus cycles and if necessary should be performed by the requesting device.

All CPU access cycles are started by the bus controller asserting the MALE* signal. The bus controller then samples the AF32* signal one SCLK clock cycle later. If AF32* is active, the cycle is assumed to be a local bus cycle and the bus controller terminates this cycle when it detects the READY* signal active. In response to an MALE*, if AF32* is detected inactive, the control is passed to the AT bus control portion of the bus controller. At the end of the bus access cycle, the AT bus control logic of the bus controller generates READY* to terminate the processor access cycle.

CPU Bus

Interface to the 80386 requires interpretation of the status lines upon assertion of the ADS* signal. It also requires synchronization and generation of a READY* response to the CPU upon completion of the requested operation.

By interpreting the CPU status lines and the ADS* signal, the bus controller generates control signals MALE* and SMCMD*. In response to each ADS* signal generated by the CPU, an MALE* signal is generated by the bus controller to indicate the start of a new CPU access cycle.

In a non-pipelined CPU cycle, MALE* is generated in response to the ADS* signal being asserted by the 80386. In a pipelined cycle, MALE* is generated when the assertion of the READY* signal is detected for the previous CPU cycle.

If the AF32* signal is not active one cycle after MALE* is asserted, control is passed to the AT bus control section of the bus controller. The bus controller then waits for READY* to become active to terminate the access cycle. The READY* signal can also be generated by the 82C302 (U27) page/interleave memory controller, which controls the system memory access.

SMCMD* indicates a memory cycle for both CPU and non-CPU accesses. During CPU cycles, SMCMD* is generated for all memory cycles by decoding the M/(IO)*, D/(C)* and W/(R)* signals. During non-CPU cycles, SMCMD* is active when XMEMR* or XMEMW* signals are active.

Bus Timeout

An optional feature allows generation of a non-maskable interrupt (NMI) if an internal memory cycle does not complete within a certain timeout period. This occurs if AF32* is asserted in response to MALE* and READY* is not returned to the bus controller within 128 CLK2 cycles. A control bit in the 82C301 Bus Controller's Port B register enables this feature and will be explained later in this section.

AT Bus

The bus controller gains control of the AT bus when AF32* is detected inactive. It also performs the necessary synchronization of control and status signals between the AT bus and the processor. The bus controller supports 8-, 16-, or 32-bit transfers between the processor and 8-, 16-, or 32-bit memory or I/O devices located on the I/O bus.

An AT bus cycle is initiated by asserting the ALE signal that is decoded from the CPU status signals and is terminated by asserting READY*. On the trailing edge of ALE, the signals MCS16* and IOCS16* are sampled to determine the bus size conversion required. The bus controller then enters the command cycle. The bus controller provides the sequencing and timing controls for status and command phases of different AT bus cycles. These controls provide timing emulation of lower-speed I/O channels to maintain compatibility with AT or PC/XT I/O adapters and memory cards. The command cycle is terminated by detecting OWS* or IOCHRDY* active.

I/O Channel Speed Control

The bus controller AT bus logic can be programmed to insert wait states in units of ATSCLK and to delay the generation of the XIOW*, XIMR*, and XMEMW* commands in one-half units of ATSCLK within the selected wait states. The command phase delay can be selectively defined for I/O cycles and for 8-, 16-, or 32-bit wide memory cycles by setting the corresponding fields in the Port B register location 05H. This location controls the I/O channel wait state generation for 8-, 16-, and 32-bit accesses.

Data Conversion

The bus controller performs data conversions for CPU accesses to devices not on the local bus when AF32* is not asserted. AT bus conversions are performed for the following types of transfers:

- 32-bit to 8-/16-bit
- 24-bit to 8-/16-bit
- 16-bit to 8-/16-bit

Larger transfers are broken into smaller AT bus reads or writes and the action codes A0-A3 to the control buffer are generated. Byte Addresses XA0-XA1 are generated to drive the lower two bits of the AT bus.

The bus controller responds to IOCS16*, MCS16*, IOCS32*, and MCS32* to determine what size of data the I/O channel needs. If none of the above signals is asserted, 8-bit transfers are assumed and the request is converted into two, three, or four I/O channel cycles, based on BE0-BE3. For either MCS16* or IOCS16*, the bus controller converts a 32-bit access into two 16-bit AT bus accesses.

The bus controller also supports 32-bit transfers between the processor, memory, and the I/O devices on the I/O channel. IOCS32* and MCS32* inputs allow a device to request a 32-bit transfer. It is assumed that the necessary extensions to the AT bus are made to utilize this feature. IOCS32* and MCS32* override IOCS16* and MCS16*.

In performing these data conversions, the bus controller generates a 4-bit action code AC0-AC3 to control the buffers in the 82A305 data buffer. This is done for the alignment of data path and direction control between the D, MD, and SD data buses. The functional description for these codes is given in the section on "82A305 Data Buffers."

Port B Register

The bus controller contains an AT-compatible configuration register called the Port B register. The following table gives the bit definitions for it:

Port	Bit	Default	Type	Function
61H	7	0	Read Only	System Parity Check
	6	0	Read Only	I/O Channel Check
	5	0	Read Only	Timer 2 Out
	4	0 or 1	Read Only	Refresh Detect
	3	1	Read/Write	Enable I/O Channel Check
	2	1	Read/Write	Enable Parity Check
	1	0	Read/Write	Speaker Data
	0	0	Read/Write	Timer 2 Gate Speaker

Table 2. Bit Definitions for Port B Register.

The bus controller can also be programmed to vary the number of wait states and command delays to the system bus as well as to change the bus clock speed. This is done by writing the address of the Index Register to alter to the Index Register port (22H) and then writing or reading the desired value to or from the Data Register port (23H). Tables 3 - 5 provide the definitions of the Programmable Index Registers:

Note: Use a JMP\$+2 instruction between I/O port accesses to allow bus settle time.

Index 04H		Version/Processor Clock Select/NMI Source
Bit(s)	Function	Default(s)
7,6	Version (0=initial version)	0,0
5	Reserved	0
4	Processor Clock Select 0 = Use processor oscillator input 1 = Use AT bus state machine clock (SYSCLKx2)	0
3	Enable/Disable power fail NMI 0 = Disable 1 = Enable	0
2	Enable/Disable READY timeout NMI 0 = Disable 1 = Enable	0
1	Enable/Disable power fail warning pin 0 = Disable 1 = Enable	0
0	READY timeout has/not occurred 0 = Ready timeout has occurred 1 = Ready timeout has not occurred	0

Table 3. Index 04H Bit Definitions.

Index 05H**Command Delay**

The value for each of the command delay fields is as follows:

0	=	0-cycle delay
1	=	1-cycle delay
2	=	2-cycle delay
3	=	3-cycle delay

Bit(s)	Function	Default(s)
7,6	AT bus 32-bit memory command delay	0,0
5,4	AT bus 16-bit memory command delay	0,0
3,2	AT bus 8-bit memory command delay	0,1
1,0	AT bus I/O cycle command delay	0,1

Index 06H**Wait State/Bus Clock Source**

Bit(s)	Function	Default(s)
7,6	Wait states per 32-bit transfer 0 = 3-cycle delay 1 = 2-cycle delay 2 = 1-cycle delay 3 = 0-cycle delay	0,1
5,4	Wait states per 16-bit transfer 0 = 3-cycle delay 1 = 2-cycle delay 2 = 1-cycle delay 3 = 0-cycle delay	0,1
3,2	Wait states per 8-bit transfer 0 = 5-cycle delay 1 = 4-cycle delay 2 = 3-cycle delay 3 = 2-cycle delay	0,1
1,0	Bus clock source select 0 = Use processor clock/3 for AT bus state machine (5.33MHz) 1 = Use processor clock/2 for AT bus state machine (8 MHz) 2 = Reserved 3 = Use ATCLK input pin for AT bus state machine (6 MHz)	0,1

Table 5. Index 06H Bit Definitions.**82A306 Control Buffer**

The control buffer contains the circuitry for decode and generation of the AF32* signal, parity generation, parity checking, byte enable latching of data for the memory cycle, and color reference signal generation.

14MHz Oscillator and Divider

The color reference oscillator is provided by the bus controller IC (U13), thus eliminating the 8284 type device normally used in AT-compatible systems. A divide by 12 counter is also included to generate the OSC/12 (1.19MHz) signal used on the main logic board. The raw clock is supplied by crystal Y2.

AF32* Generation

The AF32* signal generated by the 82A306 is not used in the Tandy® 4000 system.

Byte Enable Latch

The register that holds the byte enables valid during a memory cycle is located in the control buffer. An additional input (FBE*) is provided to force all byte enables active during certain memory operations.

Parity Checking and Generation

The control buffer provides the necessary exclusive OR'ing to generate full (byte) write and read parity from the partial parity bits PPH0-PPH3 and PPL0-PPL3, which are generated on the two (nibble wide) 82A305 data buffers (U23, U24).

For a memory read access, read parity bits PPH0-PPH3 and PPL0-PPL3 are checked against the parity bits MP0-MP3 read from memory. Parity bits PPH0-PPH3, PPL0-PPL3, and MP0-MP3 are latched by CAS* and PCHK* so that they are kept valid during parity checking. The results of the byte-wise comparison are further gated by byte enables to ignore errors for bytes that are not valid. The OR'ed byte-wise parity error is then latched as the output LPAR* if PEN* input is asserted.

During a memory write access, write parity for each byte is generated from PPH0-PPH3 and PPL0-PPL3 and can be gated onto the memory parity bus MP0-MP3 if the control buffer is enabled by the WPE* signal controlling the tri-state drivers.

Bus Drivers

24mA bus drivers are provided for some of the control signals on the I/O channel. These signals include SYSCLK, OSC, OSC/12, RESETDRV, SBHE*, BALE, IOR*, IOW*, MEMR*, MEMW*, SMEMR*, SMEMW*, and OUTL*. The functionality for these signals is covered under "I/O Decode" and "Memory Control and Refresh."

MEMORY CONTROL AND REFRESH

82C302 Memory Controller

Memory Configuration

The memory control functions in this system utilize Page Mode Access DRAMs. Memory is organized as 36-bit banks (32 bits for data and 4 bits for parity) using four 9-bit SIMM boards per bank. The system will support up to 16Mb of 32-bit DRAMs using 1Mb x 9 modules.

The minimum configuration is a single bank operating in Non-Interleaved Mode.

The memory controller is designed such that the memory can be upgraded from one to two banks, making it a two-way interleaved organization. Because of the interleaved page operation, the third and fourth banks must be added as a pair. Furthermore, the DRAM types must be identical in each bank of a pair. However, each pair of banks can use different DRAM types. One or both banks of smaller DRAM types can later be upgraded to banks of larger DRAMs.

The following table shows the available configurations:

Number of Banks	Number/Size of SIMMs	Total Memory
1	4 - 256Kb x 9	1 Mb
1	4 - 1Mb x 9	4 Mb
2	8 - 256Kb x 9	2 Mb
2	8 - 1Mb x 9	8 Mb
4	16 - 256Kb x 9	4 Mb
4	8 - 256Kb x 9	
	8 - 1Mb x 9	10 Mb
4	16 - 1Mb x 9	16 Mb

Table 6. Available Memory Configurations.

Page-Interleaved Operation

The memory controller uses a page-interleaved design that is different from most interleaved memory designs. Normal two-way interleaving uses two banks of DRAMs with even (double word) addresses stored in one bank and odd addresses stored in the other. If accesses are sequential (or at least made to alternating even and odd addresses) the RAS precharge time of one set might overlap the access time of the second set. Typically the hit rate (the fraction of times that the required bank is available) is 50%. This is especially true since operand accesses can be interspersed with instruction fetches.

Page Mode operation available with most DRAMs operates because the access to the row address of the internal DRAM array makes available a large number of bits (512 bits in a 256Kb X 1 DRAM) that are subsequently selected using the column address. Once a row address has been made, higher-speed random access can be made to any bit within the row.

Page Mode access and cycle times are typically half those of the normal access and cycle times. During Page Mode operation any access to the currently active RAS page would occur in the page access rather than the normal access time. Any subsequent access could be to anywhere in the same page without incurring any penalty due to RAS precharge.

When memory is configured to take advantage of this DRAM organization, significantly better performance can be achieved over normal interleaving.

Memory Access and Arbitration

The memory controller controls the DRAM memory access from three sources: the CPU, the DMA, and the refresh request. These accesses are arbitrated based on the inputs of HLDA1* and REF*.

CPU Access

The CPU-initiated accesses are decoded according to the memory map defined in the Configuration Registers. These registers are at 08H to 0FH and are covered later in this section. These are the only accesses that use the Page Mode operation of the DRAMs.

The memory controller maintains four page registers, in which it stores the page addresses of the most recently accessed DRAM pages of the two-way interleaved banks. These four registers are called Active Page Registers. Accesses to the active pages are called hits and are faster because the DRAM is operated in the Page Mode with the RAS signal staying asserted.

The memory controller supports memory configurations with either one, two, or four banks. Because one Active Page Register is provided for each bank, the number of active pages varies with the amount of memory installed. In a non-interleaved minimum memory configuration, only one Active Page Register is in use. For each Active Page Register in use, the corresponding RAS* stays asserted after the previous access. If an access does not hit any active pages, a miss cycle, the normal DRAM access cycle is entered by first de-asserting the RAS* associated with the bank accessed.

RAS and CAS Generation

The memory controller is based on 2Kb page-interleaved organization. The following table shows the address lines used for the different organizations:

For Non-Interleaved Operation (One Bank Only)

<u>DRAM Type</u>	<u>Row</u>	<u>Column</u>
256Kb DRAMs	A11-A19	A2-A10
1Mb DRAMs	A12-A21	A2-A11

For Interleaved Memory (Two or Four Banks)

<u>DRAM Type</u>	<u>Row</u>	<u>Column</u>
256Kb DRAMs	A12-A20	A2-A10
1Mb DRAMs	A12-A21	A22, A2-A10

Table 7.
Address Lines for Available Memory Configurations.

In interleaved memory cases, Bit All determines which one of the even page banks or odd page banks is accessed. For configurations using only 256Kb DRAMs, All and A21 are used to control RAS0*-RAS3*. For those using only 1Mb DRAMs, All and A23 are used. For those using both 256Kb and 1Mb DRAMs, the 1Mb DRAMs must occupy the first two banks and the 256Kb DRAMs must occupy the second two banks. This constraint is there to ensure that there will not be a hole in the address space without actual DRAMs.

RAS Timeout

When using DRAM Page Mode, you must observe the maximum RAS pulse width. For most DRAMs, this is 10 microseconds. Using the OSC/12 (1.19MHz) clock generated by the control buffer timers are maintained within the memory controller for each bank to assure data integrity . RAS is de-asserted for each bank when its counter times out at about 10 microsecond intervals. The Memory Configuration Register bit 13H Bit 7 can be programmed to set the desired RAS time-out interval for each bank. See Table 13, later in this section, for details.

Refresh

To reduce power supply noise caused by surges during RAS transitions, RAS pulses to each bank are automatically staggered by one CLK2 cycle. Because all RAS's could be active for Page Mode operation, a refresh cycle requires that all RAS's be first de-asserted then asserted with the correct refresh address.

Direct-Memory Access

Direct-memory accesses are initiated by asserting the signal HLDAL. The XMEMR* and XMEMW* signals determine whether the access is a read or write memory access. The bytes accessed are controlled externally with the BE0*-BE3* signals generated by the bus controller. The memory controller makes one memory access per DMA bus cycle and does not attempt to pack or unpack data transfers to make full 32-bit transfers.

DRAM Access Logic

The DRAM control logic generates the necessary RAS, CAS, and DWE signals for all DRAM accesses. CPU, DMA, and refresh access cycles use DA8-DA9, XDA0-XDA7, and bank select signals BA0-BAL. (Note that in the current version of the memory controller, the signals BA0-BAL default to zero.) The memory controller control logic provides the signal MDEN* to control the buffer chips. MDEN* enables data buffers for the MD bus for non-refresh DRAM cycles.

NOTE: "Default Configuration" refers to IC power up. The BIOS will reprogram these values.

Memory Mapping Logic

The memory controller Configuration Registers, located at Port 22H Index 08H-13H define what is a valid memory access and what is a ROM memory access, according to the local bus addresses. Registers 08H and 09H determine how ROM areas (as defined by an IBM AT) between the 768Kb to 1Mb address range are accessed.

For valid local memory accesses, the memory controller asserts the AF32* signal to indicate that it has control of the local bus and also asserts the READY* signal at the end of the access cycle. If an access is a ROM access, the memory controller asserts ROMCS* to provide controls for the ROMs. In this case, the READY* signal must be provided to the CPU and memory controller by the bus controller.

Clock, Reset, and Miscellaneous Logic

The RESET4 signal causes all internal registers to be reset to their default values. Configuration Registers not specified with default values are not reinitialized and might not retain their old values. The memory controller control logic generates the PCHK* and PEN* signals to be used for enabling parity error checking.

Configuration/Diagnostic Registers

The memory controller contains Index Registers used to configure the memory. These are accessed like the registers in the bus controller. Write the index address to Port 22H and then the programming value out to the Data Port 23H. XDEN* is asserted for these accesses to control the buffer connecting the XD and XDA buses. The Data Port can also be read.

Memory Configuration Registers

The Configuration Registers 08H to 0FH are used to control how the CPU memory accesses are defined. They define all addresses as ROM accesses, DRAM accesses, other local CPU bus accesses, or I/O channel accesses. These provisions are made because the low 1 megabyte is occupied by both DRAMs and ROMs and also devices on the AT bus. For ROM accesses, the memory controller generates the ROMCS* signal to control the ROM access. For DRAM access, the memory controller generates the necessary DRAM controls to the system memory under its control. It generates AF32* for all other local CPU bus accesses, and it does not control the I/O channel accesses.

The bus controller (U14) provides three 256Kb areas where the ROMs can be located. The functions of these areas are as follows:

Area	Location	Function
Low ROM	Just below the 1 megabyte address	8086-compatible operation
Middle ROM	Below the 16 megabyte address	80286-compatible operation
High ROM	Below the 4 gigabyte address	80386 operation

Table 8. Functions of ROM Areas.

On system reset, the default Configuration Register setting causes accesses to these three ROM areas to generate ROMCS*. The high ROM area is always recognized by the memory controller as BIOS ROM access area. The other two ROM areas can be mapped to be either ROM or RAM access.

After reset, Register 08H Bits 3 and 4, (See Table 9), can be programmed to map the entire middle ROM area to DRAM with write protection if desired.

Register 08H Bit 2 determines whether the bus controller recognizes the addresses generated beyond 16 megabytes as local CPU bus cycles.

Register 08H Bit 1 is used to enable Registers 0AH-0FH, which control the low megabyte DRAM (40000H-FFFFFH) address mapping for 256Kb to 1Mb addresses in 16Kb blocks. This bit defaults upon reset so that only the 0-256Kb areas are accessible. Accesses to the low megabyte DRAM can be made by enabling the mapping after the necessary configuration registers are correctly programmed.

Register 08H Bit 0 defaults to single-bank memory configuration upon reset. Therefore it must be programmed to enable page/interleaved operation.

Register 09H controls the address mapping and write protection for the low ROM area (C0000H-FFFFFH) in 64Kb blocks (See Table 10.) Registers 0AH-0FH define whether each 16Kb address range is a DRAM block in the system memory or on the I/O channel. (See Table 11.)

DRAM Array Configuration and Timing

The Configuration Registers 10H-13H provide the DRAM type definition and starting address for each pair of banks, Banks 0 and 1 and Banks 2 and 3.

Register 10H Bits 6 and 7 and Register 12H Bits 6 and 7 define whether the DRAMs are enabled, and if so, whether the system uses 256Kb DRAMs or 1Mb DRAMs. (See Table 12.) These bits default to 256Kb DRAMs upon reset.

Register 10H Bits 0-5 and Register 12H Bits 0-5 define the Address Bits 20-25 of the starting address of the pairs of banks. (See Table 12.) Some of these banks might not be valid because the memory banks must start at some predefined boundaries. For 256Kb DRAMs, all six bits (20-25) are valid if a only single bank is enabled. The banks starting address can be on any 1 megabyte boundary. Otherwise, only Bits 21-25 are valid starting address bits on 2Mb boundaries. For 1Mb DRAMs, only bits 23-25 are valid, forcing the banks starting address to be on 8Mb boundaries.

Register 11H Bit 7 and Register 13H Bit 7 define the RAS precharge time required when a page miss occurs so that DRAMs of different speeds can be supported for each pair of banks. (See Table 13.)

Register 11H Bit 6 and Register 13H Bit 6 define the wait state to be inserted to meet the DRAM speed. (See Table 13.) These parameters default to the slower timing upon reset so that the system can be powered up with minimal assumptions about the DRAM speed and the memory configuration.

Tables 9-15 describe the functions of the memory controller's programmable registers:

Identification Register (08H)

Bit(s)	Function	Default Value
7	Controller type	0
6-5	Version 0 = initial	0,0
4	Read/Write or Read Only of 256Kb RAM at 16128K 00FC000H 0 = read/write 1 = read only	0
3	Disable/Enable BIOS ROM below 16Mb 0 = disable 1 = enable	0
2	Assert/Do not assert AF32* for addresses above 16Mb 0 = do not assert 1 = assert	0
1	Use/Ignore memory configuration Registers 0AH-0FH 0 = ignore 1 = use	1
0	Disable/Enable interleave 0 = disable (use single bank) 1 = enable	0(*)

- * 0 for one bank of 32-bit memory (1 megabyte)
1 for two banks of 32-bit memory (2 megabytes or more)

Table 9. Register 08H.

RAM/ROM Boot Area Configuration Register (09H)

Bit(s)	Function	Default Value
7-4	Disable/Enable writing to RAM located in BIOS 0 = read/write 1 = read only	0
	Bit 7 (RAM at 768K C0000-CFFFF)	0,1*
	Bit 6 (RAM at 832K D0000-DFFFF)	0,1*
	Bit 5 (RAM at 896K E0000-EFFFFH)	0,1*
	Bit 4 (RAM at 960K F0000-FFFFFH)	0,1*
3-0	Disable/Enable substituting BIOS ROM below 1 megabyte with RAM at the same address 0 = disable 1 = enable	1
	Bit 3 (ROM at 768K C0000-CFFFF)	0,0*
	Bit 2 (ROM at 832K D0000-DFFFF)	0,0*
	Bit 1 (ROM at 896K E0000-EFFFFH)	0,0*
	Bit 0 (ROM at 960K F0000-FFFFFH)	1,0*

* Default values change after ROM is initialized (ROM code is moved into DRAM.) The first value is the default before initialization. The second is the default after initialization.

Table 10. Register 09H.

Selective Enabling/Disabling of 16Kb Blocks of RAM
(Registers 0AH,0BH, 0CH, 0DH, 0EH, and 0FH)

0 = System board controls address
1 = Address is on the I/O channel

Register	Block (Address Map)	Default Value
0AH	040000-05FFFFH	00H
0BH	060000-07FFFFH	00H
0CH	080000-08FFFFH	00H
0DH	0A0000-0BFFFFH	FFH*
0EH	0C0000-0DFFFFH	FFH*
0FH	0E0000-0FFFFFH	FFH*

* Default depends on options in I/O channel and use of expansion memory.

Table 11. Registers 0AH-0FH.

Bank Type/Starting Address
(10H for Banks 0/1, 12H for Banks 2/3)

Bit(s)	Function	Default Value
7-6	Bank enabled/disabled 0 = bank disabled 1 = bank enabled with 256Kb words (default) 2 = bank enabled with 1Mb words 3 = reserved	40H for 10H; varies for 12H*
5-0	Starting address bits 25-20 256Kb DRAMs 1Mb per bank (first bank only) 25-21 256Kb DRAMs 1Mb per bank (two banks required) 25-23 1Mb DRAMs 4Mb per bank (two banks required)	

* Index 12H default equals 00 if there are no Banks 2/3. It equals 42H for Banks 2/3 if 256Kb DRAMs are used.

Table 12. Registers 10H and 12H.

Bank Timing
(11H for Banks 0/1, 13H for Banks 2/3)

Bit(s)	Function	Default Value
7	RAS precharge time for a page miss 0 = 3 CLK2 times (93 ns) 1 = 5 CLK2 times (155 ns)	00H, 00H
6	Access wait states 0 = 0 wait states 1 = 1 wait state	
5-0	Reserved	

Table 13. Registers 11H and 13H.

Error Source/Address (MSBs)
(Register 28H)

Bit(s)	Function	Default Value
7	Enable/Disable parity check 0 = enable 1 = disable	0
6-2	Not Used	
1-0	High parity address bits (25,24) (read only)	

Table 14. Register 28H.

Parity Error Address (LSBs)
(Register 29H)

Bit(s)	Function	Default Value
7-0	Error address bits (23-16) (read only)	

Table 15. Register 29H.

Memory Expansion

The main logic board contains sockets for eight SIMM boards (Banks 0 and 1) of DRAMs. Using 1Mb x 9 SIMMs, this gives 8 megabytes of on-board memory. If more memory is desired, a 32-bit memory expansion card is available that contains eight more SIMM boards (Banks 2 and 3). Because of the interleaved nature of the memory, both Banks 2 and 3 must be used with this card. Using this option can expand the total system memory to a total of 16 megabytes. This card is plugged in the Number 1 expansion slot. It is incompatible with standard AT-type option cards.

ADDRESS BUFFERS

82A303 High Address Buffer (U20)

Address Decode

The address decode circuit provides as outputs the signals LIOCS*, LMEGCS*, L64MEG*, and HIROM*. These signals are active if the address accesses satisfy the conditions defined in Table 16. The signal decodes for LIOCS* and LMEGCS* are controlled by HLDAL and are latched onto the trailing edge of MALE*. The L64MEG* and HIROM* signals are simply decoded from the address signals.

Signal	Decode Condition
LIOCS*	A12-A15 = 00H
LMEGCS*	A20-A31 = 00H
L64MEG*	A26-A31 = 00H
HIROM*	A26-A31 = 3FH

Table 16. Address Access Conditions.

Address Bus Interfaces

The 82A303 interconnects the local X and system address buses. Bidirectional drivers connect each bus and the internal buses. These drivers have 24 mA current drivers for direct connection to the system address bus. Table 17 shows how the drivers are configured between the buses for each type of active bus request. Note that the default configuration is set up so that the CPU address bus drives the memory address bus for local memory CPU access cycles.

For all CPU-sourced accesses, the addresses are latched on the trailing edge of MALE*.

Active Signal	Source Bus	Target Bus
HLDAL	XA	SA, A, MA
MASTER*	SA	XA, A, MA
REF*	--	SA driven low
ATEN	A	XA, SA, MA
default	A	MA

Table 17. Bus Direction Request Types.

27-Bit Extensions

The standard AT implementation supports only 24-bit addresses. The system architecture allows for address extension on the SA and XA buses to 27 bits (128Mb). This is done by grounding the enable pin XBHE for the XA bus and the enable pin SBHE for the SA bus. Internal pullups are provided so that if the enable pins are left unconnected Bits 24-27 of the respective bus are forced low.

82A304 LOW ADDRESS BUFFER (U21)

Address Decode

The signals IO2XCS*, 8042CS*, PORTBCS*, NMICS*, 287CS*, and AS provide the lower address decodes for the corresponding devices after being qualified by the LIOCS* signal generated by the high address buffer decoder. The resulting decode is as defined by the IBM PC AT I/O addresses and is shown in Table 18. For applications that require these devices to be relocated, the EXDEC signal can be tied low to ignore the LIOCS* qualification and the MA10-MA11 address bits.

Signal	Address Decoded
IO2XCS*	022H, 023H
8042CS*	060H, 064H
PORTBCS*	061H
NMICS*	070H
287CS*	0E0H to OFFH

Table 18. Address Decode.

Address Bus Interfaces

The 82A304 interfaces between bits 00 to 11 of A, SA, XA, and MA address buses. The buffers and multiplexers are controlled by the HLDAL, MASTER*, REF*, and ATEN* signals to drive the signals from the source to the target buses as defined in Table 19 for each active signal. When REF* is asserted, the refresh counter is gated to the SA bus as the refresh row address and is incremented. When none of the listed signals is active, the default configuration is such that the A bus drives the MA bus for memory accesses by the CPU.

The SA00-SAll signals are 24mA address buffers for direct interface to the AT bus.

Active Signal	Source Bus	Target Bus
HLDAL	XA	SA, MA, A
MASTER*	SA	XA, MA, A
REF*	Counter	SA
ATEN*	A2-A11	SA2-SAll, XA2-XAll, MA4-MA11
	XA0-XA1	SA0-SAll
default	A	MA

Table 19. Bus Direction Request Types.

DATA BUFFERS AND CONVERSION LOGIC

82A305 Data Buffers (U23, U24)

The data buffers interface between the local, memory, and system (I/O channel) data buses and provide data alignment and size conversion for AT I/O channel operations. It is designed as a nibble slice to reduce pin count. Two of the devices are incorporated into the system design and are used to interface all buses.

Bus Controls

The data buffers control the bus buffers according to the signals HLDAl, ATEN*, MDEN*, LDEN*, SDIR, MRD*, and AC0-AC3. The first group of signals (HLDAl, ATEN*, MDEN*, and LDEN*) determine which buses are connected, and the second group of signals (SDIR, MRD*, and AC0-AC3) determines the direction of the buffers' drivers.

All drivers are active for the active buses, and external bus controls are required if selected data bits need to be controlled. For the DRAM interface, the LBE0*-LBE3* signals must be used to ensure that only the valid data bytes are written to the DRAMs during a write cycle. Table 20 outlines bus connections for different bus cycles.

Bus Cycles	From Bus	To Bus	Direction Control
HLDAl = 0 ATEN* = 1	D MD	MD D	MRD* = 1 MRD* = 0
HLDAl = 0 ATEN* = 0	D SD	SD D	SDIR = 1 SDIR = 0
HLDAl = 1	SD	MD,D	SDIR = 0 MRD* = 1
	MD	SD	SDIR = 1 MRD* = 0
	D	SD	SDIR = 1 MRD* = 1 LDEN* = 0

Table 20. Bus Control Definitions.

Data Conversion

The data buffers provide the data bus connections so that data conversions are done correctly for CPU accesses to the AT bus. The action code signals AC0-AC3 are used to control how the bus bits are connected between the I/O channel SD bus and the CPU local Bus D or the system memory MD bus. The action code signals are provided by the 82C301 bus controller for CPU-to-AT bus access cycles and are qualified by the ACEN signal. The meanings of the action code signals are given in Table 21.

AC0-AC3	From Bus	To Bus
0	MD,D Bits 0-15	SD Bits 0-15
1	MD,D Bits 8-15	SD Bits 8-15
		SD Bits 0-7
2	MD,D Bits 16-31	SD Bits 0-15
3	MD,D Bits 24-31	SD Bits 8-15
		SD Bits 0-7
4	MD,D Bits 0-31	SD Bits 0-31
5	SD Bits 0-7	MD,D Bits 0-7
6	SD Bits 0-7	MD,D Bits 8-15
7	SD Bits 0-7	MD,D Bits 16-23
8	SD Bits 0-7	MD,D Bits 24-31
9	SD Bits 0-15	MD,D Bits 0-15
A	SD Bits 0-15	MD,D Bits 16-31
B	Reserved	
C	SD Bits 0-31	MD,D Bits 0-31
D	Reserved	
E	Reserved	
F	Reserved	

Table 21. Action Code Definitions.

System I/O (AT) Bus

The Tandy 4000's system I/O (AT) bus is 100% compatible with the IBM AT bus. It has an open-bus structure that allows multiple microprocessors to share system resources. A total of eight expansion slots are available (six AT- and two PC type). Some features of this bus are:

- 24-bit memory addressing
- 8- or 16-bit data accesses
- I/O addressing range of 100 to 3FF hex
- Interrupt and DMA support
- I/O wait state generation

ROM Subsystem

The on-board ROM consists of two EPROM modules, each capable of holding up to 32 kilobytes of data, or 64 kilobytes total. One module contains the even address BIOS code. The other contains the odd address code.

I/O DECODE

82C206 INTEGRATED PERIPHERALS CONTROLLER (IPC)

The 82C206 (U8) is an LSI implementation of the standard peripherals required to implement an IBM AT system board. This device contains the equivalent of:

- Two 8237A DMA controllers
- A 74LS612 mapper
- Two 8259A interrupt controllers
- An 8254 counter/timer
- An MC146818 real-time clock with RAM

The IPC provides all the standard peripherals required for a system board implementation, except the keyboard interface controller.

DMA Controllers. The two DMA controllers are connected in such a way as to provide 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.) The BIOS programs the DMA controllers at power up as follows:

```
Master Clear issued to both controllers
All eight channels are first set to -
    Single mode select
    Verify transfer
    Autoinitialization disabled
    Address increment select
Then channel 4 is set to cascade mode and
enabled.
```

Mapper. Included as part of the DMA subsystem is the Page Register (DMAPAGE) device, which is used to supplement the DMA and drive the upper address lines when required.

Interrupt Controllers. The IPC provides 16 channels of interrupt, partitioned into two cascaded controllers (INTC1 and INTC2) with eight inputs each. Three of the sixteen channels are connected internally to various devices. The remaining 13 interrupt channels are user-definable. The BIOS programs the interrupt controllers at power up as follows:

Slave Controller - First Interrupt at 70h

Call address interval = 4 bytes
Slave ID = 2
8086/8088 Mode
Normal EOI
Non-buffered mode
Not special fully nested mode
Cascade mode
Edge triggered mode

Master Controller - First interrupt at 08h

Call address interval = 4 bytes
Slave on IRQ2
8086/8088 mode
Normal EOI
Non-buffered mode
Not special fully nested mode
Cascade mode
Edge triggered mode

The Pre-unmasked IRQs at boot time are:

IRQ0 Timer Tick Clock
IRQ1 Keyboard
IRQ2 Cascade for Slave
IRQ6 Floppy Disk
IRQ9 Software redirected to INT 0Ah for old IRQ2
IRQ14 Hard Disk (If Present)

The three internally connected channels are:

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

Counter/Timer. The counter/timer (CTC) subsystem contains three independent counters: Counters 0, 1, and 2. All three counters are driven from a clock input pin (TMRCLK) that is independent from the other clock inputs to the device.

Counter 0 is connected to Interrupt 0 of INTC1. It is intended to be used as a multi-level interrupt to the system for such tasks as timekeeping and task-switching.

Counter 1 can be programmed to generate pulses or square waves for use by external devices.

Counter 2 is a full-function counter/timer that has a gate input for controlling the internal counter. This channel can be used as an interval counter, a timer, or a gated rate/pulse generator.

Real-Time Clock. The real-time clock (RTC) subsystem contains a clock/calendar to maintain the real time and date. In addition, it contains 114 bytes of RAM. To keep the clock/calendar and RAM active when the system is turned off, you must connect the device to an external battery. The Tandy 4000 uses a lithium battery attached to the drive tower for this purpose.

Control of Subsystems

Interconnection and control of all the major subsystems is accomplished via a top-level control section. For discussion purposes, this control section is divided into two subsections:

- **Clock and Wait State Control Section**--controls the generation of DMA wait states and the negation of IOCHRDY (if programmed to do so) during CPU access of the device.
- **Top-Level Decoder**--accommodates the 200-plus registers in the IPC and maintains I/O decode compatibility with the IBM AT. The top-level decoder generates enables to the various subsystems. This subsystem also handles control and direction of the XD0-XD7 data buffers.

Top-Level Decoder

The IPC top-level decoder provides eight separate enables to various internal subsystems of the device. The following is a truth table for the top-level decoder:

	ACK*	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	X	000-00F	DMA1
1	0	0	0	0	1	0	0	0	0	0	X	020-021	INTC1
1	0	0	0	0	1	0	0	0	0	1	X	022-023	CONFIG
1	0	0	0	1	0	0	0	0	0	0	X	040-043	CTC
1	0	0	0	1	1	1	0	0	0	0	1	071	RTC
1	0	0	1	0	0	0	X	X	X	X	X	080-08F	DMAPAGE
1	0	0	1	0	1	0	0	0	0	0	X	0A0-0A1	INTC2
1	0	0	1	1	0	X	X	X	X	X	X	0C0-0DF	DMA2
0	X	X	X	X	X	X	X	X	X	X	X		Disabled
X	1	X	X	X	X	X	X	X	X	X	X		Disabled
X	X	1	X	X	X	X	X	X	X	X	X		Disabled

Table 22. Truth Table for Top-Level Decoder.

The enabling of the IPC XD0-XD7 output buffers is also controlled by the top level decoder. The output buffers are enabled whenever an enable is generated to an internal subsystem and the XIOR* signal is asserted.

The decoder is enabled by three signals: ACK*, XA9, and XA8. To enable any internal device, ACK* must be "1" and both XA8 and XA9 must be "0."

The decode scheme employed in the IPC is designed to comply with the IBM AT requirements. If you wish to take advantage of the areas that are unused by inserting additional peripherals in the I/O map, you can do so. The IPC does not respond to the unused address spaces established by the top-level decoder. The extra peripherals can be tied directly to the XD0-XD7 data lines because the IPC output buffers are not enabled unless an internal subsystem is enabled.

Clock and Wait State Control

The clock and wait state control subsystem performs three functions: control of the DMA command length, control of the CPU read or write cycle length, and selection of the DMA clock rate. All these functions are user-selectable by writing to the Configuration Register located at Address 23H.

Writing and reading this register is accomplished by first writing a 01H to Location 22H to select the IPC Configuration Register and then performing either a read or write to Location 23H. The following is a layout of Configuration Register 23H Index 01H. Explanation of the individual bits follows.

msb	b6	b5	b4	b3	b2	b1	lsb b0
RW1	RW0	16W1	16W0	8W1	8W0	EMR	CLK

RW1-RW0 - When the higher-speed CPUs are accessing the IPC, 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 (LOW) on the signal IOCHRDY whenever a valid decode from the top-level decoder is detected and either signal XIOR* or signal XIOW* is asserted. IOCHRDY will remain low for the number of wait states programmed into Bits 6 and 7 of the Configuration Register.

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

Table 23.
Configuration Register Bits 7 and 6.

Wait states are in increments of one SCLK cycle and are not affected by the DMA clock divider.

16W1-16W0 - Wait states can be independently controlled for both 8-bit and 16-bit DMA cycles, using Bits 4 and 5. This allows you 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

Table 24.
Configuration Register Bits 5 and 4.

8W1-8W0 - Wait states can be inserted in 8-bit DMA cycles by programming Bits 2 and 3 in the Configuration Register.

8W1 8W0 8-Bit DMA Wait States

0	0	1
0	1	2
1	0	3
1	1	4

Table 25.
Configuration Register Bits 3 and 2.

Further control of the cycle length is available through the use of the IOCHRDY pin on the IPC. During DMA, this pin is used as an input to the wait state generation logic to extend the cycle if necessary. To extend the cycle, the peripheral drives the input low (0). The cycle can then be completed by releasing IOCHRDY and allowing it to return high (1).

EMR - The EMR bit, Bit 1, enables the extended DMA read function. Normally the assertion of DMAMEMR* is delayed one clock cycle later than XIOR*, (I/O bus read), in the IBM AT implementation. This might not be desirable in some systems. Programming a "1" into Bit 1 starts DMAMEMR* at the same time as XIOR*.

CLK - The CLK bit, Bit 0, allows you to insert a divider between the DMA controller subsystems and the SCLK input pin or to connect the two directly. When Bit 0 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 Bit 0 bypasses the divider and causes the IPC to use the SCLK input directly. Whenever the state of Bit 0 is changed, an internal synchronizer controls the actual switching of the clock to prevent a short clock pulse from causing a DMA malfunction.

Configuration Register Defaults

The Configuration Register contents are pre-loaded by reset to an initial value of 0COH. This value establishes a default that is IBM AT-compatible. The BIOS then programs a value of 040H upon initialization and corresponds to:

Read/Write Cycles	2 wait states
16-Bit DMA Transfers	1 wait state
8-Bit DMA Transfers	1 wait state

DMAMEMR* is delayed one clock cycle later than XIOR*. DMA clock is equal to SCLK/2.

Description of DMA Controller Functions

As previously stated, the equivalent of two 8237A DMA controllers is implemented in the IPC. Each controller is a four-channel DMA device that generates the memory addresses and control signal 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 of the two DMA devices, thereby maintaining IBM AT compatibility.

DMA cycle length control is provided internally in the IPC, 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 subsystem to transfer blocks of as many as 65536 words. The register associated with each counter allows the channel to reinitialize without re-programming.

From this point on, the description of the DMA subsystem pertains to both DMA1 and DMA2 unless otherwise specified.

DMA Operation

During normal operation of the IPC, the DMA subsystem will be in one of three conditions:

- Idle
- Program
- Active

In the idle condition, the DMA controller will be executing cycles of only one state. The idle state S1 is the default condition, and the DMA controller remains in this condition unless the device is initialized and one of the DMA requests becomes 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 IPC generates the necessary memory addresses and command signals to accomplish a memory-to-I/O, I/O-to-memory, or memory-to-memory transfer.

I/O-to-memory and memory-to-I/O transfers take place in one cycle. 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 so the transfer is completed in one cycle. Memory-to-memory transfers, on the other hand, require the DMA subsystem to store data from the read operation in an internal register. The contents of this register are then written to memory on the subsequent cycle.

During transfers between memory and I/O, two commands are activated during the same cycle. In a memory-to-I/O transfer, the IPC asserts both DMAMEMR* (DMA memory read) and XIOW* (I/O bus write), allowing data to be transferred directly to the requesting device from memory. Note that the IPC does not latch data from nor drive data out on this type of cycle.

The number of clock cycles required to transfer a word of data can be varied by programming the DMA subsystem or, optionally extended by the peripheral device. During an active cycle, the DMA subsystem will sequence through a series of states. Each state is one DMA clock cycle long, and the number of states in a cycle varies depending on how the device is programmed and what type of cycle is being performed. The states are labeled S0-S4. (See "Active Condition," later in this section, for further explanation of these states.)

Idle Condition

When no device is requesting service, the DMA subsystem is in an idle condition. During this time, the IPC will sample the DREQ input pins every clock cycle. The internal select from the top-level decoder and HLDA are also sampled at the same time to determine whether the CPU is attempting to access the internal registers. When either of the above situations occurs, the DMA subsystem exits the idle condition. Note that the program condition has priority over the active condition because a CPU cycle has already started.

Program Condition

The program condition is entered whenever HLDA is inactive and an internal select is active. The internal select is derived from the top-level decoder described previously. During this time, Address Lines XA0-XA3 become inputs if DMA1 is selected. If DMA2 is selected, XA1-XA4 become inputs. Note that when DMA2 is selected XA0 is ignored. The XA0-XA4 inputs are used to select the DMA controller registers that are to be read or written. The DMA controller register assignments are shown in Tables 26 and 27.

Because of the large number of internal registers in the DMA subsystem, an internal flip-flop is used to supplement the addressing of the Count and Address Registers. This bit is used to select between the high and low bytes of these registers. The flip-flop will toggle each time a read or write occurs to any of the Word Count or Address Registers in the DMA subsystem. This internal flip-flop is cleared by a reset or a Master Clear command and can be set or cleared by the CPU issuing the appropriate command.

DMA Controller #1**Flip-
Add. XIOR* XIOW* Flop Operation**

000h	0	1	0	Read CH0 Current Address low byte
000h	0	1	1	Read CH0 Current Address high byte
000h	1	0	0	Write CH0 Base & Current Address low byte
000h	1	0	1	Write CH0 Base & Current Address high byte
001h	0	1	0	Read CH0 Current Word Count low byte
001h	0	1	1	Read CH0 Current Word Count high byte
001h	1	0	0	Write CH0 Base & Current Word Count low byte
001h	1	0	1	Write CH0 Base & Current Word Count high byte
002h	0	1	0	Read CH1 Current Address low byte
002h	0	1	1	Read CH1 Current Address high byte
002h	1	0	0	Write CH1 Base & Current Address low byte
002h	1	0	1	Write CH1 Base & Current Address high byte
003h	0	1	0	Read CH1 Current Word Count low byte
003h	0	1	1	Read CH1 Current Word Count high byte
003h	1	0	0	Write CH1 Base & Current Word Count low byte
003h	1	0	1	Write CH1 Base & Current Word Count high byte
004h	0	1	0	Read CH2 Current Address low byte
004h	0	1	1	Read CH2 Current Address high byte
004h	1	0	0	Write CH2 Base & Current Address low byte
004h	1	0	1	Write CH2 Base & Current Address high byte
005h	0	1	0	Read CH2 Current Word Count low byte
005h	0	1	1	Read CH2 Current Word Count high byte
005h	1	0	0	Write CH2 Base & Current Word count low byte
005h	1	0	1	Write CH2 Base & Current Word count high byte
006h	0	1	0	Read CH3 Current Address low byte
006h	0	1	1	Read CH3 Current Address high byte
006h	1	0	0	Write CH3 Base & Current Address low byte
006h	1	0	1	Write CH3 Base & Current Address high byte
007h	0	1	0	Read CH3 Current Word Count low byte
007h	0	1	1	Read CH3 Current Word Count high byte
007h	1	0	0	Write CH3 Base & Current Word Count low byte
007h	1	0	1	Write CH3 Base & Current Word Count high byte

DMA Controller #1 Continued**Add. XIOR* XIOW* Flip-Flop Operation**

008h	0	1	X	Read Status Register
008h	1	0	X	Write Command Register
009h	0	1	X	Read DMA Request Register
009h	1	0	X	Write DMA Request Register
00Ah	0	1	X	Read Command Register
00Ah	1	0	X	Write single-bit DMA Request Mask Register
00Bh	0	1	X	Read Mode Register
00Bh	1	0	X	Write Mode Register
00Ch	0	1	X	Set byte pointer flip-flop
00Ch	1	0	X	Clear byte pointer flip-flop
00Dh	0	1	X	Read Temporary Register
00Dh	1	0	X	Master clear
00Eh	0	1	X	Clear Mode Register counter
00Eh	1	0	X	Clear all DMA Request Mask Register bits
00Fh	0	1	X	Read all DMA Request Mask Register bits
00Fh	1	0	X	Write all DMA Request Mask Register bits

Table 26. DMA Controller #1 Register Assignments.

DMA Controller #2**Flip-
Add. XIOR* XIOW* Flop Operation**

0C0h	0	1	0	Read CH4 Current Address low byte
0C0h	0	1	1	Read CH4 Current Address high byte
0C0h	1	0	0	Write CH4 Base & Current Address low byte
0C0h	1	0	1	Write CH4 Base & Current Address high byte
0C2h	0	1	0	Read CH4 Current Word Count low byte
0C2h	0	1	1	Read CH4 Current Word Count high byte
0C2h	1	0	0	Write CH4 Base & Current Word Count low byte
0C2h	1	0	1	Write CH4 Base & Current Word Count high byte
0C4h	0	1	0	Read CH5 Current Address low byte
0C4h	0	1	1	Read CH5 Current Address high byte
0C4h	1	0	0	Write CH5 Base & Current Address low byte
0C4h	1	0	1	Write CH5 Base & Current Address high byte
0C6h	0	1	0	Read CH5 Current Word Count low byte
0C6h	0	1	1	Read CH5 Current Word Count high byte
0C6h	1	0	0	Write CH5 Base & Current Word Count low byte
0C6h	1	0	1	Write CH5 Base & Current Word Count high byte
0C8h	0	1	0	Read CH6 Current Address low byte
0C8h	0	1	1	Read CH6 Current Address high byte
0C8h	1	0	0	Write CH6 Base & Current Address low byte
0C8h	1	0	1	Write CH6 Base & Current Address high byte
0CAh	0	1	0	Read CH6 Current Word Count low byte
0CAh	0	1	1	Read CH6 Current Word Count high byte
0CAh	1	0	0	Write CH6 Base & Current Word Count low byte
0CAh	1	0	1	Write CH6 Base & Current Word Count high byte

DMA Controller #2 Continued

Flip- Add. XIOR* XIOW* Flop Operation			
0CCh	0	1	0 Read CH7 Current Address low byte
0CCh	0	1	1 Read CH7 Current Address high byte
0CCh	1	0	0 Write CH7 Base & Current Address low byte
0CCh	1	0	1 Write CH7 Base & Current Address high byte
0CEh	0	1	0 Read CH7 Current Word Count low byte
0CEh	0	1	1 Read CH7 Current Word Count high byte
0CEh	1	0	0 Write CH7 Base & Current Word Count low byte
0CEh	1	0	1 Write CH7 Base & Current Word Count high byte
0D0h	0	1	X Read Status Register
0D0h	1	0	X Write Command Register
0D2h	0	1	X Read DMA Request Register
0D2h	1	0	X Write DMA Request Register
0D4h	0	1	X Read Command Register
0D4h	1	0	X Write single-bit DMA Request Mask Register
0D6h	0	1	X Read Mode Register
0D6h	1	0	X Write Mode Register
0D8h	0	1	X Set byte pointer flip-flop
0D8h	1	0	X Clear byte pointer flip-flop
0DAh	0	1	X Read Temporary Register
0DAh	1	0	X Master clear
0DCh	0	1	X Clear Mode Register counter
0DCh	1	0	X Clear all DMA Request Mask Register bits
0DEh	0	1	X Read all DMA Request Mask Register bits
0DEh	1	0	X Write all DMA Request Mask Register bits

Table 27. DMA Controller #2 Register Assignments.

In the program condition, the DMA subsystem supports several special commands for controlling the device. These commands do not make use of the data bus but are derived from a set of addresses, the internal select, and XIOW* or XIOR* signals. The special commands are: Master Clear, Clear Mask Register, Clear Mode Register Counter, and Set and Clear Byte Pointer Flip-Flop.

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

Active Condition

The IPC DMA subsystem enters the active condition whenever a software request occurs or whenever a DMA request on an unmasked channel occurs. In either case the device must not be in the program condition. The IPC then begins a DMA transfer cycle.

In a read cycle, for example, after receiving a DREQ signal, the IPC issues an HRQ signal to the system. Until an HLDA signal is returned, the DMA subsystem remains in an idle condition. On the next clock cycle, the DMA subsystem exits the idle condition and enters state S0.

During S0, the device resolves priority and issues the DACK signal on the highest-priority channel requesting service. The DMA subsystem then proceeds to state S1 where the multiplexed addresses are output and latched.

State S2 is then entered, at which time the IPC asserts DMAMEMR*. The device then moves into S3, where the XIOW* command signal is asserted. The DMA subsystem remains in S3 until the wait state counter decrements to zero and the IOCHRDY signal is true. Note that at least one additional S3 occurs unless compressed timing is selected.

Once a ready condition is detected, the DMA subsystem enters S4, where both signals are de-asserted. In Burst Mode and Demand Mode, discussed later, subsequent cycles begin in S2 unless the intermediate addresses require updating. In these subsequent cycles, the lower addresses are changed in S2.

The DMA subsystem 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

Single Transfer Mode directs the DMA subsystem to execute only one transfer cycle at a time. The DREQ signal must be held active until the DACK signal becomes active. If the DREQ signal is held active throughout the cycle, the IPC de-asserts the HRQ signal and releases the bus once the transfer is complete. After the HLDA signal becomes inactive, the IPC again asserts the HRQ signal and executes another cycle on the same channel unless a request from a higher-priority channel has been received. In the Single Transfer Mode, the CPU is assured 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 T/C (Terminal Count) signal is generated. If the autoinitialization option has been enabled, the channel reinitializes itself. If not, the DMA subsystem sets the DMA request bit mask and suspends transferring on the channel.

Block Transfer Mode

When Block Transfer Mode is selected, the IPC begins transfers in response to either a DREQ signal or a software request and continues until the terminal count (FFFFh) is reached. At that time, the T/C signal is pulsed and the Status Register terminal count bit is set. In Block Transfer Mode, the DREQ signal need only be held active until the DACK signal is asserted. Autoinitialization is optional in this mode also.

Demand Transfer Mode

In Demand Transfer Mode, the DMA subsystem begins transfers in response to the assertion of the DREQ signal and continues until either the terminal count is reached or the DREQ signal becomes inactive.

Demand Transfer Mode is normally used for peripherals that have limited buffering ability. The peripheral can initiate a transfer and continue until its buffer capacity is exhausted. Then, the peripheral can re-establish service by again asserting the DREQ signal. During idle periods between transfers, the CPU is released to operate and can monitor the operation by reading intermediate values from the Base Address, Current Address, Base Word Count, and Current Word Count Registers. Once the DREQ signal is de-asserted, higher-priority channels are allowed to intervene. Reaching terminal count will result in the generation of a T/C signal pulse, the setting of a terminal count bit in the Status Register, and autoinitialization (if enabled).

Cascade Mode

Cascade Mode is used to interconnect multiple DMA controllers, thus extending 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 receives an HLDA signal from the CPU in response to a DREQ signal caused by the HRQ signal from a slave DMA controller, the master DMA controller ignores all inputs except the HLDA signal from the CPU and the DREQ signal on the active channel. This prevents conflicts between the DMA devices.

Channel 0 of DMA2 is internally connected to DMA1 for Cascade Mode. Additional devices can be cascaded to the available channels in either DMA1 or DMA2 since cascading is not limited to two levels of DMA controllers.

When programming cascaded controllers, begin with the device that is actually generating the HRQ signal to the system. (This is normally the first-level device.) Then, proceed to the second-level devices. Reset causes the DACK signal outputs to become active low and places them in the inactive state.

To allow the internal cascade between DMA1 and DMA2 to function correctly, do not modify the active low state of the DACK signal. The IPC has an inverter between the DACK0 signal of DMA2 and the HLDA signal of DMA1, and the first-level device's DMA request mask bits will prevent second-level cascaded devices from generating unwanted hold requests during the initialization process.

DMA Transfers

Four types of transfers are provided in the IPC DMA subsystem:

- **Read Transfers**--move data from memory to an I/O device by generating the memory address and asserting the DMAMEMR* and XIOW* signals during the same cycle.
- **Write Transfers**--move data from an I/O device to memory by generating the memory address and asserting DMAMEW* and XIOR* signals.
- **Memory to Memory Transfers**--move block of memory from one location to another. DMA channels 0 and 1 may be used as memory to memory channels by setting bit 0 in the Command register. Once programmed the process can be started by either a software or external request to channel 0 of the DMA controller. Channel 0 provides the source block address during the read portion of the cycle and channel 1 provides the address for the write portion of the cycle. During the read cycle, a byte of data is latched into the internal temporary register of the IPC. The contents are then output to the XD0 - XD7 data lines during the write portion of the cycle and written to memory. Channel 0 may also be programmed to maintain the same source address on every cycle by setting bit 1 in the Command register.
- **Verify Transfers**--useful for diagnostics. In these pseudo-transfers, the DMA subsystem will operate as if it is performing a read or write transfer by generating the HRQ signal, addresses, and the DACK signal but will do so without asserting a command signal. Because no transfer actually takes place, the IOCHRDY signal is ignored during verify transfer cycles.

Autoinitialization

Each of the four DMA channel Mode Registers contains a bit that causes the channel 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 relocated into the Current Address and Current Word Count Registers. Both the Base and Current Registers are loaded during a CPU write cycle.

The Base Registers remain 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 Channels 0 and 1 must be programmed with the same starting value for full autoinitialization. If Channel 0 reaches terminal count before Channel 1, then Channel 0 reloads the starting address and word count and continues transferring data from the beginning of the source block. If Channel 1 reaches terminal count first, it reloads the current registers, and Channel 0 remains uninitialized.

DREQ Priority

The IPC supports three schemes for establishing DREQ priority. These are fixed priority, specific rotation, and automatic rotation. The default 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.

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

Address Generation

Eight intermediate bits of the address are multiplexed onto the data lines during active cycles of the DMA. This reduces the number of pins required by the DMA subsystem. During state S1, the intermediate addresses are output on Data Lines XD0-XD7. These addresses are externally latched by U2 and U3, (74ALS373), and used to drive the system address bus.

Because DMA1 is used to transfer 8-bit data and DMA2 is used to transfer 16-bit data, a 1-bit skew occurs in the intermediate address fields. DMA1 will therefore output Addresses A8-A15 on the data bus at this time, whereas DMA2 will output A9-A16. A separate set of latch and enable signals are provided for both DMA1 and DMA2 to accommodate the address skew.

Eight-Bit Cycles. During 8-bit DMA cycles, in which DMA1 is active, the IPC outputs the lower 8 bits of address on XA0-XA7. The intermediate 8 bits of address are output on XD0-XD7, and the ADSTB8 signal is asserted for one DMA clock cycle. The falling edge of the ADSTB8 signal is used to latch the intermediate addresses A8-A15. An enable signal, AEN8, is used to control the output drivers of the external latch (U2). A16-A23 are also generated at this time from a DMA Page Register in the IPC. Note that A16 is output on the XA16 pin of the device.

Sixteen-Bit Cycles. During 16-bit cycles, in which DMA2 is active, the IPC must output the lower 8 bits of the addresses on XA1-XA8. The intermediate addresses (A9-A16) are output on XD0-XD7. Control for a separate latch (U3) is provided by signals ADSTB16 and AEN16. The DMA Page Register now generates A17-A23. During 16-bit DMA transfers, XA0 and XA16 remain inactive.

DMA Page Register. The DMA Page Register is a set of sixteen 8-bit registers in the IPC that are used to generate the high order addresses during DMA cycles. Only eight of the registers are actually used. The remaining eight are included to maintain IBM AT compatibility. With the exception of Channel 0 of DMA2 (which is used for internal cascading to DMA1), each DMA channel has a register associated with it. Assignment of each register is shown in the following table, along with the register's read/write addresses.

Address	Register Function
080h	Unused
081h	8-bit DMA Channel 2 (DACK2)
082h	8-bit DMA Channel 3 (DACK3)
083h	8-bit DMA Channel 1 (DACK1)
084h	Unused
085h	Unused
086h	Unused
087h	8-bit DMA Channel 0 (DACK 0)
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

Table 28. DMA Address Extension Register Map.

During demand and block transfers, the IPC generates multiple sequential transfers. For most of these transfers, the information in the external address latches remains the same, eliminating the need to be relatched. Because 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 updates the latch contents only when necessary. Therefore, the IPC executes S1 cycles only when necessary, resulting in an overall through-put improvement.

Compressed Timing

The DMA subsystem in the IPC can be programmed to transfer a word in as few as three 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 four DMA clock cycles because S3 is executed twice because of the one wait state insertion.

In systems capable of supporting higher 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, the T/C signal will be output in S2, and S1 cycles will be executed as necessary to update the address latch.

Compressed timing is not allowed for memory-to-memory transfers.

Description of Registers

Current Address Register

Each DMA channel has a 16-bit Current Address Register that holds the address used during transfers. Each channel can be programmed to increment or decrement this register whenever a transfer is completed. This register can be read or written by the CPU in consecutive 8-bit bytes. If autoinitialization is selected, the 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 DMA channel also 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 goes from zero to FFFFh. When this roll-over occurs, the IPC generates the T/C signal. Then, it either suspends operation on that channel and sets the appropriate Request Mask bit or it autoinitializes and continues.

Base Address Register

Associated with each Current Address Register is a Base Address Register. This is a write-only register that is loaded by the CPU when writing to the Current Address Register.

The purpose of the Base Address Register is to store the initial value of the Current Address Register for autoinitialization. The contents of the Base Address Register are loaded into the Current Address Register whenever terminal count is reached and the Autoinitialization bit is set.

Base Word Count Register

The Base Word Count Register preserves the initial value of the Current Word Count Register. It is also a write-only register that is loaded by writing to the Current Word Count Register. The Base Word Count Register is loaded into the Current Word Count Register during autoinitialization.

Command Register

The Command Register controls the overall operation of the DMA subsystem. This register can be read or written by the CPU and is cleared by either a reset or a Master Clear command. The format and explanation of this register are as follows:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
DAK	DRQ	EW	RP	CT	CD	AH	M-M

DAK - Bit 7 determines the DACK signal level. Programming a 1 in this bit position makes the DACK signal an active high signal.

DRQ - Bit 6 determines the DREQ signal active level. Writing a 1 in this bit position causes the DREQ signal to become active low.

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

RP - Writing a 1 to Bit 4 causes the IPC to use a rotating priority scheme for honoring DMA requests. The default condition (0) is fixed priority.

CT - Writing a 1 to Bit 3 enables compressed timing. The default (0) condition causes the DMA to operate with normal timing.

CD - Bit 2 is the master disable for the DMA controller. Writing a 1 to this location disables the DMA subsystem (DMA1 or DMA2). Disabling is normally used whenever the CPU needs to reprogram one of the channels to prevent DMA cycles from occurring.

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

M-M - Writing a 1 in Bit 0 enables Channels 0 and 1 to be used for memory-to-memory transfers.

Mode Register

Each DMA channel has a Mode Register associated with it. All four Mode Registers reside at the same I/O address.

Bits 0 and 1 of the Write Mode Register determine 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 operations, Bits 0 and 1 will both be a logical 1. The definitions of the Mode Register bits are as follows:

msb	b6	b5	b4	b3	b2	b1	lsb
b7	M0	DEC	AI	TT1	TT0	CS1	b0 CS0
M1							

M1-M0 - Mode selection for each channel is accomplished by Bits 6 and 7. The following table shows the definitions.

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

Table 29. Definition of Mode Selection Bits.

DEC - Determines the direction of the address counter. A 1 in Bit 5 decrements the address after each transfer.

AI - Writing a 1 in Bit 4 enables the autoinitialization function.

TT1-TT0 - Bits 2 and 3 control the type of transfer which is to be performed. The following table gives the definitions:

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

Table 30. Mode Register
Bits 2 and 3 (Transfer Type).

CS1-CS0 - Bits 0 and 1 are the Channel Select bits. They determine which channel's Mode Register will be written. Read back of a Mode Register will result in Bits 0 and 1 being 1's. The following table explains the configuration of Bits 0 and 1.

CS1	CS0	Channel
0	0	Channel 0 select
0	1	Channel 1 select
1	0	Channel 2 select
1	1	Channel 3 select

Table 31. Mode Register
Bits 0 and 1 (Channel Select).

Request Register

The Request Register is a 4-bit register used to generate software requests. (DMA service can be requested either externally or under software control.)

Request Register bits can be set or reset independently by the CPU. The request mask has no effect on software-generated requests. All four bits are read in one operation and appear in the lower four bits of the byte. Bits 4-7 are read as 1's. All four request bits are cleared to zero by reset. The following table gives an explanation of the Request Register:

	msb	b7	b6	b5	b4	b3	b2	b1	lsb b0	RS0
X		X		X		X	RB	RS1		

RB - Writing a 1 to Bit 2 sets the Request bit. Bits 0 and 1 select the bit (channel) to be manipulated.

RS1-RS0 - Bits 0 and 1 determine which channel's Mode Register will be written. Read back of the Mode Register will result in Bits 0 and 1 both being 1's. The following is a table of channel selection:

RS1	RS0	Channel
0	0	Channel 0 select
0	1	Channel 1 select
1	0	Channel 2 select
1	1	Channel 3 select

Table 32.
Request Register Write Operations.

The format for a Request Register read operation is as follows.

	msb	b7	b6	b5	b4	b3	b2	b1	lsb b0	RC0
1		1	1	1	1	RC3	RC2	RC1		

RC3-RC0 - During a Request Register read, the state of the Request bit associated with each channel is returned in Bits 0-3 of the byte. The bit position corresponds to the channel number.

Request Mask Register

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

msb	b6	b5	b4	b3	b2	bl	lsb b0
X	X	X	X	X	MB	MS1	MS0

MB - Bit 2 sets or resets the Request Mask bit for the channel selected by Bits 0 and 1 (MS0 and MS1). Writing a 1 to Bit 2 sets the mask, inhibiting external requests.

MS1-MS0 - Bits 0 and 1 select the specific mask bit to be set or reset. The following table explains the bit configurations:

MS1	MS0	Channel
0	0	Channel 0 select
0	1	Channel 1 select
1	0	Channel 2 select
1	1	Channel 3 select

Table 33. Request Mask Register Bits 0 and 1 (Mask Bit Selection).

Alternatively, all four mask bits (Bits 0-3) can be programmed in one operation by writing to the Write All Mask Bits address. Data format for this and the Read All Mask Bits function is shown below.

msb	b6	b5	b4	b3	b2	bl	lsb b0
X	X	X	X	MB3	MB2	MB1	MB0

MB3-MB0 - Each bit position in the field represents the mask bit of a channel. The mask bit number corresponds to the channel number associated with the mask bit.

All four mask bits are set following a reset or a Master Clear command. Individual channel mask bits will be 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 whether a channel has reached terminal count and whether an external service request is pending.

Bits 0-3 of this register are cleared by a reset, a Master Clear command, or each time a status read takes place. Bits 4-7 are cleared by a reset, a Master Clear command, or the pending request being de-asserted. Bits 4-7 are not affected by the state of the Mask Register bits. The channel number corresponds to the bit position.

The Status Register is a read-only register. It conforms to the following format:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
DRQ3	DRQ2	DRQ1	DRQ0	TC3	TC2	TC1	TC0

Special Commands

Five special commands are provided to make the task of programming the device easier.

These commands are activated as a result of a specific address and assertion of an XIOR* or XIOW* signal. Information on the data lines is ignored by the IPC whenever an XIOW* signal activated command is issued; thus, data returned on XIOR* signal activated commands is invalid.

Clear Byte Pointer Flip-Flop--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. This command is normally executed prior to reading or writing an Address or Word Count Register.

Set Byte Pointer Flip-Flop--allows the CPU to adjust the pointer to the high byte of an Address or Word Count Register.

Master Clear--has the same effect as a hardware reset (clears the Command Register, Status Register, Request Register, Temporary Register, Mode Register counter, and byte pointer flip-flop; sets the Request Mask Register). Immediately following a Master Clear or a reset, the DMA will be in the idle condition.

Clear Request Mask Register--enables all four DMA channels to accept requests by clearing the mask bits in the register.

Clear Mode Register Counter--clears the additional counter, which is used to allow access to the four Mode Registers while only using one address. After clearing the counter, you can read all four Mode Registers by doing successive reads to the Read Mode Register address. The registers are read in order from Channel 0 to Channel 3.

Interrupt Controller

Overview

Two interrupt controllers, INTC1 and INTC2, are included in the IPC. Each of the controllers is equivalent to an 8259A device operating in iAPX86 Mode.

The two devices are connected and must be programmed to operate in Cascade Mode for proper operation of all 16 interrupt channels.

INTC1 is located at Addresses 020h-021h and is configured for master operation in Cascade Mode. INTC2 is a slave device and is located at Addresses 0A0h-0A1h.

The interrupt request output signal from INTC2 (INT) is internally connected to the Interrupt Request Input Channel 2 (IR2) of INTC1. The address decoding and cascade interconnection matches that of the IBM AT.

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 (IRO) of INTC1. Interrupt request from the real-time clock is connected to Channel 0 (IRO) of INTC2.

Unless otherwise noted, the following descriptions of the interrupt subsystems pertain to both INTC1 and INTC2. Wherever register addresses are used, the address for the INTC1 register is listed first and the address for the INTC2 register follows, in parentheses. Example: 020h (0A0h).

Controller Operation

The Interrupt Request Register (IRR) is used to store requests from all the channels that are requesting service. The IRR bits are labeled using the channel names IR7-IR0.

The In-Service Register (ISR) contains all the channels that are currently being serviced. The ISR bits are labeled IS7-IS0, corresponding to IR7-IR0.

The Interrupt Mask Register (IMR) allows the CPU to disable any or all of the interrupt channels.

The interrupt controller's priority resolver evaluates inputs from the above three registers, issues an interrupt request, and latches the corresponding bit into the In-Service Register.

During interrupt acknowledge (INTA) cycles, a master controller outputs a code to the slave device. This code is compared in the cascade buffer/comparator with a 3-bit ID code previously written. If a match occurs in the slave controller, the controller generates an interrupt vector. The contents of the Vector Register are used to provide the CPU with an interrupt vector during INTA cycles.

Interrupt Sequence

The IPC allows the CPU to perform an indirect jump to a service routine in response to a interrupt request for service from a peripheral device. The indirect jump is based on a vector provided by the IPC 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.)

The following events occur during an interrupt sequence:

1. One or more of the interrupt requests (IR7-IR0) becomes active, setting the corresponding Interrupt Request Register bit(s).
2. The interrupt controller resolves priority (based on the state of the IRR, IMR, and ISR registers) and asserts the INTR signal (if appropriate).
3. The CPU accepts the interrupt and responds with the INTA cycle.
4. During the first INTA cycle, the highest-priority ISR bit is set and the corresponding IRR bit is reset. The internal cascade address is generated, and the XD7-XD0 outputs remain tri-stated.

5. The CPU executes a second INTA cycle, during which the IPC drives an 8-bit vector onto the data pins XD7-XD0, which is in turn latched by the CPU. The format of this vector is:

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	V7	V6	V5	V4	V3	1	1	1
IR6	V7	V6	V5	V4	V3	1	1	0
IR5	V7	V6	V5	V4	V3	1	0	1
IR4	V7	V6	V5	V4	V3	1	0	0
IR3	V7	V6	V5	V4	V3	0	1	1
IR2	V7	V6	V5	V4	V3	0	1	0
IR1	V7	V6	V5	V4	V3	0	0	1
IR0	V7	V6	V5	V4	V3	0	0	0

Table 34. Interrupt Vector Byte.

Note that V7-V3 are programmable by writing to Initialization Control Word 2. This will be covered later in the section.

6. At the end of the second INTA cycle, the ISR bit will be cleared if the Automatic End of Interrupt Mode is selected. (This mode is covered later in this section.) Otherwise, the ISR must be cleared by an End Of Interrupt (EOI) command from the CPU at the end of the interrupt service routine.

If there is no interrupt request at the beginning of the first INTA cycle, INTCl issues an interrupt level 7 during the second INTA cycle.

The following table defines the interrupt request sources:

Controller	Channel No.	Interrupt Request Source
INTC1	IR0	IRQ0 - Counter/Timer Out0
INTC1	IR1	IRQ1 - Keyboard
INTC1	IR2	IRQ2 - INTC2 Cascade Interrupt
INTC1	IR3	IRQ3 - Serial Port 2
INTC1	IR4	IRQ4 - Serial Port 1
INTC1	IR5	IRQ5 - Parallel Port 2
INTC1	IR6	IRQ6 - Diskette Controller
INTC1	IR7	IRQ7 - Parallel Port 1
INTC2	IR0	IRQ8 - Real-Time Clock IRQ
INTC2	IR1	IRQ9 - Software Redirected to IRQ2
INTC2	IR2	IRQ10 - Reserved
INTC2	IR3	IRQ11 - Reserved
INTC2	IR4	IRQ12 - Reserved
INTC2	IR5	IRQ13 - Co-Processor Interrupt
INTC2	IR6	IRQ14 - Hard Disk Controller
INTC2	IR7	IRQ15 - Reserved

Table 35. Interrupt Request Sources.

End Of Interrupt

An EOI is defined as a condition that causes an In Service Register bit to be reset. To define the ISR bit to reset, you can either use a CPU command (for a specific EOI) or you can instruct the priority resolver to clear the highest-priority ISR bit (for a non-specific EOI).

In modes that do not alter the fully nested structure, a non-specific EOI is sufficient. The IPC can determine the correct ISR bit to reset because the highest-priority ISR bit is necessarily the last level acknowledged and serviced.

In conditions where the fully nested structure is not preserved, a specific EOI must be generated at the end of the interrupt service routine.

An ISR bit that is masked, in Special Mask Mode by an Interrupt Mask Register 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 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 the Operational Command Word 2 (OCW2) Register.

Fixed Priority Mode

Fixed Priority Mode is the default condition. It 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 below:

Priority: Lowest Highest
Interrupt: 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, the following events occur:

1. Priority is resolved.
 2. The vector of the highest-priority request is placed on the bus.
 3. The In Service Register bit for that channel is set. This ISR bit remains set until an EOI is issued to that channel.

As long as the ISR bit is set, all interrupts of equal or lower priority are inhibited. A higher-priority interrupt that occurs during an interrupt service routine will be acknowledged only if the CPU has internally re-enabled interrupts.

Specific Rotation Mode

Specific rotation allows the system software to re-assign priority levels by issuing a command that redefines the highest-priority channel. The following is an example of specific rotation when a command is issued with Channel 5 selected.

Before Rotation

Priority: Lowest 5 4 3 2 Highest
Interrupt: 7 6 5 4 3 2 1 0

After Rotation

Priority: Lowest Highest
Interrupt: 5 4 3 2 1 0 7 6

Automatic Rotation Mode

In applications in which a number of equal-priority peripherals are requesting interrupts, you can use automatic rotation to equalize the priority assignment. In Automatic Rotation Mode, a peripheral, after being serviced, is assigned the lowest priority. All peripherals connected to the controller are serviced at least once in eight interrupt requests to the CPU from the controller.

Automatic rotation will occur, if enabled, as a result of an EOI (automatic or CPU-generated). The following is an example of automatic rotation in which IR4 is the highest-priority request being serviced.

Before Rotation

ISR	Status	Bit	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
			0	1	0	1	0	0	0	0

Priority:	Lowest						Highest
Interrupt:	7	6	5	4	3	2	1 0

After Rotation (IR4 Service Completed)

ISR	Status	Bit	IS7	IS6	IS5	IS4	IS3	IS2	IS1	ISO
			0	1	0	0	0	0	0	0

Priority:	Lowest						Highest	
Interrupt:	4	3	2	1	0	7	6	5

Programming the Interrupt Controller

Two types of commands are used to control the IPC interrupt controllers: initialization command words (ICWs) and operational command words (OCWs).

Initialization Command Words

The initialization process consists of writing a sequence of four bytes to each interrupt controller. The initialization sequence is started by writing the first initialization command word (ICW1) to Address 020h (0A0h) with a 1 on Bit 4 of the data byte. The interrupt controller interprets this as the start of an initialization sequence and does the following:

1. Resets the Initialization Command Word Counter to 0.
2. Latches ICW1 to the device.
3. Selects Fixed Priority Mode.
4. Assigns IR7 the highest priority.
5. Clears the Interrupt Mask Register.
6. Sets the Slave Mode address to 7.
7. Disables Special Mask Mode.
8. Selects the Interrupt Request Register for status read operations.

The next three I/O writes to Address 021h (0A1h) will load ICW2-ICW4. (All four bytes must be written for the controller to be properly initialized.) The initialization sequence can be terminated at any point by writing to Address 020h (0A0h) with a 0 in Data Bit 4. Note that this will allow OCW2 or OCW3 to be written.

Registers ICW1-ICW4 are write-only registers in the following formats:

ICW1 - Address 020h (0A0h)

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	SI	LTM	X	SM	X

SI - Bit 4 indicates to the interrupt controller that an initialization sequence is starting. This bit must be a 1 to write ICW1.

LTM - 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 will generate an interrupt request. To generate the proper interrupt vector, the interrupt request must be active until the first INTA cycle is started. (An IR7 vector is generated if the IRR input is de-asserted early.) The interrupt request must be removed prior to EOI to prevent a second interrupt from occurring.

SM - Bit 1 selects between Single Mode and Cascade Mode. Single Mode is used when only one interrupt controller (INTC1) is used. Therefore, it is not recommended for use with the 82C206.

Cascade Mode allows the two interrupt controllers to be connected through IR2 of INTC1. INTC1 allows INTC2 to generate its own interrupt vectors if Cascade Mode is selected and the highest-priority interrupt request pending is from an INTC2 input. INTC1 and INTC2 must be programmed for Cascade Mode for both controllers to operate.

ICW2 - Address 021h (0A1h)

								msb	lsb							
b7	b6	b5	b4	b3	b2	b1	b0	V7	V6	V5	V4	V3	X	X	X	

V7-V3 - These bits are the upper five bits of the interrupt vector and are programmable by the CPU. The lower three bits of the vector are generated by the priority resolver during INTA. (See Table 34, earlier in the section.) INTC1 and INTC2 need not be programmed with the same value in ICW2.

ICW3 Format for INTC1 - Address 021h

								msb	lsb							
b7	b6	b5	b4	b3	b2	b1	b0	S7	S6	S5	S4	S3	S2	S1	S0	

S7-S0 - These bits determine which interrupt requests have Slave Mode controllers connected. ICW3 in INTC1 must be written with a 04h for INTC2 to function.

ICW3 Format for INTC2 - Address 0A1h

msb	b7	b6	b5	b4	b3	b2	b1	lsb b0
	0	0	0	0	0	ID2	ID1	ID0

ID2-ID0 - These bits determine which Slave Mode address the controller will respond to during the cascaded INTA sequence. ICW3 in INTC2 should be written with a 02h for Cascade Mode operation. Note that Bits 3-7 should be 0.

ICW4 - Address 021h (0A1h)

msb	b7	b6	b5	b4	b3	b2	b1	lsb b0
	X	X	X	EMI	X	X	AEOI	X

EMI - Bit 4 will 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 INTC2 being blocked by INTCl. Correct handling of this mode requires the CPU to issue a non-specific EOI command to INTC2 and to check its In-Service Register for zero when exiting an interrupt service routine. If a zero value exists, a non-specific EOI command should be sent to INTCl. If not, no command is issued.

AEOI - Auto EOI (AEOI) is enabled when ICW4 is written with a 0 in Bit 1. The interrupt controller performs a non-specific EOI on the trailing edge of the second INTA cycle. Do not use AEOI in a device with fully nested interrupts unless the device is a cascade master.

Operational Command Words

Operational command words (OCWs) allow the IPC interrupt controllers to be controlled or reconfigured at any time while operating. Each interrupt has three operational command words that can be programmed to affect the proper operating configuration and a Status Register to monitor controller operation.

Operational Command Word 1 (OCW1) is located at Address 021h (0A1h). It can be written any time the controller is not in Initialization Mode. OCW2 and 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). The following format descriptions are for OCW1-OCW3. These registers are read/write registers.

OCW1 - Address 021h (0A1h)

msb									lsb		
b7	b6	b5	b4	b3	b2	b1	b0				
M7	M6	M5	M4	M3	M2	M1	M0				

M7-M0 - Bits 0-7 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, M1 controls IR1, M2 controls IR2, and so on.) Setting an IMR bit has no affect on lower-priority requests. All IMR bits are cleared by writing ICW1.

OCW2 - Address 020h (0A0h)

msb									lsb		
b7	b6	b5	b4	b3	b2	b1	b0				
R	SL	EOI	SI	2/3	L2	L1	L0				

R - Bit 7, in conjunction with Bits 5 (EOI) and 6 (SL), selects operational function. Writing a 1 in Bit 7 selects one of the rotation functions. The following table shows the R selections:

R	SL	EOI	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 0 to all three bit positions.

Table 36. R Selections.

SL - Bit 6, in conjunction with Bits 7 and Bit 5, selects operational function. Writing a 1 in Bit 6 causes a specific or immediate function to occur. All specific commands require Bits 2-0 (L2-L0) to be valid except for no operation. The following table shows the SL selections:

R	SL	EOI	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

Table 37. SL Selections.

EOI - Bit 5, in conjunction with Bits 6 and 7, selects operational function. Writing a 1 in Bit 5 causes a function related to EOI to occur. The following table shows the EOI selections.

R	SL	EOI	Function
0	0	1	Non-specific EOI command
0	1	1	Specific EOI command
1	0	1	Rotate on non-specific EOI
1	1	1	Rotate on Specific EOI

Table 38. EOI Selections.

SI - Writing a 0 in Bit 4 takes the interrupt controller out of Initialization Mode and allows OCW2 and OCW3 to be written.

2/3 - If the I/O write places a 0 in Bit 4 (SI), writing a 0 in Bit 3 (2/3) selects OCW2. Writing a 1 selects OCW3.

L2-L0 - These three bits are internally decoded to select the interrupt channel to be affected by the specific command. L2-L0 must be valid during three of the four specific cycles shown in Table 37. Following is a table that defines the L2-L0 selections:

L2	L1	L0	Interrupt
0	0	0	IR0
0	0	1	IR1
0	1	0	IR2
0	1	1	IR3
1	0	0	IR4
1	0	1	IR5
1	1	0	IR6
1	1	1	IR7

L2-L0 Definitions

OCW3 - Address 020h (0A0h)

msb						lsb	
b7	b6	b5	b4	b3	b2	b1	
0	ESMM	SMM	SI	2/3	PM	RR	RIS

ESMM - Writing a 1 in Bit 6 enables the Set/Reset Special Mask Mode function, controlled by Bit 5 (SMM). ESMM allows the other functions on OCW3 to be accessed and manipulated without affecting the Special Mask Mode state.

SMM - Writing a 1 to both Bits 5 and 6 enables the Special Mask Mode. Writing a 1 to Bit 6 and a 0 to Bit 5 disables Special Mask Mode. During Special Mask Mode, writing a 1 to any bit position inhibits interrupts. Writing a 0 enables interrupts on the associated channel by causing the priority resolver to ignore the condition of the ISR.

SI - Same as for OCW2.

2/3 - Same as for OCW2.

PM - Writing a 1 to Bit 2 of OCW3 enables Polled Mode, 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 has Bit 7 set if an interrupt is pending. If Bit 7 of the byte is set, the level of the highest pending request is encoded on Bits 2-0. The Interrupt Request Register remains frozen until the read cycle is completed, at which time the PM bit is reset.

RR - When Bit 1 is 1, reading the status port at Address 020h (0A0h) causes the contents of the Interrupt Request Register or In Service Register (determined by Bit 0) to be placed on XDO-XD7. Asserting Bit 2 (PM) forces RR reset.

RIS - Bit 0 selects between the IRR and the ISR registers during status read operations if Bit 1 (RR) is 1.

Timer/Counter

The counter/timer circuit (CTC) in the IPC contains three 16-bit counters (Counters 0-3) that can be programmed to count in binary or binary coded decimal (BCD.) Each counter operates independently of the others and can be programmed as a timer or a counter.

All three counters are driven from a common set of control logic. The control logic decodes control information written to the CTC and provides the control necessary to load, read, configure, and control each counter.

Counter 2 can be operated in any of the following six modes:

- Mode 0 - Interrupt on Terminal Count
- Mode 1 - Hardware-Retriggerable One-Shot
- Mode 2 - Rate Generator
- Mode 3 - Square Wave Generator
- Mode 4 - Software-Triggered Strobe
- Mode 5 - Hardware-Retriggerable Strobe

Counters 0 and 1 can be programmed for all six modes. However, the usefulness of Modes 1 and 5 is limited because of the lack of an external hardware trigger signal.

All three counters are driven from a common clock input pin (TMRCLK) that is independent from other clock inputs to the IPC. Counter 0's output (Out0) is connected to IR0 of INTC1 and can be used as an interrupt to the system for keeping time and task switching. Counter 1 can be programmed to generate pulses or square waves for use by external devices. Counter 2 is a full-function counter/timer. It can be used as an interval timer, a counter, or a gated rate/pulse generator.

Counter Description

Each counter contains the following:

- A Control Register.
- A Status Register.
- A 16-bit counting element (CE).
- A pair of 8-bit counter input latches (CIL,CIH).
- A pair of 8-bit counter output latches (COL,COH).
- A clock input for loading and decrementing the CE.
- A mode-defined GATE input for controlling the counter. (Only Gate 2 is accessible.)
- An OUT signal. (OUT0 is not externally accessible.)

The OUT signal's state and function is controlled by the Counter Mode and the condition of the CE. (See "Mode Definitions," at the end of this section.)

The Control Register stores the mode and command information used to control the counter. This register can 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 the count format information.

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

The CE is a loadable 16-bit synchronous down counter. It is loaded or decremented on the falling edge of the TMRCLK signal. It contains the maximum count when a 0 is loaded, which is equivalent to 65536 in binary operation or 10000 in BCD. The CE does not stop when it reaches 0. In Modes 2 and 3, the CE will be reloaded when it reaches 0. In all other modes, it will wrap around to FFFF in binary operation or 9999 in BCD.

The CE is indirectly loaded by writing one or two bytes (optional) to the counter input latches. The latches are in turn loaded into the CE. This allows the CE to be loaded or reloaded in one TMRCLK cycle.

The CE is also read indirectly by reading the contents of the counter output latches. COL and COH are transparent latches that can be read while transparent or latched. (See "Latch Counter Command," later in this section.)

Programming the CTC

After power-up, the condition of the CTC Control Registers, the Counter Registers, the CE, and the output of all counters is undefined. Each counter must be programmed before it can be used. This is either done by the BIOS or by the software application.

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. The Control Register is a write-only location. The following table shows the addresses and format of the control words and the Control Word Register.

msb	b7	b6	b5	b4	b3	b2	b1	lsb	b0
	F3	F2	F1	F0	M2	M1	M0		BCD

Address	Function
040h	Counter 0 Read/Write
041h	Counter 1 Read/Write
042h	Counter 2 Read/Write
<u>043h</u>	Control Register Write Only

Table 39. Control Word Addresses.

Control Word (043h)

F3-F0 - Bits 7-4 determine the command to be performed. The following table lists the available commands.

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")
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 "Read Back Command")

Table 40. Counter Commands.

M2-M0-Bits 3-1 determine the counter's mode during a Read/Write Counter command or select the counter during a Read Back command. (See the specific commands, later in this section.) Bits 3-1 become "don't care" during Latch Counter commands.

BCD - Bit 0 selects the binary coded decimal counting format during Read/Write Counter commands. During ~~a~~ a Read Back command, this bit must be a 0.

Read/Write Counter Command

When writing to a counter, observe the following conventions:

- Write each counter's control word before writing the initial count.
- When writing the initial count, follow the format specified in the control word (for example, least significant byte only or least significant byte, then most significant byte).

Providing the programming format is observed, a new initial count can be written into the counter at any time after programming without rewriting the control word.

During Read/Write Counter commands, M2-M0 are defined as follows:

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

Table 41. Mode Select.

Latch Counter Command

When a Latch Counter command is issued, the counter's output latches (COL,COH) latch the current state of the CE. COL and COH remain latched until read by the CPU or until 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 can be read directly.

It is possible to issue Latch Counter commands to multiple counters before reading the first counter to which a command was issued. However, multiple Latch Counter commands issued to the **same** counter without reading the counter will cause all but the first command to be ignored.

Read Back Command

The Read Back command allows you to check the count value, mode, and state of the OUT signal, and the Null Count Flag of the selected counter(s). The format for the Read Back command is as follows:

msb	b7	b6	b5	b4	b3	b2	b1	lsb	b0
	1	1	LC	LS	C2	C1	C0		0

LC - Writing a 0 in Bit 5 causes the selected counter(s) to latch the state of the CE in COL and COH.

LS - Writing a 0 in Bit 4 causes the selected counter(s) to latch the current condition of its Control Register, null count, and output into the Status Register. The next read of the counter causes the contents of the Status Register to be read. (See "Status Read," later in the section.)

C2-C0 - 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 latches remain latched until either the latch is read or the counter is reprogrammed.

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

Status Byte

msb	b7	b6	b5	b4	b3	b2	b1	lsb	b0
	OUT	NC	F1	F0	M2	M1	M0		BCD

OUT - Bit 7 contains the state of the counter's OUT signal.

NC - Bit 6 contains the condition of the Null Count Flag. This flag is used to indicate that the contents of the CE are valid. NC 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.

F1-F0 - Bits 5 and 4 contain the F1 and F0 command bits that were written to the Command Register of the counter during initialization. This information is useful in determining whether the high byte, the low byte, or both must be transferred during counter read/write operations.

M2-M1 - Bits 2 and 1 reflect the mode of the counter and are interpreted in the same manner as in Write Command operations.

BCD - Bit 0 indicates that the CE is operating in BCD format.

Counter Operation

Because of the previously stated restrictions in Counter 0 and Counter 1, Counter 2 will be used as the example in describing counter operation. However, the descriptions of Modes 0, 2, 3 and 4 are relevant to all three 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 signal.
- **Counter Load**--The transfer of the 16-bit value in CIL and CIH to the CE.
- **Initialized**--A control word is written and the counter input latches are loaded.

Counter 2 operates in one of the modes that are described in the following sections.

Mode 0: Interrupt on Terminal Count

Writing the control word causes the OUT2 signal to go low and to remain low until the CE reaches 0. At that time, the signal goes back high and remains high until a new count or control word is written. Counting is enabled when the GATE2 signal is at a logical "1". Disabling the count has no effect on the OUT2 signal.

The CE is loaded with the first TMRCLK pulse after the control word and initial count are loaded. When both CIL and CIH are written, the CE is loaded after CIH is written. This TMRCLK pulse does not decrement the count. Therefore, for an initial count of n , the OUT2 signal does not go high until $n + 1$ TMRCLK pulses after initialization. Writing a new initial count to the counter reloads the CE on the next TMRCLK pulse and causes counting to continue from the new count.

If an initial count is written with the GATE2 signal equal to 0, the CE is still loaded on the next TMRCLK pulse. In this case, however, counting does not begin until the GATE2 signal equals 1. The OUT2 signal, therefore, goes high n TMRCLK pulses after the GATE2 signal equals 1.

Mode 1: Hardware-Retriggerable One-Shot

Writing the control word causes the OUT2 signal to go high initially. Once initialized, the counter is armed and a trigger causes the OUT2 signal to go low on the next TMRCLK pulse. The OUT2 signal then remains low until the counter reaches 0. An initial count of n results in a one-shot pulse that is n TMRCLK cycles long.

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

Mode 2: Rate Generator

Mode 2 functions as a divide-by- n counter, with OUT2 as the carry. Writing the control word during initialization sets the OUT2 signal high.

When the initial count decrements to 1, the OUT2 signal goes low on the next TMRCLK pulse. The following TMRCLK pulse returns OUT2 high and reloads the CE. Then, the process is repeated. In Mode 2, the counter continues counting (if the GATE2 signal equals 1) and generates an OUT2 pulse every n TMRCLK cycles. A count of 1 is illegal in Mode 2.

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

Writing a new count while counting does not affect counter 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 event count, the duty cycle of OUT2 will be 50%. For odd count values, OUT2 is high one TMRCLK cycle longer than it is low.

Mode 4: Software-Triggered Strobe

Writing the control word causes the OUT2 signal to go high initially. Expiration of the initial count causes the OUT2 signal 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. It 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 the new count. This allows the sequence to be retriggered by software.

Mode 5: Hardware-Triggered Strobe

Writing the control word causes OUT2 to go high initially. Counting is started by trigger. The expiration of the initial count causes OUT2 to go low for one TMRCLK cycle. GATE2=0 disables counting.

The CE is loaded on the TMRCLK pulse after a trigger. Because loading the CE inhibits decrementing, OUT2 will go low for one TMRCLK cycle, $n + 1$ TMRCLK cycles after the trigger.

If a new count 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 retriggerable.

GATE2 Definition

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, the output of which 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.

The following table describes each channel's control and output.
(Note: Channel 1 is programmed as a rate generator to produce a
15-microsecond period signal used for refresh request.)

Channel	Signal Name	Description
0		System Timer
0	Gate 0	Tied active
0	Clk In 0	1.190 MHz Osc
0	Clk Out 0	IRQ0
1		Refresh Request Generator
1	Gate 1	Tied active
1	Clk In 1	1.190 MHz Osc
1	Clk Out 1	REFREQ
2		Speaker Tone Generation
2	Gate 2	Controlled by Bit 0 of Port 061H
2	Clk In 2	1.190 MHz Osc
2	Clk Out 2	Used to drive speaker

Table 42. Channel Control and Output.

Real-Time Clock

This section of the IPC combines a complete time-of-day clock with alarm, 100-year calendar, programmable period interrupt, and 114 bytes of low power static RAM. This section of the device can operate in a battery-powered mode to protect the contents stored in the SRAM and keep the clock functioning.

Register Access

Reading and writing to the 128 locations in the real-time clock is accomplished by first writing the index address out to 070H and then writing the desired data out to 071H. The index addresses for the real-time clock locations are:

Index	Function
00H	Seconds
01H	Seconds alarm
02H	Minutes
03H	Minutes alarm
04H	Hours
05H	Hours alarm
06H	Day of week
07H	Date of month
08H	Month
09H	Year
0AH	Register A
0BH	Register B
0CH	Register C
0DH	Register D
0EH	Diagnostic status byte
0FH	Shutdown status byte
10H	Diskette drive type byte (Drives A and B)
11H	Reserved
12H	Hard disk type byte (Drives C and D) types 1-14
13H	Reserved
14H	Equipment byte
15H	Low base memory byte
16H	High base memory byte
17H	Low expansion memory byte
18H	High expansion memory byte
19H	Disk C extended byte
1AH	Disk D extended byte
1BH-2BH	Reserved
2CH	Bit 0 = swap disk bit
2DH	Reserved
2EH-2FH	2-byte CMOS checksum
30H	Low expansin memory byte
31H	High expansion memory byte
32H	Date century byte
33H	Information flags (set during power on)
34H-7FH	Reserved

Table 43. Index Addresses for the Real-Time Clock Locations.

Time Calendar and Alarm Bytes

The CPU can obtain the time and calendar information by reading the appropriate locations in the real-time clock. Initialization of the time, calendar, and alarm information is accomplished by writing to these locations. Information is stored in these locations in binary-coded decimal format.

Before initialization of the internal registers can be performed, the SET bit in Register B should be set to a 1 to prevent real-time clock updates from occurring. The CPU then initializes the first 10 locations in 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 1Hz rate.

The 24/12 in Register B 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 hour's byte in both the time and alarm bytes will indicate p.m. 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 2ms. Information read while the real-time clock is performing an update 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, you need only program the time that the interrupt is to occur into the three 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" byte. For instance, an interrupt can be generated every hour by programming C0h into Register 5, or it can be generated once a second by programming the same value into all three alarm registers.

Static RAM

The 114 bytes of RAM from Index Addresses 0Eh-7Fh are not affected by the real-time clock. These bytes are accessible during the update cycle and can be used for anything. In the Tandy 4000, these locations are used for nonvolatile configuration storage and calibration parameters because this device is normally battery-powered when the system is turned off.

Control and Status Registers

The IPC contains four registers (A-D) used to control the operation and monitor the status of the real-time clock. These registers are located at Index Addresses 0Ah-0Dh and are accessible by the CPU at all times.

Register A (0Ah)

msb	b7	b6	b5	b4	b3	b2	b1	b0	lsb
	UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0	

Except for UIP, Register A is a read/write register.

UIP - Bit 7 is the Update In Progress flag. It is a status bit used to indicate that an update cycle is about to take place. A "1" indicates that an update cycle is taking place or is imminent. UIP goes active (high) 244us prior to the start of an update cycle and remains 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.

DV0-DV2 - Bits 4-6 are used to control the divider/prescaler on the real-time clock. While the IPC can operate at frequencies higher than 32.768KHz, this is not recommended for battery-powered operation because of the increased power consumption at these higher frequencies. The following table gives the settings for DV0-DV2:

DV2	DV1	DV0	OSCL Frequency	Mode
0	1	0	32.768 KHz	Operate
1	1	X		Reset Divider

Table 44. Bits 6, 5, and 4 of Register A.

RS0-RS3 - Bits 0-3 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 the periodic interrupts, however, do 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. Below are the interrupt rates for which the real-time clock can be programmed:

<u>Rate Selection</u>				<u>Time Base</u>
RS3	RS2	RS1	RS0	32.768 KHz
0	0	0	0	None
0	0	0	1	3.90526ms
0	0	1	0	7.8125ms
0	0	1	1	122.070us
0	1	0	0	244.141us
0	1	0	1	488.281us
0	1	1	0	976.562us
0	1	1	1	1.935125ms
1	0	0	0	3.90625ms
1	0	0	1	7.8125ms
1	0	1	0	15.625ms
1	0	1	1	31.25ms
1	1	0	0	62.5ms
1	1	0	1	125ms
1	1	1	0	250ms
1	1	1	1	500ms

Table 45. Interrupt Rates for Real-Time Clock.

Register B (0Bh)

msb	b7	b6	b5	b4	b3	b2	b1	lsb b0
SET	PIE	AIE	UIE	0	DM	24/12	DSE	

Register B is a read/write register.

SET - Writing a "0" to Bit 7 enables the update cycle and allows the real-time clock to function normally. When Bit 7 is set to a "1," the update cycle is inhibited and any cycle in progress is aborted. The SET bit is not affected by the RESET* input signal.

PIE - Bit 6 is the Periodic Interrupt Enable bit. It controls the generation of interrupts based on the value programmed into the RS0-RS3 bits (Bits 0-3) of Register A. This allows you to disable this function without affecting the programmed rate. Writing a "1" to Bit 6 enables the generation of periodic interrupts. Bit 6 is cleared to "0" by RESET*.

AIE - The generation of alarm interrupts is enabled by setting Bit 5 to a "1". Once this bit is enabled, the real-time clock generates an alarm whenever a match occurs between the programmed alarm and clock information. Programming the "don't care" condition into one or more of the Alarm Registers enables the generation of periodic interrupts at rates of 1 second or greater. Bit 5 is cleared by the RESET* signal.

UIE - This read/write bit is used to update the update-end flag bit in Register C. This causes the proper IRQ to be asserted. The RESET* pin going low or the SET bit going high resets the UIE bit.

DM - This bit determines whether the time and calendar updates are to use binary or BCD format. This bit is written or read by the processor program, but not modified by any internal functions or RESET*. A 1 signifies binary data while a 0 specifies BCD data.

24/12 - The 24/12 control bit, Bit 1, is used to establish the format of both hours and hours alarm bytes. If this bit is a "1," the real-time clock interprets and updates the information in these two bytes using the 24-hour mode. This bit can be read or written by the CPU. It is not affected by RESET*.

DSE - The real-time clock can be instructed to handle daylight-saving time changes by setting Bit 0 to a "1." This enables two exceptions to the normal timekeeping sequence to occur. Setting Bit 1 to a "0" disables the execution of these two exceptions. PSRSTB* has no affect on Bit 1.

Register C (0Ch)

	msb	b6	b5	b4	b3	b2	b1	lsb
IRQF	PF	AF	UF	0	0	0	0	0

Register C is a read-only register.

IRQF - Bit 7, the Interrupt Request Flag bit, is set to a "1" when any condition that can cause an interrupt is true and the interrupt enable for that condition is true. The condition that causes this bit to be set also generates an interrupt. The logic expression for this flag is as follows:

$$\begin{aligned} \text{IRQF} = & \text{ PF \& PIE or} \\ & \text{ AF \& AIE or} \\ & \text{ UF \& UIE} \end{aligned}$$

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

PF - Bit 6, the Periodic Interrupt flag, is set to a "1" when a transition, which is selected by RS0 - RS3 (Bits 0-3 of Register A), occurs in the divider chain. Bit 6 will become active, independent of the condition of the PIE control bit. The PF bit then generates an interrupt and sets IRQF if PIE is a "1".

AF - A "1" appears in Bit 5, the AF bit, whenever a match has occurred between the Time Registers and the Alarm Registers during an update cycle. This flag is also independent of its enable (AIE) and generates an interrupt if AIE is true.

UF - See "Update Cycle," later in this section.

Register D (0Dh)

	msb	b6	b5	b4	b3	b2	b1	lsb
VRT	0	0	0	0	0	0	0	0

Register D is a read-only register.

VRT - Bit 7, 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 signal is low. This pin is normally derived from the power supply that supplies power (Vcc) to the device. It allows you to determine whether the registers have been initialized since power was applied to the device. PSRSTB* has no affect on this bit. This signal can only be set by reading Register D. All unused register bits are "0" when read. They are not writable.

Update Cycle

During normal operation, the real-time clock will perform an update cycle once every second. The performance of an update cycle depends on the divider bits DV0-DV2 not being cleared and the SET bit in Register B being cleared. The function of the update cycle is to increment the Clock/Calendar Registers and compare them to the Alarm Registers. If a match occurs between the two sets of registers, an alarm is issued and an interrupt is issued if the alarm and interrupt control bits are enabled.

During an update, the lower ten registers are unavailable to the CPU. This is done to prevent the possible corruption of data in the registers or the reading of incorrect data. To avoid contention problems between the real-time clock and the CPU, a flag is provided in Register A to alert you of a pending update cycle.

This Update In Process (UIP) bit, Bit 7, 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 is cleared and the Update Flag (UF) bit in Register C is set. CPU access is always allowed to Registers A through D during update cycles.

Two methods of reading and writing to the real-time clock are recommended. Both enable you 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 that might occur must not require more than 244us to complete (from the beginning of the read of Register A to the completion of the last read or write to the Clock/Calendar Registers).
- The second method of accessing the lower ten registers is to read Register C once, disregard the contents, and then continue reading this register until the UF bit is a "1". The UF bit becomes true immediately after an update is completed. You then have 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, an alternate source of power is provided by connecting a battery to the Vcc supply pin of the device. A circuit is provided to switch from system power to batter power in such a way as to eliminate power drain on the battery when the entire IPC is active.

A pin is provided on the device to protect the contents of the real-time clock and reduce power consumption whenever the system is powered down. This pin (PWRGD) should be low whenever the system power supply is not within specifications for proper system operation. To prevent noise on the inactive pins, the PWRGD signal input disables all unnecessary inputs while the system is powered down. PWRGD remains inactive to allow the remainder of the device to operate properly when power is applied to the system.

One pin (PSRSTB) is provided to initialize the device whenever power is applied to the IPC. The PSRSTB pin does not alter the RAM or clock/calendar contents, but it does initialize the necessary Control Register bits. Assertion of the PSRSTB signal disables the generation of interrupts and sets a flag indicating that the contents of the device might not be valid.

Floppy Disk Controller

General Overview

The floppy disk controller resides on the main logic board and is located on the system data bus (SD0-SD15). It is a Western Digital WD37C65 integrated floppy disk control system. It supports two floppy drives through an internal, daisy-chained flat cable.

The controller supports data rates of 250 Kb/s, 300 Kb/s, or 500 Kb/s. High- and low-capacity 5½-inch and 720K (and 1.44M) 3½-inch floppy disk drives can be used on the same internal controller. The FDC interface uses the DMA (DRQ2-DACK2*) and interrupt request (IRQ6) controls from the system bus. The on-board controller can be strapped to either the primary (3F2-3F7 hex) or the secondary (372-377 hex) FDC port address. The control signals can be disabled through software, allowing two devices to share the same DMA and IRQ channels.

Host Interface

The host interface is the host microprocessor peripheral bus. This bus is composed of eight control signals and eight data lines. In the either the Special or PC/AT Mode, IRQ and DMA requests are tri-stated and qualified by DMA enable, internally provided by the Operations Register. The data bus, DMA, and IRQ outputs are designed to handle 20 LS TTL loads. Inputs, except the data bus, are Schmidt Trigger receivers.

During the command or result phase, the Main Status Register must be read by the CPU before each byte of information is written into or read from the Data Register. After each byte is written into or read from the Data Register, the CPU should wait 12us before reading the Main Status Register.

Bits D6 and D7 in the Main Status Register must be "0" and "1," respectively, before each byte of the command word can be written into the FDC. Many of the commands require multiple bytes. As a result, the Main Status Register must be read prior to each byte transfer to the FDC.

During the result phase, Bits D6 and D7 in the Main Status Register must both be "1's" before reading each byte from the Data Register. Note that this reading of the Main Status Register before each byte transfer to the FDC is required only in the command and result phases, not during the execution phase.

During the execution phase, the Main Status Register need not be read. If the FDC is in the Non-DMA Mode, the receipt of each data byte from the FDD is indicated by an interrupt signal on Pin 16 (IRQ). The generation of a read signal (RD*) or write signal (WR*) clears the interrupt and outputs the data onto the data bus. If the CPU cannot handle interrupts quickly enough, it might poll the Main Status Register and Bit D7 (RQM) functions as the interrupt signal. If a write command is in process, the WR* signal performs the reset to the interrupt signal.

Note that in the Non-DMA Mode, it is necessary to examine the Main Status Register to determine the cause of the interrupt. It might be a data interrupt or a command interrupt, either normal or abnormal. If the FDC is in the DMA Mode, no interrupt signals are generated during the execution phase. The FDC generates DMA requests when each byte of data is available. The DMA controller responds to these requests with both the DACK* and RD* signal. When the DACK* signal goes low, the DMA Request (DMA) signal is cleared. If a write command has been issued, a WR* signal appears instead of the RD* signal. After the execution phase has been completed, the terminal count has occurred, or the EOT sector has been read or written, an interrupt occurs (IRQ). This signifies the beginning of the result phase. When the first byte of data is read during the result phase, the interrupt is automatically cleared.

The RD* or WR* signals should be asserted while the DACK* is true. The CS* signal is used in conjunction with the RD* and WR* signals as a gating function during programmed I/O operations. CS* has no effect during DMA operations. It is very important to note that in the result phase all bytes must be read. The Read Data command, for example, has several bytes of data in the result phase. All seven bytes must be read to successfully complete the Read Data command. The FDC will not accept a new command until all seven bytes have been read. Other commands might require fewer bytes to be read during the result phase.

The FDC contains five status registers. The Main Status Register, mentioned previously, can be read by the processor at any time. The other four status registers (ST0, ST1, ST2, and ST3) are available only during a result phase and can be read only after completing a command. The particular command that has been executed determines how many status registers are read.

The bytes of data that are sent to the FDC to form the command phase, and those that are read from the FDC in the result phase, must occur in the order shown in the appropriate table (49-63) in the "Commands" section. The command code must be sent first and the other bytes sent in the prescribed sequence.

No foreshortening of the command or result phases is allowed. After the last byte of data in the command phase is sent to the FDC, the execution phase automatically starts. Similarly, when the last byte of data is read out of the result phase, the command ends automatically and the FDC is ready for a new command.

Control Register (3F7h Primary, 377h Secondary)

The Control Register provides support logic that latches the two lsb's of the data bus upon receiving the Load Control Register (LDCR*) and Write (WR*) signals. These bits are used to select the desired data rate, which in turn controls the internal clock generation.

Clock switchover is internally "deglitched," allowing continuous operation after changing data rates. If the Control Register is not used, the data rate is governed by the clock or crystal. The frequency must be 64 times the MFM data rate, up to a maximum frequency of 16MHz. This implies a maximum data rate of 250 Kb/s unless the Control Register is used. Bits 0 and 1 are used to set the transfer rate, bits 3-7 are reserved. Table 46 presents the Control Register.

CR1 Bit	CR0 Bit	DRV Type	Data Rate	Comments	RPM Out
0	1				
0	0	X	500 Kb/s	MFM	1
0	0	X	250 Kb/s	FM	1
0	1	0	250 Kb/s	MFM	0
0	1	1	300 Kb/s	MFM	0
1	0	X	250 Kb/s	MFM, RST Default	1
1	0	X	125 Kb/s	FM, RST Default	1
1	1	X	125 Kb/s	FM	0

Table 46. Control Register.

Note: Drive type 0 = dual speed, 1 = single speed. Same for RPM bit returned. 0 = dual speed, 1 = single speed.

Main Status Register (3F4h Primary, 374h Secondary)

The Main Status Register is an 8-bit register that contains the status of the FDC. It can be accessed at any time. Only the Master Status Register can be used to facilitate the transfer of data between the processor and FDC.

Bits 6 (DIO) and Bit 7 (RQM) in the Main Status Register indicate when data is ready and in which direction data will be transferred on the data bus. The maximum time between the last RD* or WR* signal during a command or result phase and DIO and RQM getting set is 12us if the 500 Kb/s MFM data rate is selected. The maximum time from the trailing edge of the last RD* in the result phase to CB (FDC Busy) going low is 12us. The locations and functions of the bits in the Main Status Register are as follows:

	msb		lsb
RQM	b7	b6 b5 b4	b3 b2 b1 b0
DIO	EXM	CB	D3B D2B D1B D0B

RQM - When Bit 7 is "1," the Data Register is ready to send data to or receive data from the processor.

DIO - Bit 6 indicates the direction of data transfer between the FDC and the Data Register. A "1" indicates that the transfer is from the register to the processor. A "0" indicates that the transfer is from the processor to the register.

EXM - Bit 5 is set only during the execution phase of Non-DMA Mode. When this bit goes low, it indicates that the execution phase has ended and the result phase has started.

CB - A "1" in Bit 4 indicates that a read or write command is in progress. The FDC will not accept any other command.

D3B - A "1" in Bit 3 indicates that Drive 3 is in the Seek Mode and the FDC will not accept read or write commands.

D2B - A "1" in Bit 2 indicates that Drive 2 is in the Seek Mode and the FDC will not accept read or write commands.

D1B - A "1" in Bit 1 indicates that Drive 1 is in the Seek Mode and the FDC will not accept read or write commands.

D0B - A "1" in Bit 0 indicates that Drive 0 is in the Seek Mode and the FDC will not accept read or write commands.

Status Register 0

	msb	b7	b6	b5	b4	b3	b2	b1	lsb	b0
	IC1	IC2	SE	EC	NR	HS	US1	US0		

IC1-IC2 - Bits 6 and 7 are Interrupt Code bits. The following table shows their functions:

IC1	IC2	Meaning
0	0	Normal termination of command
0	1	Abnormal termination of command
1	0	Invalid command issued
1	1	Abnormal termination of command because of FDD ready signal changing state

Table 47. Bits 7 and 6 of Status Register 0.

SE - When the FDC completes the SEEK command, the Seek End bit, Bit 5, is set to a "1."

EC - Bit 4, the Equipment Check bit, is set if a fault signal is received from the FDD or if the Track 0 signal fails to occur after 255 step pulses (Recalibrate command).

NR - Drive ready is always presumed true. Therefore, the Not Ready bit, Bit 3, will always be a "0."

HS - Bit 2, the Head Select bit, is used to indicate the state of the head at interrupt.

US1-US0 - Bits 0 and 1 are the Unit Select bits. They are used to indicate the drive unit number at the time of interrupt.

US0	US1	Drive Number
0	0	0
0	1	1

Unit Select Bit Table

Status Register 1

	msb	b7	b6	b5	b4	b3	b2	b1	lsb	b0
EN	X	DE	OR	X	ND	NW	MA			

EN - Bit 7, the End of Cylinder bit, is set when the FDC tries to access a sector beyond the final sector of a cylinder.

b6 - Always 0

DE - The Data Error bit, Bit 5, is set if the FDC detects a CRC error in either the ID field or the data field.

OR - Bit 4, the Overrun bit, is set if the FDC is not serviced by the host system during data transfers within a certain time interval.

b3 - Always 0

ND - Bit 3, the No Data bit, is set:

- . During the execution of a Read Data, Write Deleted Data, or Scan command if the FDC cannot find the sector specified in the Internal Data Register (IDR)
- . During the execution of a Read ID command if the FDC cannot read the ID field without generating an error
- . During a Read a Track command if the starting sector cannot be found

NW - The Not Writable bit, Bit 1, is set if the FDC detects a WP* (Write Protect) signal from the FDD during a Write Data, Write Deleted Data, or Format a Track command.

MA - Bit 0 is the Missing Address Mark bit. It is set if the FDC cannot detect the ID address mark after twice encountering the index hole. At the same time, the MD (Missing Address Mark in Data Field) bit in Status Register 2 is set.

Status Register 2

	msb	b7	b6	b5	b4	b3	b2	b1	lsb	b0
		0	CM	DD	WC	SH	SN	BC		MD

b7 - Not used. Bit 7 is always 0.

CM - Bit 6 is the Control Mark bit. It is set during the execution of the Read Data or Scan command if the FDC encounters a sector that contains a Deleted Data Address Mark.

DD - Bit 5, the Data Error bit, is set if the FDC detects a CRC error in the data field.

WC - Bit 4 is the Wrong Cylinder bit. It is set if the contents of the cylinder on the medium are different from that stored in the IDR.

SH - The Scan Equal bit, Bit 3, is set during execution of the Scan command if the condition Equal is satisfied.

SN - Bit 2, the Scan Not bit, is set during the execution of the Scan command if the FDC cannot find a sector on the cylinder that meets the condition.

BC - Bit 1 is the Bad Cylinder bit. It is set when the contents of the cylinder on the medium are different from those stored in the IDR and the contents of the cylinder are FF.

MD - Bit 0 is the Missing Address Mark in Data Field bit. It is set when data is read from the medium and the FDC cannot find an Data Address Mark or Deleted Data Address Mark.

Status Register 3

msb	b6	b5	b4	b3	b2	b1	lsb
0	WP*	RY	T0	WP*	HS	USL	US0

b7 - Not used. Bit 7 is always 0.

WP* - Bit 6, the Write Protect bit, is set when write protect status is received from the FDD.

RY - The Ready bit, Bit 5, is always set to logical "1".

T0 - Bit 4 is the Track 0 bit. It is set when the Track 0 signal is received from the FDD.

WP* - Bit 3, the Write Protect bit, is set when write protect status is received from the FDD.

HS - Bit 2 is used to indicate the status of the side select signal from the FDD.

USL - Bit 1 is used to indicate the status of the unit select signal from the FDD.

US0 - Bit 0 is used to indicate the status of the unit select signal from the FDD.

Data Register (3F5h Primary, 375h Secondary)

The 8-bit Data Register stores data, commands, parameters, and FDD status information. Data bytes are read from, or written to, the Data Register to program or obtain the results after a particular command.

Operations Register (3F2h Primary, 372h Secondary)

The Operations Register provides support logic that latches the data bus upon receiving the LDOR* and WR* signals. The Operations Register replaces the typical latched port used in floppy subsystems to control disk drive spindle motors and to select the desired disk drive. The following table represents the Operations Register.

msb	lsb						
b7	b6	b5	b4	b3	b2	b1	b0
MS	X	MOEN2	MOEN1	DMAEN	SRST*	X	DSEL

b7 - Bit 7 is the Mode Select bit. During a soft reset condition, this bit can be used to select between Special Mode and AT Mode.

b6 - Not used.

MOEN2 - Bit 5 is the Motor On Enable bit, which is the inverted output of MO2*. This bit is only active in AT Mode.

MOEN1 - Bit 4 is the Motor On Enable bit, which is the inverted output of M01*. This bit is active only in AT Mode.

DMAEN - Bit 3 is the DMA Enable bit, which is active in both the Special and PC/AT Modes. This bit qualifies the DMA and IRQ outputs as well as the DACK* input.

SRST* - Bit 3 indicates a soft reset when it is low.

b1 - Not used.

DSEL - Bit 0 is the Drive Select bit. If it is low and MOEN1="1," then DS1 is active. If it is high and MOEN2="1," then DS2 is active. Bit 0 is only used in AT Mode.

Base, Special, and AT Modes

Base Mode

After a hardware reset, the RST signal active, the FDC will be held in soft reset, SRST* signal active, with the normally driven signals and the DMA request and IRQ request outputs tri-stated. Base Mode can be initiated at this time by a chip access by the host. Although the access can be any read or write, it is strongly suggested that the first access be a read of the Main Status Register.

Once Base Mode is entered, the soft reset is released, and IRQ and DMA are driven. Base Mode prohibits the use of the Operations Register. Therefore, there can be no qualifying by the DMAEN signal and no soft resets. The drive select output signals, DS1 to DS4, offer a 1 of 4 decoding of the unit select bits resident in the command structure. The RWC* signal represents reduced write current and indicates when write precompensation is necessary.

Special Mode

Special Mode allows the use of the Operations Register for the DMAEN signal as a qualifier to do a software driven device reset (SRST*). To enter Special Mode, the Operations Register is loaded with 1 X 0 0 X 0 X X, setting Mode Select to "1," which disables MOEN1 and MOEN2 and also causes SRST* to be active. Then, a read of the Control Register address, LDCR* and RD*, sets the device into Special Mode. The DS1 through DS4 signals, as well as the RWC* signal, are again offered in this mode.

AT Mode

For AT compatibility, users will write to the Operations Register, LDOR* and WR*. This action, performed after a hardware reset, or in the Base Mode, initiates AT Mode.

AT Mode can also be entered from Special Mode by loading the Operations Register with 0 X 0 0 X 0 X X, setting Mode S to a logical "0," disabling MOEN1 and MOEN2, and causing SRST* to be active. Then, a read of the Control Register address sets the device in AT Mode.

The DS* outputs are now replaced with the DSEL and MOEN signals buffered from the Operations Register. The RWC* pin is now RPM* so that users with two-speed drives can reduce spindle speed from a nominal 369 RPM to 300 RPM when active low or reduce write current when a slower data rate is selected for a given drive.

Polling Routine

After any reset, either RST or SRST*, US0 and US1 go into a polling routine. In between commands (and in between step pulses in the SEEK command), the FDC polls all four FDDs, looking for a change in the ready line from any of the drives.

Because the drive is always presumed ready, an interrupt is generated only following a reset. This occurs because a reset forces not ready status, which then promptly becomes ready. In Special or AT Mode, if DMAEN is not valid prior to lms after reset goes inactive, then IRQ may be already set and pending when finally enabled onto the bus.

The polling of the ready line by the FDC occurs continuously between commands. Each drive is polled every 1.024ms except during read/write commands.

Device Resets

The FDC supports both hardware reset (RST) pin 19, and software reset (SRST*), through the use of the Operations Register.

The RST pin causes a device reset for the active duration, RST causes a default to Base Mode, and default selects 250Kb/s MFM (or 125Kb/s FM code dependent) as the data rate (16MHz input clock).

SRST* resets the microcontroller as does the RST but does not affect the current data rate selection or the mode.

Both RST and SRST*, when active, disable the high current driver outputs to the FDD.

Data Separator

The data separator, built in the WD37C65 FDC, is a WD92C32 Phase Lock Loop Floppy Disk Data Separator (DPLL). It is designed to address high-performance error rates on floppy disk drives and to provide superior performance in terms of available bit jitter tolerance. It contains the necessary logic to achieve classical 2nd order, type 2, phase locked loop performance.

Write Precompensation

The FDC maintains the standard first-level algorithm to determine when write precompensation should be applied. The EARLY and LATE signals are used internally to select the appropriate delay in the write data pulse stream. The encoded WRITE DATA signal is synchronized to the 16MHz clock, CLK1 pin (23), and clocked through a shift register. Signals EARLY, NOM, and LATE determine the amount of delay through the shift register before a multiplexer gates the chosen bit to the output. The output data pulse width has a 25% duty cycle and is equal to one-half the WCLK period.

When the PCVAL signal is active, all data is precompensated by + or - 125ns, regardless of track number and data rate. However, this is only for MFM encoding. There is no write precompensation for FM. If the PCVAL signal is not active, and if a track inside number 28 is accessed, then + or - 187ns precompensation is generated. When the non-standard data rate using CLK2 is chosen, the MFM precompensation is always two clock cycles. For 9.6MHz, this is + or - 208ns. In this case, the PCVAL signal is disabled.

Clock Generation

This logical block internal to the WD37C65 FDC provides all the clocks needed by the FDC: the sampling clock (SCLK), the write clock (WCLK), and the master clock (MCLK).

SCLK drives the data separator used during data recovery. This clock frequency is always 32 times the selected data rate.

WCLK is used by the encoder logic to place MFM or FM on the serial WD-stream to the disk. WCLK always has a frequency of 2 times the selected data rate.

MCLK is used by the microsequencer. MCLK and MCLK* clock all latches in a two-phase scheme. One microinstruction cycle is four MCLK cycles. MCLK has a frequency equal to 8 times the selected MFM data rate or 16 times the FM data rate. The following table presents the clock data rates:

Data Rate	Code	SCLK	MCLK	WCLK
500 Kb/s	MFM	16 MHz	4.0 MHz	1.0 MHz
250 Kb/s	FM	8 MHz	4.0 MHz	500 KHz
250 Kb/s	MFM	8 MHz	2.0 MHz	500 KHz
125 Kb/s	FM	4 MHz	2.0 MHz	250 KHz
300 Kb/s	MFM	9.6 MHz	2.4 MHz	600 KHz

Table 48. Clock Data Rates.

Commands

The FDC is capable of performing 15 different commands. Each command is initiated by a multibyte transfer from the processor. The result after the execution of the command can also be a multibyte transfer back to the processor.

Command operation consists of three phases:

- **Command Phase**--The FDC receives all information required to perform a particular operation from the processor.
- **Execution Phase**--The FDC performs the operation as instructed.
- **Result Phase**--After completion of the operation, status and other housekeeping information are made available to the processor.

Tables 49-63 provide details about the FDC commands.

Command Symbol Description

Symbol	Name	Description
A0	Address Line 0	Controls selection of Main Status Register (A0=0) or Data Register (A0=1)
C	Cylinder Number	Current selected cylinder number
D	Data	Data pattern to be written into a sector
D7-D0	Data bus	8 bit data bus D0=lsb, D7=msb
DTL	Data Length	When N is 0, DTL stands for the data length to be written into a sector
EOT	End of Track	Final sector number on a cylinder. During read or write operations the FDC will stop data transfer after it reaches a sector number equal to EOT.
GPL	Gap Length	Gap 3 length. During a format command it determines the size of Gap 3.
H	Head Address	Head number, 0 or 1, as specified in the ID field.
HLT	Head Load Time	Head Load Time of FDD from 2-254ms in increments of 2ms.
HS	Head Select	Selected head number 0 or 1.
HUT	Head Unload Time	Head unload time after a read or write operation from 16-240ms in 16ms increments.
MF	FM or MFM	If low, FM mode is selected, if high, MFM mode is selected.

Command Symbol Legend

Symbol	Name	Description
MT	Multitrack	If MT is high, a multitrack operation is performed.
N	Number	Number of data bytes written in a sector.
NCN	New Cylinder Number	The new cylinder number that will be reached after a seek operation.
ND	Non DMA Mode	The operation is performed in the non-DMA mode.
PCN	Present Cylinder Number	The cylinder number at the end of the Sense Interrupt Status command.
R	Record	Sector number which will be read or written.
R/W	Read/Write	Read or Write signal.
SC	Sector	Number of sectors per cylinder.
SK	Skip	Skip Deleted Data Address Mark.
SRT	Step Rate Time	Stepping rate for the FDD 1-16ms in 1ms increments.
ST0	Status 0	Stands for one of the four status registers where information is stored after a command is executed.
ST1	Status 1	
ST2	Status 2	
ST3	Status 3	
STP		During scan operations if STP = 1 then contiguous sectors are compared with the data sent, if STP = 2 then alternate sectors are compared.

Command Symbol Legend

Read Data

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	MT	MF	SK	0	0	1	1	0	Command codes.
	W	X	X	X	X	HS	US1	US0		Sector ID information
	W					Cylinder number				prior to command
	W					Head address				execution. The
	W					Sector number				four bytes are
	W					Number of bytes				compared with
	W					Final sector number				the header on
	W					Gap length				the floppy disk.
	W					Data length				
Execution										Data transfer between FDD (Floppy Disk Drive) and system.
Results	R				ST0					Status information after command execution.
	R				ST1					
	R				ST2					
	R					Cylinder number				Sector ID information
	R					Head address				after command
	R					Sector number				execution.
	R					Number of bytes				

Table 49. Read Data Command.

Read Deleted Data

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	MT	MF	SK	0	1	1	0	0	Command codes.
	W	X	X	X	X	X	HS	US1	US0	
	W									Cylinder number
	W									Head address
	W									Sector number
	W									Number of bytes
	W									Final sector number
	W									Gap length
	W									Data length
Execution										Data transfer between FDD and system.
Results	R			ST0						Status information after command execution.
	R			ST1						
	R			ST2						
	R									Cylinder number
	R									Head address
	R									Sector number
	R									Number of bytes
										Sector ID information after command execution.

Table 50. Read Deleted Data Command.

Write Data

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	MT	MF	0	0	0	1	0	1	Command codes.
	W	X	X	X	X	HS	US1	US0		Sector ID information
	W									Head address
	W									Sector number
	W									Number of bytes
	W									Final sector number
	W									Gap length
	W									Data length
Execution										Data transfer between FDD and system.
Results	R			ST0						Status information after command execution.
	R			ST1						
	R			ST2						
	R				Cylinder number					Sector ID information
	R				Head address					after command execution.
	R				Sector number					
	R				Number of bytes					

Table 51. Write Data Command.

Write Deleted Data

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	MT	MF	0	0	1	0	0	1	Command codes.
	W	X	X	X	X	HS	US1	US0		Sector ID
	W					Cylinder number				information
	W					Head address				prior to command
	W					Sector number				execution. The
	W					Number of bytes				four bytes are
	W					Final sector number				compared with
	W					Gap length				the header on
	W					Data length				the floppy disk.
Execution										Data transfer between FDD and system.
Results	R			ST0						Status
	R			ST1						information
	R			ST2						after command
	R					Cylinder number				execution.
	R					Head address				Sector ID
	R					Sector number				information
	R					Number of bytes				after command
										execution.

Table 52. Write Deleted Data Command.

Read a Track

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	0	MF	SK	0	0	0	1	0	Command codes.
	W	X	X	X	X	HS	US1	US0		Sector ID
	W					Cylinder number				information
	W					Head address				prior to command
	W					Sector number				execution.
	W					Number of bytes				
	W					Final sector number				
	W					Gap length				
	W					Data length				
Execution										Data transfer between FDD and system. FDD reads all data fields from index hole to EOT.
Results	R				ST0					Status
	R				ST1					information
	R				ST2					after command
	R					Cylinder number				execution.
	R					Head address				Sector ID
	R					Sector number				information
	R					Number of bytes				after command
										execution.

Table 53. Read a Track Command.

Read ID

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	0	MF	0	0	1	0	1	0	Command codes.
	W	X	X	X	X	X	HS	US1	US0	
Execution										The first correct ID information on the cylinder is stored in the Data Register.
Results	R		ST0							Status information after command execution.
	R		ST1							
	R		ST2							
	R		Cylinder number							Sector ID information after command execution.
	R		Head address							
	R		Sector number							
	R		Number of bytes							

Table 54. Read ID Command.

Format a Track

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	0	MF	0	0	1	1	0	1	Command codes.
	W	X	X	X	X	X	HS	US1	US0	
	W									Number of bytes Bytes per sector.
	W									Number of sectors Sectors per track.
	W									Gap length Gap 3.
	W									DT Filler byte.
Execution										FDC formats the entire track.
Results	R			ST0						Status information after command execution.
	R			ST1						
	R			ST2						
	R			Cylinder number						Sector ID information after command execution.
	R			Head address						
	R			Sector number						
	R			Number of bytes						

Table 55. Format a Track Command.

Scan Equal

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	MT	MF	SK	1	0	0	0	1	Command codes.
	W	X	X	X	X	HS	US1	US0		
	W									Cylinder number
	W									Head address
	W									Sector number
	W									Number of bytes
	W									Final sector number
	W									Gap length
	W									STP
										1 = compare contiguous sectors; 2 = compare alternate sectors.
Execution										Data compared between FDD and system.
Results	R				ST0					Status information after command execution.
	R				ST1					
	R				ST2					
	R					Cylinder number				Sector ID information after command execution.
	R					Head address				
	R					Sector number				
	R					Number of bytes				

Table 56. Scan Equal Command.

Scan Low or Equal

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	MT	MF	SK	1	1	0	0	1	Command codes.
	W	X	X	X	X	HS	US1	US0		Sector ID
	W									information
	W									prior to command
	W									execution.
	W									1 = compare
	W									contiguous
	W									sectors;
	W									2 = compare
	W									alternate
	W									sectors.
Execution										Data compared
										between FDD and
										system.
Results	R			ST0						Status
	R			ST1						information
	R			ST2						after command
	R			Cylinder number						execution.
	R			Head address						Sector ID
	R			Sector number						information
	R			Number of bytes						after command
										execution.

Table 57. Scan Low or Equal Command.

Scan High or Equal

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	MT	MF	SK	1	1	1	0	1	Command codes.
	W	X	X	X	X	X	HS	US1	US0	
	W						Cylinder number			Sector ID information
	W						Head address			prior to command
	W						Sector number			execution.
	W						Number of bytes			
	W						Final sector number			
	W						Gap length			
	W						STP			1 = compare contiguous sectors; 2 = compare alternate sectors.
Execution										Data compared between FDD and system.
Results	R				ST0					Status information after command execution.
	R				ST1					
	R				ST2					
	R				Cylinder number					Sector ID information
	R				Head address					after command
	R				Sector number					execution.
	R				Number of bytes					

Table 58. Scan High or Equal Command.

Recalibrate

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W W	0 X	0 X	0 X	0 X	0 X	1 0	1 US1	1 US0	Command codes.
Execution										Heads retract to Track 0.

Table 59. Recalibrate Command.

Sense Interrupt Status

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	0	0	0	0	1	0	0	0	Command codes.
Results	R		ST0							Status information about the FDC at the end of a SEEK operation.
	R		Present cylinder							

Table 60. Sense Interrupt Status Command.

Specify

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	0	0	0	0	0	0	1	1	Command codes.
	W	Step rate time, head unload time								
	W	Head load time, Non-DMA Mode								

Table 61. Specify Command.

Sense Drive Status

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	0	0	0	0	0	0	1	0	Command codes.
	W	X	X	X	X	X	HS	US1	US0	
Results	R	ST3								Status information about FDC.

Table 62. Sense Drive Status Command.

Seek

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	0	0	0	0	1	1	1	1	Command codes.
	W	X	X	X	X	X	0	US1	US0	
	W	New cylinder number								
Execution										Heads positioned to the proper cylinder.

Table 63. Seek Command.

80287 Numeric Processor

The optional 80287 numeric processor is a 40-pin IC that can be used in the available socket (U25) on the main logic board. The optional 80287 receives its input clock from the oscillator Y5, an 8MHz crystal. Chip select and interrupt control for the Numeric Processor and the Weitek are provided by PAL U22. The numeric processor executes mathematical calculations independently, allowing the CPU to perform other tasks. It runs asynchronously to the 80386 and interfaces through the system bus through IRQ13 of the interrupt controller.

Serial/Parallel Interface

The serial/parallel card (standard) is compatible with the IBM AT. The serial portion of the card supports the EIA RS-232C interface standard through a 9-pin male D-shell connector located on the rear of the unit for external chassis access.

The UART, an National Semiconductor NS16450, used on this card is fully programmable and supports asynchronous communications with 5, 6, 7, or 8 character bits; 1, 1.5, or 2 stop bits; with or without parity. The baud-rate generator supports operation from 50 to 9600 baud. A prioritized interrupt scheme controls transmit, receive, error, line status, and data set interrupts.

The Serial portion of the adapter is accessible through Interrupt IR4 for COM1 and IR3 for COM2. The serial port is addressible through I/O port 3F8h-3FFh for COM1 and 2F8h-2FFh for COM2. For a detailed set of instructions and programming information, refer to the NS16450 in the Devices section.

The parallel portion of the board is accessible through Interrupt IR7 for LPT1 and IR5 for LPT2. The following is a table of addresses that can be used in programming the parallel port. The first address is for the primary port and the second is for the secondary port.

0378h (37Ch) Printer - Data Latch

Bit Description

- 0 Bit 0 - LSB
- 1 Bit 1
- 2 Bit 2
- 3 Bit 3
- 4 Bit 4
- 5 Bit 5
- 6 Bit 6
- 7 Bit 7 - MSB

0379h (37Dh) Printer - Read Status

Bit Description

- 0 Not Used
- 1 Not Used
- 2 Not Used
- 3 "0" = Error
- 4 "1" = Printer Select
- 5 "1" = Out of Paper
- 6 "0" = Acknowledge
- 7 "0" = Busy

Address Description

037Ah (37Eh) Printer - Control Latch

Bit Description

- 0 "1" = Strobe
- 1 "1" = Auto FD XT
- 2 "0" = Initialize
- 3 "1" = Select Printer
- 4 "1" = Enable Interrupt
- 5 Not Used
- 6 Not Used
- 7 Not Used

037Bh Not Used

Keyboard Control

The keyboard controller (U1) is a single-chip microcomputer (Intel 8042) that is programmed to be IBM AT compatible.

The controller receives serial data from the keyboard, checks the parity of the data, translates scan codes, and presents the data to the system as a byte of data in its output buffer, I/O address 060h. The controller will interrupt the CPU on interrupt line IRI when data is available to be read.

Data is sent to the keyboard by writing to the controller's input buffer, I/O address 064h for issuing a command, and 060h for data. (Status Register bits inform the CPU of any error encountered in receiving the data.) The byte of data is serialized and sent to the keyboard, with odd parity bits automatically inserted. The keyboard acknowledges all data transmissions, and the acknowledgment separates each byte transmitted.

The keyboard sends serial format data to the controller, using an 11-bit frame. (Data sent is synchronized by a keyboard-supplied clock.) The first bit is a start bit. It is followed by 8 data bits, an odd parity bit, and a stop bit. At the end of a transmission, the controller disables the interface until the system accepts the data byte.

For a received byte of data with a parity error, a resend command is sent to the keyboard.

Whenever the controller is unable to receive the data correctly, FF hex is placed in its output buffer, and the parity bit in the Status Register is set to 1.

The controller will also time a byte of data from the keyboard, and if the byte transmission does not end within 2 milliseconds, FF hex is placed in the controller's output buffer, and the receive time-out bit in the Status Register is set. (No retries will be attempted on a receive time-out error).

For scan codes received from the keyboard, the controller converts these codes to system scan codes before they are put into the controller's output buffer.

Data is sent to the keyboard in the same serial format used to receive data from the keyboard. If the keyboard does not start clocking the data out of the controller within 15 milliseconds, or if it completes that clocking within 2 milliseconds, FE hex is placed in the controller's output buffer, and the transmit time-out error bit is set in the Status Register.

If the response contains a parity error, FE hex is placed in the controller's output buffer, and the transmit time-out and parity error bits are set in the Status Register. The controller is programmed to a set time limit for the keyboard to respond. If 25 milliseconds are exceeded, the controller places FE hex in its output buffer and sets the transmit and receive time-out error bits in the Status Register. (No retry is made by the controller for any transmission error.)

The controller also senses the status of the display type jumper E1-E2.

Status Register Bit Definitions

Bit 0, Output Buffer Full - When this bit is a 0 it indicates that the output buffer is empty. A 1 indicates that there is data in the output buffer but the system has not yet read the data. When the system does read the data this bit is returned to a 0.

Bit 1, Input Buffer Full - A 0 in this position means the input buffer at I/O address 60h or 64h is empty. A 1 indicates that data has been written into the buffer but the controller (8042) has not read it yet. When the data is read by the controller this bit returns to a 0.

Bit 2, System Flag - This bit may be set to 0 or 1 by writing to the flag bit in the controller's command byte. It is set to 0 on power up reset.

Bit 3, Command/Data - The input buffer may be addressed as I/O address 60h or 64h. Address 60h is defined as a data port while address 64h is a command port. Writing to address 64h sets this bit to a 1, and writing to address 60h sets it to 0. The controller uses this bit to determine if the byte in the input buffer is a data or command byte.

Bit 4, Inhibit Switch - This bit is updated when data is placed in the output buffer. It reflects the state of the keyboard inhibit switch. A 0 in this bit means the keyboard is inhibited.

Bit 5, Transmit Time Out - A 1 in this bit indicates that a transmit by the keyboard controller was not properly completed. This bit works in conjunction with bits 6 and 7. If the transmit byte was not clocked out in the specified time limit, this will be the only bit set. If the transmit byte was clocked out and a response was not sent within the specified time limit, this bit and the receive time out bit is set. If the transmit byte was sent and a response was received with a parity error, this bit and the parity error bits are set.

Bit 6, Response Time Out - A 1 indicates that a transmission was started by the keyboard but did not finish within the programmed time limit.

Bit 7, Parity Error - A 0 indicates the last byte received from the keyboard had odd parity. A 1 indicates the last byte received had even parity. The keyboard should send data with odd parity.

Keyboard Controller Command Set (I/O Address 64h)

20 -- Read Keyboard Controller's Command Byte The controller sends the current command byte to the output buffer.

60 -- Write Keyboard Controller's Command Byte The next byte of data written to I/O address 60h is put in the controller's command byte. Bit definitions for the command byte are as follows:

Bit 7 - Reserved. Should be a 0.

Bit 6 - IBM Personal Computer Compatibility Mode.
Writing a 1 to this bit causes the controller to convert the scan codes it receives to those used by the IBM Personal Computer.

Bit 5 - IBM Personal Computer Mode.
Writing a 1 to this bit programs the keyboard to support the IBM Personal Computer keyboard interface. The controller will not check parity or convert scan codes.

Bit 4 - Disable Keyboard.
Writing a 1 to this bit disables the keyboard interface by driving the clock line low. Data is not sent or received.

Bit 3 - Inhibit Override.
A 1 in this bit disables the keyboard inhibit function.

Bit 2 - System Flag.
The value written to this bit is put in the system flag bit of the status register.

Bit 1 - Reserved. Should be written to a 0.

Bit 0 - Enable Output Buffer Full Interrupt.
Writing a 1 to this bit causes the controller to generate an interrupt when it places data into the output buffer.

AA -- Self Test Causes the controller to perform internal diagnostics. 55h is placed in the output buffer if no errors are encountered.

AB -- Interface Test Causes the controller to test the clock and data lines. The result is placed in the output buffer as follows:

00 - No error detected
01 - Clock line stuck low
02 - clock line stuck high
03 - data line stuck low
04 - data line stuck high

AC -- Diagnostic Dump Sends 16 bytes of the controller's RAM, current state of the input port, current state of the output port, and the controller's program status word to the system. All items are sent in scan code format.

AD -- Disable Keyboard Feature This sets bit 4 of the controller's command byte. This disables keyboard interfacing by driving the clock line low. Data will not be sent or received.

AE -- Enable Keyboard Interface This command clears bit 4 of the command byte. This releases the keyboard interface.

C0 -- Read Input Port Tells the controller to read the input port and place the data in the output buffer. This command should be used only if the output buffer is empty.

D0 -- Read Output Port Causes the controller to read the output port and place the contents in the output buffer. This command should only be used if the output buffer is empty.

D1 -- Write Output Port The next byte of data written to I/O address 60h is placed in the output port. Note that bit 0 of the controller's output port is connected to system reset and should not be written low.

E0 -- Read Test Inputs This causes the controller to read it's T0 and T1 inputs. The data is placed in the output buffer with data bit 0 representing T0 and data bit 1 representing T1.

F0-FF -- Pulse Output Port Bits 0-3 of the controllers output port may be pulsed low for approximately 6usec. Bits 0-3 of this command indicate which bits are to be pulsed. A 0 causes the bit to be pulsed, and a 1 causes the bit not to be pulsed. Note that bit 0 of the controller's output port is connected to system reset and should not be written low.

Interface Requirements

System I/O (AT) Bus

This section identifies the I/O interface requirements for both the PC-compatible and AT-compatible option cards. There are eight expansion slots in the Tandy 4000. Two are PC-type slots. Each of these has a 62-position connector. The other six are AT-type slots, each of which has one 62-position connector and one 36-position connector. The last slot is for a 32-bit memory expansion option. System board power consumption is rated at 3.5 Amps for the +5V supply and 45 mA for the +12V supply, including the FDC. This allows an effective 16.3A available for the +5V and 6.5A total for the +12V on the system I/O bus and disk drive power connectors.

PC Interface Compatibility

The following pin assignments are used on the 62-position connector found on both the AT- and the PC-type option slots:

Pin	Signal Name	Pin	Signal Name
A1	IOCHCHK*	B1	Ground
A2	SD7	B2	RESETDRV
A3	SD6	B3	+5 V
A4	SD5	B4	IRQ9
A5	SD4	B5	-5 V
A6	SD3	B6	DRQ2
A7	SD2	B7	-12 V
A8	SD1	B8	0WS*
A9	SD0	B9	+12 V
A10	IOCHRDY	B10	Ground
A11	AEN	B11	SMEMMW*
A12	SA19	B12	SMEMR*
A13	SA18	B13	IOW*
A14	SA17	B14	IOR*
A15	SA16	B15	DACK3*
A16	SA15	B16	DRQ3
A17	SA14	B17	DACK1*
A18	SA13	B18	DRQ1
A19	SA12	B19	REFRESH*
A20	SA11	B20	SYSCLK*
A21	SA10	B21	IRQ7
A22	SA9	B22	IRQ6
A23	SA8	B23	IRQ5
A24	SA7	B24	IRQ4
A25	SA6	B25	IRQ3
A26	SA5	B26	DACK2*
A27	SA4	B27	T/C
A28	SA3	B28	BALE
A29	SA2	B29	+5 V
A30	SA1	B30	OSC
A31	SA0	B31	Ground

Table 64. Pin Assignments for 62-Position Connector.

PC Bus Pin Definitions

The following signal descriptions for the System I/O Bus are for PC and AT bus-compatible option cards. Note that all signal lines are TTL compatible levels and that I/O adapters should be designed with a maximum of two low power Shottky (LS) loads per line.

CLK (B20). CLK is the System clock. It has a 50% duty cycle and is used only for synchronization with the CPU. It is not intended for uses requiring a fixed frequency. It is an 8 MHz signal.

SA0 through SA19 (A12-A31). These lines are 20 address bits used to address memory and I/O devices within the Tandy 4000. They are gated on the system bus when the BALE signal is high and are latched on the falling edge of the BALE signal. Generation of these signals is accomplished by the CPU or a DMA controller. SA0-SA19 are active high.

BALE (B28). BALE is a Buffered Address Latch Enable generated by the CPU Bus Controller. It is used to latch valid addresses from the CPU, and can be used by an I/O board to indicate a valid CPU address, in conjunction with AEN. BALE is pulled to a high state during DMA cycles, which include Refresh cycles. BALE is active high.

AEN (A11). AEN is an Address Enable signal used to remove the CPU and other devices from the bus to allow DMA transfers to take place. During AEN active, the DMA controller has control of the address bus, the data bus, the READ command lines, and the WRITE command lines. AEN is active high.

SD0 through SD7 (A2-A9). These signals are the data bus bits 0 through 7 from the CPU to memory and I/O devices on the bus. SD0 is the least significant bit (lsb), and SD7 is the most significant bit (msb).

RESETDRV (B2). RESETDRV is used to reset or initialize the expansion logic during power-up time, line voltage outage, or when the Reset switch on the front panel is pressed. RESETDRV is active high.

IOCHCHK* (A1). This signal indicates an uncorrectable system error when active. The IOCHCHK* signal provides the system board with parity information about memory or devices on the bus. IOCHCHK* is active low.

IOCHRDY (A10). This signal is used to lengthen I/O or memory cycles when driven low by the active device. (This signal should not be held low more than 15 microseconds.) Any slow device using this line should drive it low immediately upon detecting its valid address and a READ or WRITE command. See the timing diagram for setup times. IOCHRDY is active high (Ready condition).

IRQ9 and IRQ3 through IRQ7 (B4, B21-B25). These signals are used to tell the CPU that an I/O device needs attention. The Interrupt Requests are prioritized with IRQ9 (software redirected to IRQ2) having the highest priority and IRQ7 the lowest. An Interrupt Request is generated when any IRQ signal is driven high and held high until the CPU acknowledges the interrupt.

IOR* (B14). IOR* is a read signal that instructs an I/O device to drive its data onto the data bus (SD0-SD7). This line can be driven by the CPU Bus Controller or by the DMA controller. IOR* is active low.

IOW* (B13). IOW* is a write signal that instructs an I/O device to read, or latch, the data from the data bus (SD0-SD7). This line can be driven by the CPU Bus Controller or by the DMA controller. IOW* is active low.

SMEMR* (B12). SMEMR* is a read signal that instructs a memory device to drive its data onto the data bus (SD0-SD7). This line can be driven by the CPU Bus Controller or by the DMA controller through the CPU Bus Controller. SMEMR* is active only when the memory address is within the first 1 megabyte range (000000-0FFFFFH). SMEMR* is active low.

SMEMW* (B11). SMEMW* is a write signal that instructs a memory device to read, or latch, the data from the data bus (SD0-SD7). This line can be driven by the CPU Bus Controller or by the DMA controller through the CPU Bus Controller. SMEMW* is active only when the memory address is within the first 1 megabyte range (000000-0FFFFFH). SMEMW* is active low.

DRQ1, DRQ2, and DRQ3 (B18, B6, B16). These lines are asynchronous DMA requests by peripheral devices to gain DMA service. They are prioritized with DRQ1 having the highest priority, DRQ2 next, and DRQ3 lowest. A DMA request is generated by driving a DRQ line active high and holding it until the corresponding DACK (DMA acknowledge) signal goes active. DRQ1, DRQ2, and DRQ3 perform only 8-bit transfers. All DRQ lines are active high.

DACK1*, DACK2*, and DACK3* (B17, B26, B15). These lines are DMA acknowledge signals used to acknowledge DMA requests DRQ1, DRQ2, and DRQ3. All DACK signals are active low.

REFRESH* (B19). This signal is used to indicate a refresh cycle that can be used by a memory board to refresh Dynamic memory. REFRESH* is active low and generated every 15 usec.

T/C (B27). T/C is a signal that provides a pulse when the terminal count for any DMA channel is reached. T/C is active high.

OSC (B30). OSC is an oscillator signal that is a high-speed clock with a 70 nanosecond period (14.31818 megahertz). It has a 50% duty cycle.

AT Interface Compatibility

The following pin assignments are used on the 36-position connector found only on the AT-type slots.

Board Pin No. [†]	Signal Name	Board Pin No. ^{††}	Signal Name
C1†	SBHE	D1††	MEMCS16*
C2	SA23	D2	IOCS16*
C3	SA22	D3	IRQ10
C4	SA21	D4	IRQ11
C5	SA20	D5	IRQ12
C6	SA19	D6	IRQ15
C7	SA18	D7	IRQ14
C8	SA17	D8	DACK0*
C9	MEMR*	D9	DRQ0
C10	MEMW*	D10	DACK5*
C11	SD08	D11	DRQ5
C12	SD09	D12	DACK6*
C13	SD10	D13	DRQ6
C14	SD11	D14	DACK7*
C15	SD12	D15	DRQ7
C16	SD13	D16	+5 V
C17	SD14	D17	MASTER*
C18	SD15	D18	Ground

Table 65. Pin Assignments for 36-Position Connector.

[†] Note: Pin No.s that begin with a C on the PCB are labeled A on the schematic.

^{††} Note: Pin No.s that begin with a D on the PCB are labeled B on the schematic.

AT Bus Pin Definitions

SBHE (C1). Bus High Enable indicates a transfer on the upper eight bits of the data bus, SD8-SD15. Sixteen bit devices use this signal to condition the data bus buffers tied to SD8-SD15. SBHE is active high.

SA17-SA23 (C2 through C8). These signals are used to address memory and I/O devices in the system. These signals are valid when BALE is high. These signals are not latched during microprocessor cycles, therefore they do not stay valid for the entire cycle. They generate decodes for 1 wait state memory cycles. These decodes should be latched by the I/O adapters on the falling edge of BALE. These signals may also be driven by other DMA controllers or microprocessors that reside on the bus. SA17-SA23 are active high.

MEMR* (C9). This signal, along with SMEMR*, is used to instruct memory devices to drive data onto the data bus and is active on all memory read cycles. MEMR* may be driven by any DMA controller or microprocessor on the I/O bus. When a microprocessor on the bus wishes to assert MEMR*, it must have address lines valid on the bus for at least one system clock period before asserting MEMR*. MEMR* is active low.

MEMW* (C9). This signal, along with SMEMW*, is used to instruct memory devices to store data present on the data bus and is active on all memory write cycles. MEMW* may be driven by any DMA controller or microprocessor on the I/O bus. When a microprocessor on the bus wishes to assert MEMW*, it must have address lines valid on the bus for at least one system clock period before asserting MEMW*. MEMW* is active low.

SD8-SD15 (C11 through C18). These are the upper eight bus bits for the memory and I/O devices. 16 bit devices will use SD0-SD7 of the PC bus and SD8-SD15 of the AT bus.

MEMCS16* (D1). This signal is used to tell the system board if the present data transfer is a 16-bit N wait state memory cycle where N is determined by the 82C301. The default is 2. It is decoded from the SA17-SA23 lines. MEMCS16* is active low.

IOCS16* (D2). This signal is used to tell the system board if the present data transfer is a 16-bit N wait state I/O cycle where N is determined by the 82C301. The default is 2. IOCS16* is active low.

IRQ10-IRQ12, IRQ14-IRQ15 (D3 through D7). Interrupt Requests IRQ10-IRQ12, IRQ14-IRQ15 are used to signal the microprocessor that an I/O device needs attention. An interrupt request is generated by raising an IRQ line high. The line must be held high until the microprocessor acknowledges the request. IRQ13 is reserved for the system board.

DRQ0, DRQ5-DRQ7 (D9, D11, D13, D15). DMA request 0 and DMA request 5 through 7 are asynchronous channel requests used by peripheral devices on the I/O bus to gain DMA service. A request is generated by bringing the DRQ line high, and the line must be held high until a DMA Acknowledge line goes active. DRQ0 will perform 8-bit transfers, and DRQ5-DRQ7 will perform 16-bit transfers. These signals are active high.

DACK0*, DACK5* - DACK7* (D8, D10, D12, D14). These lines are used to acknowledge DRQ0 and DRQ5-DRQ7. These signals are active low.

+5V (D16). Bus connector power.

MASTER* (D17). This signal is used with the DRQ lines to gain control of the system. A processor or DMA controller on the I/O bus can issue a DRQ to a DMA channel and receive a DACK*. In response to the DACK*, the processor or DMA controller can pull the MASTER* signal low, thus allowing it to control the system data, address, and control lines. After MASTER* is low, the I/O device must wait one system clock period before driving the address and data lines, and two clock periods before issuing a read or write command. Warning, if this signal is held low for more than 15 microseconds, system memory may be lost due to lack of refresh.

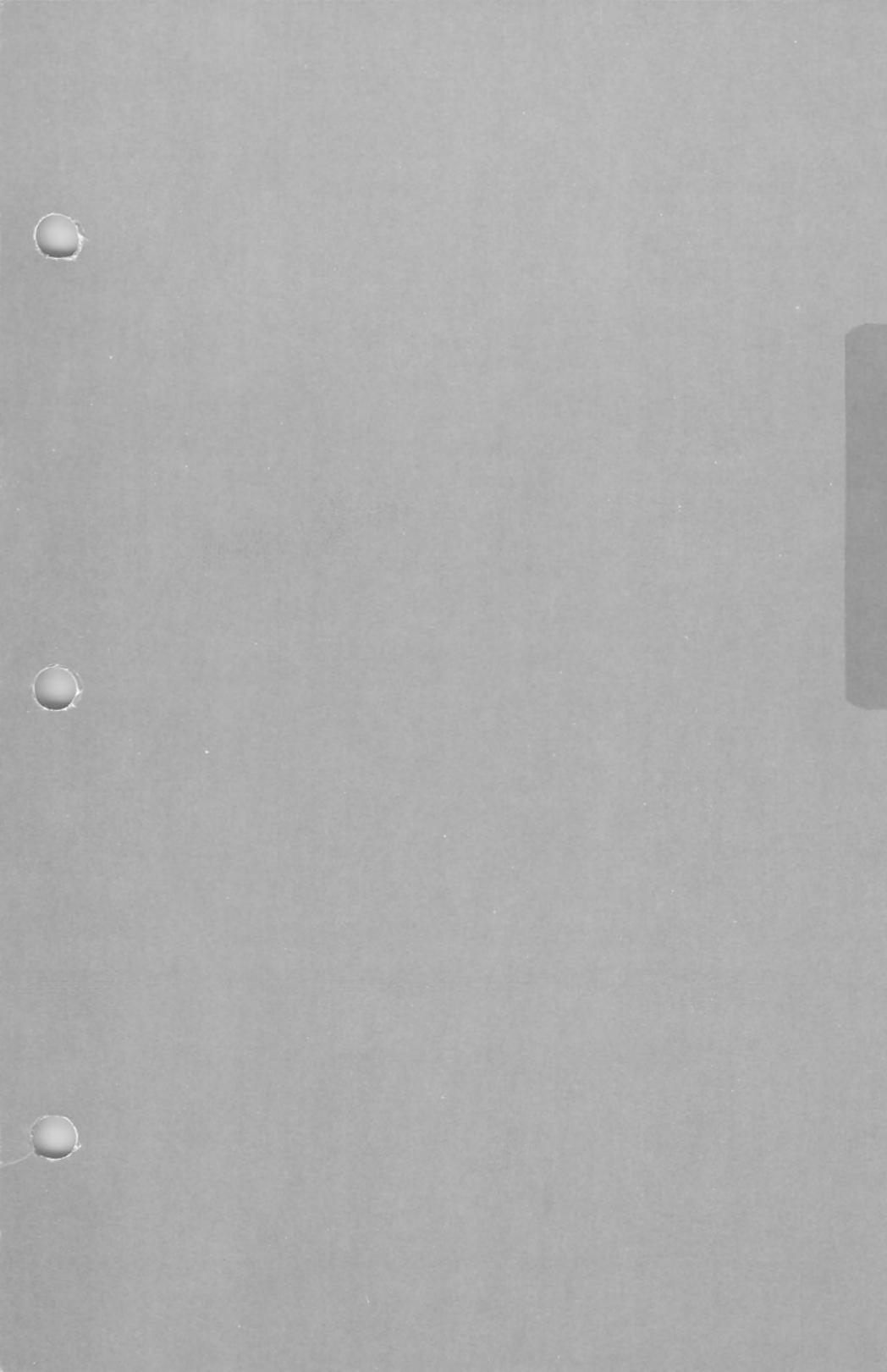
GROUND (D18). Ground for I/O bus.

Memory Expansion Bus

The memory expansion bus (MEB) is located in Slot #1 of the card cage. It is designed to support one 32-bit memory card with either 2 or 8 megabytes of 100nS DRAM. The memory expansion bus is presented to two 36-pin card edge connectors with the following signal description:

Pin	Signal Name	Pin	Signal Name
A1	+5 V	B1	+5 V
A2	Ground	B2	Ground
A3	MD00	B3	MD01
A4	MD02	B4	MD03
A5	MD04	B5	MD05
A6	MD06	B6	MD07
A7	MD08	B7	MD09
A8	MD10	B8	MD11
A9	MD12	B9	MD13
A10	MD14	B10	MD15
A11	MD16	B11	MD17
A12	MD18	B12	MD19
A13	MD20	B13	MD21
A14	MD22	B14	MD23
A15	MD24	B15	MD25
A16	MD26	B16	MD27
A17	MD28	B17	MD29
A18	MD30	B18	MD31
C1	+5 V	D1	+5 V
C2	Ground	D2	Ground
C3	+5 V	D3	+5 V
C4	Ground	D4	Ground
C5	MA0	D5	MA1
C6	MA2	D6	MA3
C7	MA4	D7	MA5
C8	MA6	D8	MA7
C9	MA8	D9	MA9
C10	Ground	D10	+5 V
C11	+5 V	D11	DWE*
C12	RAS2*	D12	CAS2*
C13	RAS3*	D13	CAS3*
C14	LBE0*	D14	MP0
C15	LBE1*	D15	MP1
C16	LBE2*	D16	MP2
C17	LBE3*	D17	MP3
C18	+5 V	D18	Ground

Table 66. Signal Description for 36-Pin Memory Card Edge.







TANDY COMPUTER PRODUCTS



4000 Devices





4000 Devices
Contents

Device	Manufacturer
82C206	Chips and Technology
82C301,82C302,82A303, 82A304,82A305,82A306	Chips and Technology
8042	intel
80287-8 (Option)	intel
80386	intel
NS16450 (UART)	National Semiconductor
M5M4256J-10,-12,-5 (DRAM)	Mitsubishi
MC14066	Motorola
MC14069	Motorola
U7 (System Data Bus Buffer Control)	Tandy
U9 (X and S Data Bus Buffer Control and FDC Port Decode)	Tandy
U10 (FDC Address Decode)	Tandy
U22 (Coprocessor Control)	Tandy
TMS27C128 (EPROM)	Texas Instruments
WD37C65	Western Digital



82C206 INTEGRATED PERIPHERALS CONTROLLER

- 100% Compatible to IBM™ PC AT
- Fully compatible to Intel™'s 8237 DMA controller, 8259 Interrupt controller, 8254 Timer/Counter, and Motorola™'s 146818 Real Time Clock
- Offers 7 DMA channels, 13 Interrupt request channels, 2 Timer/Counter channels, and a Real Time Clock
- Reduced recovery time (120 ns) between control
- 114 bytes of CMOS RAM memory
- 8 MHz DMA clock with programmable internal divider for 4 MHz operation
- Programmable wait states for the DMA cycle
- 16 Mbytes DMA address space
- Single chip 84-pin CMOS implementation

The 82C206 Integrated Peripheral Controller incorporates two 8237 DMA controllers, two 8259 Interrupt controllers, one 8254 Timer/Counter, one MC146818 Real Time Clock, 74LS612 memory mapper, in addition to several other TTL/SSI interface logic chips to offer a single chip integration of all the peripherals attached to the peripheral bus (X-Bus) in the IBM™ PC AT™. While offering a complete compatibility to the IBM PC AT architecture, the chip offers enhanced features and improved speed performance. These include an additional 64 bytes of user RAM for the Real Time Clock, and drastically reduced recovery specifications for the 8237.

8259 and 8254. Variable wait state option is provided for the DMA cycles. Programmable delays are provided for the CPU access to the internal registers of the chip. The chip also provides an option to select 8 or 4 MHz system clock.

The 82C206, along with the CS8220 PC AT Compatible CHIPSet, provides a highly integrated high performance solution for a PC AT compatible implementation.

The 82C206 is implemented using advanced CMOS technology and is packaged in an 84-pin PLCC.

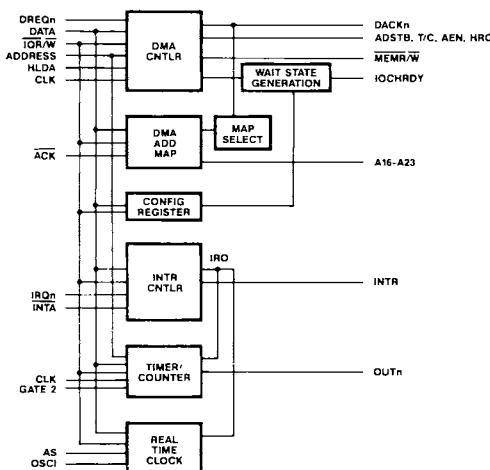
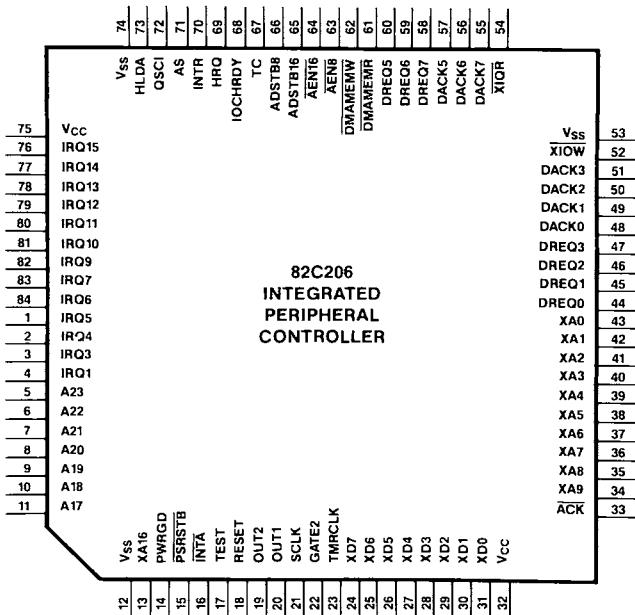


Figure 1. 82C206 Integrated Peripheral Controller Block Diagram



CHIPS

82C206 Pin Description

Pin No.	Name	Type	Function
24-31	XD7-XD0	I/O	<p>DATA BUS: The Data Bus lines are 3-state bi-directional lines connected to the system data bus (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 (Address register, Status register, the Temporary register or a Word Count register), the three Interrupt Controller registers (Interrupt Request register, In Service register and the Interrupt Mask register), the Timer/Counters registers (namely the contents of these counters or states of the counters), the Real Time Clocks internal registers and page registers of memory mapper.</p> <p>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/Counters registers, the DMA Page register and the Real Time Clock registers and internal RAM.</p> <p>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 ADSTB8 or ADSTB16. During memory-to-memory operations, data from the memory comes into the DMA Controller on the data bus during read from the memory. In the write to memory transfer, the data bus outputs place the data into the new memory location.</p> <p>During the interrupt sequence, the interrupt controllers output the interrupt vector byte on the data bus.</p> <p>Data bus XD7-XD0 also acts as the multiplexed address/data bus for the Real Time Clock.</p>

CHIPS

82C206 Pin Description (Continued)

Pin No.	Name	Type	Function
35-43 34	XA8-XA0 XA9	I/O I	ADDRESS BUS: This is the system address bus used to address various registers of the 82C206. It is tied to the external bus (XA bus) in a PC/AT compatible design. During a non-DMA cycle, A3-A0 act as inputs and are used by the CPU to address the registers of the DMA Controller corresponding to DMA channels 0-3 and A4-A1 address the registers of the DMA Controller corresponding to DMA channels 5-7. In the active DMA cycle, A7-A0 are outputs and carry address information for DMA channels 0-3. Correspondingly, A8-A1 are address outputs for 16 bit DMA channels 5-7. During program condition, A9-A0 are used to address configuration register and the internal registers of DMA Controller, INT Controller, Timer, RTC and Memory Mapper.
18	RESET	I	RESET: Reset is active high input which affects 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 Controllers: 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 mode is disabled and status read is set to IRR.
54	XIOR	I/O	I/O READ: I/O Read is a bidirectional active low three-state line. In an idle cycle (non DMA, non-interrupt), it is an input control signal used by the CPU to read the 82C206 internal registers. In an active DMA cycle, it is an output control signal used by the DMA Controller to access data from a peripheral during a DMA write transfer.
52	XIOW	I/O	I/O WRITE: I/O Write is a bidirectional active low three-state signal. In an idle cycle (non-DMA, non-interrupt) it is an input control signal used by the CPU to load information to the 82C206 internal registers. In an active DMA cycle, it is an output control signal used by the DMA Controller to load data to the peripheral during a DMA Read transfer.
61	DMAMEMR	O	DMA MEMORY READ: DMAMEMR is an active low three-state output used to access data from the selected memory location during DMA Read or memory-to-memory transfer.

CHIPS

82C206 Pin Description (Continued)

Pin No.	Name	Type	Function
62	DMAMEMW	O	DMA MEMORY WRITE: DMAMEMW is an active low three-state output used to write data to the selected memory location during DMA write or a memory-to-memory transfer. In a PC/AT compatible design, this signal is connected to XMEMW.
21	SCLK	I	CLOCK INPUT: The Clock Input is used to generate the timing signals which control DMA operations. This input may be driven from DC to 10 MHz. The Clock may be stopped in either state for standby operation. The internal clock used for DMA is either the SCLK or SCLK/2 depending on the setting of DMA CLOCK SELECT bit in the configuration register.
68	IOCHRDY	I/O	I/O CHANNEL READY: In the input mode, a low on IOCHRDY 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 up. This signal is used to extend memory read and write pulses for the DMA controllers to accommodate slow memories or I/O devices. IOCHRDY must satisfy set-up and hold times with respect to DMACLK in order to work reliably. In the output mode, this pin is an open drain output and provides an active low output whenever any 82C206 register is addressed for read or write. This output will remain low for a programmed number of DMACLK cycles (as configured by bits 6 and 7 of 82C206 configuration register) and then goes high, if pulled up by an external register. IOCHRDY provides means of introducing a programmed number of wait-states (as counted by DMACLK cycles) for I/O read/write cycles to 82C206. In a PC/AT architecture based design this pin should be wire-ored to PC/AT's IOCHRDY signal.
73	HLDA	I	HOLD ACKNOWLEDGE: The active high Hold Acknowledge from the CPU indicates that it has relinquished control of the system busses.

82C206 Pin Description (Continued)

Pin No.	Name	Type	Function
44-	DREQ0-	I	DMA REQUEST: The DMA Request (DREQ) are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. 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 DREQ signal. Polarity of DREQ is programmable. Reset initializes these lines to active high. DREQ must be maintained until the corresponding DACK goes active. DREQ will not be recognized while the clock is stopped. Unused DREQ inputs should be pulled High or Low (inactive) and the corresponding mask bit set.
47	DREQ3		
60-	DREQ5-		
58	DREQ7		
			DREQ0-DREQ3 support 8-bit transfers between 8-bit I/O and 8 or 16-bit system memory. DREQ5-DREQ7 support 16-bit data transfers between 16-bit peripheral and 16-bit system memory. DREQ4 is not available as it is used to cascade DREQ0-DREQ3.
67	TC	O	TERMINAL COUNT: Terminal Count (TC) is an active high signal. Information concerning the completion of DMA services is available at the TC output pin. A pulse is generated by the DMA Controller when terminal count (TC) for any channel is reached, except for channel 0 in memory-to-memory mode. During memory-to-memory transfers TC will be output when the TC for channel 1 occurs. When a TC pulse occurs, the DMA Controller will terminate the service, and if auto-initialize is enabled, the base registers will be written to the current registers of that channel. The mask bit and TC bit in the status word will be set for the currently active channel unless the channel is programmed for auto-initialize. In that case, the mask bit remains clear.

CHIPS

82C206 Pin Description (Continued)

Pin No.	Name	Type	Function
69	HRQ	O	HOLD REQUEST: The Hold Request (HRQ) output is used to request control of the system bus. When a DREQ occurs and the corresponding mask bit is clear, or a software DMA request is made, the DMA Controller issues HRQ. The HLDA signal then informs the controller when access to the system busses is permitted. For stand-alone operation where the DMA Controller always controls the busses, HRQ may be tied to HLDA. This will result in one SO state before the transfer.
48- 51 57- 55	DACK0- DACK3 DACK5- DACK7	O	DMA ACKNOWLEDGE: DMA Acknowledge is used to notify the individual peripherals when one has been granted a DMA cycle. The active polarity of these lines is programmable. Reset initializes them to active low. Because these signals are used internally for cascading the DMA channels and for DMA page register selection, these signals must be programmed to be active low.
66	ADSTB8	O	ADDRESS STROBE: This is an active high signal used to control latching of the upper address byte (A8-A15) for 8-bit peripherals. It will drive directly the strobe input of external transparent octal latches. During block operations, ADSTB8 will only be issued when the upper address byte must be updated, thus speeding operation through elimination of S1 states. ADSTB8 is active for DMA channels 0-3.
64	AEN16	O	ADDRESS ENABLE for 16-BIT DMA TRANSFERS: This signal enables the 8-bit latch containing the upper 8 address bits (A9-A16) on to system address bus. It is inactive when external bus master controls the system bus (MASTER=0). This signal is active low.
65	ADSTB16	O	ADDRESS STROBE for 16-BIT TRANSFERS (channels 5-7). This is an active high signal used to control latching of the upper address byte A9-A16 for 16-bit DMA transfers. Its function is just like ADSTB8.
63	AEN8	O	ADDRESS ENABLE for 8-BIT DMA TRANSFERS: This signal is the output enable for the 8-bit latch containing upper 8 address bits (A8-A15). It enables A8-A15 system address bus. It is inactive when external bus master controls the system bus (MASTER=0). This signal is active low.

CHIPS

82C206 Pin Description (Continued)

Pin No.	Name	Type	Function
33	ACK (MSE)	I	MODULE SELECT ENABLE: When high, it enables the chip select function on one of the modules (DMA Controller, INT Controller, TIMER, RTC, DMA Page register or the Configuration register) for the programming function, i.e. CPU read or write of the command, status or other register of various modules of the 82C206. When low, the 82C206 is essentially disconnected from the system bus. The 82C206 at this time could be performing an active DMA or an interrupt cycle. In a PC/AT compatible design, this pin is tied to ACK signal.
11-5 13	A23-A17 XA16	O	DMA PAGE REGISTER ADDRESS: XA16 and A17-A23 are 3-state output pins. XA16 is the least significant bit of the DMA page register and is used for DMA transfers for 8-bit peripherals only (channel 0-3). XA16 is not used for DMA transfers to 16-bit peripherals (channel 5-7) as XA9-XA16 is provided by demultiplexing the data bus. A17-A23 are the upper 7 bits of the DMA page register.
76-82 83, 84 1-3 4	IRQ15-IRQ9 IRQ7, IRQ6 IRQ5-IRQ3 IRQ1	I	INTERRUPT REQUESTS: Asynchronous inputs. An interrupt request is executed by raising an IRQ input 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).
16	INTA	I	INTERRUPT ACKNOWLEDGE: This pin is used to enable the interrupt controllers interrupt vector data on to the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
70	INTR	O	INTERRUPT: This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU, and is usually connected to the CPU's interrupt pin.
23	TMRCLK	I	TIMER CLOCK: Clock input for Counter 0, Counter 1 and Counter 2.
22	GATE2	I	GATE 2: Gate input for Counter 2. In a PC/AT compatible design, the Counter 2 is used for tone generation for speaker. In this design, the GATE 2 input is driven by Bit 0 of I/O Port 61H (called TIM2GATE SPK).
20	OUT1	O	OUT 1: Output of Timer 1. In a PC/AT compatible design, Timer 1 is programmed as a rate generator to produce 15 μ sec period signal used for interrupt request for refresh cycles.
19	OUT 2	O	OUT 2: Output of Timer 2. In a PC/AT compatible design, OUT 2 is used to drive the speaker.

CHIPS

82C206 Pin Description (Continued)

Pin No.	Name	Type	Function
71	AS	I	ADDRESS STROBE: Address strobe is a positive pulse whose falling edge latches the address from the XD bus.
72	OSCI	I	OSCILLATOR INPUT: The time base for the time functions is connected to this pin. External square waves of 32.768 KHz may be connected to this input.
15	PSRSTB	I	This input 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. When PSRSTB and TEST are both low, the following occurs: (a) Periodic Interrupt Enable (PIE) bit is cleared to zero. (b) Alarm Interrupt Enable (AIE) bit is cleared to zero. (c) Update ended Interrupt Enable (UIE) bit is cleared to zero. (d) Update ended Interrupt Flag (UF) bit is cleared to zero. (e) Interrupt Request status Flag (IRQF) is cleared to zero. (f) Periodic Interrupt Flag (PF) bit is cleared to zero. (g) The part is not accessible. (h) Alarm interrupt Flag (AF) bit is cleared to zero. (i) Square Wave output enable list is cleared to zero.
14	PWRGD	I	PWRGOOD: The Power Good pin must be high for bus cycles in which the CPU accesses the RTC. When PWRGD is low, all address, data, data strobe and R/W pins are disconnected from the processor.
17	TEST	I	TEST: Test is an active high input. It initializes various internal registers so that the test program starts in a known state. It should be tied low for normal operation.
32, 75	V _{CC}	—	Power Supply
12, 53, 74	V _{SS}	—	Ground

CHIPS

82C206—INTEGRATED PERIPHERALS CONTROLLER

The 82C206 is a LSI implementation of the standard peripherals required to implement an IBM PC/AT system board. This device contains the equivalent of two 8237A DMA Controllers, a 74LS612 Mapper, two 8259A Interrupt Controllers, an 8254 Counter/Timer, and a MC146818 Real Time Clock with RAM. The 82C206 provides all of the standard peripherals required for a system board implementation except the keyboard interface controller. Figure 1 illustrates the subsystems contained within the 82C206.

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 Page Register (DMAPAGE) device which is used to supplement the DMA and drive the upper address lines when required.

Sixteen channels of interrupt are provided in the 82C206. These channels are partitioned into two cascaded controllers (INTC1, INTC2) with 8 inputs each. Of these 16 channels, three are connected internally to various devices, allowing 13 user definable channels of interrupt. 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 which contains three independent counters. All three counters are driven from a clock input pin which is independent from the other clock inputs to the device. Counter 0 is connected to Interrupt 0 of INTC1. It is intended to be used as a multi-level 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 gate input for controlling the internal counter. This channel can be used as an interval counter, a timer, or as a gated rate/pulse generator.

A Real Time Clock (RTC) is included in the 82C206 for maintaining the time and date. This subsystem also contains 114 bytes of 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 system power is turned off.

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

The first section is the Clock and Wait State Control section. 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 Top Level Decode.

In order to accommodate over 200 registers in the 82C206 and maintain I/O decode compatibility with the IBM PC/AT, a multilevel decode scheme is employed. The Top Level Decode subsystem performs the function of generating enables to the various subsystems. Control and direction of the XD0-XD7 data bus buffers are also handled by this subsystem.

Each of these subsystems will now be described separately.

Top Level Decode

The 82C206 Top Level Decode provides 8 separate enables to various subsystems of the device. Figure 2 contains a truth table for the Top Level Decoder. The enabling of the 82C206 XD0-XD7 output buffers is also controlled by this section. The output buffers are enabled whenever an enable is generated to an internal subsystem and the XIOR signal is asserted.

CHIPS

ACK	XA9	XA8	XA7	XA6	XA5	XA4	XA3	XA2	XA1	XA0	ADDRESS RANGE(Hex)	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	0	0	0	1	071	RTC
1	0	0	1	0	0	0	X	X	X	X	080-08F	DMAPAGE
1	0	0	1	0	1	0	0	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	

Figure 2. 82C206 Internal Decode

The decoder is enabled by three signals. These three signals are ACK, XA9 and XA8. To enable any internal device ACK must be "1" and both XA9 and XA8 must be "0".

The decode scheme employed in the 82C206 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 which are unused by inserting additional peripherals in the I/O map, he may do so since the subsystems in the 82C206 will not respond to the unused address spaces established by the Top Level Decoder. The extra peripherals may be tied directly to the XD0-XD7 data lines since the 82C206 output buffers are not enabled unless an internal subsystem is enabled.

Clock and Wait State Control

The Clock and Wait State 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 Configuration Register located at address 023H.

Writing and reading this register is accomplished by first writing a 01H to location 022H to select the 82C206 Configuration

Register, and then performing either a read or write to location 023H.

Configuration Register (023H) (Index 01H)

msb							
b7	b6	b5	b4	b3	b2	b1	b0
RW1	RW0	16W1	16W0	8W1	8W0	EMR	CLK

RW1-RW0—When the higher speed CPU's are accessing the 82C206, the cycle can be extended by programming up to four wait states into the Configuration Register. This will cause the 82C206 to assert a not ready condition on IOCHRDY (low) whenever a valid decode from the Top Level Decoder is detected and either XIOR or XIOW is asserted. IOCHRDY will remain low for the number of wait states programmed into the Configuration Register bits 6 and 7.

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

Wait states are in increments of one SCLK cycle and are not affected by the DMA Clock Divider.

CHIPS

16W1-16W0—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

8W1-8W0—Wait states may be inserted in 8-bit DMA cycles by programming these two bits in the Configuration Register.

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 82C206. 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 (0) by the peripheral to extend the cycle. The cycle can then be completed by releasing IOCHRDY and allowing it to return high (1).

EMR—This bit enables the extended DMAMEMR function. Normally the assertion of DMAMEMR is delayed one clock cycle later than XIOR in the IBM PC/AT implementation. This may not be desirable in some systems. A "1" programmed into this bit position will start DMAMEMR at the same time as XIOR.

CLK—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.

The Configuration Register contents are pre-loaded by RESET to an initial value of 0C0 hex. This value establishes a default which is IBM 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

DMAMEMR delayed 1 DMA clock cycle later than XIOR

DMA clock is equal to SCLK/2

DMA FUNCTIONAL DESCRIPTION

The equivalent of two 8237A DMA Controllers is implemented in the 82C206. Each controller is a four channel DMA device which will generate 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 of the two DMA devices, thereby maintaining IBM PC/AT compatibility.

DMA cycle length control is provided internally in the 82C206 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.

CHIPS

From this point on the description of the DMA subsystem pertains to both DMA1 and DMA2 unless otherwise noted.

DMA Operation

During normal operation of the 82C206, the DMA subsystem will be in either the Idle condition, the Program condition or the Active condition. In the Idle condition the DMA controller will be executing cycles consisting of only one state. The idle state SI is the default condition and the DMA will remain 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 82C206 will generate 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 store data from the read operation in an internal register. The contents of this register is then written 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 82C206 will assert both DMAMEMR and XIOW allowing data to be transferred directly to the requesting device from memory. Note that 82C206 does not latch data from nor drive data out on this type of cycle.

The number of clock cycles required to transfer a word of data may be varied by programming the DMA or, optionally extended by the peripheral device. During an Active cycle the DMA will sequence through a series of states. Each state will be one DMA clock cycle in

length and the number of states in a cycle will vary depending on how the device is programmed and what type of cycle is being performed. The states are labeled S0-S4 and will be explained in detail in the section entitled Active Condition.

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 82C206 will sample the DREQ input pins every clock cycle. The internal select from the top level decoder and HLDA are also sampled at the same time to determine if the CPU is attempting to access the internal registers. When either of the above two situations occurs, the DMA will exit 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 an internal select is active. The internal select is derived from the top level decode described previously. During this time address lines XA0-XA3 become inputs if DMA1 is selected, or XA1-XA4 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. Figure 3 lists the register address assignment. Due to the large number of internal registers in the DMA subsystem, an internal flip-flop is used to supplement the addressing of the count and address registers. This bit is used to select between the high and low bytes of these registers. The flip-flop will toggle each time a read or write occurs to any of the word count or address registers in the DMA. This internal flip-flop will be cleared by hardware RESET or a Master Clear command and may be set or cleared by the CPU issuing the appropriate command.

Special commands are supported by the DMA subsystem in the Program condition to control the device. These commands do not make use of the data bus but are derived from a set of addresses, the internal select and XIOW or

ADDRESS OPERATION

DMA1	DMA2	XIOR	XIOW	Flip-Flop	Register Function
000h	0C0h	0	1	0	Read Channel 0 Current Address Low Byte
		0	1	1	Read Channel 0 Current Address High Byte
		1	0	0	Write Channel 0 Base and Current Address Low Byte
		1	0	1	Write Channel 0 Base and Current Address High Byte
001h	0C2h	0	1	0	Read Channel 0 Current Word Count Low Byte
		0	1	1	Read Channel 0 Current Word Count High Byte
		1	0	0	Write Channel 0 Base and Current Word Count Low Byte
		1	0	1	Write Channel 0 Base and Current Word Count High Byte
002h	0C4h	0	1	0	Read Channel 1 Current Address Low Byte
		0	1	1	Read Channel 1 Current Address High Byte
		1	0	0	Write Channel 1 Base and Current Address Low Byte
		1	0	1	Write Channel 1 Base and Current Address High Byte
003h	0C6h	0	1	0	Read Channel 1 Current Word Count Low Byte
		0	1	1	Read Channel 1 Current Word Count High Byte
		1	0	0	Write Channel 1 Base and Current Word Count Low Byte
		1	0	1	Write Channel 1 Base and Current Word Count High Byte
004h	0C8h	0	1	0	Read Channel 2 Current Address Low Byte
		0	1	1	Read Channel 2 Current Address High Byte
		1	0	0	Write Channel 2 Base and Current Address Low Byte
		1	0	1	Write Channel 2 Base and Current Address High Byte
005h	0CAh	0	1	0	Read Channel 2 Current Word Count Low Byte
		0	1	1	Read Channel 2 Current Word Count High Byte
		1	0	0	Write Channel 2 Base and Current Word Count Low Byte
		1	0	1	Write Channel 2 Base and Current Word Count High Byte
006h	0CCh	0	1	0	Read Channel 3 Current Address Low Byte
		0	1	1	Read Channel 3 Current Address High Byte
		1	0	0	Write Channel 3 Base and Current Address Low Byte
		1	0	1	Write Channel 3 Base and Current Address High Byte
007h	0CEh	0	1	0	Read Channel 3 Current Word Count Low Byte
		0	1	1	Read Channel 3 Current Word Count High Byte
		1	0	0	Write Channel 3 Base and Current Word Count Low Byte
		1	0	1	Write Channel 3 Base and Current Word Count High Byte
008h	0D0h	0	1	X	Read Status Register
		1	0	X	Write Command Register
009h	0D2h	0	1	X	Read DMA Request Register
		1	0	X	Write DMA Request Register
00Ah	0D4h	0	1	X	Read Command Register
		1	0	X	Write Single Bit DMA Request Mask Register
00Bh	0D6h	0	1	X	Read Mode Register
		1	0	X	Write Mode Register
00Ch	0D8h	0	1	X	Set Byte Pointer Flip-Flop
		1	0	X	Clear Byte Pointer Flip-Flop
00Dh	0DAh	0	1	X	Read Temporary Register
		1	0	X	Master Clear
00Eh	0DCh	0	1	X	Clear Mode Register Counter
		1	0	X	Clear All DMA Request Mask Register Bits
00Fh	0DEh	0	1	X	Read All DMA Request Mask Register Bits
		1	0	X	Write All DMA Request Mask Register Bits

Figure 3. DMA Registers

CHIPS

XIOR. These commands are Master Clear, Clear Mask Register, Clear Mode Register Counter, Set and Clear Byte Pointer Flip-Flop.

The 82C206 will enable 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 82C206 can occur if a request for service occurs on an unmasked channel which is being programmed. The channel should be masked or the DMA disabled to prevent the 82C206 from attempting to service a device with a channel which is partially programmed.

Active Condition

The 82C206 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 82C206 will then begin a DMA transfer cycle.

In a read cycle for example, after receiving a DREQ, the 82C206 will issue a HRQ to the system. Until a HLDA is returned the DMA will remain in an idle condition. On the next clock cycle the DMA will exit Idle and enter state S0. During S0 the device will resolve priority and issue DACK on the highest priority channel requesting service. The DMA will then proceed to state S1 where the multiplexed addresses are output and latched. State S2 is then entered, at which time the 82C206 will assert DMAMEMR. The device then transitions into S3 where the XIOW command is asserted. The 82C206 DMA will then remain in S3 until the Wait State Counter has decremented to zero and IOCHRDY is true. Note that at least one additional S3 will occur unless Compressed Timing is selected. Once a ready condition is detected, the DMA will enter S4 where both commands are deasserted. In Burst Mode and Demand Mode (discussed below), subsequent cycles will begin in S2 unless the intermediate addresses require updating. In these subsequent cycles the lower addresses are changed in S2.

The DMA can be programmed on a channel

by channel basis to operate in one of four modes. The four modes are listed below.

Single Transfer Mode—This mode directs the DMA to execute only one transfer cycle at a time. DREQ must be held active until DACK becomes active. If DREQ is held active throughout the cycle, the 82C206 will deassert HRQ and release the bus once the transfer is complete. After HLDA has gone inactive the 82C206 will again assert HRQ and execute 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 T/C pulse is generated. If the autoinitialization option has been enabled, the channel will reinitialize itself. If Autoinitialize is not selected the DMA will set the DMA request bit mask and suspend transferring on that channel.

Block Transfer Mode—When Block Transfer Mode is selected, the 82C206 will begin transfers in response to either a DREQ or a software request and will continue until a terminal count (FFFFh) is reached, at which time T/C is pulsed and the status register terminal count bit is set. In this mode DREQ need only be held active until DACK is asserted. Auto-initialization is optional in this mode also.

Demand Transfer Mode—In Demand Transfer mode the DMA will begin transfers in response to the assertion of DREQ and will continue until either terminal count is reached or DREQ becomes inactive. This mode is normally used for peripherals which have limited buffering ability. 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 regis-

ters. Once DREQ has been deasserted, higher priority channels are allowed to intervene. Reaching terminal count will result in the generation of a T/C 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 a HLDA from the CPU in response to a DREQ caused by the HRQ from a slave DMA controller, the master DMA controller will ignore all inputs except HLDA from the CPU and DREQ on the active channel. This prevents conflicts between the DMA devices.

Figure 4 shows the cascade interconnection for two levels of DMA devices. Note that Channel 0 of DMA2 is internally connected for Cascade mode to DMA1. Additional devices can be cascaded to the available channels in either DMA1 or DMA2 since cascade is not limited to two levels of DMA controllers.

When programming cascaded controllers, begin with the device which is actually generating HRQ to the system (first level device) and then proceed to the second level devices. RESET causes the DACK outputs to become active low and are placed 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 82C206 has an inverter between DACK0 of DMA2 and HLDA of DMA1. The first level device's DMA request mask bits will prevent second level cascaded devices from generating unwanted hold requests during the initialization process.

DMA Transfers

Four types of transfer modes are provided in the 82C206 DMA subsystem. These transfer types are:

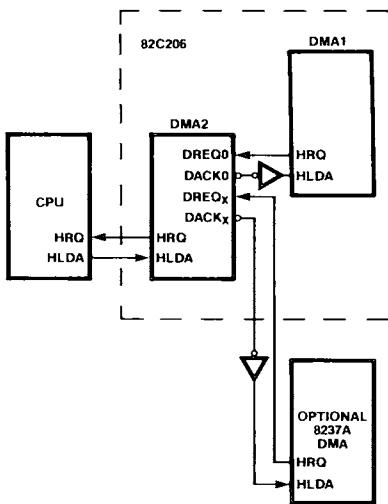


Figure 4. Cascade Mode Interconnect

Read Transfer—Read transfers move data from memory to an I/O device, by generating the memory address and asserting DMAMEMR and XIOW during the same cycle.

Write Transfer—Write transfers move data from an I/O device to memory by generating the memory address and asserting XIOW and DMAMEMW.

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 a bit in the Command Register. Once programmed to perform a memory-to-memory transfer the process can be started by generating either a software 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

CHIPS

latched in the internal Temporary Register of the 82C206. The contents of this register are then output on the XD0-7 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 82C206 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.

Autoinitialization

Each of the four DMA channel Mode Registers contains a bit which will cause the channel 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 registers remain 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.

DREQ Priority

The 82C206 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 will be 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 Figure 5.

In instances where multiple requests occur at the same time the 82C206 will issue a HRQ but will not freeze the priority logic until HLDA is returned. Once HLDA becomes active the priority logic is frozen and DACK is asserted on the highest requesting channel. Priority will not be re-evaluated until HLDA has been deactivated.

Address Generation

Eight intermediate bits of the address are multiplexed onto the data lines during Active cycles of the DMA. This reduces the number of pins required by the DMA subsystem. During state S1, the intermediate addresses are output on data lines XD0-XD7. These addresses should be externally latched and used to drive the system address bus. Since DMA1 is used to transfer 8-bit data and DMA2 is used to transfer 16-bit data, a one bit skew occurs in the intermediate address fields. DMA1 will therefore output addresses A8-A15 on the data bus at this time whereas DMA2 will output A9-A16. A separate set of latch and enable signals are provided for both DMA1 and DMA2 to accommodate the address skew.

During 8-bit DMA cycles, in which DMA1 is active, the 82C206 will output the lower 8-bits

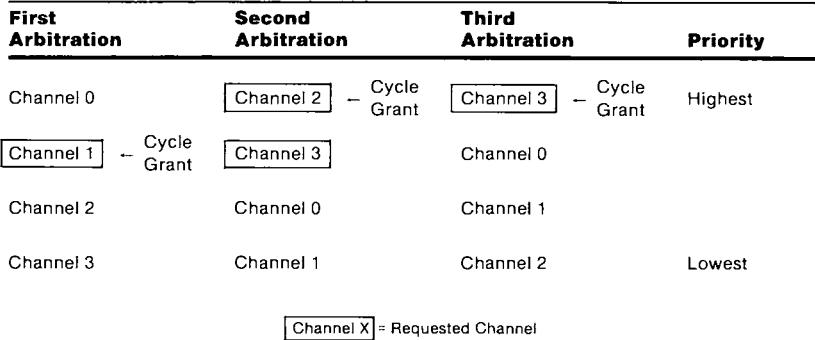


Figure 5. Rotating Priority Scheme

of address on XA0-XA7. The intermediate 8-bits of address will be output on XD0-XD7 and ADSTB8 will be asserted for one DMA clock cycle. The falling edge of ADSTB8 is used to latch the intermediate addresses A8-A15. An enable signal, AEN8, is used to control the output drivers of the external latch. A16-A23 are also generated at this time from a DMA Page Register in the 82C206. Note that A16 is output on the XA16 pin of the device.

16-bit DMA cycles from DMA2 require the 82C206 to output the lower 8-bits of the address on XA1-XA8. The intermediate addresses A9-A16 are output on XD0-XD7. Control for a separate latch is provided by signals ADSTB16 and AEN16. The DMA Page Register now generates A17-A23. During 16-bit DMA transfers XA0 and XA16 remain inactive.

The DMA Page Register is a set of 16 8-bit registers in the 82C206 which are used to generate the high order addresses during DMA cycles. Only 8 of the registers are actually used but all 16 were 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 Figure 6 along with its Read/Write address.

Address	Register Function
080h	Unused
081h	8-bit DMA Channel 2 (DACK2)
082h	8-bit DMA Channel 3 (DACK3)
083h	8-bit DMA Channel 1 (DACK1)
084h	Unused
085h	Unused
086h	Unused
087h	8-bit DMA Channel 0 (DACK0)
088h	Unused
089h	16-bit DMA Channel 2 (DACK6)
08Ah	16-bit DMA Channel 3 (DACK7)
08Bh	16-bit DMA Channel 1 (DACK5)
08Ch	Unused
08Dh	Unused
08Eh	Unused
08Fh	Refresh Cycle

Figure 6.
DMA Address Extension Register Map

During Demand and Block Transfers, the 82C206 generates multiple sequential transfers. For most of these transfers the informa-

CHIPS

tion in the external address latches will remain 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 82C206 will only update the latch contents when necessary. The 82C206 will therefore, only execute S1 cycles when necessary, resulting in an overall through-put improvement.

Compressed Timing

The DMA subsystem in the 82C206 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 higher through-put, the 82C206 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, T/C will be output in S2 and S1 cycles will be executed as necessary to update the address latch. Note that Compressed Timing is not allowed for memory-to-memory transfers.

Register Description

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 will be 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 will be one greater than the value programmed into the register. The register is decremented after each transfer until it goes from zero to FFFFh. When this roll-over occurs the 82C206 will generate T/C and either suspend operation on that channel and set the appropriate Request Mask Bit or Autoinitialize and continue.

Base Address Register

Associated with each Current Address Register is a Base Address Register. This is a write only register which is loaded by the CPU when writing to the Current Address Register. The purpose of this register is to store the initial value of the Current Address Register for Autoinitialization. The contents of this register are loaded into the Current Address Register whenever terminal count is reached and the Autoinitialize Bit is set.

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 into 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 RESET or a Master Clear command.

msb	b6	b5	b4	b3	b2	b1	lsb b0
DAK	DRQ	EW	RP	CT	CD	AH	M-M

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

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

CHIPS

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

RP—Writing a 1 to bit 4 causes the 82C206 to utilize a rotating priority scheme for honoring DMA requests. The default condition is fixed priority.

CT—Compressed timing is enabled by writing a 1 to bit 3 of this register. The default 0 condition causes the DMA to operate with normal timing.

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

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

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

Mode Register

Each DMA channel has a Mode Register associated with it. All four Mode Registers reside at the same I/O address. Bits 0 and 1 of the Write Mode Register command determine which channel's Mode Register gets 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 operations, bits 0 and 1 will both be 1.

msb								lsb	
b7	b6	b5	b4	b3	b2	b1	b0		
M1	M0	DEC	AI	TT1	TT0	CS1	CS0		
(Read/Write Register)									

M1-M0—Mode selection for each channel is accomplished by bits 6 and 7.

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

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

AI—The Autoinitialization function is enabled by writing a 1 in bit 4 of the Mode Register.

TT1-TT0—Bits 2 and 3 control the type of transfer which is to be performed.

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

CS1-CS0—Channel Select bits 1 and 0 determine which channel's Mode Register will be written. Read back of a mode register will result in bits 1 and 0 both being ones.

CS1	CS0	CHANNEL
0	0	Channel 0 select
0	1	Channel 1 select
1	0	Channel 2 select
1	1	Channel 3 select

Request Register

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

CHIPS

reset independently by the CPU. The Request Mask has no effect on software generated requests. All four bits are read in one operation and appear in the lower four bits of the byte. Bits 4 through 7 are read as ones. All four request bits are cleared to zero by RESET.

msb								lsb		
b7	b6	b5	b4	b3	b2	b1	b0			
X	X	X	X	X	RB	RS1	RS0			

(Write Operation)

RB—The request bit is set by writing a 1 to bit 2. RS1-RS0 select which bit (channel) is to be manipulated.

RS1-RS0—Channel Select 0 and 1 determine which channel's Mode Register will be written. Read back of the mode register will result in bits 0 and 1 both being ones.

RS1	RS0	CHANNEL
0	0	Channel 0 select
0	1	Channel 1 select
1	0	Channel 2 select
1	1	Channel 3 select

Format for the Request Register read operation is shown below.

msb								lsb		
b7	b6	b5	b4	b3	b2	b1	b0			
1	1	1	1	RC3	RC2	RC1	RC0			

(Read Operation)

RC3-RC0—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 below.

msb								lsb		
b7	b6	b5	b4	b3	b2	b1	b0			
X	X	X	X	X	MB	MS1	MS0			

(Set/Reset Operation)

MB—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.

MS1-MS0—These two bits select the specific mask bit which is to be set or reset.

MS1	MS0	CHANNEL
0	0	Channel 0 select
0	1	Channel 1 select
1	0	Channel 2 select
1	1	Channel 3 select

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

msb								lsb		
b7	b6	b5	b4	b3	b2	b1	b0			
X	X	X	X	MB3	MB2	MB1	MB0			

(Read/Write Operation)

MB3-MB0—Each bit position in the field represents the mask bit of a channel. The mask bit number corresponds to the channel number associated with the mask bit.

All four mask bits are set following a RESET or a Master Clear command. Individual channel mask bits will be set as a result of terminal count being reached, if Autoinitialize is disabled. The entire register can be cleared,

CHIPS

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 RESET, Master Clear or each time a Status Read takes place. Bits 4-7 are cleared by RESET, 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.

msb	lsb						
b7	b6	b5	b4	b3	b2	b1	b0
DRQ3	DRQ2	DRQ1	DRQ0	TC3	TC2	TC1	TC0

(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 XDO-XD7. During the second cycle of the transfer, the data in the Temporary Register is output on the XDO-XD7 pins. Data from the last memory-to-memory transfer will remain in the register unless a RESET or Master Clear occurs.

Special Commands

Five Special Commands are provided to make the task of programming the device easier. These commands are activated as a result of a specific address and assertion of either a XIOR or XIOW. Information on the data lines is ignored by the 82C206 whenever an XIOW activated command is issued, thus data returned on XIOR 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 registers. 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 RESET, 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 registers will be read is Channel 0 first, Channel 3 last.

INTERRUPT CONTROLLER FUNCTIONAL DESCRIPTION

The programmable interrupt controllers in the 82C206 function 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 used as an index by the CPU to determine which interrupt service routine to execute.

A variety of priority assignment modes are provided, which can be reconfigured at any time during system operation, allowing the complete interrupt subsystem to be restructured, based on the system environment.

Overview

Two interrupt controllers, INTC1 and INTC2, are included in the 82C206. Each of the interrupt controllers is 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 7) for proper operation of all 16 interrupt channels. INTC1 is located at addresses 020H-021H and is configured for Master operation (defined below) in Cascade Mode. INTC2 is a Slave device (defined below) and is located at 0A0H-0A1H. The Interrupt Request output signal from INTC2 (INT) is internally connected to the interrupt request input Channel 2 (IR2) of INTC1. The address decoding and Cascade interconnection match-
es that of the IBM PC/AT.

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. Figure 8 lists the 16 interrupt channels and their interrupt request source.

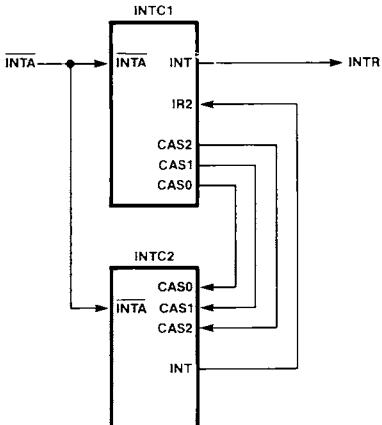


Figure 7. Internal Cascade Interconnect

Description of the Interrupt Subsystem will pertain to both INTC1 and INTC2 unless otherwise noted. Wherever register addresses are used, the address for the INTC1 register will be listed first and the address for the INTC2 register will follow in parenthesis. Example 020H (0A0H)

Controller Operation

Figure 9 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 which are requesting service. Interrupt Request Register bits are labeled using the Channel Name IR7-IR0. 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 IS7-IS0 and correspond to IR7-IR0. The Interrupt Mask Register (IMR) allows the CPU to disable any or all of the interrupt channels. The Priority Resolver evaluates inputs from the above three registers, issues an interrupt request, and latches the corresponding bit into the In-Service Register. During interrupt acknowledge cycles, a master controller outputs a code to the slave device which is compared in the Cascade Buffer/Comparator 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.

Interrupt Sequence

The 82C206 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 82C206 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). The events which occur during an interrupt sequence are as follows:

1—One or more of the interrupt requests



Controller Number	Channel Name	Interrupt Request Source
INTC1	IR0	Counter/Timer Out0
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

Figure 8. Interrupt Request Source

(IR7-IR0) becomes 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 reset. The internal Cascade address is generated and the XD7-XD0 outputs remain tri-stated.

5—The CPU will execute a second INTA cycle, during which the 82C206 will drive an 8-bit vector onto the data pins XD7-XD0, which is in turn latched by the CPU. The format of this vector is shown in Figure 12. Note that V7-V3 in Figure 12 are programmable by writing to Initialization Control

Word 2 (see Initialization Command Words section below).

6—At the end of the second INTA cycle, the ISR bit will be cleared if the Automatic End Of Interrupt mode is selected (see End Of Interrupt section below). Otherwise, the ISR bit must be cleared by an End Of Interrupt (EOI) command from the CPU at the end of the interrupt service routine.

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

End Of Interrupt

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



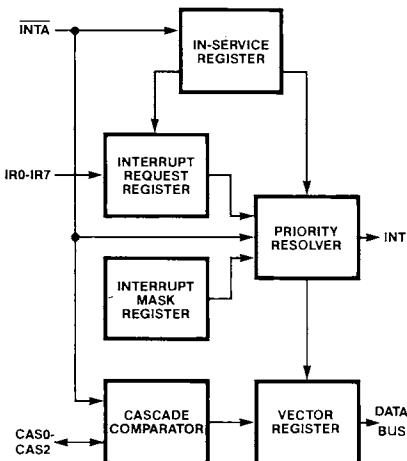


Figure 9. Interrupt Controller Block Diagram

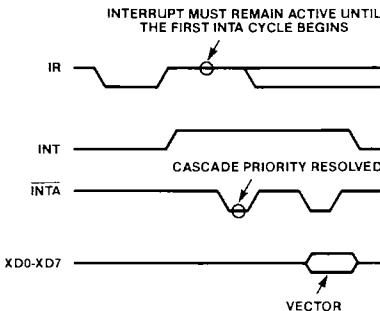


Figure 10. Interrupt Sequence

The 82C206 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 necessarily the last level acknowledged and serviced. In conditions where the fully nested structure is not preserved, a specific EOI must be generated at the end of the interrupt

service routine. An ISR bit that is masked, in Special Mask Mode by a IMR bit, will not be cleared by a non-specific EOI command. The interrupt controller can optionally generate an Automatic End Of Interrupt (AOI) 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, IRO 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:

	Lowest							Highest						
Priority Status	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 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.

Before Rotation

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

(Specific Rotation command issued with
Channel 5 specified)

Channel 3 Spec After Rotation

Priority Status Lowest Highest

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, if enabled, due to the occurrence of EOI (automatic or CPU generated).

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

After Rotation (IR4 service completed)

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

Programming The Interrupt Controller

Two types of commands are used to control the 82C206 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 an initialization sequence and does the following:

- 1—The Initialization Command Word Counter is reset to zero.
 - 2—ICW1 is latched into the device
 - 3—Fixed Priority Mode is selected
 - 4—IR7 is assigned the highest priority
 - 5—The Interrupt Mask Register is cleared
 - 6—The Slave Mode Address is set to 7
 - 7—Special Mask Mode is disabled
 - 8—The IRR is selected for Status Read operations

The next three I/O writes to address 021H (0A1H) will load ICW2-ICW4. See Figure 11 for a flow chart of the initialization sequence. The initialization sequence can be terminated

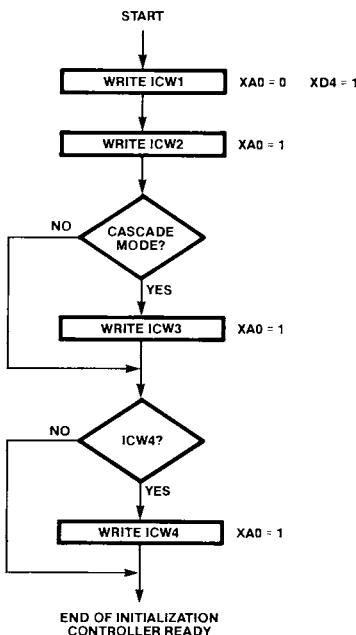


Figure 11. Initialization Sequence

CHIPS

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	V7	V6	V5	V4	V3	1	1	1
IR7	V7	V6	V5	V4	V3	1	1	0
IR7	V7	V6	V5	V4	V3	1	0	1
IR7	V7	V6	V5	V4	V3	0	1	1
IR7	V7	V6	V5	V4	V3	0	1	0
IR7	V7	V6	V5	V4	V3	0	0	1
IR7	V7	V6	V5	V4	V3	0	0	0

Figure 12. Interrupt Vector Byte

at any point (all 4 bytes must be written for the controller to be properly initialized) by writing to address 020H (0AOH) with a 0 in data bit 4. Note, this will cause OCW2 or OCW3 to be written.

ICW1—Address 020H (0AOH)

msb	lsb							
b7	b6	b5	b4	b3	b2	b1	b0	
X	X	X	SI	LTM	X	SM	X	

(Write Only Register)

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

LTM—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 will generate an interrupt request. The IR must be active until the first INTA cycle is started to generate the proper interrupt vector (an IR7 vector will be generated if the IRR input is deasserted early) and the IR must be removed prior to EOI to prevent a second interrupt from occurring.

SM—Bit 1 selects between Single Mode and Cascade Mode. Single Mode is used whenever only one interrupt controller (INTC1) is used and is not recommended for this device. Cascade Mode allows the two interrupt controllers to be connected through IR2 of

INTC1. INTC1 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.

ICW2—Address 021H (0A1H)

msb	lsb							
b7	b6	b5	b4	b3	b2	b1	b0	
V7	V6	V5	V4	V3	X	X	X	

(Write Only Register)

V7-V3—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 12). INTC1 and INTC2 need not be programmed with the same value in ICW2.

ICW3 Format for INTC1—Address 021H

msb	lsb							
b7	b6	b5	b4	b3	b2	b1	b0	
S7	S6	S5	S4	S3	S2	S1	S0	

(Write Only Register)

S7-S0—Select which IR inputs have Slave Mode controllers connected. ICW3 in INTC1 must be written with a 04H for INTC2 to function.

ICW3 Format for INTC2—Address 0A1H

msb	lsb							
b7	b6	b5	b4	b3	b2	b1	b0	
0	0	0	0	0	ID2	ID1	ID0	

(Write Only Register)

ID2-ID0—Determine the Slave Mode address the controller will respond to during the cascaded INTA sequence. ICW3 in INTC2 should be written with a 02H for Cascade Mode operation. Note, b7-b3 should be zero.

CHIPS

ICW4—Address 021H (0A1H)

msb								lsb							
b7	b6	b5	b4	b3	b2	b1	b0								
X	X	X	EMI	X	X	AEOI	X								

(Write Only Register)

EMI—Bit 4 will 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. 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.

AEOI—Auto End Of Interrupt is enabled when ICW4 is written with a zero in bit 1. The interrupt controller will perform 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.

Operational Command Words

Operational Command Words (OCWs) allow the 82C206 interrupt controllers to be controlled or reconfigured at any time while operating. Each interrupt has 3 OCWs which can be programmed to affect the proper operating configuration and a Status Register to monitor controller operation.

Operational Command Word 1 (OCW1) is located at address 021h (0A1h) and may be written any time the controller is not 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 will place the controller in operational mode and load OCW2 (if data bit 3 = 0) or OCW3 (if data bit 3 = 1).

OCW1—Address 021H (0A1H)

msb								lsb							
b7	b6	b5	b4	b3	b2	b1	b0								
M7	M6	M5	M4	M3	M2	M1	M0								

(Read/Write Register)

M7-M0—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 effect on lower priority requests. All IMR bits are cleared by writing ICW1.

OCW2—Address 020H (0A0H)

msb								lsb							
b7	b6	b5	b4	b3	b2	b1	b0								
R	SL	EOI	SI	2/3	L2	L1	LO								

(Write Only Register)

R—This bit in conjunction with SL and EOI selects operational function. Writing a 1 in bit 7 causes one of the rotate functions to be selected.

R	SL	EOI	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.

SL—This bit in conjunction with R and EOI selects operational function. Writing a 1 in this bit position causes a specific or immediate function to occur. All specific commands require L2-L0 to be valid except no operation.

CHIPS

R	SL	EOI	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

EOI—This bit in conjunction with R and SL selects operational function. Writing a 1 in this bit position causes a function related to EOI to occur.

R	SL	EOI	Function
0	0	1	Non-specific EOI Command
0	1	1	Specific EOI Command
1	0	1	Rotate on non-specific EOI
1	1	1	Rotate on specific EOI

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

2/3—If the I/O write places a 0 in bit 4 (SI), then writing a 0 in bit 3 (2/3) selects OCW2 and writing a 1 will select OCW3.

L2-L0—These three bits are internally decoded to select which interrupt channel is to be affected by the Specific command. L2-L0 must be valid during three of the four specific cycles (see SL above).

OCW3—Address 020H (0A0H)

msb	b6	b5	b4	b3	b2	b1	lsb
0	ESMM	SMM	SI	2/3	PM	RR	RIS

(Write Only Register)

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

SMM—If ESMM and SMM both are written with a 1 the Special Mask Mode is enabled. Writing a 1 to ESMM and a 0 to SMM 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 of the ISR.

SI—See SI above.

2/3—See 2/3 above.

PM—Polled Mode is enabled by writing a 1 to bit 2 of OCW3, causing the 82C206 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 IRR will remain frozen until the read cycle is completed at which time the PM bit is reset.

RR—When the RR bit (bit 1) is 1, reading the Status Port at address 020h (0A0h) will cause the contents of IRR or ISR (determined by RIS) to be placed on XD7-XD0. Asserting PM forces RR reset.

RIS—This bit selects between the IRR and the ISR during Status Read operations if RR = 1.

COUNTER/TIMER FUNCTIONAL DESCRIPTION

The Counter/Timer (CTC) in the 82C206 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 13 are controlled from a common set of control logic. The Control Logic decodes control in-

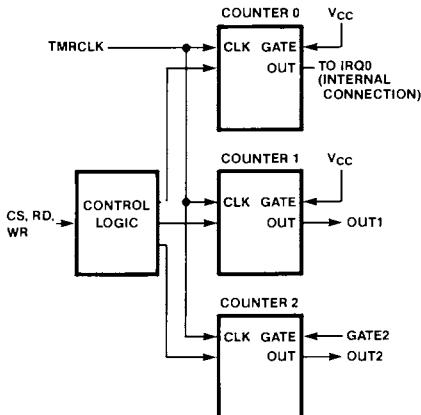


Figure 13. Counter/Timer Block Diagram

formation written to the CTC and provides the controls necessary to load, read, configure and control each counter. Counter 0 and Counter 1 can be programmed for all six modes, but Mode 1 and Mode 5 have limited usefulness due to the lack of an external hardware trigger signal. Counter 2 can be operated in any of six modes listed below.

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

All three counters in the CTC are driven from a common clock input pin (TMRCLK) which is independent from other clock inputs to the 82C206. Counter 0's output (Out0) is connected to IRO of INTC1 (see Interrupt Controller Functional Description) and may be used as an interrupt to the system for time keeping and task switching. Counter 1 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.

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 (OUT0 is not externally accessible). The OUT signal's state and function are controlled by the Counter Mode and condition of the CE (see Mode Definitions).

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

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

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

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

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

CHIPS

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 14). The Control Word is a write only location.

Control Word—(043H)

msb	b6	b5	b4	b3	b2	b1	lsb b0
F3	F2	F1	F0	M2	M1	M0	BCD

(Write Only Register)

F3-F0—Bits 7-4 determine the command to be performed.

M2-M0—Bits 3-1 determine the counter's mode during Read/Write Counter Commands (see Read/Write Counter Command) or select the counter during a Read-Back Com-

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

Figure 14. Counter/Timer Address Map

mand (see Read-Back Command). Bits 3-1 become "don't care" during Latch Counter Commands.

BCD—Bit 0 selects binary coded decimal counting format during Read/Write Counter Commands. Note, during Read-Back Command this bit must be 0.

Read/Write Counter Command

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

1—Each counter's Control Word must be written before the initial count is written.

2—Writing the initial count must follow the

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 significant byte

CHIPS

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 M3-M0 are defined as follows:

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

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

Read-Back Command

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

The format of the Read-Back Command is:

msb	b6	b5	b4	b3	b2	b1	lsb
1	1	LC	LS	C2	C1	C0	0

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

LS—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 will result in the contents of the Status Register being read (see Status Read).

C2-C0—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 latches remain latched until either the latch is read or the counter is reprogrammed.

If LS = LC = 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.

Status Byte

msb	b6	b5	b4	b3	b2	b1	lsb
OUT	NC	F1	F0	M2	M1	M0	BCD

OUT—Bit 7 contains the state of the OUT signal of the counter

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

CHIPS

whenever the counter is loaded from the counter input registers.

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

M2-M1—These bits reflect the mode of the counter and are interpreted in the same manner as in Write Command operations.

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

Counter Operation

Due to the previously stated restrictions in Counter 0 and Counter 1, Counter 2 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 82C206 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 goes back high and remains high until a new count or Control Word is written. Counting is enabled when GATE2 = 1. Disabling the count has no effect on OUT2.

The CE is loaded with the first TMRCLK pulse after the Control Word and initial count

are loaded. When both CIL and CIH are written, the CE is loaded after CIH is written (see Write Operations). This TMRCLK pulse does not decrement the count, so for an initial count of N, OUT2 does not go high until N+1 TMRCLK pulses after initialization. Writing a new initial count to the counter reloads the CE on the next TMRCLK pulse and counting continues from the new count.

If an initial count is written with GATE2 = 0, it will still be 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 cause the CE to be reloaded, extending the length of the pulse. Writing a new count to CIL and CIH will not affect the current one-shot unless the counter is retriggered.

Mode 2—Rate generator

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

When the initial count is decremented to 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 will generate an OUT2 pulse every N TMRCLK cycles. Note that a count of 1 is illegal in Mode 2.

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

Writing a new count while counting does not

CHIPS

affect current operation unless a trigger is received. Otherwise, the new count will be 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 the new count. This allows the sequence to be "retriggerable" 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. GATE2 = 0 disables counting.

The CE is loaded on the TMRCLK pulse after a trigger. Since loading the CE inhibits decrementing, OUT2 will go low for one TMRCLK cycle, $(N+1)$ TMRCLK cycles after the trigger.

If a new count 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 "retriggerable".

Mode		Condition	
		Low	Rising
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	—

Figure 15. Gate Pin Function

CHIPS

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 flop-flop resets immediately after being sampled. Note that in Modes 2 and 3 the GATE2 input is both edge and level sensitive.

REAL TIME CLOCK FUNCTIONAL DESCRIPTION

This section of the 82C206 combines a complete time-of-day clock with alarm, one hundred year calendar, a programmable periodic interrupt, and 114 bytes of low power static RAM. Provisions are made to enable the device to operate in a low power (battery powered) mode and protect the contents of both the RAM and clock during system power-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 and then strobing the AS input pin. The address will then be latched into the Index Address Register on the falling edge of AS. 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 XIOR or XIOW with an address on the XA9-XA0 inputs of 071H.

Since AS will most likely be generated by an I/O operation which will result in the assertion of XIOW, it is recommended that an address of 070H be applied to the XA9-XA0 inputs during this time. This will prevent the modification of other registers in the 82C206.

Address Map

Figure 16 illustrates the internal register/RAM organization of the Real Time Clock portion of the 82C206. The 128 addressable locations in the Real Time Clock are divided into 10

bytes which normally contain the time, calendar, and alarm data, four control and status bytes and 114 general purpose RAM bytes. All 128 bytes are readable by the CPU. The CPU may also write to all locations except Registers C, D, Bit 7 of Register A and Bit 7 of the Seconds Byte which is always 0.

Index	Function
00	SECONDS
01	SECONDS ALARM
02	MINUTES
03	MINUTES ALARM
04	HOURS
05	HOURS ALARM
06	DAY OF WEEK
07	DATE OF MONTH
08	MONTH
09	YEAR
0A	REGISTER A
0B	REGISTER B
0C	REFISTER C
0D	REGISTER D
0E	USER RAM
0F	USER RAM
.	.
7E	USER RAM
7F	USER RAM

Figure 16. Address Map for Real Time Clock

Time Calendar and Alarm Bytes

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

Before initialization of the internal registers can be performed, the SET bit in Register B should be set to a "1" to prevent Real Time Clock updates from occurring. The CPU then initializes the first 10 locations in 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.

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

Figure 17. Time, Calendar,
Alarm Data Format

Figure 17 shows the format for the ten clock, calendar and alarm locations. The 24/12 bit in Register B 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 sec-

ond, 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 an update 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 every hour by programming a C0H into Register 5, or an interrupt can be generated once a second by programming the same value into all three alarm registers.

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 nonvolatile storage for configuration and calibration parameters since this device is normally battery powered when the system is turned off.

Control and Status Registers

The 82C206 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.

REGISTER A (0AH)

msb	lsb							
b7	b6	b5	b4	b3	b2	b1	b0	
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0	

(Read/Write register except UIP)

CHIPS

UIP—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.

DV2-DV0—These three bits are used to control the Divider/Prescaler on the Real Time Clock. While the 82C206 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.

DV2	DV1	DVO	OSCI Freq.	Mode
0	0	0	4.194304MHz	Operate
0	0	1	1.048576MHz	Operate
0	1	0	32.768KHz	Operate
1	1	X		Reset Divider

Divider Options

RS3-RS0—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. Below are the interrupt rates for which the Real Time Clock can be programmed.

Rate Selection				Time Base	
RS3	RS2	RS1	RS0	4.194304 MHz	1.048576 MHz
0	0	0	0	None	None
0	0	0	1	30.517 μ s	3.90526 ms
0	0	1	0	61.035 μ s	7.8125 ms
0	0	1	1	122.070 μ s	122.070 μ s
0	1	0	0	244.141 μ s	244.141 μ s
0	1	0	1	488.281 μ s	488.281 μ s
0	1	1	0	976.562 μ s	976.562 μ s
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

Periodic Interrupt Rate

CHIPS

REGISTER B (0BH)

msb								lsb							
b7	b6	b5	b4	b3	b2	b1	b0								
SET	PIE	AIE	UIE	0	024/12DSE										

(Read/Write Register)

SET—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. The SET bit is not affected by the RESET input pin.

PIE—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.

AIE—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 will enable the generation of periodic interrupts at rates of one second or greater. This bit is cleared by Reset.

24/12—The 24/12 control bit is used to establish the format of both the Hours and Hours Alarm bytes. If this bit is a "1", the Real Time Clock will interpret and update the 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.

DSE—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. PSRSTB has no affect on this bit.

REGISTER C (0CH)

msb								lsb							
b7	b6	b5	b4	b3	b2	b1	b0								
IRQF	PF	AF	UF	0	0	0	0								

(Read only register)

IRQF—The Interrupt Request Flag bit is set to a "1" when any of the conditions which can cause an interrupt is 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:

$$\begin{aligned} \text{IRQF} = & \text{ PF} \& \text{ PIE} \\ & + \text{ AF} \& \text{ AIE} \\ & + \text{ UF} \& \text{ UIE} \end{aligned}$$

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.

PF—The Periodic 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".

AF—A "1" appears in the AF bit whenever a match has occurred between the time registers and alarm registers during an update cycle. This flag is also independent of its enable (AIE) and will generate an interrupt if AIE is true.

REGISTER D (0DH)

msb								lsb							
b7	b6	b5	b4	b3	b2	b1	b0								
VRT	0	0	0	0	0	0	0								

(Read only register)

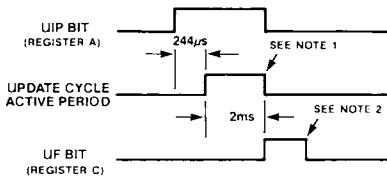
VRT—The Valid RAM and Time Bit indicates the condition of the contents of the Real Time Clock. This bit is cleared to a "0" whenever the PS input pin is LOW. This pin

CHIPS

is normally derived from the 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 affect on this bit and it can only be set by reading Register D. All unused register bits will be "0" when read and are not writable.

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 DV2-DV0 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.



NOTE:

1. REGISTERS 0-9 ARE UNAVAILABLE TO BE READ OR WRITTEN DURING THIS TIME.
2. UF BIT CLEARED BY CPU READ OF REGISTER C

Figure 18. Update Cycle

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 the reading of incorrect data. To avoid contention problems between the Real Time Clock and the CPU, a flag is provided in Register A to alert the user of an impending update cycle. This Update In Process Bit (UIP) is asserted 244 μ s 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 18 illustrates the update cycle. CPU access is always allowed to Registers A through D during update cycles.

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

The first method is to read Register A, determine the state of the UIP bit 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 244 μ s 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 will become 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 will 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 82C206. 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 19 may be used to eliminate power drain on the battery when the entire 82C206 is active. The circuit shown here will allow for reliable transitions between system and battery power without undue battery power drain.

CHIPS

The user should also ensure that the V_{in} maximum specification is never exceeded when powering the system up or down. Failure to observe this specification may result in damage to the device.

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

One pin is provided to initialize the device whenever power is applied to the 82C206. This pin (PSRSTB) will not alter the RAM or Clock/Calendar contents but it will initialize the necessary control register bits. (See Pin Description for a list of the control register

bits affected by PSRSTB) Assertion of PSRSTB disables the generation of interrupts and sets a flag indicating that the contents of the device may not be valid. A recommended circuit for controlling the PSRSTB input is also shown in Figure 19.

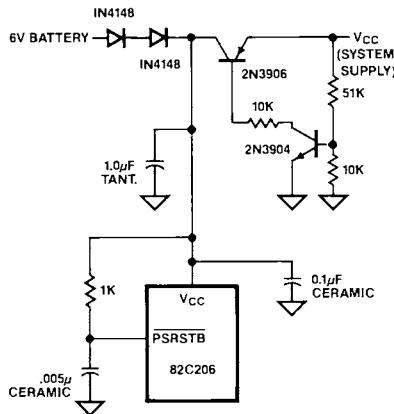


Figure 19. Power Conversion and Reset Circuitry

CHIPS

82C206 Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V_{CC}	—	7.0	V
Input Voltage	V_I	-0.5	5.5	V
Output Voltage	V_O	-0.5	5.5	V
Operating Temperature	T_{OP}	-25	85	C
Storage Temperature	T_{STG}	-40	125	C

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

82C206 Operating Conditions

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V_{CC}	4.75	5.25	V
Ambient Temperature	T_A	0	70	C

DC Characteristics ($T_A = 0\text{--}70^\circ\text{C}$, $V_{CC} = 5 \pm 5\%$)

Symbol	Parameter	Min.	Max.	Units	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	$V_{CC} + 0.5$	V	
V_{OL}	Output Low Voltage	0.4	V		$I_{OL} = 2.0 \text{ mA}$
V_{OH}	Output High Voltage	2.4	V		$I_{OH} = -2.0 \text{ mA}$
I_{OL}	Input Current	-10	10	μA	$V_{IN} = V_{CC}$ to 0V
I_{OL}	Output Leakage Current	-10	10	μA	$V_{OU} = V_{CC}$ to 0.45
I_{CC}	V_{CC} Supply Current	30	mA		CLK Freq = 8 MHz
I_{CCSB}	V_{CC} Standby Supply Current	10	μA		CLK Freq = DC

Capacitance ($T_A = 25^\circ\text{C}$, $V_{CC} = \text{GND} = 0\text{V}$)

Symbol	Parameter	Min.	Max.	Units	Test Conditions
C_{IN}	Input Capacitance	10	pF		$FC = 1 \text{ MHz}$
$C_{I/O}$	I/O Capacitance	20	pF		Unmeasured pins
C_{OUT}	Output Capacitance	20	pF		Returned to V_{SS}

AC Characteristics ($T_A = 0\text{--}70^\circ\text{C}$, $V_{CC} = 5\text{V} \pm 5\%$)

Symbol	Parameter	6 MHz		8 MHz		Units
		Min.	Max.	Min.	Max.	
t1	Address Setup to Command Active	25		25		nsec
t2	Command Active Period	250		200		nsec
t3	Address Hold Time from Command Inactive	0		0		nsec
t4	Data Valid Delay	200		160		nsec
t5	Data Hold Time from XIOR Inactive	10		10		nsec
t6	XDO-XD7 Active from XIOR	5	40	5	40	nsec
t7	Data Setup to XIOW Inactive	200		160		nsec
t8	Data Hold Time from XIOW Inactive	0		0		nsec
t9	Command Recovery Time	150		120		nsec
t10	Interrupt Request width (Low)	120		100		nsec
t11	Interrupt Request width (High)	250		200		nsec
t12	INT Output Delay	400		300		nsec
t20	Real Time Clock Cycle Time	500		500		nsec
t21	AS Pulse Width	200		160		nsec
t22	Data Valid Setup to AS Inactive	200		160		nsec
t23	Data Hold Time from AS Inactive	0		0		nsec
t24	OSCI Period	500		500		nsec
t25	OSCI High Time	200		200		nsec
t26	OSCI Low Time	200		200		nsec
t27	PSRSTB High Delay from V_{CC}	5		5		μsec
t28	PSRSTB Low Pulse Width	5		5		μsec
t29	VRT Bit Valid Delay	2		2		μsec
t40	TMRCLK Period	200	DC	125	DC	nsec
t41	TMRCLK Low Time	80		50		nsec
t42	TMRCLK High Time	80		50		nsec
t43	GATE2 Setup to TMRCLK	80		50		nsec
t44	GATE2 Hold Time from TMRCLK	80		50		nsec
t45	GATE2 Low Time	80		50		nsec
t46	GATE2 High Time	80		50		nsec
t47	OUT2 Delay from TMRCLK	200		120		nsec
t48	OUT2 Delay from GATE2	200		120		nsec

CHIPS

AC Characteristics ($T_A = 0\text{--}70^\circ\text{C}$, $V_{CC} = 5\text{V} \pm 5\%$) (Continued)

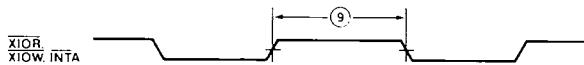
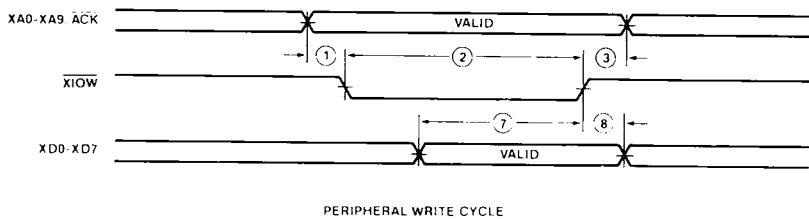
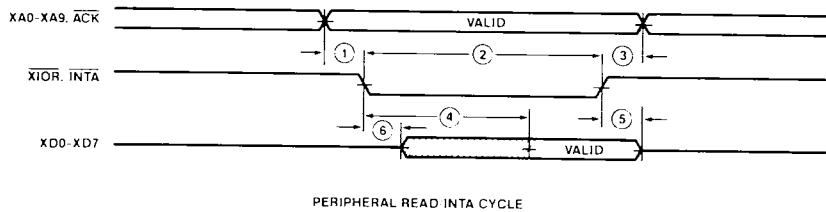
Symbol	Parameter	6 MHz		8 MHz		Units
		Min.	Max.	Min.	Max.	
t50	SCLK Period (1 X SCLK)	186		125		nsec
t50A	SCLK Period (2 X SCLK)	93		62		nsec
t51	SCLK Low Time (1 X SCLK)	75		43		nsec
t51A	SCLK Low Time (2 X SCLK)	32		22		nsec
t52	SCLK High Time (1 X SCLK)	82		55		nsec
t52A	SCLK High Time (2 X SCLK)	40		27		nsec
t53	DREQx Setup to SCLK	0		0		nsec
t54	HRQ Valid From SCLK		120		75	nsec
t55	HLDA Setup to SCLK		75		45	nsec
t56	AENx Valid Delay from SCLK		175		105	nsec
t57	AENx Invalid Delay from SCLK	TBD	130	TBD	80	nsec
t58	ADSTBx Valid Delay from SCLK		80		50	nsec
t59	ADSTBx Invalid Delay from SCLK		120		120	nsec
t60	XD0-XD7 Active Delay from SCLK		110		60	nsec
t61	XD0-XD7 Valid Setup to ADSTBx Low		80		65	nsec
t62	XD0-XD7 Hold Time from ADSTBx Low		25		25	nsec
t63	XD0-XD7 Tristate Delay from SCLK			170	135	nsec
t64	Address Valid Delay from SCLK			110	60	nsec
t65	Address Hold Time from DMAMEMR High		66		50	nsec
t66	Address Tristate Delay from SCLK			90	55	nsec
t67	DACKx Delay from SCLK		170		105	nsec
t68	Command Enable Delay from SCLK			150	90	nsec
t69	Command Active Delay from SCLK			190	120	nsec
t70	Write Cmd Inactive Delay from SCLK			130	80	nsec
t71	Address Hold Time from Write High		116		75	nsec
t72	Command Tristate Delay from SCLK			120	75	nsec

**AC Characteristics** ($T_A = 0\text{--}70^\circ\text{C}$, $V_{CC} = 5\text{V} \pm 5\%$) (Continued)

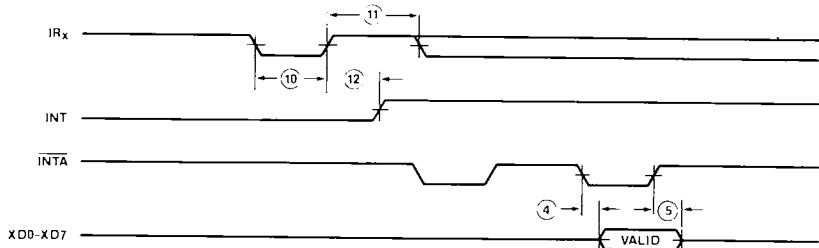
Symbol	Parameter	6 MHz		8 MHz		Units
		Min.	Max.	Min.	Max.	
t73	Read Cmnd Inactive Delay from SCLK		190		115	nsec
t74	TC Delay from SCLK		100		60	nsec
t75	XD0-XD7 Setup to Read Cmnd Inactive	155		90		nsec
t76	XD0-XD7 Hold from Read Cmnd Inactive	0		0		nsec
t77	XD0-XD7 Valid Delay from SCLK		190		120	nsec
t78	XD0-XD7 Hold from Write Inactive	15		15		nsec
t79	IOCHRDY Input Setup to SCLK	50		35		nsec
t80	IOCHRDY Input Hold Time from SCLK	35		20		nsec

CHIPS

82C206 TIMING DIAGRAMS

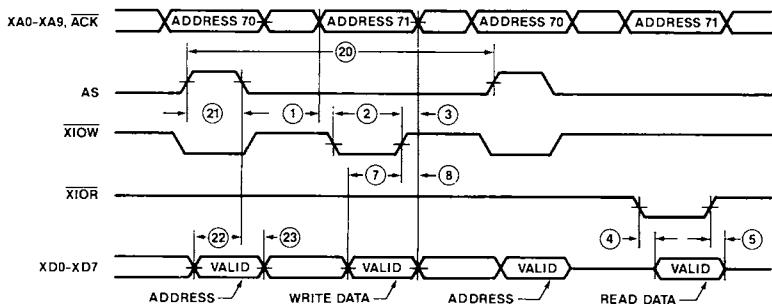
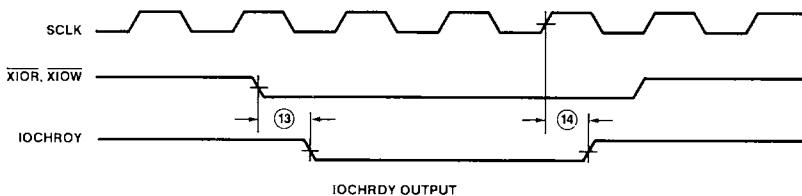


COMMAND RECOVERY

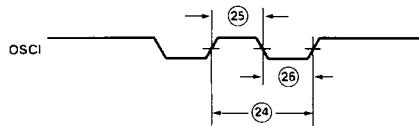


INTA SEQUENCE

82C206 TIMING DIAGRAMS



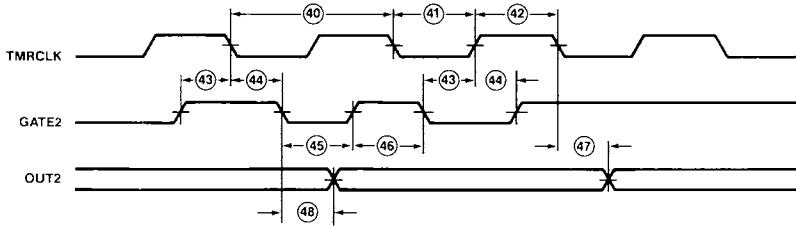
REAL TIME CLOCK ACCESS CYCLE



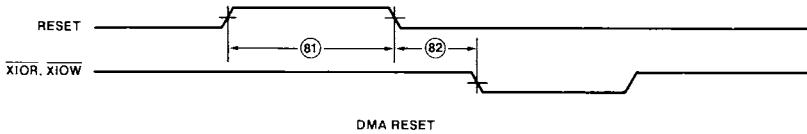
REAL TIME CLOCK POWER-UP SEQUENCE

CHIPS

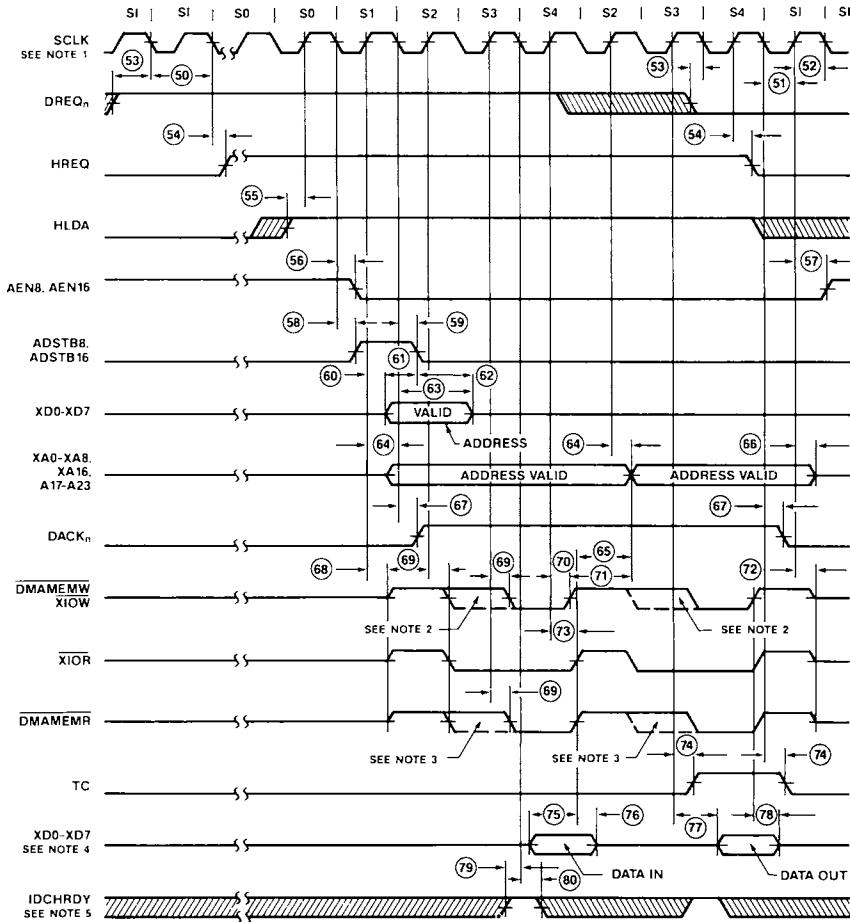
82C206 TIMING DIAGRAMS



COUNTER/TIMER PARAMETERS



82C206 TIMING DIAGRAMS



NOTES:

- All timings referenced to SCLK are independent of the state of the clock select bit in the configuration register. SCLK shown in this diagram is the undivided clock directly from the input.
- Extended Write mode selected.
- Extended Read mode selected.
- Data Bus during memory to Memory Transfer.
- IOCHRDY Input Timing.

CHIPS

Load Circuit Measurement Conditions

Parameter	Output Type	Symbol	C_L (pF)	R_1 (Ω)	R_L (Ω)	SW_1	SW_2
Propagation Delay Time	Totem pole 3-state	t_{PLH} t_{PHL}	50	—	1.0K	OFF	ON
Time	Bidirectional						
Propagation Delay Time	Open drain or Open Collector	t_{PLH} t_{PHL}	50	0.5K	—	ON	OFF
Disable Time	3-state Bidirectional	t_{PZL} t_{PHZ}	5	0.5K	1.0K	ON OFF	ON
Enable Time	3-state Bidirectional	t_{PZH} t_{PZL}	50	0.5K	1.0K	ON OFF	ON

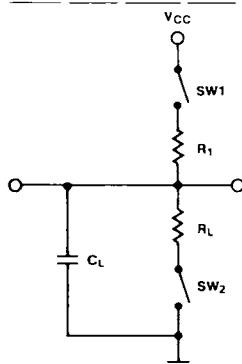
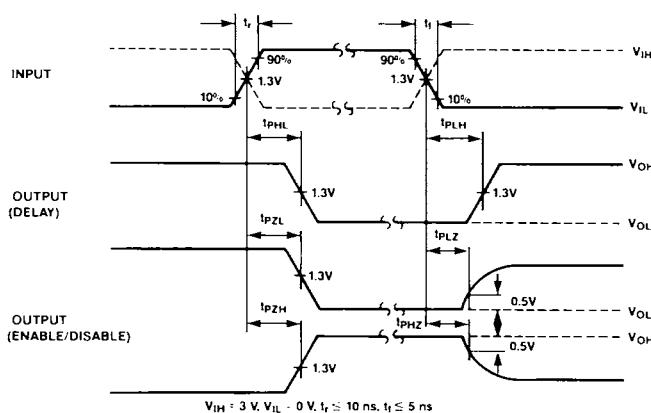
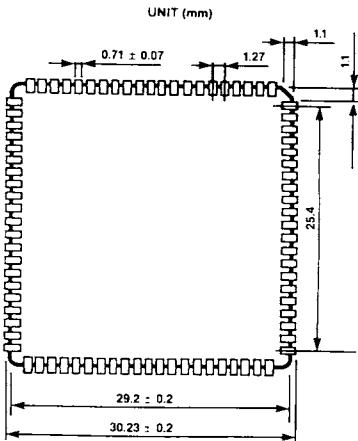
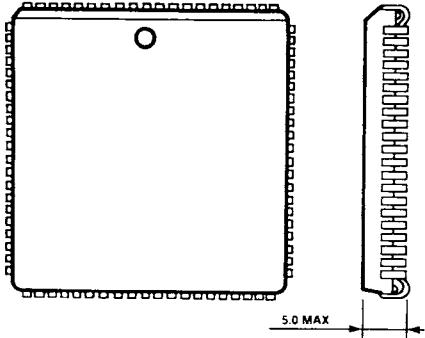


Figure 10. Load Circuit and AC Characteristics Measurement Waveform

CHIPS

84-PIN PLASTIC LEADED CHIP CARRIER



Ordering Information

Order Number	Package Type	Operating Range
P82C206	PLCC-84	C (Note 1)

NOTE:

1. PLCC-84 Plastic Leaded Chip Carrier 84-Pin Package
C = Commercial Range, 0-70°C, ±5% Supply Voltage



CHIPS



CHIPS

Chips and Technologies, Incorporated
521 Cottonwood Drive, Milpitas, CA 95035 408-434-0600 Telex 272929 CHIPS UR

IBM, AT are the trademarks of International Business Machines, San Jose.
INTEL and iAPX 86 is a trademark of Intel Corporation, Santa Clara.

MOTOROLA™ is a trademark of Motorola.

CHIPSet is a trademark of Chips and Technologies, Incorporated.

CHIPS makes no warranty for the use of its products and bears no responsibility for any errors
which may appear in this document.



82C301, 82C302, 82A303, 82A304, 82A305, 82A306 CS8230: AT/386 CHIPSet™

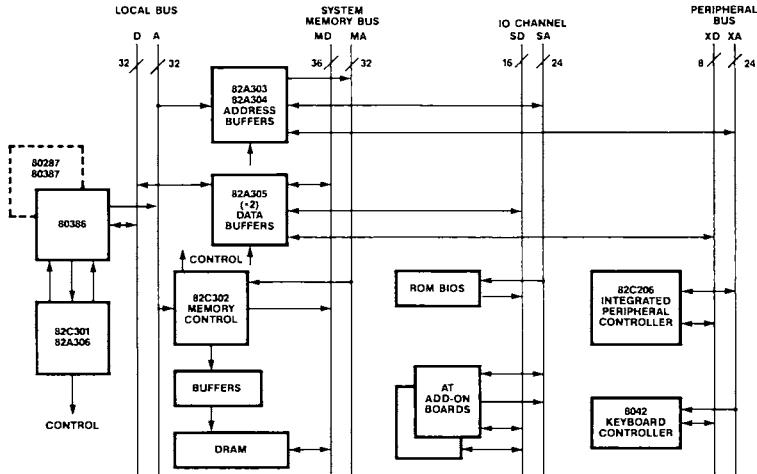
- 100% IBM™ PC AT compatible
- Flexible architecture allows usage in any iAPX 386™ design
- Operates in Page mode with Interleave memory subsystem
- 16 MHz zero wait operation
- Independent clock to support correct AT bus timing
- 1MB to 16MB of DRAM memory support
- A complete PC AT requires only 40 IC's plus memory

The CS8230 AT/386 CHIPSet™ is a seven chip VLSI implementation of most of the system logic to control an iAPX 386 based system. The CHIPSet is designed to offer a 100% PC AT compatible integrated solution. The flexible architecture of the CHIPSet allows it to be used in any iAPX386 based system design, such as CAD/CAE workstations, office systems, industrial and financial transaction systems.

CS8230 CHIPSet combined with CHIP's 82C206, Integrated Peripherals Controller, provides a complete PC AT compatible system using only 40 components plus memory devices.

The CS8230 CHIPSet™ consists of one 82C301 Bus Controller, one 82C302 Page/Interleave MemoryController, one each of 82A303 and 82A304 Address Bus Interfaces, two 82A305 Data Bus Interfaces, and a 82A306 Control Signal Buffer.

The CHIPSet supports a local CPU bus, a 32-bit system memory bus, and AT buses as shown in the system diagram below. The 82C301 and 82A306 provide the generation and synchronization of control signals for all buses. The 82C301 also supports an independent AT bus clock, and allows for dynamic selection of the processor clock between the 16 MHz clock and the AT bus clock. The



AT/386 System Block Diagram

CHIPS

82A306 provides buffers for bus control signals in addition to other miscellaneous logic functions.

The 82C302 Page/Interleave Memory Controller provides an interleaved memory subsystem design with page mode operation. It supports 1 MB to 16 MB of DRAMs with combinations of 256Kbit and 1Mbit DRAMs. The processor can operate at 16 MHz with zero wait state memory accesses by using 100 nsec DRAMs.

The 82A303 and 82A304 interface between all address buses, generate RAS/CAS addresses for the system memory and the addresses needed for proper data path conversion. Two 82A305 are used to interface between the local, system memory, and AT data buses. In addition to having high current drive, they also perform the conversion necessary between the different sized data paths.

CHIPS

System Overview

The CS 8230 is designed for use in 80386-based systems and provides complete support for the IBM PC AT bus. There are four buses supported by the CS 8230 as shown in the AT/386 system block diagram: CPU local bus (A and D), system memory bus (MA and MD), IO Channel bus (SA and SD), and X bus (XA and XD). The system memory bus is used to interface to DRAM's controlled by the 82C302. The IO channel bus refers to the bus supporting the AT bus adapters which could be either a 8 bit devices or 16 bit devices. The X bus refers to the peripheral bus to which the DMA controllers and timers are attached in an IBM PC AT. The X bus has only an 8-bit data path. The term "AT bus" is used to refer to the IO channel bus and X bus. Provisions are also made for user extension of the IO channel to 32 bit bus.

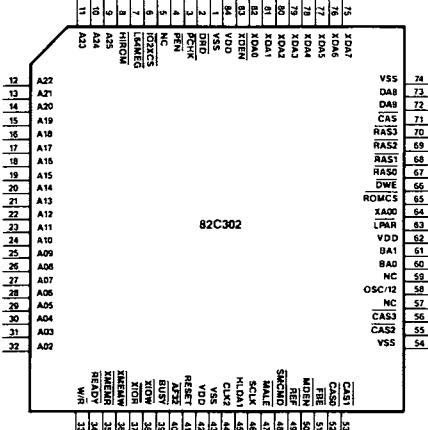
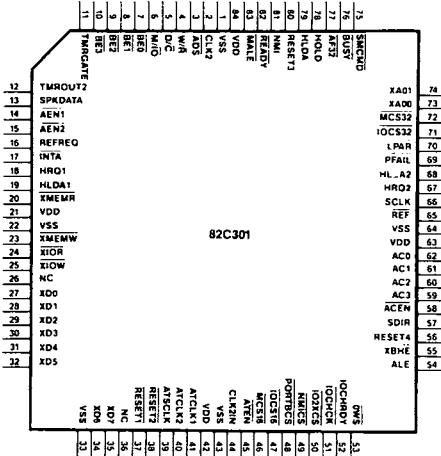
Notation and Glossary

The following notations are used to refer to the configuration and diagnostic registers internal to the 82C301 and 82C302.

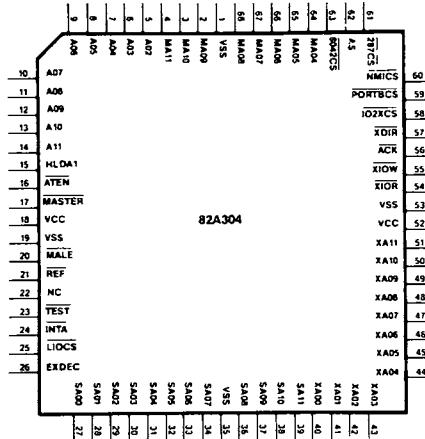
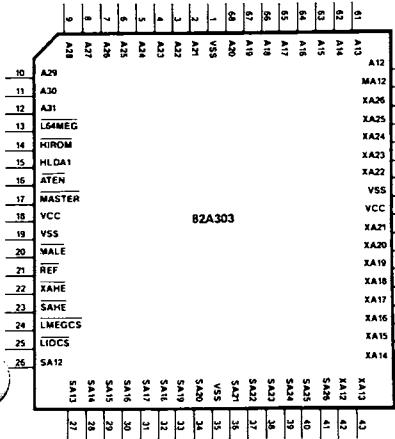
REGnH denotes the internal register with the index n in hexadecimal notation.

REGnH<x:y> denotes the bit field from bits y to x of the internal register with the index n in hexadecimal notation.

CHIPS



CHIPS



CHIPS

82C301 Pin Description

Pin No.	Symbol	Pin Type	Description
Clocks			
44	CLK2IN	I	CLOCK 2 INPUT from a packaged TTL crystal oscillator having twice the rated frequency of the processor.
2	CLK2	O	CLK2 output to the Clock 2 input of 80386 and the memory controller. This clock output is derived from CLK2IN and has a 50% duty cycle. The clock can also be programmed to be the same as the BCLK.
66	SCLK	O	SCLK is CLK2 divided by two and is an output generated as a reference to verify the phase relationship of the internal clock and CLK2.
41	ATCLK1	I	BUS CLOCK INPUT source from Crystal or Oscillator. This clock input is used for the AT Bus operation and is required only if the AT bus state machine clock (BCLK) will not be derived from the CLK2 input. This signal should be tied LO if not used.
40	ATCLK2	O	BUS CLOCK CRYSTAL OUTPUT is connected to the crystal oscillator circuit if a packaged oscillator is not used. A series resistor should be used to reduce the amplitude of the resonant circuit. It should be left unconnected if a packaged TTL oscillator is used.
39	ATSCLK	O	AT SYSTEM CLOCK is buffered to drive the clock signal SYSCLK on the AT bus I/O channel. It is half the frequency of BCLK and should have a nominal value in the range of 6 to 8 MHz for maintaining correct AT I/O bus timing with IBM PC AT.
Control			
37	RESET1	I	Active LO. RESET1 is connected to the power good signal generated by the PWRGOOD from the power supply. When LOW it will activate RESET3 and RESET4 for resetting the system.
38	RESET2	I	Active LO. RESET2 (8042RC) is an active LOW signal generated from the keyboard controller 8042 for a "warm reset" not requiring the system power to be shut off. It forces a CPU reset by activating RESET3 signal.
56	RESET4	O	Active HI. RESET4 is the System Reset used to reset the AT Bus, 82C206 IPC, 8042 keyboard controller 82C302 or 82C312 memory controller. RESET4 is synchronized with the processor clock.

CHIPS

82C301 Pin Description (Continued)

Pin No.	Symbol	Pin Type	Description
CPU Interface			
80	RESET3	O	Active HI. RESET3 is the reset to the 80386 when RESET1 or RESET2 is active. This is also activated when shutdown condition in the CPU is detected.
82	READY	I/O	Active LO. READY is driven LO during AT bus cycles indicating that the current CPU bus cycle is to be completed. It is also asserted if 'Time Out' condition is detected. During all other cycles it is an input to 82C301. Ready is an open collector output requiring an external pull up resistor. It connects to the 80386 READY pin.
3	ADS	I	Active LO. ADDRESS STATUS input connected to the 80386 ADS pin.
4	W/R	I/O	READ/WRITE STATUS input from the 80386 W/R signal. It indicates a write bus cycle if it is HI and a read cycle if it is LO.
5	D/C	I	DATA/CONTROL STATUS input from the 80386 D/C signal.
6	M/I _O	I	MEMORY/IO STATUS input from the 80386 M/I _O signal.
78	HOLD	O	Active HI. HOLD REQUEST output to the 80386 HOLD input pin. This signal is used to request the CPU to relinquish the bus cycles to another requesting master such as HRQ1, HRQ2 and REFREQ.
79	HLDA	I	Active HI. HOLD ACKNOWLEDGE input connected to processor HLDA signal. When the signal is HIGH it indicates that the processor has relinquished the system bus in response to the HOLD request.
10-7	BE<3:0>	B	Active LO. BYTE ENABLE signals input from the 80386 BE<3:0> during a CPU cycle. BE3 controls the most significant while BE0 controls the least significant byte. BE<3:0> are generated by 82C301 during DMA cycles based on the status signals XA0, XA1 and XBHE.
81	NMI	O	Active HI. NON-MASKABLE INTERRUPT connects to the 80386 NMI pin and is generated by 82C301 to cause an NMI.

CHIPS

82C301 Pin Description (Continued)

Pin No.	Symbol	Pin Type	Description
Decodes			
48	PORTBCS	I	Active LO. PORT B CHIP SELECT is the address decode input from the 82A304 as enable for the Port B register at address 061H.
49	NMICS	I	Active LO. NMI CHIP SELECT is the address decode input from the 82A304 as enable for the NMI enable bit at address 070H.
50	IO2XCS	I	Active LO. IO2X CHIP SELECT is the address decode input from the 82A304 as chip select for the IO registers at 022H and 023H used to access the 82C301 internal configuration registers.
53	OWS	I	Active LO. ZERO WAIT STATE acknowledge input from the IO channel. When active it causes immediate termination of the current AT bus memory or IO cycle.
IO Channel Interface			
52	IOCHRDY	I	Active HI. IO CHANNEL READY input from the AT bus. When LOW it indicates a 'not ready' condition and forces the insertion of wait states in I/O or memory accesses. When HIGH it will allow the completion of the current memory or I/O access.
51	IOCHCK	I	Active LO. IO CHANNEL CHECK input from the AT bus which causes an NMI to be generated if enabled. It is used to signal an Error condition from a device residing on the AT bus.
70	LPAR	I	Active LO. PARITY ERROR input from local memory system which causes an NMI to be generated if enabled.
69	PFAIL	I	Active LO. POWER FAIL WARNING signal input from the power supply.
54	ALE	O	Active HI. ADDRESS LATCH ENABLE to AT bus. This signal controls the address latches used to hold the address during a bus cycle. The signal should be buffered to drive the AT bus.
DMA Interface			
19	HLDA1	O	Active HI. HOLD ACKNOWLEDGE 1 is active when a bus cycle is granted in response to HRQ1.
68	HLDA2	O	Active HI. HOLD ACKNOWLEDGE 2 is active when a bus cycle is granted in response to HRQ2.

CHIPS

82C301 Pin Description (Continued)

Pin No.	Symbol	Pin Type	Description
18	HRQ1	I	Active HI. HOLD REQUEST 1 is active when a DMA/ Master is requesting a bus cycle. For an AT compatible architecture it should be connected to the HOLD REQUEST signal from DMA1 and DMA2.
67	HRQ2	I	Active HI. HOLD REQUEST 2 is active when a DMA/ Master is requesting a bus cycle. This should be grounded if not used.
14	AEN1	I	Active LO. ADDRESS ENABLE for 8 bit DMA transfers.
15	AEN2	I	Active LO. ADDRESS ENABLE for 16 bit DMA transfers.
Control Strobes			
46	MCS16	I	Active LO. MCS16 When active causes 16 bit memory accesses on IO channel.
72	MCS32	I	Active LO. MCS32 when active causes 32 bit memory accesses on IO channel.
47	IOCS16	I	Active LO. IOCS16 when active causes 16 bit IO accesses on IO channel.
71	IOCS32	I	Active LO. IOCS32 when active causes 32 bit IO accesses on IO channel.
75	SMCMD	O	Active LO. SYSTEM MEMORY COMMAND when active indicates the current access cycle is a memory cycle.
Refresh			
16	REFREQ	I	Active HI. REFresh REQuest when active initiates a DRAM refresh sequence be initiated. This signal is obtained from the timer controller, 8254, in a PC AT implementation.
69	REF	I/O	Active LO. REFresh is an open drain signal. It initiates a refresh cycle for the DRAMs. As an input it can be used to force a refresh cycle from an I/O device. An external pull up is required.
X Bus Interface			
20	XMEMR	I/O	Active LO. X BUS MEMORY READ is a control strobe directing memory to place data on the data bus. It is sourced either from the 82C301 when the 80386 is the master or from the DMA.

CHIPS

82C301 Pin Description (Continued)

Pin No.	Symbol	Pin Type	Description
23	XMEMW	I/O	Active LO. X BUS MEMORY WRITE is a control strobe directing memory to accept data from the data bus. It is sourced either from the 82C301 when the 80386 is the master or from the DMA.
24	XIOR	I/O	Active LO. X BUS IO READ is a control strobe directing an IO port to place data on the data bus. It is sourced either from the 82C301 when the 80386 is the master or from the DMA.
25	XIOW	I/O	Active LO. X BUS IO WRITE is a control strobe directing an IO port to accept data from the data bus. It is sourced either from the 82C305 when the 80386 is the master or from the DMA.
55	XBHE	I/O	Active LO. X BUS BYTE HIGH ENABLE indicates that the high byte (bits <15:08>) on the bus has valid data. It is sourced from the 82C301 when 80386 or DMA2 (16 bit) is the master.
35-34	XD<7:6>	I/O	X DATA BUS bits <7:6>
32-27	XD<5:0>	I/O	X DATA BUS bits <5:0>
57	SDIR	O	SYSTEM BUS DIRECTION controls the direction of data transfer between the IO channel and the local bus. When LO it enables data transfer from the IO channel to local bus.
58	ACEN	O	Active LO. ACTION CODE ENABLE when active validates the action code signals AC<3:0>.
59-62	AC<3:0>	O	ACTION CODE is a four-bits encoded command for bus size control and byte assembly operations performed by the 82A305s.

Memory Control

77	AF32	I	Active LO. AF32 when active indicates that a bus cycle is a local bus access without any data size conversion or AT cycle simulation treated as a 32-bit access.
76	BUSY	I	Active LO. BUSY from memory controller.
83	MALE	O	Active LO. Address Latch Enable for accesses to on board memory/IO. It also indicates start of a new CPU cycle.
11	TMRGATE	O	Active HI. TIMER GATE signal enables the timer on 8254 Timer to generate the tone signal for the speaker.
12	TMROUT2	I	Active HI. TIMER OUT 2 is the output from the timer 8254. It can be read from port B.

CHIPS

82C301 Pin Description (Continued)

Pin No.	Symbol	Pin Type	Description
13	SPKDATA	O	Active HI. SPEAKER DATA is used to gate the 8254 tone signal to the speaker.
17	INTA	O	Active LO. Interrupt acknowledge output to the interrupt controller.
45	ATEN	O	Active LO. AT ENABLE when active indicates the current CPU access is an AT bus cycle.
73	XA00	I/O	Address bit 0. It is sourced from the 82C301 when 80386 or DMA (16 bit) is a bus master.
74	XA01	I/O	Address but 1. It is sourced from the 82C301 when 80386 is a bus master.
26,36	NC		Reserved
21,42	VDD		
63,84	VDD		Power
1	VSS		
22,33	VSS		Ground
43,64	VSS		

CHIPS

82C302 Pin Description

Pin No.	Symbol	Pin Type	Description
Clocks and Control			
44	CLK2	I	Processor Clock
46	SCLK	O	Generated CLK2/2 for reference.
41	RESET	I	Active HI. When active resets 82C302
49	REF	I	Active LO. DRAM refresh control signal.
47	MALE	I	Active LO. Address Latch Enable
33	W/R	I	System WRITE/READ status input
48	SMCMD	I	Active LO. System Memory Command. Indicates that the current command is for memory.
37	XIOR	I	Active LO. I/O READ command used to qualify IO2XCS.
38	XIOW	I	Active LO. I/O WRITE command used to qualify IO2XCS.
35	XMEMR	I	Active LO. X Bus memory READ command.
36	XMEMW	I	Active LO. X Bus memory WRITE command.
45	HLDA1	I	Active HI. HOLD ACKNOWLEDGE 1 input from 82C301.
63	LPAR	I	Active LO. Parity error indication during a DRAM read. The failing address will be latched inside the chip for diagnostic purposes.
8	HIROM	I	Active LO. High ROM address chip select asserted when the highest 16 MBytes of memory is addressed ($A<31:A24>=FFH$). Unlatched. This is used in conjunction with the remaining address bits to generate the ROMCS signal.
7	L64MEG	I	Active LO. Low 64M address that is asserted when $A<31:26>=00H$. Unlatched.
65	ROMCS	O	Active LO. Chip select for the BIOS EPROMs that is qualified with W/R and SMCMD.
6	IO2XCS	I	Active LO. IO address 22 and 23 chip selects. I/O port 22 is the index register for the configuration register set and I/O 23 is accessed as the 8 bit configuration register selected by the index written to I/O port 22.
09-32	A<25:02>	I	Address from the CPU local bus.
64	XA00	I	Address from the X Bus

82C302 Pin Description (Continued)

Pin No.	Symbol	Pin Type	Description
34	READY	I/O	Active LO. System ready indicating the end of current 386 bus cycle. It goes inactive when the requested memory transfer has been completed. It becomes an input when the current bus cycle is for the AT IO expansion channel (AF32 = 0).
39	BUSY	O	Active LO. Indicates that the memory controller is still servicing a previous request. This should be connected to IOCHRDY through an open collector buffer. This signal should not be confused with the BUSY of 80386.
40	AF32	O	Active LO, open drain. If asserted indicates that the current address is for local memory on the system board (DRAM or possibly EPROM). Otherwise the current address is assumed to be on the AT IO channel.

Memory Expansion

61-60	BA<1:0>	O	Addresses that may be externally decoded to gate RAS to the correct block of 4 banks of DRAMs. These will always be zero.
2	DRD	O	Active LO. DRAM Read controls the direction of data transfer between the DRAM and local bus. When LO it controls the transfer from the memory data bus toward CPU and from the CPU to memory otherwise.

DRAM Interface

70-67	RAS<3:0>	O	Active LO. Row Address Strobe. There is one for each bank.
71	CAS	O	Active LO. Column Address Strobe. Used to latch data in the 82A305 data buffer.
56-55 53-52	CAS<3:2> CAS<1:0>	O	Active LO. Column Address Strobe. A strobe per bank that must be externally gated with byte enables for each byte of DRAM chips.
58	OSC/12	I	1.19MHz Clock input used for RAS pulse width timeout. Replaces CA3 on Cache controller.
66	DWE	O	Active LO. DRAM Write Enable.
51	FBE	O	Active LO. Force Byte Enable. Will always be inactive.
72-73	DA<9:8>	O	Remaining DRAM address bits.

CHIPS

82C302 Pin Description (Continued)

Pin No.	Symbol	Pin Type	Description
75-82	XDA<7:0>	I/O	Multiplexed bidirectional data pins for XD<7:0>. DA<7:0> are the lower address bits for the DRAM array.
83	XDEN	O	Active LO. XD bus buffer Enable. XDEN is asserted during IO access cycles to 022H and 023H if 022H access is for an internal register of 82C302. XDEN is used to control the chip enable for the buffer between the XD and XDA buses.
3	PCHK	O	Active LO. Parity Check Strobe.
4	PEN	O	Active LO. Overall Parity Enable.
50	MDEN	O	Active LO. MEMORY DATA BUFFER ENABLE. This signal is by default always LO and is connected to MDEN of 82A305.
Miscellaneous			
5,57,59	NC		Reserved
42,62,84	VDD		Power
1,43 54,74	VSS		Ground

CHIPS

82A303 Pin Description

Pin No.	Symbol	Pin Type	Description
Control			
17	MASTER	I	Active LO. Bus MASTER is generated by a device that is active on the expansion bus. After MASTER is force LO by an I/O device, the I/O CPU must wait for one system clock period before forcing the address and data lines. MASTER must not be held LO for more than 15 microseconds, or else data in the system memory may be lost due to lack of a refresh cycle.
15	HLDA1	I	Active HI. HOLD ACKNOWLEDGE from 82C301.
20	MALE	I	Active LO. MEMORY ADDRESS LATCH ENABLE clocks addresses into the address registers on the rising edge.
21	REF	I	Active LO. REFRESH. Schmitt Trigger.
16	ATEN	I	Active LO. AT BUS ENABLE is active when the CPU makes an AT bus access.
25	LIOCS	O	Active LO. LOW IO ADDRESS CHIP SELECT is asserted when A<15:12>=0.
24	LMEGCS	O	Active LO. LOW 1 MB SELECT is active when the access address decodes to the low 1MB address space: A<31:20> = 0.
13	L64MEG	O	Active LO. LOW 64 MB SELECT is active when the access address decodes to the low 64MB address space: A<31:26> = 0.
14	HIROM	O	Active LO. HI ROM SELECT is active when A<31:26> = 3FH.
Processor/Bus Interface			
12-2	A<31:21>	I/O	
68-60	A<20:12>	I/O	Local Address Bus
58-54	XA<26:22>	I/O	
51-42	XA<21:12>	I/O	X Address Bus
22	XAHE	I	Active LO. Enable bits 26:24 from the XA bus. A pullup is provided so that the input can be left open if only 24 bits are sourced externally.
41-36	SA<26:21>	I/O	
34-26	SA<20:12>	I/O	24mA. System Address Bus

82A303 Pin Description (Continued)

Pin No.	Symbol	Pin Type	Description
23	SAHE	I	Active LO. Enable bits 26:24 from the SA bus. A pullup is provided so that the input can be left open if only 24 bits are sourced externally.
59	MA12	O	Memory Address Bus Latched on the trailing edge of MALE.

Miscellaneous

18,52	VCC	Power
1,19 35,53	VSS	Ground

CHIPS

82A304 Pin Description

Pin No.	Symbol	Pin Type	Description
Control			
17	MASTER	I	Active LO. BUS MASTER is generated by a device active on the expansion bus.
15	HLDA1	I	Active HI. HOLD ACKNOWLEDGE 1 from 82C301.
20	MALE	I	Active LO. MEMORY ADDRESS LATCH ENABLE clocks addresses into the address registers on the rising (trailing) edge.
21	REF	I	Active LO. REFRESH Schmitt trigger.
16	ATEN	I	Active LO. AT BUS ENABLE is active when the CPU makes an AT bus access.
25	LIOCS	I	Active LO. LOW IO ADDRESS CHIP SELECT.
54	XIOR	I	Active LO. X BUS IO Read.
55	XIOW	I	Active LO. X BUS IO Write
57	XDIR	O	X BUS DIRECTION is used to control the drivers between the X bus and S bus. The drivers should be used such that S bus signals are driven toward X bus when XDIR is LO and in the other direction when HI.
26	EXDEC	I	Active HI. EXTENDED IO DECODE. A strapping option that when LO ignores A<11:10> and LIOCS (which is decoded based on A<15:12>) for decoding the system board IO ports. An internal pullup is provided.
58	IO2XCS	O	Active LO. IO 2x SELECT is decode of IO address 022H or 023H.
63	8042CS	O	Active LO. 8042 SELECT is decode of 8042 address at 060H or 064H.
59	PORTBCS	O	Active LO. PORTB SELECT is decode of Port B address at 061H
60	NMICS	O	Active LO. NMI SELECT is decode of NMI address at 070H.
61	287CS	O	Active LO. 80287 SELECT is decode of 287 address at 0E0-0FFH.
56	ACK	O	Active LO. ACKNOWLEDGE indicates that AEN1 OR AEN2 has been asserted. This signal is used to generate AEN signal on the AT I/O channel.
62	AS	O	Active HI. Address Strobe for the RTC. IO address 7xH is conditioned with XIOW.

CHIPS

82A304 Pin Description (Continued)

Pin No.	Symbol	Pin Type	Description
24	INTA	I	Active LO. INTERRUPT ACKNOWLEDGE bus cycle indication.
Processor/Bus Interface			
14-5	A<11:02>	I/O	Local address
51-40	XA<11:00>	I/O	X bus address
39-36	SA<11:08>	I/O	24mA. System address
34-27	SA<07:00>	I/O	
4-2	MA<11:09>	O	Memory address
68-64	MA<08:04>	O	
23	TEST	I	Active LO. TEST when active resets the refresh counter to zero. A pullup is provided.
22	NC		Reserved
Miscellaneous			
18,52	VCC		Power
1,19	VSS		Ground
35,53	VSS		

CHIPS

82A305 Pin Description

Pin No.	Symbol	Pin Type	Description
Control			
15-12	AC<3:0>	I	Action Code (bus size/assembly command)
16	ACEN	I	Active LO. Action Code Enable when active validates the action codes.
17	SDIR	I	System bus DIRection. When LO enables data transfers from the System to Local bus and in the other direction otherwise.
25	ATEN	I	Active LO. AT bus ENable
23	HLDA1	I	Active HI. HoLD Acknowledge.
20	MDEN	I	Active LO. MEMORY DATA BUFFER ENABLE. When LO enables the memory data buffers for transfer between the processor and memory subsystem. When HI disables these bus buffers. Should be connected to MDEN on the 82C302.
19	LDEN	I	Active LO. Selects LD as a source for the SD bus during MASTER or DMA reads. When HI selects MD. Asserting MRD overrides LDEN and gates MD to the SD bus. A pullup is provided.
21	MRD	I	Active LO. Memory Bus DIRection. When LO enables data movement for a processor read from the memory to local bus. MRD when HI enables drivers from local to memory bus.
22	DLE	I	Active LO. Data Latch Enable.
Data Paths (Bit numbers are for the upper 4 bits slice of each byte and should be 4 less for the lower nibble slice).			
11, 9	D<31:30>	I/O	
8, 6	D<29:28>	I/O	
5-2	D<23:20>	I/O	Local Data Bus
68-65	D<15:12>	I/O	
64-61	D<07:04>	I/O	
45,43	MD<31:30>	I/O	
42,40	MD<29:28>	I/O	
39-36	MD<23:20>	I/O	Memory Data Bus
34-31	MD<15:12>	I/O	
30-27	MD<07:04>	I/O	
49-46	PP<03:00>	O	Memory Partial Parity

82A305 Pin Description (Continued)

Pin No.	Symbol	Pin Type	Description
59,57	SD<15:14>	I/O	
56,55	SD<13:12>	I/O	
54-53	SD<07:06>	I/O	IO Channel Data Bus
51-50	SD<05:04>	I/O	
7,24	NC		Reserved.
41,58	NC		

Power Supplies

18,52	VCC	Power
1,10	VSS	
26,35	VSS	Ground
44,60	VSS	

CHIPS

82A306 Pin Description

Pin No.	Symbol	Pin Type	Description
64	ATSCLK	I	AT IO channel SYSCLK input.
59	SYSCLK	O	24mA. Buffered SYSCLK to AT IO channel. Nominally one half of the bus state machine clock frequency.
Control			
54	CX1	I	14.318MHz oscillator input from crystal.
55	CX2	O	14.318MHz oscillator output to crystal.
56	OSC	O	24mA. System 14.318MHz output.
57	OSC/12	O	24mA. 14.318MHz/12 = 1.19MHz output.
17	MALE	I	Active LO. Address Latch Enable for on board access.
12-15	BE<3:0>	I	Active LO. BYTE ENABLE.
4-7	LBE<3:0>	O	Active LO. LATCHED BYTE ENABLE on the trailing edge of MALE.
24	FBE	I	Active LO. FORCE BYTE ENABLE Forces all byte enables LBE active independent of MALE and the BE<3:0> inputs.
67	REF	I	Active LO. REFRESH.
8	A<31>	I	Local Address Bus bit 31.
10	M/IO	I	80386 Status used to generate AF32 for the 80387 and other 32 bit IO devices.
9	D/C	I	80386 Status used to generate AF32 for the 80387 and other 32 bit IO devices.
68	LMEGCS	I	Active LO. LOW MEGABYTE CHIP SELECT.
11	AF32	O	Active LO. Tri-state output. AF32 when active indicates a local bus memory access cycle on the system board. It is generated from M/IO, D/C, A<31>, and HLDA1.
66	MASTER	I	Active LO. Bus MASTER input from the AT IO channel.
3	RESET	I	Active HI. RESET input. Should be connected to RESET4 of 82C301.
62	RESETB	O	Active HI. Buffered RESET to X bus.
61	RDRV	O	Active HI. 24mA. RESET to AT bus.
2	ALE	I	Active HI. ALE for AT bus.
40	XBHE	I/O	Active LO. X Bus BHE.
44	XMEMR	I/O	Active LO. X Bus Memory Read.
43	XMEMW	I/O	Active LO. X Bus Memory Write.

82A306 Pin Description (Continued)

Pin No.	Symbol	Pin Type	Description
42	XIOR	I/O	Active LO. X Bus IO Read.
41	XIOW	I/O	Active LO. X Bus IO Write.
60	BALE	O	Active HI. 24mA. Buffered ALE to AT bus.
45	SBHE	I/O	Active LO. 24mA. System bus BHE.
51	SMEMR	O	Active LO. 24mA. System bus MEMory Read.
50	SMEMW	O	Active LO. 24mA. System bus MEMory Write.
49	MEMR	I/O	Active LO. 24mA. Memory Read.
48	MEMW	I/O	Active LO. 24mA. Memory Write.
47	IOR	I/O	Active LO. 24mA. IO Read.
46	IOW	I/O	Active LO. 24mA. IO Write.
16	HLDA1	I	Active HI. HOLD ACKNOWLEDGE from 82C301.
22	SCLK	I	CLK2/2 clock input. Should be connected to SCLK output of 82C302.
27-30	PPH<3:0>	I	PARTIAL PARITY HIGH computed by 82A305 for the high nibble data bits.
31-34	PPL<3:0>	I	PARTIAL PARITY LOW computed by 82A305 for the low nibble data bits.
36-39	MP<3:0>	I/O	Data Parity bits for the DRAMs.
21	CAS	I	Active LO. Read Parity latch enable.
20	PCHK	I	Active LO. PARITY CHECK STROBE for generating LPAR from the partial parity and data parity bits.
26	PEN	I	Active LO. Overall PARITY CHECK ENABLE.
23	WPE	I	Active LO. WRITE PARITY ENABLE. Enables the sourcing of write parity onto the MP bus. A pullup is provided.
25	LPAR	O	Active LO. LATCHED PARITY ERROR signal.
65	TEST	I	Active LO. Enables testing of the OSC/12 counter. A pullup is provided.
63	IN1	I	Input to an uncommitted 24mA non-inverting buffer.
58	OUT1	O	24mA. Output of the IN1 buffer.

Power Supply

18,52	VCC	Power
1,19	VSS	
35,53	VSS	Ground

82C301 BUS CONTROLLER

- Optional Independent AT Bus Clock
- Processor Clock Selection
- AT Bus Timing Configuration
- CPU Interface and Bus Control
- Port B Register

OVERVIEW

The 82C301 provides clock generation circuitry to solve two basic problems. One is to provide system designers the choice of a particular AT bus clock most adequate for their applications. The other is to allow the processor to run at the full speed and optionally at a speed to match timing dependent application software. Because many AT adapter boards are designed with built in timing assumptions, independent programmable controls are provided for AT bus command timing and wait state generation for IO accesses and for 8, 16, and 32 bit memory accesses.

The 82C301 interfaces directly with the 80386 and implements the state machines required for controlling all bus accesses. It also features a status register known as Port B register used in a standard IBM PC AT.

FUNCTIONAL DESCRIPTION

The 82C301 has the following function blocks as illustrated in figure 1-1:

- Clock generation and reset control
- CPU bus access state machine
- AT bus access state machine
- Port B register and NMI logics
- Bus Arbitration and refresh logic

Clock Generation and Reset Control

The 82C301 provides three major system clocks: the processor clock CLK2, the BCLK clock for AT bus state machine of 82C301, and the AT bus clock SYSCLK. The BCLK

(SYSCLK × 2) is a clock internal to the 82C301 and is used in this document to describe the system operation.

The clock generation circuitry shown in figure 1-2 contains two external clocks CLK2IN and ATCLK1 used as selectable clock sources. CLK2IN is assumed to be greater than 20MHz and should come from a packaged crystal oscillator while an oscillator circuit is provided for the ATCLK1 signal so that it can be connected to either a packaged oscillator or a crystal. This ATCLK1 input is required only if BCLK need be derived from an independent clock source other than CLK2IN. When required ATCLK1 frequency should be between 10 and 20 MHz for a typical PC AT. Notice that by design SYSCLK is always the BCLK divided by two.

The clock switching logic is guaranteed to provide a "clean" transition with no phases shorter than the minimum values or longer than the maximum values. This allows the clock selections be done dynamically.

Processor clock selection

Referring to figure 1-2, the CLK2 and SCLK signal can be selected from two sources:

- the external oscillator connected to CLK2IN,
- the AT bus state machine clock BCLK.

This selection is made by programming bit 4 of the configuration register 4 which defaults to CLK2IN upon reset. By design, if SCLK is selected as the source for BCLK, CLK2 must not be sourced from BCLK. In all but some special cases, CLK2IN should match the rated processor speed, and the BCLK can be either subdivisions of CLK2IN or ATCLK1.

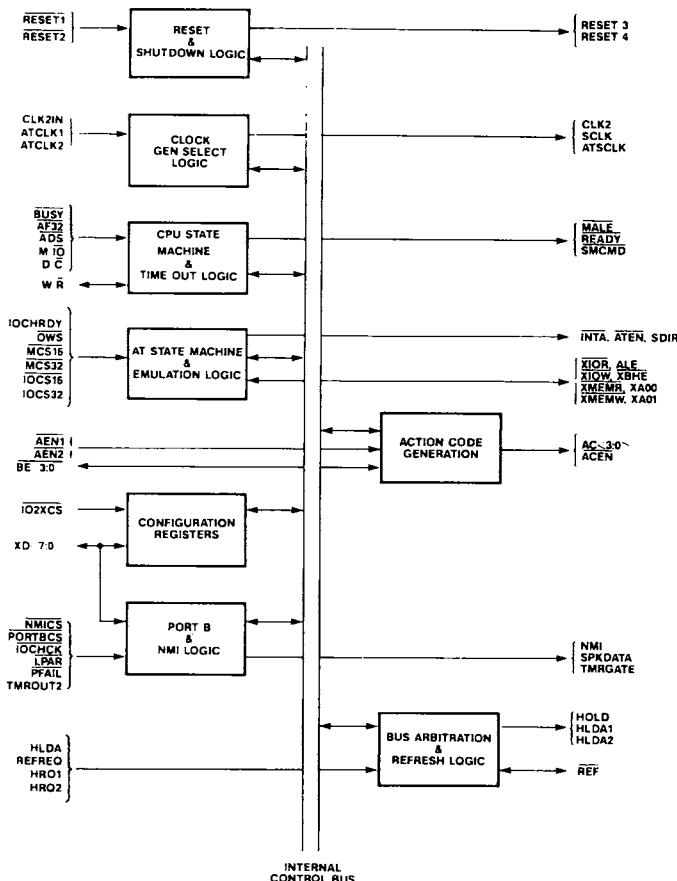


Figure 1-1. 82C301 Functional Block Diagram

CHIPS

AT bus clock selection

The 82C301 provides flexible software controlled selection of the clock used for the AT bus state machine. The clock can be synchronous (related but not necessarily equal) to the processor CLK2 or unrelated (requiring synchronization between the AT and processor bus state machines). While synchronization logic has been provided in all interface signals between the CPU and the AT state machines, it is highly recommended that the AT bus state machine clock (BCLK) be sourced from SCLK. An internal programmable divider has been provided allowing BCLK frequency of CLK2/2 or CLK2/3. This eliminates the need for an additional oscillator for some system designs. If the divide by 3 option is selected the resulting waveform will have an approximately 50% duty cycle.

The SYSCLK signal generated by the 82C301 is one half of the AT bus state machine clock BCLK. Since this clock is used to drive the AT bus, it is recommended that the the divide

ratio be set for a SYSCLK of about 6 and 8MHz. The table 1-1 shows the combination of clock frequencies obtainable from CLK2IN with this selection scheme.

CLK2IN	SCLK	Ratio	BCLK	SYSCLK
24	12	/2	12	6
32	16	/2	16	8
32	16/3	10.7	5.4	

Table 1-1. Examples of BCLK and SYSCLK derived from CLK2

Reset control

When RESET1 signal is asserted 82C301 asserts RESET3 and RESET4 for a system reset. For warm restart not requiring the extensive reset, RESET2 can be asserted to generate the RESET3 for resetting only the processor and some specific devices. RESET3 is also asserted when CPU shut down condition is detected. This differentiation of reset is

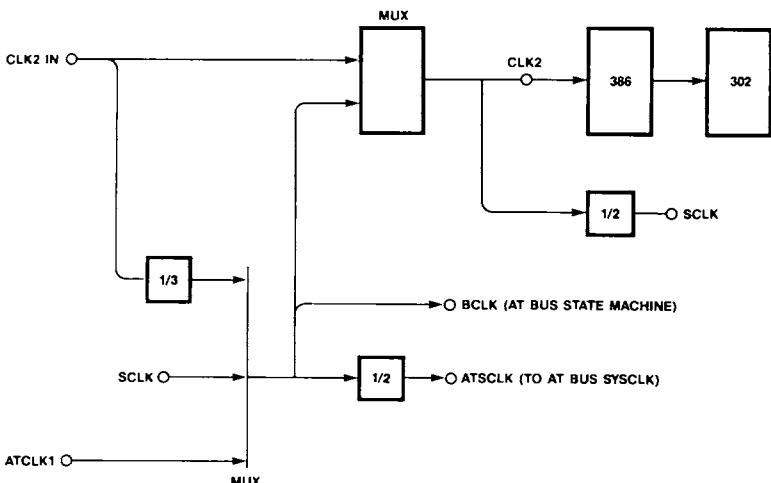


Figure 1-2. CLK2 and ATSCLK Clock Selection

provided so that some register states can be maintained through the reset if so desired.

Bus Arbitration, CPU bus and AT bus State Machines

The 82C301 performs the synchronization and control required between the local processor bus, the memory subsystem and the AT IO channel. It controls all bus activities and handles the HRQ1, HRQ2, and REFREQ by generating HOLD request to the CPU and arbitrating among these requests in a non-preemptive manner. Upon CPU asserting HLDA the arbitration logic in turn responds by asserting HLDA1 (for HRQ1) or HLDA2 (for HRQ2), and the requesting DMA or master device has the control of the bus until it de-asserts HRQ1 (or HRQ2) to terminate the HLDA cycle. During the HLDA cycle, the 82C301 generates both SMCMD and action codes AC<3:0> to control the buffer enable and directions for the address and data buffers. Bus size conversions are not supported by 82C301 for these bus cycles and if necessary should be performed by the requesting device.

The CPU state machine and AT state machine control CPU accesses to the devices on the local bus and non-local buses respectively. The CPU state machine supports only 32 bit transfers between the 80386 and system memory (or memory mapped IO) and no bus size conversions are done. Thus BS16 input on the 80386 is not used in a CS8230 system and should be connected to a HI level. The AT state machine responsible for all non-local bus CPU accesses controls the AT bus and supports bus size matching.

All CPU access cycles are started by 82C301 asserting MALE. The CPU state machine then samples AF32 one SCLK clock cycle later. If AF32 is active, it is assumed to be a local bus cycle and the CPU state machine terminates this cycle when it detects READY signal active. In response to an MALE, if the AF32 is detected inactive the control is passed to AT state machine. At the end of the bus access cycle, the AT state machine generates READY to terminate the processor access cycle as well as the CPU state machine cycle.

CPU State Machine

Interface to the 80386 requires interpretation of the status lines upon assertion of ADS and synchronization and generation of a READY response to the CPU upon completion of the requested operation. By interpreting the CPU status lines and ADS, the 82C301 generates control signals MALE and SMCMD. In response to each ADS generated by CPU, an MALE is generated by the 82C301 to indicate the start of a new CPU access cycle. In a non-pipelined CPU cycle, MALE is generated in response to ADS being asserted by the 80386. In a pipelined cycle, MALE is generated when the assertion of READY is detected for the previous CPU cycle. If AF32 is not active one cycle after MALE is asserted, control is passed to the AT bus state machine. The CPU state machine then waits for READY becoming active to terminate the access cycle. In CS 8230 CHIPset, the READY can be generated by 82C302 which controls the system memory access.

SMCMD indicates a memory cycle for both CPU and non-CPU accesses. During CPU cycles it is generated for all memory cycles by decoding M/I/O, D/C and W/R signals. During non-CPU cycles it is active when XMEMR or XMEMW is active.

NA Pipeline Control

The 82C301 supports both pipelined and non-pipelined cycles of the 80386. The NA (Next Address) input on the 80386 can be always asserted in a CS8230 system for higher performance.

Bus Timeout

An optional feature allows generation of an NMI if an internal memory cycle does not complete within a certain timeout period. This occurs if AF32 is asserted in response to MALE and READY is not returned to the 82C301 within 128 CLK2 cycles. A control bit in the 82C301 configuration registers enables this feature.

AT Bus State Machine

The AT state machine gains the control of the buses when AF32 is detected inactive by the

CHIPS

CPU state machine. It uses BCLK having a frequency twice that of the IO channel clock SYSCLK. When ATCLK1 is selected as the source for BCLK, it also performs the necessary synchronization of control and status signals between the AT bus and the processor. The 82C301 supports 8, 16 or 32 bit transfers between the processor and 8, 16 or 32 bit memory or IO devices located on the IO channel.

An AT bus cycle is initiated by asserting ALE decoded from the CPU status signals and is terminated by asserting READY. On the falling (or trailing) edge of the ALE, MCS16, IOCS16, MCS32, IOCS32 are sampled to determine the bus size conversion required. It then enters the command cycle. The AT bus state machine provides the sequencing and timing controls for status and command phases of different AT bus cycles. These controls provide for timing emulation of lower speed IO channels to maintain compatibility with AT or PC/XT IO adapters and memory cards. The command cycle is terminated by detecting OWS or IOCHRDY active.

IO Channel Speed Control

The AT state machine can be programmed to insert wait states in units of ATSCLK and to delay the generation of XIOR, XIOW, XMEMR, and XMEMW commands in one half units of ATSCLK (BCLK) within the selected wait states. The command phase delay can be selectively defined for IO cycles and for 8, 16, and 32 bit wide memory cycles by setting the corresponding fields in REG05H. REG06H controls the IO Channel wait state generation for 8, 16, and 32 bit accesses.

The bus clock BCLK is selected by setting REG06H<1:0>. It should be noted that the processor clock source should be set to CLK2IN whenever the BCLK is selected to be SCLK.

Data Conversion

The AT bus access state machine performs data conversion for CPU accesses to devices not on the local bus when AF32 is not asserted.

AT bus data conversions are performed for the following types of transfers:

- 32 bit to 8/16 bit,
- 24 bit to 8/16 bit,
- 16 bit to 8/16 bit.

Larger transfers are broken into smaller AT bus reads or writes and the action code AC<3:0> to the 82A306 is generated. Byte addresses XA<01:00> are generated to drive the lower two bits of the AT address bus.

The 82C301 responds to IOCS16, MCS16, IOCS32, and MCS32 in determining what size of data the IO channel needs. If none of the above signals are asserted, 8 bit transfers are assumed and the request is converted into 2, 3 or 4 IO channel cycles based on BE<3:0>. For either MCS16 or IOCS16, the AT bus state machine converts a 32-bit access into two 16 bit AT bus accesses.

The bus state machine also supports 32-bits transfer between the processor and memory and IO devices on the IO channel. IOCS32 and MCS32 inputs allow a device to request a 32-bits transfer. It is assumed that the necessary extensions to the AT bus are made to utilize this feature. IOCS32 and MCS32 override IOCS16 and MCS16.

In performing these data conversions, a 4-bits action code AC<3:0> is generated to control the buffers in 82A305 for the alignment of data path, and direction control between D, MD, and SD data buses. The definition for the action codes is given in the functional description of 82A305.

Port B Register

The 82C301 provides access to Port B defined for a PC AT as shown in figure 1-3. PORTBCS enables the access to Port B register and is provided as an output from 82A304. Table 1-2 gives the Port B register bit definition.

IO ADDR	7	6	5	4	3	2	1	0
62H	PCK	CHK	T2O	RFD	EIC	ERP	SPK	T2G

Figure 1-3. Port B register definition

Addr	Bits	Function
62H		Port B Register
7	Read only.	PCK - System memory parity check.
6	Read only.	CHK - IO channel check.
5	Read only.	T2O - Timer 2 out
4	Read only.	RFD - Refresh Detect.
3	Read/write.	EIC - Enable IO 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

Table 1-2. Port B Register Definition

CS 8230 Internal Register Access Ports

The CS 8230 have internal registers used for system configurations and for diagnostics. These are accessed through IO ports 22H and 23H normally found in the interrupt controller. An indexing scheme is used to reduce the number of IO addresses required to access all registers needed to configure and control CS 8230 chips. Each access (either read or write) to an internal register is done by first writing its index into port 22H. This index then controls the multiplexers gating the appropriate register data accessible as port 23H. Every access to port 23H must be preceded by writing the index value to port 22H even if the same data port is being accessed again.

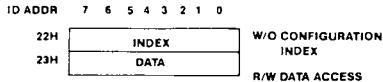


Figure 1-4. Configuration Register Access Ports

Configuration Registers

There are 3 bytes of configuration and diagnostic registers in 82C301 as shown in figure 1-5. The definitions for these registers are given in table 1-3.

INDEX	7	6	5	4	3	2	1	0
04H	VERS	-	PC	FE	TE	PF	TO	VERSION/PROCESSOR CLOCK/NMI SOURCES
05H	M32	M 16	MB		IO			COMMAND DELAY
06H	32 WS	16 WS	8 WS	B CLK				WAIT STATE/BUS CLOCK

Figure 1-5. 82C301 Internal Configuration Registers

CHIPS

Index	Bits	Function
04H		Version/Processor clock select/NMI source
	7:6	Read only. Version
	0	Initial version
	5	Reserved
	4	Processor Clock Select. If SCLK is selected as the source for BCLK, CLK2 source must not be selected as BCLK.
	0	Use processor oscillator input. Default.
	1	Use AT bus state machine clock (SYSCLKx2).
	3	Power Fail Warning Enable
	0	Power Fail NMI not enabled. Default.
	1	Power Fail NMI enabled
	2	Local Bus READY timeout NMI Enable
	0	READY timeout NMI not enabled. Default.
	1	READY timeout NMI enabled
	1	Read only. Power Fail warning active during last NMI arbitration.
	0	Power Fail warning pin not active. Default.
	1	Power Fail warning pin was active.
	0	Read only. Local bus READY timeout
	0	READY timeout has not occurred. Default.
	1	READY timeout has occurred
05H		Command delay
		The value for each one of the command delay field is defined as:
	0	0 cycle delay
	1	1 cycle delay
	2	2 cycle delay
	3	3 cycle delay
	7:6	AT Bus 32 bit memory command delay Specifies between 0 and 3 BCLK cycles for command delay during an AT bus 32 bit memory cycle. Default is 0.
	5:4	AT Bus 16 bit memory command delay Specifies between 0 and 3 BCLK cycles for command delay during an AT bus 16 bit memory cycle. Default is 0.

Table 1-3. 82C301 Configuration Register Definitions

Index	Bits	Function
05H	3:2	AT Bus 8 bit memory command delay Specifies between 0 and 3 BCLK cycles for command delay during an AT bus 8 bit memory cycle. Default is 1.
	1:0	AT Bus I/O Cycle command delay Specifies between 0 and 3 BCLK cycles for command delay during an AT bus IO cycle. Default is 1.
06H		Wait State/Bus Clock Source
	7:6	32 bit AT Bus wait state select 0-3 wait states per 32 bit transfer on the AT bus. Each wait state is 2 BCLK cycles. Default is 3. 0 3 cycle delay 1 2 cycle delay 2 1 cycle delay 3 0 cycle delay
	5:4	16 bit AT Bus wait state select 0-3 wait states per 16 bit transfer on the AT bus. Each wait state is 2 BCLK cycles. Default is 3. 0 3 cycle delay 1 2 cycle delay 2 1 cycle delay 3 0 cycle delay
	3:2	8 bit AT Bus wait state select 2-5 wait states per 8 bit transfer on the AT bus. Each wait state is 2 BCLK cycles. Default is 5. 0 5 cycle delay 1 4 cycle delay 2 3 cycle delay 3 2 cycle delay
	1:0	Bus Clock Source Select 0 Use Proc Clock/3 for AT bus state machine. Default. 1 Use Proc Clock/2 for AT bus state machine. 2 Reserved. 3 Use ATCLK input pin for the AT bus state machine.

Table 1-3. 82C301 Configuration Register Definitions (Continued)

CHIPS

82C301 Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V _{CC}	—	7.0	V
Input Voltage	V _I	-0.5	5.5	V
Output Voltage	V _O	-0.5	5.5	V
Operating Temperature	T _{op}	-25	85	C
Storage Temperature	T _{stg}	-40	125	C

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

82C301 Operating Conditions

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V _{CC}	4.75	5.25	V
Ambient Temperature	T _A	0	70	C

82C301 DC Characteristics

Parameter	Symbol	Min.	Max.	Units
Input Low Voltage	V _{IL}	—	0.8	V
Input High Voltage	V _{IH}	—	2.0	V
Output Low Voltage I _{OL} =8mA (Note 1)	V _{OL}	—	0.45	V
Output High Voltage I _{OH} =-200 μA	V _{OH}	—	2.4	V
Input Current 0 < V _{IN} < V _{CC}	I _{IL}	—	±10	μA
Output Short Circuit Current V _O =0V	I _{OS}	TBD	TBD	mA
Input Clamp Voltage	V _{IC}	—	TBD	V
Power Supply Current @ 16 MHz Clock	I _{CC}	—	40	mA
Output Hi-Z Leak Current 0.45 < V _{OUT} < V _{CC}	I _{OZ1}	—	±10	μA
CLK2 Output Low Voltage @ I _{OL} = 5 mA	V _{OLC}	—	0.45	V
CLK2 Output High Voltage @ I _{OH} = -1 mA	V _{OHC}	—	4.0	V

NOTE:

- REF has I_{OL} = 16mA. CLK2, MALE have I_{OL} = 8mA. All other outputs and I/O pins have I_{OL} = 4mA. In all cases all I_{OL} = I_{OH} for the pin.

82C301 AC Characteristics
 $(T_A = 0^\circ\text{C} \text{ to } 70^\circ\text{C}, V_{CC} = 5V \pm 5\%)$

Sym	Description	Min.	Typ.	Max.	Units
t101	CLK2 period		31		ns
t102	CLK2 low time (at 32 MHz)	9			ns
t103	CLK2 high time (at 32 MHz)	9			ns
t104	CLK2 rise time		8		ns
t105	CLK2 fall time		8		ns
t106	SCLK delay from CLK2!		4		ns
t107	RESET3, RESET4 set-up time		15		ns
t108	RESET3, RESET4 hold time		8		ns
t109	SMCMD delay from MALE active		7		ns
t110	AF32 set-up time to CLK2!	22			ns
t111	AF32 hold time to CLK2!	0			ns
t112	HOLD delay from CLK2!		25		ns
t113	READY input set-up time to CLK2!	13			ns
t114	READY input hold time from CLK2!	5			ns
t115	ATEN active delay from CLK2!		20		ns
t116	ATEN inactive delay from CLK2!		20		ns
t117	MALE active delay from CLK2!		15		ns
t118	MALE inactive delay from CLK2!		15		ns
t119	READY output active delay from CLK2!		20		ns
t120	READY output inactive delay from CLK2!		20		ns
t121	ATSCLK period		125		ns
t122	ATSCLK low time		62		ns
t123	ATSCLK high time		62		ns
t124	ATSCLK rise time		8		ns
t125	ATSCLK fall time		8		ns
t126	ALE delay from ATSCLK (! or !)		5		ns
t127	XIOR, XMEMR, INTA active delay from ATSCLK (! or !)		10		ns
t128	XIOR, XMEMR, INTA inactive delay from ATSCLK!		10		ns
t129	IOCHRDY set-up time to ATSCLK!		17		ns

CHIPS

82C301 AC Characteristics (Continued)

($T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5\text{V} \pm 5\%$)

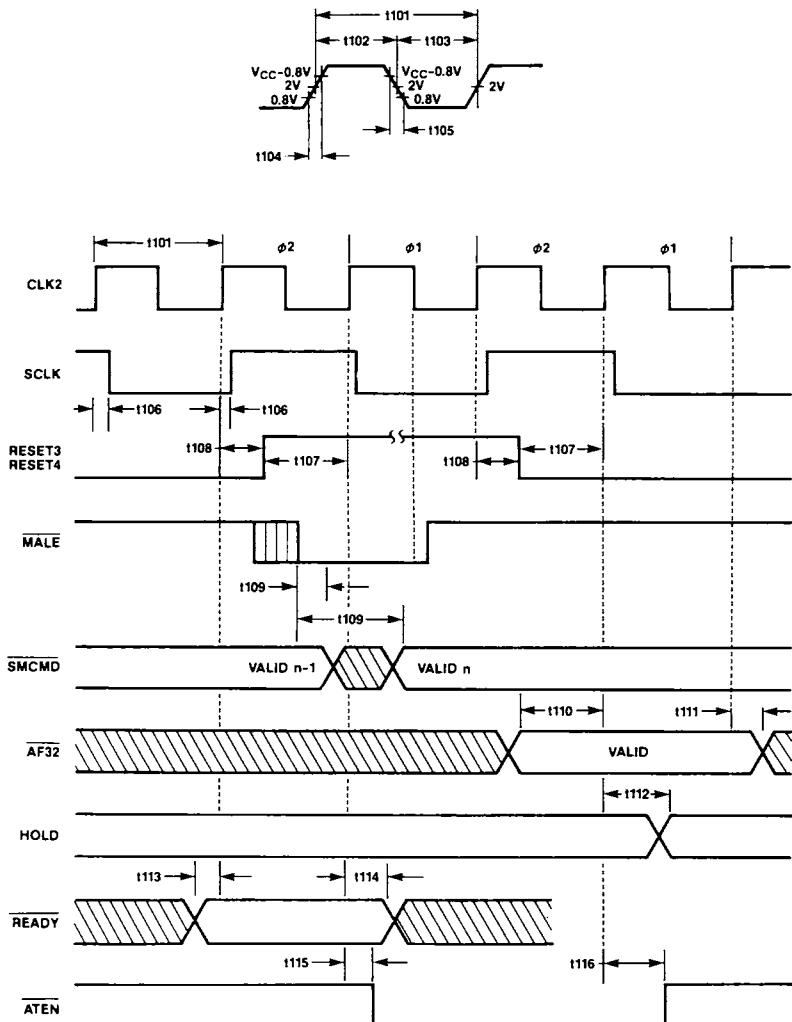
Sym	Description	Min.	Typ.	Max.	Units
t130	IOCHRDY hold time to ATSCLKI	0			ns
t131	MCS16, IOCS16 set-up time to ATSCLKI	35			ns
t132	MCS16, IOCS16 hold time from ATSCLKI	25			ns
t133	MCS32, IOCS32 set-up time to ATSCLKI	35			ns
t134	MCS32, IOCS32 hold time from ATSCLKI	25			ns
t135	XA00, XA01, SBHE active delay from ATSCLKI	20			ns
t136	XA00, XA01, SBHE inactive delay from ATSCLKI	10			ns
t137	ACEN active delay (read cycle) from ATSCLKI	20			ns
t138	ACEN inactive delay (read cycle) from ATSCLKI	10			ns
t139	AC<3:0> active delay from ATSCLKI	20			ns
t140	AC<3:0> inactive delay from ATSCLKI	10			ns
t145	XMEMR, XMEMW active delay from ATSCLKI (with zero command delay)	15			ns
t146	ACEN active delay (write cycle) from ATSCLKI	10			ns
t147	ACEN inactive delay (write cycle) from ATSCLKI	0			ns
t148	OWS set-up time to ATSCLKI	17			ns
t149	OWS hold time from ATSCLKI	0			ns
t151	NMICS set-up time to XIOW active	20			ns
t152	NMICS hold time from XIOW inactive	20			ns
t153	Data (XD7) set-up time to XIOW inactive	30			ns
t154	Data (XD7) hold time from XIOW inactive	20			ns
t155	NMI delay from XIOW inactive	25			ns
t156	PORTBCS set-up time to XIOR, XIOW active	20			ns
t157	PORTBCS hold time from XIOR, XIOW inactive	20			ns
t158	Data (XD<7:0>) valid delay from XIOR active	15			ns
t159	Data (XD<7:0>) hold time from XIOR inactive	15			ns
t160	IO2XCS set-up time to XIOR, XIOW active	10			ns
t161	IO2XCS hold time from XIOR, XIOW inactive	15			ns
t162	LPAR, IOCHK, PFAIL pulse width	15			ns
t165	REFREQ pulse width	15			ns

82C301 AC Characteristics (Continued)(T_A = 0°C to 70°C, V_{CC} = 5V ± 5%)

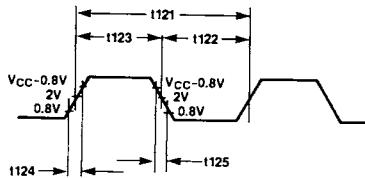
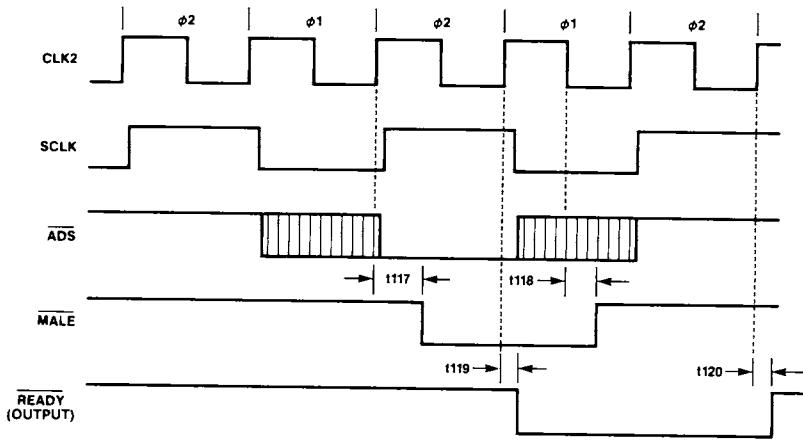
Sym	Description	Min.	Typ.	Max.	Units
t166	REF set-up time to ATSCLK1	10		ns	
t167	XMEMR active delay (refresh cycle) from ATSCLK1	15		ns	
t168	XMEMR inactive delay (refresh cycle) from ATSCLK1	15		ns	
t169	IOCHRDY set-up time (refresh cycle) to ATSCLK1	25		ns	
t170	IOCHRDY hold time (refresh cycle) from ATSCLK1	0		ns	
t171	BE<3:0> active delay from XA0, XA1, XBHE valid	15		ns	
t172	BE<3:0> inactive delay	15		ns	
t173	SMCMD active delay from XMEMR, XMEMW active	20		ns	
t174	SMCMD inactive delay from XMEMR, XMEMW inactive	20		ns	
t175	ACEN active delay from HLDA1 active	20		ns	
t176	ACEN inactive delay from HLDA1 inactive	20		ns	
t177	AC<3:0> active delay from XA0, XA1, XBHE not valid		TBD		
t178	AC<3:0> inactive delay from XA0, XA1, XBHE valid		TBD		
t179	AC<3:0> active delay from XMEMR active		TBD		
t180	AC<3:0> inactive delay from XMEMR inactive		TBD		

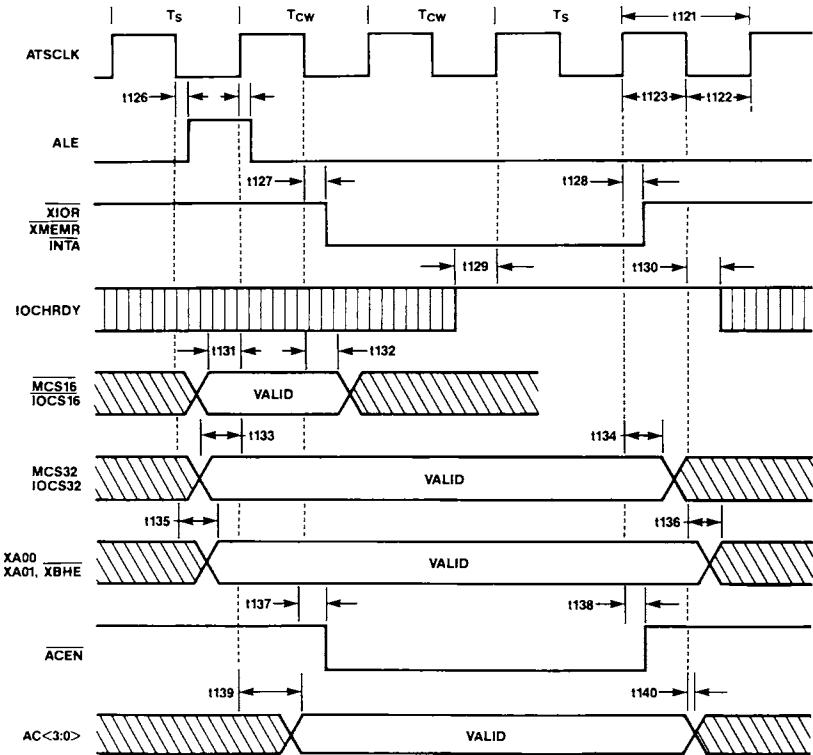
CHIPS

82C301 TIMING DIAGRAMS

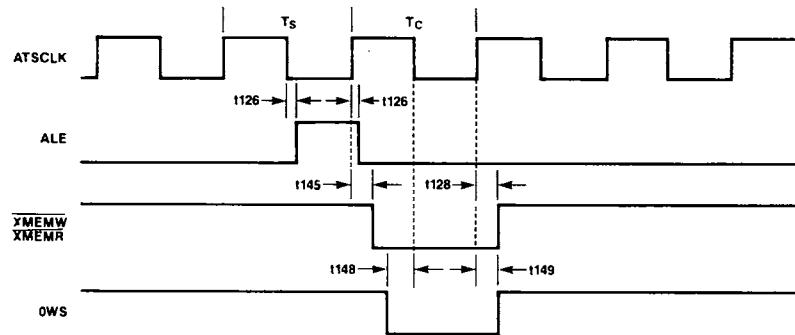
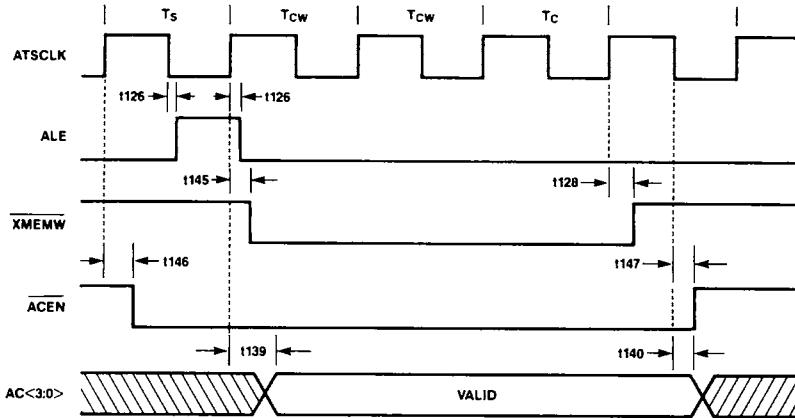


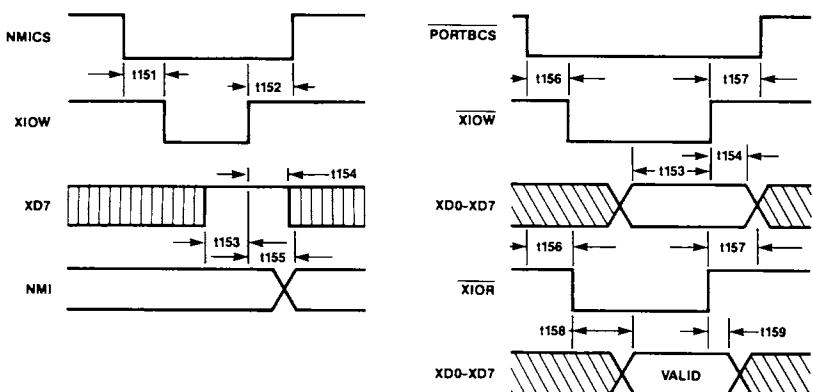
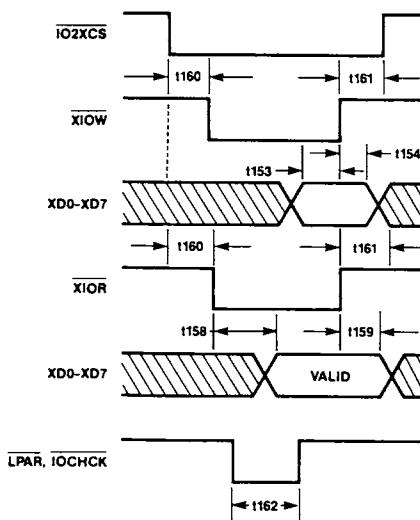
82C301 TIMING DIAGRAMS

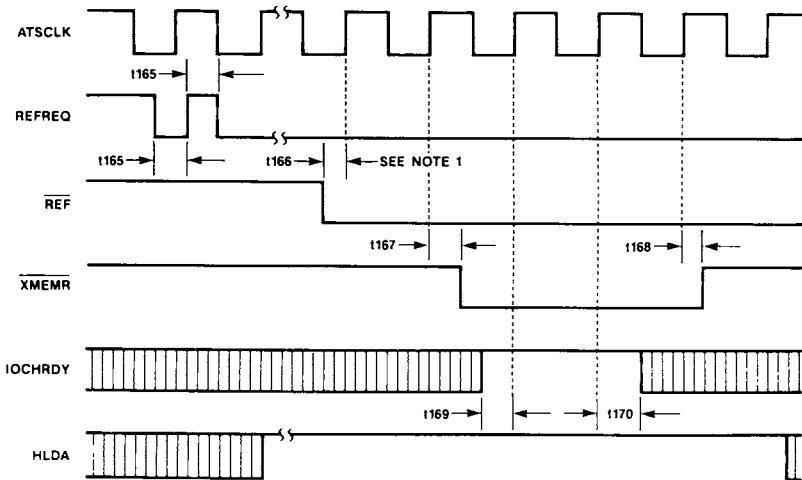


82C301 TIMING DIAGRAMS


82C301 TIMING DIAGRAMS



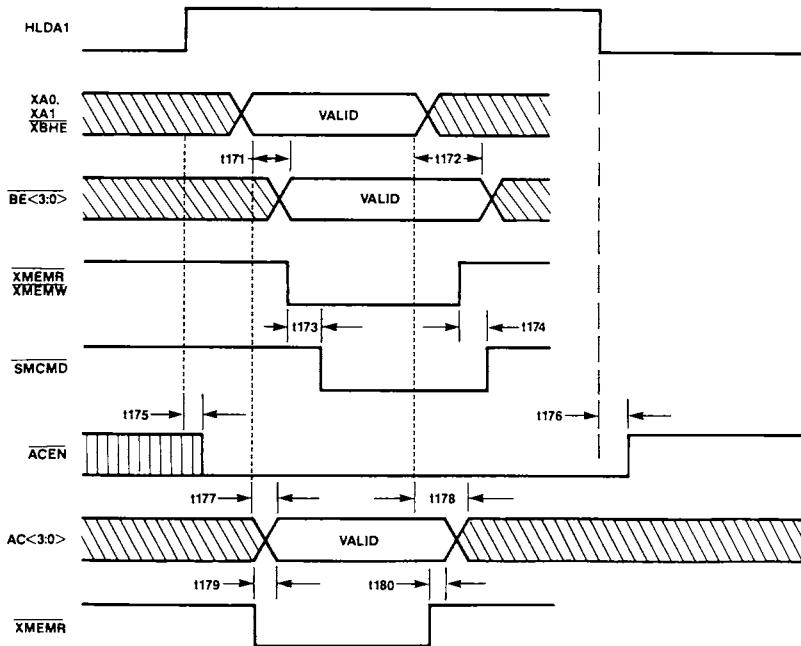
82C301 TIMING DIAGRAMS


82C301 TIMING DIAGRAMS**NOTE**

REF is an asynchronous signal and the setup time is specified only to guarantee starting a refresh cycle on that clock cycle instead of next one.

CHIPS

82C301 TIMING DIAGRAMS





CHIPS

82C302 PAGE/INTERLEAVE MEMORY CONTROLLER

- Page mode access with interleaved memory banks achieves higher performance than conventional DRAM arrays.
- Zero wait state access at 16MHz using 100nS DRAMs.
- Minimum configuration of 1 bank of 36 bits, 1MB using 256K x 1 or 4MB using 1M x 1 DRAMs.
- Maximum configuration of 4 banks of 36 bits, 4MB using 256K x 1 or 16MB using 1M x 1 DRAMs.
- Memory configurations of 1, 2 and 4 banks.
- Staggered refresh to reduce power supply noise.

OVERVIEW

The 82C302 performs the memory control functions in a 80386-based systems that utilizes page mode access DRAMs. The memory configurations can be one bank (non-interleaved) or multiple banks (2 or 4) interleaved on 2KB-page basis.

Array Configuration

The 82C302 organizes memory as banks of 36 bits consisting of 32 bits of data and 4 bits of parity. A common design may use either 36 by-1 DRAMs or 8 by-4 and 4 by-1 DRAMs. The minimum configuration can be a single bank operating in non-interleaved mode or can be one to two pairs of banks operating in two-way page interleaved mode at higher performance.

The memory controller is designed such that the memory can be up-graded from one to two banks by making it a two-way interleaved organization. Because of the interleaved page operation, the third and fourth banks must be added as a pair. Furthermore, the DRAM types must be identical in each bank of a pair due to the interleaved configuration. However, each pair of banks can use different DRAM

types, with one or two banks of smaller DRAM types and later upgraded with additional pairs of banks of larger DRAMs.

Page Interleaved Operation

The 82C302 uses a page interleaved design that is different from most interleaved memory designs. Normal two-way interleaving uses two banks of DRAMs with even (double word) addresses stored in one bank and odd addresses in the other. If accesses are sequential (or at least to alternating even and odd addresses) the RAS precharge time of one set can be overlapped with the access time of the second set. Typically the hit rate (fraction of times that the required bank is available) is 50%. This is especially true since operand accesses (which tend to be more random) can be interspersed with (most likely sequential) instruction fetches.

Page mode operation available with most DRAMs operates because the access to the row address of the internal DRAM array makes available a large number of bits (512 bits in a 256K x 1) that are subsequently selected using the column address. Once a row access has been made higher speed random access can be made to any bit (1 of 512) within the row. The page mode access and cycle times are typically half that of the normal access and cycle times respectively. If 36 256K x 1 DRAMs are used to implement a bank, a page would have 512×4 bytes = 2KB. Thus memory could be interleaved on a 2KB page rather than 4B basis. Any access to the currently active RAS page would occur in the page access rather than the normal access time and any subsequent access could be to anywhere in the same 2KB without incurring any penalty due to RAS precharge.

When memory is configured to take advantage of this DRAM organization, significantly better performance can be achieved over normal interleaving. There are two reasons for this:

- The page mode access is faster than the normal access time. This permits more relaxed timing in order to achieve the same 0 wait-state "hit" access.

- The frequency of the next access being fast (same or alternate page vs. alternate address in interleaved mode) is significantly higher. This is because of the principle of locality of reference, instructions and data tend to be clustered together.

However, the complexity is somewhat higher in the page mode controller, making VLSI an ideal implementation vehicle.

FUNCTIONAL DESCRIPTION

The 82C302 performs four major functions as shown in figure 2-1:

- DRAM memory access arbitration
- DRAM memory access cycle control
- DRAM refresh
- Memory mapping

Memory Access State Machines and Arbitration

The 82C302 controls the DRAM memory access from three sources: CPU, DMA, and refresh requests. These accesses are arbitrated based on the inputs HLDA1 and REF and are handled by three state machines controlling each type of accesses. The CPU cycle state machine controls the memory operation for CPU accesses, the DMA cycle state machine for DMA accesses, and the refresh cycle state machine controls the DRAM refresh operation.

The refresh state machine is in control whenever REF is active. When HLDA1 is active the DMA state machine is in control. In all other cases, the CPU state machine is in control for valid DRAM memory accesses as defined by the memory map in the configuration registers. The arbitration is not preemptive in that the current active state machine always runs to completion before relinquishing the control. Therefore, it is possible for the HLDA1 with active XMEMW or XMEMR to prevent refresh cycles to take place.

CPU Access State Machine

The CPU initiated accesses are decoded according to the memory map defined in the configuration registers. These are the only accesses that uses the page mode operation of the DRAMs. The 82C302 maintains four page registers storing the page addresses of the most recently accessed DRAM pages of the two-way page-interleaved banks. These four registers are called active page registers. Accesses to the active pages are called "hits" and are faster because the DRAM is operated in the page mode with the RAS staying asserted.

The 82C302 supports memory configurations with either one, two, or four banks. Since one active register is provided for each bank, the number of active pages varies with the amount of memory installed. In a non-interleaved minimum memory configuration only one active page register is in use. For each active page register in use, the corresponding RAS stays asserted after the previous access. If an access does not hit any active pages, a "miss" cycle, normal DRAM access cycle is entered by first de-asserting the RAS associated with the bank accessed. Refer to the timing diagram for the timing sequence for each of these cases.

RAS and CAS Generation

The 82C302 is based on 2K byte page-interleaved organization. To maintain this organization, the following table shows the address lines used for the different organizations:

CHIPS

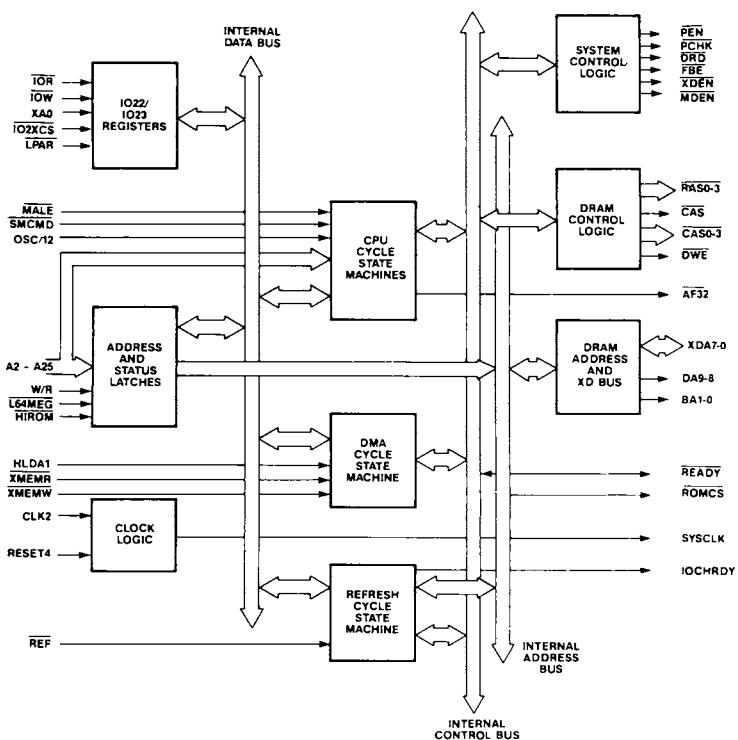


Figure 2-1. 82C302 Functional Block Diagram

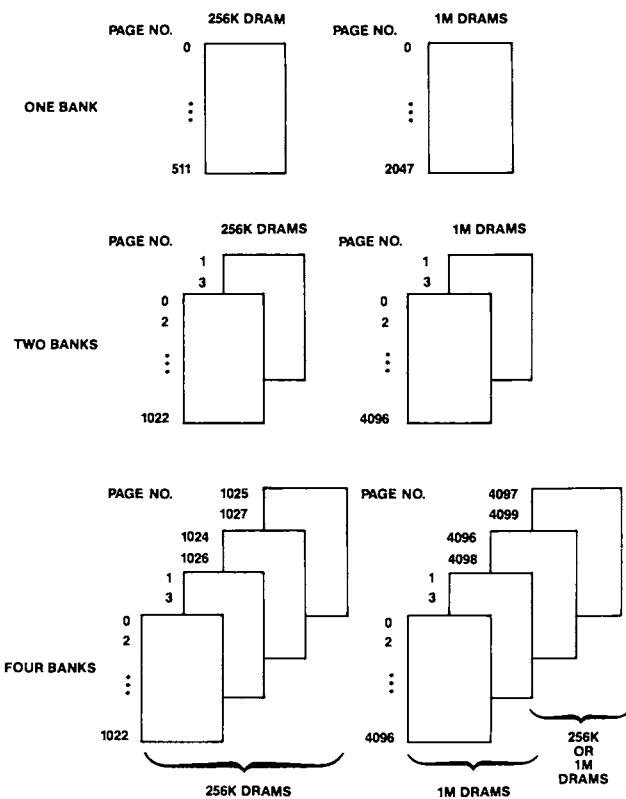


Figure 2-2. Memory Addressing

CHIPS

For non-interleaved operation (one bank only):

	Row	Column
256K DRAM's	A<19:11>	A<10:2>
1M DRAM's	A<21:12>	A<11:2>

For interleaved memory(two or four banks):

	Row	Column
256K DRAM's	A<20:12>	A<10:2>
1M DRAM's	A<21:12>	A<22>, A<10:2>

Table 2-1. Row and Column Address Definition

In interleaved memory cases bit A<11> determines which one of the even page banks or odd page banks is accessed in the two-way interleaved organization. For configurations using only 256K DRAM's, A<11> and A<21> are used to control RAS<3:0>; and for 1M DRAM only configurations A<11> and A<23> are used. When 256K and 1M DRAMs are used, it is required that the 1M DRAMs occupy the first two-banks and the 256K DRAMs occupy the second two-banks. This constrain is there to ensure that the there will not be a hole in the address space without actual DRAM's. Figure 2-2 shows the memory addressing scheme for the allowable memory configurations.

RAS Timeout

When using DRAM page mode, the maximum RAS pulse width must be observed. For most DRAMs this is 10 microseconds (although some have 30 or 100 microsecond limits). Timers are maintained for each bank to assure data integrity using the OSC/12 (1.19MHz = 840nS) clock available on the system board. RAS is de-asserted for each bank when its counter times out at about 10 microseconds intervals. The configuration register bit REG13H<7> can be programmed to set desired RAS time out intervals.

CPU Access Cycles Sequences

There are many basic CPU memory access patterns: memory read-hit access, memory write-hit access, memory read-miss access, and memory write-miss access, and CPU IO access to 82C301 configuration registers. These basic access sequences and timing for the critical signals are shown in the timing charts. In addition to these basic patterns, the configuration register REG13H<6> may be programmed to have one wait state inserted for supporting slow DRAM's. Note that the default setting after the system reset is for one wait state insertion.

DMA Access State Machine

DMA accesses are initiated by asserting HLDA1. The XMEMR an XMEMW determines if it is a read or a write memory access. The bytes accessed are controlled externally with the BE<3:0> signals generated by the 82C301 Bus Controller. The DMA state machine makes one memory access per DMA bus cycle and does not attempt to pack or unpack data transfers to make full 32-bit transfers. Refer to the timing charts for a DMA access cycle sequence and timing.

Dynamic RAM Access Logic

The DRAM control logic generates the necessary RAS, CAS and DWE signals for all DRAM accesses. CPU, DMA, and refresh access cycles use DA<9:8> and XDA<7:0> and bank select signals BA<1:0>. (Note that in the current version of 82C302, the signals BA<1:0> default to zero.) The system control logic provides MDEN to control the buffer chips. MDEN enables data buffers for MD bus in 82A305's for non-refresh DRAM access cycles.

Refresh Cycles

To reduce power supply noise generation due to the surges caused during RAS transitions, RAS pulses to each bank are staggered by one CLK2 cycle, as shown in Figure 2-3. Because all RAS's could be active for page mode operation, a refresh cycle requires that all RAS's be first de-asserted then asserted with the correct refresh address.

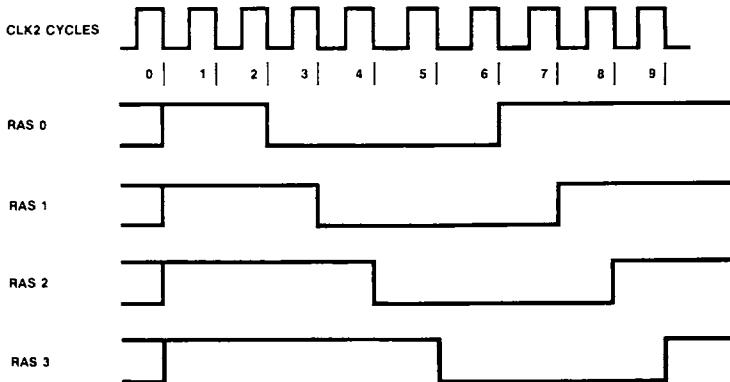


Figure 2-3. Staggered RAS pulses during refresh

Memory Mapping Logic

The configuration registers REG08H to REG13H define what is a valid local memory access, and what is a ROM memory access according to the local bus addresses. REG08H and REG09H determines how ROM areas (as defined by an IBM PC AT) between 768K to 1M address range are accessed.

For valid local memory accesses it asserts the AF32 to indicate that it has control of the local bus and also asserts the READY signal at the end of the access cycle. If an access is a ROM access, it asserts ROMCS to provide controls for the ROM's or PROM's; in this case, the READY signal must be provided to the CPU and 82C302 by another source (82C301 will provide this signal in a chip set solution).

Clock, Reset and Other Miscellaneous Logic

The RESET4 input causes all internal registers to be reset to their default values. Configuration registers not specified with a default value is not reinitialized and may not retain its old value. The system control logic generates the PCHK and PEN signals to be used for enabling parity error checking.

Configuration/Diagnostic Registers

There are 14 bytes of configuration and diagnostic registers in the 82C302. These are accessed through IO ports 22H and 23H normally found in the interrupt controller. Accesses to these configuration and diagnostic registers are done first by writing the index of the desired register into port 22H and then followed by an access (either read or write) to 23H for the data. XDEN is asserted for these accesses to control the buffer connecting the XD and XDA buses.

Memory Configuration Registers

The configuration registers REG08H to REG0FH are used to control how the CPU memory accesses are defined. They define all address as ROM accesses, system memory accesses (or DRAM accesses for short), other local CPU bus accesses, or as IO channel accesses. These provisions are made because the low one megabyte is both occupied by DRAM's, ROM's and also devices on the AT bus. For ROM accesses it generates the ROMCS to control the PROM access; for system memory accesses it generates the necessary DRAM controls to the system memory

CHIPS

under its control; it generates AF32 for all other local CPU bus accesses; and it does not control the IO channel accesses.

The 82C301 provides three 256KB areas where the ROM's can be located. The low ROM space is located just below the 1MB address, the middle ROM space is located below 16MB address, and the high ROM space is below 4GB address. The low ROM is used for 8086 compatible operation, the middle ROM is for 80286, and the high ROM is for 80386. Upon system reset, the default configuration register setting causes accesses to these three ROM areas to generate ROMCS. With the exception of the high ROM area which is always recognized as ROM accesses, the other two ROM areas can be mapped to be either ROM or RAM accesses.

After reset, REG08<4:3> may be programmed to make the entire middle ROM area mapped to DRAM and with write protection if desired. REG08H<2> determines if the 82C301 recognizes the addresses generated beyond 16 MB as local CPU bus cycles. REG08H<1> is used to enable REG0AH to REG0FH which controls the "Low Meg DRAM" (40000H to FFFFFH) address mapping for 256KB to 1MB addresses in 16 KB blocks. This bit defaults upon reset so that only the 0 to 256KB areas are accessible.

Accesses to the low megabyte DRAM can be made by enabling the mapping after the necessary configuration registers are correctly programmed. REG08H<0> defaults to single bank memory configuration upon reset and must be programmed to enable page/interleaved operation.

The REG09H control the address mapping and write protection for the low ROM area (from C0000H to FFFFFH) in 64KB blocks. REG0AH to REG0FH define for each 16 KB address range if it is a DRAM block in the system memory or on the IO channel.

INDEX	7	6	5	4	3	2	1	0	
D8H	0	VERS	MW	MR	HM	SM	N1		IDENTIFICATION
D9H	R3	R2	R1	R0	D3	D2	D1	D0	ROM CONFIGURATION
DAH	358K				256K				MEMORY ENABLE (16KB RESOLUTION)
DBH	496K				384K				*
0CH	624K				512K				*
ODH	752K				640K				*
OEH	880K				768K				*
OFH	1008K				896K				MEMORY ENABLE (16KB RESOLUTION)

Figure 2-5. Control and Address Space Map Register Summary



Index	Bits	Function
08H		Identification
	7	Controller Type Part type
	0	Interleaved Memory Controller (82C302)
	6:5	Version
	0	Initial
	4	MW - Middle Boot Space Write Protect. This bit is used in conjunction with bit 3 allowing the BIOS code to be copied into RAM and write protected at this location as well as below 1MB. It should only be used if there is RAM present at this address (16MB installed). Executing out of RAM will result in better performance than out of narrower (usually 8 or 16 bits) EPROMs.
	0	Read/Write of 256KB RAM at 16128K 00FC0000H. Default.
	1	Read-Only of 256KB RAM at 16128K 00FC0000H
	3	MR - Middle Boot ROM disable
	0	The boot/BIOS ROM located just below 16MB is enabled. This is necessary for 286 compatibility. Default.
	1	The boot/BIOS ROM located just below 16MB is disabled.
	2	HM - 16MB IO Channel Memory Limit
	0	AF32 will not be asserted for addresses \geq 16MB. This should only be used if external logic can recognize addresses above 16MB. Default.
	1	AF32 is asserted for addresses \geq 16MB (01000000H). Since IO channel memory cannot normally be configured above 16MB, accessing above 16MB will cause a READY timeout if that feature is enabled. This is necessary during setup because memory address above 16MB that are not enabled for local memory could wrap into a valid IO channel memory location.
	1	SM - Minimum memory configuration after reset. Used during initialization.
	0	256K only enabled. Default. Ignore memory address configuration registers 0AH to 0FH.
	1	Normal configuration controlled by registers 0AH to 0FH.
0	NI -	Single bank/interleave select
	0	Disable interleave (single bank). Default.
	1	Enable interleave

Table 2-2. Memory Configuration Register Definition

CHIPS

Index	Bits	Function
09H		RAM/ROM Configuration in boot area.
	7	RAM at 768K C0000-CFFFFH (EGA)
	6	RAM at 832K D0000-DFFFFH
	5	RAM at 896K E0000-EFFFFH
	4	RAM at 960K F0000-FFFFFH (BIOS)
		Bits 7:4 disable writing to RAM located in the BIOS area in 64KB blocks. BIOS data.
	0	Read/Write. Default.
	1	Read-Only
	3	ROM at 768K C0000-CFFFFH (EGA)
	2	ROM at 832K D0000-DFFFFH
	1	ROM at 896K E0000-EFFFFH
	0	ROM at 960K F0000-FFFFFH (BIOS)
		Bits 3:0 enable substitution of the BIOS ROM located below 1MB with RAM at the same location in 64KB blocks. This should be done after the BIOS code is copied from the ROM and the RAM locations have been write protected using bits 7:4.
	0	Disabled
	1	Enabled. Default.
0AH		Address Map 256K 040000-05FFFFH (16K Resolution)
0BH		Address Map 384K 060000-07FFFFH
0CH		Address Map 512K 080000-09FFFFH
0DH		Address Map 640K 0A0000-0BFFFFH
0EH		Address Map 768K 0C0000-0DFFFFH
0FH		Address Map 896K 0E0000-0FFFFFH
	0	Address is on or controlled by the system board
	1	Address is on the IO Channel.
		This permits 16K blocks of memory to be disabled allowing ROMs, memory expansion mechanisms (EMS or XMA) or memory mapped IO devices to reside within the lower 1MB address space.

Table 2-2. Memory Configuration Register Definition (Continued)

DRAM Array Configuration and Timing

The configuration registers REG10H to REG13H provides the DRAM type definition and starting address for each pair of banks, banks 0 and 1, and banks 2 and 3. The REG10H<7:6> and REG12H<7:6> defines if the DRAM's are enabled, uses 256K DRAM's, or uses 1M DRAM's. These bits defaults to 256K DRAM's upon reset. The REG10H<6:0> and REG12H<6:0> defines the address bits <25:20> of the starting address of the pairs of banks. Some of these bits may not be valid because the memory banks must start at some predefined boundaries. For 256K DRAM's, all bits <25:20> are valid if only single bank is enabled-it can be on any 1MB boundary; otherwise only bits <25:21> are valid starting address bits on 2MB boundaries. For 1M DRAM's, only bits <25:23> are valid forcing it on 8 MB boundaries. The REG11H<7> and REG13H<7> define the RAS precharge time

required when a page miss occurs so that DRAM's of different speeds can be supported for each pair of banks. The REG11H<6> and REG13H<6> define the wait state to be inserted to meet the DRAM speed. These parameters default to the slower timing upon reset so that the system can be powered up with minimal assumptions on the DRAM speed and the memory configuration. Refer to Table 2-3 for details of the bit definitions.

INDEX	7	6	5	4	3	2	1	0	
10H									BANKS 0/1
11H	RP	WS							
12H									BANKS 2/3
13H	RP	WS							

Figure 2.6. DRAM Configuration/Timing Register Summary

Index	Bits	Function
10H		bank 0/1 Type/Start Address
12H		bank 2/3 Type/Start Address
7:6		DRAM Type
	0	none (bank disabled)
	1	256K words, default value for REG10H and REG12H
	2	1M words
	3	Reserved
5:0		Starting Address 25:20 The DRAM type determines which address bits are valid in the address recognition process. This field of REG10H defaults to zero after reset.
	25:20	256K DRAM's. 1MB boundary 1MB per bank, single bank only. Valid for the first register only.
	25:21	256K DRAM's. 2MB boundary 1MB per bank, two banks required for interleaved operation.
	25:23	1M DRAM's. 8MB boundary 4MB per bank, two banks required for interleaved operation.

Table 2-3. DRAM Configuration and Timing Register Definition

CHIPS

Index	Bits	Function
11H		banks 0/1 Timing
13H		banks 2/3 Timing
	7	DRAM RAS precharge. Specifies the amount of time for RAS precharge when a page miss occurs.
	0	3 CLK2 times (93 nS at 16MHz)
	1	5 CLK2 times (155nS at 16MHz). Default.
	6	Access wait states Specifies the number of wait states in SCLK units to allow the use of slower DRAMs.
	0	0 wait-states
	1	1 wait-states. Default.
	5:0	Reserved

Table 2-3. DRAM Configuration and Timing Register Definition (Continued)

Diagnostic Access Register

REG28H<7> controls the parity check enable and defaults to "disable" after reset. This bit generates the PEN signal for enabling the parity check by 82A306. When parity errors occur REG28H<1:0> and REG29H<7:0> will latch the error address <25:16>.

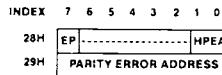


Figure 2.7. Diagnostic Access Register Summary

Index	Bits	Function
28H		Error Source/Address (MSBs)
	7	Parity check disable
	0	Enabled
	1	Disabled
	6:2	Not used, returns unpredictable value.
	1:0	High Parity Error Address bits <25:24>
29H		Parity Error Address (LSBs)
	7:0	Error address bits <23:16>

Table 2-4. Diagnostic Access Registers Definition

CHIPS

82C302 Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V_{CC}	—	7.0	V
Input Voltage	V_I	-0.5	5.5	V
Output Voltage	V_O	-0.5	5.5	V
Operating Temperature	T_{OP}	-25	85	C
Storage Temperature	T_{STG}	-40	125	C

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

82C302 Operating Conditions

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V_{CC}	4.75	5.25	V
Ambient Temperature	T_A	0	70	C

82C302 DC Characteristics

Parameter	Symbol	Min.	Max.	Units
Input Low Voltage	V_{IL}	—	0.8	V
Input High Voltage	V_{IH}	—	2.0	V
Output Low Voltage $I_{OL}=8\text{mA}$ (Note 1)	V_{OL}	—	0.45	V
Output High Voltage $I_{OH}=-200\ \mu\text{A}$	V_{OH}	—	2.4	V
Input Current $0 < V_{IN} < V_{CC}$	I_{IL}	—	± 10	μA
Output Short Circuit Current $V_O=0\text{V}$	I_{OS}	TBD	TBD	mA
Input Clamp Voltage	V_{IC}	—	TBD	V
Power Supply Current @ 8 MHz Clock	I_{CC}	—	20	mA
Output HI-Z Leak Current	I_{OZ1}	—	± 10	μA

NOTE:

1. SYSCLK, DWE, RAS<3:0>, CAS, CAS<3:0> have $I_{OL}=8\text{mA}$. All other outputs and I/O pins have $I_{OL}=4\text{mA}$. In all cases all $I_{OL}=I_{OH}$ for the pin.

CHIPS

82C302 AC Characteristics

($T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5\text{V} \pm 5\%$)

Sym	Description	Min.	Typ.	Max.	Units
t200	CLK2 input cycle time (t0 used as reference)	31	42		ns
t201	CLK2 fall time	2	5		ns
t202	CLK2 rise time	2	5		ns
t203	CLK2 low time	10			ns
t204	CLK2 high time	10			
t205	RESET hold time	6			ns
t206	RESET set-up time	5			ns
t207	SCLK delay time	6	12	22	ns
DMA Sequence					
t210	RAS <i>i</i> de-assertion time from HLDA1	13			ns
t211	RAS <i>i</i> active delay from commands active	16			ns
t212	Address set-up time to commands active	35			ns
t213	Address hold time from commands inactive	0			ns
t214	AF32 active time from commands active	26			ns
t215	DRD active time from commands active	10			ns
t216	Row address set-up time to RAS active	10			ns
t217	Row address hold time from RAS active	0.5t0			ns
t218	CAS <i>i</i> active delay from RAS active for DMA memory read cycle	1.0t0			ns
t219	CAS <i>i</i> active delay from RAS active for DMA memory write cycle	1.5t0			ns
t220	DWE active delay from RAS active	0.5t0			ns
t221	READY active delay from RAS active	1.5t0			ns
t222	RAS <i>i</i> de-assertion time from commands inactive	13			ns
t223	Column address hold time from commands active	29			ns
t224	CAS <i>i</i> de-assertion from commands inactive	22			ns
t225	AF32 tri-state delay from commands inactive	21			ns
t226	DWE de-assertion time from commands inactive	11			ns
t227	READY de-assertion time from commands inactive	16			ns
t228	DRD de-assertion time from commands inactive	12			ns

82C302 AC Characteristics (Continued)
 $(T_A = 0^\circ\text{C} \text{ to } 70^\circ\text{C}, V_{CC} = 5V \pm 5\%)$

Sym	Description	Min.	Typ.	Max.	Units
ROMCS Sequence					
t235	ROMCS active delay from CLK2I		27		ns
t236	ROMCS inactive delay from CLK2I		22		ns
t237	READY input set-up time to CLK2I	8			ns
t238	READY input hold time from CLK2I	0			ns
Refresh Sequence					
t240	IOCHRDY going low from REF active		14		ns
t241	IOCHRDY floating from CLK2I		18		ns
t242	RAS0 precharge time		3t0		ns
t243	RASI (i = 0 to 3) pulse width		4t0		ns
t244	RAS(i+1) active delay from RASI active		110		ns
t245	Refresh address set-up time to RASI		3t0		ns
t246	Refresh address hold time from RASI		2t0		ns
t247	RASI inactive delay from CLK2I		14		ns
t248	RASI active delay from CLK2I		15		ns
IO Read/Write Sequence					
t250	IO2XCS set-up time to XIORI or XIOWI	10			ns
t251	XAO set-up time to XIORI or XIOWI	10			ns
t252	IO2XCS hold time from XIORI or XIOWI	15			ns
t253	XAO hold time from XIORI or XIOWI	15			ns
t254	XDEN active delay from XIORI or XIOWI		15		ns
t255	XDEN inactive delay from XIORI or XIOWI		12		ns
t256	XDA<7:0> input set-up time to XIOWI	10			ns
t257	XDA<7:0> input hold time to XIOWI	8			ns
t258	XDA<7:0> output valid delay from XIORI		37		ns
t259	XDA<7:0> hold time from XIORI		14		ns
CPU to Memory Sequence					
t260	MALE active set-up time to CLK2I		TBD		
t261	MALE inactive delay from CLK2I		TBD		
t262	Address/Status set-up time to CLK2I		TBD		

CHIPS

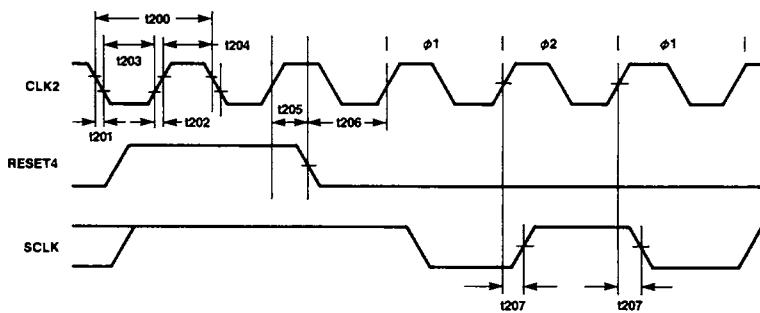
82C302 AC Characteristics (Continued)

($T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5V \pm 5\%$)

Sym	Description	Min.	Typ.	Max.	Units
t263	Address/Status hold time from MALE!			TBD	
t264	L64MEG, HIROM set-up time to CLK2!			TBD	
t265	L64MEG, HIROM hold time to MALE!			TBD	
t266	SMCMD active delay from MALE!			TBD	
CPU Cycle Timing					
t270	AF32 active delay from CLK2!	26		ns	
t271	AF32 inactive delay from CLK2!	17		ns	
t272	CASi active delay from MALE! for read-hit cycle	19		ns	
t273	CASi inactive delay from CLK2!	16		ns	
t274	CAS active delay from MALE! for read-hit cycle	20		ns	
t275	CAS inactive delay from CLK2!	18		ns	
t276	Column Address stable from MALE!	25		ns	
t277	DRD active delay from CLK2!	21		ns	
t278	DRD inactive delay from CLK2!	17		ns	
t279	FBE active delay from CLK2!	23		ns	
t280	FBE inactive delay from CLK2!	19		ns	
t281	READY active delay from CLK2!	20		ns	
t282	READY inactive delay from CLK2!	18		ns	
t283	RASI active delay from CLK2!	18		ns	
t284	Row address set-up time to RASI!	10		ns	
t285	Row address hold time from CLK2!	12		ns	
t286	CASi active delay from CLK2!	17		ns	
t287	CAS active delay from CLK2!	18		ns	
t288	RASI inactive delay from CLK2!	17		ns	
t289	RASI precharge time	3t0		ns	
t290	CASi precharge time	1.5t0		ns	
t291	DWE active delay from CLK2!	16		ns	
t292	DWE inactive delay from CLK2!	17		ns	

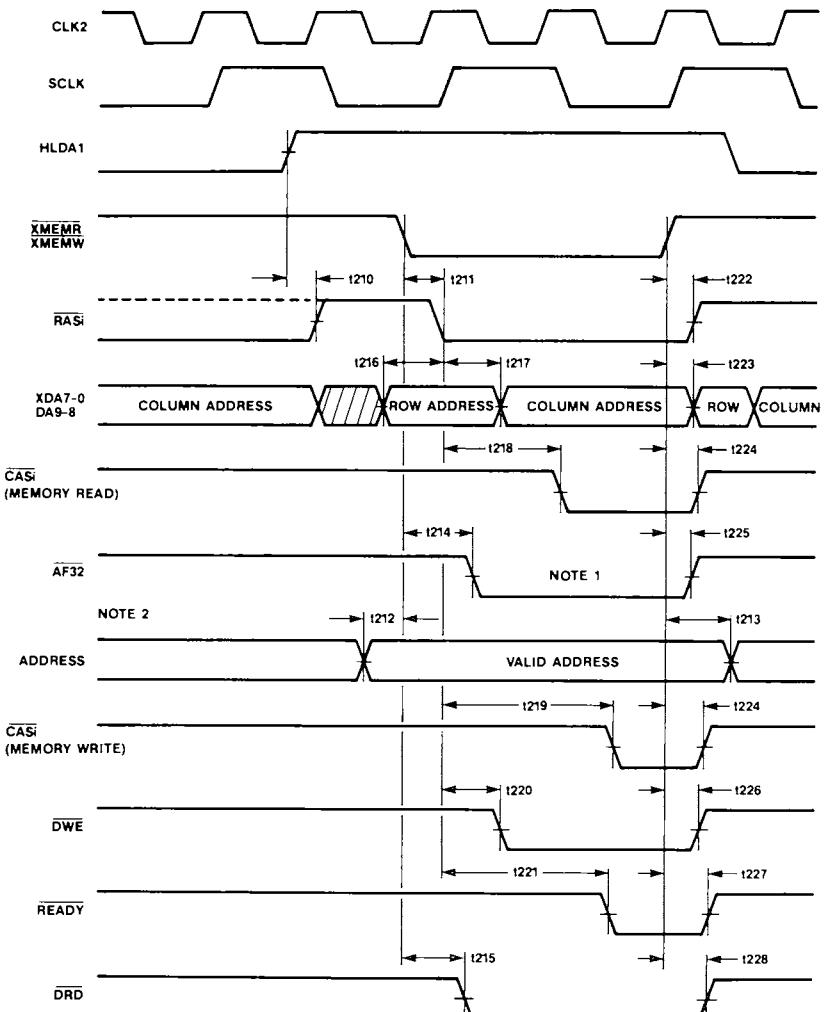


82C302 TIMING DIAGRAM (RESET SEQUENCE)



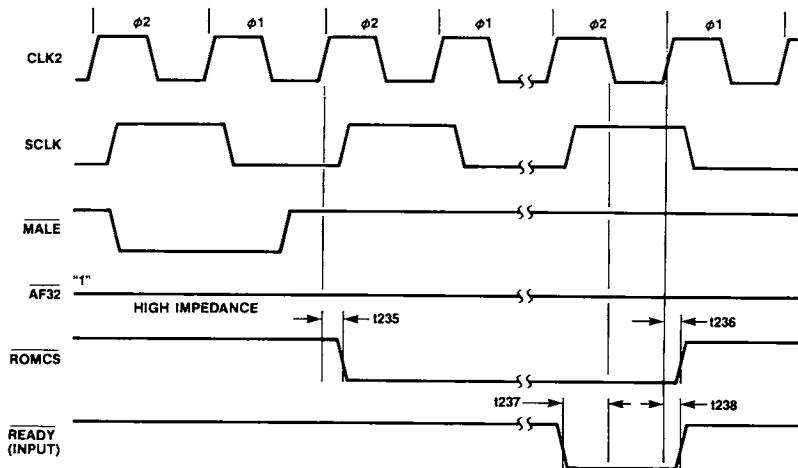
CHIPS

82C302 TIMING DIAGRAM (DMA CYCLES)



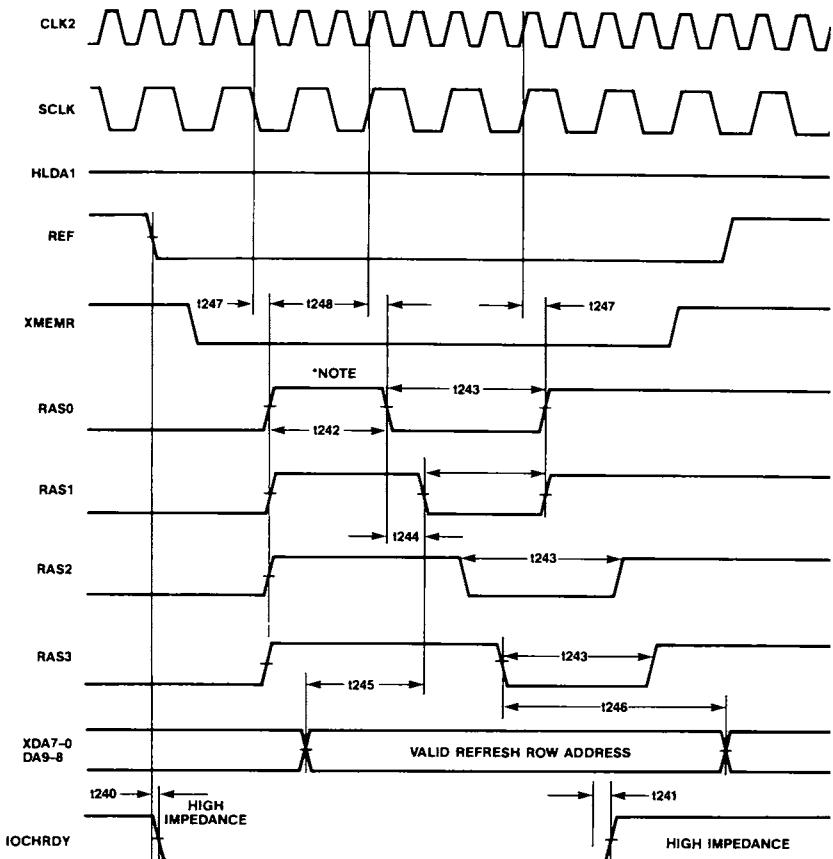
NOTE

1. During DMA cycles, only RASi will become active if AF32 = 1 (not local memory)
2. The address includes L64MEG and HIROM input signals.

82C302 TIMING DIAGRAM (ROM READ CYCLE)

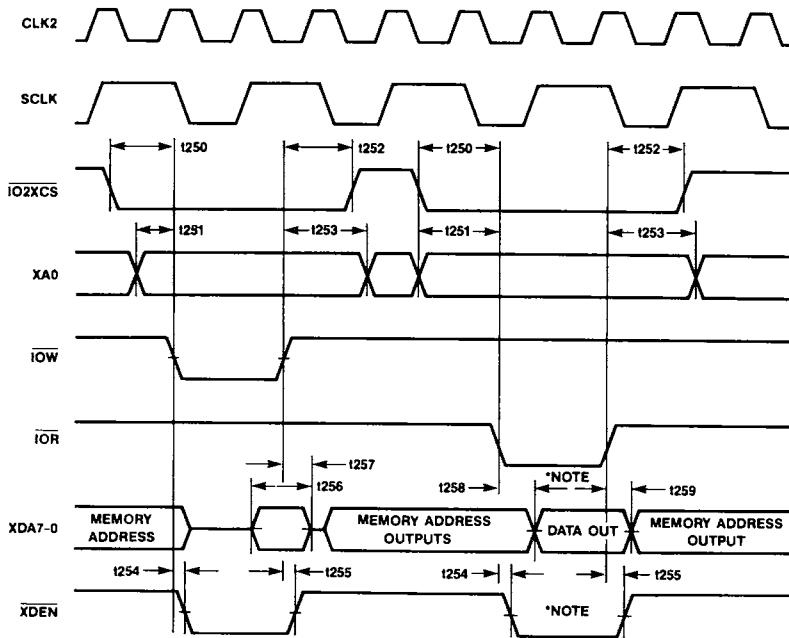
CHIPS

82C302 REFRESH CYCLE WAVEFORM



* NOTE: Add 2 more clock cycles if either Bit 7 of register 11 is 1 or Bit 7 of register 13 is 1.

82C302 TIMING DIAGRAM (IO READ/WRITE)

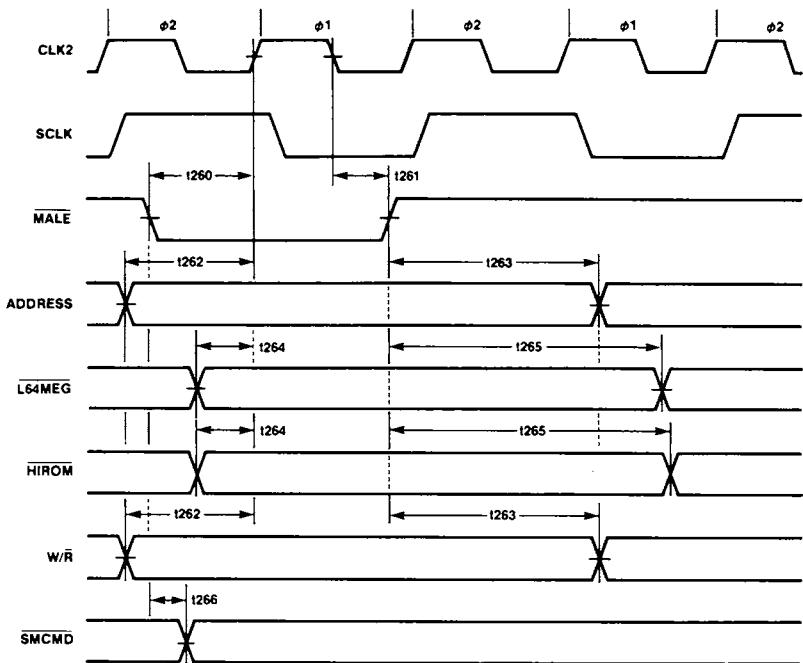


***NOTE:** No data output and XDEN is inactive if the index set up by the previous IO22 Write doesn't point to a valid IO23 register of 82C302.

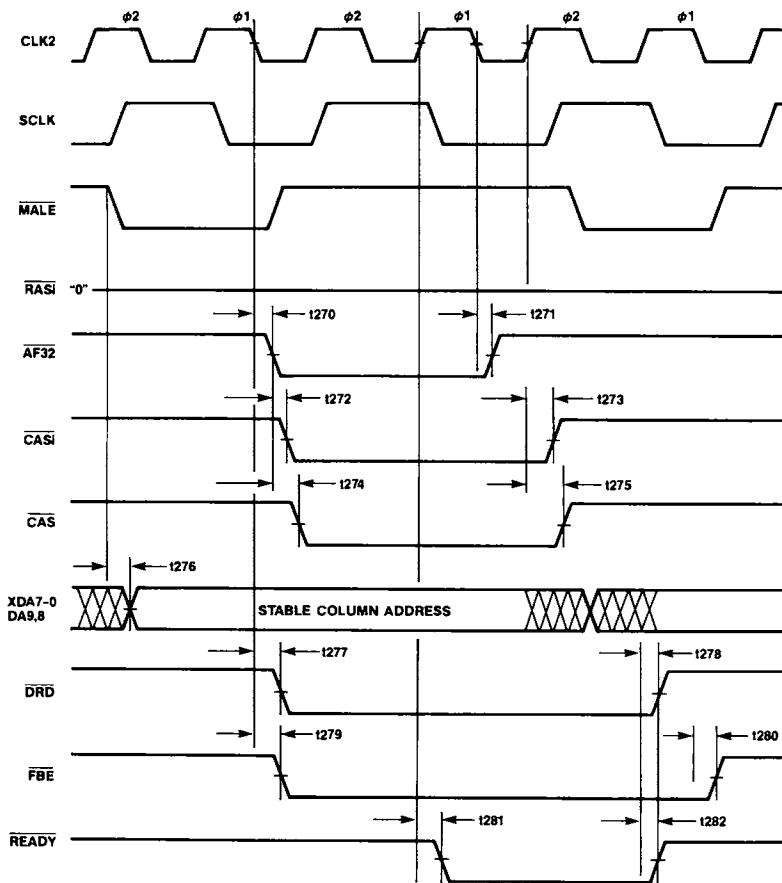
Valid registers of IO23: 08H-0FH, 10H-13H, 28H-29H.

CHIPS

82C302 TIMING DIAGRAM (INPUT SETUP/HOLD TIME FOR CPU CYCLES)



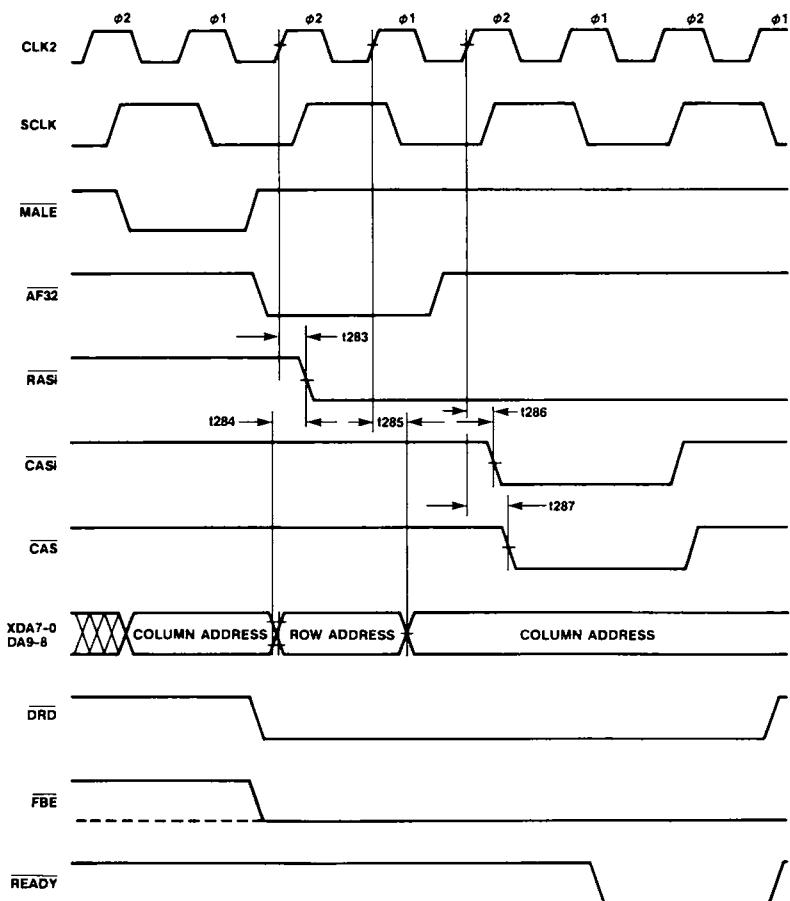
82C302 TIMING DIAGRAM (CPU TO LOCAL MEMORY CYCLE) READ HIT, ϕ WAIT STATE



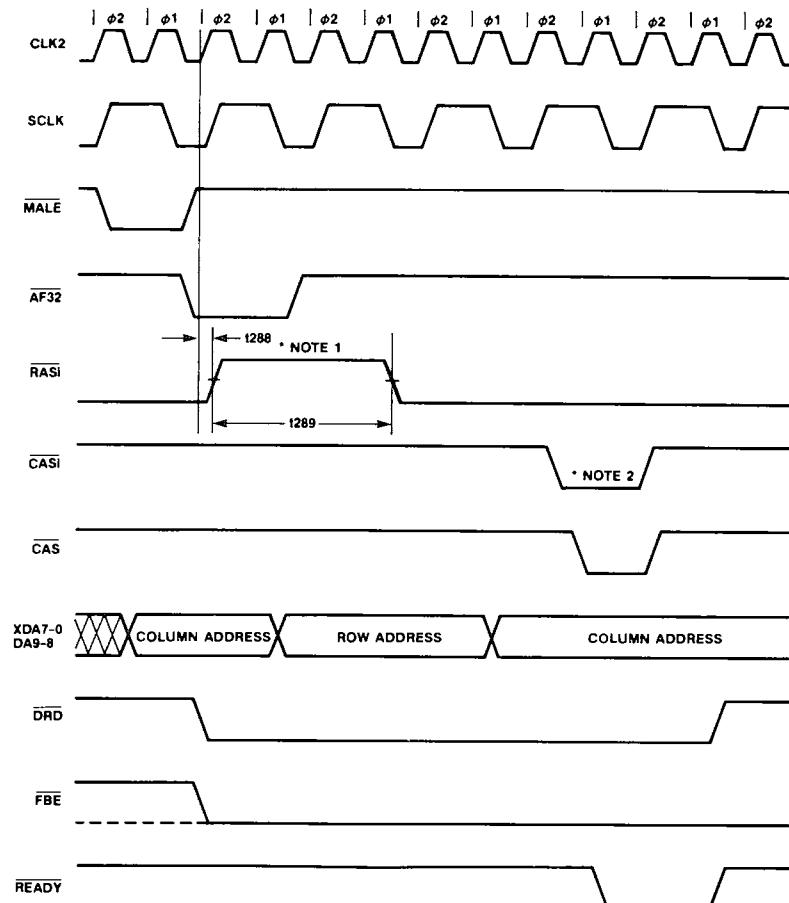
NOTE: Assume bit 6 of Register 11 (or 13) is programmed to "0".

CHIPS

82C302 TIMING DIAGRAM (CPU TO LOCAL MEMORY)
READ CYCLE WITH RAS BEING INACTIVE



82C302 TIMING DIAGRAM (CPU TO LOCAL MEMORY) READ MISS CYCLE

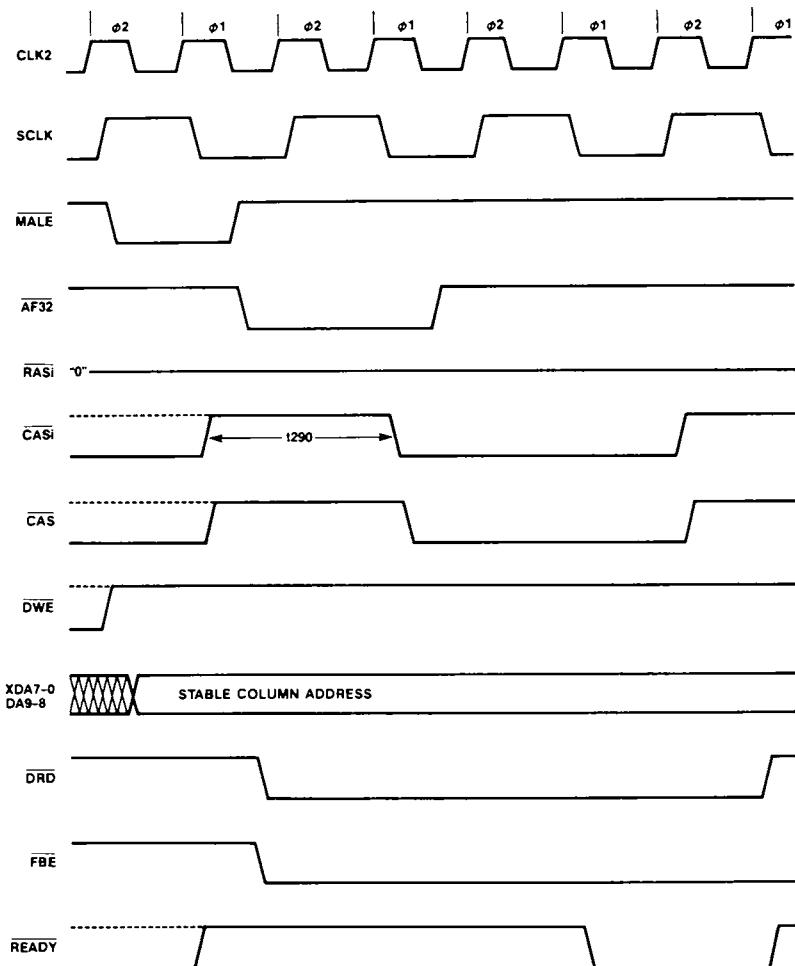


NOTES

1. RASI precharge time will be increased by 2 CLK2 cycles if bit 7 of Register 11 (or 13) is programmed to "1".
2. CASI (and CAS) pulse width will be increased by 2 CLK2 cycles if bit 6 of registers 11, (or 13) is programmed to "1".

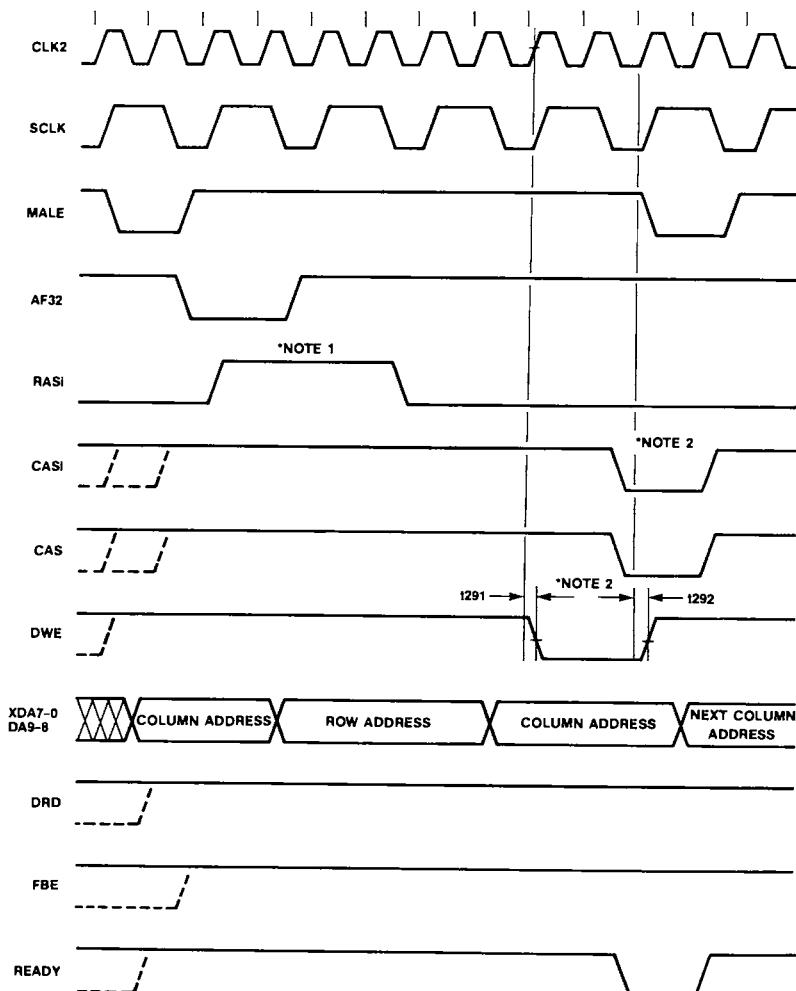
CHIPS

82C302 TIMING DIAGRAM (CPU TO LOCAL MEMORY CYCLE) READ HIT FOLLOWING A WRITE OR 1 WAIT STATE READ-HIT CYCLE



NOTE: The read cycle will be identical to this waveform (regardless whether the previous cycle is a Write or not) if bit 6 of Register 11 (or 13) is programmed to "1".

82C302 TIMING DIAGRAM (CPU TO LOCAL MEMORY) WRITE MISS CYCLE



NOTES

1. RASI precharge time will be increased by 2 CLK2 cycles if bit 7 of Register 11 (or 13) is programmed to "1".
2. CASI (and CAS) pulse widths will be increased by 2 CLK2 cycles if bit 6 of register 11, (or 13) is programmed to "1".

82A303 HIGH ADDRESS BUFFERS

- Buffer for bits 31:12 of the Local, X and System address buses.
- X and S address bus can be extended to 27 bits (128MB).
- Direct interface to AT Bus
- Advanced Schottky TTL technology

FUNCTIONAL DESCRIPTIONS

The 82A303 as shown in figure 3-1 provides two functions:

- Generation of address decoding signals required by other chips.
- Interface between the local, X and System address bus.

Address Decode

The address decoding circuit provides as outputs LIOCS, LMEGCS, L64MEG, and HIROM. These signals are active if the address accesses satisfy the conditions defined in table 3-1. The signal decodes for LIOCS and LMEGCS are controlled by HLDA1 and latched on the trailing edge of MALE. The L64MEG and HIROM are simply decoded from the address signals.

Signal	Decode Condition
LIOCS	A<15:12> = 00H
LMEGCS	A<31:20> = 00H
L64MEG	A<31:26> = 00H
HIROM	A<31:26> = 3FH

Table 3-1. High Address Decodes Definition

Address Bus Interfaces

The 82A303 interconnects the local, X and system address buses with bidirectional drivers connecting each bus and the internal buses. These drivers have 24mA current drives for direct connection to the system address bus. The table 3-2 shows how the drivers are configured between the buses for each type of active bus requests. Note that the default configuration is set up so that the CPU address bus drives the memory address bus for local memory CPU access cycles.

For all CPU sourced accesses, the addresses are latched on the trailing edge of MALE.

Active	Source	Target
HLDA1	XA	SA, A, MA
MASTER	SA	XA, A, MA
REF	-	SA driven LO
ATEN	A	XA, SA, MA
default	A	MA

Table 3-2. High Address Bus Control

27 bit Address Extensions

The standard AT implementation supports only 24-bit addresses. The CS 8230 allows for address extension on the SA and XA buses to 27 bits (128MB). This is done by grounding the enable pin XBHE for XA bus and SBHE for SA bus. Internal pullups are provided so that if the enable pins are left unconnected bits 24 to 27 of the respective bus are forced LO.

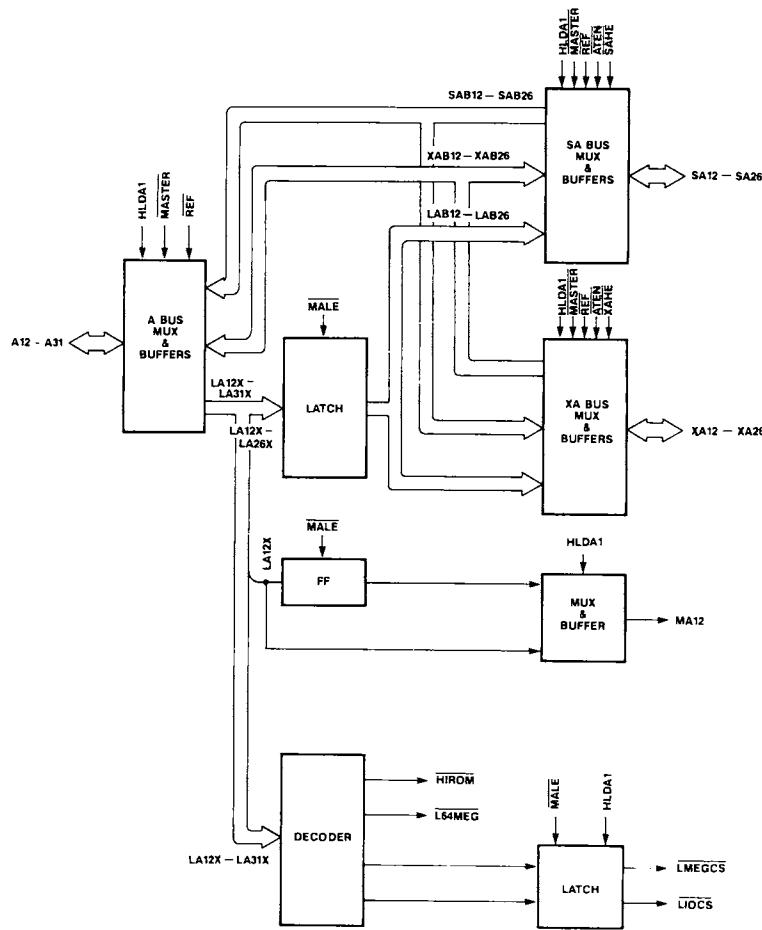


Figure 3-1. 82A303 Functional Block Diagram

CHIPS

82A303 Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V_{CC}	—	7.0	V
Input Voltage	V_I	-0.5	5.5	V
Output Voltage	V_O	-0.5	5.5	V
Operating Temperature	T_{op}	-25	85	C
Storage Temperature	T_{stg}	-40	125	C

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

82A303 Operating Conditions

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V_{CC}	4.75	5.25	V
Ambient Temperature	T_A	0	70	C

82A303 DC Characteristics

Parameter	Symbol	Min.	Max.	Units
Input Low Voltage	V_{IL}	—	0.8	V
Input High Voltage	V_{IH}	—	2.0	V
Output Low Voltage $I_{OL}=10\text{mA}$ (Note 1)	V_{OL1}	—	0.5	V
Output Low Voltage $I_{OL}=24\text{mA}$ (Note 2)	V_{OL2}	—	0.5	V
Output High Voltage $I_{OH}=-3.3\text{mA}$ (Note 3)	V_{OH}	2.4	—	V
Input Low Current $V_I = 0.5\text{V}$, $V_{CC} = 5.25\text{V}$	I_{IL}	—	-200	μA
Input High Current $V_I = 2.4\text{V}$, $V_{CC} = 5.25\text{V}$	I_{IH}	—	20	μA
Input High Current $V_I = 5.5\text{V}$, $V_{CC} = 5.25\text{V}$	I_I	—	200	μA
Output Short Circuit Current $V_O=0\text{V}$	I_{OS}	-15	-100	mA
Input Clamp Voltage $I_I = -18\text{mA}$, $V_{CC} = 4.75\text{V}$	V_{IC}	—	-1.5	V
Power Supply Current	I_{CC}	140	230	mA
Output Hi-Z Leak Current 3-State Output Pins	I_{OZ1}	-100	100	μA
Output Hi-Z Leak Current Bidirectional Pins	I_{OZ2}	-300	120	μA

NOTES

1. All bus outputs other than SA<20:12> and XA<26:21>
2. All SA<20:12> and XA<26:21> have $I_{OL} = 24\text{mA}$.
3. All outputs and bidirectional pins.

82A303 AC Characteristics(T_A = 0°C to 60°C, V_{CC} = 5V ± 5%)

Sym	Description	Min.	Typ.	Max.	Units
t301	A to MA input set-up time to <u>MALE</u> †			TBD	
t302	A to MA input hold time from <u>MALE</u> †			TBD	
t303	MA output valid delay from <u>MALE</u> †	5		33	ns
t304	A to SA,XA input set-up time to <u>MALE</u> †			TBD	
t305	A to SA,XA input hold time from <u>MALE</u> †			TBD	
t306	SA output valid delay from ATEN active	8		33	ns
t307	SA tri-state delay from ATEN inactive	6		28	ns
t308	XA output valid delay from ATEN active	11		42	ns
t309	XA tri-state delay from ATEN inactive	9		38	ns
t310	HIROM decode active from A<32:26> valid	4		19	ns
t311	HIROM decode inactive from A<32:26> invalid	2		16	ns
t312	L64MEG decode active from A<32:26> valid	4		19	ns
t313	L64MEG decode inactive from A<32:26> invalid	2		17	ns
t314	LIOCS decode active from MALE active	6		26	ns
t315	LIOCS decode inactive from MALE active	4		23	ns
t316	LMEGCS decode active from MALE active	6		26	ns
t317	LMEGCS decode inactive from MALE active	4		23	ns
t318	A data valid delay from SA data valid	4		27	ns
t319	XA data valid delay from SA data valid	7		35	ns
t320	MA data valid delay from SA data valid	9		49	ns
t321	LIOCS decode active from SA data valid	13		50	ns
t322	LIOCS decode inactive from SA data invalid	10		39	ns
t323	L64MEG decode active from SA data valid	12		47	ns
t324	L64MEG decode inactive from SA data invalid	9		35	ns
t325	LMEGCS decode active from SA data valid	14		53	ns
t326	LMEGCS decode inactive from SA data invalid	10		40	ns
t327	A data valid delay from XA data valid	4		27	ns
t328	SA data valid delay from XA data valid	6		34	ns
t329	MA data valid delay from XA data valid	9		49	ns
t330	LIOCS decode active from XA data valid	13		50	ns

CHIPS

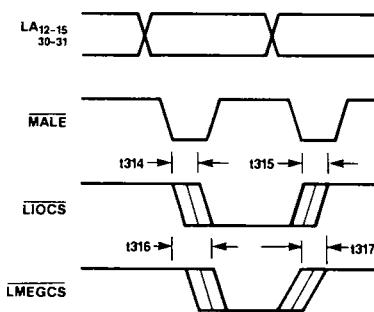
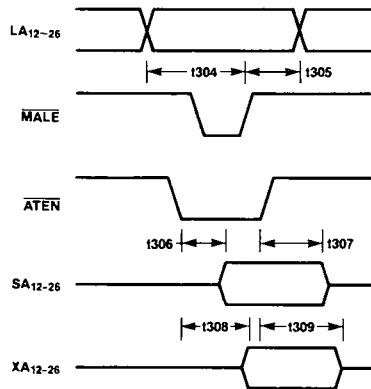
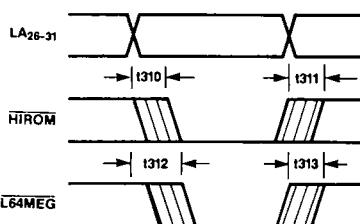
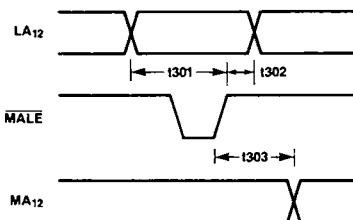
82A303 AC Characteristics (Continued)

($T_A = 0^\circ\text{C}$ to 60°C , $V_{CC} = 5V \pm 5\%$)

Sym	Description	Min.	Typ.	Max.	Units
t331	LIOCS decode inactive from XA data invalid	9	38	ns	
t332	L64MEG decode active from XA data valid	12	47	ns	
t333	L64MEG decode inactive from XA data invalid	9	35	ns	
t334	LMEGCS decode active from XA data valid	14	53	ns	
t335	LMEGCS decode inactive from XA data invalid	10	40	ns	
t336	SA valid delay from $\overline{\text{REF}}$ active	18	64	ns	
t337	SA tri-state delay from $\overline{\text{REF}}$ inactive	8	33	ns	

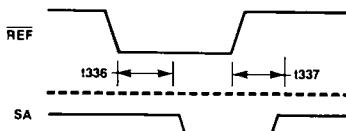
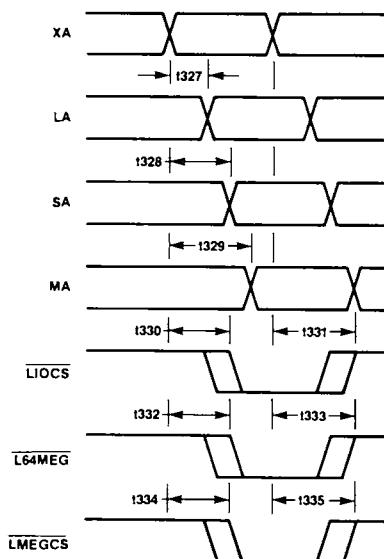
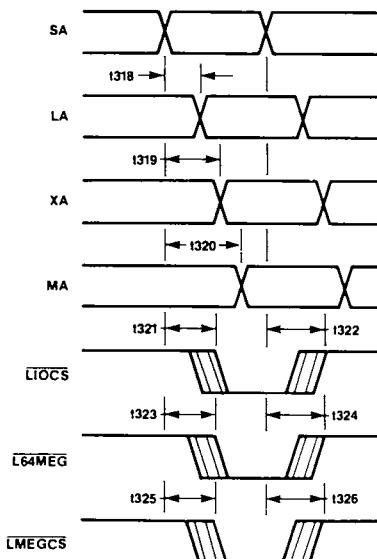
CHIPS

82A303 TIMING DIAGRAMS



CHIPS

82A304 TIMING DIAGRAMS





CHIPS

82A304 LOW ADDRESS BUFFERS

- Buffer for bits 11:00 of the Local, X and System address buses.
- Peripheral device decode
- Direct interface to AT Bus
- Refresh Address Generation
- Advanced Schottky TTL technology

FUNCTIONAL DESCRIPTIONS

Address Decode

The signals IO2XCS, 8042CS, PORTBCS, NMICS, 287CS, and AS provide the lower address decodes for the corresponding devices after being qualified by the LIOCS generated by the high address buffer decoder. The resulting decode is as defined by the IBM PC AT IO addresses and is as shown in table 4-1. For applications where these devices are required to be relocated, the EXDEC can be tied LOW to ignore the LIOCS qualification and the MA<11:10> address bits.

Signal	Addresses Decoded
IO2XCS	022H, 023H
8042CS	060H, 064H
PORTBCS	061H
NMICS	070H
287CS	0E0H to 0FFH

Table 4-1. Low Address Decode Definition

Address Bus Interfaces

The 82A304 interfaces between the bits 00 to 11 of A, SA, XA, and MA address buses. The buffers and multiplexers are controlled by the HLDA1, MASTER, REF, and ATEN to drive the signals from the source to the target buses as defined by table 4-2 for each signal when active. When REF is asserted, the refresh counter is gated to the SA bus as refresh row address and is incremented. When none of the listed signals are active, the default buffers configuration is that the A bus drives the MA bus for memory accesses by CPU.

The SA<11:00> are 24mA address buffers for direct interface to the AT bus.

Active	Source	Target
HLDA1	XA	SA, MA, A
MASTER	SA	XA, MA, A
REF	Counter	SA
ATEN	A<11:2>	SA<11:2>, XA<11:2>, MA<11:4>
		XA<1:0> SA<1:0>
default	A	MA

Table 4-2. Bus Control Definition

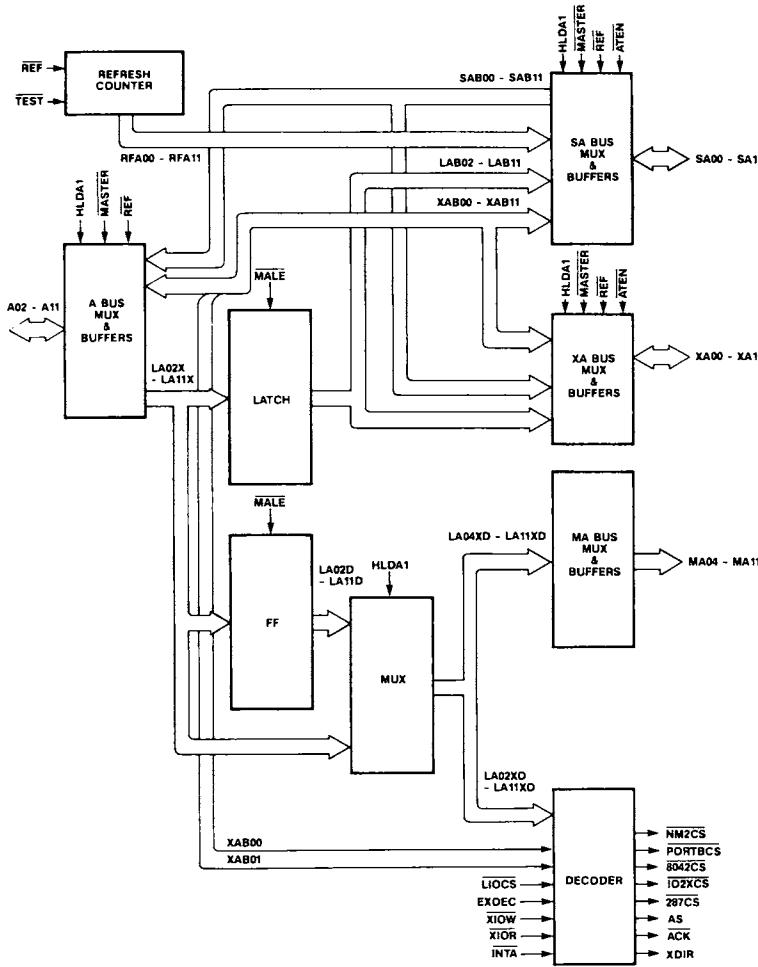


Figure 4-1. 82A304 Functional Block Diagram

CHIPS

82A304 Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V_{CC}	—	7.0	V
Input Voltage	V_I	-0.5	5.5	V
Output Voltage	V_O	-0.5	5.5	V
Operating Temperature	T_{op}	-25	85	C
Storage Temperature	T_{stg}	-40	125	C

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

82A304 Operating Conditions

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V_{CC}	4.75	5.25	V
Ambient Temperature	T_A	0	70	C

82A304 DC Characteristics

Parameter	Symbol	Min.	Max.	Units
Input Low Voltage	V_{IL}	—	0.8	V
Input High Voltage	V_{IH}	—	2.0	V
Output Low Voltage $I_{OL}=10\text{mA}$ (Note 1)	V_{OL1}	—	0.5	V
Output Low Voltage $I_{OL}=24\text{mA}$ (Note 2)	V_{OL2}	—	0.5	V
Output High Voltage $I_{OH}=3.3\text{mA}$ (Note 3)	V_{OH}	—	2.4	V
Input Low Current $V_I = 0.5\text{V}$, $V_{CC} = 5.25\text{V}$	I_{IL}	—	-200	μA
Input High Current $V_I = 2.4\text{V}$, $V_{CC} = 5.25\text{V}$	I_{IH}	—	20	μA
Input High Current $V_I = 5.5\text{V}$, $V_{CC} = 5.25\text{V}$	I_I	—	200	μA
Output Short Circuit Current $V_O=0\text{V}$	I_{OS}	-15	-100	mA
Input Clamp Voltage $I_I = -18\text{mA}$, $V_{CC} = 4.75\text{V}$	V_{IC}	—	-1.5	V
Power Supply Current	I_{CC}	140	230	mA
Output Hi-Z Leak Current 3-State Output Pins	I_{OZ1}	-100	100	μA
Output Hi-Z Leak Current Bidirectional Pins	I_{OZ2}	-300	120	μA

NOTES

1. All bus outputs other than SA<11:00>.
2. All SA<11:00> have $I_{OL} = 24\text{mA}$.
3. All outputs and bidirectional pins.

82A304 AC Characteristics
($T_A = 0^\circ C$ to $60^\circ C$, $V_{CC} = 5V \pm 5\%$)

Sym	Description	Min.	Typ.	Max.	Units
t401	A to MA input set-up time to <u>MALE1</u>			TBD	
t402	A to MA input hold time from <u>MALE1</u>			TBD	
t403	MA output valid delay from <u>MALE1</u>	5		34	ns
t404	A to SA, XA input set-up time to <u>MALE1</u>			TBD	
t405	A to SA, XA input hold time from <u>MALE1</u>			TBD	
t406	SA output valid delay from <u>ATEN</u> active	8		33	ns
t407	SA tri-state delay from <u>ATEN</u> inactive	6		28	ns
t408	XA output valid delay from <u>ATEN</u> active	10		41	ns
t409	XA tri-state delay from <u>ATEN</u> inactive	9		37	ns
t410	NMICS decode active from <u>MALE1</u>	11		44	ns
t411	NMICS decode inactive from <u>MALE1</u>	10		40	ns
t412	PORTBCS decode active from <u>MALE1</u>	11		44	ns
t413	PORTBCS decode inactive from <u>MALE1</u>	10		40	ns
t414	8042CS decode active from <u>MALE1</u>	11		44	ns
t415	8042CS decode inactive from <u>MALE1</u>	10		40	ns
t416	IO2XCS decode active from <u>MALE1</u>	11		44	ns
t417	IO2XCS decode inactive from <u>MALE1</u>	10		40	ns
t418	287CS decode active from <u>MALE1</u>	11		44	ns
t419	287CS decode inactive from <u>MALE1</u>	10		40	ns
t420	A data valid delay from SA data valid	4		26	ns
t421	XA data valid delay from SA data valid	3		35	ns
t422	MA data valid delay from SA data valid	9		49	ns
t423	NMICS decode active from SA data valid	14		58	ns
t424	NMICS decode inactive from SA data invalid	11		46	ns
t425	PORTBCS decode active from SA data valid	15		59	ns
t426	PORTBCS decode inactive from SA data invalid	11		46	ns
t427	8042CS decode active from SA data valid	15		59	ns
t428	8042CS decode inactive from SA data invalid	11		46	ns
t429	IO2XCS decode active from SA data valid	12		59	ns
t430	IO2XCS decode inactive from SA data invalid	12		46	ns

CHIPS

82A304 AC Characteristics (Continued)

($T_A = 0^\circ C$ to $60^\circ C$, $V_{CC} = 5V \pm 5\%$)

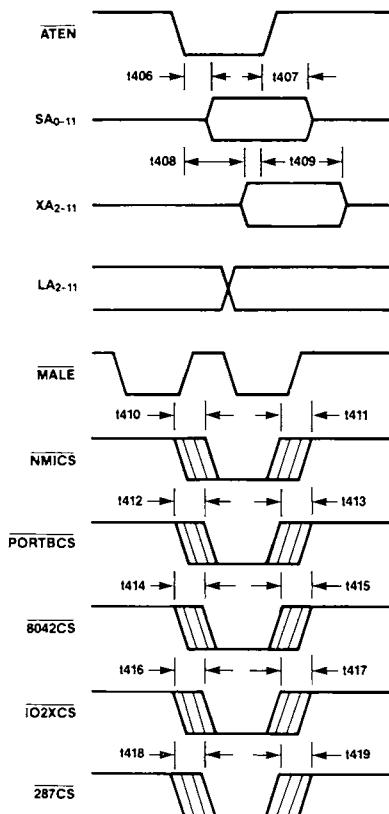
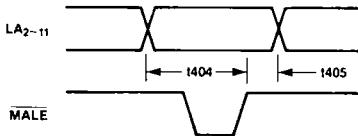
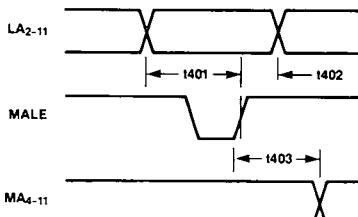
Sym	Description	Min.	Typ.	Max.	Units
t431	287CS decode active from SA data valid	16	59	ns	
t432	287CS decode inactive from SA data invalid	13	46	ns	
t433	XDIR decode active from SA data valid	15	59	ns	
t434	XDIR decode inactive from SA data invalid	15	60	ns	
t435	A data valid delay from XA data valid	4	26	ns	
t436	SA data valid delay from XA data valid	7	34	ns	
t437	MA data valid delay from XA data valid	9	49	ns	
t438	NMICS decode active from XA data valid	14	58	ns	
t439	NMICS decode inactive from XA data invalid	11	46	ns	
t440	PORTBCS decode active from XA data valid	15	59	ns	
t441	PORTBCS decode inactive from XA data invalid	11	46	ns	
t442	8042CS decode active from XA data valid	15	59	ns	
t443	8042CS decode inactive from XA data invalid	11	46	ns	
t444	IO2XCS decode active from XA data valid	12	59	ns	
t445	IO2XCS decode inactive from XA data invalid	12	46	ns	
t446	287CS decode active from XA data valid	16	59	ns	
t447	287CS decode inactive from XA data invalid	13	46	ns	
t448	XDIR decode active from XA data valid	15	55	ns	
t449	XDIR decode inactive from XA data invalid	15	55	ns	
t450	NMICS decode active from LIOCS active	7	31	ns	
t451	NMICS decode inactive from LIOCS inactive	5	24	ns	
t452	PORTBCS decode active from LIOCS active	7	31	ns	
t453	PORTBCS decode inactive from LIOCS inactive	5	24	ns	
t454	8042CS decode active from LIOCS active	7	30	ns	
t455	8042CS decode inactive from LIOCS inactive	5	24	ns	
t456	IO2XCS decode active from LIOCS active	7	30	ns	
t457	IO2XCS decode inactive from LIOCS inactive	5	24	ns	
t458	287CS decode active from LIOCS active	7	30	ns	
t459	287CS decode inactive from LIOCS inactive	5	24	ns	
t460	XDIR decode active from LIOCS active	8	32	ns	

82A304 AC Characteristics (Continued)
($T_A = 0^\circ\text{C}$ to 60°C , $V_{CC} = 5V \pm 5\%$)

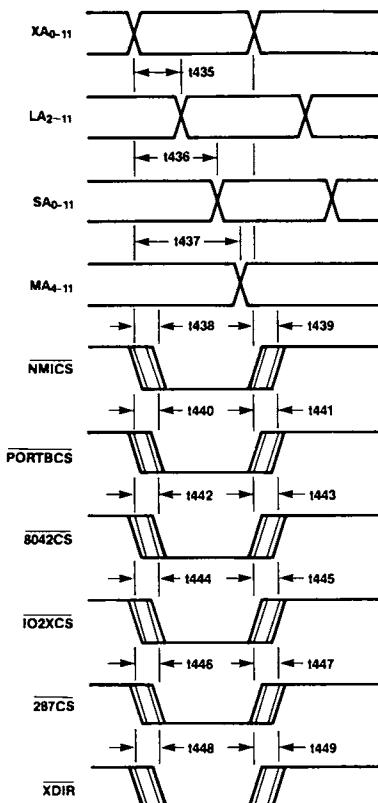
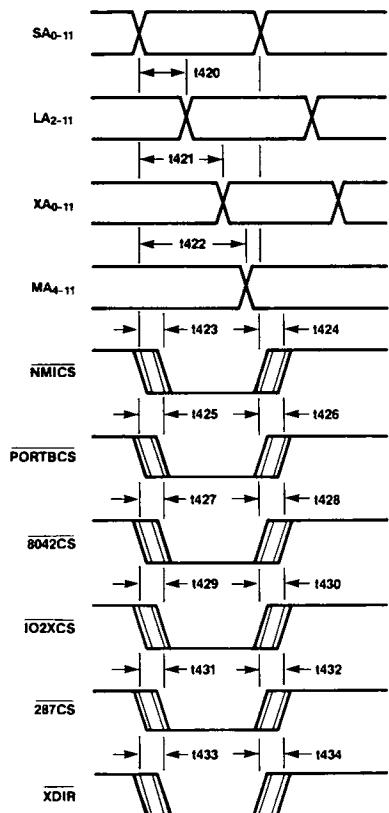
Sym.	Description	Min.	Typ.	Max.	Units
t461	XDIR decode inactive from LIOCS inactive	5	26	ns	
t462	XDIR decode active from INTA active	4	23	ns	
t463	XDIR decode inactive from INTA inactive	2	17	ns	
t464	XDIR decode active from XIOR active	6	27	ns	
t465	XDIR decode inactive from XIOR inactive	4	20	ns	
t466	ACK decode active from HLDA1 active	9	37	ns	
t467	ACK decode inactive from HLDA1 inactive	7	32	ns	
t468	ACK decode active from MASTER active	8	33	ns	
t469	ACK decode inactive from MASTER inactive	6	26	ns	
t470	SA data valid delay from REF active	18	64	ns	
t471	SA tri-state delay from REF inactive	8	33	ns	

CHIPS

82A304 TIMING DIAGRAMS

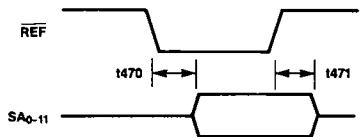
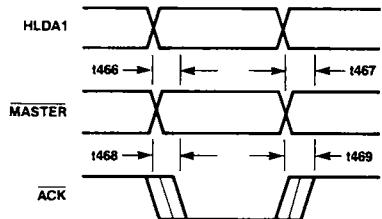
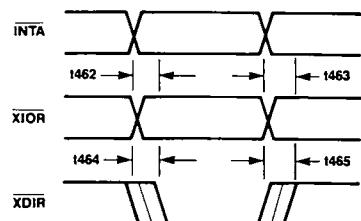
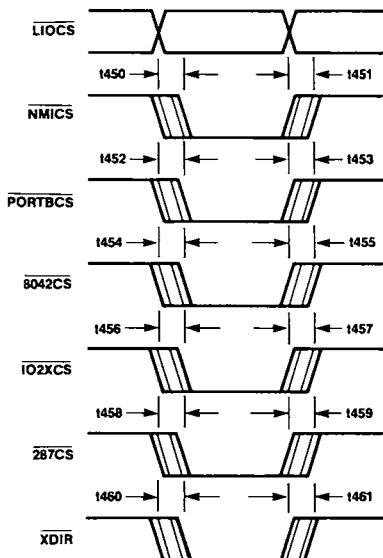


82A304 TIMING DIAGRAMS



CHIPS

82A304 TIMING DIAGRAMS





CHIPS

82A305 DATA BUFFER

- Nibble Slice of Memory and AT Data Bus Interface
- Data Size Conversion
- Advanced Schottky technology

FUNCTIONAL DESCRIPTION

The 82A305 interfaces between the Local, Memory and System AT IO channel data busses and provides data alignment and size conversion for AT IO channel operations. It is designed as a nibble slice to reduce pin count and simplify system design and two parts are used to interface all data buses.

Bus Controls

The 82A305 controls the bus buffers according to the signals HLDA1, ATEN, MDEN, LDEN, SDIR, MRD, and AC<3:0>. The first group of signals HLDA1, ATEN, MDEN, and LDEN determines which buses are connected, and the second group of signals SDIR, MRD, and AC<3:0> determines the direction of the buffers drivers. Table 5-1 shows the bus connections for different bus cycles.

All drivers are active for the active buses, and external bus controls are required if selective data bits need be controlled. For the DRAM interface, the LBE<3:0> must be used to ensure that only the valid data bytes are written into the DRAM's during a write cycle.

Data Conversion

The 82A305 provides the data bus connections so that data conversions are done correctly for CPU accesses to the AT bus. The action codes AC<3:0> are used to control how bus bits are connected between the IO channel SD bus and the CPU local bus D or the system memory MD bus. The action codes are provided by the 82C301 bus controller for CPU to AT bus access cycles and is qualified by the ACEN. The meaning of the action codes are:

AC<3:0>	FROM	TO
0	MD,D<15:0>	SD<15:0>
1	MD,D<15:8>	SD<15:8> SD<7:0>
2	MD,D<31:16>	SD<15:0>
3	MD,D<31:24>	SD<15:8> SD<7:0>
4	MD,D<31:0>	SD<31:0>
5	SD<7:0>	MD,D<7:0>
6	SD<7:0>	MD,D<15:8>
7	SD<7:0>	MD,D<23:16>
8	SD<7:0>	MD,D<31:24>
9	SD<15:0>	MD,D<15:0>
A	SD<15:0>	MD,D<31:16>
B	reserved	
C	SD<31:0>	MD,D<31:0>
D	reserved	
E	reserved	
F	reserved	

Table 5-2. Action Code Definition

Bus Cycles	From Bus	To Bus	Direction Control
HLDA1=0, ATEN=1	D MD	MD D	MRD=1 MRD=0
HLDA1=0, ATEN=0	D SD	SD D	SDIR=1 SDIR=0
HLDA1=1	SD	MD,D	SDIR=0, MRD=1
	MD	SD	SDIR=1, MRD=0
	D	SD	SDIR=1, MRD=1, LDEN=0

Table 5-1. Bus Control Definitions

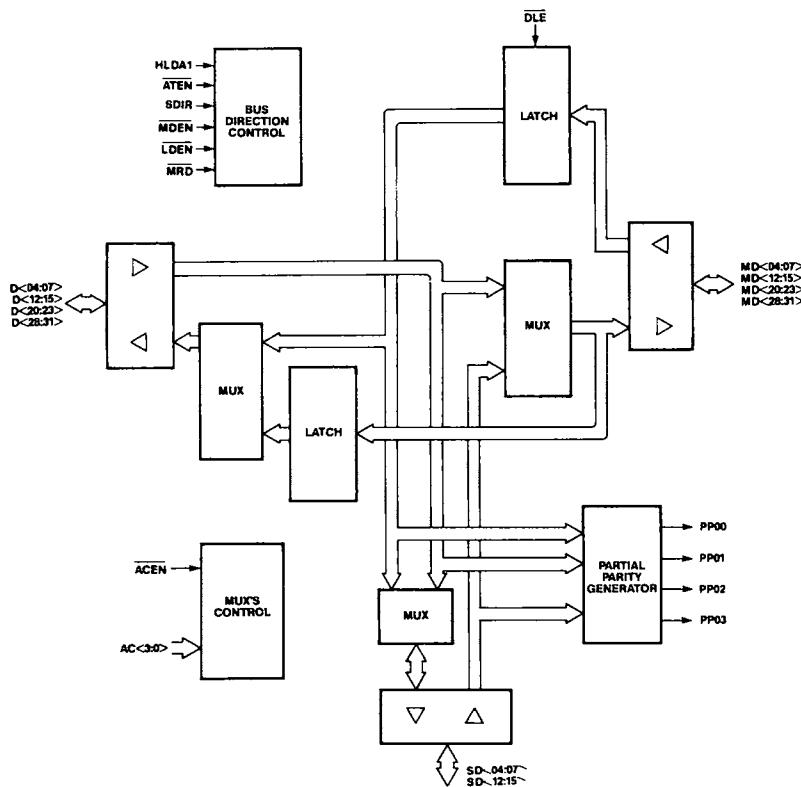


Figure 5-1. 82A305 Functional Block Diagram

CHIPS

82A305 Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V_{CC}	—	7.0	V
Input Voltage	V_I	-0.5	5.5	V
Output Voltage	V_O	-0.5	5.5	V
Operating Temperature	T_{op}	-25	85	C
Storage Temperature	T_{stg}	-40	125	C

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

82A305 Operating Conditions

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V_{CC}	4.75	5.25	V
Ambient Temperature	T_A	0	70	C

82A305 DC Characteristics

Parameter	Symbol	Min.	Max.	Units
Input Low Voltage	V_{IL}	—	0.8	V
Input High Voltage	V_{IH}	—	2.0	V
Output Low Voltage $I_{OL} = 10\text{mA}$ (Note 1)	V_{OL1}	—	0.5	V
Output Low Voltage $I_{OL} = 24\text{mA}$	V_{OL2}	—	0.5	V
Output High Voltage $I_{OH} = 3.3\text{mA}$ (Note 2)	V_{OH}	—	2.4	V
Input Low Current $V_I = 0.5\text{V}$, $V_{CC} = 5.25\text{V}$	I_{IL}	—	-200	μA
Input High Current $V_I = 2.4\text{V}$, $V_{CC} = 5.25\text{V}$	I_{IH}	—	20	μA
Input High Current $V_I = 5.5\text{V}$, $V_{CC} = 5.25\text{V}$	I_I	—	200	μA
Output Short Circuit Current $V_O = 0\text{V}$	I_{OS}	-15	-100	mA
Input Clamp Voltage $I_I = -18\text{mA}$, $V_{CC} = 4.75\text{V}$	V_{IC}	—	-1.5	V
Power Supply Current	I_{CC}	140	230	mA
Output HI-Z Leak Current 3-State Output Pins	I_{OZ1}	-100	100	μA
Output HI-Z Leak Current Bidirectional Pins	I_{OZ2}	-300	120	μA

NOTES

1. All bus outputs and $PP<3:0>$ have $I_{OL} = 10\text{mA}$.
2. All outputs and bidirectional pins.

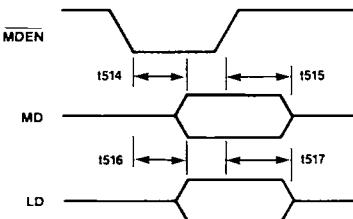
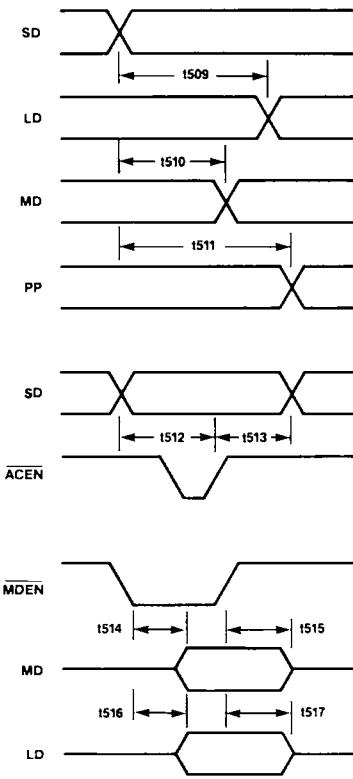
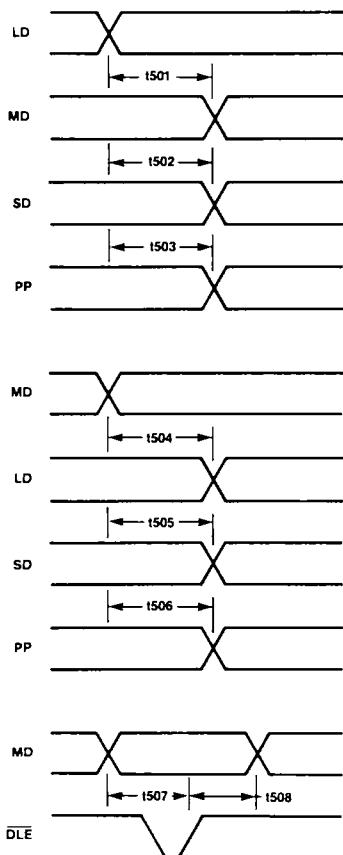
82A305 AC Characteristics

(TA = 0°C to 60°C, VCC = 5V ± 5%)

Sym	Description	Min.	Typ.	Max.	Units
t501	MD data valid delay from LD data valid	4	27	ns	
t502	SD data valid delay from LD data valid	4	27	ns	
t503	PP data valid delay from LD data valid	5	29	ns	
t504	LD data valid delay from MD data valid	4	28	ns	
t505	SD data valid delay from MD data valid	4	28	ns	
t506	PP data valid delay from MD data valid	5	30	ns	
t507	MD data set-up time to <u>DLE1</u>		TBD		
t508	MD data hold time from <u>DLE1</u>		TBD		
t509	LD data valid delay from SD data valid	8	38	ns	
t510	MD data valid delay from SD data valid	4	27	ns	
t511	PP data valid delay from SD data valid	5	33	ns	
t512	SD data set-up time to <u>ACEN1</u>		TBD		
t513	SD data hold time from <u>ACEN1</u>		TBD		
t514	MD data valid delay from <u>MDEN1</u>	7	29	ns	
t515	MD tri-state delay from <u>MDEN1</u>	5	23	ns	
t516	LD data valid delay from <u>MDEN1</u>	7	30	ns	
t517	LD tri-state delay from <u>MDEN1</u>	5	23	ns	

CHIPS

82A305 TIMING DIAGRAMS





CHIPS

82A306 CONTROL BUFFER

- 14.318MHz oscillator and divide by 12 counter
- Byte enable latch
- Parity Checking
- Direct interface to AT Bus
- Advanced Schottky TTL technology

FUNCTIONAL DESCRIPTION

14MHz Oscillator and Divider

The color reference oscillator is provided eliminating the 8224 normally used in AT compatible systems. A divide by 12 counter is also included to generate the OSC/12 (1.19MHz) signal used on the system board.

AF32 Generation

The AF32 is used in the CS 8230 system to indicate that the current bus cycle is a CPU local bus cycle.

Byte Enable Latch

The register that holds the byte enables valid during a memory cycle is located on the 82A306. An additional input FBE is provided to force all byte enables active during certain memory operations. A pullup resistor is provided on the FBE input for implementations not requiring this feature.

Parity Checking and Generation

The 82A306 provides the necessary exclusive OR'ing to generate full (byte) write and read parity from the partial parity bits PPH<3:0> and PPL<3:0> generated on the two (nibble wide) data buffers 82A305.

For a memory read access, read parity PPH<3:0> and PPL<3:0> are checked against the parity bits MP<3:0> read from memory. These parity bits are latched by CAS and PCHK so that they are kept valid during parity checking. The results of the byte-wise comparison are further gated by byte enables to ignore errors for bytes which are not valid. The ORed byte-wise parity error is then latched as the output LPAR if PEN input is asserted.

During a memory write access, write parity for each byte is generated from PPH<3:0> and PPL<3:0> and can be gated onto the memory parity bus MP<3:0> if enabled by WPE controlling the tri-state drivers. If an external parity generation circuit is used, an internal pullup resistor is provided for WPE to disable the write parity output buffers if left unconnected.

Bus Drivers

24mA drivers are provided for some of the control signals on the IO channel. These include SYSCLK, OSC, OSC/12, RDRV, SBHE, BALE, IOR, IOW, MEMR, MEMW, SMEMR, SMEMW and OUT1.

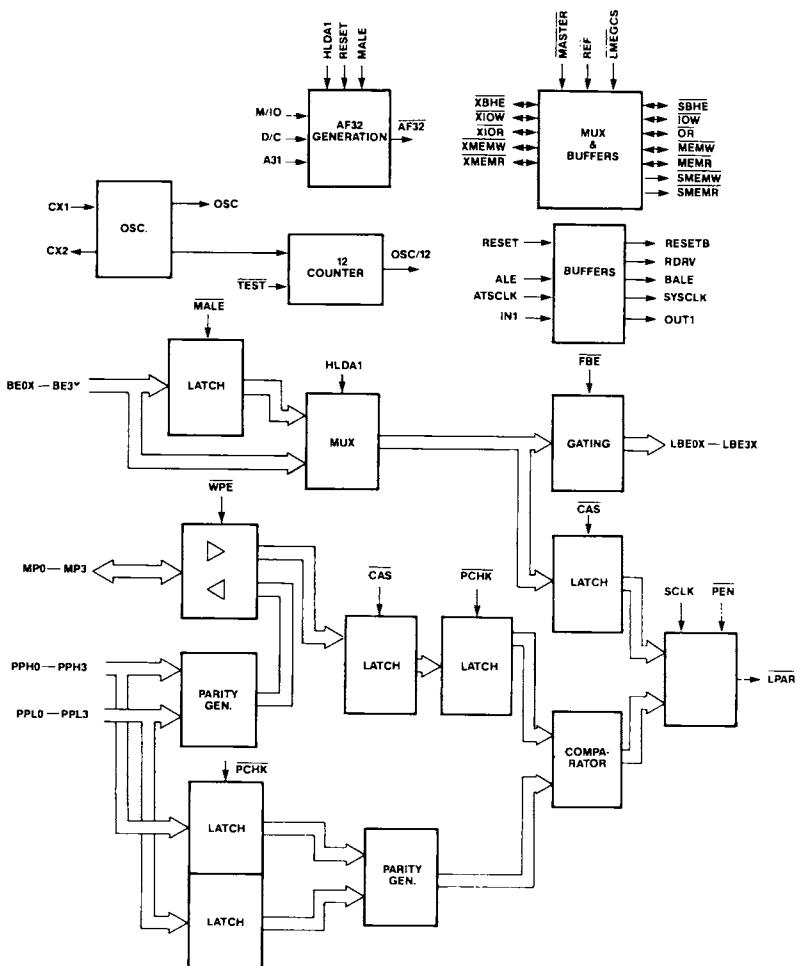


Figure 6-1. 82A306 Functional Block Diagram

CHIPS

82A306 Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V _{CC}	—	7.0	V
Input Voltage	V _I	-0.5	5.5	V
Output Voltage	V _O	-0.5	5.5	V
Operating Temperature	T _{op}	-25	85	C
Storage Temperature	T _{sig}	-40	125	C

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

82A306 Operating Conditions

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V _{CC}	4.75	5.25	V
Ambient Temperature	T _A	0	70	C

82A306 DC Characteristics

Parameter	Symbol	Min.	Max.	Units
Input Low Voltage	V _{IL}	0.8	V	
Input High Voltage	V _{IH}	2.0	V	
Output Low Voltage I _{OL} =10mA (Note 1)	V _{OL1}	0.5	V	
Output Low Voltage I _{OL} =24mA (Note 2)	V _{OL2}	0.5	V	
Output High Voltage I _{OH} =3.3mA (Note 3)	V _{OH}	2.4	V	
Input Low Current V _I = 0.5V, V _{CC} = 5.25V	I _{IL}	-200	μA	
Input High Current V _I = 2.4V, V _{CC} = 5.25V	I _{IH}	20	μA	
Input High Current V _I = 5.5V, V _{CC} = 5.25V	I _I	200	μA	
Output Short Circuit Current V _O =0V	I _{OS}	-15	-100	mA
Input Clamp Voltage I _I = -18mA, V _{CC} = 4.75V	V _{IC}	-1.5	V	
Power Supply Current	I _{CC}	140	230	mA
Output HI-Z Leak Current 3-State Output Pins	I _{OZ1}	-100	100	μA
Output HI-Z Leak Current Bidirectional Pins	I _{OZ2}	-300	120	μA

NOTES

1. MP<3:0>, XIOW, XIOR, XBHE, XMEMW, XMEMR, RESTEB, LBE<3:0> all have I_{OL} = 10mA.
2. SBHE, IOW, IOR, MEMW, MEMR, SMEMW, SMEMR, OSC, OSC/12, OUT1, SYSCLK, BALE, RDRV all have I_{OL} = 24mA.
3. All outputs and bidirectional pins.

82A306 AC Characteristics(T_A = 0°C to 60°C, V_{CC} = 5V ± 5%)

Sym	Description	Min.	Typ.	Max.	Units
t601	OSC _I delay from CX11	4	22	ns	
t602	OSC _I delay from CX11	5	26	ns	
t603	OSC/12 _I delay from CX11	9	35	ns	
t604	OSC/12 _I delay from CX11	9	37	ns	
t605	BE<3:0> set-up time to MALE _I		TBD		
t606	BE<3:0> hold time to MALE _I		TBD		
t607	LBE<3:0> valid delay from MALE _I	7	35	ns	
t608	LBE<3:0> valid delay from BE<3:0> valid	3	25	ns	
t609	LBE<3:0> LO delay from FBE _I	5	25	ns	
t610	LBE<3:0> de-asserted from FBE _I	3	19	ns	
t611	PPH<3:0>, PPL<3:0> set-up time to PCHK _I		TBD		
t612	PPH<3:0>, PPL<3:0> hold time to PCHK _I		TBD		
t613	MP<3:0> valid delay from corresponding PPH<3:0> and PPL<3:0>	2	21	ns	
t614	MP<3:0> set-up time from CAS _I		TBD		
t615	MP<3:0> hold time from CAS _I		TBD		
t616	LPARe _I delay from SCLK _I	4	23	ns	
t617	LPAE _I delay from SCLK _I	6	24	ns	
t618	LPARe _I delay from PEN _I	1	13	ns	
t619	LPAE _I delay from PEN _I	3	19	ns	
t620	MEMW _I (or MEMR _I) delay from XMEMW _I (or XMEMR _I)	3	19	ns	
t621	MEMW _I (or MEMR _I) delay from XMEMW _I (or XMEMR _I)	1	14	ns	
t622	XMEMW _I (or XMEMR _I) delay from MEMW _I (or MEMR _I)	4	21	ns	
t623	XMEMW _I (or XMEMR _I) delay from MEMW _I (or MEMR _I)	1	14	ns	
t624	SMEMW _I (or SMEMR _I) delay from XMEMW _I (or XMEMR _I)	5	23	ns	
t625	SMEMW _I (or SMEMR _I) LO to HI-Z transition delay from LMEGCS _I	4	23	ns	

CHIPS

82A306 AC Characteristics (Continued)

($T_A = 0^\circ\text{C}$ to 60°C , $V_{CC} = 5\text{V} \pm 5\%$)

Sym	Description	Min.	Typ.	Max.	Units
t626	SMEMW (or SMEMR) HI-Z to LO transition delay from LMEGCS!	6	28	ns	
t627	SMEMW (or SMEMR) LO to HI-Z transition delay from REF!	6	28	ns	
t628	SMEMW (or SMEMR) HI-Z to LO transition delay from REF!	8	32	ns	
t629	SMEMW! (or SMEMR!) delay from XMEMW! (or XMEMR!)	3	19	ns	
t630	SMEMW (or SMEMR) HI to HI-Z transition delay from LMEGCS!	4	23	ns	
t631	SMEMW (or SMEMR) HI-Z to HI transition delay from LMEGCS!	6	28	ns	
t632	SMEMW (or SMEMR) HI to HI-Z transition delay from REF!	6	28	ns	
t633	SMEMW (or SMEMR) HI-Z to HI transition delay from REF!	8	32	ns	
t634	SMEMW! (or SMEMR!) delay from MEMW! (or MEMR!)	5	23	ns	
t635	SMEMW! (or SMEMR!) delay from MEMW! (or MEMR!)	3	19	ns	
t636	IOW! (or IOR!) delay from XIOW! (or XIOR!)	3	18	ns	
t637	IOW! (or IOR!) delay from XIOW! (or XIOR!)	1	14	ns	
t638	XIOW! (or XIOR!) delay from IOW! (or IOR!)	4	21	ns	
t639	XIOW! (or XIOR!) delay from IOW! (or IOR!)	1	14	ns	
t640	SBHE! delay from XBHE!	3	18	ns	
t641	SBHE! delay from XBHE!	1	14	ns	
t642	XBHE! delay from SBHE!	4	21	ns	
t643	XBHE! delay from SBHE!	1	14	ns	
t644	RESETB! delay from RESET!	3	20	ns	
t645	RESETB! delay from RESET!	1	14	ns	
t646	RDRV! delay from RESET!	3	18	ns	

CHIPS

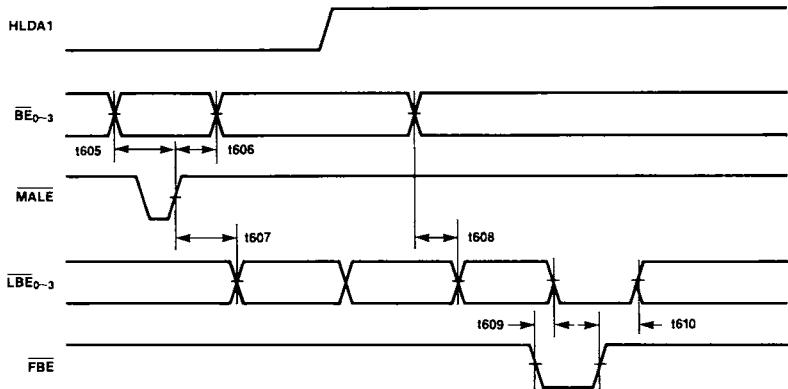
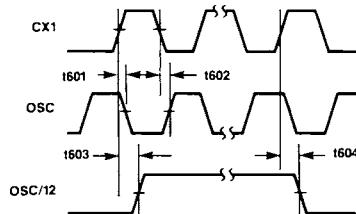
82A306 AC Characteristics (Continued)

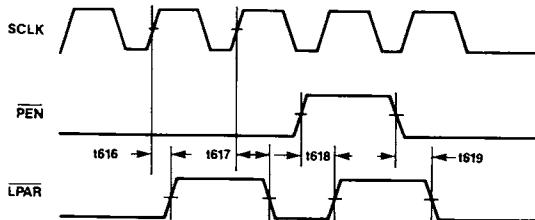
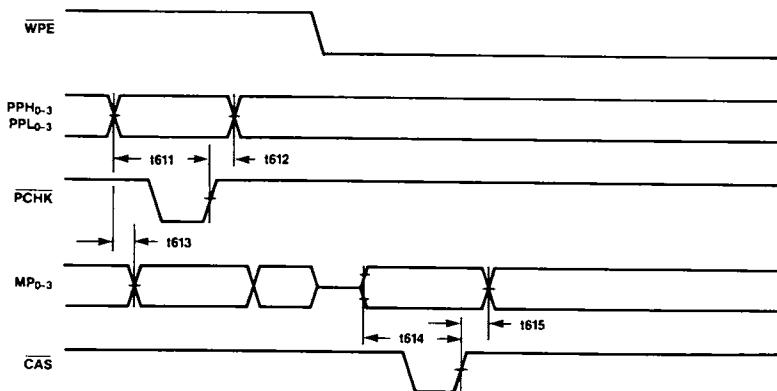
($T_A = 0^\circ\text{C}$ to 60°C , $V_{CC} = 5\text{V} \pm 5\%$)

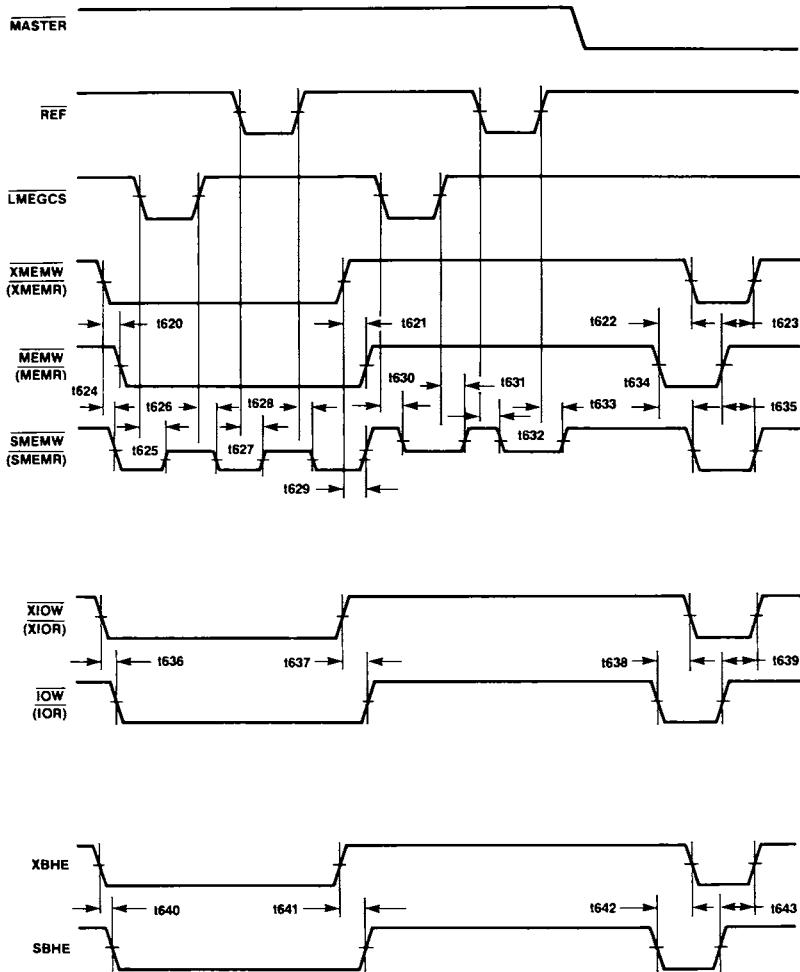
Sym	Description	Min.	Typ.	Max.	Units
t647	RDRV1 delay from RESET1	1	14	ns	
t648	BALE1 delay from ALT1 SYSCLK1 delay from ATSCLK1 OUT11 delay from IN11	2	17	ns	
t649	BALE1 delay from ALT1 SYSCLK1 delay from ATSCLK1 OUT11 delay from IN11	1	13	ns	
t650	M/IO, D/C, A31 set-up time to MALE1		TBD		
t651	M/IO, D/C, A31 hold time to MALE1		TBD		
t652	AF32 HI-Z to LO transition delay from MALE1	7	32	ns	
t653	AF32 LO to HI-Z transition delay from MALE1	6	29	ns	
t654	AF32 LO to HI-Z transition delay from RESET1	6	28	ns	
t655	AF32 LO to HI-Z transition delay from HLDA11	6	29	ns	

CHIPS

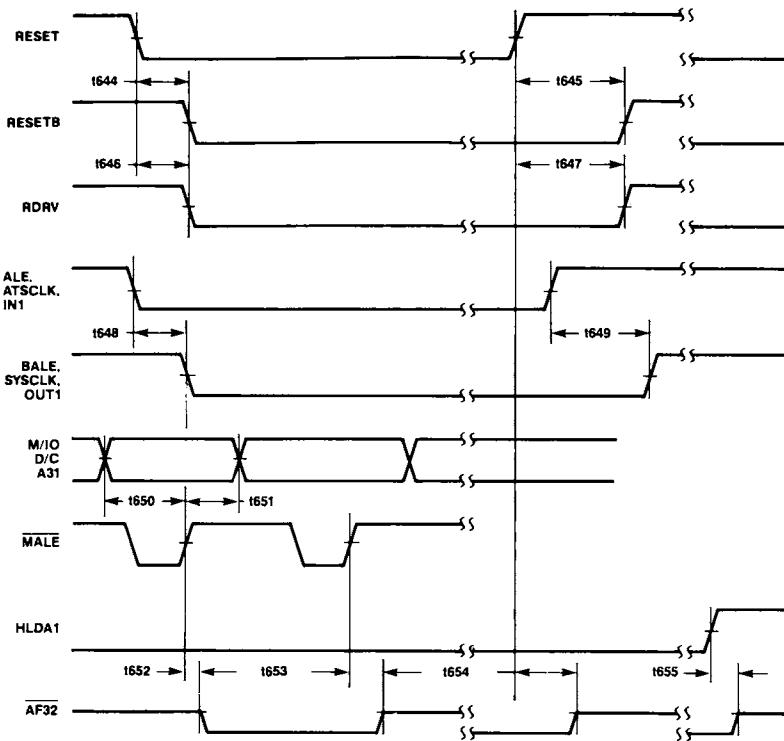
82A306 TIMING DIAGRAMS



82A306 TIMING DIAGRAMS

82A306 TIMING DIAGRAMS


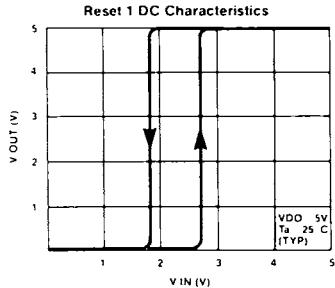
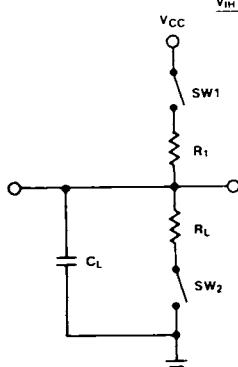
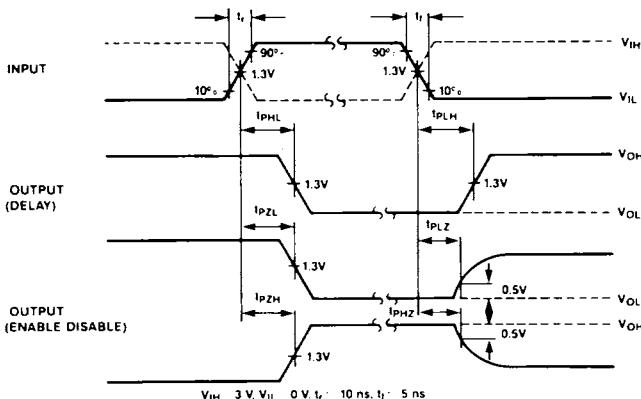
82A306 TIMING DIAGRAMS



CHIPS

Load Circuit Measurement Conditions

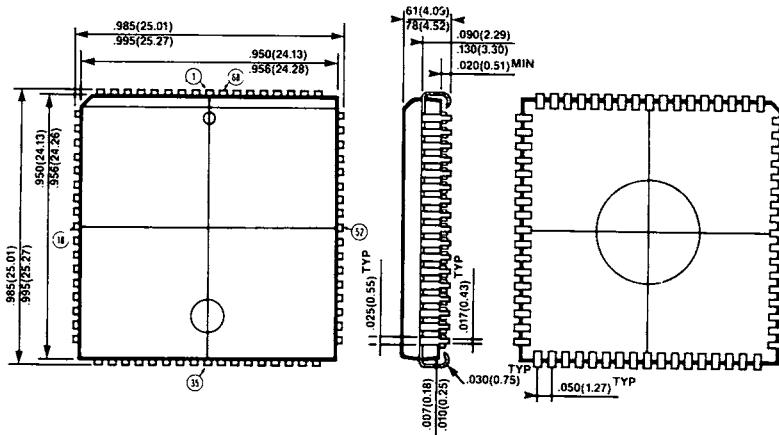
Parameter	Output Type	Symbol	C_L (pF)	R_1 (Ω)	R_L (Ω)	SW_1	SW_2
Propagation Delay Time	Totem pole 3-state	t_{PLH} t_{PHL}	50	—	1.0K	OFF	ON
Time	Bidirectional						
Propagation Delay Time	Open drain or Open Collector	t_{PLH} t_{PHL}	50	0.5K	—	ON	OFF
Disable Time	3-state Bidirectional	t_{PZL} t_{PZH}	5	0.5K	1.0K	ON OFF	ON
Enable Time	3-state Bidirectional	t_{PZL} t_{PZH}	50	0.5K	1.0K	ON OFF	ON



Load Circuit and AC Characteristics Measurement Waveform

CHIPS

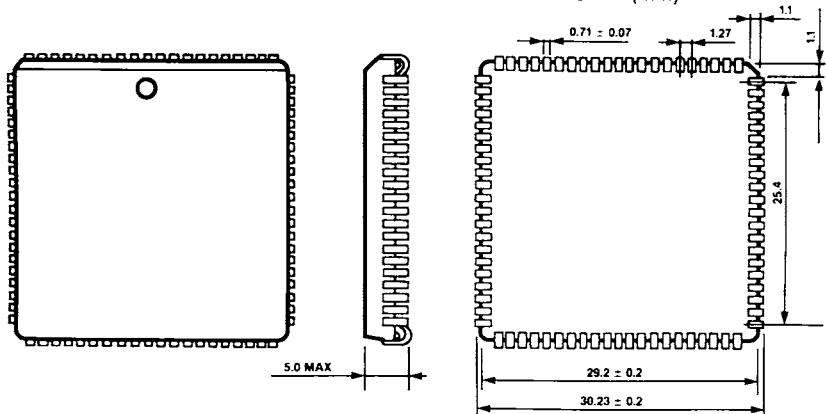
68-LEAD PLASTIC CHIP CARRIER



DIMENSIONS IN INCHES (MILLIMETERS) S = 3.6/1

CHIPS

84-PIN PLASTIC LEADED CHIP CARRIER



82A303 Absolute Maximum Ratings

Order Number	Package Type Note 1	Remarks
P82C301	PLCC-84	C (Note 2)
P82C302	PLCC-84	C
P82A303	PLCC-68	C
P82A304	PLCC-68	C
P82A305	PLCC-68	C
P82A306	PLCC-68	C
CS8230	—	Standard CHIPSet (Note 3)

NOTES

1. PLCC = Plastic Leaded Chip Carrier 84 Pins
2. C = Commercial Range, 0° to 70°C, $V_{DD} = 4.75$ to 5.25 V
3. CS8230 consists of P82C301, P82C302, P82A303, P82A304, 2 of P82A305, P82A306.





CHIPS



CHIPS

Chips and Technologies, Incorporated
521 Cottonwood Drive, Milpitas, CA 95035 408-434-0600 Telex 272929 CHIPS UR

IBM and AT are registered trademarks of International Business Machines.

iAPX 386 is a registered trademark of Intel Corporation.

CHIPSet is a trademark of Chips and Technologies.

CHIPS makes no warranty for the use of its products and bears no responsibility for any errors which may appear in this document.



8042/8742 UNIVERSAL PERIPHERAL INTERFACE 8-BIT MICROCOMPUTER

- 8042/8742: 12 MHz
- Pin, Software and Architecturally Compatible with 8041A/8741A
- 8-Bit CPU plus ROM, RAM, I/O, Timer and Clock in a Single Package
- 2048 x 8 ROM/EPROM, 128 x 8 RAM, 8-Bit Timer/Counter, 18 Programmable I/O Pins
- One 8-Bit Status and Two Data Registers for Asynchronous Slave-to-Master Interface
- DMA, Interrupt, or Polled Operation Supported
- Fully Compatible with all Intel and Most Other Microprocessor Families
- Interchangeable ROM and EPROM Versions
- Expandable I/O
- RAM Power-Down Capability
- Over 90 Instructions: 70% Single Byte
- Available in EXPRESS —Standard Temperature Range

The Intel 8042/8742 is a general-purpose Universal Peripheral Interface that allows the designer to grow his own customized solution for peripheral device control. It contains a low-cost microcomputer with 2K of program memory, 128 bytes of data memory, 8-bit CPU, I/O ports, 8-bit timer/counter, and clock generator in a single 40-pin package. Interface registers are included to enable the UPI device to function as a peripheral controller in the MCS-48™ MCS-51™, MCS-80™, MCS-85™, iAPX-88, iAPX-86 and other 8-, 16-bit systems.

The 8042/8742 is software, pin, and architecturally compatible with the 8041A, 8741A. The 8042/8742 doubles the on-chip memory space to allow for additional features and performance to be incorporated in upgraded 8041A/8741A designs. For new designs, the additional memory and performance of the 8042/8742 extends the UPI concept to more complex motor control tasks, 80-column printers and process control applications as examples.

To allow full user flexibility, the program memory is available as ROM in the 8042 version or as UV-erasable EPROM in the 8742 version. The 8742 and the 8042 are fully pin compatible for easy transition from prototype to production level designs.

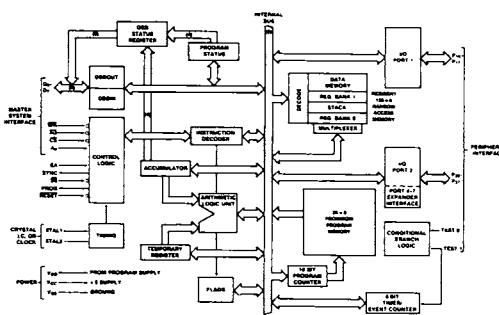


Figure 1. Block Diagram

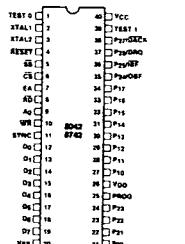


Figure 2. Pin Configuration



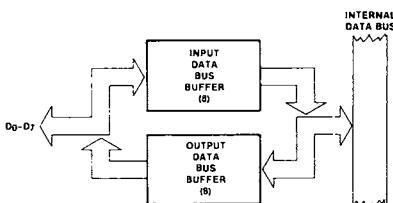
Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
TEST 0, TEST 1	1 39	I	Test Inputs: Input pins which can be directly tested using conditional branch instructions. Frequency Reference: TEST 1 (T_1) also functions as the event timer input (under software control). TEST 0 (T_0) is used during PROM programming and verification in the 8742.
XTAL 1, XTAL 2	2 3	I	Inputs: Inputs for a crystal, LC or an external timing signal to determine the internal oscillator frequency.
RESET	4	I	Reset: Input used to reset status flip-flops and to set the program counter to zero. RESET is also used during PROM programming and verification.
SS	5	I	Single Step: Single step input used in conjunction with the SYNC output to step the program through each instruction. (8742 only)
CS	6	I	Chip Select: Chip select input used to select one UPI microcomputer out of several connected to a common data bus.
EA	7	I	External Access: External access input which allows emulation, testing and PROM/ROM verification. This pin should be tied low if unused.
RD	8	I	Read: I/O read input which enables the master CPU to read data and status words from the OUTPUT DATA BUS BUFFER or status register.
A_0	9	I	Command/Data Select: Address input used by the master processor to indicate whether byte transfer is data ($A_0=0$, F1 is reset) or command ($A_0=1$, F1 is set).
WR	10	I	Write: I/O write input which enables the master CPU to write data and command words to the UPI INPUT DATA BUS BUFFER.

Symbol	Pin No.	Type	Name and Function
SYNC	11	O	Output Clock: Output signal which occurs once per UPI-42 instruction cycle. SYNC can be used as a strobe for external circuitry; it is also used to synchronize single step operation.
D_0-D_7 (BUS)	12-19	I/O	Data Bus: Three-state, bidirectional DATA BUS BUFFER lines used to interface the UPI-42 microcomputer to an 8-bit master system data bus.
$P_{10}-P_{17}$	27-34	I/O	Port 1: 8-bit, PORT 1 quasi-bidirectional I/O lines.
$P_{20}-P_{27}$	21-24 35-38	I/O	Port 2: 8-bit, PORT 2 quasi-bidirectional I/O lines. The lower 4 bits ($P_{20}-P_{23}$) interface directly to the 8243 I/O expander device and contain address and data information during PORT 4-7 access. The upper 4 bits ($P_{24}-P_{27}$) can be programmed to provide interrupt Request and DMA Handshake capability. Software control can configure P_{24} as Output Buffer Full (OBF) interrupt, P_{25} as Input Buffer Full (IBF) interrupt, P_{26} as DMA Request (DREQ), and P_{27} as DMA ACKnowledge (DACK).
PROG	25	I/O	Program: Multifunction pin used as the program pulse input during PROM programming. During I/O expander access the PROG pin acts as an address/data strobe to the 8243. This pin should be tied high if unused.
V_{CC}	40		Power: +5V main power supply pin.
V_{DD}	26		Power: +5V during normal operation. +21V during programming operation. Low power standby pin in ROM version.
V_{SS}	20		Ground: Circuit ground potential.

UPI-42 FEATURES

1. Two Data Bus Buffers, one for input and one for output. This allows a much cleaner Master/Slave protocol.



2. 8 Bits of Status

ST ₇	ST ₆	ST ₅	ST ₄	F ₁	F ₀	IBF	OBF
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀

ST₄-ST₇ are user definable status bits. These bits are defined by the "MOV STS, A" single byte, single cycle instruction. Bits 4-7 of the accumulator are moved to bits 4-7 of the status register. Bits 0-3 of the status register are not affected.

MOV STS, A Op Code: 90H							
1	0	0	1	0	0	0	0
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀

3. RD and WR are edge triggered. IBF, OBF, F₁, and INT change internally after the trailing edge of RD or WR.



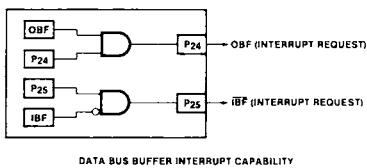
During the time that the host CPU is reading the status register, the 8042/8742 is prevented from updating this register or is 'locked out.'

4. P₂₄ and P₂₅ are port pins or Buffer Flag pins which can be used to interrupt a master processor. These pins default to port pins on Reset.

If the "EN FLAGS" instruction has been executed, P₂₄ becomes the OBF (Output Buffer Full) pin. A "1" written to P₂₄ enables the OBF pin (the pin outputs the OBF Status Bit). A "0" written to P₂₄ disables the OBF pin (the pin remains low). This pin can be used to indicate that valid data is available from the UPI-41A (in Output Data Bus Buffer).

If "EN FLAGS" has been executed, P₂₅ becomes the IBF (Input Buffer Full) pin. A "1" written to P₂₅ enables the IBF pin (the pin outputs the inverse of the IBF Status Bit). A "0" written to P₂₅ disables the IBF

pin (the pin remains low). This pin can be used to indicate that the UPI-42 is ready for data.



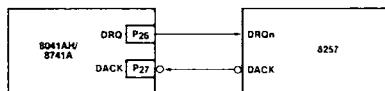
EN FLAGS Op Code: 0F5H

1	1	1	1	0	1	0	1
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀

5. P₂₆ and P₂₇ are port pins or DMA handshake pins for use with a DMA controller. These pins default to port pins on Reset.

If the "EN DMA" instruction has been executed, P₂₆ becomes the DRQ (DMA ReQuest) pin. A "1" written to P₂₆ causes a DMA request (DRQ is activated). DRQ is deactivated by DACK·RD, DACK·WR, or execution of the "EN DMA" instruction.

If "EN DMA" has been executed, P₂₇ becomes the DACK (DMA ACKnowledge) pin. This pin acts as a chip select input for the Data Bus Buffer registers during DMA transfers.



EN DMA Op Code: 0E5H

1	1	1	0	0	1	0	1
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀

6. The RESET input on the 8042/8742 includes a 2-stage synchronizer to support reliable reset operation for 12 MHz operation.

7. When EA is enabled on the 8042/8742, the program counter is placed on Port 1 and the lower three bits of Port 2 (MSB = P₂₂; LSB = P₁₀). On the 8042/8742 this information is multiplexed with PORT DATA (see port timing diagrams at end of this data sheet).

APPLICATIONS

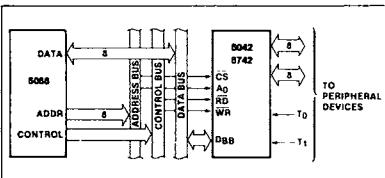


Figure 3. 8088-8042/8742 Interface

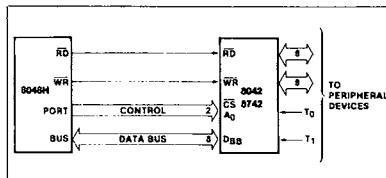


Figure 4. 8048H-8042/8742 Interface

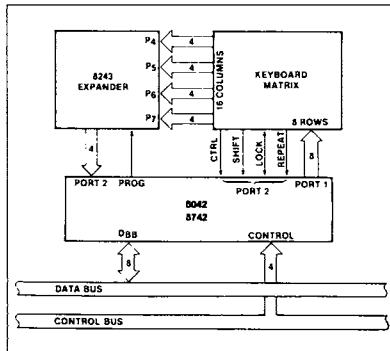


Figure 5. 8042/8742-8243 Keyboard Scanner

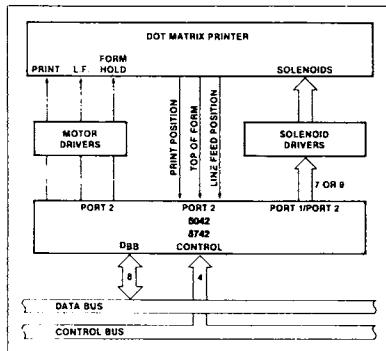


Figure 6. 8042/8742 80-Column Matrix Printer Interface

PROGRAMMING, VERIFYING, AND ERASING THE 8742 EPROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next word and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock Input
Reset	Initialization and Address Latching
Test 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input
P20-12	Data Output During Verify
Address Input	
VDD	Programming Power Supply
PROG	Program Pulse Input

WARNING

An attempt to program a missocketed 8742 will result in severe damage to the part. An indication of a properly socketed part is the appearance of the SYNC clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify sequence is:

1. $A_0 = 0V$, $CS = 5V$, $EA = 5V$, $RESET = 0V$, $TEST0 = 5V$, $V_{DD} = 5V$, clock applied or internal oscillator operating, BUS floating, PROG = 5V
2. Insert 8742 in programming socket
3. TEST 0 = Dv (select program mode)
4. EA = 18V (active program mode)*
5. Address applied to BUS and P20-22
6. RESET = 5v (latch address)
7. Data applied to BUS**
8. $V_{DD} = 21V$ (programming power)**
9. PROG = V_{CC} followed by one 50 ms pulse to 18V**
10. $V_{DD} = 5V$
11. TEST 0 = 5v (verify mode)



12. Read and verify data on BUS
13. TEST O = 0v
14. RESET = 0v and repeat from step 5
15. Programmer should be at conditions of step 1 when 8742 is removed from socket
*When verifying ROM, EA = 12V.
**Not used in verifying ROM procedure.

8742 Erasure Characteristics

The erasure characteristics of the 8742 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (\AA). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000 \AA range. Data show that constant exposure to room level fluorescent lighting could erase the typical 8742 in approximately 3 years while it would take ap-

proximately one week to cause erasure when exposed to direct sunlight. If the 8742 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 8742 window to prevent unintentional erasure.

The recommended erasure procedure for the 8742 is exposure to shortwave ultraviolet light which has a wavelength of 2537 \AA . The integrated dose (i.e., UV intensity \times exposure time) for erasure should be a minimum of 15 w-sec/cm². The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000 $\mu\text{W}/\text{cm}^2$ power rating. The 8742 should be placed within one inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias 0°C to 70°C
 Storage Temperature -65°C to +150°C
 Voltage on Any Pin With Respect
 to Ground -0.5V to +7V
 Power Dissipation 1.5 Watt

*NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS (TA = 0° to +70°C, VCC = VDD = +5V ± 10%)

Symbol	Parameter	8042/8742		Units	Notes
		Min.	Max.		
VIL	Input Low Voltage (Except XTAL1, XTAL2, RESET)	-0.5	0.8	V	
VIL1	Input Low Voltage (XTAL1, XTAL2, RESET)	-0.5	0.6	V	
VIH	Input High Voltage (Except XTAL1, XTAL2, RESET)	2.0	VCC	V	
VIH1	Input High Voltage (XTAL1, RESET)	3.5	VCC	V	
VIH2	Input High Voltage (XTAL2)	2.2	VCC	V	
VOL	Output Low Voltage (D0-D7)	0.45	V	IOL = 2.0 mA	
VOL1	Output Low Voltage (P10P17, P20P27, Sync)	0.45	V	IOL = 1.6 mA	
VOL2	Output Low Voltage (PROG)	0.45	V	IOL = 1.0 mA	
VOH	Output High Voltage (D0-D7)	2.4	V	IOH = -400 μA	
VOH1	Output High Voltage (All Other Outputs)	2.4		IOH = -50 μA	
IIL	Input Leakage Current (T0, T1, RD, WR, CS, A0, EA)	± 10	μA	VSS ≤ VIN ≤ VCC	
IOLF	Output Leakage Current (D0-D7, High Z State)	± 10	μA	VSS ≤ 0.45 ≤ VOUT ≤ VCC	
IIL1	Low Input Load Current (P10P17, P20P27)	0.3	mA	VIL = 0.8V	
IIL11	Low Input Load Current (RESET, SS)	0.2	mA	VIL = 0.8V	
IDD	VDD Supply Current	20	mA	Typical = 5 mA	
ICC + IDD	Total Supply Current	135	mA	Typical = 60 mA	
IIH	Input Leakage Current (P10-P17, P20-P27)	100	μA	VIN = VCC	
CIN	Input Capacitance		10	pF	
CO	I/O Capacitance		20	pF	

D.C. CHARACTERISTICS—PROGRAMMING (TA = 25°C ± 5°C, VCC = 5V ± 5%, VDD = 21V ± 0.5V)

Symbol	Parameter	Min.	Max.	Units	Test Conditions
VDOH	VDD Program Voltage High Level	20.5	21.5	V	
VDDL	VDD Voltage Low Level	4.75	5.25	V	
VPH	PROG Program Voltage High Level	17.5	18.5	V	
VPL	PROG Voltage Low Level	VCC-0.5	VCC	V	
VEAH	EA Program or Verify Voltage High Level	17.5	18.5	V	
VEAL	EA Voltage Low Level		5.25	V	
IDD	VDD High Voltage Supply Current	30.0	mA		
IPROG	PROG High Voltage Supply Current	1.0	mA		
IEA	EA High Voltage Supply Current	1.0	mA		



8042/8742

PRELIMINARY

A.C. CHARACTERISTICS ($T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $V_{SS} = 0\text{V}$, $V_{CC} = V_{DD} = +5\text{V} \pm 10\%$)**DBB READ**

Symbol	Parameter	8042		8742		Units
		Min.	Max.	Min.	Max.	
t_{AR}	CS, A_0 Setup to RD↓	0		0		ns
t_{RA}	CS, A_0 Hold After RD↑	0		0		ns
t_{RR}	RD Pulse Width	160		160		ns
t_{AD}	CS, A_0 to Data Out Delay		130		130	ns
t_{RD}	RD↓ to Data Out Delay		130		130	ns
t_{DF}	RD↑ to Data Float Delay		85		85	ns

DBB WRITE

Symbol	Parameter	Min.	Max.	Min.	Max.	Units
t_{AW}	CS, A_0 Setup to WR↓	0		0		ns
t_{WA}	CS, A_0 Hold After WR↑	0		0		ns
t_{WW}	WR Pulse Width	160		160		ns
t_{DW}	Data Setup to WR↑	130		130		ns
t_{WD}	Data Hold After WR↑	0		0		ns

CLOCK

Symbol	Parameter	8042		8742		Units
		Min.	Max.	Min.	Max.	
t_{CY}	Cycle Time	1.25	9.20	1.25	9.20	$\mu\text{s}^{(1)}$
t_{CYC}	Clock Period	83.3	613	83.3	613	ns
t_{PWH}	Clock High Time	33		38		ns
t_{PWL}	Clock Low Time	33		38		ns
t_R	Clock Rise Time		10		10	ns
t_F	Clock Fall Time		10		10	ns

NOTE:

- $t_{CY} = 15/t(\text{XTAL})$

A.C. CHARACTERISTICS ($T_A = 25^\circ C \pm 5^\circ C$, $V_{CC} = 5V \pm 5\%$, $V_{DD} = 21V \pm 0.5V$)
PROGRAMMING

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AW}	Address Setup Time to RESET \dagger	$4t_{CY}$			
t_{WA}	Address Hold Time After RESET \dagger	$4t_{CY}$			
t_{DW}	Data in Setup Time to PROG \dagger	$4t_{CY}$			
t_{WD}	Data in Hold Time After PROG \dagger	$4t_{CY}$			
t_{PH}	RESET Hold Time to Verify	$4t_{CY}$			
t_{VDDW}	V_{DD} Setup Time to PROG \dagger	0	1.0	ms	
t_{VDDH}	V_{DD} Hold Time After PROG \dagger	0	1.0	ms	
t_{PW}	Program Pulse Width	50	60	ms	
t_{TW}	Test 0 Setup Time for Program Mode	$4t_{CY}$			
t_{WT}	Test 0 Hold Time After Program Mode	$4t_{CY}$			
t_{DO}	Test 0 to Data Out Delay		$4t_{CY}$		
t_{WW}	RESET Pulse Width to Latch Address	$4t_{CY}$			
t_r, t_f	V_{DD} and PROG Rise and Fall Times	0.5	100	μs	
t_{CY}	CPU Operation Cycle Time	4.0		μs	
t_{RE}	RESET Setup Time Before EA \dagger	$4t_{CY}$			

NOTE:If TEST 0 is high, t_{DO} can be triggered by RESET \dagger .
A.C. CHARACTERISTICS DMA

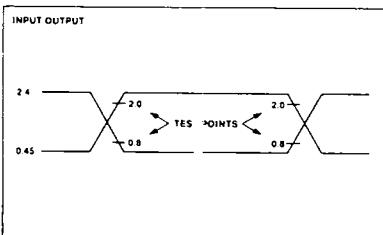
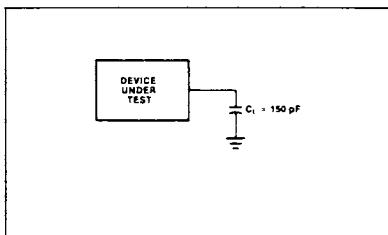
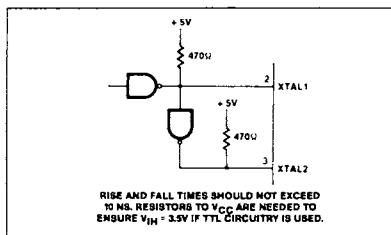
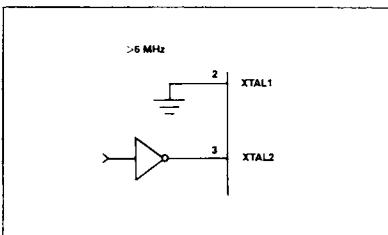
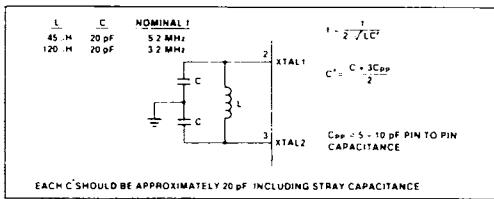
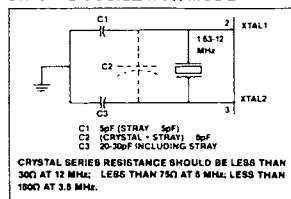
Symbol	Parameter	8042		8742		Units
		Min.	Max.	Min.	Max.	
t_{ACC}	DACK to WR or RD	0		0		ns
t_{CAC}	RD or WR to DACK	0		0		ns
t_{ACD}	DACK to Data Valid		130		130	ns
t_{CRO}	RD or WR to DRQ Cleared		110		130	ns ^[1]

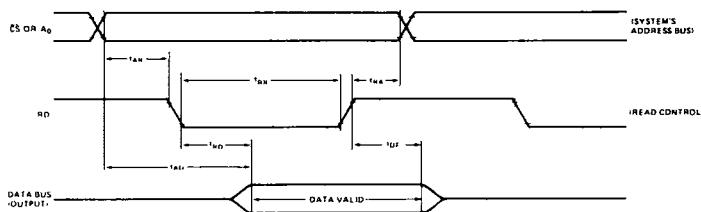
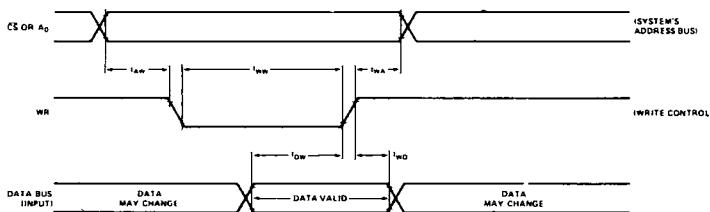
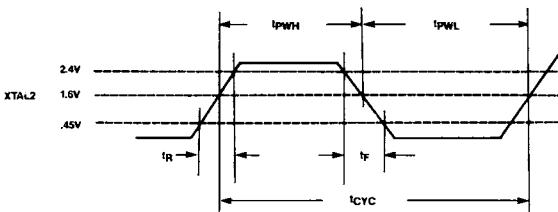
NOTE:1. $C_L = 150 \text{ pF}$.
A.C. CHARACTERISTICS PORT 2 ($T_A = 0^\circ C$ to $+ 70^\circ C$, $V_{CC} = + 5V \pm 10\%$)

Symbol	Parameter	$t(t_{CY})$	8042/8742 [3]		Units
			Min.	Max.	
t_{CP}	Port Control Setup Before Falling Edge of PROG	$1/15 t_{CY} - 28$	55		ns ^[1]
t_{PC}	Port Control Hold After Falling Edge of PROG	$1/10 t_{CY}$	125		ns ^[2]
t_{PR}	PROG to Time P2 Input Must Be Valid	$(8/15 t_{CY} - 16)$		650	ns ^[1]
t_{PF}	Input Data Hold Time		0	150	ns ^[2]
t_{DP}	Output Data Setup Time	$2/10 t_{CY}$	250		ns ^[1]
t_{PD}	Output Data Hold Time	$1/10 t_{CY} - 80$	45		ns ^[2]
t_{PP}	PROG Pulse Width	$6/10 t_{CY}$	750		ns

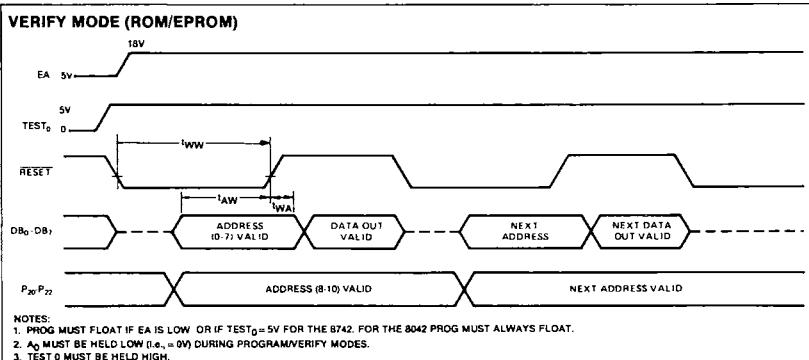
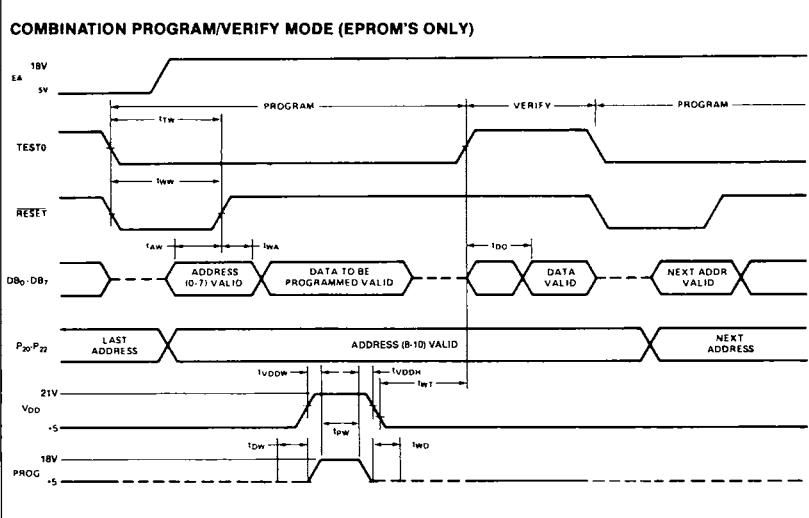
NOTES:

1. $C_L = 80 \text{ pF}$.
2. $C_L = 20 \text{ pF}$.
3. $t_{CY} = 1.25 \mu s$.

A.C. TESTING INPUT, OUTPUT WAVEFORM**A.C. TESTING LOAD CIRCUIT****DRIVING FROM EXTERNAL SOURCE-TWO OPTIONS****LC OSCILLATOR MODE****CRYSTAL OSCILLATOR MODE**

WAVEFORMS**READ OPERATION—DATA BUS BUFFER REGISTER****WRITE OPERATION—DATA BUS BUFFER REGISTER****CLOCK TIMING**

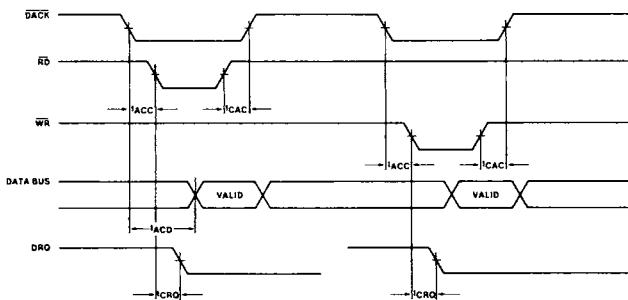
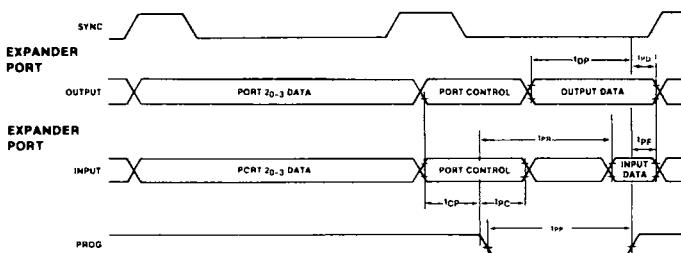
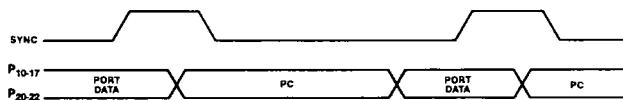
WAVEFORMS (Continued)



The 8742 EPROM can be programmed by the following Intel products:

1. Universal PROM Programmer (UPP 103) peripheral of the Intellic® Development System with a UPP-549 Personality Card.

2. iUP-200/iUP-201 PROM Programmer with the iUP-87/44 Personality Module.

WAVEFORMS (Continued)**DMA****PORT 2****PORT TIMING DURING EA**

ON THE RISING EDGE OF SYNC AND EA IS ENABLED, PORT DATA IS VALID AND CAN BE STROBED. ON THE TRAILING EDGE OF SYNC THE PROGRAM COUNTER CONTENTS ARE AVAILABLE.



Table 2. UPI™ Instruction Set

Mnemonic	Description	Bytes	Cycles
ACCUMULATOR			
ADD A, Rr	Add register to A	1	1
ADD A, @Rr	Add data memory to A	1	1
ADD A, #data	Add immediate to A	2	2
ADDC A, Rr	Add register to A with carry	1	1
ADDC A, @Rr	Add data memory to A with carry	1	1
ADDC A, #data	Add immediate to A with carry	2	2
ANL A, Rr	AND register to A	1	1
ANL A, @Rr	AND data memory to A	1	1
ANL A, #data	AND immediate to A	2	2
ORL A, Rr	OR register to A	1	1
ORL A, @Rr	OR data memory to A	1	1
ORL A, #data	OR immediate to A	2	2
XRL A, Rr	Exclusive OR register to A	1	1
XRL A, @Rr	Exclusive OR data memory to A	1	1
XRL A, #data	Exclusive OR immediate to A	2	2
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal Adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
INPUT/OUTPUT			
IN A, Pp	Input port to A	1	2
OUTL Pp, A	Output A to port	1	2
ANL Pp, #data	AND immediate to port	2	2
ORL Pp, #data	OR immediate to port	2	2
IN A, DBB	Input DBB to A, clear IBF	1	1
OUT DBB, A	Output A to DBB, set OBF	1	1
MOV STS, A	A ₄ -A ₇ to Bits 4-7 of Status	1	1
MOVD A, Pp	Input Expander port to A	1	2
MOVD Pp, A	Output A to Expander port	1	2
ANLD Pp, A	AND A to Expander port	1	2
ORLD Pp, A	OR A to Expander port	1	2
DATA MOVES			
MOV A, Rr	Move register to A	1	1
MOV A, @Rr	Move data memory to A	1	1
MOV A, #data	Move immediate TO A	2	2
MOV Rr, A	Move A to register	1	1
MOV @Rr, A	Move A to data memory	1	1
MOV Rr, #data	Move immediate to register	2	2
MOV @Rr, #data	Move immediate to data memory	2	2
MOV A, PSW	Move PSW to A	1	1
MOV PSW, A	Move A to PSW	1	1
XCH A, Rr	Exchange A and register	1	1
XCH A, @Rr	Exchange A and data memory	1	1
XCHD A, @Rr	Exchange digit of A and register	1	1
MOV P A, @A	Move to A from current page	1	2
MOV P3 A, @A	Move to A from page 3	1	2
TIMER/COUNTER			
MOV A, T	Read Timer/Counter	1	1
MOV T, A	Load Timer/Counter	1	1
STRT T	Start Timer	1	1
STRT CNT	start Counter	1	1
STOP TCNT	Stop Timer/Counter	1	1
EN TCNTI	Enable Timer/Counter Interrupt	1	1
DIS TCNTI	Disable Timer/Counter Interrupt	1	1
CONTROL			
EN DMA	Enable DMA Handshake Lines	1	1
EN I	Enable IBF Interrupt	1	1
DIS I	Disable IBF Interrupt	1	1
EN FLAGS	Enable Master Interrupts	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1
NOP	No Operation	1	1
REGISTERS			
INC Rr	Increment register	1	1
INC @Rr	Increment data memory	1	1
DEC Rr	Decrement register	1	1
SUBROUTINE			
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2

Table 2. UPI™ Instruction Set (Continued)

Mnemonic	Description	Bytes	Cycles
FLAGS			
CLR C	Clear Carry	1	1
CPL C	Complement Carry	1	1
CLR F0	Clear Flag 0	1	1
CPL F0	Complement Flag 0	1	1
CLR F1	Clear F1 Flag	1	1
CPL F1	Complement F1 Flag	1	1
BRANCH			
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ Rr, addr	Decrement register and jump	2	2
JC addr	Jump on Carry=1	2	2
JNC addr	Jump on Carry=0	2	2
JZ addr	Jump on A Zero	2	2
JNZ addr	Jump on A not Zero	2	2
JT0 addr	Jump on T0=1	2	2
JNT0 addr	Jump on T0=0	2	2
JT1 addr	Jump on T1=1	2	2
JNT1 addr	Jump on T1=0	2	2
JF0 addr	Jump on F0 Flag=1	2	2
JF1 addr	Jump on F1 Flag=1	2	2
JTF addr	Jump on Timer Flag =1, Clear Flag	2	2
JNIBF addr	Jump on IBF Flag =0	2	2
JOBF addr	Jump on OBF Flag =1	2	2
JBb addr	Jump on Accumula- tor Bit	2	2



80287 80-BIT HMOS NUMERIC PROCESSOR EXTENSION (80287-3, 80287-6, 80287-8, 80287-10)

- High Performance 80-Bit Internal Architecture
- Implements Proposed IEEE Floating Point Standard 754
- Expands 80286 Data types to Include 32-, 64-, 80-Bit Floating Point, 32-, 64-Bit Integers and 18-Digit BCD Operands
- Object Code Compatible with 8087
- Built-in Exception Handling
- Operates in Both Real and Protected Mode 80286 Systems
- 8x80-Bit, Individually Addressable, Numeric Register Stack
- Protected Mode Operation Completely Conforms to the 80286 Memory Management and Protection Mechanisms
- Directly Extends 80286 Instruction Set to Trigonometric, Logarithmic, Exponential and Arithmetic Instructions for All Data types
- Operates with 80386 CPU without Software Modification
- Available in EXPRESS—Standard Temperature Range
- Available in 40 pin-CERDIP package (see Packaging Spec: Order #231369)

The Intel 80287 is a high performance numerics processor extension that extends the 80286 architecture with floating point, extended integer and BCD data types. The 80286/80287 computing system fully conforms to the proposed IEEE Floating Point Standard. Using a numerics oriented architecture, the 80287 adds over fifty mnemonics to the 80286/80287 instruction set, making the 80286/80287 a complete solution for high performance numeric processing. The 80287 is implemented in N-channel, depletion load, silicon gate technology (HMOS) and packaged in a 40-pin cerdip package. The 80286/80287 is object code compatible with the 8086/8087 and 8088/8089.

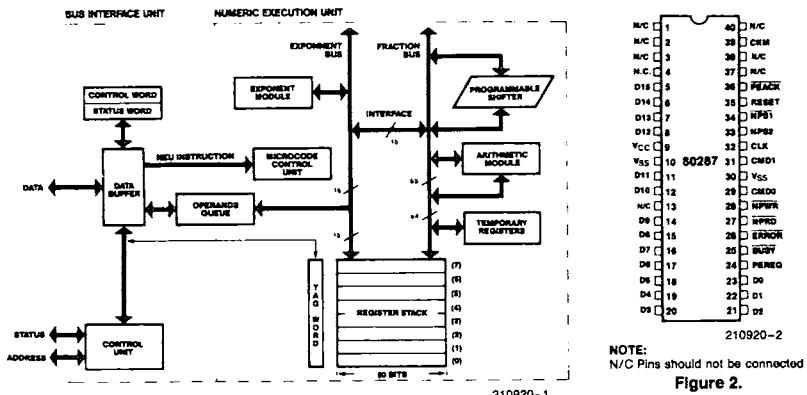


Figure 1. 80287 Block Diagram



80287

Table 1. 80287 Pin Description

Symbol	Type	Name and Function
CLK	I	CLOCK INPUT: this clock provides the basic timing for internal 80287 operations. Special MOS level inputs are required. The 82284 or 8284A CLK outputs are compatible to this input.
CKM	I	CLOCK MODE SIGNAL: indicates whether CLK input is to be divided by 3 or used directly. A HIGH input will cause CLK to be used directly. This input must be connected to V _{CC} or V _{SS} as appropriate. This input must be either HIGH or LOW 20 CLK cycles before RESET goes LOW.
RESET	I	SYSTEM RESET: causes the 80287 to immediately terminate its present activity and enter a dormant state. RESET is required to be HIGH for more than 4 80287 CLK cycles. For proper initialization the HIGH-LOW transition must occur no sooner than 50 µs after V _{CC} and CLK meet their D.C. and A.C. specifications.
D15-D0	I/O	DATA: 1-bit bidirectional data bus. Inputs to these pins may be applied asynchronous to the 80287 clock.
BUSY	O	BUSY STATUS: asserted by the 80287 to indicate that it is currently executing a command.
ERROR	O	ERROR STATUS: reflects the ES bit of the status word. This signal indicates that an unmasked error condition exists.
PEREQ	O	PROCESSOR EXTENSION DATA CHANNEL OPERAND TRANSFER REQUEST: a HIGH on this output indicates that the 80287 is ready to transfer data. PEREQ will be disabled upon assertion of PEACK or upon actual data transfer, whichever occurs first, if no more transfers are required.
PEACK	I	PROCESSOR EXTENSION DATA CHANNEL OPERAND TRANSFER ACKNOWLEDGE: acknowledges that the request signal (PEREQ) has been recognized. Will cause the request (PEREQ) to be withdrawn in case there are no more transfers required. PEACK may be asynchronous to the 80287 clock.
NPRD	I	NUMERIC PROCESSOR READ: Enables transfer of data from the 80287. This input may be asynchronous to the 80287 clock.
NPWR	I	NUMERIC PROCESSOR READ: Enables transfer of data from the 80287. This input may be asynchronous to the 80287 clock.
NPS1, NPS2	I	NUMERIC PROCESSOR SELECTS: indicate the CPU is performing an ESCAPE instruction. Concurrent assertion of these signals (i.e., NPST is LOW and NPS2 is HIGH) enables the 80287 to perform floating point instructions. No data transfers involving the 80287 will occur unless the device is selected via these lines. These inputs may be asynchronous to the 80287 clock.
CMD1, CMD0	I	COMMAND LINES: These, along with select inputs, allow the CPU to direct the operation of the 80287. These inputs may be asynchronous to the 80287 clock.

Table 1. 80187 Pin Description (Continued)

Symbols	Type	Name and Function
Vss	I	System ground, both pins must be connected to ground.
Vcc	I	+ 5V supply

FUNCTIONAL DESCRIPTION

The 80287 Numeric Processor Extension (NPX) provides arithmetic instructions for a variety of numeric data types in 80286/80287 systems. It also executes numerous built-in transcendental functions (e.g., tangent and log functions). The 80287 executes instructions in parallel with an 80286. It effec-

tively extends the register and instruction set of an 80286 system for existing 80286 data types and adds several new data types as well. Figure 3 presents the program visible register model of the 80286/80287. Essentially, the 80287 can be treated as an additional resource or an extension to the 80286 that can be used as a single unified system, the 80286/80287.

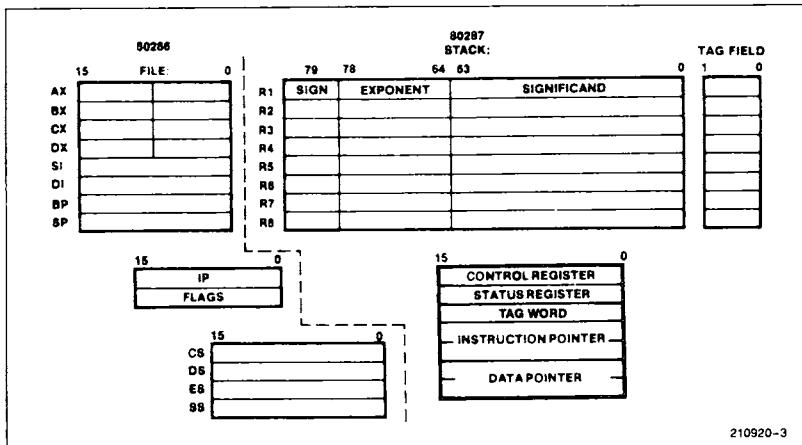


Figure 3. 80286/80287 Architecture

The 80287 has two operating modes similar to the two modes of the 80286. When reset, 80287 is in the real address mode. It can be placed in the protected virtual address mode by executing the SETPM ESC instruction. The 80287 cannot be switched back to the real address mode except by reset. In the real address mode, the 80286/80287 is completely software compatible with 8086/8087 and 8088/8087.

Once in protected mode, all references to memory for numerics data or status information, obey the 80286 memory management and protection rules giving a fully protected extension of the 80286 CPU. In the protected mode, 80286/80287 numerics software is also completely compatible with 8086/8087 and 8088/8087.



SYSTEM CONFIGURATION WITH 80286

As a processor extension to an 80286, the 80287 can be connected to the CPU as shown in Figure 4A. The data channel control signals (PEREQ, PEACK), the BUSY signal and the NPRD, NPWR signals, allow the NPX to receive instructions and data from the CPU. When in the protected mode, all information received by the NPX is validated by the 80286 memory management and protection unit. Once started, the 80287 can process in parallel with and independent of the host CPU. When the NPX detects an error or exception, it will indicate this to the CPU by asserting the ERROR signal.

The NPX uses the processor extension request and acknowledgement pins of the 80286 CPU to implement data transfers with memory under the protection model of the CPU. The full virtual and physical address space of the 80286 is available. Data for the 80287 in memory is addressed and represented in the same manner as for an 8087.

The 80287 can operate either directly from the CPU clock or with a dedicated clock. For operation with the CPU clock ($CKM = 0$), the 80287 works at one-third the frequency of the system clock (i.e., for an 8 MHz 80286, the 16 MHz system clock is divided down to 5.3 MHz). The 80287 provides a capability to internally divide the CPU clock by three to produce the required internal clock (33% duty cycle). To use a higher performance 80287 (8 MHz), an 8284A clock driver and appropriate crystal may be used to directly drive the 80287 with a 1/2 duty cycle clock on the CLK input ($CKM = 1$). The following table describes the relationship between the clock speed and the 287 operating speed as a function of the CKM state.

287 Speed	CLK Speed	
	CKM = 0	CKM = 1
5 MHz	12 MHz	5 MHz
6 MHz	16 MHz	6 MHz
8 MHz	20 MHz	8 MHz
10 MHz	25 MHz	10 MHz

SYSTEM CONFIGURATION WITH 80386

The 80287 can also be connected as a processor extension to the 80386 CPU as shown in Figure 4b. All software written for 8086/8087 and 80286/80287 is object code compatible with 80386/80287 and can benefit from the increased speed of the 80386 CPU.

Note that the PEACK input pin is pulled high. This is because the 80287 is not required to keep track of the number of words transferred during an operand transfer when it is connected to the 80386 CPU. Unlike the 80286 CPU, the 80386 CPU knows the exact length of the operand being transferred to/from the 80287. After an ESC instruction has been sent to the 80287, the 80386 processor extension data channel will initiate the data transfer as soon as it receives the PEREQ signal from the 80287. The transfer is automatically terminated by the 80386 CPU as soon as all the words of the operand have been transferred.

Because of the very high speed local bus of the 80386 CPU, the 80287 cannot reside directly on the CPU local bus. A local bus controller logic is used to generate the necessary read and write cycle timings as well as the chip select timings for the 80287. The 80386 CPU uses I/O addresses 800000F8 through 800000FF to communicate with the 80287. This is beyond the normal I/O address space of the CPU and makes it easier to generate the chip select signals using A31 and M/I/O. It may also be noted that the 80386 CPU automatically generates 16-bit bus cycles whenever it communicates with the 80287.

HARDWARE INTERFACE

Communication of instructions and data operands between the 80286 and 80287 is handled by the CMD0, CMD1, NPS1, NPS2, NPRD, and NPWR signals. I/O port addresses 00F8H, 00FAH, and 00FCH are used by the 80286 for this communication. When any of these addresses are used, the NPS1 input must be LOW and NPS2 input HIGH. The IORC and IOWC outputs of the 82288 identify I/O space transfers (see Figure 4A). CMD0 should be connected to latched 80286 A1 and CMD1 should be connected to latched 80286 A2.

I/O ports 00F8H to 00FFH are reserved for the 80286/80287 interface. To guarantee correct operation of the 80287, programs must not perform any I/O operations to these ports.

The PEREQ, PEACK, BUSY, and ERROR signals of the 80287 are connected to the same-named 80286 input. The data pins of the 80287 should be directly connected to the 80286 data bus. Note that all bus drivers connected to the 80286 local bus must be inhibited when the 80286 reads from the 80287. The use of M/I/O in the decoder prevents INTA bus cycles from disabling the data transceivers.

PROGRAMMING INTERFACE

Table 2 lists the seven data types the 80287 supports and presents the format for each type. These

values are stored in memory with the least significant digits at the lowest memory address. Programs retrieve these values by generating the lowest address. All values should start at even addresses for maximum system performance.

Internally the 80287 holds all numbers in the temporary real format. Load instructions automatically convert operands represented in memory as 16-, 32-, or 64-bit integers, 32- or 64-bit floating point number or

18-digit packed BCD numbers into temporary real format. Store instructions perform the reverse type conversion.

80287 computations use the processor's register stack. These eight 80-bit registers provide the equivalent capacity of 40 16-bit registers. The 80287 register set can be accessed as a stack, with instructions operating on the top one or two stack elements, or as a fixed register set, with instructions operating on explicitly designated registers.

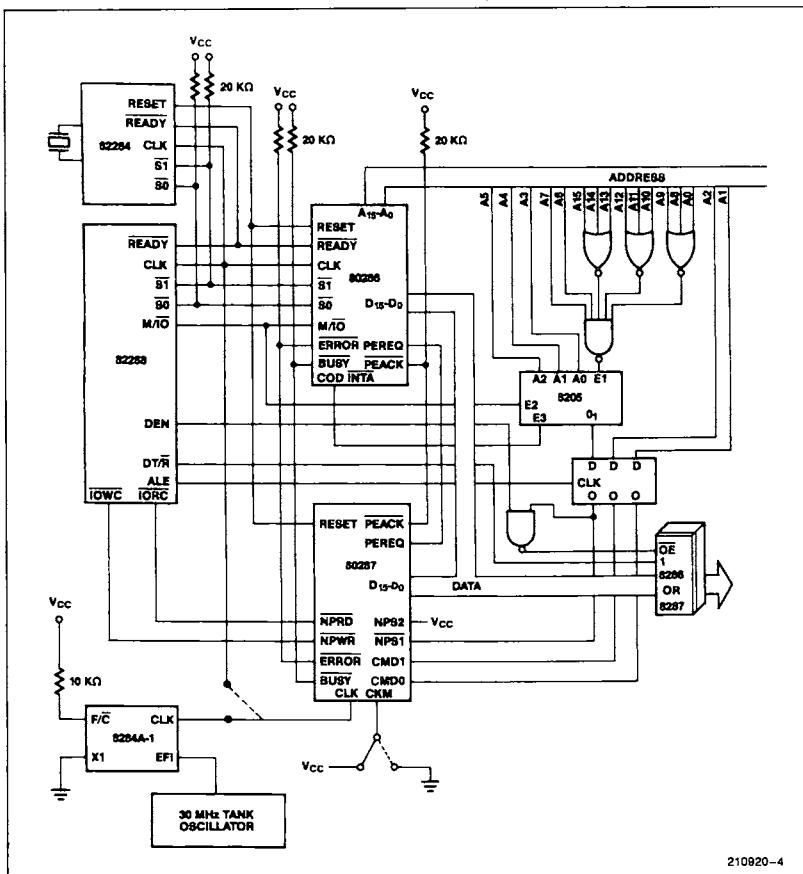


Figure 4A. 80286/80287 System Configuration

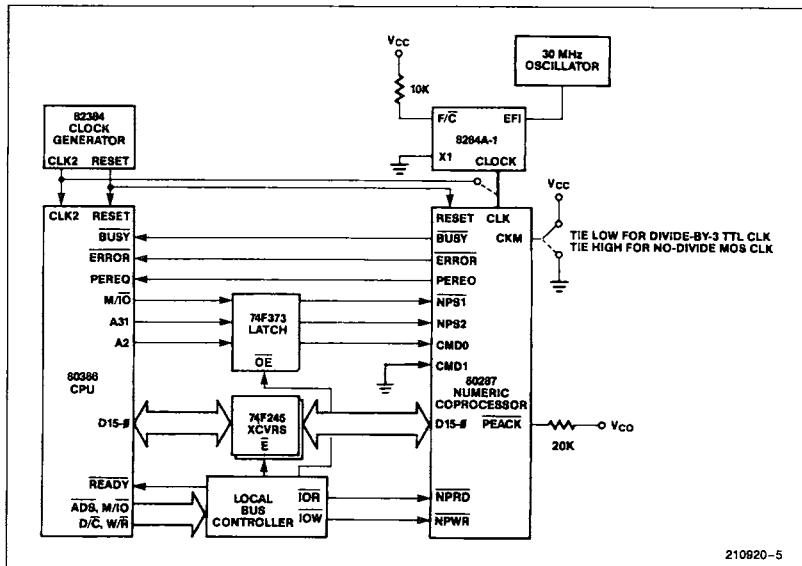


Figure 4B. 80386/80287 System Configuration



80287

Table 2. 80287 Data Type Representation in Memory

Data Formats	Range	Precision	Most Significant Byte	HIGHEST ADDRESSED BYTE
			7 0 7 0 7 0 7 0 7 0 7 0 7 0 7 0 7 0 7 0 7 0 7 0 7 0	
Word Integer	10^4	16 Bits	[] (TWOS COMPLEMENT)	
			16 0	
Short Integer	10^9	32 Bits	[] (TWOS COMPLEMENT)	
			31 0	
Long Integer	10^{19}	84 Bits	[] (TWOS COMPLEMENT)	
			63 0	
Packed BCD	10^{18}	18 Digits	[S X d ₁₇ d ₁₆ d ₁₅ d ₁₄ d ₁₃ d ₁₂ d ₁₁ d ₁₀ d ₉ d ₈ d ₇ d ₆ d ₅ d ₄ d ₃ d ₂ d ₁ d ₀]	
			79 72 0	
Short Real	$10^{\pm 38}$	24 Bits	[S BIASED EXPONENT] [SIGNIFICAND]	
			31 23 1 0	
Long Real	$10^{\pm 308}$	53 Bits	[S BIASED EXPONENT] [SIGNIFICAND]	
			63 52 1 0	
Temporary Real	$10^{\pm 4932}$	84 Bits	[S BIASED EXPONENT] [I] [SIGNIFICAND]	
			79 64 63 1 0	

NOTES:

1. S = Sign bit (0 = positive, 1 = negative)
2. d_n = Decimal digit (two per byte)
3. X = Bits have no significance; 80287 ignores when loading, zeros when storing.
4. ▲ = Position of implicit binary point
5. I = Integer bit of significant; stored in temporary real, implicit in short and long real.
6. Exponent Bias (normalized values):
 - Short Real: 127 (7FH)
 - Long Real: 1023 (3FFFH)
 - Temporary Real: 16383 (3FFFFH)
7. Packed BCD: $(-1)^q (D_{17} \dots D_0)$
8. Real: $(-1)^q (2^{E-BIAS}) (F_0 F_1 \dots)$

210920-8

Table 6 lists the 80287's instructions by class. No special programming tools are necessary to use the 80287 since all new instructions and data types are directly supported by the 80286 assembler and

appropriate high level languages. All 8086/8088 development tools which support the 8087 can also be used to develop software for the 80286/80287 in real address mode.

SOFTWARE INTERFACE

The 80286/80287 is programmed as a single processor. All communication between the 80286 and the 80287 is transparent to software. The CPU automatically controls the 80287 whenever a numeric instruction is executed. All memory addressing modes, physical memory, and virtual memory of the CPU are available for use by the NPX.

Since the NPX operates in parallel with the CPU, any errors detected by the NPX may be reported after the CPU has executed the ESCAPE instruction which caused it. To allow identification of the failing numeric instruction, the NPX contains two pointer registers which identify the address of the failing numeric instruction and the numeric memory operand if appropriate for the instruction encountering this error.

INTERRUPT DESCRIPTION

Several interrupts of the 80286 are used to report exceptional conditions while executing numeric programs in either real or protected mode. The interrupts and their functions are shown in Table 3.

PROCESSOR ARCHITECTURE

As shown in Figure 1, the NPX is internally divided into two processing elements, the bus interface unit (BIU) and the numeric execution unit (NEU). The NEU executes all numeric instructions, while the BIU receives and decodes instructions, requests operand transfers to and from memory and executes processor control instructions. The two units are able to operate independently of one another allowing the BIU to maintain asynchronous communication with the CPU while the NEU is busy processing a numeric instruction.

BUS INTERFACE UNIT

The BIU decodes the ESC instruction executed by the CPU. If the ESC code defines a math instruction, the BIU transmits the formatted instruction to the NEU. If the ESC code defines an administrative instruction, the BIU executes it independently of the NEU. The parallel operation of the NPX with the CPU is normally transparent to the user. The BIU generates the BUSY and ERROR signals for 80286/80287 processor synchronization and error notification, respectively.

The 80287 executes a single numeric instruction at a time. When executing most ESC instructions, the

Table 3. 80286 Interrupt Vectors Reserved for NPX

Interrupt Number	Interrupt Function
7	An ESC instruction was encountered when EM or TS of the 80286 MSW was set. EM = 1 indicates that software emulation of the instruction is required. When TS is set, either an ESC or WAIT instruction will cause interrupt 7. This indicates that the current NPX context may not belong to the current task.
9	The second or subsequent words of a numeric operand in memory exceeded a segment's limit. This interrupt occurs after executing an ESC instruction. The saved return address will not point at the numeric instruction causing this interrupt. After processing the addressing error, the 80286 program can be restarted at the return address with IRET. The address of the failing numeric instruction and numeric operand are saved in the 80287. An interrupt handler for this interrupt <i>must</i> execute FNINIT before any other ESC or WAIT instruction.
13	The starting address of a numeric operand is not in the segment's limit. The return address will point at the ESC instruction, including prefixes, causing this error. The 80287 has not executed this instruction. The instruction and data address is 80287 refer to a previous, correctly executed, instruction.
16	The previous numeric instruction caused an unmasked numeric error. The address of the faulty numeric instruction or numeric data operand is stored in the 80287. Only ESC or WAIT instructions can cause this interrupt. The 80286 return address will point at a WAIT or ESC instruction, including prefixes, which may be restarted after clearing the error condition in the NPX.

80286 tests the **BUSY** pin and waits until the 80287 indicates that it is not busy before initiating the command. Once initiated, the 80286 continues program execution while the 80287 executes the ESC instruction. In 8088/8087 systems, this synchronization is achieved by placing a WAIT instruction before an ESC instruction. For most ESC instructions, the 80287 does not require a WAIT instruction before the ESC opcode. However, the 80287 will operate correctly with these WAIT instruction. In all cases, a WAIT or ESC instruction should be inserted after any 80287 store to memory (except FSTSW and FSTCW) or load from memory (except FLDENV or FRSTOR) before the 80286 reads or changes the value to be sure the numeric value has already been written or read by the NPX.

Data transfers between memory and the 80287, when needed, are controlled by the PEREQ PEACK, NPPRD, NPWRD, NPS1, NPS2 signals. The 80286 does the actual data transfer with memory through its processor extension data channel. Numeric data transfers with memory performed by the 80286 use the same timing as any other bus cycle. Control signal for the 80287 are generated by the 80286 as

shown in Figure 4a, and meet the timing requirements shown in the AC requirements section.

NUMERIC EXECUTION UNIT

The NEU executes all instructions that involve the register stack; these include arithmetic, logical, transcendental, constant and data transfer instructions. The data path in the NEU is 84 bits wide (68 significand bits, 15 exponent bits and a sign bit) which allows internal operand transfers to be performed at very high speeds.

When the NEU begins executing an instruction, it activates the BIU **BUSY** signal. This signal is used in conjunction with the CPU WAIT instruction or automatically with most of the ESC instructions to synchronize both processors.

REGISTER SET

The 80287 register set is shown in Figure 5. Each of the eight data registers in the 8087's register stack

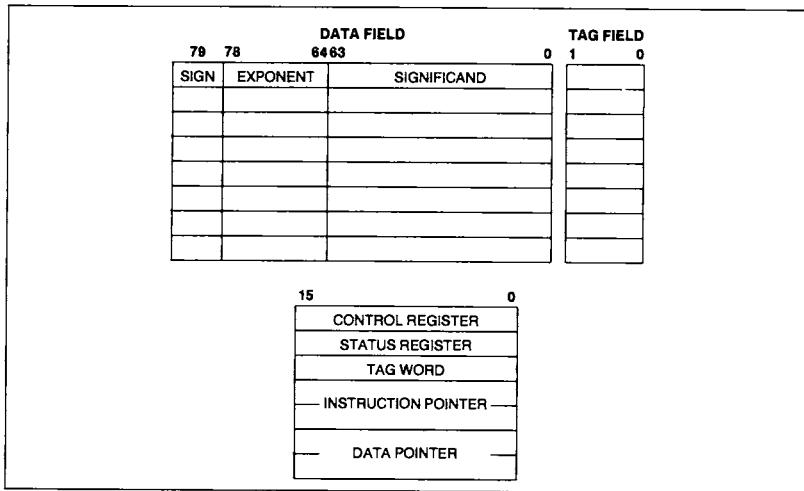


Figure 5. 80287 Register Set

is 80 bits wide and is divided into "fields" corresponding to the NPX's temporary real data type.

At a given point in time the TOP field in the status word identifies the current top-of-stack register. A "push" operation decrements TOP by 1 and loads a value into the new top register. A "pop" operation stores the value from the current top register and then increments TOP by 1. Like 80286 stacks in memory, the 80287 register stack grows "down" toward lower-addressed registers.

Instructions may address the data registers either implicitly or explicitly. Many instructions operate on the register at the TOP of the stack. These instructions implicitly address the register pointed by the TOP. Other instructions allow the programmer to explicitly specify the register which is to be used. This explicit register addressing is also "top-relative."

STATUS WORD

The 16-bit status word (in the status register) shown in Figure 6 reflects the overall state of the 80287. It may be read and inspected by CPU code. The busy bit (bit 15) indicates whether the NEU is executing an instruction ($B = 1$) or is idle ($B = 0$).

The instructions FSTSW, FSTSW AX, FSTENV, and FSAVE which store the status word are executed exclusively by the BIU and do not set the busy bit themselves or require the Busy bit be cleared in order to be executed.

The four numeric condition code bits ($C_0 - C_3$) are similar to the flags in a CPU: instructions that perform arithmetic operations update these bits to reflect the outcome of NPX operations. The effect of these instructions on the condition code is summarized in Tables 4a and 4b.

Bits 14–12 of the status word point to the 80287 register that is the current top-of-stack (TOP) as described above. Figure 6 shows the six error flags in bits 5–0 of the status word. Bits 5–0 are set to indicate that the NEU has detected an exception while executing an instruction. The section on exception handling explains how they are set and used.

Bit 7 is the error summary status bit. This bit is set if any unmasked exception bit is set and cleared otherwise. If this bit is set, the ERROR signal is asserted.

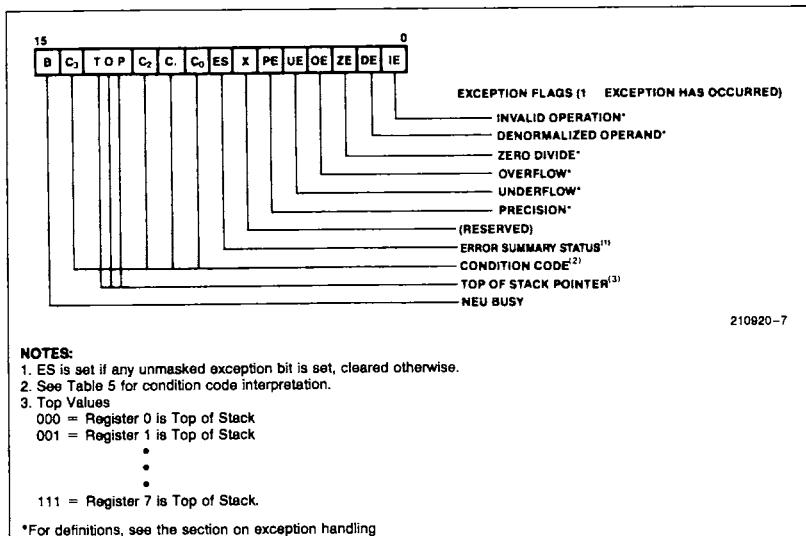


Figure 6. 80287 Status Word

TAG WORD

The tag word marks the content of each register as shown in Figure 7. The principal function of the tag word is to optimize the NPX's performance. The eight two-bit tags in the tag word can be used, however, to interpret the contents of 80287 registers.

INSTRUCTION AND DATA POINTERS

The instruction and data pointers (See Figures 8a and 8b) are provided for user-written error handlers. Whenever the 80287 executes a new instruction, the BIU saves the instruction address, the operand address (if present) and the instruction opcode. 80287 instructions can store this data into memory.

The instruction and data pointers appear in one of two formats depending on the operating mode of the 80287. In real mode, these values are the 20-bit physical address and 11-bit opcode formatted like the 8087. In protection mode, these values are the

32-bit virtual address used by the program which executed an ESC instruction. The same FLDENV/FSTENV/FSAVE/FRSTOR instructions as those of the 8087 are used to transfer these values between the 80287 registers and memory.

The saved instruction address in the 80287 will point at any prefixes which preceded the instruction. This is different than in the 8087 which only pointed at the ESCAPE instruction opcode.

CONTROL WORD

The NPX provides several processing options which are selected by loading a word from memory into the control word. Figure 9 shows the format and encoding of fields in the control word.

The low order byte of this control word configures the 80287 error and exception masking. Bits 5-0 of the control word contain individual masks for each of the six exceptions that the 80287 recognizes. The high order byte of the control word configures the

Table 4a. Condition Code Interpretation

Instruction Type	C ₃	C ₂	C ₁	C ₀	Interpretation
Compare, Test	0	0	X	0	ST > Source or 0 (FTST)
	0	0	X	1	ST < Source or 0 (FTST)
	1	0	X	0	ST = Source or 0 (FTST)
	1	1	X	1	ST is not comparable
Remainder	Q ₁	0	Q ₀	Q ₂	Complete reduction with three low bits of quotient (See Table 5b)
	U	1	U	U	Incomplete Reduction
Examine	0	0	0	0	Valid, positive unnormalized
	0	0	0	1	Invalid, positive, exponent = 0
	0	0	1	0	Valid, negative, unnormalized
	0	0	1	1	Invalid, negative, exponent = 0
	0	1	0	0	Valid, positive, normalized
	0	1	0	1	Infinity, positive
	0	1	1	0	Valid, negative, normalized
	0	1	1	1	Infinity, negative
	1	0	0	0	Zero, positive
	1	0	0	1	Empty
	1	0	1	0	Zero, Negative
	1	0	1	1	Empty
	1	1	0	0	Invalid, positive, exponent = 0
	1	1	0	1	Empty
	1	1	1	0	Invalid, negative, exponent = 0
	1	1	1	1	Empty

NOTES:

1. ST = Top of Stack
2. X = value is not affected by instruction
3. U = value is undefined following instruction
4. Q_n = Quotient bit n

Table 4b. Condition Code Interpretation after FPREM (See Note 1) Instruction as a Function of Dividend Value

Dividend Range	Q ₂	Q ₁	Q ₀
Dividend < 2 * Modulus	C ₃	C ₁	Q ₀
Dividend < 4 * Modulus	C ₃	Q ₁	Q ₀
Dividend ≥ 4 * Modulus	Q ₂	Q ₁	Q ₀

NOTE:

1. Previous value of indicated bit, not affected by FPREM instruction execution.

80287 operating mode including precision, rounding, and infinity control. The precision control bits (bits 9–8) can be used to set the 80287 internal operating precision at less than the default of temporary real (80-bit) precision. This can be useful in providing compatibility with the early generation arithmetic processors of smaller precision than the 80287. The rounding control bits (bits 11–10) provide for directed rounding and true chop as well as the unbiased round to nearest even mode specified in the IEEE standard. Control over closure of the number space at infinity is also provided (either affine closure: ±∞, or projective closure: ∞, is treated as unsigned, may be specified).

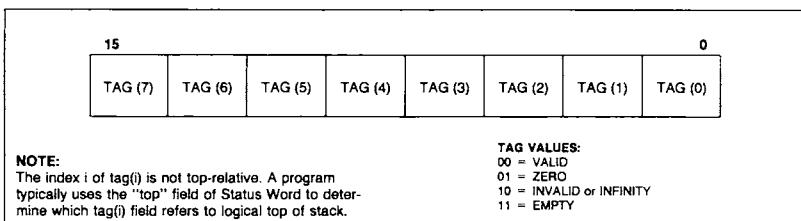


Figure 7. 80287 Tag Word

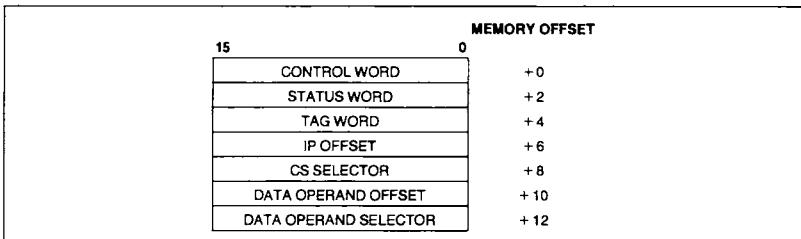


Figure 8a. Protected Mode 80287 Instruction and Data Pointer Image in Memory

EXCEPTION HANDLING

The 80287 detects six different exception conditions that can occur during instruction execution. Any or all exceptions will cause the assertion of external ERROR signal and ES bit of the Status Word if the appropriate exception masks are not set.

The exceptions that the 80287 detects and the 'default' procedures that will be carried out if the exception is masked, are as follows:

Invalid Operation: Stack overflow, stack underflow, indeterminate form ($0/0$, ∞ , $-\infty$, etc) or the use of a Non-Number (NAN) as an operand. An exponent value of all ones and non-zero significand is reserved to identify NaNs. If this exception is masked, the 80287 default response is to generate a specific

NAN called INDEFINITE, or to propagate already existing NaNs as the calculation result.

Overflow: The result is too large in magnitude to fit the specified format. The 80287 will generate an encoding for infinity if this exception is masked.

Zero Divisor: The divisor is zero while the dividend is a non-infinite, non-zero number. Again, the 80287 will generate an encoding for infinity if this exception is masked.

Underflow: The result is non-zero but too small in magnitude to fit in the specified format. If this exception is masked the 80287 will denormalize (shift right) the fraction until the exponent is in range. The process is called gradual underflow.

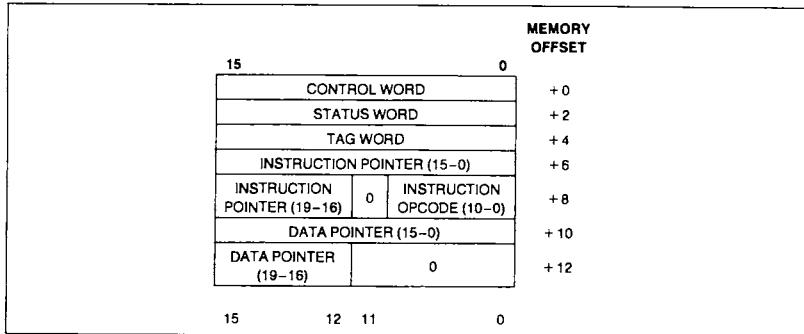


Figure 8b. Real Mode 80287 Instruction and Data Pointer image in Memory

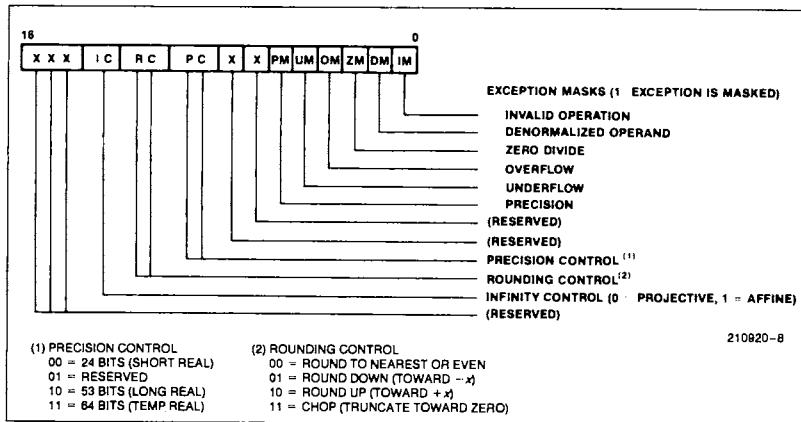


Figure 9. 80287 Control Word

Denormalized Operand: At least one of the operands is denormalized; it has the smallest exponent but a non-zero significand. Normal processing continues if this exception is masked off.

Inexact Result: The true result is not exactly representable in the specified format; the result is rounded according to the rounding mode, and this flag is set. If this exception is masked, processing will simply continue.

If the error is not masked, the corresponding error bit and the error status bit (ES) in the control word will be set, and the ERROR output signal will be asserted. If the CPU attempts to execute another ESC or WAIT instruction, exception 7 will occur.

The error condition must be resolved via an interrupt service routine. The 80287 saves the address of the floating point instruction causing the error as well as the address of the lowest memory location of any memory operand required by that instruction.

8086/8087 COMPATIBILITY:

The 80286/80287 supports portability of 8086/8087 programs when it is in the real address mode. However, because of differences in the numeric error handling techniques, error handling routines *may* need to be changed. The differences between an 80286/80287 and 8086/8087 are:

1. The NPX error signal does not pass through an interrupt controller (8087 INT signal does).

Therefore, any interrupt controller oriented instructions for the 8086/8087 may have to be deleted.

2. Interrupt vector 16 must point at the numeric error handler routine.
3. The saved floating point instruction address in the 80287 includes any leading prefixes before the ESCAPE opcode. The corresponding saved address of the 8087 does not include leading prefixes.
4. In protected mode, the format of the saved instruction and operand pointers is different than for the 8087. The instruction opcode is not saved—it must be read from memory if needed.
5. Interrupt 7 will occur when executing ESC instructions with either TS or EM or MSW = 1. If TS or MSW = 1 then WAIT will also cause interrupt 7. An interrupt handler should be added to handle this situation.
6. Interrupt 9 will occur if the second or subsequent words of a floating point operand fall outside a segment's size. Interrupt 13 will occur if the starting address of a numeric operand falls outside a segment's size. An interrupt handler should be added to report these programming errors.

In the protected mode, 8086/8087 application code can be directly ported via recompilation if the 80286 memory protection rules are not violated.



80287

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias	0°C to 70°C
Storage Temperature	-65°C to +150°C
Case Temperature	0°C to 85°C
Voltage on any Pin with Respect to Ground	-1.0 to +7V
Power Dissipation	3.0 Watt

*Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS $T_A = 0^\circ\text{C}$ to 70°C , $T_C = 0^\circ\text{C}$ to 85°C , $V_{CC} = 5\text{V} \pm 5\%$ **ALL SPEEDS SELECTIONS**

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input LOW Voltage	-0.5	0.8	V	
V_{IH}	Input HIGH Voltage	2.0	$V_{CC} + 0.5$	V	
V_{IHC}	Clock Input HIGH Voltage CKM = 1; CKM = 0:	2.0 3.8	$V_{CC} + 1$ $V_{CC} + 1$	V V	
V_{ILC}	Clock Input LOW Voltage CKM = 1 CKM = 0	-0.5 -0.5	0.8 0.6	V V	
V_{OL}	Output LOW Voltage		0.45	V	$I_{OL} = 3.0\text{ mA}$
V_{OH}	Output HIGH Voltage	2.4		V	$I_{OH} = -400\text{ }\mu\text{A}$
I_{LU}	Input Leakage Current	•	± 10	μA	$OV \leq V_{IN} \leq V_{CC}$
I_{LO}	Output Leakage Current	•	± 10	μA	$0.45V \leq V_{OUT} \leq V_{CC}$
I_{CC}	Power Supply Current		600 475 375	mA mA mA	$T_A = 0^\circ\text{C}$ $T_A = 25^\circ\text{C}$ $T_A = 70^\circ\text{C}$
C_{IN}	Input Capacitance	•	10	pF	$f_C = \text{MHz}$
C_O	Input/Output Capacitance (D0-D15)	•	20	pF	$V_C = 1\text{ MHz}$
C_{CLK}	CLK Capacitance	•	12	pF	$f_C = 1\text{ MHz}$



80287

A.C. CHARACTERISTICS $T_A = 0^\circ\text{C}$ to 70°C , $T_{\text{CASE}} = 0^\circ\text{C}$ to 85°C , $V_{\text{CC}} = 5\text{V} \pm 5\%$
TIMING REQUIREMENTS

A.C. timings are referenced to 0.8V and 2.0V points on signals unless otherwise noted.

Symbol	Parameter	80287-3 5 MHz		80287-6 6 MHz		80287-8 8 MHz		80287-10 10 MHz Preliminary		Units	Test Conditions
		Min	Max	Min	Max	Min	Max	Min	Max		
T_{CLCL}	CLK Period CKM = 1: CKM = 0:	200 62.5	500 250	166 62.5	500 166	125 50	500 166	100 40	500 166	ns	
T_{CLCH}	CLK LOW Time CKM = 1: CKM = 0:	118 15	230	100 15	343 146	68 15	343 146	62 11	343 146	ns	At 0.8V At 0.6V
T_{CHCL}	CLK HIGH Time CKM = 1: CKM = 0:	69 20	235	50 20	230 151	43 20	230 151	28 18	230 151	ns	At 2.0V At 3.6V
T_{CH1CH2}	CLK Rise Time		10		10		10		10	ns	1.0V to 3.6V if CKM = 0
T_{CL2CL1}	CLK Fall Time		10		10		10		10	ns	3.6V to 1.0V if CKM = 0
T_{DYZH}	Data Setup to NPWR Inactive	75		75		75		75		ns	
T_{WHDX}	Data Hold from NPWR Inactive	30		30		18		18		ns	
T_{WLWH} T_{TRLH}	NPWR NPROD Active Time	95		95		90		90		ns	At 0.8V
T_{AVRL} T_{AVWL}	Command Valid to NPWR or NPROD Active	0		0		0		0		ns	
T_{MHRL}	Minimum Delay from PEREQ Active to NPROD Active	130		130		130		100		ns	
T_{KLKH}	PEAK Active Time	85		85		85		60		ns	At 0.8V
T_{KHKL}	PEAK Inactive Time	250		250		250		200		ns	At 2.0V
T_{KHCH}	PEAK Inactive to NPWR, NPROD Inactive	50		50		40		40		ns	
T_{CHKL}	NPWR, NPROD Inactive to PEAK Active	-30		-30		-30		-30		ns	
T_{WHAX} T_{RHAX}	Command Hold from NPWR, NPROD Inactive	30		30		30		22		ns	
T_{KLCL}	PEAK Active Setup to NPWR NPROD Active	50		50		40		40		ns	



80287

A.C. CHARACTERISTICS $T_A = 0^\circ\text{C}$ to 70°C , $T_{\text{CASE}} = 0^\circ\text{C}$ to 85°C , $V_{\text{CC}} = 5\text{V} \pm 5\%$ (Continued)**TIMING REQUIREMENTS** (Continued)

A.C. timings are referenced to 0.8V and 2.0V points on signals unless otherwise noted.

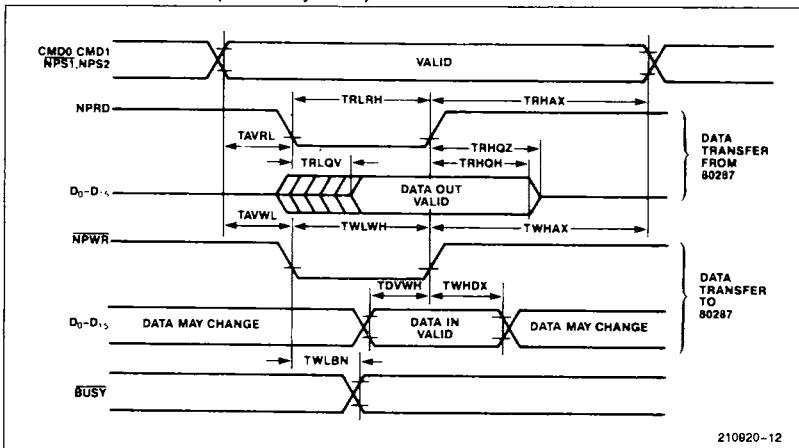
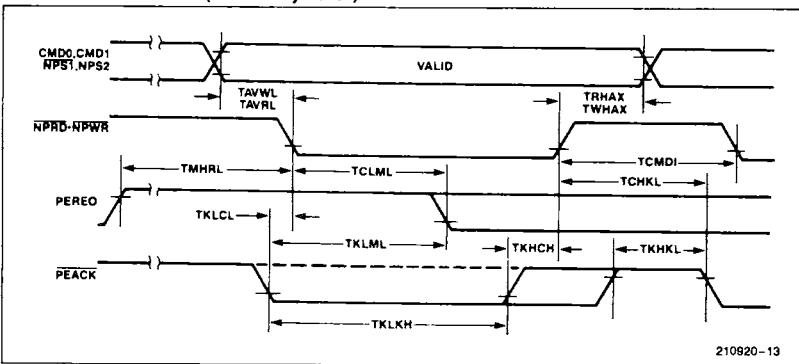
Symbol	Parameter	80287-3 5 MHz		80287-6 6 MHz		80287-8 8 MHz		80287-10 10 MHz Preliminary		Units	Test Conditions
		Min	Max	Min	Max	Min	Max	Min	Max		
T_{IVCL}	NPWR, NPROD to CLK Setup Time	70		70		70		53		ns	(Note 1)
T_{CLIH}	NPWR, NPROD from CLK Hold Time	45		45		45		37		ns	(Note 1)
T_{RSCL}	RESET to CLK Setup Time	20		20		20		20		ns	(Note 1)
T_{CLRS}	RESET from CLK Hold Time	20		20		20		20		ns	(Note 1)

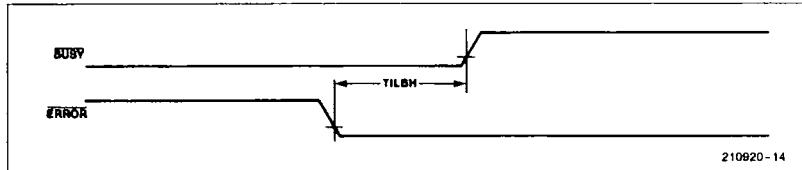
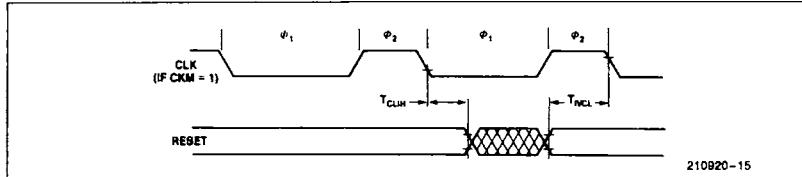
TIMING RESPONSES

Symbol	Parameter	80287-3 5 MHz		80287-6 6 MHz		80287-8 8 MHz		80287-10 10 MHz Preliminary		Units	Test Conditions
		Min	Max	Min	Max	Min	Max	Min	Max		
T_{RHOZ}	NPROD Inactive to Data Float		37.5		37.5		35		21	ns	(Note 2)
T_{RLOV}	NPROD Active to Data Valid		60		60		60		60	ns	(Note 3)
T_{ILBH}	ERROR Active to BUSY Inactive	100		100		100		100		ns	(Note 4)
T_{WLBV}	NPWR Active to BUSY Active		100		100		100		100	ns	(Note 5)
T_{KLML}	PEAK Active to PEREQ Inactive		127		127		127		100	ns	(Note 6)
T_{CMDI}	Command Inactive Time									ns	At 2.0V
	Write-to-Write	95		95		95		75			
	Read-to-Read	250		95		95		75			
	Write-to-Read	105		95		95		75			
	Read-to-Write	95		95		95		75			
T_{RHOD}	Data Hold from NPROD Inactive	5		3		3		3		ns	(Note 7)

NOTES:

1. This is an asynchronous input. This specification is given for testing purposes only, to assure recognition at a specific CLK edge.
2. Float condition occurs when output current is less than I_{LO} on D0-D15.
3. D0-D15 IoSINF : $XL = 100 \text{ pF}$.
4. BUSY loading: $CL = 100 \text{ pF}$.
5. BUSY loading: $CL = 100 \text{ pF}$.
6. On last data transfer on numeric instruction.
7. D0-D15 loading: $CL = 100 \text{ pF}$.

WAVEFORMS**DATA TRANSFER TIMING (Initiated by 80286)****DATA CHANNEL TIMING (Initiated by 80287)**

WAVEFORMS (Continued)**ERROR OUTPUT TIMING****CLK, RESET TIMING (CKM = 1)****NOTE:**

Reset, NPWR, NRD are inputs asynchronous to CLK. Timing requirements on this page are given for testing purposes only, to assure recognition at a specific CLK edge.

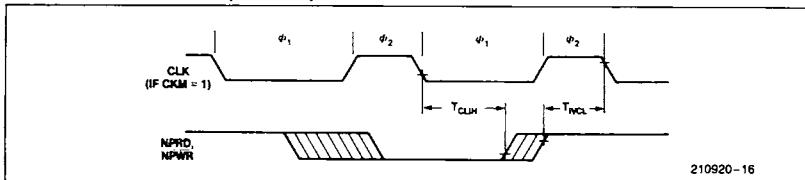
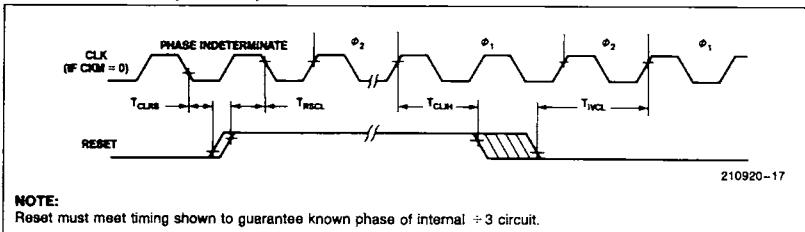
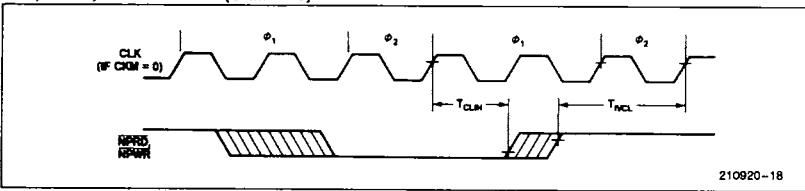
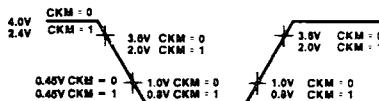
WAVEFORMS (Continued)**CLK, NPROD, NPWR TIMING (CKM = 1)****CLK, RESET TIMING (CKM = 0)****CLK, NPROD, NPWR TIMING (CKM = 0)**

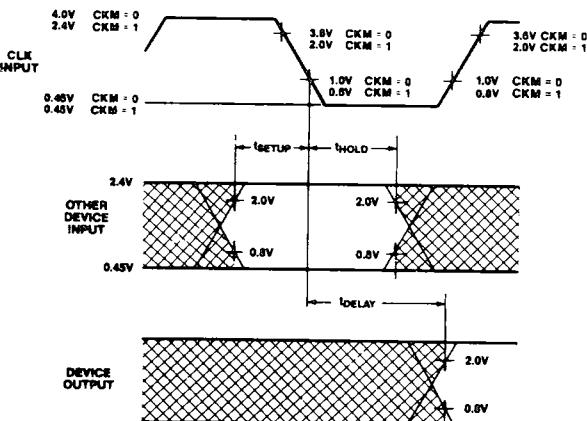
Table 6. 80287 Extensions to the 80286 Instruction Set

Data Transfer	Options 8,16 Bit Displacement				Clock Count Range					
	32 Bit Real	32 Bit Integer	64 Bit Real	16 Bit Integer						
FLD ~ LOAD	MF	-			00	01	10	11		
Integer/Real Memory to ST(0)	ESCAPE	MF 1	MOD 0 0 0 R/M		DISP		38-56	52-60	40-60	46-54
Long Integer Memory to ST(0)	ESCAPE	1 1 1	MOD 1 0 1 R/M		DISP			60-68		
Temporary Real Memory to ST(0)	ESCAPE	0 1 1	MOD 1 0 1 R/M		DISP		53-65			
BCD Memory to ST(0)	ESCAPE	1 1 1	MOD 1 0 0 R/M		DISP		290-310			
ST(i) to ST(0)	ESCAPE	0 0 1	1 1 0 0 0 ST(i)				17-22			
FST = STORE										
ST(0) to Integer/Real Memory	ESCAPE	MF 1	MOD 0 1 0 R/M		DISP		84-90	82-92	96-104	80-90
ST(0) to ST(i)	ESCAPE	1 0 1	1 1 0 1 0 ST(i)				15-22			
FSTP = STORE AND POP										
ST(0) to Integer/Real Memory	ESCAPE	MF 1	MOD 0 1 1 R/M		DISP		86-92	84-94	98-106	82-92
ST(0) to Long Integer Memory	ESCAPE	1 1 1	MOD 1 1 1 R/M		DISP			94-105		
ST(0) to Temporary Real Memory	ESCAPE	0 1 1	MOD 1 1 1 R/M		DISP		52-58			
ST(0) to BCD Memory	ESCAPE	1 1 1	MOD 1 1 0 R/M		DISP		520-540			
ST(0) to ST(i)	ESCAPE	1 0 1	1 1 0 1 1 ST(i)				17-24			
FXCH = Exchange ST(i) and ST(0)	ESCAPE	0 0 1	1 1 0 0 1 ST(i)				10-15			
Comparison										
FCOM = Compare										
Integer/Real Memory to ST(0)	ESCAPE	MF 0	MOD 0 1 0 R/M		DISP		60-70	78-91	65-75	72-86
ST(i) to ST(0)	ESCAPE	0 0 0	1 1 0 1 0 ST(i)				40-50			
FCOMP = Compare and Pop										
Integer/Real Memory to ST(0)	ESCAPE	MF 0	MOD 0 1 1 R/M		DISP		63-73	80-93	67-77	74-88
ST(i) to ST(0)	ESCAPE	0 0 0	1 1 0 1 1 ST(i)				45-52			
FCOMPP = Compare ST(1) to ST(0) and Pop Twice	ESCAPE	1 1 0	1 1 0 1 1 0 0 1				45-55			
FTST = Test ST(0)	ESCAPE	0 0 1	1 1 1 0 0 1 0 0				38-48			
FXAM = Examine ST(0)	ESCAPE	0 0 1	1 1 1 0 0 1 0 1				12-23			

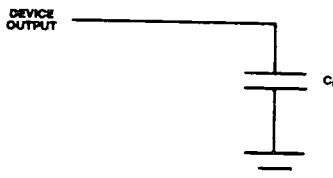
210920-19



AC Drive and Measurement Points—CLK Input



AC Setup, Hold and Delay Time Measurement—General



AC Test Loading on Outputs



80287

Table 6. 80287 Extensions to the 80286 Instruction Set (Continued)

Constants	MF	-	Optional 8,16 Bit Displacement	Clock Count Range			
				32 Bit Real	32 Bit Integer	64 Bit Real	16 Bit Integer
FLDZ = LOAD + 0.0 into ST(0)	ESCAPE	0 0 1	1 1 1 0 1 1 1 0			00	01
						10	11
FLD1 = LOAD + 1.0 into ST(0)	ESCAPE	0 0 1	1 1 1 0 1 0 0 0			11-17	
FLDP1 = LOAD π into ST(0)	ESCAPE	0 0 1	1 1 1 0 1 0 1 1			15-21	
FLDLST = LOAD $\log_2 10$ into ST(0)	ESCAPE	0 0 1	1 1 1 0 1 0 0 1			16-22	
FLDLG2E = LOAD $\log_2 e$ into ST(0)	ESCAPE	0 0 1	1 1 1 0 1 0 1 0			16-22	
FLDLG2 = LOAD $\log_{10} 2$ into ST(0)	ESCAPE	0 0 1	1 1 1 0 1 1 1 0			15-21	
FLDLN2 = LOAD \log_2 into ST(0)	ESCAPE	0 0 1	1 1 1 0 1 1 0 1			16-24	
						17-23	
Arithmetic							
FADD = Addition							
Integer/Real Memory with ST(0)	ESCAPE	MF 0	MOD 0 0 0 R/M	DISP		90-120	108-143
						95-125	102-137
ST(i) and ST(0)	ESCAPE	d P 0	1 1 0 0 0 ST(i)			70-100	(Note 1)
FSUB = Subtraction							
Integer/Real Memory with ST(0)	ESCAPE	MF 0	MOD 1 0 R R/M	DISP		90-120	108-143
						95-125	102-137
ST(i) and ST(0)	ESCAPE	d P 0	1 1 0 R R/M			70-100	(Note 1)
FMUL = Multiplication							
Integer/Real Memory with ST(0)	ESCAPE	MF 0	MOD 0 0 1 R/M	DISP		110-125	130-144
						112-168	124-138
ST(i) and ST(0)	ESCAPE	d P 0	1 1 0 0 1 R/M			90-145	(Note 1)
FDIV = Division							
Integer/Real Memory with ST(0)	ESCAPE	MF 0	MOD 1 1 R R/M	DISP		215-225	230-243
						220-230	224-238
ST(i) and ST(0)	ESCAPE	d P 0	1 1 1 1 R R/M			193-203	(Note 1)
FSQRT = Square Root of ST(0)	ESCAPE	0 0 1	1 1 1 1 1 0 1 0			180-186	
FSCALE = Scale ST(0) by ST(1)	ESCAPE	0 0 1	1 1 1 1 1 1 0 1			32-38	
FPREM = Partial Remainder of ST(0) + ST(1)	ESCAPE	0 0 1	1 1 1 1 1 0 0 0			15-190	
FRNDINT = Round ST(0) to Integer	ESCAPE	0 0 1	1 1 1 1 1 1 0 0			18-50	
						210920-20	

NOTE:

1. If P = 1 then add 5 clocks.



80287

Table 6. 80287 Extensions to the 80286 Instruction Set (Continued)

		Optional 8,16 Bit Displacement	Clock Count Range
FXTRACT - Extract Components of ST(0)	ESCAPE 0 0 1 1 1 1 1 0 1 0 0		27-55
FABS - Absolute Value of ST(0)	ESCAPE 0 0 1 1 1 1 0 0 0 0 1		10-17
FCNS - Change Sign of ST(0)	ESCAPE 0 0 1 1 1 1 0 0 0 0 0		10-17
Transcendental			
FPTAN - Partial Tangent of ST(0) - ST(1)	ESCAPE 0 0 1 1 1 1 1 0 0 1 0		30-540
FPATAN = Partial Arc tangent of ST(0) - ST(1)	ESCAPE 0 0 1 1 1 1 1 0 0 1 1		250-800
F2XM1 = $2^{ST(0)} - 1$	ESCAPE 0 0 1 1 1 1 1 0 0 0 0		310-630
FYL2X = ST(1) • Log ₂ ST(0)	ESCAPE 0 0 1 1 1 1 1 0 0 0 1		900-1100
FYL2XP1 = ST(1) • Log ₂ ST(0) + 1	ESCAPE 0 0 1 1 1 1 1 1 0 0 1		700-1000
Processor Control			
FINIT - Initialize NPX	ESCAPE 0 1 1 1 1 1 0 0 0 1 1		2-8
FSETPM = Enter Protected Mode	ESCAPE 0 1 1 1 1 1 0 0 1 0 0		2-8
FBTW AX = Store Control Word	ESCAPE 1 1 1 1 1 1 0 0 0 0 0		10-16
FLDCW = Load Control Word	ESCAPE 0 0 1 MOD 1 0 1 R/M	DISP	7-14
FSTCW = Store Control Word	ESCAPE 0 0 1 MOD 1 1 1 R/M	DISP	12-18
FSTSW = Store Status Word	ESCAPE 1 0 1 MOD 1 1 1 R/M	DISP	12-18
FCLEX = Clear Exceptions	ESCAPE 0 1 1 1 1 1 0 0 0 1 0		2-8
FSTENV = Store Environment	ESCAPE 0 0 1 MOD 1 1 0 R/M	DISP	40-50
FLDENV = Load Environment	ESCAPE 0 0 1 MOD 1 0 0 R/M	DISP	35-45
FSAVE = Save State	ESCAPE 1 0 1 MOD 1 1 0 R/M	DISP	205-215
FRSTOR = Restore State	ESCAPE 1 0 1 MOD 1 0 0 R/M	DISP	205-215
FINCSTP = Increment Stack Pointer	ESCAPE 0 0 1 1 1 1 1 0 1 1 1		8-12
FDECSTP = Decrement Stack Pointer	ESCAPE 0 0 1 1 1 1 1 0 1 1 0		8-12

210920-21

Table 6. 80287 Extensions to the 80286 Instruction Set (Continued)

	Clock Count Range
FFREE = Free ST(i)	ESCAPE 1 0 1 1 1 0 0 0 ST(i) 9-16
FNOP = No Operation	ESCAPE 0 0 1 1 1 0 1 0 0 0 10-18 210920-22

NOTES:

- if mod = 00 then DISP = 0*, disp-low and disp-high are absent
 if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
 if mod = 10 then DISP = disp-high; disp-low
 if mod = 11 then r/m is treated as an ST(i) field
- if r/m = 000 then EA = (BX) + (SI) + DISP
 if r/m = 001 then EA = (BX) + (DI) + DISP
 if r/m = 010 then EA = (BP) + (SI) + DISP
 if r/m = 011 then EA = (BP) + (DI) + DISP
 if r/m = 100 then EA = (SI) + DISP
 if r/m = 101 then EA = (DI) + DISP
 if r/m = 110 then EA = (BP) + DISP
 if r/m = 111 then EA = (BX) + DISP
 *except if mod = 000 and r/m = 110 then EA = disp-high; disp-low.
- MF = Memory Format
 00—32-bit Real
 01—32-bit Integer
 10—64-bit Real
 11—16-bit Integer
- ST(0) = Current stack top
 ST(i) = ith register below stack top
- d = Destination
 0—Destination is ST(0)
 1—Destination is ST(i)
- P = Pop
 0—No pop
 1—Pop ST(0)
- R = Reverse: When d = 1 reverse the sense of R
 0—Destination (op) Source
 1—Source (op) Destination
- For FSQRT: $-0 \leq ST(0) \leq +\infty$
 For FSCALE: $-2^{15} \leq ST(1) < +2^{15}$ and ST(1) integer
 For F2XM1: $0 \leq ST(0) \leq 2^{-1}$
 For FYL2X: $0 < ST(0) < \infty$
 $-\infty < ST(1) < +\infty$
 For FYL2XP1: $0 \leq IST(0) < (2 - \sqrt{2})/2$
 For FPTAN: $0 \leq ST(0) \leq \pi/4$
 For FPATAN: $0 \leq ST(0) < ST(1) < +\infty$
- ESCAPE bit pattern is 11011.



ADVANCE INFORMATION

80386 HIGH PERFORMANCE 32-BIT CHMOS MICROPROCESSOR WITH INTEGRATED MEMORY MANAGEMENT

- Flexible 32-Bit Microprocessor
 - 8, 16, 32-Bit Data Types
 - 8 General Purpose 32-Bit Registers
- Very Large Address Space
 - 4 Gigabyte Physical
 - 64 Terabyte Virtual
 - 4 Gigabyte Maximum Segment Size
- Integrated Memory Management Unit
 - Virtual Memory Support
 - Optional On-Chip Paging
 - 4 Levels of Protection
 - Fully Compatible with 80286
- Object Code Compatible with All 8086 Family Microprocessors
- Virtual 8086 Mode Allows Running of 8086 Software In a Protected and Paged System
- Hardware Debugging Support
- Optimized for System Performance
 - Pipelined Instruction Execution
 - On-Chip Address Translation Caches
 - 16 and 20 MHz Clock
 - 32 and 40 Megabytes/Sec Bus Bandwidth
- High Speed Numerics Support via 80287 and 80387 Coprocessors
- Complete System Development Support
 - Software: C, PL/M, Assembler System Generation Tools
 - Debuggers: PSCOPE, ICE™-386
- High Speed CHMOS III Technology
- 132 Pin Grid Array Package

(See Packaging Specification, Order #231369)

The 80386 is an advanced 32-bit microprocessor designed for applications needing very high performance and optimized for multitasking operating systems. The 32-bit registers and data paths support 32-bit addresses and data types. The processor addresses up to four gigabytes of physical memory and 64 terabytes (2^{46}) of virtual memory. The integrated memory management and protection architecture includes address translation registers, advanced multitasking hardware and a protection mechanism to support operating systems. In addition, the 80386 allows the simultaneous running of multiple operating systems.

Instruction pipelining, on-chip address translation, and high bus bandwidth ensure short average instruction execution times and high system throughput. The 80386 processor is capable of execution at sustained rates of between 4 and 5 million instructions per second.

The 80386 offers new testability and debugging features. Testability features include a self-test and direct access to the page translation cache. Four new breakpoint registers provide breakpoint traps on code execution or data accesses, for powerful debugging of even ROM-based systems.

Object-code compatibility with all 8086 family members (8086, 8088, 80186, 80188, 80286) means the 80386 offers immediate access to the world's largest microprocessor software base.

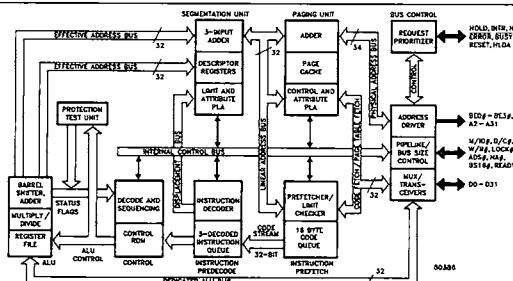


Figure 1-1. 80386 Pipelined 32-Bit Microarchitecture

231630-49

UNIX™ is a Trademark of AT&T Bell Labs.
MS-DOS is a Trademark of MICROSOFT Corporation.



2. BASE ARCHITECTURE

2.1 INTRODUCTION

The 80386 consists of a central processing unit, a memory management unit and a bus interface.

The central processing unit consists of the execution unit and instruction unit. The execution unit contains the eight 32-bit general purpose registers which are used for both address calculation, data operations and a 64-bit barrel shifter used to speed shift, rotate, multiply, and divide operations. The multiply and divide logic uses a 1-bit per cycle algorithm. The multiply algorithm stops the iteration when the most significant bits of the multiplier are all zero. This allows typical 32-bit multiplies to be executed in under one microsecond. The instruction unit decodes the instruction opcodes and stores them in the decoded instruction queue for immediate use by the execution unit.

The memory management unit (MMU) consists of a segmentation unit and a paging unit. Segmentation allows the managing of the logical address space by providing an extra addressing component, one that allows easy code and data relocatability, and efficient sharing. The paging mechanism operates beneath and is transparent to the segmentation process, to allow management of the physical address space. Each segment is divided into one or more 4K byte pages. To implement a virtual memory system, the 80386 supports full restartability for all page and segment faults.

Memory is organized into one or more variable length segments, each up to four gigabytes in size. A given region of the linear address space, a segment, can have attributes associated with it. These attributes include its location, size, type (i.e. stack, code or data), and protection characteristics. Each task on an 80386 can have a maximum of 16,381 segments of up to four gigabytes each, thus providing 64 terabytes (trillion bytes) of virtual memory to each task.

The segmentation unit provides four-levels of protection for isolating and protecting applications and the operating system from each other. The hardware enforced protection allows the design of systems with a high degree of integrity.

The 80386 has two modes of operation: Real Address Mode (Real Mode), and Protected Virtual Address Mode (Protected Mode). In Real Mode the 80386 operates as a very fast 8086, but with 32-bit extensions if desired. Real Mode is required primarily

to setup the processor for Protected Mode operation. Protected Mode provides access to the sophisticated memory management, paging and privilege capabilities of the processor.

Within Protected Mode, software can perform a task switch to enter into tasks designated as Virtual 8086 Mode tasks. Each such task behaves with 8086 semantics, thus allowing 8086 software (an application program, or an entire operating system) to execute. The Virtual 8086 tasks can be isolated and protected from one another and the host 80386 operating system, by the use of paging, and the I/O Permission Bitmap.

Finally, to facilitate high performance system hardware designs, the 80386 bus interface offers address pipelining, dynamic data bus sizing, and direct Byte Enable signals for each byte of the data bus. These hardware features are described fully beginning in Section 5.

2.2 REGISTER OVERVIEW

The 80386 has 32 register resources in the following categories:

- General Purpose Registers
- Segment Registers
- Instruction Pointer and Flags
- Control Registers
- System Address Registers
- Debug Registers
- Test Registers.

The registers are a superset of the 8086, 80186 and 80286 registers, so all 16-bit 8086, 80186 and 80286 registers are contained within the 32-bit 80386.

Figure 2-1 shows all of 80386 base architecture registers, which include the general address and data registers, the instruction pointer, and the flags register. The contents of these registers are task-specific, so these registers are automatically loaded with a new context upon a task switch operation.

The base architecture also includes six directly accessible segments, each up to 4 Gbytes in size. The segments are indicated by the selector values placed in 80386 segment registers of Figure 2-1. Various selector values can be loaded as a program executes, if desired.

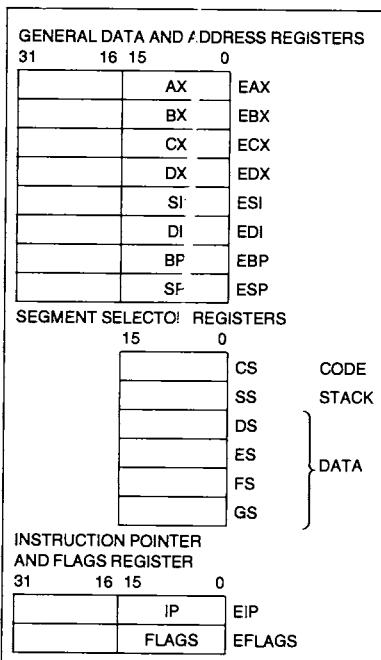


Figure 2-1. 80386 Base Architecture Registers

The selectors are also task-specific, so the segment registers are automatically loaded with new context upon a task switch operation.

The other types of registers, Control, System Address, Debug, and Test, are primarily used by system software.

2.3 REGISTER DESCRIPTIONS

2.3.1 General Purpose Registers

General Purpose Registers: The eight general purpose registers of 32 bits hold data or address quantities. The general registers, Figure 2-2, support data operands of 1, 8, 16, 32 and 64 bits, and bit fields of 1 to 32 bits. They support address operands of 16 and 32 bits. The 32-bit registers are named EAX, EBX, ECX, EDX, ESI, EDI, EBP, and ESP.

The least significant 16 bits of the registers can be accessed separately. This is done by using the 16-bit names of the registers AX, BX, CX, DX, SI, DI, BP, and SP.

Finally 8-bit operations can individually access the lowest byte (bits 0–7) and the higher byte (bits 8–15) of general purpose registers AX, BX, CX and DX. The lowest bytes are named AL, BL, CL and DL, respectively. The higher bytes are named AH, BH, CH and DH, respectively. The individual byte accessibility offers additional flexibility for data operations, but is not used for effective address calculation.

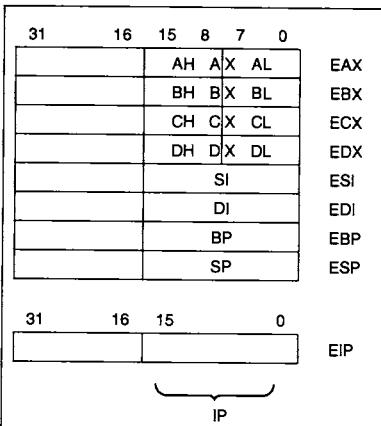


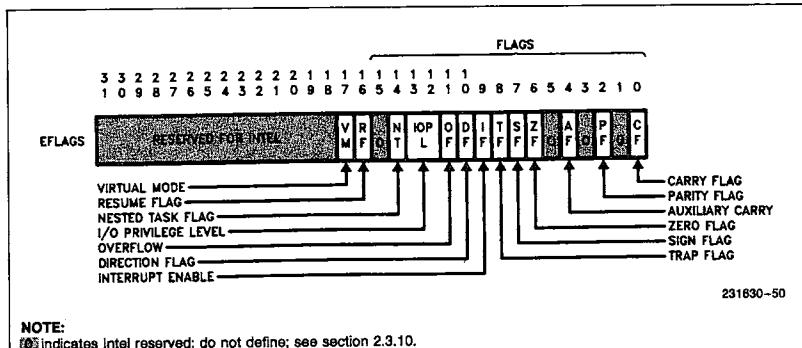
Figure 2-2. General Registers and Instruction Pointer

2.3.2 Instruction Pointer

The instruction pointer, Figure 2-2, is a 32-bit register named EIP. EIP holds the offset of the next instruction to be executed. The offset is always relative to the base of the code segment (CS). The lower 16 bits (bits 0–15) of EIP contain the 16-bit instruction pointer named IP, which is used by 16-bit addressing.

2.3.3 Flags Register

The Flags Register is a 32-bit register named EFLAGS. The defined bits and bit fields within EFLAGS, shown in Figure 2-3, control certain operations and indicate status of the 80386. The lower 16 bits (bit 0–15) of EFLAGS contain the 16-bit flag register named FLAGS, which is most useful when executing 8086 and 80286 code.

**NOTE:**

indicates Intel reserved: do not define; see section 2.3.10.

Figure 2-3. Flags Register

VM (Virtual 8086 Mode, bit 17)

The VM bit provides Virtual 8086 Mode within Protected Mode. If set while the 80386 is in Protected Mode, the 80386 will switch to Virtual 8086 operation, handling segment loads as the 8086 does, but generating exception 13 faults on privileged opcodes. The VM bit can be set only in Protected Mode, by the IRET instruction (if current privilege level = 0) and by task switches at any privilege level. The VM bit is unaffected by POPF. PUSHF always pushes a 0 in this bit, even if executing in virtual 8086 Mode. The EFLAGS image pushed during interrupt processing or saved during task switches will contain a 1 in this bit if the interrupted code was executing as a Virtual 8086 Task.

RF (Resume Flag, bit 16)

The RF flag is used in conjunction with the debug register breakpoints. It is checked at instruction boundaries before breakpoint processing. When RF is set, it causes any debug fault to be ignored on the next instruction. RF is then automatically reset at the successful completion of every instruction (no faults are signalled) except the IRET instruction, the POPF instruction, (and JMP, CALL, and INT instructions causing a task switch). These instructions set RF to the value specified by the memory image. For example, at the end of the breakpoint service routine, the IRET

instruction can pop an EFLAG image having the RF bit set and resume the program's execution at the breakpoint address without generating another breakpoint fault on the same location.

NT

(Nested Task, bit 14)

This flag applies to Protected Mode. NT is set to indicate that the execution of this task is nested within another task. If set, it indicates that the current nested task's Task State Segment (TSS) has valid back link to the previous task's TSS. This bit is set or reset by control transfers to other tasks. The value of NT in EFLAGS is tested by the IRET instruction to determine whether to do an inter-task return or an intra-task return. A POPF or an IRET instruction will affect the setting of this bit according to the image popped, at any privilege level.

IOPL

(Input/Output Privilege Level, bits 12-13)

This two-bit field applies to Protected Mode. IOPL indicates the numerically maximum CPL (current privilege level) value permitted to execute I/O instructions without generating an exception 13 fault or consulting the I/O Permission Bitmap. It also indicates the maximum CPL value allowing alteration of the IF (INTR Enable Flag) bit when new values are popped into the EFLAG register. POPF and IRET instruction can alter the IOPL field when executed at CPL = 0. Task switches can always alter the IOPL field, when the new flag image is loaded from the incoming task's TSS.

OF	(Overflow Flag, bit 11)	ZF	(Zero Flag, bit 6)
	OF is set if the operation resulted in a signed overflow. Signed overflow occurs when the operation resulted in carry/borrow into the sign bit (high-order bit) of the result but did not result in a carry/borrow out of the high-order bit, or vice-versa. For 8/16/32 bit operations, OF is set according to overflow at bit 7/15/31, respectively.		ZF is set if all bits of the result are 0. Otherwise it is reset.
DF	(Direction Flag, bit 10)	AF	(Auxiliary Carry Flag, bit 4)
	DF defines whether ESI and/or EDI registers postdecrement or postincrement during the string instructions. Postincrement occurs if DF is reset. Postdecrement occurs if DF is set.		The Auxiliary Flag is used to simplify the addition and subtraction of packed BCD quantities. AF is set if the operation resulted in a carry out of bit 3 (addition) or a borrow into bit 3 (subtraction). Otherwise AF is reset. AF is affected by carry out of, or borrow into bit 3 only, regardless of overall operand length: 8, 16 or 32 bits.
IF	(INTR Enable Flag, bit 9)	PF	(Parity Flags, bit 2)
	The IF flag, when set, allows recognition of external interrupts signalled on the INTR pin. When IF is reset, external interrupts signalled on the INTR are not recognized. IOPL indicates the maximum CPL value allowing alteration of the IF bit when new values are popped into EFLAGS or FLAGS.		PF is set if the low-order eight bits of the operation contains an even number of "1's" (even parity). PF is reset if the low-order eight bits have odd parity. PF is a function of only the low-order eight bits, regardless of operand size.
TF	(Trap Enable Flag, bit 8)	CF	(Carry Flag, bit 0)
	TF controls the generation of exception 1 trap when single-stepping through code. When TF is set, the 80386 generates an exception 1 trap after the next instruction is executed. When TF is reset, exception 1 traps occur only as a function of the breakpoint addresses loaded into debug registers DR0-DR3.		CF is set if the operation resulted in a carry out of (addition), or a borrow into (subtraction) the high-order bit. Otherwise CF is reset. For 8-, 16- or 32-bit operations, CF is set according to carry/borrow at bit 7, 15 or 31, respectively.
SF	(Sign Flag, bit 7)		
	SF is set if the high-order bit of the result is set, it is reset otherwise. For 8-, 16-, 32-bit operations, SF reflects the state of bit 7, 15, 31 respectively.		Note in these descriptions, "set" means "set to 1," and "reset" means "reset to 0."

2.3.4 Segment Registers

Six 16-bit segment registers hold segment selector values identifying the currently addressable memory segments. Segment registers are shown in Figure 2-4. In Protected Mode, each segment may range in size from one byte up to the entire linear and physi-

SEGMENT REGISTERS		DESCRIPTOR REGISTERS (LOADED AUTOMATICALLY)					
15	0						
Selector	CS-	Physical Base Address	Segment Limit				Other Segment Attributes from Descriptor
Selector	SS-						— —
Selector	DS-						— — —
Selector	ES-						— — — —
Selector	FS-						— — — —
Selector	GS-						— — — —

Figure 2-4. 80386 Segment Registers, and Associated Descriptor Registers

cal space of the machine, 4 Gbytes (2^{32} bytes). In Real Address Mode, the maximum segment size is fixed at 64 Kbytes (2^{16} bytes).

The six segments addressable at any given moment are defined by the segment registers CS, SS, DS, ES, FS and GS. The selector in CS indicates the current code segment; the selector in SS indicates the current stack segment; the selectors in DS, ES, FS and GS indicate the current data segments.

2.3.5 Segment Descriptor Registers

The segment descriptor registers are not programmer visible, yet it is very useful to understand their content. Inside the 80386, a descriptor register (programmer invisible) is associated with each programmer-visible segment register, as shown by Figure 2-4. Each descriptor register holds a 32-bit segment base address, a 32-bit segment limit, and the other necessary segment attributes.

When a selector value is loaded into a segment register, the associated descriptor register is automatically updated with the correct information. In Real Address Mode, only the base address is updated directly (by shifting the selector value four bits to the left), since the segment maximum limit and attributes are fixed in Real Mode. In Protected Mode, the base address, the limit, and the attributes are all updated per the contents of the segment descriptor indexed by the selector.

Whenever a memory reference occurs, the segment descriptor register associated with the segment being used is automatically involved with the memory reference. The 32-bit segment base address becomes a component of the linear address calculation, the 32-bit limit is used for the limit-check operation, and the attributes are checked against the type of memory reference requested.

2.3.6 Control Registers

The 80386 has three control registers of 32 bits, CR0, CR2 and CR3, to hold machine state of a global nature (not specific to an individual task). These registers, along with System Address Registers described in the next section, hold machine state that affects all tasks in the system. To access the Control Registers, load and store instructions are defined.

CR0: Machine Control Register (Includes 80286 Machine Status Word)

CR0, shown in Figure 2-5, contains 6 defined bits for control and status purposes. The low-order 16 bits of CR0 are also known as the Machine Status Word, MSW, for compatibility with 80286 Protected Mode. LMSW and SMSW instructions are taken as special aliases of the load and store CR0 operations, where only the low-order 16 bits of CR0 are involved. For compatibility with 80286 operating systems the 80386's LMSW instructions work in an identical fashion to the LMSW instruction on the 80286. (i.e. It only operates on the low-order 16-bits of CR0 and it ignores the new bits in CR0.) New 80386 operating systems should use the MOV CR0, Reg instruction.

The defined CR0 bits are described below.

PG (Paging Enable, bit 31)

the PG bit is set to enable the on-chip paging unit. It is reset to disable the on-chip paging unit.

ET (Processor Extension Type, bit 4)

ET indicates the processor extension type (either 80287 or 80387) as detected by the level of the ERROR# input following 80386 reset. The ET bit may also be set or reset by loading CR0 under program control if desired. If ET is set, the 80387-compatible 32-bit protocol is used. If ET is reset, 80287-compatible 16-bit protocol is used.

Note that for strict 80286 compatibility, ET is not affected by the LMSW instruction. When the MSW or CR0 is stored, bit 4 accurately reflects the current state of the ET bit.

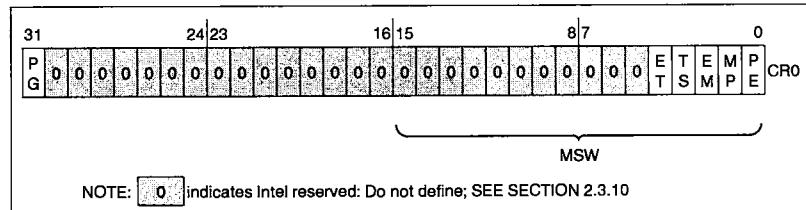


Figure 2-5. Control Register 0

TS (Task Switched, bit 3)

TS is automatically set whenever a task switch operation is performed. If TS is set, a coprocessor ESCape opcode will cause a Coprocessor Not Available trap (exception 7). The trap handler typically saves the 80287/80387 context belonging to a previous task, loads the 80287/80387 state belonging to the current task, and clears the TS bit before returning to the faulting coprocessor opcode.

EM (Emulate Coprocessor, bit 2)

The EMulate coprocessor bit is set to cause all coprocessor opcodes to generate a Coprocessor Not Available fault (exception 7). It is reset to allow coprocessor opcodes to be executed on an actual 80287 or 80387 coprocessor (this the default case after reset). Note that the WAIT opcode is not affected by the EM bit setting.

MP (Monitor Coprocessor, bit 1)

The MP bit is used in conjunction with the TS bit to determine if the WAIT opcode will generate a Coprocessor Not Available fault (exception 7) when TS = 1. When both MP = 1 and TS = 1, the WAIT opcode generates a trap. Otherwise, the WAIT opcode does not generate a trap. Note that TS is automatically set whenever a task switch operation is performed.

PE (Protection Enable, bit 0)

The PE bit is set to enable the Protected Mode. If PE is reset, the processor operates again in Real Mode. PE may be set by loading MSW or CR0. PE can be reset only by a load into CR0. Resetting the PE bit is typically part of a longer instruction sequence needed for proper transition from Protected Mode to Real Mode. Note that for strict 80286 compatibility, PE cannot be reset by the LMSW instruction.

CR1: reserved

CR1 is reserved for use in future Intel processors.

CR2: Page Fault Linear Address

CR2, shown in Figure 2-6, holds the 32-bit linear address that caused the last page fault detected. The

error code pushed onto the page fault handler's stack when it is invoked provides additional status information on this page fault.

CR3: Page Directory Base Address

CR3, shown in Figure 2-6, contains the physical base address of the page directory table. The 80386 page directory table is always page-aligned (4 Kbyte-aligned). Therefore the lowest twelve bits of CR3 are ignored when written and they store as undefined.

A task switch through a TSS which changes the value in CR3, or an explicit load into CR3 with any value, will invalidate all cached page table entries in the paging unit cache. Note that if the value in CR3 does not change during the task switch, the cached page table entries are not flushed.

2.3.7 System Address Registers

Four special registers are defined to reference the tables or segments supported by the 80286/80386 protection model. These tables or segments are:

GDT (Global Descriptor Table),

IDT (Interrupt Descriptor Table),

LDT (Local Descriptor Table),

TSS (Task State Segment).

The addresses of these tables and segments are stored in special registers, the System Address and System Segment Registers illustrated in Figure 2-7. These registers are named GDTR, IDTR, LDTR and TR, respectively. Section 4 **Protected Mode Architecture** describes the use of these registers.

GDTR and IDTR

These registers hold the 32-bit linear base address and 16-bit limit of the GDT and IDT, respectively.

The GDT and IDT segments, since they are global to all tasks in the system, are defined by 32-bit linear addresses (subject to page translation if paging is enabled) and 16-bit limit values.

31	24 23	16 15	8 7	C
PAGE FAULT LINEAR ADDRESS REGISTER				
PAGE DIRECTORY BASE REGISTER				CR2
NOTE:  indicates Intel reserved: Do not define; SEE SECTION 2.3.10				

Figure 2-6. Control Registers 2 and 3

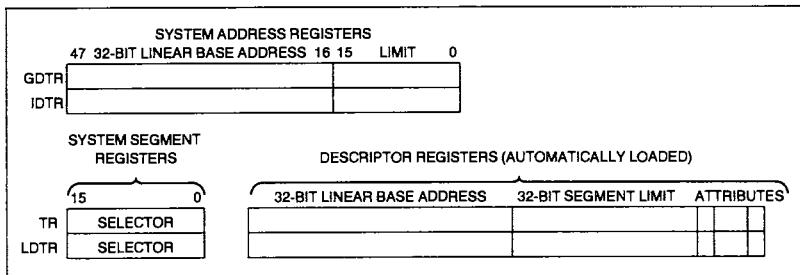


Figure 2-7. System Address and System Segment Registers

LDTR and TR

These registers hold the 16-bit selector for the LDT descriptor and the TSS descriptor, respectively.

The LDT and TSS segments, since they are task-specific segments, are defined by selector values stored in the system segment registers. Note that a segment descriptor register (programmer-invisible) is associated with each system segment register.

2.3.8 Debug and Test Registers

Debug Registers: The six programmer accessible debug registers provide on-chip support for debugging. Debug Registers DR0–3 specify the four linear breakpoints. The Debug Control Register DR7 is used to set the breakpoints and the Debug Status Register DR6, displays the current state of the breakpoints. The use of the debug registers is described in section 2.12 **Debugging support**.

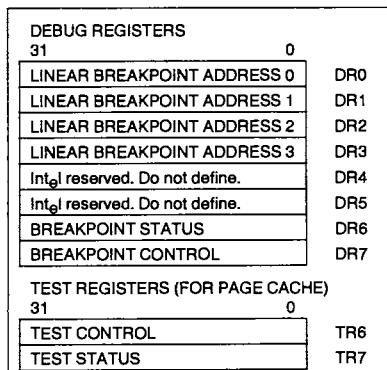


Figure 2-8. Debug and Test Registers

Test Registers: Two registers are used to control the testing of the RAM/CAM (Content Addressable Memories) in the Translation Lookaside Buffer portion of the 80386. TR6 is the command test register, and TR7 is the data register which contains the data of the Translation Lookaside buffer test. Their use is discussed in section 2.11 **Testability**.

Figure 2-8 shows the Debug and Test registers.

2.3.9 Register Accessibility

There are a few differences regarding the accessibility of the registers in Real and Protected Mode. Table 2-1 summarizes these differences. See Section 4 **Protected Mode Architecture** for further details.

2.3.10 Compatibility

VERY IMPORTANT NOTE: COMPATIBILITY WITH FUTURE PROCESSORS

In the preceding register descriptions, note certain 80386 register bits are Intel reserved. When reserved bits are called out, treat them as fully undefined. This is essential for your software compatibility with future processors! Follow the guidelines below:

- 1) Do not depend on the states of any undefined bits when testing the values of defined register bits. Mask them out when testing.
- 2) Do not depend on the states of any undefined bits when storing them to memory or another register.
- 3) Do not depend on the ability to retain information written into any undefined bits.
- 4) When loading registers always load the undefined bits as zeros.



Table 2-1. Register Usage

Register	Use in Real Mode		Use in Protected Mode		Use in Virtual 8086 Mode	
	Load	Store	Load	Store	Load	Store
General Registers	Yes	Yes	Yes	Yes	Yes	Yes
Segment Registers	Yes	Yes	Yes	Yes	Yes	Yes
Flag Register	Yes	Yes	Yes	Yes	IOPL	IOPL*
Control Registers	Yes	Yes	PL = 0	PL = 0	No	Yes
GDTR	Yes	Yes	PL = 0	Yes	No	Yes
IDTR	Yes	Yes	PL = 0	Yes	No	Yes
LDTR	No	No	PL = 0	Yes	No	No
TR	No	No	PL = 0	Yes	No	No
Debug Control	Yes	Yes	PL = 0	PL = 0	No	No
Test Registers	Yes	Yes	PL = 0	PL = 0	No	No

NOTES:

PL = 0: The registers can be accessed only when the current privilege level is zero.

*IOPL: The PUSHF and POPF instructions are made I/O Privilege Level sensitive in Virtual 8086 Mode.

- 5) However, registers which have been previously stored may be reloaded without masking.

Depending upon the values of undefined register bits will make your software dependent upon the unspecified 80386 handling of these bits. Depending on undefined values risks making your software incompatible with future processors that define usages for the 80386-undefined bits. AVOID ANY SOFTWARE DEPENDENCE UPON THE STATE OF UNDEFINED 80386 REGISTER BITS.

2.4 INSTRUCTION SET

2.4.1 Instruction Set Overview

The instruction set is divided into nine categories of operations:

- Data Transfer
- Arithmetic
- Shift/Rotate
- String Manipulation
- Bit Manipulation
- Control Transfer
- High Level Language Support
- Operating System Support
- Processor Control

These 80386 instructions are listed in Table 2-2.

All 80386 instructions operate on either 0, 1, 2, or 3 operands; where an operand resides in a register, in the instruction itself, or in memory. Most zero operand instructions (e.g. CLI, STI) take only one byte. One operand instructions generally are two bytes long. The average instruction is 3.2 bytes long. Since the 80386 has a 16-byte instruction queue, an average of 5 instructions will be prefetched. The use of two operands permits the following types of common instructions:

- Register to Register
- Memory to Register
- Immediate to Register
- Register to Memory
- Immediate to Memory.

The operands can be either 8, 16, or 32 bits long. As a general rule, when executing code written for the 80386 (32-bit code), operands are 8 or 32 bits; when executing existing 80286 or 8086 code (16-bit code), operands are 8 or 16 bits. Prefixes can be added to all instructions which override the default length of the operands, (i.e. use 32-bit operands for 16-bit code, or 16-bit operands for 32-bit code).

2.4.2 80386 Instructions

Table 2-2a. Data Transfer

GENERAL PURPOSE	
MOV	Move operand
PUSH	Push operand onto stack
POP	Pop operand off stack
PUSHA	Push all registers on stack
POPA	Pop all registers off stack
XCHG	Exchange Operand, Register
XLAT	Translate
CONVERSION	
MOVZX	Move byte or Word, Dword, with zero extension
MOVSX	Move byte or Word, Dword, sign extended
CBW	Convert byte to Word, or Word to Dword
CWD	Convert Word to DWORD
CWDE	Convert Word to DWORD extended
CDQ	Convert DWORD to QWORD
INPUT/OUTPUT	
IN	Input operand from I/O space
OUT	Output operand to I/O space
ADDRESS OBJECT	
LEA	Load effective address
LDS	Load pointer into D segment register
LES	Load pointer into E segment register
LFS	Load pointer into F segment register
LGS	Load pointer into G segment register
LSS	Load pointer into S (Stack) segment register
FLAG MANIPULATION	
LAHF	Load A register from Flags
SAHF	Store A register in Flags
PUSHF	Push flags onto stack
POPF	Pop flags off stack
PUSHFD	Push EFlags onto stack
POPFD	Pop EFlags off stack
CLC	Clear Carry Flag
CLD	Clear Direction Flag
CMC	Complement Carry Flag
STC	Set Carry Flag
STD	Set Direction Flag

Table 2-2b. Arithmetic Instructions

ADDITION	
ADD	Add operands
ADC	Add with carry
INC	Increment operand by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition
SUBTRACTION	
SUB	Subtract operands
SBB	Subtract with borrow
DEC	Decrement operand by 1
NEG	Negate operand
CMP	Compare operands
DAS	Decimal adjust for subtraction
AAS	ASCII Adjust for subtraction
MULTIPLICATION	
MUL	Multiply Double/Single Precision
IMUL	Integer multiply
AAM	ASCII adjust after multiply
DIVISION	
DIV	Divide unsigned
IDIV	Integer Divide
AAD	ASCII adjust before division

Table 2-2c. String Instructions

MOVS	Move byte or Word, Dword string
INS	Input string from I/O space
OUTS	Output string to I/O space
CMPS	Compare byte or Word, Dword string
SCAS	Scan Byte or Word, Dword string
LODS	Load byte or Word, Dword string
STOS	Store byte or Word, Dword string
REP	Repeat
REPE/REPZ	Repeat while equal/zero
RENE/REPNZ	Repeat while not equal/not zero

Table 2-2d. Logical Instructions

LOGICALS	
NOT	"NOT" operands
AND	"AND" operands
OR	"Inclusive OR" operands
XOR	"Exclusive OR" operands
TEST	"Test" operands

Table 2-2d. Logical Instructions (Continued)

SHIFTS	
SHL/SHR	Shift logical left or right
SAL/SAR	Shift arithmetic left or right
SHLD/ SHRD	Double shift left or right
ROTATES	
ROL/ROR	Rotate left/right
RCL/RCR	Rotate through carry left/right

Table 2-2e. Bit Manipulation Instructions

SINGLE BIT INSTRUCTIONS	
BT	Bit Test
BTS	Bit Test and Set
BTR	Bit Test and Reset
BTC	Bit Test and Complement
BSF	Bit Scan Forward
BSR	Bit Scan Reverse
BIT STRING INSTRUCTIONS	
IBTS	Insert Bit String
XBTS	Exact Bit String

Table 2-2f. Program Control Instructions

CONDITIONAL TRANSFERS	
SETCC	Set byte equal to condition code
JA/JNBE	Jump if above/not below nor equal
JAE/JNB	Jump if above or equal/not below
JB/JNAE	Jump if below/not above nor equal
JBE/JNA	Jump if below or equal/not above
JC	Jump if carry
JE/JZ	Jump if equal/zero
JG/JNLE	Jump if greater/not less nor equal
JGE/JNL	Jump if greater or equal/not less
JL/JNGE	Jump if less/not greater nor equal
JLE/JNG	Jump if less or equal/not greater
JNC	Jump if not carry
JNE/JNZ	Jump if not equal/not zero
JNO	Jump if not overflow
JNP/JPO	Jump if not parity/parity odd
JNS	Jump if not sign
JO	Jump if overflow
JP/JPE	Jump if parity/parity even
JS	Jump if Sign

Table 2-2f. Program Control Instructions (Continued)

UNCONDITIONAL TRANSFERS	
CALL	Call procedure/task
RET	Return from procedure
JMP	Jump
ITERATION CONTROLS	
LOOP	Loop
LOOPE/ LOOPZ	Loop if equal/zero
LOOPNE/ LOOPNZ	Loop if not equal/not zero
JCXZ	JUMP if register CX = 0
INTERRUPTS	
INT	Interrupt
INTO	Interrupt if overflow
IRET	Return from Interrupt/Task
CLI	Clear interrupt Enable
SLI	Set interrupt Enable

Table 2-2g. High Level Language Instructions

BOUND	Check Array Bounds
ENTER	Setup Parameter Block for Entering Procedure
LEAVE	Leave Procedure

Table 2-2h. Protection Model

SGDT	Store Global Descriptor Table
SIDT	Store Interrupt Descriptor Table
STR	Store Task Register
SLDT	Store Local Descriptor Table
LGDT	Load Global Descriptor Table
LIDT	Load Interrupt Descriptor Table
LTR	Load Task Register
LLDT	Load Local Descriptor Table
ARPL	Adjust Requested Privilege Level
LAR	Load Access Rights
LSL	Load Segment Limit
VERR/ VERW	Verify Segment for Reading or Writing
LMSW	Load Machine Status Word (lower 16 bits of CR0)
SMSW	Store Machine Status Word

Table 2-2i. Processor Control Instructions

HLT	Halt
WAIT	Wait until BUSY# negated
ESC	Escape
LOCK	Lock Bus

2.5 ADDRESSING MODES

2.5.1 Addressing Modes Overview

The 80386 provides a total of 11 addressing modes for instructions to specify operands. The addressing modes are optimized to allow the efficient execution of high level languages such as C and FORTRAN, and they cover the vast majority of data references needed by high-level languages.

2.5.2 Register and Immediate Modes

Two of the addressing modes provide for instructions that operate on register or immediate operands:

Register Operand Mode: The operand is located in one of the 8-, 16- or 32-bit general registers.

Immediate Operand Mode: The operand is included in the instruction as part of the opcode.

2.5.3 32-Bit Memory Addressing Modes

The remaining 9 modes provide a mechanism for specifying the effective address of an operand. The linear address consists of two components: the segment base address and an effective address. The effective address is calculated by using combinations of the following four address elements:

DISPLACEMENT: An 8-, or 32-bit immediate value, following the instruction.

BASE: The contents of any general purpose register. The base registers are generally used by compilers to point to the start of the local variable area.

INDEX: The contents of any general purpose register except for ESP. The index registers are used to access the elements of an array, or a string of characters.

SCALE: The index register's value can be multiplied by a scale factor, either 1, 2, 4 or 8. Scaled index mode is especially useful for accessing arrays or structures.

Combinations of these 4 components make up the 9 additional addressing modes. There is no performance penalty for using any of these addressing combinations, since the effective address calculation is pipelined with the execution of other instructions.

The one exception is the simultaneous use of Base and Index components which requires one additional clock.

As shown in Figure 2-9, the effective address (EA) of an operand is calculated according to the following formula.

$$\text{EA} = \text{Base Reg} + (\text{Index Reg} * \text{Scaling}) + \text{Displacement}$$

Direct Mode: The operand's offset is contained as part of the instruction as an 8-, 16- or 32-bit displacement.

EXAMPLE: INC Word PTR [500]

Register Indirect Mode: A BASE register contains the address of the operand.

EXAMPLE: MOV [ECX], EDX

Based Mode: A BASE register's contents is added to a DISPLACEMENT to form the operands offset.

EXAMPLE: MOV ECX, [EAX + 24]

Index Mode: An INDEX register's contents is added to a DISPLACEMENT to form the operands offset.

EXAMPLE: ADD EAX, TABLE[ESI]

Scaled Index Mode: An INDEX register's contents is multiplied by a scaling factor which is added to a DISPLACEMENT to form the operands offset.

EXAMPLE: IMUL EBX, TABLE[ESI*4], 7

Based Index Mode: The contents of a BASE register is added to the contents of an INDEX register to form the effective address of an operand.

EXAMPLE: MOV EAX, [ESI] [EBX]

Based Scaled Index Mode: The contents of an INDEX register is multiplied by a SCALING factor and the result is added to the contents of a BASE register to obtain the operands offset.

EXAMPLE: MOV ECX, [EDX*8] [EAX]

Based Index Mode with Displacement: The contents of an INDEX Register and a BASE register's contents and a DISPLACEMENT are all summed together to form the operand offset.

EXAMPLE: ADD EDX, [ESI] [EBP + 00FFFFFF0H]

Based Scaled Index Mode with Displacement: The contents of an INDEX register are multiplied by a SCALING factor, the result is added to the contents of a BASE register and a DISPLACEMENT to form the operand's offset.

EXAMPLE: MOV EAX, LOCALTABLE[EDI*4] [EBP + 80]

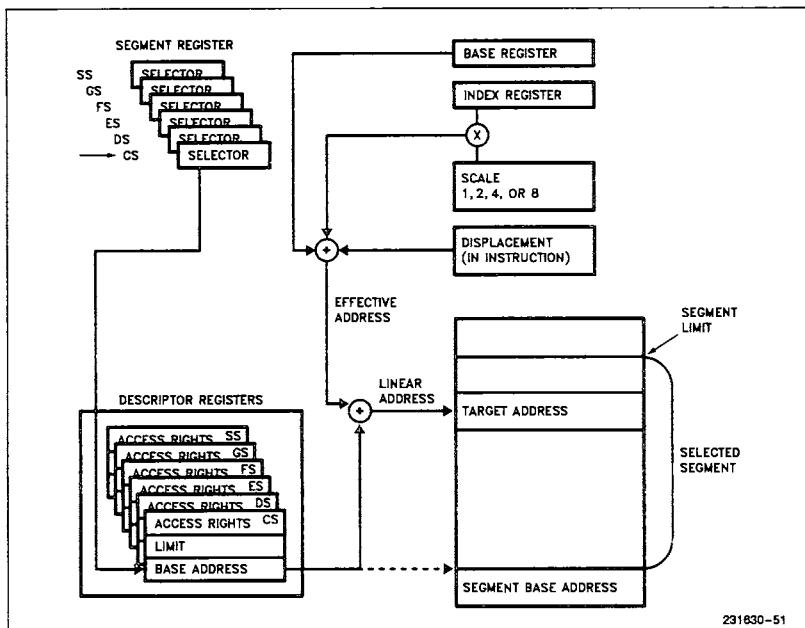


Figure 2-9. Addressing Mode Calculations

231630-51

2.5.4 Differences Between 16 and 32 Bit Addresses

In order to provide software compatibility with the 80286 and the 8086, the 80386 can execute 16-bit instructions in Real and Protected Modes. The processor determines the size of the instructions it is executing by examining the D bit in the CS segment Descriptor. If the D bit is 0 then all operand lengths and effective addresses are assumed to be 16 bits long. If the D bit is 1 then the default length for operands and addresses is 32 bits. In Real Mode the default size for operands and addresses is 16-bits.

Regardless of the default precision of the operands or addresses, the 80386 is able to execute either 16 or 32-bit instructions. This is specified via the use of override prefixes. Two prefixes, the **Operand Size Prefix** and the **Address Length Prefix**, override the value of the D bit on an individual instruction basis. These prefixes are automatically added by Intel assemblers.

Example: The processor is executing in Real Mode and the programmer needs to access the EAX registers. The assembler code for this might be MOV EAX, 32bitMEMORYOP. ASM 386 automatically determines that an Operand Size Prefix is needed and generates it.

Example: The D bit is 0, and the programmer wishes to use Scaled Index addressing mode to access an array. The Address Length Prefix allows the use of MOV DX, TABLE[ESI*2]. The assembler uses an Address Length Prefix since, with D=0, the default addressing mode is 16-bits.

Example: The D bit is 1, and the program wants to store a 16-bit quantity. The Operand Length Prefix is used to specify only a 16-bit value; MOV MEM16, DX.

Table 2-3. BASE and INDEX Registers for 16- and 32-Bit Addresses

	16-Bit Addressing	32-Bit Addressing
BASE REGISTER INDEX REGISTER	BX,BP SI,DI	Any 32-bit GP Register Any 32-bit GP Register Except ESP 1, 2, 4, 8 0, 8, 32 bits
SCALE FACTOR DISPLACEMENT	none 0, 8, 16 bits	

The OPERAND LENGTH and Address Length Prefixes can be applied separately or in combination to any instruction. The Address Length Prefix does not allow addresses over 64K bytes to be accessed in Real Mode. A memory address which exceeds FFFFH will result in a General Protection Fault. An Address Length Prefix only allows the use of the additional 80386 addressing modes.

When executing 32-bit code, the 80386 uses either 8-, or 32-bit displacements, and any register can be used as base or index registers. When executing 16-bit code, the displacements are either 8, or 16 bits, and the base and index register conform to the 286 model. Table 2-3 illustrates the differences.

2.6 DATA TYPES

The 80386 supports all of the data types commonly used in high level languages:

Bit: A single bit quantity.

Bit Field: A group of up to 32 contiguous bits, which spans a maximum of four bytes.

Bit String: A set of contiguous bits, on the 80386 bit strings can be up to 4 gigabits long.

Byte: A signed 8-bit quantity.

Unsigned Byte: An unsigned 8-bit quantity.

Integer (Word): A signed 16-bit quantity.

Long Integer (Double Word): A signed 32-bit quantity. All operations assume a 2's complement representation.

Unsigned Integer (Word): An unsigned 16-bit quantity.

Unsigned Long Integer (Double Word): An unsigned 32-bit quantity.

Signed Quad Word: A signed 64-bit quantity.

Unsigned Quad Word: An unsigned 64-bit quantity.

Offset: A 16- or 32-bit offset only quantity which indirectly references another memory location.

Pointer: A full pointer which consists of a 16-bit segment selector and either a 16- or 32-bit offset.

Char: A byte representation of an ASCII Alphanumeric or control character.

String: A contiguous sequence of bytes, words or dwords. A string may contain between 1 byte and 4 Gbytes.

BCD: A byte (unpacked) representation of decimal digits 0-9.

Packed BCD: A byte (packed) representation of two decimal digits 0-9 storing one digit in each nibble.

When the 80386 is coupled with a numerics Coprocessor such as the 80287 or the 80387 then the following common Floating Point types are supported.

Floating Point: A signed 32-, 64-, or 80-bit real number representation. Floating point numbers are supported by the 80287 and 80387 numerics coprocessor.

Figure 2-10 illustrates the data types supported by the 80386 and the 80387/80287.

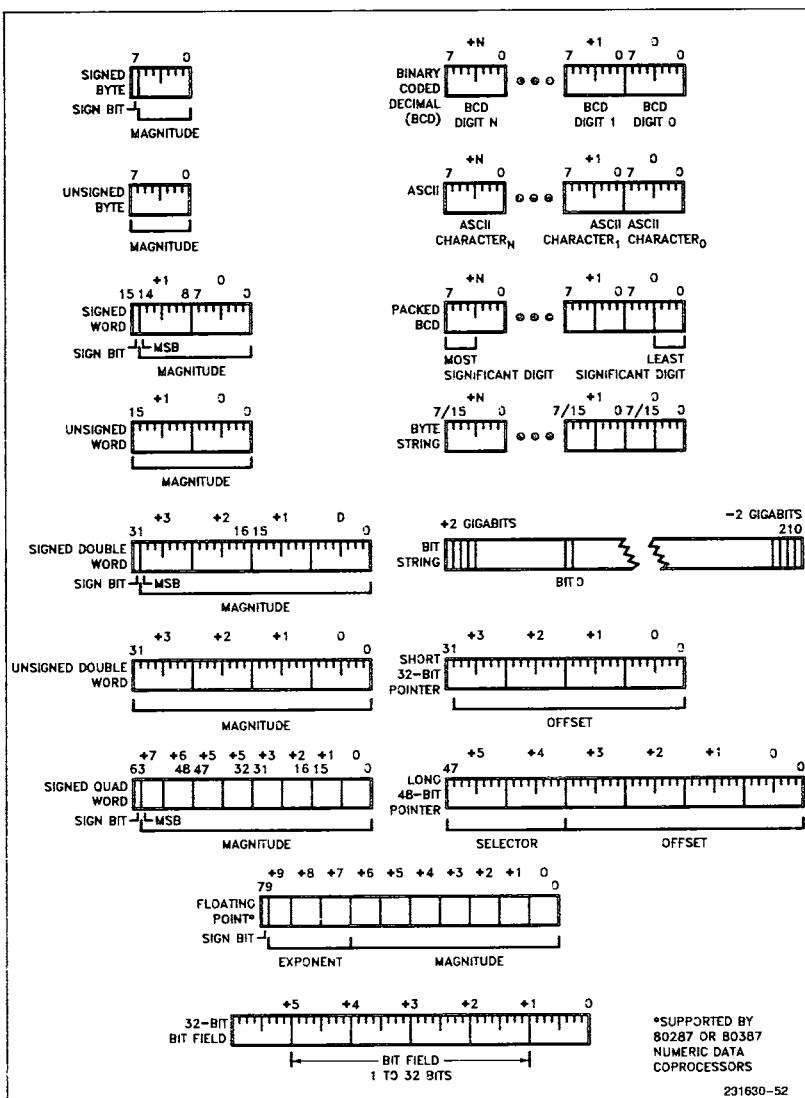


Figure 2-10. 80386 Supported Data Types

2.7 MEMORY ORGANIZATION

2.7.1 Introduction

Memory on the 80386 is divided up into 8-bit quantities (bytes), 16-bit quantities (words), and 32-bit quantities (dwords). Words are stored in two consecutive bytes in memory with the low-order byte at the lowest address, the high-order byte at the highest address. Dwords are stored in four consecutive bytes in memory with the low-order byte at the lowest address, the high-order byte at the highest address. The address of a word or dword is the byte address of the low-order byte.

In addition to these basic data types the 386 supports two larger units of memory: pages and segments. Memory can be divided up into one or more variable length segments, which can be swapped to disk or shared between programs. Memory can also be organized into one or more 4K byte pages. Finally, both segmentation and paging can be combined, gaining the advantages of both systems. The 80386 supports both pages and segments in order to provide maximum flexibility to the system designer. Segmentation and paging are complementary. Segmentation is useful for organizing memory in logical modules, and as such is a tool for the application programmer, while pages are useful for the system programmer for managing the physical memory of a system.

2.7.2 Address Spaces

The 80386 has three distinct address spaces: **logical**, **linear**, and **physical**. A **logical address**

(also known as a **virtual address**) consists of a selector and an offset. A selector is the contents of a segment register. An offset is formed by summing all of the addressing components (BASE, INDEX, DISPLACEMENT) discussed in section 2.5.3 **Memory Addressing Modes** into an effective address. Since each task on 80386 has a maximum of 16K ($2^{14} - 1$) selectors, and offsets can be 4 gigabytes, (2^{32} bits) this gives a total of 2^{46} bits or 64 terabytes of **logical address space per task**. The programmer sees this virtual address space.

The segmentation unit translates the **logical address** space into a 32-bit **linear address** space. If the paging unit is not enabled then the 32-bit **linear address** corresponds to the **physical address**. The paging unit translates the **linear address space** into the **physical address space**. The **physical address** is what appears on the address pins.

The primary difference between Real Mode and Protected Mode is how the segmentation unit performs the translation of the **logical address** into the **linear address**. In Real Mode, the segmentation unit shifts the selector left four bits and adds the result to the offset to form the **linear address**. While in Protected Mode every selector has a **linear base address** associated with it. The **linear base address** is stored in one of two operating system tables (i.e. the Local Descriptor Table or Global Descriptor Table). The selector's **linear base address** is added to the offset to form the final **linear address**.

Figure 2-11 shows the relationship between the various address spaces.

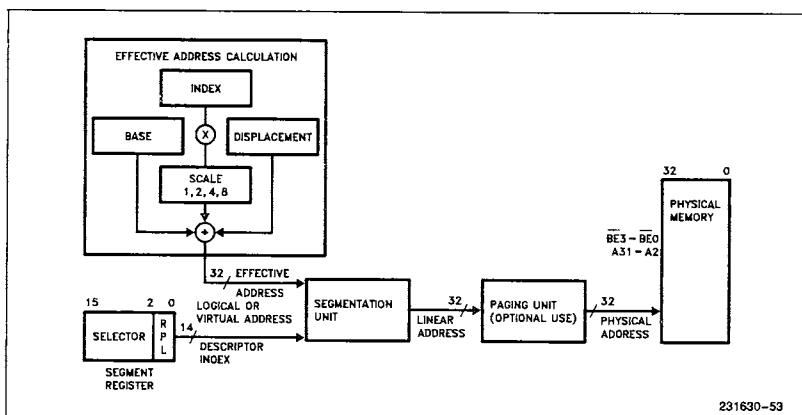


Figure 2-11. Address Translation

2.7.3 Segment Register Usage

The main data structure used to organize memory is the segment. On the 386, segments are variable sized blocks of linear addresses which have certain attributes associated with them. There are two main types of segments: code and data, the segments are of variable size and can be as small as 1 byte or as large as 4 gigabytes (2³² bytes).

In order to provide compact instruction encoding, and increase processor performance, instructions do not need to explicitly specify which segment register is used. A default segment register is automatically chosen according to the rules of Table 2-4 (Segment Register Selection Rules). In general, data references use the selector contained in the DS register; Stack references use the SS register and Instruction fetches use the CS register. The contents of the Instruction Pointer provides the offset. Special segment override prefixes allow the explicit use of a given segment register, and override the implicit rules listed in Table 2-4. The override prefixes also allow the use of the ES, FS and GS segment registers.

There are no restrictions regarding the overlapping of the base addresses of any segments. Thus, all 6 segments could have the base address set to zero and create a system with a four gigabyte linear address space. This creates a system where the virtual address space is the same as the linear address space. Further details of segmentation are discussed in section 4.1.

2.8 I/O SPACE

The 80386 has two distinct physical address spaces: Memory and I/O. Generally, peripherals are placed in I/O space although the 80386 also supports memory-mapped peripherals. The I/O space consists of 64K bytes, it can be divided into 64K 8-bit ports, 32K 16-bit ports, or 16K 32-bit ports, or any combination of ports which add up to less than 64K bytes. The 64K I/O address space refers to physical memory rather than linear address since I/O instructions do not go through the segmentation or paging hardware. The M/I/O# pin acts as an additional address line thus allowing the system designer to easily determine which address space the processor is accessing.

Table 2-4. Segment Register Selection Rules

Type of Memory Reference	Implied (Default) Segment Use	Segment Override Prefixes Possible
Code Fetch	CS	None
Destination of PUSH, PUSHA instructions	SS	None
Source of POP, POPA instructions	SS	None
Other data references, with effective address using base register of:		
[EAX]	DS	CS,SS,ES,FS,GS
[EBX]	DS	CS,SS,ES,FS,GS
[ECX]	DS	CS,SS,ES,FS,GS
[EDX]	DS	CS,SS,ES,FS,GS
[EBX]	DS	CS,SS,ES,FS,GS
[ESI]*	DS	CS,SS,ES,FS,GS
[EDI]*	DS	CS,SS,ES,FS,GS
[EBP]	SS	CS,DS,ES,FS,GS
[ESP]	SS	CS,DS,ES,FS,GS

* Data references for the memory destination of the STOS and MOVS instructions (and REP STOS and REP MOVS) use DI as the base register and ES as the segment, with no override possible.

The I/O ports are accessed via the IN and OUT I/O instructions, with the port address supplied as an immediate 8-bit constant in the instruction or in the DX register. All 8- and 16-bit port addresses are zero extended on the upper address lines. The I/O instructions cause the M/IO# pin to be driven low.

I/O port addresses 00F8H through 00FFH are reserved for use by Intel.

2.9 INTERRUPTS

2.9.1 Interrupts and Exceptions

Interrupts and exceptions alter the normal program flow, in order to handle external events, to report errors or exceptional conditions. The difference between interrupts and exceptions is that interrupts are used to handle asynchronous external events while exceptions handle instruction faults. Although a program can generate a software interrupt via an INT N instruction, the processor treats software interrupts as exceptions.

Hardware interrupts occur as the result of an external event and are classified into two types: maskable or non-maskable. Interrupts are serviced after the execution of the current instruction. After the interrupt handler is finished servicing the interrupt, execution proceeds with the instruction immediately after the interrupted instruction. Sections 2.9.3 and 2.9.4 discuss the differences between Maskable and Non-Maskable interrupts.

Exceptions are classified as faults, traps, or aborts depending on the way they are reported, and whether or not restart of the instruction causing the exception is supported. **Faults** are exceptions that are detected and serviced before the execution of the faulting instruction. A fault would occur in a virtual memory system, when the processor referenced a page or a segment which was not present. The operating system would fetch the page or segment from disk, and then the 80386 would restart the instruction. **Traps** are exceptions that are reported immediately after the execution of the instruction which caused the problem. User defined interrupts are examples of traps. **Aborts** are exceptions which do not permit the precise location of the instruction causing the exception to be determined. Aborts are used to report severe errors, such as a hardware error, or illegal values in system tables.

Thus, when an interrupt service routine has been completed, execution proceeds from the instruction immediately following the interrupted instruction. On the other hand, the return address from an exception fault routine will always point at the instruction causing the exception and include any leading instruction prefixes. Table 2-5 summarizes the possible interrupts for the 80386 and shows where the return address points.

The 80386 has the ability to handle up to 256 different interrupts/exceptions. In order to service the interrupts, a table with up to 256 interrupt vectors must be defined. The interrupt vectors are simply pointers to the appropriate interrupt service routine. In Real Mode (see section 3.1), the vectors are 4 byte quantities, a Code Segment plus a 16-bit offset; in Protected Mode, the interrupt vectors are 8 byte quantities, which are put in an Interrupt Descriptor Table (see section 4.1). Of the 256 possible interrupts, 32 are reserved for use by Intel, the remaining 224 are free to be used by the system designer.

2.9.2 Interrupt Processing

When an interrupt occurs the following actions happen. First, the current program address and the Flags are saved on the stack to allow resumption of the interrupted program. Next, an 8-bit vector is supplied to the 80386 which identifies the appropriate entry in the interrupt table. The table contains the starting address of the interrupt service routine. Then, the user supplied interrupt service routine is executed. Finally, when an IRET instruction is executed the old processor state is restored and program execution resumes at the appropriate instruction.

The 8-bit interrupt vector is supplied to the 80386 in several different ways: exceptions supply the interrupt vector internally; software INT instructions contain or imply the vector; maskable hardware interrupts supply the 8-bit vector via the interrupt acknowledge bus sequence. Non-Maskable hardware interrupts are assigned to interrupt vector 2.

2.9.3 Maskable Interrupt

Maskable interrupts are the most common way used by the 80386 to respond to asynchronous external hardware events. A hardware interrupt occurs when the INTR is pulled high and the Interrupt Flag bit (IF) is enabled. The processor only responds to interrupts between instructions, (REPeat String instruc-



Table 2-5. Interrupt Vector Assignments

Function	Interrupt Number	Instruction Which Can Cause Exception	Return Address Points to Faulting Instruction	Type
Divide Error	0	DIV, IDIV	YES	FAULT
Debug Exception	1	any instruction	YES	TRAP*
NMI Interrupt	2	INT 2 or NMI	NO	NMI
One Byte Interrupt	3	INT	NO	TRAP
Interrupt on Overflow	4	INTO	NO	TRAP
Array Bounds Check	5	BOUND	YES	FAULT
Invalid OP-Code	6	Any Illegal Instruction	YES	FAULT
Device Not Available	7	ESC, WAIT	YES	FAULT
Double Fault	8	Any Instruction That Can Generate an Exception		ABORT
Coprocessor Segment Overrun	9	ESC	NO	ABORT
Invalid TSS	10	JMP, CALL, IRET, INT	YES	FAULT
Segment Not Present	11	Segment Register Instructions	YES	FAULT
Stack Fault	12	Stack References	YES	FAULT
General Protection Fault	13	Any Memory Reference	YES	FAULT
Page Fault	14	Any Memory Access or Code Fetch	YES	FAULT
Coprocessor Error	16	ESC, WAIT	YES	FAULT
Intel Reserved	17-32			
Two Byte Interrupt	0-255	INT n	NO	TRAP

* Some debug exceptions may report both traps on the previous instruction, and faults on the next instruction.

tions, have an "interrupt window", between memory moves, which allows interrupts during long string moves). When an interrupt occurs the processor reads an 8-bit vector supplied by the hardware which identifies the source of the interrupt, (one of 224 user defined interrupts). The exact nature of the interrupt sequence is discussed in section 5.

The IF bit in the EFLAG registers is reset when an interrupt is being serviced. This effectively disables servicing additional interrupts during an interrupt service routine. However, the IF may be set explicitly by the interrupt handler, to allow the nesting of interrupts. When an IRET instruction is executed the original state of the IF is restored.

2.9.4 Non-Maskable Interrupt

Non-maskable interrupts provide a method of servicing very high priority interrupts. A common example of the use of a non-maskable interrupt (NMI) would

be to activate a power failure routine. When the NMI input is pulled high it causes an interrupt with an internally supplied vector value of 2. Unlike a normal hardware interrupt, no interrupt acknowledgment sequence is performed for an NMI.

While executing the NMI servicing procedure, the 80386 will not service further NMI requests, until an interrupt return (IRET) instruction is executed or the processor is reset. If NMI occurs while currently servicing an NMI, its presence will be saved for servicing after executing the first IRET instruction. The IF bit is cleared at the beginning of an NMI interrupt to inhibit further INTR interrupts.

2.9.5 Software Interrupts

A third type of interrupt/exception for the 80386 is the software interrupt. An INT n instruction causes

the processor to execute the interrupt service routine pointed to by the nth vector in the interrupt table.

A special case of the two byte software interrupt INT n is the one byte INT 3, or breakpoint interrupt. By inserting this one byte instruction in a program, the user can set breakpoints in his program as a debugging tool.

A final type of software interrupt, is the single step interrupt. It is discussed in section 2.12.

2.9.6 Interrupt and Exception Priorities

Interrupts are externally-generated events. Maskable Interrupts (on the INTR input) and Non-Maskable Interrupts (on the NMI input) are recognized at instruction boundaries. When NMI and maskable INTn are both recognized at the same instruction boundary, the 80386 invokes the NMI service routine first. If, after the NMI service routine has been invoked, maskable interrupts are still enabled, then the 80386 will invoke the appropriate interrupt service routine.

Table 2-6a. 80386 Priority for Invoking Service Routines In Case of Simultaneous External Interrupts

- | |
|---------|
| 1. NMI |
| 2. INTR |

Exceptions are internally-generated events. Exceptions are detected by the 80386 if, in the course of executing an instruction, the 80386 detects a problematic condition. The 80386 then immediately invokes the appropriate exception service routine. The state of the 80386 is such that the instruction causing the exception can be restarted. If the exception service routine has taken care of the problematic condition, the instruction will execute without causing the same exception.

It is possible for a single instruction to generate several exceptions (for example, transferring a single operand could generate two page faults if the operand and location spans two "not present" pages). However, only one exception is generated upon each attempt to execute the instruction. Each exception service routine should correct its corresponding exception, and restart the instruction. In this manner, exceptions are serviced until the instruction executes successfully.

As the 80386 executes instructions, it follows a consistent cycle in checking for exceptions, as shown in Table 2-6b. This cycle is repeated as each instruc-

tion is executed, and occurs in parallel with instruction decoding and execution.

Table 2-6b. Sequence of Exception Checking

Consider the case of the 80386 having just completed an instruction. It then performs the following checks before reaching the point where the next instruction is completed:

1. Check for Exception 1 Traps from the instruction just completed (single-step via Trap Flag, or Data Breakpoints set in the Debug Registers).
2. Check for external NMI and INTR.
3. Check for Exception 1 Faults in the next instruction (Instruction Execution Breakpoint set in the Debug Registers for the next instruction).
4. Check for Segmentation Faults that prevented fetching the entire next instruction (exceptions 11 or 13).
5. Check for Page Faults that prevented fetching the entire next instruction (exception 14).
6. Check for Faults decoding the next instruction (exception 6 if illegal opcode; exception 6 if in Real Mode or in Virtual 8086 Mode and attempting to execute an instruction for Protected Mode only (see 4.6.4); or exception 13 if instruction is longer than 15 bytes, or privilege violation in Protected Mode (i.e. not at IOPL or at CPL = 0)).
7. If WAIT opcode, check if TS=1 and MP=1 (exception 7 if both are 1).
8. If ESCAPE opcode for numeric coprocessor, check if EM=1 or TS=1 (exception 7 if either are 1).
9. If WAIT opcode or ESCAPE opcode for numeric coprocessor, check ERROR# input signal (exception 16 if ERROR# input is asserted).
10. Check in the following order for each memory reference required by the instruction:
 - a. Check for Segmentation Faults that prevent transferring the entire memory quantity (exceptions 11, 12, 13).
 - b. Check for Page Faults that prevent transferring the entire memory quantity (exception 14).

Note that the order stated supports the concept of the paging mechanism being "underneath" the segmentation mechanism. Therefore, for any given code or data reference in memory, segmentation exceptions are generated before paging exceptions are generated.



2.9.7 Instruction Restart

The 80386 fully supports restarting all instructions after faults. If an exception is detected in the instruction to be executed (exception categories 4 through 10 in Table 2-6c), the 80386 invokes the appropriate exception service routine. The 80386 is in a state that permits restart of the instruction, for all cases but those in Table 2-6c. Note that all such cases are easily avoided by proper design of the operating system.

Table 2-6c. Conditions Preventing Instruction Restart

- A. An instruction causes a task switch to a task whose Task State Segment is partially "not present". (An entirely "not present" TSS is restartable.) Partially present TSS's can be avoided either by keeping the TSS's of such tasks present in memory, or by aligning TSS segments to reside entirely within a single 4K page (for TSS segments of 4K bytes or less).
- B. A coprocessor operand wraps around the top of a 64K-byte segment or a 4G-byte segment, and spans three pages, and the page holding the middle portion of the operand is "not present." This condition can be avoided by starting at a page boundary any segments containing coprocessor operands if the segments are approximately 64K-200 bytes or larger (i.e. large enough for wraparound of the coprocessor operand to possibly occur).

Note that these conditions are avoided by using the operating system designs mentioned in this table.

2.9.8 Double Fault

A Double Fault (exception 8) results when the processor attempts to invoke an exception service routine for the segment exceptions (10, 11, 12 or 13), but in the process of doing so, detects an exception other than a Page Fault (exception 14).

One other cause of generating a Double Fault (exception 8) is the 80386 detecting any other exception when it is attempting to invoke the Page Fault (exception 14) service routine (for example, if a Page Fault is detected when the 80386 attempts to invoke the Page Fault service routine). Of course in any functional system, not only in 80386-based systems, the entire page fault service routine must remain "present" in memory.

When a Double Fault occurs, the 80386 invokes the exception service routine for exception 8.

2.10 RESET AND INITIALIZATION

When the processor is initialized or Reset the registers have the values shown in Table 2-7. The 80386 will then start executing instructions near the top of physical memory, at location FFFFFFFFOH. When the first InterSegment Jump or Call is executed, address lines A20-31 will drop low for CS-relative memory cycles, and the 80386 will only execute instructions in the lower one megabyte of physical memory. This allows the system designer to use a ROM at the top of physical memory to initialize the system and take care of Resets.

RESET forces the 80386 to terminate all execution and local bus activity. No instruction execution or bus activity will occur as long as Reset is active. Between 350 and 450 CLK2 periods after Reset becomes inactive the 80386 will start executing instructions at the top of physical memory.

Table 2-7. Register Values after Reset

Flag Word	UUUUU002H Note 1
Machine Status Word (CR0)	UUUUUUU0H Note 2
Instruction Pointer	0000FFFFH
Code Segment	F000H Note 3
Data Segment	0000H
Stack Segment	0000H
Extra Segment (ES)	0000H
Extra Segment (FS)	0000H
Extra Segment (GS)	0000H
DX register	component and stepping ID Note 5
All other registers	undefined Note 4

NOTES:

1. EFLAG Register. The upper 14 bits of the EFLAGS register are undefined, VM (Bit 17) and RF (BIT 16) are 0 as are all other defined flag bits.

2. CR0: (Machine Status Word). All of the defined fields in the CR0 are 0 (PG Bit 31, TS Bit 3, EM Bit 2, MP Bit 1, and PE Bit 0) except for ET Bit 4 (processor extension type). The ET Bit is set during Reset according to the type of Coprocessor in the system. If the coprocessor is an 80387 then ET will be 1, if the coprocessor is an 80287 or no coprocessor is present then ET will be 0. All other bits are undefined.

3. The Code Segment Register (CS) will have its Base Address set to FFFF0000H and Limit set to 0FFFFH.

4. All undefined bits are Intel Reserved and should not be used.

5. DX register always holds component and stepping identifier (see 5.7). EAX register holds self-test signature if self-test was requested (see 5.6).



2.11 TESTABILITY

2.11.1 Self-Test

The 80386 has the capability to perform a self-test. The self-test checks the function of all of the Control ROM and most of the non-random logic of the part. Approximately one-half of the 80386 can be tested during self-test.

Self-Test is initiated on the 80386 when the RESET pin transitions from HIGH to LOW, and the BUSY# pin is low. The self-test takes about 2^{19} clocks, or approximately 33 milliseconds with a 16 MHz 80386. At the completion of self-test the processor performs reset and begins normal operation. The part has successfully passed self-test if the contents of the EAX register are zero (0). If the results of EAX are not zero then the self-test has detected a flaw in the part.

2.11.2 TLB Testing

The 80386 provides a mechanism for testing the Translation Lookaside Buffer (TLB) if desired. This particular mechanism is unique to the 80386 and may not be continued in the same way in future processors. When testing the TLB paging must be turned off (PG = 0 in CR0) to enable the TLB testing hardware and avoid interference with the test data being written to the TLB.

There are two TLB testing operations: 1) write entries into the TLB, and, 2) perform TLB lookups. Two Test Registers, shown in Figure 2-12, are provided for the purpose of testing. TR6 is the "test command register", and TR7 is the "test data register". The fields within these registers are defined below.

C: This is the command bit. For a write into TR6 to cause an immediate write into the TLB entry, write a 0 to this bit. For a write into TR6 to cause an immediate TLB lookup, write a 1 to this bit.

Linear Address: This is the tag field of the TLB. On a TLB write, a TLB entry is allocated to this linear address and the rest of that TLB entry is set per the value of TR7 and the value just written into TR6. On a TLB lookup, the TLB is interrogated per this value and if one and only one TLB entry matches, the rest of the fields of TR6 and TR7 are set from the matching TLB entry.

Physical Address: This is the data field of the TLB. On a write to the TLB, the TLB entry allocated to the linear address in TR6 is set to this value. On a TLB lookup, the data field (physical address) from the TLB is read out to here.

PL: On a TLB write, PL = 1 causes the REP field of TR7 to select which of four associative blocks of the TLB is to be written, but PL = 0 allows the internal pointer in the paging unit to select which TLB block is written. On a TLB lookup, the PL bit indicates whether the lookup was a hit (PL gets set to 1) or a miss (PL gets reset to 0).

V: The valid bit for this TLB entry. All valid bits can also be cleared by writing to CR3.

D, D#: The dirty bit for/from the TLB entry.

U, U#: The user bit for/from the TLB entry.

W, W#: The writable bit for/from the TLB entry.

For D, U and W, both the attribute and its complement are provided as tag bits, to permit the option of a "don't care" on TLB lookups. The meaning of these pairs of bits is given in the following table:

X	X#	Effect During TLB Lookup	Value of Bit X after TLB Write
0	0	Miss All	Bit X Becomes Undefined
0	1	Match if X = 0	Bit X Becomes 0
1	0	Match if X = 1	Bit X Becomes 1
1	1	Match all	Bit X Becomes Undefined

For writing a TLB entry:

1. Write TR7 for the desired physical address, PL and REP values.
2. Write TR6 with the appropriate linear address, etc. (be sure to write C = 0 for "write" command).

For looking up (reading) a TLB entry:

1. Write TR6 with the appropriate linear address (be sure to write C = 1 for "lookup" command).
2. Read TR7 and TR6. If the PL bit in TR7 indicates a hit, then the other values reveal the TLB contents. If PL indicates a miss, then the other values in TR7 and TR6 are indeterminate.

2.12 DEBUGGING SUPPORT

The 80386 provides several features which simplify the debugging process. The three categories of on-chip debugging aids are:

- 1) the code execution breakpoint opcode (0CCH),
- 2) the single-step capability provided by the TF bit in the flag register, and
- 3) the code and data breakpoint capability provided by the Debug Registers DR0-3, DR6, and DR7.

31	12	11											0
LINEAR ADDRESS	V	D	D*	U	U*	W	W*	0	0	0	0	C	TR6
PHYSICAL ADDRESS	0	0	0	0	0	0	0	P	L	REP	0	0	TR7

NOTE: **0** indicates Intel reserved: Do not define; SEE SECTION 2.3.10

Figure 2-12. Test Registers

2.12.1 Breakpoint Instruction

A single-byte-opcode breakpoint instruction is available for use by software debuggers. The breakpoint opcode is 0CCh, and generates an exception 3 trap when executed. In typical use, a debugger program can "plant" the breakpoint instruction at all desired code execution breakpoints. The single-byte breakpoint opcode is an alias for the two-byte general software interrupt instruction, INT n, where n=3. The only difference between INT 3 (0CCh) and INT n is that INT 3 is never IOPL-sensitive but INT n is IOPL-sensitive in Protected Mode and Virtual 8086 Mode.

2.12.2 Single-Step Trap

If the single-step flag (TF, bit 8) in the EFLAG register is found to be set at the end of an instruction, a single-step exception occurs. The single-step exception is auto vectored to exception number 1. Precisely, exception 1 occurs as a trap after the instruction following the instruction which set TF. In typical practice, a debugger sets the TF bit of a flag register image on the debugger's stack. It then typically transfers control to the user program and loads the flag image with a signal instruction, the IRET instruction. The single-step trap occurs after executing one instruction of the user program.

Since the exception 1 occurs as a trap (that is, it occurs after the instruction has already executed), the CS:EIP pushed onto the debugger's stack points to the next unexecuted instruction of the program being debugged. An exception 1 handler, merely by ending with an IRET instruction, can therefore efficiently support single-stepping through a user program.

2.12.3 Debug Registers

The Debug Registers are an advanced debugging feature of the 80386. They allow data access breakpoints as well as code execution breakpoints. Since the breakpoints are indicated by on-chip registers, an instruction execution breakpoint can be placed in

ROM code or in code shared by several tasks, neither of which can be supported by the INT3 breakpoint opcode.

The 80386 contains six Debug Registers, providing the ability to specify up to four distinct breakpoints addresses, breakpoint control options, and read breakpoint status. Initially after reset, breakpoints are in the disabled state. Therefore, no breakpoints will occur unless the debug registers are programmed. Breakpoints set up in the Debug Registers are autovectored to exception number 1.

2.12.3.1 LINEAR ADDRESS BREAKPOINT REGISTERS (DR0-DR3)

Up to four breakpoint addresses can be specified by writing into Debug Registers DR0-DR3, shown in Figure 2-13. The breakpoint addresses specified are 32-bit linear addresses. 80386 hardware continuously compares the linear breakpoint addresses in DR0-DR3 with the linear addresses generated by executing software (a linear address is the result of computing the effective address and adding the 32-bit segment base address). Note that if paging is not enabled the linear address equals the physical address. If paging is enabled, the linear address is translated to a physical 32-bit address by the on-chip paging unit. Regardless of whether paging is enabled or not, however, the breakpoint registers hold linear addresses.

2.12.3.2 DEBUG CONTROL REGISTER (DR7)

A Debug Control Register, DR7 shown in Figure 2-13, allows several debug control functions such as enabling the breakpoints and setting up other control options for the breakpoints. The fields within the Debug Control Register, DR7, are as follows:

LENi (breakpoint length specification bits)

A 2-bit LEN field exists for each of the four breakpoints. LEN specifies the length of the associated breakpoint field. The choices for data breakpoints are: 1 byte, 2 bytes, and 4 bytes. Instruction execu-

Figure 2-13. Debug Registers

tion breakpoints must have a length of 1 (LENi = 00). Encoding of the LENi field is as follows:

The following is an example of various size breakpoint fields. Assume the breakpoint linear address in DR2 is 00000005H. In that situation, the following illustration indicates the region of the breakpoint field for lengths of 1, 2, or 4 bytes.

LENI Encoding	Breakpoint Field Width	Usage of Least Significant Bits in Breakpoint Address Register I, (I = 0 – 3)
00	1 byte	All 32-bits used to specify a single-byte breakpoint field.
01	2 bytes	A1–A31 used to specify a two-byte, word-aligned breakpoint field. A0 in Breakpoint Address Register is not used.
10	Undefined—do not use this encoding	
11	4 bytes	A2–A31 used to specify a four-byte, dword-aligned breakpoint field. A0 and A1 in Breakpoint Address Register are not used.

The LENi field controls the size of breakpoint field by controlling whether all low-order linear address bits in the breakpoint address register are used to detect the breakpoint event. Therefore, all breakpoint fields are aligned; 2-byte breakpoint fields begin on Word boundaries, and 4-byte breakpoint fields begin on Dword boundaries.

DR2 = 00000005H; LEN2 = 00B			
31			0
			00000008H
		bkpt fld2	00000004H
			00000000H
 DR2 = 00000005H; LEN2 = 01B			
31			0
			00000008H
	←	bkpt fld2	→ 00000004H
			00000000H
 DR2 = 00000005H; LEN2 = 11B			
31			0
			00000008H
	←	bkpt fld2	→ 00000004H
			00000000H



RWi (memory access qualifier bits)

A 2-bit RW field exists for each of the four breakpoints. The 2-bit RW field specifies the type of usage which must occur in order to activate the associated breakpoint.

RW Encoding	Usage Causing Breakpoint
00	Instruction execution only
01	Data writes only
10	Undefined—do not use this encoding
11	Data reads and writes only

RW encoding 00 is used to set up an instruction execution breakpoint. RW encodings 01 or 11 are used to set up write-only or read/write data breakpoints.

Note that **instruction execution breakpoints are taken as faults** (i.e. before the instruction executes), but **data breakpoints are taken as traps** (i.e. after the data transfer takes place).

Using LENi and RWi to Set Data Breakpoint i

A data breakpoint can be set up by writing the linear address into DRi ($i = 0\text{--}3$). For data breakpoints, RWi can = 01 (write-only) or 11 (write/read). LEN can = 00, 01, or 11.

If a data access entirely or partly falls within the data breakpoint field, the data breakpoint condition has occurred, and if the breakpoint is enabled, an exception 1 trap will occur.

Using LENi and RWi to Set Instruction Execution Breakpoint i

An instruction execution breakpoint can be set up by writing address of the beginning of the instruction (including prefixes if any) into DRi ($i = 0\text{--}3$). RWi must = 00 and LEN must = 00 for instruction execution breakpoints.

If the instruction beginning at the breakpoint address is about to be executed, the instruction execution breakpoint condition has occurred, and if the breakpoint is enabled, an exception 1 fault will occur before the instruction is executed.

Note that an instruction execution breakpoint address must be equal to the **beginning** byte address of an instruction (including prefixes) in order for the instruction execution breakpoint to occur.

GD (Global Debug Register access detect)

The Debug Registers can only be accessed in Real Mode or at privilege level 0 in Protected Mode. The

GD bit, when set, provides extra protection against any Debug Register access even in Real Mode or at privilege level 0 in Protected Mode. This additional protection feature is provided to guarantee that a software debugger (or ICE-386) can have full control over the Debug Register resources when required. The GD bit, when set, causes an exception 1 fault if an instruction attempts to read or write any Debug Register. The GD bit is then automatically cleared when the exception 1 handler is invoked, allowing the exception 1 handler free access to the debug registers.

GE and LE (Exact data breakpoint match, global and local)

If either GE or LE is set, any data breakpoint trap will be reported exactly after completion of the instruction that caused the operand transfer. Exact reporting is provided by forcing the 80386 execution unit to wait for completion of data operand transfers before beginning execution of the next instruction.

If exact data breakpoint match is not selected, data breakpoints may not be reported until several instructions later or may not be reported at all. When enabling a data breakpoint, it is therefore recommended to enable the exact data breakpoint match.

When the 80386 performs a task switch, the LE bit is cleared. Thus, the LE bit supports fast task switching out of tasks, that have enabled the exact data breakpoint match for their task-local breakpoints. The LE bit is cleared by the processor during a task switch, to avoid having exact data breakpoint match enabled in the new task. Note that exact data breakpoint match must be re-enabled under software control.

The 80386 GE bit is unaffected during a task switch. The GE bit supports exact data breakpoint match that is to remain enabled during all tasks executing in the system.

Note that **instruction execution breakpoints are always reported exactly**, whether or not exact data breakpoint match is selected.

Gi and Li (breakpoint enable, global and local)

If either Gi or Li is set then the associated breakpoint (as defined by the linear address in DRi, the length in LENi and the usage criteria in RWi) is enabled. If either Gi or Li is set, and the 80386 detects the i th breakpoint condition, then the exception 1 handler is invoked.

When the 80386 performs a task switch to a new TSS, all Li bits are cleared. Thus, the Li bits support fast task switching out of tasks that use some task-local breakpoint registers. The Li bits are cleared by

the processor during a task switch, to avoid spurious exceptions in the new task. Note that the breakpoints must be re-enabled under software control.

All 80386 Gi bits are unaffected during a task switch. The Gi bits support breakpoints that are active in all tasks executing in the system.

2.12.3.3 DEBUG STATUS REGISTER (DR6)

A Debug Status Register, DR6 shown in Figure 2-13, allows the exception 1 handler to easily determine why it was invoked. Note the exception 1 handler can be invoked as a result of one of several events:

- 1) DR0 Breakpoint fault/trap.
- 2) DR1 Breakpoint fault/trap.
- 3) DR2 Breakpoint fault/trap.
- 4) DR3 Breakpoint fault/trap.
- 5) Single-step (TF) trap.
- 6) Task switch trap.
- 7) Fault due to attempted debug register access when GD=1.

The Debug Status Register contains single-bit flags for each of the possible events invoking exception 1. Note below that some of these events are faults (exception taken before the instruction is executed), while other events are traps (exception taken after the debug events occurred).

The flags in DR6 are set by the hardware but never cleared by hardware. Exception 1 handler software should clear DR6 before returning to the user program to avoid future confusion in identifying the source of exception 1.

The fields within the Debug Status Register, DR6, are as follows:

Bi (debug fault/trap due to breakpoint 0-3)

Four breakpoint indicator flags, B0-B3, correspond one-to-one with the breakpoint registers in DR0-DR3. A flag Bi is set when the condition described by DRi, LENi, and RWi occurs.

If Gi or Li is set, and if the ith breakpoint is detected, the processor will invoke the exception 1 handler. The exception is handled as a fault if an instruction execution breakpoint occurred, or as a trap if a data breakpoint occurred.

IMPORTANT NOTE: A flag Bi is set whenever the hardware detects a match condition on **enabled** breakpoint i. Whenever a match is detected on at least one **enabled** breakpoint i, the hardware immediately sets all Bi bits corresponding to breakpoint conditions matching at that instant, whether enabled or not. Therefore, the exception 1 handler may see

that multiple Bi bits are set, but only set Bi bits corresponding to **enabled** breakpoints (Li or Gi set) are **true** indications of why the exception 1 handler was invoked.

BD (debug fault due to attempted register access when GD bit set)

This bit is set if the exception 1 handler was invoked due to an instruction attempting to read or write to the debug registers when GD bit was set. If such an event occurs, then the GD bit is automatically cleared when the exception 1 handler is invoked, allowing handler access to the debug registers.

BS (debug trap due to single-step)

This bit is set if the exception 1 handler was invoked due to the TF bit in the flag register being set (for single-stepping). See section 2.12.2.

BT (debug trap due to task switch)

This bit is set if the exception 1 handler was invoked due to a task switch occurring to a task having a 386 TSS with the T bit set. (See Figure 4-15a). Note the task switch into the new task occurs normally, but before the first instruction of the task is executed, the exception 1 handler is invoked. With respect to the task switch operation, the operation is considered to be a trap.

2.12.3.4 USE OF RESUME FLAG (RF) IN FLAG REGISTER

The Resume Flag (RF) in the flag word can suppress an instruction execution breakpoint when the exception 1 handler returns to a user program at a user address which is also an instruction execution breakpoint. See section 2.3.3.

3. REAL MODE ARCHITECTURE

3.1 REAL MODE INTRODUCTION

When the processor is reset or powered up it is initialized in Real Mode. Real Mode has the same base architecture as the 8086, but allows access to the 32-bit register set of the 80386. The addressing mechanism, memory size, interrupt handling, are all identical to the Real Mode on the 80286.

All of the 80386 instructions are available in Real Mode (except those instructions listed in 4.6.4). The default operand size in Real Mode is 16-bits, just like the 8086. In order to use the 32-bit registers and addressing modes, override prefixes must be used. In addition, the segment size on the 80386 in Real Mode is 64K bytes so 32-bit effective addresses must have a value less than 0000FFFFH. The primary

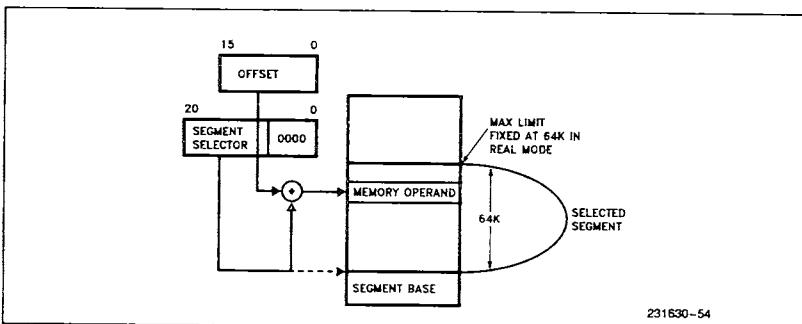


Figure 3-1. Real Address Mode Addressing

231630-54

purpose of Real Mode is to set up the processor for Protected Mode Operation.

The LOCK prefix on the 80386, even in Real Mode, is more restrictive than on the 80286. This is due to the addition of paging on the 80386 in Protected Mode and Virtual 8086 Mode. Paging makes it impossible to guarantee that repeated string instructions can be LOCKed. The 80386 can't require that all pages holding the string be physically present in memory. Hence, a Page Fault (exception 14) might have to be taken during the repeated string instruction. Therefore the LOCK prefix can't be supported during repeated string instructions.

These are the only instruction forms where the LOCK prefix is legal on the 80386:

Opcode	Operands (Dest, Source)
BIT Test and SET/RESET/COMPLEMENT	Mem, Reg/immed
XCHG	Reg, Mem
XCHG	Mem, Reg
ADD, OR, ADC, SBB, AND, SUB, XOR	Mem, Reg/immed
NOT, NEG, INC, DEC	Mem

An exception 6 will be generated if a LOCK prefix is placed before any instruction form or opcode not listed above. The LOCK prefix allows indivisible read/modify/write operations on memory operands using the instructions above. For example, even the "ADD Reg/immed, Mem" is not LOCKable, because the Mem operand is not the destination (and therefore no memory read/modify/operation is being performed).

Since, on the 80386, repeated string instructions are not LOCKable, it is not possible to LOCK the bus for a long period of time. Therefore, the LOCK prefix is not IOPL-sensitive on the 80386. The LOCK prefix can be used at any privilege level, but only on the instruction forms listed above.

3.2 MEMORY ADDRESSING

In Real Mode the maximum memory size is limited to 1 megabyte. Thus, only address lines A2-A19, BEO-BE are active. (Exception, the high address lines A20-A31 are high during CS-relative memory cycles until an intersegment jump or call is executed (see section 2.10)).

Since paging is not allowed in Real Mode the linear addresses are the same as physical addresses. Physical addresses are formed in Real Mode by adding the contents of the appropriate segment register which is shifted left by four bits to an effective address. This addition results in a physical address from 0000000H to 0010FFFFH. This is compatible with 80286 Real Mode. Since segment registers are shifted left by 4 bits this implies that Real Mode segments always start on 16 byte boundaries.

All segments in Real Mode are exactly 64K bytes long, and may be read, written, or executed. The 80386 will generate an exception 13 if a data operand or instruction fetch occurs past the end of a segment (i.e. if an operand has an offset greater than FFFFH, example a word with a low byte at FFFFH and the high byte at 0000H)

Segments may be overlapped in Real Mode. Thus, if a particular segment does not use all 64K bytes another segment can be overlayed on top of the unused portion of the previous segment. This allows the programmer to minimize the amount of physical memory needed for a program.

3.3 RESERVED LOCATIONS

There are two fixed areas in memory which are reserved in Real address mode: system initialization area and the interrupt table area. Locations 00000H through 003FFH are reserved for interrupt vectors. Each one of the 256 possible interrupts has a 4-byte jump vector reserved for it. Locations FFFFFFFFOH through FFFFFFFFH are reserved for system initialization.

3.4 INTERRUPTS

Many of the exceptions shown in Table 2-5 and discussed in section 2.9 are not applicable to Real Mode operation, in particular exceptions 10, 11, 14, will not happen in Real Mode. Other exceptions have slightly different meanings in Real Mode; Table 3-1 identifies these exceptions.

3.5 SHUTDOWN AND HALT

The HLT instruction stops program execution and prevents the processor from using the local bus until restarted. Either NMI, INTR with interrupts enabled (IF = 1), or RESET will force the 80386 out of halt. If interrupted, the saved CS:IP will point to the next instruction after the HLT.

Shutdown will occur when a severe error is detected that prevents further processing. In Real Mode, shutdown can occur under two conditions:

An interrupt or an exception occur (Exceptions 8 or 13) and the interrupt vector is larger than the Interrupt Descriptor Table (i.e. There is not an interrupt handler for the interrupt).

A CALL, INT or PUSH instruction attempts to wrap around the stack segment when SP is not even. (e.g. pushing a value on the stack when SP = 0001 resulting in a stack segment greater than FFFFH)

An NMI input can bring the processor out of shutdown if the Interrupt Descriptor Table limit is large enough to contain the NMI interrupt vector (at least 0017H) and the stack has enough room to contain the vector and flag information (i.e. SP is greater than 0005H). Otherwise shutdown can only be exited via the RESET input.

4. PROTECTED MODE ARCHITECTURE

4.1 INTRODUCTION

The complete capabilities of the 80386 are unlocked when the processor operates in Protected Virtual Address Mode (Protected Mode). Protected Mode vastly increases the linear address space to four gigabytes (2^{32} bytes) and allows the running of virtual memory programs of almost unlimited size (64 terabytes or 2^{48} bytes). In addition Protected Mode allows the 80386 to run all of the existing 8086 and 80286 software, while providing a sophisticated memory management and a hardware-assisted protection mechanism. Protected Mode allows the use of additional instructions especially optimized for supporting multitasking operating systems. The base architecture of the 80386 remains the same, the registers, instructions, and addressing modes described in the previous sections are retained. The main difference between Protected Mode, and Real Mode from a programmer's view is the increased address space, and a different addressing mechanism.

Table 3-1

Function	Interrupt Number	Related Instructions	Return Address Location
Interrupt table limit too small	8	INT Vector is not within table limit	Before Instruction
CS, DS, ES, FF, GS Segment overrun exception	13	Word memory reference beyond offset = FFFFH. An attempt to execute past the end of CS segment.	Before Instruction
SS Segment overrun exception	12	Stack Reference beyond offset = FFFFH	Before Instruction

4.2 ADDRESSING MECHANISM

Like Real Mode, Protected Mode uses two components to form the logical address, a 16-bit selector is used to determine the linear base address of a segment, the base address is added to a 32-bit effective address to form a 32-bit linear address. The linear address is then either used as the 32-bit physical address, or if paging is enabled the paging mechanism maps the 32-bit linear address into a 32-bit physical address.

The difference between the two modes lies in calculating the base address. In Protected Mode the selector is used to specify an index into an operating

system defined table (see Figure 4-1). The table contains the 32-bit base address of a given segment. The physical address is formed by adding the base address obtained from the table to the offset.

Paging provides an additional memory management mechanism which operates only in Protected Mode. Paging provides a means of managing the very large segments of the 80386. As such, paging operates beneath segmentation. The paging mechanism translates the protected linear address which comes from the segmentation unit into a physical address. Figure 4-2 shows the complete 80386 addressing mechanism with paging enabled.

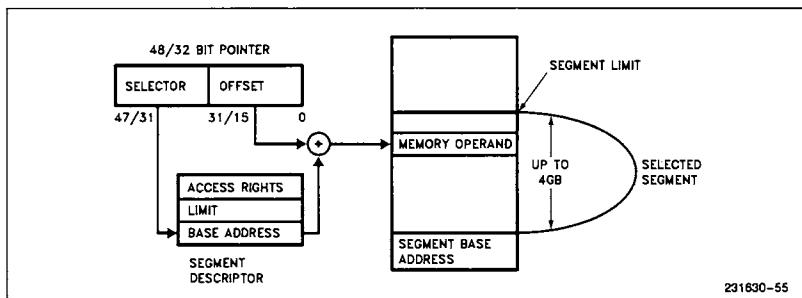


Figure 4-1. Protected Mode Addressing

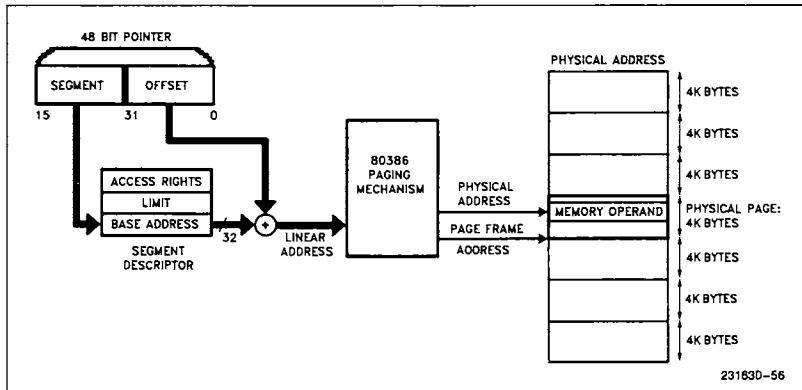


Figure 4-2. Paging and Segmentation

4.3 SEGMENTATION

4.3.1 Segmentation Introduction

Segmentation is one method of memory management. Segmentation provides the basis for protection. Segments are used to encapsulate regions of memory which have common attributes. For example, all of the code of a given program could be contained in a segment, or an operating system table may reside in a segment. All information about a segment is stored in an 8 byte data structure called a descriptor. All of the descriptors in a system are contained in tables recognized by hardware.

4.3.2 Terminology

The following terms are used throughout the discussion of descriptors, privilege levels and protection:

PL: Privilege Level—One of the four hierarchical privilege levels. Level 0 is the most privileged level and level 3 is the least privileged. More privileged levels are numerically smaller than less privileged levels.

RPL: Requestor Privilege Level—The privilege level of the original supplier of the selector. RPL is determined by the least two significant bits of a selector.

DPL: Descriptor Privilege Level—This is the least privileged level at which a task may access that descriptor (and the segment associated with that descriptor). Descriptor Privilege Level is determined by bits 6:5 in the Access Right Byte of a descriptor.

CPL: Current Privilege Level—The privilege level at which a task is currently executing, which equals the privilege level of the code segment being executed.

CPL can also be determined by examining the lowest 2 bits of the CS register, except for conforming code segments.

EPL: Effective Privilege Level—The effective privilege level is the least privileged of the RPL and DPL. Since smaller privilege level values indicate greater privilege, EPL is the numerical maximum of RPL and DPL.

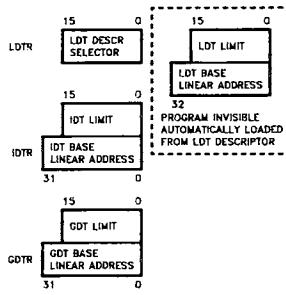
Task: One instance of the execution of a program. Tasks are also referred to as processes.

4.3.3 Descriptor Tables

4.3.3.1 DESCRIPTOR TABLES INTRODUCTION

The descriptor tables define all of the segments which are used in an 80386 system. There are three types of tables on the 80386 which hold descriptors: the Global Descriptor Table, Local Descriptor Table, and the Interrupt Descriptor Table. All of the tables are variable length memory arrays. They can range in size between 8 bytes and 64K bytes. Each table can hold up to 8192 8 byte descriptors. The upper 13 bits of a selector are used as an index into the descriptor table. The tables have registers associated with them which hold the 32-bit linear base address, and the 16-bit limit of each table.

Each of the tables has a register associated with it the GDTR, LDTR, and the IDTR (see Figure 4-3). The LGDT, LLDT, and LIDT instructions, load the base and limit of the Global, Local, and Interrupt Descriptor Tables, respectively, into the appropriate register. The SGDT, SLDT, and SIDT store the base and limit values. These tables are manipulated by the operating system. Therefore, the load descriptor table instructions are privileged instructions.



231630-57

Figure 4-3. Descriptor Table Registers

4.3.3.2 GLOBAL DESCRIPTOR TABLE

The Global Descriptor Table (GDT) contains descriptors which are possibly available to all of the tasks in a system. The GDT can contain any type of segment descriptor except for descriptors which are used for servicing interrupts (i.e. interrupt and trap descriptors). Every 386 system contains a GDT. Generally the GDT contains code and data segments used by the operating systems and task state segments, and descriptors for the LDTs in a system.

The first slot of the Global Descriptor Table corresponds to the null selector and is not used. The null selector defines a null pointer value.

4.3.3.3 LOCAL DESCRIPTOR TABLE

LDTs contain descriptors which are associated with a given task. Generally, operating systems are designed so that each task has a separate LDT. The LDT may contain only code, data, stack, task gate, and call gate descriptors. LDTs provide a mechanism for isolating a given task's code and data segments from the rest of the operating system, while the GDT contains descriptors for segments which are common to all tasks. A segment cannot be accessed by a task if its segment descriptor does not exist in either the current LDT or the GDT. This provides both isolation and protection for a task's segments, while still allowing global data to be shared among tasks.

Unlike the 6 byte GDT or IDT registers which contain a base address and limit, the visible portion of the LDT register contains only a 16-bit selector. This selector refers to a Local Descriptor Table descriptor in the GDT.

4.3.3.4 INTERRUPT DESCRIPTOR TABLE

The third table needed for 80386 systems is the Interrupt Descriptor Table. (See Figure 4-4.) The IDT contains the descriptors which point to the location of up to 256 interrupt service routines. The IDT may contain only task gates, interrupt gates, and trap gates. The IDT should be at least 256 bytes in size in order to hold the descriptors for the 32 Intel Reserved Interrupts. Every interrupt used by a system must have an entry in the IDT. The IDT entries are referenced via INT instructions, external interrupt vectors, and exceptions. (See 2.9 Interrupts).

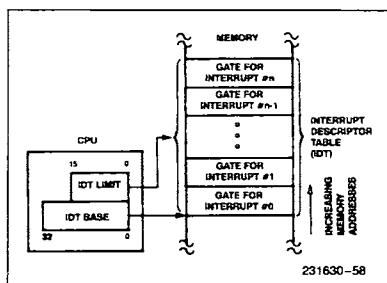


Figure 4-4. Interrupt Descriptor Table Register Use

4.3.4 Descriptors

4.3.4.1 DESCRIPTOR ATTRIBUTE BITS

The object to which the segment selector points to is called a descriptor. Descriptors are eight byte quantities which contain attributes about a given region of linear address space (i.e. a segment). These

31										0	BYTE ADDRESS					
SEGMENT BASE 15...0										0	0					
BASE	Base Address of the segment									P	DPL	S	TYPE	A	BASE 23...16	+ 4
LIMIT	The length of the segment									Present Bit 1 = Present 0 = Not Present						
P	Descriptor Privilege Level 0-3															
DPL	Segment Descriptor 0 = System Descriptor 1 = Code or Data Segment Descriptor															
S	Type of Segment															
Type of Segment																
A	Accessed Bit															
G	Granularity Bit 1 = Segment length is page granular 0 = Segment length is byte granular															
D	Default Operation Size (recognized in code segment descriptors only) 1 = 32-bit segment 0 = 16-bit segment															
O	Bit must be zero (0) for compatibility with future processors															

Figure 4-5. Segment Descriptors

attributes include the 32-bit base linear address of the segment, the 20-bit length and granularity of the segment, the protection level, read, write or execute privileges, the default size of the operands (16-bit or 32-bit), and the type of segment. All of the attribute information about a segment is contained in 12 bits in the segment descriptor. Figure 4-5 shows the general format of a descriptor. All segments on the 80386 have three attribute fields in common: the P bit, the DPL bit, and the S bit. The Present P bit is 1 if the segment is loaded in physical memory, if P=0 then any attempt to access this segment causes a not present exception (exception 11). The Descriptor Privilege Level **DPL** is a two-bit field which specifies the protection level 0-3 associated with a segment.

The 80386 has two main categories of segments system segments and non-system segments (for code and data). The segment **S** bit in the segment descriptor determines if a given segment is a system segment or a code or data segment. If the S bit is 1 then the segment is either a code or data segment, if it is 0 then the segment is a system segment.

4.3.4.2 386 CODE, DATA DESCRIPTORS (S=1)

Figure 4-6 shows the general format of a code and data descriptor and Table 4-1 illustrates how the bits in the Access Rights Byte are interpreted.

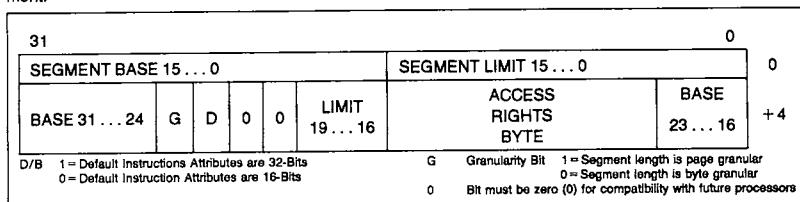


Figure 4-6. Segment Descriptors

Table 4-1. Access Rights Byte Definition for Code and Data Descriptions

Bit Position		Name	Function	
Type Field Definition	7	Present (P)	P = 1	Segment is mapped into physical memory. P = 0 No mapping to physical memory exists, base and limit are not used.
	6-5	Descriptor Privilege Level (DPL)	Segment privilege attribute used in privilege tests.	
	4	Segment Descriptor (S)	S = 1	Code or Data (includes stacks) segment descriptor S = 0 System Segment Descriptor or Gate Descriptor
	3	Executable (E)	E = 0	Descriptor type is data segment: ED = 0 Expand up segment, offsets must be < limit. ED = 1 Expand down segment, offsets must be > limit.
	2	Expansion Direction (ED)	ED = 0	If Data Segment (S = 1, E = 0)
	1	Writeable (W)	ED = 1	Expand down segment, offsets must be > limit. W = 0 Data segment may not be written into. W = 1 Data segment may be written into.
	3	Executable (E)	E = 1	Descriptor type is code segment: C = 1 Code segment may only be executed when CPL ≥ DPL and CPL remains unchanged.
	2	Conforming (C)	C = 1	If Code Segment (S = 1, E = 1)
	1	Readable (R)	R = 0	R = 0 Code segment may not be read. R = 1 Code segment may be read.
	0	Accessed (A)	A = 0	A = 0 Segment has not been accessed. A = 1 Segment selector has been loaded into segment register or used by selector test instructions.



Code and data segments have several descriptor fields in common. The accessed A bit is set whenever the processor accesses a descriptor. The A bit is used by operating systems to keep usage statistics on a given segment. The G bit, or granularity bit, specifies if a segment length is byte-granular or page-granular. 80386 segments can be one mega-byte long with byte granularity ($G = 0$) or four gigabytes with page granularity ($G = 1$), (i.e., 220 pages each page is 4K bytes in length). The granularity is totally unrelated to paging. A 80386 system can consist of segments with byte granularity, and page granularity, whether or not paging is enabled.

The executable E bit tells if a segment is a code or data segment. A code segment ($E=1, S=1$) may be execute-only or execute/read as determined by the Read R bit. Code segments are execute only if $R=0$, and execute/read if $R=1$. Code segments may never be written into.

NOTE-

NOTE.
Code segments may be modified via aliases. Aliases are writeable data segments which occupy the same range of linear address space as the code segment.

The D bit indicates the default length for operands and effective addresses. If D=1 then 32-bit operands and 32-bit addressing modes are assumed. If D=0 then 16-bit operands and 16-bit addressing modes are assumed. Therefore all existing 286 code segments will execute on the 80386 assuming the D bit is set 0.

Another attribute of code segments is determined by the conforming C bit. Conforming segments, C = 1, can be executed and shared by programs at different privilege levels. (See section 4.4 Protection.)

Segments identified as data segments ($E = 0$, $S = 1$) are used for two types of 80386 segments, stack and data segments. The expansion direction (**ED**) bit specifies if a segment expands downward (stack) or upward (data). If a segment is a stack segment all offsets must be greater than the segment limit. On a data segment all offsets must be less than or equal to the limit. In other words, stack segments start at the base linear address plus the maximum segment limit and grow down to the base linear address plus the limit. On the other hand, data segments start at the base linear address and expand to the base linear address plus limit.

The write W bit controls the ability to write into a segment. Data segments are read-only if W=0. The stack segment must have W=1.

The B bit controls the size of the stack pointer register. If B = 1, then PUSHes, POPs, and CALLs all use the 32-bit ESP register for stack references and assume an upper limit of FFFFFFFFh. If B = 0, stack instructions all use the 16-bit SP register and assume an upper limit of FFFFh.

4.3.4.3 SYSTEM DESCRIPTOR FORMATS

System segments describe information about operating system tables, tasks, and gates. Figure 4-7 shows the general format of system segment descriptors, and the various types of system segments. 80386 system descriptors contain a 32-bit base linear address and a 20-bit segment limit. 80286 system descriptors have a 24-bit base address and a 16-bit segment limit. 80286 system descriptors are identified by the upper 16 bits being all zero.

SEGMENT BASE 15 . . . 0				SEGMENT LIMIT 15 . . . 0								
BASE 31 . . . 24		G	0	0	0	LIMIT 19 . . . 16	P	DPL	0	TYPE	BASE 23 . . . 16	
Type	Defines						Type	Defines				
0	Invalid						8	Invalid				
1	Available 286 TSS						9	Available 386 TSS				
2	LDT						A	Undefined (Intel Reserved)				
3	Busy 286 TSS						B	Busy 386 TSS				
4	286 Call Gate						C	386 Call Gate				
5	Task Gate (for 286 or 386 Task)						D	Undefined (Intel Reserved)				
6	286 Interrupt Gate						E	386 Interrupt Gate				
7	286 Trap Gate						F	386 Trap Gate				

Figure 4-7. System Segments Descriptors



80386

ADVANCE INFORMATION

4.3.4.4 LDT DESCRIPTORS (S=0, TYPE=2)

LDT descriptors (S=0 TYPE=2) contain information about Local Descriptor Tables. LDTs contain a table of segment descriptors, unique to a particular task. Since the instruction to load the LDTR is only available at privilege level 0, the DPL field is ignored. LDT descriptors are only allowed in the Global Descriptor Table (GDT).

4.3.4.5 TSS DESCRIPTORS (S=0, TYPE=1, 3, 9, B)

A Task State Segment (TSS) descriptor contains information about the location, size, and privilege level of a Task State Segment (TSS). A TSS in turn is a special fixed format segment which contains all the state information for a task and a linkage field to permit nesting tasks. The TYPE field is used to indicate whether the task is currently BUSY (i.e. on a chain of active tasks) or the TSS is available. The TYPE field also indicates if the segment contains a 286 or a 386 TSS. The Task Register (TR) contains the selector which points to the current Task State Segment.

4.3.4.6 GATE DESCRIPTORS (S=0, TYPE=4-7, C, F)

Gates are used to control access to entry points within the target code segment. The various types of

gate descriptors are call gates, task gates, interrupt gates, and trap gates. Gates provide a level of indirection between the source and destination of the control transfer. This indirection allows the processor to automatically perform protection checks. It also allows system designers to control entry points to the operating system. Call gates are used to change privilege levels (see section 4.4 Protection), task gates are used to perform a task switch, and interrupt and trap gates are used to specify interrupt service routines.

Figure 4-8 shows the format of the four types of gate descriptors. Call gates are primarily used to transfer program control to a more privileged level. The call gate descriptor consists of three fields: the access byte, a long pointer (selector and offset) which points to the start of a routine and a word count which specifies how many parameters are to be copied from the caller's stack to the stack of the called routine. The word count field is only used by call gates when there is a change in the privilege level, other types of gates ignore the word count field.

Interrupt and trap gates use the destination selector and destination offset fields of the gate descriptor as a pointer to the start of the interrupt or trap handler routines. The difference between interrupt gates and trap gates is that the interrupt gate disables interrupts (resets the IF bit) while the trap gate does not.

31		24		16		8		5		0																																			
SELECTOR		OFFSET 15...0								0																																			
OFFSET 31...16		P	DPL	0	TYPE	0	0	0	WORD COUNT	+4																																			
Gate Descriptor Fields																																													
<table><thead><tr><th>Name</th><th>Type</th><th>Value</th><th>Description</th></tr></thead><tbody><tr><td rowspan="6">P</td><td>4</td><td>286 call gate</td><td></td></tr><tr><td>5</td><td>Task gate (for 286 or 386 task)</td><td></td></tr><tr><td>6</td><td>286 interrupt gate</td><td></td></tr><tr><td>7</td><td>286 trap gate</td><td></td></tr><tr><td>C</td><td>386 call gate</td><td></td></tr><tr><td>E</td><td>386 interrupt gate</td><td></td></tr><tr><td rowspan="2">DPL</td><td>F</td><td>386 trap gate</td><td></td></tr><tr><td>0</td><td>Descriptor contents are not valid</td><td></td></tr><tr><td>TYPE</td><td>1</td><td>Descriptor contents are valid</td><td></td></tr></tbody></table>												Name	Type	Value	Description	P	4	286 call gate		5	Task gate (for 286 or 386 task)		6	286 interrupt gate		7	286 trap gate		C	386 call gate		E	386 interrupt gate		DPL	F	386 trap gate		0	Descriptor contents are not valid		TYPE	1	Descriptor contents are valid	
Name	Type	Value	Description																																										
P	4	286 call gate																																											
	5	Task gate (for 286 or 386 task)																																											
	6	286 interrupt gate																																											
	7	286 trap gate																																											
	C	386 call gate																																											
	E	386 interrupt gate																																											
DPL	F	386 trap gate																																											
	0	Descriptor contents are not valid																																											
TYPE	1	Descriptor contents are valid																																											
DPL—least privileged level at which a task may access the gate. WORD COUNT 0-31—the number of parameters to copy from caller's stack to the called procedure's stack. The parameters are 32-bit quantities for 386 gates, and 16-bit quantities for 286 gates.																																													
DESTINATION	SELECTOR	16-bit selector	Selector to the target code segment or Selector to the target task state segment for task gate																																										
DESTINATION	OFFSET	offset 16-bit 286 32-bit 386	Entry point within the target code segment																																										

Figure 4-8. Gate Descriptor Formats

Task gates are used to switch tasks. Task gates may only refer to a task state segment (see section 4.4.6 **Task Switching**) therefore only the destination selector portion of a task gate descriptor is used, and the destination offset is ignored.

Exception 13 is generated when a destination selector does not refer to a correct descriptor type, i.e., a code segment for an interrupt, trap or call gate, a TSS for a task gate.

The access byte format is the same for all gate descriptors. P=1 indicates that the gate contents are valid. P=0 indicates the contents are not valid and causes exception 11 if referenced. DPL is the descriptor privilege level and specifies when this descriptor may be used by a task (see section 4.4 **Protection**). The S field, bit 4 of the access rights byte, must be 0 to indicate a system control descriptor. The type field specifies the descriptor type as indicated in Figure 4-8.

4.3.4.7 DIFFERENCES BETWEEN 386 AND 286 DESCRIPTORS

In order to provide operating system compatibility between the 80286 and 80386, the 386 supports all of the 80286 segment descriptors. Figure 4-9 shows the general format of an 80286 system segment descriptor. The only differences between 286 and 386 descriptor formats are that the values of the type fields, and the limit and base address fields have been expanded for the 386. The 80286 system segment descriptors contained a 24-bit base address and 16-bit limit, while the 386 system segment descriptors have a 32-bit base address, a 20-bit limit field, and a granularity bit.

By supporting 80286 system segments the 80386 is able to execute 286 application programs on a 80386 operating system. This is possible because the processor automatically understands which de-

scriptors are 286-style descriptors and which descriptors are 386-style descriptors. In particular, if the upper word of a descriptor is zero, then that descriptor is a 286-style descriptor.

The only other differences between 286-style descriptors and 386 descriptors is the interpretation of the word count field of call gates and the B bit. The word count field specifies the number of 16-bit quantities to copy for 286 call gates and 32-bit quantities for 386 call gates. The B bit controls the size of PUSHes when using a call gate; if B=0 PUSHes are 16 bits, if B=1 PUSHes are 32 bits.

4.3.4.8 SELECTOR FIELDS

A selector in Protected Mode has three fields: Local or Global Descriptor Table Indicator (TI), Descriptor Entry Index (Index), and Requestor (the selector's) Privilege Level (RPL) as shown in Figure 4-10. The TI bits select one of two memory-based tables of descriptors (the Global Descriptor Table or the Local Descriptor Table). The Index selects one of 8K descriptors in the appropriate descriptor table. The RPL bits allow high speed testing of the selector's privilege attributes.

4.3.4.9 SEGMENT DESCRIPTOR CACHE

In addition to the selector value, every segment register has a segment descriptor cache register associated with it. Whenever a segment register's contents are changed, the 8-byte descriptor associated with that selector is automatically loaded (cached) on the chip. Once loaded, all references to that segment use the cached descriptor information instead of reaccessing the descriptor. The contents of the descriptor cache are not visible to the programmer. Since descriptor caches only change when a segment register is changed, programs which modify the descriptor tables must reload the appropriate segment registers after changing a descriptor's value.

31	0
SEGMENT BASE 15...0	SEGMENT LIMIT 15...0 0
Intel Reserved Set to 0	P DPL S TYPE BASE + 4
BASE Base Address of the segment LIMIT The length of the segment Present Bit 1 = Present 0 = Not Present	DPL Descriptor Privilege Level 0-3 S System Descriptor 0 = System 1 = User TYPE Type of Segment

Figure 4-9. 286 Code and Data Segment Descriptors

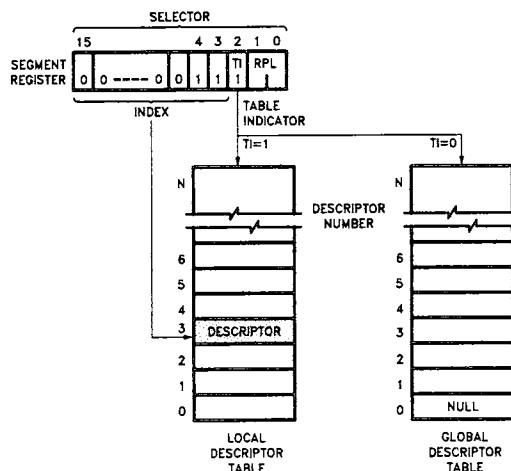


Figure 4-10. Example Descriptor Selection

4.3.4.10 SEGMENT DESCRIPTOR REGISTER SETTINGS

The contents of the segment descriptor cache vary depending on the mode the 80386 is operating in. When operating in Real Address Mode, the segment base, limit, and other attributes within the segment cache registers are defined as shown in Figure 4-11.

For compatibility with the 8086 architecture, the base is set to sixteen times the current selector value, the limit is fixed at 0000FFFFH, and the attributes are fixed so as to indicate the segment is present and fully usable. In Real Address Mode, the internal "privilege level" is always fixed to the highest level, level 0, so I/O and other privileged opcodes may be executed.

SEGMENT		DESCRIPTOR CACHE REGISTER CONTENTS									
		32 - BIT BASE (UPATED DURING SELECTOR LOAD INTO SEGMENT REGISTER)	32 - BIT LIMIT (FIXED)	OTHER ATTRIBUTES (FIXED)							
CONFORMING PRIVILEGE											
STACK SIZE											
EXECUTABLE											
WRITEABLE											
READABLE											
EXPANSION DIRECTION											
GRANULARITY											
ACCESSED											
PRIVILEGE LEVEL											
PRESENT											
		BASE	LIMIT								
CS	16X CURRENT CS SELECTOR*	0000FFFFH	Y 0 Y B U Y Y Y -N								
SS	16X CURRENT SS SELECTOR	0000FFFFH	Y 0 Y B U Y Y N W-								
DS	16X CURRENT DS SELECTOR	0000FFFFH	Y 0 Y B U Y Y N -								
ES	16X CURRENT ES SELECTOR	0000FFFFH	Y 0 Y B U Y Y N -								
FS	16X CURRENT FS SELECTOR	0000FFFFH	Y 0 Y B U Y Y N -								
GS	16X CURRENT GS SELECTOR	0000FFFFH	Y 0 Y B U Y Y N -								

231830-80

*Except the 32-bit CS base is initialized to FFFF000H after reset until first intersegment control transfer (e.g. intersegment CALL, or intersegment JMP, or INT). (See Figure 4-13 Example.)

Key:

Y	= yes	D	= expand down
N	= no	B	= byte granularity
0	= privilege level 0	P	= page granularity
1	= privilege level 1	W	= push/pop 16-bit words
2	= privilege level 2	F	= push/pop 32-bit dwords
3	= privilege level 3	-	= does not apply to that segment cache register
U	= expand up		

Figure 4-11. Segment Descriptor Caches for Real Address Mode
(Segment Limit and Attributes are Fixed)

When operating in Protected Mode, the segment base, limit, and other attributes within the segment cache registers are defined as shown in Figure 4-12. In Protected Mode, each of these fields are defined

according to the contents of the segment descriptor indexed by the selector value loaded into the segment register.

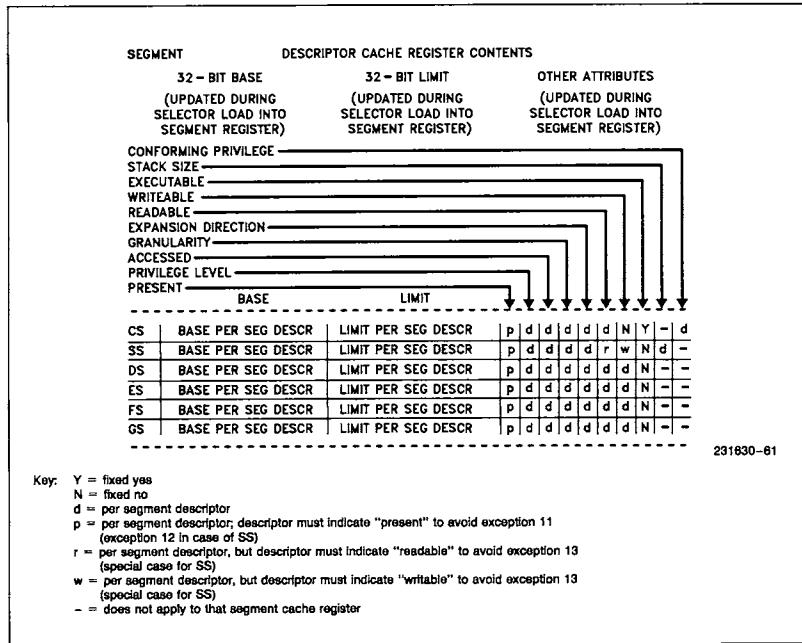


Figure 4-12. Segment Descriptor Caches for Protected Mode (Loaded per Descriptor)

When operating in a Virtual 8086 Mode within the Protected Mode, the segment base, limit, and other attributes within the segment cache registers are defined as shown in Figure 4-13. For compatibility with the 8086 architecture, the base is set to sixteen times the current selector value, the limit is fixed at

0000FFFFH, and the attributes are fixed so as to indicate the segment is present and fully usable. The virtual program executes at lowest privilege level, level 3, to allow trapping of all IOPL-sensitive instructions and level-0-only instructions.

SEGMENT		descriptor cache register contents															
32 - BIT BASE (UPDATED DURING SELECTOR LOAD INTO SEGMENT REGISTER)		32 - BIT LIMIT (FIXED)		OTHER ATTRIBUTES (FIXED)													
CONFIRMING PRIVILEGE																	
STACK SIZE																	
EXECUTABLE																	
WRITABLE																	
READABLE																	
EXPANSION DIRECTION																	
GRANULARITY																	
ACCESSED																	
PRIVILEGE LEVEL																	
PRESENT																	
		BASE	LIMIT														
CS	16X CURRENT CS SELECTOR	0000FFFFH	Y 3 Y B U Y Y Y - N														
SS	16X CURRENT SS SELECTOR	0000FFFFH	Y 3 Y B U Y Y N W -														
DS	16X CURRENT DS SELECTOR	0000FFFFH	Y 3 Y B U Y Y N - -														
ES	16X CURRENT ES SELECTOR	0000FFFFH	Y 3 Y B U Y Y N - -														
FS	16X CURRENT FS SELECTOR	0000FFFFH	Y 3 Y B U Y Y N - -														
GS	16X CURRENT GS SELECTOR	0000FFFFH	Y 3 Y B U Y Y N - -														

Key:

Y	= yes	D	= expand down
N	= no	B	= byte granularity
O	= privilege level 0	P	= page granularity
1	= privilege level 1	W	= push/pop 16-bit words
2	= privilege level 2	F	= push/pop 32-bit dwords
3	= privilege level 3	-	= does not apply to that segment cache register
U	= expand up		

231630-62

Figure 4-13. Segment Descriptor Caches for Virtual 8086 Mode within Protected Mode
(Segment Limit and Attributes are Fixed)

4.4 PROTECTION

4.4.1 Protection Concepts

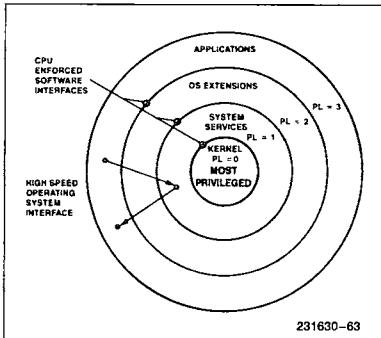


Figure 4-14. Four-Level Hierarchical Protection

The 80386 has four levels of protection which are optimized to support the needs of a multi-tasking operating system to isolate and protect user programs from each other and the operating system. The privilege levels control the use of privileged instructions, I/O instructions, and access to segments and segment descriptors. Unlike traditional microprocessor-based systems where this protection is achieved only through the use of complex external hardware and software the 80386 provides the protection as part of its integrated Memory Management Unit. The 80386 offers an additional type of protection on a page basis, when paging is enabled (See section 4.5.3 Page Level Protection).

The four-level hierarchical privilege system is illustrated in Figure 4-14. It is an extension of the user/supervisor privilege mode commonly used by mini-computers and, in fact, the user/supervisor mode is fully supported by the 80386 paging mechanism. The privilege levels (PL) are numbered 0 through 3. Level 0 is the most privileged or trusted level.

4.4.2 Rules of Privilege

The 80386 controls access to both data and procedures between levels of a task, according to the following rules.

- Data stored in a segment with privilege level p can be accessed only by code executing at a privilege level at least as privileged as p .
- A code segment/procedure with privilege level p can only be called by a task executing at the same or a lesser privilege level than p .

4.4.3 Privilege Levels

4.4.3.1 TASK PRIVILEGE

At any point in time, a task on the 80386 always executes at one of the four privilege levels. The Current Privilege Level (CPL) specifies the task's privilege level. A task's CPL may only be changed by control transfers through gate descriptors to a code segment with a different privilege level. (See section 4.4.4 Privilege Level Transfers) Thus, an application program running at PL = 3 may call an operating system routine at PL = 1 (via a gate) which would cause the task's CPL to be set to 1 until the operating system routine was finished.

4.4.3.2 SELECTOR PRIVILEGE (RPL)

The privilege level of a selector is specified by the RPL field. The RPL is the two least significant bits of the selector. The selector's RPL is only used to establish a less trusted privilege level than the current privilege level for the use of a segment. This level is called the task's effective privilege level (EPL). The EPL is defined as being the least privileged (i.e. numerically larger) level of a task's CPL and a selector's RPL. Thus, if selector's RPL = 0 then the CPL always specifies the privilege level for making an access using the selector. On the other hand if RPL = 3 then a selector can only access segments at level 3 regardless of the task's CPL. The RPL is most commonly used to verify that pointers passed to an operating system procedure do not access data that is of higher privilege than the procedure that originated the pointer. Since the originator of a selector can specify any RPL value, the Adjust RPL (ARPL) instruction is provided to force the RPL bits to the originator's CPL.

4.4.3.3 I/O PRIVILEGE AND I/O PERMISSION BITMAP

The I/O privilege level (IOPL, a 2-bit field in the EFLAG register) defines the least privileged level at which I/O instructions can be unconditionally performed. I/O instructions can be unconditionally performed when $CPL \leq IOPL$. (The I/O instructions are IN, OUT, INS, OUTS, REP INS, and REP OUTS.) When $CPL > IOPL$, and the current task is associated with a 286 TSS, attempted I/O instructions cause an exception 13 fault. When $CPL > IOPL$, and the current task is associated with a 386 TSS, the I/O Permission Bitmap (part of a 386 TSS) is consulted on whether I/O to the port is allowed, or an exception 13 fault is to be generated instead. For diagrams of the I/O Permission Bitmap, refer to Figures 4-15a and 4-15b. For further information on how the I/O Permission Bitmap is used in Protected Mode or in



Virtual 8086 Mode, refer to section 4.6.4 Protection and I/O Permission Bitmap.

The I/O privilege level (IOPL) also affects whether several other instructions can be executed or cause an exception 13 fault instead. These instructions are called "IOPL-sensitive" instructions and they are CLI and STI. (Note that the LOCK prefix is *not* IOPL-sensitive on the 80386.)

The IOPL also affects whether the IF (interrupts enable flag) bit can be changed by loading a value into the EFLAGS register. When CPL \leq IOPL, then the IF bit can be changed by loading a new value into the EFLAGS register. When CPL > IOPL, the IF bit cannot be changed by a new value POP'ed into (or otherwise loaded into) the EFLAGS register; the IF bit merely remains unchanged and no exception is generated.

Table 4-2. Pointer Test Instructions

Instruction	Operands	Function
ARPL	Selector, Register	Adjust Requested Privilege Level: adjusts the RPL of the selector to the numeric maximum of current selector RPL value and the RPL value in the register. Set zero flag if selector RPL was changed.
VERR	Selector	VERify for Read: sets the zero flag if the segment referred to by the selector can be read.
VERW	Selector	VERify for Write: sets the zero flag if the segment referred to by the selector can be written.
LSL	Register, Selector	Load Segment Limit: reads the segment limit into the register if privilege rules and descriptor type allow. Set zero flag if successful.
LAR	Register, Selector	Load Access Rights: reads the descriptor access rights byte into the register if privilege rules allow. Set zero flag if successful.

4.4.3.4 PRIVILEGE VALIDATION

The 80386 provides several instructions to speed pointer testing and help maintain system integrity by verifying that the selector value refers to an appropriate segment. Table 4-2 summarizes the selector validation procedures available for the 80386.

This pointer verification prevents the common problem of an application at PL = 3 calling a operating systems routine at PL = 0 and passing the operating system routine a "bad" pointer which corrupts a data structure belonging to the operating system. If the operating system routine uses the ARPL instruction to ensure that the RPL of the selector has no greater privilege than that of the caller, then this problem can be avoided.

4.4.3.5 DESCRIPTOR ACCESS

There are basically two types of segment accesses: those involving code segments such as control transfers, and those involving data accesses. Determining the ability of a task to access a segment involves the type of segment to be accessed, the instruction used, the type of descriptor used and CPL, RPL, and DPL as described above.

Any time an instruction loads data segment registers (DS, ES, FS, GS) the 80386 makes protection validation checks. Selectors loaded in the DS, ES, FS, GS registers must refer only to data segments or readable code segments. The data access rules are specified in section 4.2.2 Rules of Privilege. The only exception to those rules is readable conforming code segments which can be accessed at any privilege level.

Finally the privilege validation checks are performed. The CPL is compared to the EPL and if the EPL is more privileged than the CPL an exception 13 (general protection fault) is generated.

The rules regarding the stack segment are slightly different than those involving data segments. Instructions that load selectors into SS must refer to data segment descriptors for writeable data segments. The DPL and RPL must equal the CPL. All other descriptor types or a privilege level violation will cause exception 13. A stack not present fault causes exception 12. Note that an exception 11 is used for a not-present code or data segment.

4.4.4 Privilege Level Transfers

Inter-segment control transfers occur when a selector is loaded in the CS register. For a typical system most of these transfers are simply the result of a call

Table 4-3. Descriptor Types Used for Control Transfer

Control Transfer Types	Operation Types	Descriptor Referenced	Descriptor Table
Intersegment within the same privilege level	JMP, CALL, RET, IRET*	Code Segment	GDT/LDT
Intersegment to the same or higher privilege level Interrupt within task may change CPL	CALL	Call Gate	GDT/LDT
	Interrupt Instruction, Exception, External Interrupt	Trap or Interrupt Gate	IDT
Intersegment to a lower privilege level (changes task CPL)	RET, IRET*	Code Segment	GDT/LDT
	CALL, JMP	Task State Segment	GDT
Task Switch	CALL, JMP	Task Gate	GDT/LDT
	IRET** Interrupt Instruction, Exception, External Interrupt	Task Gate	IDT

*NT (Nested Task bit of flag register) = 0

**NT (Nested Task bit of flag register) = 1

or a jump to another routine. There are five types of control transfers which are summarized in Table 4-3. Many of these transfers result in a privilege level transfer. Changing privilege levels is done only via control transfers, by using gates, task switches, and interrupt or trap gates.

Control transfers can only occur if the operation which loaded the selector references the correct descriptor type. Any violation of these descriptor usage rules will cause an exception 13 (e.g. JMP through a call gate, or IRET from a normal subroutine call).

In order to provide further system security, all control transfers are also subject to the privilege rules.

The privilege rules require that:

- Privilege level transitions can only occur via gates.
- JMPs can be made to a non-conforming code segment with the same privilege or to a conforming code segment with greater or equal privilege.
- CALLs can be made to a non-conforming code segment with the same privilege or via a gate to a more privileged level.
- Interrupts handled within the task obey the same privilege rules as CALLs.
- Conforming Code segments are accessible by privilege levels which are the same or less privileged than the conforming-code segment's DPL.
- Both the requested privilege level (RPL) in the selector pointing to the gate and the task's CPL

must be of equal or greater privilege than the gate's DPL.

- The code segment selected in the gate must be the same or more privileged than the task's CPL.
- Return instructions that do not switch tasks can only return control to a code segment with same or less privilege.
- Task switches can be performed by a CALL, JMP, or INT which references either a task gate or task state segment who's DPL is less privileged or the same privilege as the old task's CPL.

Any control transfer that changes CPL within a task causes a change of stacks as a result of the privilege level change. The initial values of SS:ESP for privilege levels 0, 1, and 2 are retained in the task state segment (see section 4.4.6 Task Switching). During a JMP or CALL control transfer, the new stack pointer is loaded into the SS and ESP registers and the previous stack pointer is pushed onto the new stack.

When RETurning to the original privilege level, use of the lower-privileged stack is restored as part of the RET or IRET instruction operation. For subroutine calls that pass parameters on the stack and cross privilege levels, a fixed number of words (as specified in the gate's word count field) are copied from the previous stack to the current stack. The inter-segment RET instruction with a stack adjustment value will correctly restore the previous stack pointer upon return.

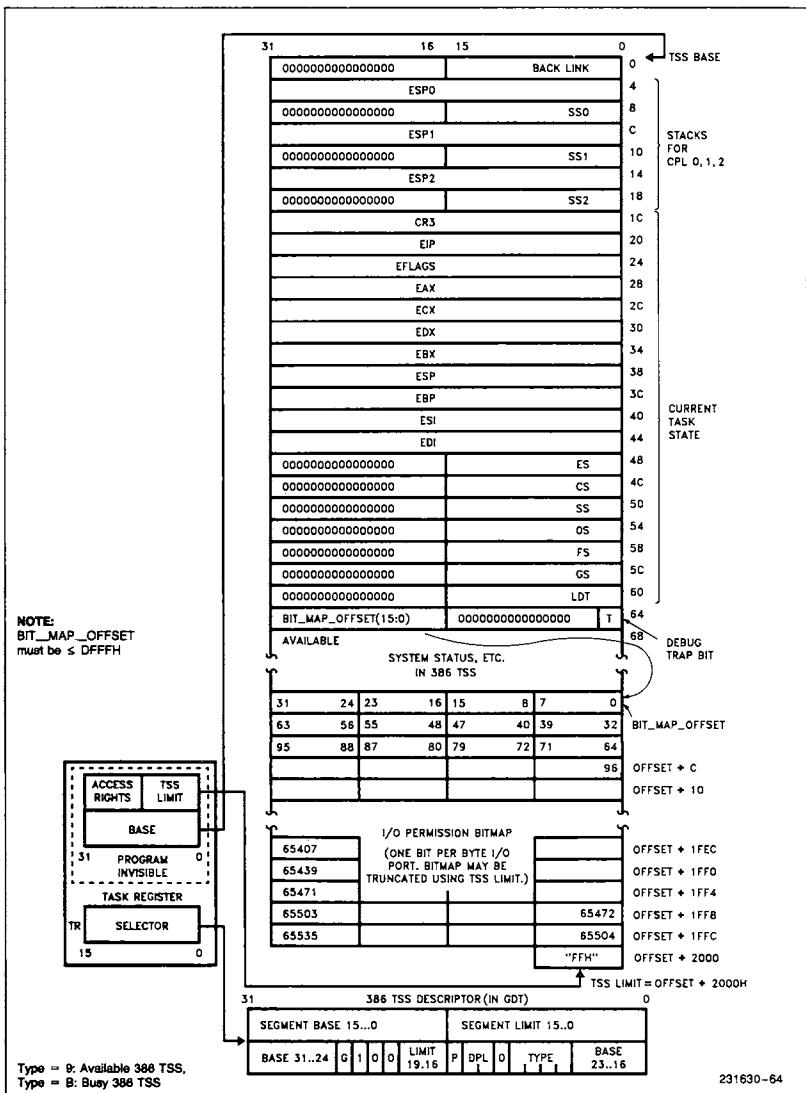


Figure 4-15a. 386 TSS and TSS Registers

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
31	1	1	1	1	0	1	1	0	0	0	0	1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0			
63	0	0	1	0	0	0	1	1	1	0	0	1	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1			
95	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
127	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	etc.																															

I/O Ports Accessible: 2 → 9, 12, 13, 15, 20 → 24, 27, 33, 34, 40, 41, 48, 50, 52, 53, 58 → 60, 62, 83, 95 → 127 231630-71

Figure 4-15b. Sample I/O Permission Bit Map

4.4.5 Call Gates

Gates provide protected, indirect CALLs. One of the major uses of gates is to provide a secure method of privilege transfers within a task. Since the operating system defines all of the gates in a system, it can ensure that all gates only allow entry into a few trusted procedures (such as those which allocate memory, or perform I/O).

Gate descriptors follow the data access rules of privilege; that is, gates can be accessed by a task if the EPL is equal to or more privileged than the gate descriptor's DPL. Gates follow the control transfer rules of privilege and therefore may only transfer control to a more privileged level.

Call Gates are accessed via a CALL instruction and are syntactically identical to calling a normal subroutine. When an inter-level 386 call gate is activated, the following actions occur.

1. Load CS:EIP from gate check for validity
2. SS is pushed zero-extended to 32 bits
3. ESP is pushed
4. Copy Word Count 32-bit parameters from the old stack to the new stack
5. Push Return address on stack

The procedure is identical for 286 Call gates, except that 16-bit parameters are copied and 16-bit registers are pushed.

Interrupt Gates and Trap gates work in a similar fashion as the call gates, except there is no copying of parameters. The only difference between Trap and Interrupt gates is that control transfers through an interrupt gate disable further interrupts (i.e. the IF bit is set to 0), and Trap gates leave the interrupt status unchanged.

4.4.6 Task Switching

A very important attribute of any multi-tasking/multi-user operating systems is its ability to rapidly switch between tasks or processes. The 80386 directly supports this operation by providing a task switch

instruction in hardware. The 80386 task switch operation saves the entire state of the machine (all of the registers, address space, and a link to the previous task), loads a new execution state, performs protection checks, and commences execution in the new task, in about 17 microseconds. Like transfer of control via gates, the task switch operation is invoked by executing an inter-segment JMP or CALL instruction which refers to a Task State Segment (TSS), or a task gate descriptor in the GDT or LDT. An INT n instruction, exception, trap, or external interrupt may also invoke the task switch operation if there is a task gate descriptor in the associated IDT descriptor slot.

The TSS descriptor points to a segment (see Figure 4-15) containing the entire 80386 execution state while a task gate descriptor contains a TSS selector. The 80386 supports both 286 and 386 style TSSs. Figure 4-16 shows a 286 TSS. The limit of a 386 TSS must be greater than 0064H (002BH for a 286 TSS), and can be as large as 4 Gigabytes. In the additional TSS space, the operating system is free to store additional information such as the reason the task is inactive, time the task has spent running, and open files belong to the task.

Each task must have a TSS associated with it. The current TSS is identified by a special register in the 80386 called the Task State Segment Register (TR). This register contains a selector referring to the task state segment descriptor that defines the current TSS. A hidden base and limit register associated with TR are loaded whenever TR is loaded with a new selector. Returning from a task is accomplished by the IRET instruction. When IRET is executed, control is returned to the task which was interrupted. The current executing task's state is saved in the TSS and the old task state is restored from its TSS.

Several bits in the flag register and machine status word (CR0) give information about the state of a task which are useful to the operating system. The Nested Task (NT) (bit 14 in EFLAGS) controls the function of the IRET instruction. If NT = 0, the IRET instruction performs the regular return; when NT = 1, IRET performs a task switch operation back to the previous task. The NT bit is set or reset in the following fashion:

15	0
BACK LINK SELECTOR TO TSS	0
SP FOR CPL 0	2
SS FOR CPL 0	4
SP FOR CPL 1	6
SS FOR CPL 1	8
SP FOR CPL 2	A
SS FOR CPL 2	C
IP (ENTRY POINT)	E
FLAGS	10
AX	12
CX	14
DX	16
BX	18
SP	1A
BP	IC
SI	IE
DI	20
ES SELECTOR	22
CS SELECTOR	24
SS SELECTOR	26
DS SELECTOR	28
TASK'S LDT SELECTOR	2A
AVAILABLE	

231630-65

Figure 4-16. 286 TSS

When a CALL or INT instruction initiates a task switch, the new TSS will be marked busy and the back link field of the new TSS set to the old TSS selector. The NT bit of the new task is set by CALL or INT initiated task switches. An interrupt that does not cause a task switch will clear NT. (The NT bit will be restored after execution of the interrupt handler) NT may also be set or cleared by POPF or IRET instructions.

The 386 task state segment is marked busy by changing the descriptor type field from TYPE 9H to TYPE BH. A 286 TSS is marked busy by changing the descriptor type field from TYPE 1 to TYPE 3. Use of a selector that references a busy task state segment causes an exception 13.

The Virtual Mode (VM) bit 17 is used to indicate if a task, is a virtual 8086 task. If VM = 1, then the tasks will use the Real Mode addressing mechanism. The virtual 8086 environment is only entered and exited via a task switch (see section 4.6 Virtual Mode).

The coprocessor's state is not automatically saved when a task switch occurs, because the incoming

task may not use the coprocessor. The Task Switched (TS) Bit (bit 3 in the CR0) helps deal with the coprocessor's state in a multi-tasking environment. Whenever the 80386 switches tasks, it sets the TS bit. The 80386 detects the first use of a processor extension instruction after a task switch and causes the processor extension not available exception 7. The exception handler for exception 7 may then decide whether to save the state of the coprocessor. A processor extension not present exception (7) will occur when attempting to execute an ESC or WAIT instruction if the Task Switched and Monitor coprocessor extension bits are both set (i.e. TS = 1 and MP = 1).

The T bit in the 386 TSS indicates that the processor should generate a debug exception when switching to a task. If T = 1 then upon entry to a new task a debug exception 1 will be generated.

4.4.7 Initialization and Transition to Protected Mode

Since the 80386 begins executing in Real Mode immediately after RESET it is necessary to initialize the system tables and registers with the appropriate values.

The GDT and IDT registers must refer to a valid GDT and IDT. The IDT should be at least 256 bytes long, and GDT must contain descriptors for the initial code, and data segments. Figure 4-17 shows the tables and Figure 4-18 the descriptors needed for a simple Protected Mode 80386 system. It has a single code and single data/stack segment each four gigabytes long and a single privilege level PL = 0.

The actual method of enabling Protected Mode is to load CR0 with the PE bit set, via the MOV CR0, R/M instruction. This puts the 80386 in Protected Mode.

After enabling Protected Mode, the next instruction should execute an intersegment JMP to load the CS register and flush the instruction decode queue. The final step is to load all of the data segment registers with the initial selector values.

An alternate approach to entering Protected Mode which is especially appropriate for multi-tasking operating systems, is to use the built in task-switch to load all of the registers. In this case the GDT would contain two TSS descriptors in addition to the code and data descriptors needed for the first task. The first JMP instruction in Protected Mode would jump to the TSS causing a task switch and loading all of the registers with the values stored in the TSS. The Task State Segment Register should be initialized to point to a valid TSS descriptor since a task switch saves the state of the current task in a task state segment.

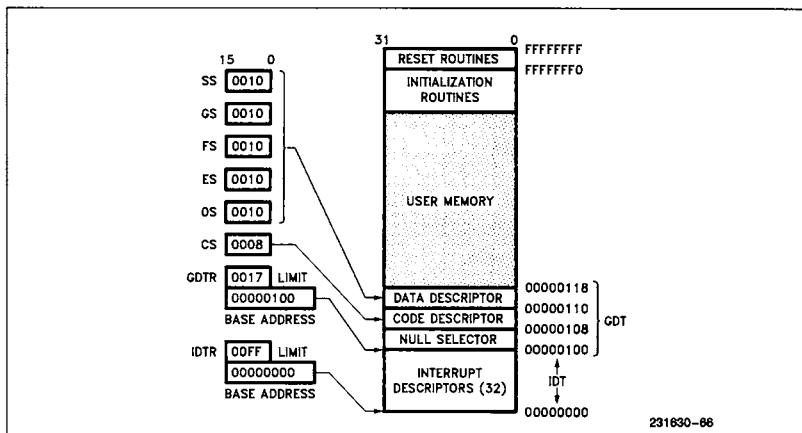


Figure 4-17. Simple Protected System

		SEGMENT BASE 15...0 0118 (H)				SEGMENT LIMIT 15...0 FFFF (H)						
		2	BASE 31...24 00 (H)	G 1	D 1	0 0	0 0	LIMIT 19.16 F (H)	1 0 0 1	0 0 1 0	BASE 23...16 00 (H)	
		CODE DESCRIPTOR	SEGMENT BASE 15...0 0118 (H)				SEGMENT LIMIT 15...0 FFFF (H)					
		1	BASE 31...24 00 (H)	G 1	D 1	0 0	0 0	LIMIT 19.16 F (H)	1 0 0 1	1 0 1 0	BASE 23...16 00 (H)	
		0	NULL DESCRIPTOR									
		31	24		16		15		8		0	

Figure 4-18. GDT Descriptors for Simple System

4.4.8 Tools for Building Protected Systems

In order to simplify the design of a protected multi-tasking system, Intel provides a tool which allows the system designer an easy method of constructing the data structures needed for a Protected Mode 80386 system. This tool is the builder BLD-386™. BLD-386 lets the operating system writer specify all of the segment descriptors discussed in the previous sections (LDTs, IDTs, GDTs, Gates, and TSSs) in a high-level language.

4.5 PAGING

4.5.1 Paging Concepts

Paging is another type of memory management useful for virtual memory multitasking operating systems. Unlike segmentation which modularizes programs and data into variable length segments, pa-

ging divides programs into multiple uniform size pages. Pages bear no direct relation to the logical structure of a program. While segment selectors can be considered the logical "name" of a program module or data structure, a page most likely corresponds to only a portion of a module or data structure.

By taking advantage of the locality of reference displayed by most programs, only a small number of pages from each active task need be in memory at any one moment.

4.5.2 Paging Organization

4.5.2.1 PAGE MECHANISM

The 80386 uses two levels of tables to translate the linear address (from the segmentation unit) into a physical address. There are three components to the paging mechanism of the 80386: the page directory, the page tables, and the page itself (page frame). All memory-resident elements of the 80386 paging mechanism are the same size, namely, 4K bytes. A uniform size for all of the elements simplifies memory allocation and reallocation schemes, since there is no problem with memory fragmentation. Figure 4-19 shows how the paging mechanism works.

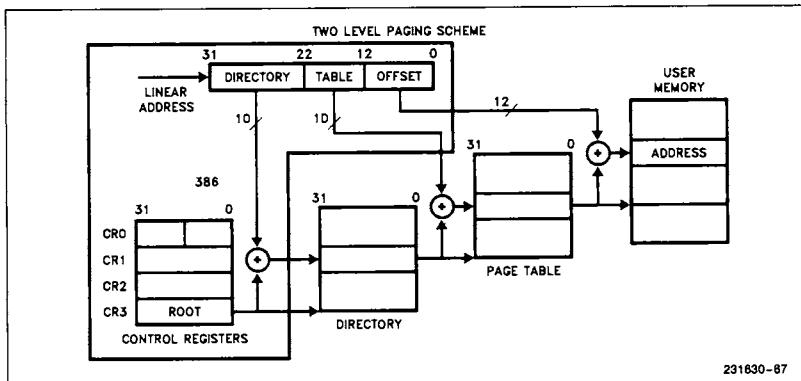


Figure 4-19. Paging Mechanism

31	12	11	10	9	8	7	6	5	4	3	2	1	0
PAGE TABLE ADDRESS 31..12	OS RESERVED			0	0	D	A	0	0	U — S	R — W	P	

Figure 4-20. Page Directory Entry (Points to Page Table)

31	12	11	10	9	8	7	6	5	4	3	2	1	0
PAGE FRAME ADDRESS 31..12	OS RESERVED	0	0	D	A	0	0	U — S	R — W	P			

Figure 4-21. Page Table Entry (Points to Page)

4.5.2.4 PAGE TABLES

Each Page Table is 4K bytes and holds up to 1024 Page Table Entries. Page Table Entries contain the starting address of the page frame and statistical information about the page (see Figure 4-21). Address bits A12-A21 are used as an index to select one of the 1024 Page Table Entries. The 20 upper-bit page frame address is concatenated with the lower 12 bits of the linear address to form the physical address. Page tables can be shared between tasks and swapped to disks.

4.5.2.5 PAGE DIRECTORY/TABLE ENTRIES

The lower 12 bits of the Page Table Entries and Page Directory Entries contain statistical information about pages and page tables respectively. The P (Present) bit 0 indicates if a Page Directory or Page Table entry can be used in address translation. If P = 1 the entry can be used for address translation if P = 0 the entry can not be used for translation, and all of the other bits are available for use by the software. For example the remaining 31 bits could be used to indicate where on the disk the page is stored.

The A (Accessed) bit 5, is set by the 80386 for both types of entries before a read or write access occurs to an address covered by the entry. The D (Dirty) bit 6 is set to 1 before a write to an address covered by that page table entry occurs. The D bit is undefined for Page Directory Entries. When the P, A and D bits are updated by the 80386, the processor generates a Read-Modify-Write cycle which locks the bus and prevents conflicts with other processors or peripherals. Software which modifies these bits should use the LOCK prefix to ensure the integrity of the page tables in multi-master systems.

The 3 bits marked OS Reserved in Figure 4-20 and Figure 4-21 (bits 9-11) are software definable. OSs are free to use these bits for whatever purpose they wish. An example use of the OS Reserved bits would be to store information about page aging. By keeping track of how long a page has been in memory since being accessed, an operating system can implement a page replacement algorithm like Least Recently Used.

The (User/Supervisor) U/S bit 2 and the (Read/Write) R/W bit 1 are used to provide protection attributes for individual pages.

4.5.3 Page Level Protection (R/W, U/S Bits)

The 80386 provides a set of protection attributes for paging systems. The paging mechanism distinguishes between two levels of protection: User which corresponds to level 3 of the segmentation based protection, and supervisor which encompasses all of the other protection levels (0, 1, 2). Programs executing at Level 0, 1 or 2 bypass the page protection, although segmentation based protection is still enforced by the hardware.

The U/S and R/W bits are used to provide User/Supervisor and Read/Write protection for individual pages or for all pages covered by a Page Table Directory Entry. The U/S and R/W bits in the first level Page Directory Table apply to all pages described by the page table pointed to by that directory entry. The U/S and R/W bits in the second level Page Table Entry apply only to the page described by that entry. The U/S and R/W bits for a given page are obtained by taking the most restrictive of the U/S and R/W from the Page Directory Table Entries and the Page Table Entries and using these bits to address the page.

Example: If the U/S and R/W bits for the Page Directory entry were 10 and the U/S and R/W bits for the Page Table Entry were 01, the access rights for the page would be 01, the numerically smaller of the two. Table 4-4 shows the affect of the U/S and R/W bits on accessing memory.

Table 4-4. Protection Provided by R/W and U/S

U/S	R/W	Permitted Level 3	Permitted Access Levels 0, 1, or 2
0	0	None	Read/Write
0	1	None	Read/Write
1	0	Read-Only	Read/Write
1	1	Read/Write	Read/Write

However a given segment can be easily made read-only for level 0, 1, or 2 via the use of segmented protection mechanisms. (Section 4.4 Protection).

4.5.4 Translation Lookaside Buffer

The 80386 paging hardware is designed to support demand paged virtual memory systems. However, performance would degrade substantially if the processor was required to access two levels of tables for every memory reference. To solve this problem, the 80386 keeps a cache of the most recently accessed pages, this cache is called the Translation Lookaside Buffer (TLB). The TLB is a four-way set associative 32-entry page table cache. It automatically keeps the most commonly used Page Table Entries in the processor. The 32-entry TLB coupled with a 4K page size, results in coverage of 128K bytes of memory addresses. For many common multi-tasking systems, the TLB will have a hit rate of about 98%. This means that the processor will only have to access the two-level page structure on 2% of all memory references. Figure 4-22 illustrates how the TLB complements the 80386's paging mechanism.

4.5.5 Paging Operation

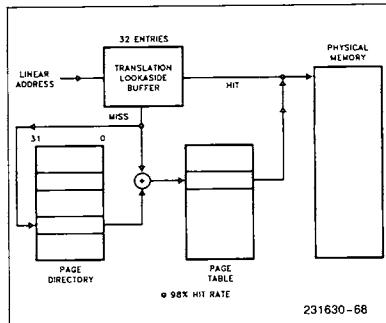


Figure 4-22. Translation Lookaside Buffer

The paging hardware operates in the following fashion. The paging unit hardware receives a 32-bit linear address from the segmentation unit. The upper 20 linear address bits are compared with all 32 entries in the TLB to determine if there is a match. If there is a match (i.e. a TLB hit), then the 32-bit physical address is calculated and will be placed on the address bus.

However, if the page table entry is not in the TLB, the 80386 will read the appropriate Page Directory

Entry. If P = 1 on the Page Directory Entry indicating that the page table is in memory, then the 80386 will read the appropriate Page Table Entry and set the Access bit. If P = 1 on the Page Table Entry indicating that the page is in memory, the 80386 will update the Access and Dirty bits as needed and fetch the operand. The upper 20 bits of the linear address, read from the page table, will be stored in the TLB for future accesses. However, if P = 0 for either the Page Directory Entry or the Page Table Entry, then the processor will generate a page fault, an Exception 14.

The processor will also generate an exception 14, page fault, if the memory protection attributes (i.e. U/S or R/W) (e.g. trying to write to a read-only page). CR2 will hold the linear address which caused the page fault. Since Exception 14 is classified as a fault, CS: EIP will point to the instruction causing the page fault. The 16-bit error code pushed as part of the page fault handler will contain status bits which indicate the cause of the page fault.

The 16-bit error code is used by the operating system to determine how to handle the page fault. Figure 4-23A shows the format of the page-fault error code and the interpretation of the bits.

NOTE:

Even though the bits in the error code (U/S, W/R, and P) have similar names as the bits in the Page Directory/Table Entries, the interpretation of the error code bits is different. Figure 4-23B indicates what type of access caused the page fault.

15	3	2	1	0
U	U	U	U	U/S W/R P

Figure 4-23A. Page Fault Error Code Format

U/S: The U/S bit indicates whether the access causing the fault occurred when the processor was executing in User Mode (U/S = 1) or in Supervisor mode (U/S = 0)

W/R: The W/R bit indicates whether the access causing the fault was a Read (W/R = 0) or a Write (W/R = 1)

P: The P bit indicates whether a page fault was caused by a not-present page (P = 0), or by a page level protection violation (P = 1)

U: UNDEFINED

U/S	W/R	Access Type
0	0	Supervisor* Read
0	1	Supervisor Write
1	0	User Read
1	1	User Write

*Descriptor table access will fault with U/S = 0, even if the program is executing at level 3.

Figure 4-23B. Type of Access Causing Page Fault

4.5.6 Operating System Responsibilities

The 80386 takes care of the page address translation process, relieving the burden from an operating system in a demand-paged system. The operating system is responsible for setting up the initial page tables, and handling any page faults. The operating system also is required to invalidate (i.e. flush) the TLB when any changes are made to any of the page table entries. The operating system must reload CR3 to cause the TLB to be flushed.

Setting up the tables is simply a matter of loading CR3 with the address of the Page Directory, and allocating space for the Page Directory and the Page Tables. The primary responsibility of the operating system is to implement a swapping policy and handle all of the page faults.

A final concern of the operating system is to ensure that the TLB cache matches the information in the paging tables. In particular, any time the operating system sets the P present bit of page table entry to zero, the TLB must be flushed. Operating systems may want to take advantage of the fact that CR3 is stored as part of a TSS, to give every task or group of tasks its own set of page tables.

4.6 VIRTUAL 8086 ENVIRONMENT

4.6.1 Executing 8086 Programs

The 80386 allows the execution of 8086 application programs in both Real Mode and in the Virtual 8086 Mode (Virtual Mode). Of the two methods, Virtual 8086 Mode offers the system designer the most flexibility. The Virtual 8086 Mode allows the execution of 8086 applications, while still allowing the system designer to take full advantage of the 80386 protection mechanism. In particular, the 80386 allows the simultaneous execution of 8086 operating systems and its applications, and an 80386 operat-

ing system and both 80286 and 80386 applications. Thus, in a multi-user 80386 computer, one person could be running an MS-DOS spreadsheet, another person using MS-DOS, and a third person could be running multiple Unix utilities and applications. Each person in this scenario would believe that he had the computer completely to himself. Figure 4-24 illustrates this concept.

4.6.2 Virtual 8086 Mode Addressing Mechanism

One of the major differences between 80386 Real and Protected modes is how the segment selectors are interpreted. When the processor is executing in Virtual 8086 Mode the segment registers are used in an identical fashion to Real Mode. The contents of the segment register is shifted left 4 bits and added to the offset to form the segment base linear address.

The 80386 allows the operating system to specify which programs use the 8086 style address mechanism, and which programs use Protected Mode addressing, on a per task basis. Through the use of paging, the one megabyte address space of the Virtual Mode task can be mapped to anywhere in the 4 gigabyte linear address space of the 80386. Like Real Mode, Virtual Mode effective addresses (i.e., segment offsets) that exceed 64K byte will cause an exception 13. However, these restrictions should not prove to be important, because most tasks running in Virtual 8086 Mode will simply be existing 8086 application programs.

4.6.3 Paging In Virtual Mode

The paging hardware allows the concurrent running of multiple Virtual Mode tasks, and provides protection and operating system isolation. Although it is not strictly necessary to have the paging hardware enabled to run Virtual Mode tasks, it is needed in order to run multiple Virtual Mode tasks or to relocate the address space of a Virtual Mode task to physical address space greater than one megabyte.

The paging hardware allows the 20-bit linear address produced by a Virtual Mode program to be divided into up to 256 pages. Each one of the pages can be located anywhere within the maximum 4 gigabyte physical address space of the 80386. In addition, since CR3 (the Page Directory Base Register) is loaded by a task switch, each Virtual Mode task can use a different mapping scheme to map pages to different physical locations. Finally, the paging hardware allows the sharing of the 8086 operating system code between multiple 8086 applications.

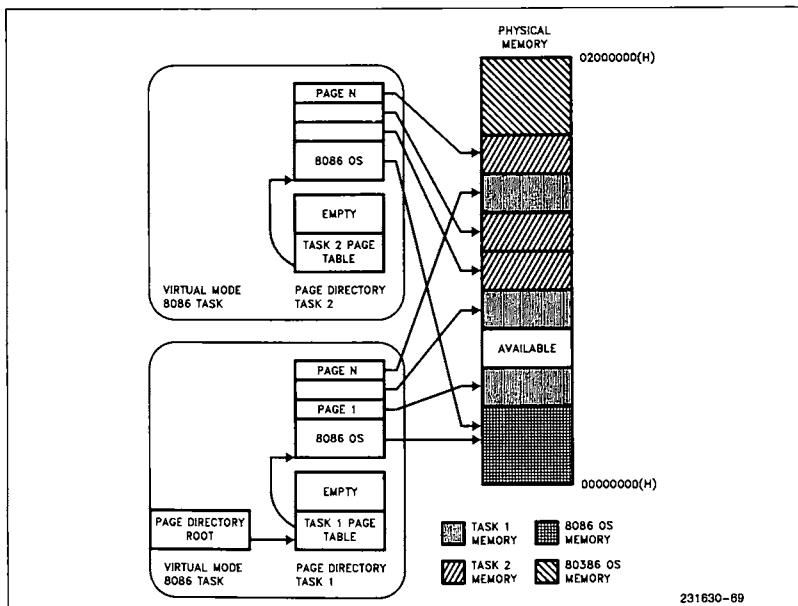


Figure 4-24 shows how the 80386 paging hardware enables multiple 8086 programs to run under a virtual memory demand paged system.

LMSW; MOV CRn,reg; MOV reg,CRn.
CLTS;
HLT;

4.6.4 Protection and I/O Permission Bitmap

All Virtual 8086 Mode programs execute at privilege level 3, the level of least privilege. As such, Virtual 8086 Mode programs are subject to all of the protection checks defined in Protected Mode. (This is different from Real Mode which implicitly is executing at privilege level 0, the level of greatest privilege.) Thus, an attempt to execute a privileged instruction when in Virtual 8086 Mode will cause an exception 13 fault.

The following are privileged instructions, which may be executed only at Privilege Level 0. Therefore, attempting to execute these instructions in Virtual 8086 Mode (or anytime CPL > 0) causes an exception 13 fault:

LIDT; MOV DRn,reg; MOV reg,DRn;
LGDT; MOV TRn,reg; MOV reg,TRn;

Several instructions, particularly those applying to the multitasking model and protection model, are available only in Protected Mode. Therefore, attempting to execute the following instructions in Real Mode or in Virtual 8086 Mode generates an exception 6 fault:

LTR; STR;
LLDT; SLDI;
LAR; VERR;
LSL; VERW;
ARPL.

The instructions which are IOPL-sensitive in Protected Mode are:

IN; STI;
OUT; CLI
INS;
OUTS;
REP INS;
REP OUTS;



In Virtual 8086 Mode, a slightly different set of instructions are made IOPL-sensitive. The following instructions are IOPL-sensitive in Virtual 8086 Mode:

```
INT n;      STI;
PUSHF;     CLI;
POPF;      IRET
```

The PUSHF, POPF, and IRET instructions are IOPL-sensitive in Virtual 8086 Mode only. This provision allows the IF flag (interrupt enable flag) to be virtualized to the Virtual 8086 Mode program. The INT n software interrupt instruction is also IOPL-sensitive in Virtual 8086 Mode. Note, however, that the INT 3 (opcode 0CCH), INTO, and BOUND instructions are not IOPL-sensitive in Virtual 8086 mode (they aren't IOPL sensitive in Protected Mode either).

Note that the I/O instructions (IN, OUT, INS, OUTS, REP INS, and REP OUTS) are not IOPL-sensitive in Virtual 8086 mode. Rather, the I/O instructions become automatically sensitive to the **I/O Permission Bitmap** contained in the **386 Task State Segment**. The I/O Permission Bitmap, automatically used by the 80386 in Virtual 8086 Mode, is illustrated by Figures 4.15a and 4.15b.

The I/O Permission Bitmap can be viewed as a 0-64 Kbit bit string, which begins in memory at offset Bit_Map_Offset in the current TSS. Bit_Map_Offset must be \leq DFFFH so the entire bit map and the byte FFH which follows the bit map are all at offsets \leq FFFFH from the TSS base. The 16-bit pointer Bit_Map_Offset (15:0) is found in the word beginning at offset 66H (102 decimal) from the TSS base, as shown in Figure 4.15a.

Each bit in the I/O Permission Bitmap corresponds to a single byte-wide I/O port, as illustrated in Figure 4.15a. If a bit is 0, I/O to the corresponding byte-wide port can occur without generating an exception. Otherwise the I/O instruction causes an exception 13 fault. Since every byte-wide I/O port must be protectable, all bits corresponding to a word-wide or dword-wide port must be 0 for the word-wide or dword-wide I/O to be permitted. If all the referenced bits are 0, the I/O will be allowed. If any referenced bits are 1, the attempted I/O will cause an exception 13 fault.

Due to the use of a pointer to the base of the I/O Permission Bitmap, the bitmap may be located anywhere within the TSS, or may be ignored completely by pointing the Bit_Map_Offset (15:0) beyond the limit of the TSS segment. In the same manner, only a small portion of the 64K I/O space need have an associated map bit, by adjusting the TSS limit to truncate the bitmap. This eliminates the commitment of 8K of memory when a complete bitmap is not required, while allowing the fully general case if desired.

EXAMPLE OF BITMAP FOR I/O PORTS 0-255: Setting the TSS limit to {bit_Map_Offset + 31 + 1**} [** see note below] will allow a 32-byte bitmap for the I/O ports #0-255, plus a terminator byte of all 1's [** see note below]. This allows the I/O bitmap to control I/O Permission to I/O port 0-255 while causing an exception 13 fault on attempted I/O to any I/O port 256 through 65,565.

****IMPORTANT IMPLEMENTATION NOTE:** Beyond the last byte of I/O mapping information in the I/O Permission Bitmap must be a byte containing all 1's. The byte of all 1's must be within the limit of the 386 TSS segment (see Figure 4-15a).

4.6.5 Interrupt Handling

In order to fully support the emulation of an 8086 machine, interrupts in Virtual 8086 Mode are handled in a unique fashion. When running in Virtual Mode all interrupts and exceptions involve a privilege change back to the host 80386 operating system. The 80386 operating system determines if the interrupt comes from a Protected Mode application or from a Virtual Mode program by examining the VM bit in the EFLAGS image stored on the stack.

When a Virtual Mode program is interrupted and execution passes to the interrupt routine at level 0, the VM bit is cleared. However, the VM bit is still set in the EFLAG image on the stack.

The 80386 operating system in turn handles the exception or interrupt and then returns control to the 8086 program. The 80386 operating system may choose to let the 8086 operating system handle the interrupt or it may emulate the function of the interrupt handler. For example, many 8086 operating system calls are accessed by PUSHing parameters on the stack, and then executing an INT n instruction. If the IOPL is set to 0 then all INT n instructions will be intercepted by the 80386 operating system. The 80386 operating system could emulate the 8086 operating system's call. Figure 4-25 shows how the 80386 operating system could intercept an 8086 operating system's call to "Open a File".

An 80386 operating system can provide a Virtual 8086 Environment which is totally transparent to the application software via intercepting and then emulating 8086 operating system's calls, and intercepting IN and OUT instructions.

4.6.6 Entering and Leaving Virtual 8086 Mode

Virtual 8086 mode is entered by executing an IRET instruction (at CPL=0), or Task Switch (at any CPL) to a 386 task whose 386 TSS has a FLAGS image containing a 1 in the VM bit position while the proc-

essor is executing in Protected Mode. That is, one way to enter Virtual 8086 mode is to switch to a task with a 386 TSS that has a 1 in the VM bit in the EFLAGS image. The other way is to execute a 32-bit IRET instruction at privilege level 0, where the stack has a 1 in the VM bit in the EFLAGS image. POPF does not affect the VM bit, even if the processor is in Protected Mode or level 0, and so cannot be used to enter Virtual 8086 Mode. PUSHF always pushes a 0 in the VM bit, even if the processor is in Virtual 8086 Mode, so that a program cannot tell if it is executing in REAL mode, or in Virtual 8086 mode.

The VM bit can be set by executing an IRET instruction only at privilege level 0, or by any instruction or interrupt which causes a task switch in Protected Mode (with VM=1 in the new FLAGS image), and can be cleared only by an interrupt or exception in Virtual 8086 Mode. IRET and POPF instructions executed in REAL mode or Virtual 8086 mode will not change the value in the VM bit.

The transition out of virtual 8086 mode to 386 protected mode occurs only on receipt of an interrupt or exception (such as due to a sensitive instruction). In Virtual 8086 mode, all interrupts and exceptions vector through the protected mode IDT, and enter an interrupt handler in protected 386 mode. That is, as part of interrupt processing, the VM bit is cleared.

Because the matching IRET must occur from level 0, if an Interrupt or Trap Gate is used to field an interrupt or exception out of Virtual 8086 mode, the Gate must perform an inter-level interrupt only to level 0. Interrupt or Trap Gates through conforming segments, or through segments with DPL>0, will raise a GP fault with the CS selector as the error code.

4.6.6.1 TASK SWITCHES TO/FROM VIRTUAL 8086 MODE

Tasks which can execute in virtual 8086 mode must be described by a TSS with the new 386 format (TYPE 9 or 11 descriptor).

A task switch out of virtual 8086 mode will operate exactly the same as any other task switch out of a task with a 386 TSS. All of the programmer visible state, including the FLAGS register with the VM bit set to 1, is stored in the TSS. The segment registers in the TSS will contain 8086 segment base values rather than selectors.

A task switch into a task described by a 386 TSS will have an additional check to determine if the incoming task should be resumed in virtual 8086 mode. Tasks described by 286 format TSSs cannot be resumed in virtual 8086 mode, so no check is required there (the FLAGS image in 286 format TSS has only the low order 16 FLAGS bits). Before loading the segment register images from a 386 TSS, the FLAGS image is loaded, so that the segment

registers are loaded from the TSS image as 8086 segment base values. The task is now ready to resume in virtual 8086 execution mode.

4.6.6.2 TRANSITIONS THROUGH TRAP AND INTERRUPT GATES, AND IRET

A task switch is one way to enter or exit virtual 8086 mode. The other method is to exit through a Trap or Interrupt gate, as part of handling an interrupt, and to enter as part of executing an IRET instruction. The transition out must use a 386 Trap Gate (Type 14), or 386 Interrupt Gate (Type 15), which must point to a non-conforming level 0 segment (DPL=0) in order to permit the trap handler to IRET back to the Virtual 8086 program. The Gate must point to a non-conforming level 0 segment to perform a level switch to level 0 so that the matching IRET can change the VM bit. 386 gates must be used, since 286 gates save only the low 16 bits of the FLAGS register, so that the VM bit will not be saved on transitions through the 286 gates. Also, the 16-bit IRET (presumably) used to terminate the 286 interrupt handler will pop only the lower 16 bits from FLAGS, and will not affect the VM bit. The action taken for a 386 Trap or Interrupt gate if an interrupt occurs while the task is executing in virtual 8086 mode is given by the following sequence.

- (1) Save the FLAGS register in a temp to push later.
Turn off the VM and TF bits, and if the interrupt is serviced by an Interrupt Gate, turn off IF also.
- (2) Interrupt and Trap gates must perform a level switch from 3 (where the VM86 program executes) to level 0 (so IRET can return). This process involves a stack switch to the stack given in the TSS for privilege level 0. Save the Virtual 8086 Mode SS and ESP registers to push in a later step. The segment register load of SS will be done as a Protected Mode segment load, since the VM bit was turned off above.
- (3) Push the 8086 segment register values onto the new stack, in the order: GS, FS, DS, ES. These are pushed as 32-bit quantities, with undefined values in the upper 16 bits. Then load these 4 registers with null selectors (0).
- (4) Push the old 8086 stack pointer onto the new stack by pushing the SS register (as 32-bits, high bits undefined), then pushing the 32-bit ESP register saved above.
- (5) Push the 32-bit FLAGS register saved in step 1.
- (6) Push the old 8086 instruction pointer onto the new stack by pushing the CS register (as 32-bits, high bits undefined), then pushing the 32-bit EIP register.
- (7) Load up the new CS:EIP value from the interrupt gate, and begin execution of the interrupt routine in protected 386 mode.

The transition out of virtual 8086 mode performs a level change and stack switch, in addition to changing-

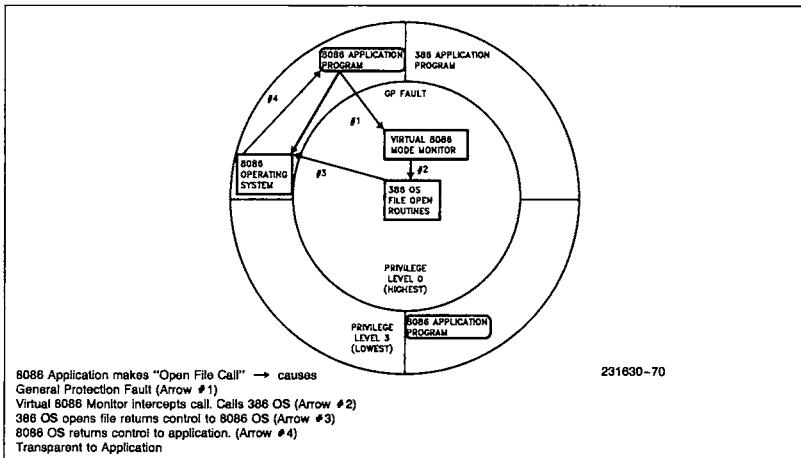


Figure 4-25. Virtual 8086 Environment Interrupt and Call Handling

ing back to protected mode. In addition, all of the 8086 segment register images are stored on the stack (behind the SS:ESP image), and then loaded with null (0) selectors before entering the interrupt handler. This will permit the handler to safely save and restore the DS, ES, FS, and GS registers as 286 selectors. This is needed so that interrupt handlers which don't care about the mode of the interrupted program can use the same prolog and epilog code for state saving (i.e. push all registers in prolog, pop all in epilog) regardless of whether or not a "native" mode or Virtual 8086 mode program was interrupted. Restoring null selectors to these registers before executing the IRET will not cause a trap in the interrupt handler. Interrupt routines which expect values in the segment registers, or return values in segment registers will have to obtain/return values from the 8086 register images pushed onto the new stack. They will need to know the mode of the interrupted program in order to know where to find/return segment registers, and also to know how to interpret segment register values.

The IRET instruction will perform the inverse of the above sequence. Only the extended 386 IRET instruction (operand size = 32) can be used, and must be executed at level 0 to change the VM bit to 1.

- (1) If the NT bit in the FLAGS register is on, an inter-task return is performed. The current state is stored in the current TSS, and the link field in the current TSS is used to locate the TSS for the interrupted task which is to be resumed.

Otherwise, continue with the following sequence.

- (2) Read the FLAGS image from SS:8[ESP] into the FLAGS register. This will set VM to the value active in the interrupted routine.
- (3) Pop off the instruction pointer CS:EIP. EIP is popped first, then a 32-bit word is popped which contains the CS value in the lower 16 bits. If VM=0, this CS load is done as a protected mode segment load. If VM=1, this will be done as an 8086 segment load.
- (4) Increment the ESP register by 4 to bypass the FLAGS image which was "popped" in step 1.
- (5) If VM=1, load segment registers ES, DS, FS, and GS from memory locations SS:[ESP+8], SS:[ESP+12], SS:[ESP+16], and SS:[ESP+20], respectively, where the new value of ESP stored in step 4 is used. Since VM=1, these are done as 8086 segment register loads. Else if VM=0, check that the selectors in ES, DS, FS, and GS are valid in the interrupted routine. Null out invalid selectors to trap if an attempt is made to access through them.
- (6) If (RPL(CS) > CPL), pop the stack pointer SS:ESP from the stack. The ESP register is popped first, followed by 32-bits containing SS in the lower 16 bits. If VM=0, SS is loaded as a protected mode segment register load. If VM=1, an 8086 segment register load is used.
- (7) Resume execution of the interrupted routine. The VM bit in the FLAGS register (restored from the interrupt routine's stack image in step 1) determines whether the processor resumes the interrupted routine in Protected mode of Virtual 8086 mode.

5. FUNCTIONAL DATA

5.1 INTRODUCTION

The 80386 features a straightforward functional interface to the external hardware. The 80386 has separate, parallel buses for data and address. The data bus is 32-bits in width, and bidirectional. The address bus outputs 32-bit address values in the most directly usable form for the high-speed local bus: 4 individual byte enable signals, and the 30 upper-order bits as a binary value. The data and address buses are interpreted and controlled with their associated control signals.

A **dynamic data bus sizing** feature allows the processor to handle a mix of 32- and 16-bit external buses on a cycle-by-cycle basis (see **5.3.4 Data Bus Sizing**). If 16-bit bus size is selected, the 80386 automatically makes any adjustment needed, even performing another 16-bit bus cycle to complete the transfer if that is necessary. 8-bit peripheral devices may be connected to 32-bit or 16-bit buses with no loss of performance. A **new address pipelining option** is provided and applies to 32-bit and 16-bit buses for substantially improved memory utilization, especially for the most heavily used memory resources.

The **address pipelining option**, when selected, typically allows a given memory interface to operate with one less wait state than would otherwise be required (see **5.4.2 Address Pipelining**). The pipelined bus is also well suited to interleaved memory designs. For 16 MHz interleaved memory designs with 100 ns access time DRAMs, zero wait states can be achieved when pipelined addressing is selected. When address pipelining is requested by the external hardware, the 80386 will output the address and bus cycle definition of the next bus cycle (if it is internally available) even while waiting for the current cycle to be acknowledged.

Non-pipelined address timing, however, is ideal for external cache designs, since the cache memory will typically be fast enough to allow non-pipelined cycles. For maximum design flexibility, the address pipelining option is selectable on a cycle-by-cycle basis.

The processor's bus cycle is the basic mechanism for information transfer, either from system to processor, or from processor to system. 80386 bus cycles perform data transfer in a minimum of only two clock periods. On a 32-bit data bus, the maximum 80386 transfer bandwidth at 16 MHz is therefore 32 Mbytes/sec, and at 20 MHz bandwidth is 40 MBytes/sec. Any bus cycle will be extended for more than two clock periods, however, if external hardware withholds acknowledgement of the cycle.

At the appropriate time, acknowledgement is signalled by asserting the 80386 READY# input.

The 80386 can relinquish control of its local buses to allow mastership by other devices, such as direct memory access channels. When relinquished, HLDA is the only output pin driven by the 80386, providing near-complete isolation of the processor from its system. The near-complete isolation characteristic is ideal when driving the system from test equipment, and in fault-tolerant applications.

Functional data covered in this chapter describes the processor's hardware interface. First, the set of signals available at the processor pins is described (see **5.2 Signal Description**). Following that are the signal waveforms occurring during bus cycles (see **5.3 Bus Transfer Mechanism**, **5.4 Bus Functional Description** and **5.5 Other Functional Descriptions**).

5.2 SIGNAL DESCRIPTION

5.2.1 Introduction

Ahead is a brief description of the 80386 input and output signals arranged by functional groups. Note the # symbol at the end of a signal name indicates the active, or asserted, state occurs when the signal is at a low voltage. When no # is present after the signal name, the signal is asserted when at the high voltage level.

Example signal: M/IO# — High voltage indicates Memory selected
— Low voltage indicates I/O selected

The signal descriptions sometimes refer to AC timing parameters, such as "t₁₂₅ Reset Setup Time" and "t₂₅ Reset Hold Time." The values of these parameters can be found in Tables 7-4 and 7-5.

5.2.2 Clock (CLK2)

CLK2 provides the fundamental timing for the 80386. It is divided by two internally to generate the internal processor clock used for instruction execution. The internal clock is comprised of two phases, "phase one" and "phase two." Each CLK2 period is a phase of the internal clock. Figure 5-2 illustrates the relationship. If desired, the phase of the internal processor clock can be synchronized to a known phase by ensuring the RESET signal falling edge meets its applicable setup and hold times, t₁₂₅ and t₂₅.

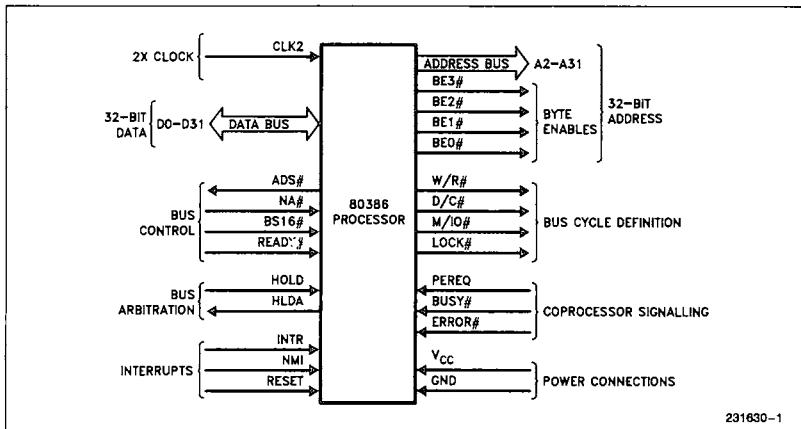


Figure 5-1. Functional Signal Groups

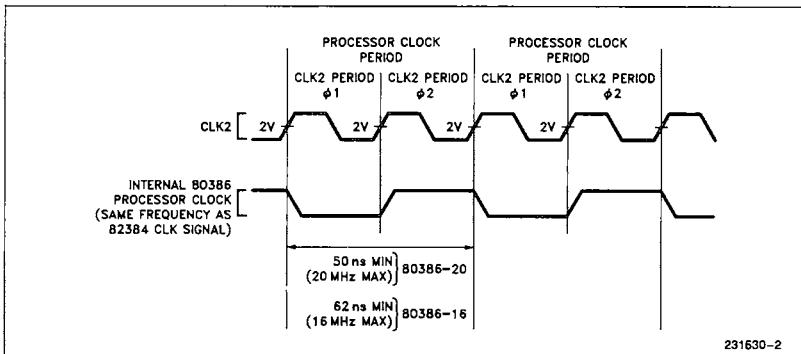


Figure 5-2. CLK2 Signal and Internal Processor Clock

5.2.3 Data Bus (D0 through D31)

These three-state bidirectional signals provide the general purpose data path between the 80386 and other devices. Data bus inputs and outputs indicate "1" when HIGH. The data bus can transfer data on 32- and 16-bit buses using a data bus sizing feature controlled by the BS16# input. See section 5.2.6 **Bus Control**. Data bus reads require that read data setup and hold times t₂₁ and t₂₂ be met for correct operation. During any write operation (and during halt cycles and shutdown cycles), the 80386 always drives all 32 signals of the data bus even if the current bus size is 16-bits.

5.2.4 Address Bus (BE0# through BE3#, A2 through A31)

These three-state outputs provide physical memory addresses or I/O port addresses. The address bus is capable of addressing 4 gigabytes of physical memory space (00000000H through FFFFFFFFH), and 64 kilobytes of I/O address space (00000000H through 0000FFFFH) for programmed I/O. I/O transfers automatically generated for 80386-to-coprocessor communication use I/O addresses 800000F8H through 800000FFH, so A31 HIGH in conjunction with M/I/O# LOW allows simple generation of the coprocessor select signal.



80386

ADVANCE INFORMATION

The Byte Enable outputs, BE0#–BE3#, directly indicate which bytes of the 32-bit data bus are involved with the current transfer. This is most convenient for external hardware.

BE0# applies to D0–D7
BE1# applies to D8–D15
BE2# applies to D16–D23
BE3# applies to D24–D31

The number of Byte Enables asserted indicates the physical size of the operand being transferred (1, 2, 3, or 4 bytes). Refer to section 5.3.6 Operand Alignment.

When a memory write cycle or I/O write cycle is in progress, and the operand being transferred occupies **only** the upper 16 bits of the data bus (D16–D31), duplicate data is simultaneously presented on the corresponding lower 16-bits of the data bus (D0–D15). This duplication is performed for optimum write performance on 16-bit buses. The pattern of write data duplication is a function of the Byte Enables asserted during the write cycle. Table 5-1 lists the write data present on D0–D31, as a function of the asserted Byte Enable outputs BE0#–BE3#.

5.2.5 Bus Cycle Definition Signals (W/R#, D/C#, M/IO#, LOCK#)

These three-state outputs define the type of bus cycle being performed. W/R# distinguishes between write and read cycles. D/C# distinguishes between data and control cycles. M/IO# distinguishes between memory and I/O cycles. LOCK# distinguishes between locked and unlocked bus cycles.

The primary bus cycle definition signals are W/R#, D/C#, and M/IO#, since these are the signals driven valid as the ADS# (Address Status output) is driven asserted. The LOCK# is driven valid at the same time as the first locked bus cycle begins, which due to address pipelining, could be later than ADS# is driven asserted. See 5.4.3.4 Pipelined Address. The LOCK# is negated when the READY# input terminates the last bus cycle which was locked.

Exact bus cycle definitions, as a function of W/R#, D/C#, and M/IO#, are given in Table 5-2. Note one combination of W/R#, D/C# and M/IO# is never given when ADS# is asserted (however, that combination, which is listed as "does not occur," will occur during Idle bus states when ADS# is not asserted). If M/IO#, D/C#, and W/R# are qualified by ADS# asserted, then a decoding scheme may use the non-occurring combination to its best advantage.

Table 5-1. Write Data Duplication as a Function of BE0#–BE3#

80386 Byte Enables				80386 Write Data				Automatic Duplication?
BE3 #	BE2 #	BE1 #	BE0 #	D24–D31	D16–D23	D8–D15	D0–D7	
High	High	High	Low	undef	undef	undef	A	No
High	High	Low	High	undef	undef	B	undef	No
High	Low	High	High	undef	C	undef	C	Yes
Low	High	High	High	D	undef	D	undef	Yes
High	High	Low	Low	undef	undef	B	A	No
High	Low	Low	High	undef	C	B	undef	No
Low	Low	High	High	D	C	D	C	Yes
High	Low	Low	Low	undef	C	B	A	No
Low	Low	Low	High	D	C	B	undef	No
Low	Low	Low	Low	D	C	B	A	No

Key:

D = logical write data d24–d31
C = logical write data d16–d23
B = logical write data d8–d15
A = logical write data d0–d7

Table 5-2. Bus Cycle Definition

M/IO #	D/C #	W/R #	Bus Cycle Type	Locked?	
Low	Low	Low	INTERRUPT ACKNOWLEDGE	Yes	
Low	Low	High	does not occur	—	
Low	High	Low	I/O DATA READ	No	
Low	High	High	I/O DATA WRITE	No	
High	Low	Low	MEMORY CODE READ	No	
High	Low	High	HALT: Address = 2 (BE0# High BE1# High BE2# Low BE3# High A2-A31 Low)	SHUTDOWN: Address = 0 (BE0# Low BE1# High BE2# High BE3# High A2-A31 Low)	No
High	High	Low	MEMORY DATA READ	Some Cycles	
High	High	High	MEMORY DATA WRITE	Some Cycles	

5.2.6 Bus Control Signals

5.2.6.1 INTRODUCTION

The following signals allow the processor to indicate when a bus cycle has begun, and allow other system hardware to control address pipelining, data bus width and bus cycle termination.

5.2.6.2 ADDRESS STATUS (ADS#)

This three-state output indicates that a valid bus cycle definition, and address (W/R#, D/C#, M/IO#, BE0#-BE3#, and A2-A31) is being driven at the 80386 pins. It is asserted during T1 and T2P bus states (see 5.4.3.2 Non-pipelined Address and 5.4.3.4 Pipelined Address for additional information on bus states).

5.2.6.3 TRANSFER ACKNOWLEDGE (READY#)

This input indicates the current bus cycle is complete, and the active bytes indicated by BE0#-BE3# and BS16# are accepted or provided. When READY# is sampled asserted during a read cycle or interrupt acknowledge cycle, the 80386 latches the input data and terminates the cycle. When READY# is sampled asserted during a write cycle, the processor terminates the bus cycle.

READY# is ignored on the first bus state of all bus cycles, and sampled each bus state thereafter until asserted. READY# must eventually be asserted to acknowledge every bus cycle, including Halt Indication and Shutdown Indication bus cycles. When be-

ing sampled, READY must always meet setup and hold times t_{19} and t_{20} for correct operation. See all sections of 5.4 Bus Functional Description.

5.2.6.4 NEXT ADDRESS REQUEST (NA#)

This is used to request address pipelining. This input indicates the system is prepared to accept new values of BE0#-BE3#, A2-A31, W/R#, D/C# and M/IO # from the 80386 even if the end of the current cycle is not being acknowledged on READY#. If this input is asserted when sampled, the next address is driven onto the bus, provided the next bus request is already pending internally. See 5.4.2 Address Pipelining and 5.4.3 Read and Write Cycles.

5.2.6.5 BUS SIZE 16 (BS16#)

The BS16# feature allows the 80386 to directly connect to 32-bit and 16-bit data buses. Asserting this input constrains the current bus cycle to use only the lower-order half (D0-D15) of the data bus, corresponding to BE0# and BE1#. Asserting BS16# has no additional effect if only BE0# and/or BE1# are asserted in the current cycle. However, during bus cycles asserting BE2# or BE3#, asserting BS16# will automatically cause the 80386 to make adjustments for correct transfer of the upper byte(s) using only physical data signals D0-D15.

If the operand spans both halves of the data bus and BS16# is asserted, the 80386 will automatically perform another 16-bit bus cycle. BS16# must always meet setup and hold times t_{17} and t_{18} for correct operation.



80386 I/O cycles automatically generated for coprocessor communication do not require BS16# be asserted. The coprocessor type, 80287 or 80387, is sensed on the ERROR# input shortly after the falling edge of RESET. The 80386 transfers only 16-bit quantities between itself and the 80287, but must transfer 32-bit quantities between itself and the 80387. Therefore, BS16# is a don't care during 80287 cycles and **must not** be asserted during 80387 communication cycles.

5.2.7 Bus Arbitration Signals

5.2.7.1 INTRODUCTION

This section describes the mechanism by which the processor relinquishes control of its local buses when requested by another bus master device. See **5.5.1 Entering and Exiting Hold Acknowledge** for additional information.

5.2.7.2 BUS HOLD REQUEST (HOLD)

This input indicates some device other than the 80386 requires bus mastership.

HOLD must remain asserted as long as any other device is a local bus master. HOLD is not recognized while RESET is asserted. If RESET is asserted while HOLD is asserted, RESET has priority and places the bus into an idle state, rather than the hold acknowledge (high impedance) state.

HOLD is level-sensitive and is a synchronous input. HOLD signals must always meet setup and hold times t_{23} and t_{24} for correct operation.

5.2.7.3 BUS HOLD ACKNOWLEDGE (HLDA)

Assertion of this output indicates the 80386 has relinquished control of its local bus in response to HOLD asserted, and is in the bus Hold Acknowledge state.

The Hold Acknowledge state offers near-complete signal isolation. In the Hold Acknowledge state, HLDA is the only signal being driven by the 80386. The other output signals or bidirectional signals (D0-D31, BE0#-BE3#, A2-A31, W/R#, D/C#, M/I/O#, LOCK# and ADS#) are in a high-impedance state so the requesting bus master may control them. Pullup resistors may be desired on several signals to avoid spurious activity when no bus master is driving them. See **7.2.3 Resistor Recommendations**. Also, one rising edge occurring on the NMI input during Hold Acknowledge is remembered, for processing after the HOLD input is negated.

In addition to the normal usage of Hold Acknowledge with DMA controllers or master peripherals, the near-complete isolation has particular attractiveness during system test when test equipment drives the system, and in hardware-fault-tolerant applications.

5.2.8 Coprocessor Interface Signals

5.2.8.1 INTRODUCTION

In the following sections are descriptions of signals dedicated to the numeric coprocessor interface. In addition to the data bus, address bus, and bus cycle definition signals, these following signals control communication between the 80386 and its 80287 or 80387 processor extension.

5.2.8.2 COPROCESSOR REQUEST (PERQ)

When asserted, this input signal indicates a coprocessor request for a data operand to be transferred to/from memory by the 80386. In response, the 80386 transfers information between the coprocessor and memory. Because the 80386 has internally stored the coprocessor opcode being executed, it performs the requested data transfer with the correct direction and memory address.

PEREQ is level-sensitive and is allowed to be asynchronous to the CLK2 signal.

5.2.8.3 COPROCESSOR BUSY (BUSY#)

When asserted, this input indicates the coprocessor is still executing an instruction, and is not yet able to accept another. When the 80386 encounters any coprocessor instruction which operates on the numeric stack (e.g. load, pop, or arithmetic operation), or the WAIT instruction, this input is first automatically sampled until it is seen to be negated. This sampling of the BUSY# input prevents overrunning the execution of a previous coprocessor instruction.

The FNINIT and FNCLEX coprocessor instructions are allowed to execute even if BUSY# is asserted, since these instructions are used for coprocessor initialization and exception-clearing.

BUSY# is level-sensitive and is allowed to be asynchronous to the CLK2 signal.

BUSY# serves an additional function. If BUSY# is sampled LOW at the falling edge of RESET, the 80386 performs an internal self-test (see **5.5.3 Bus Activity During and Following Reset**). If BUSY# is sampled HIGH, no self-test is performed.

5.2.8.4 COPROCESSOR ERROR (ERROR #)

This input signal indicates that the previous coprocessor instruction generated a coprocessor error of a type not masked by the coprocessor's control register. This input is automatically sampled by the 80386 when a coprocessor instruction is encountered, and if asserted, the 80386 generates exception 16 to access the error-handling software.

Several coprocessor instructions, generally those which clear the numeric error flags in the coprocessor or save coprocessor state, do execute without the 80386 generating exception 16 even if ERROR# is asserted. These instructions are FNINIT, FNCLEX, FSTSWX, FSTCW, FSTENV, FSAVE, FESTENV and FESAVE.

ERROR# is level-sensitive and is allowed to be asynchronous to the CLK2 signal.

ERROR# serves an additional function. If ERROR# is LOW no later than 20 CLK2 periods after the falling edge of RESET and remains LOW at least until the 80386 begins its first bus cycle, an 80387 is assumed to be present (ET bit in CR0 automatically gets set to 1). Otherwise, an 80287 (no coprocessor) is assumed to be present (ET bit in CR0 automatically is reset to 0). See 5.5.3 Bus Activity During and After Reset. Only the ET bit is set by this ERROR# pin test. Software must set the EM and MP bits in CR0 as needed. Therefore, distinguishing 80287 presence from no coprocessor requires a software test and appropriately resetting or setting the EM bit of CR0 (set EM = 1 when no coprocessor is present). If ERROR# is sampled LOW after reset (indicating 80387) but software later sets EM = 1, the 80386 will behave as if no coprocessor is present.

5.2.9 Interrupt Signals

5.2.9.1 INTRODUCTION

The following descriptions cover inputs that can interrupt or suspend execution of the processor's current instruction stream.

5.2.9.2 MASKABLE INTERRUPT REQUEST (INTR)

When asserted, this input indicates a request for interrupt service, which can be masked by the 80386 Flag Register IF bit. When the 80386 responds to the INTR input, it performs two interrupt acknowledge bus cycles, and at the end of the second, latches an 8-bit interrupt vector on D0-D7 to identify the source of the interrupt.

INTR is level-sensitive and is allowed to be asynchronous to the CLK2 signal. To assure recognition

of an INTR request, INTR should remain asserted until the first interrupt acknowledge bus cycle begins.

5.2.9.3 NON-MASKABLE INTERRUPT REQUEST (NMI)

This input indicates a request for interrupt service, which cannot be masked by software. The non-maskable interrupt request is always processed according to the pointer or gate in slot 2 of the interrupt table. Because of the fixed NMI slot assignment, no interrupt acknowledge cycles are performed when processing NMI.

NMI is rising edge-sensitive and is allowed to be asynchronous to the CLK2 signal. To assure recognition of NMI, it must be negated for at least eight CLK2 periods, and then be asserted for at least eight CLK2 periods.

Once NMI processing has begun, no additional NMIs are processed until after the next IRET instruction, which is typically the end of the NMI service routine. If NMI is re-asserted prior to that time, however, one rising edge on NMI will be remembered for processing after executing the next IRET instruction.

5.2.9.4 RESET (RESET)

This input signal suspends any operation in progress and places the 80386 in a known reset state. The 80386 is reset by asserting RESET for 15 or more CLK2 periods (80 or more CLK2 periods before requesting self test). When RESET is asserted, all other input pins are ignored, and all other bus pins are driven to an idle bus state as shown in Table 5-3. If RESET and HOLD are both asserted at a point in time, RESET takes priority even if the 80386 was in a Hold Acknowledge state prior to RESET asserted.

RESET is level-sensitive and must be synchronous to the CLK2 signal. If desired, the phase of the internal processor clock, and the entire 80386 state can be completely synchronized to external circuitry by ensuring the RESET signal falling edge meets its appropriate setup and hold times, t_{25} and t_{26} .

Table 5-3. Pin State (Bus Idle) During Reset

Pin Name	Signal Level During Reset
ADS#	High
D0-D31	High Impedance
BE0 #-BE3 #	Low
A2-A31	High
W/R#	Low
D/C#	High
M/I/O#	Low
LOCK#	High
HLDA	Low

5.2.10 Signal Summary

Table 5-4 summarizes the characteristics of all 80386 signals.

Table 5-4. 80386 Signal Summary

Signal Name	Signal Function	Active State	Input/Output	Input Synch or Asynch to CLK2	Output High Impedance During HLDA?
CLK2	Clock	—	I	—	—
D0-D31	Data Bus	High	I/O	S	Yes
BE0 # - BE3 #	Byte Enables	Low	O	—	Yes
A2-A31	Address Bus	High	O	—	Yes
W/R #	Write-Read Indication	High	O	—	Yes
D/C #	Data-Control Indication	High	O	—	Yes
M/I/O #	Memory-I/O Indication	High	O	—	Yes
LOCK #	Bus Lock Indication	Low	O	—	Yes
ADS #	Address Status	Low	O	—	Yes
NA #	Next Address Request	Low	I	S	—
BS16 #	Bus Size 16	Low	I	S	—
READY #	Transfer Acknowledge	Low	I	S	—
HOLD	Bus Hold Request	High	I	S	—
HLDA	Bus Hold Acknowledge	High	O	—	No
PEREQ	Coprocessor Request	High	I	A	—
BUSY #	Coprocessor Busy	Low	I	A	—
ERROR #	Coprocessor Error	Low	I	A	—
INTR	Maskable Interrupt Request	High	I	A	—
NMI	Non-Maskable Intrpt Request	High	I	A	—
RESET	Reset	High	I	S	—

5.3 BUS TRANSFER MECHANISM

5.3.1 Introduction

All data transfers occur as a result of one or more bus cycles. Logical data operands of byte, word and double-word lengths may be transferred without restrictions on physical address alignment. Any byte boundary may be used, although two or even three physical bus cycles are performed as required for unaligned operand transfers. See 5.3.4 Dynamic Data Bus Sizing and 5.3.6 Operand Alignment.

The 80386 address signals are designed to simplify external system hardware. Higher-order address bits are provided by A2-A31. Lower-order address in the form of BE0 # - BE3 # directly provides linear selects for the four bytes of the 32-bit data bus. Physical operand size information is thereby implicitly provided each bus cycle in the most usable form.

Byte Enable outputs BE0 # - BE3 # are asserted when their associated data bus bytes are involved with the present bus cycle, as listed in Table 5-5. During a bus cycle, any possible pattern of contiguous, asserted Byte Enable outputs can occur, but never patterns having a negated Byte Enable separating two or three asserted Enables.

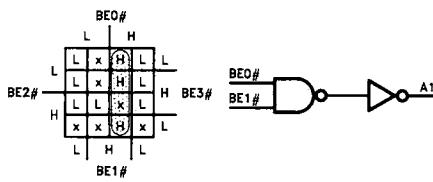
Address bits A0 and A1 of the physical operand's base address can be created when necessary (for instance, for MULTIBUS® I or MULTIBUS® II interface), as a function of the lowest-order asserted Byte Enable. This is shown by Table 5-6. Logic to generate A0 and A1 is given by Figure 5-3.

Table 5-5. Byte Enables and Associated Data and Operand Bytes

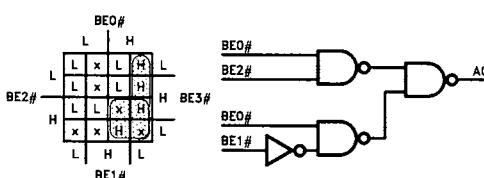
Byte Enable Signal	Associated Data Bus Signals
BE0#	D0-D7 (byte 0—least significant)
BE1#	D8-D15 (byte 1)
BE2#	D16-D23 (byte 2)
BE3#	D24-D31 (byte 3—most significant)

Table 5-6. Generating A0-A31 from BE0#-BE3# and A2-A31

80386 Address Signals			
A31	A2	BE3#	BE2#
Physical Base Address		BE1#	BE0#
A31	A2 A1 A0		
A31	0 0 X	X	Low
A31	0 1 X	X	High
A31	1 0 X	Low	High
A31	1 1 Low	High	High



231630-3



231630-4

Figure 5-3. Logic to Generate A0, A1 from BE0#-BE3#

Each bus cycle is composed of at least two bus states. Each bus state requires one processor clock period. Additional bus states added to a single bus cycle are called wait states. See 5.4 Bus Functional Description.

Since a bus cycle requires a minimum of two bus states (equal to two processor clock periods), data can be transferred between external devices and the 80386 at a maximum rate of one 4-byte Dword every two processor clock periods, for a maximum bus bandwidth of 40 megabytes/second (80386-20 operating at 20 MHz processor clock rate).

5.3.2 Memory and I/O Spaces

Bus cycles may access physical memory space or I/O space. Peripheral devices the system may either be memory-mapped, or I/O-mapped, or both. As shown in Figure 5-4, physical memory addresses range from 00000000H to FFFFFFFFH (4 gigabytes) and I/O addresses from 00000000H to 0000FFFFH (64 kilobytes) for programmed I/O. Note the I/O addresses used by the automatic I/O cycles for coprocessor communication are 800000F8H to 800000FFH, beyond the address range of programmed I/O, to allow easy generation of a coprocessor chip select signal using the A31 and M/IO# signals.

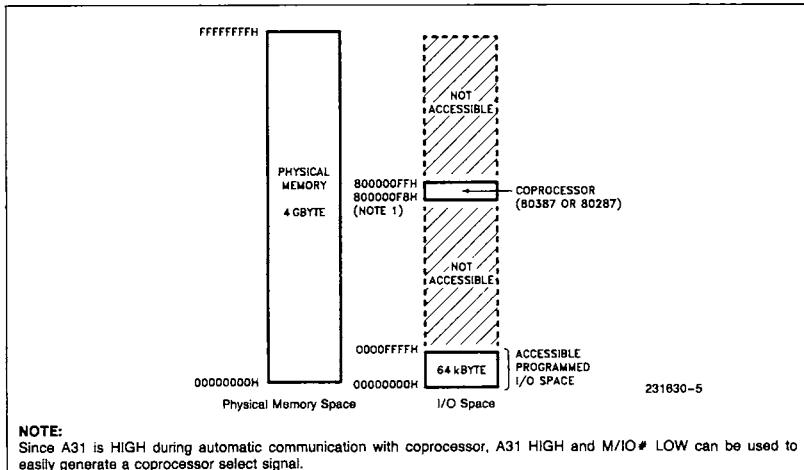


Figure 5-4. Physical Memory and I/O Spaces

5.3.3 Memory and I/O Organization

The 80386 datapath to memory and I/O spaces can be 32 bits wide or 16 bits wide. When 32-bits wide, memory and I/O spaces are organized naturally as arrays of physical 32-bit Dwords. Each memory or I/O Dword has four individually addressable bytes at consecutive byte addresses. The lowest-addressed byte is associated with data signals D0-D7; the highest-addressed byte with D24-D31.

The 80386 includes a bus control input, BS16#, that also allows direct connection to 16-bit memory or I/O spaces organized as a sequence of 16-bit words. Cycles to 32-bit and 16-bit memory or I/O devices may occur in any sequence, since the BS16# control is sampled during each bus cycle. See 5.3.4 Dynamic Data Bus Sizing. The Byte Enable signals, BE0#-BE3#, allow byte granularity when addressing any memory or I/O structure, whether 32 or 16 bits wide.

5.3.4 Dynamic Data Bus Sizing

Dynamic data bus sizing is a feature allowing direct processor connection to 32-bit or 16-bit data buses for memory or I/O. A single processor may connect to both size buses. Transfers to or from 32- or 16-bit ports are supported by dynamically determining the bus width during each bus cycle. During each bus cycle an address decoding circuit or the slave de-

vices itself may assert BS16# for 16-bit ports, or negate BS16# for 32-bit ports.

With BS16# asserted, the processor automatically converts operand transfers larger than 16 bits, or misaligned 16-bit transfers, into two or three transfers as required. All operand transfers physically occur on D0-D15 when BS16# is asserted. Therefore, 16-bit memories or I/O devices only connect on data signals D0-D15. No extra transceivers are required.

Asserting BS16# only affects the processor when BE2# and/or BE3# are asserted during the current cycle. If only D0-D15 are involved with the transfer, asserting BS16# has no effect since the transfer can proceed normally over a 16-bit bus whether BS16# is asserted or not. In other words, asserting BS16# has no effect when only the lower half of the bus is involved with the current cycle.

There are two types of situations where the processor is affected by asserting BS16#, depending on which Byte Enables are asserted during the current bus cycle:

Upper Half Only:
Only BE2# and/or BE3# asserted.

Upper and Lower Half:
At least BE1#, BE2# asserted (and perhaps also BE0# and/or BE3#).

Effect of asserting BS16# during "upper half only" read cycles:

Asserting BS16# during "upper half only" reads causes the 80386 to read data on the lower 16 bits of the data bus and ignore data on the upper 16 bits of the data bus. Data that would have been read from D16-D31 (as indicated by BE2# and BE3#) will instead be read from D0-D15 respectively.

Effect of asserting BS16# during "upper half only" write cycles:

Asserting BS16# during "upper half only" writes does not affect the 80386. When only BE2# and/or BE3# are asserted during a write cycle the 80386 always duplicates data signals D16-D31 onto D0-D15 (see Table 5-1). Therefore, no further 80386 action is required to perform these writes on 32-bit or 16-bit buses.

Effect of asserting BS16# during "upper and lower half" read cycles:

Asserting BS16# during "upper and lower half" reads causes the processor to perform two 16-bit read cycles for complete physical operand transfer. Bytes 0 and 1 (as indicated by BE0# and BE1#) are read on the first cycle using D0-D15. Bytes 2 and 3 (as indicated by BE2# and BE3#) are read during the second cycle, again using D0-D15. D16-D31 are ignored during both 16-bit cycles. BE0# and BE1# are always negated during the second 16-bit cycle (See Figure 5-14, cycles 2 and 2a).

Effect of asserting BS16# during "upper and lower half" write cycles:

Asserting BS16# during "upper and lower half" writes causes the 80386 to perform two 16-bit write cycles for complete physical operand transfer. All bytes are available the first write cycle allowing external hardware to receive Bytes 0 and 1 (as indicated by BE0# and BE1#) using D0-D15. On the second cycle the 80386 duplicates Bytes 2 and 3 on D0-D15 and Bytes 2 and 3 (as indicated by BE2# and BE3#) are written using D0-D15. BE0# and BE1# are always negated during the second 16-bit cycle. BS16# must be asserted during the second 16-bit cycle. See Figure 5-14, cycles 1 and 1a.

5.3.5 Interfacing with 32- and 16-Bit Memories

In 32-bit-wide physical memories such as Figure 5-5, each physical Dword begins at a byte address that is a multiple of 4. A2-A31 are directly used as a Dword select and BE0#-BE3# as byte selects. BS16# is negated for all bus cycles involving the 32-bit array.

When 16-bit-wide physical arrays are included in the system, as in Figure 5-6, each 16-bit physical word begins at a address that is a multiple of 2. Note the address is decoded, to assert BS16# only during bus cycles involving the 16-bit array. (If desiring to use

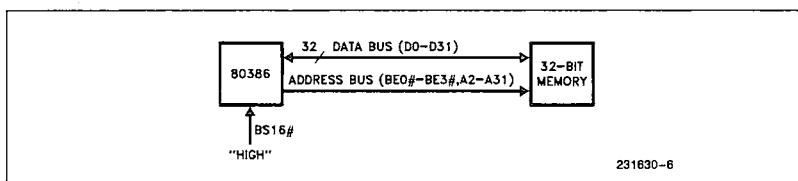


Figure 5-5. 80386 with 32-Bit Memory

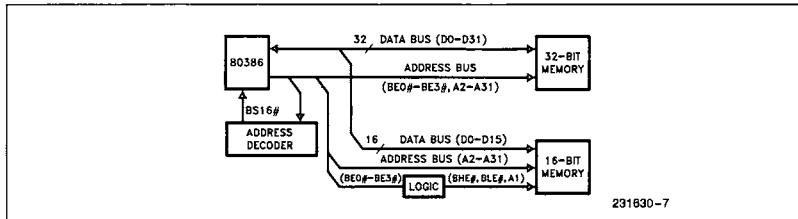


Figure 5-6. 80386 with 32-Bit and 16-Bit Memory

pipelined address with 16-bit memories then BE0# – BE3# and W/R# are also decoded to determine when BS16# should be asserted. See **5.4.3.6 Pipelined Address with Dynamic Data Bus Sizing.**)

A2–A31 are directly usable for addressing 32-bit and 16-bit devices. To address 16-bit devices, A1 and two byte enable signals are also needed.

To generate an A1 signal and two Byte Enable signals for 16-bit access, BE0# – BE3# should be decoded as in Table 5-7. Note certain combinations of BE0# – BE3# are never generated by the 80386, leading to "don't care" conditions in the decoder. Any BE0# – BE3# decoder, such as Figure 5-7, may use the non-occurring BE0# – BE3# combinations to its best advantage.

5.3.6 Operand Alignment

With the flexibility of memory addressing on the 80386, it is possible to transfer a logical operand that spans more than one physical Dword or word of memory or I/O. Examples are 32-bit Dword operands beginning at addresses not evenly divisible by

4, or a 16-bit word operand split between two physical Dwords of the memory array.

Operand alignment and data bus size dictate when multiple bus cycles are required. Table 5-8 describes the transfer cycles generated for all combinations of logical operand lengths, alignment, and data bus sizing. When multiple bus cycles are required to transfer a multi-byte logical operand, the highest-order bytes are transferred first (but if BS16# asserted requires two 16-bit cycles be performed, that part of the transfer is low-order first).

5.4 BUS FUNCTIONAL DESCRIPTION

5.4.1 Introduction

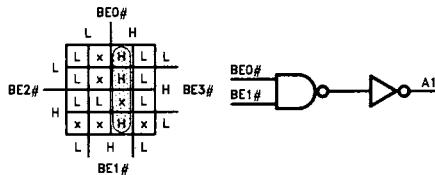
The 80386 has separate, parallel buses for data and address. The data bus is 32-bits in width, and bidirectional. The address bus provides a 32-bit value using 30 signals for the 30 upper-order address bits and 4 Byte Enable signals to directly indicate the active bytes. These buses are interpreted and controlled via several associated definition or control signals.

Table 5-7. Generating A1, BHE#, and BLE# for Addressing 16-Bit Devices

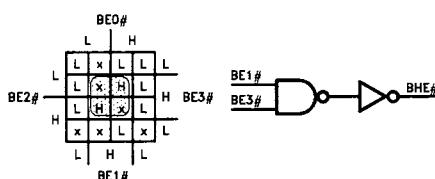
80386 Signals				16-Bit Bus Signals			Comments
BE3 #	BE2 #	BE1 #	BE0 #	A1	BHE #	BLE # (A0)	
H*	H*	H*	H*	X	X	X	x—no active bytes
H	H	H	L	L	H	L	
H	H	L	H	L	L	H	
H	H	L	L	L	L	L	
H	L	H	H	H	H	L	
H*	L*	H*	L*	X	X	X	
H	L	L	H	L	L	H	
H	L	L	L	L	L	L	
L	H	H	H	H	L	H	
L*	H*	H*	L*	X	X	X	x—not contiguous bytes
L*	H*	L*	H*	X	X	X	x—not contiguous bytes
L*	H*	L*	L*	X	X	X	x—not contiguous bytes
L	L	H	H	H	L	L	
L*	L*	H*	L*	X	X	X	x—not contiguous bytes
L	L	L	H	L	L	H	
L	L	L	L	L	L	L	

BLE# asserted when D0–D7 of 16-bit bus is active.
BHE# asserted when D8–D15 of 16-bit bus is active.
A1 low for all even words; A1 high for all odd words.

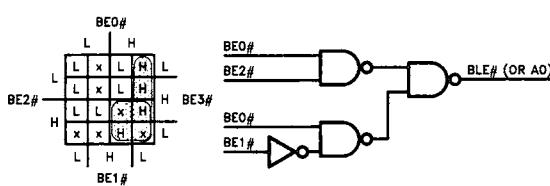
Key:
x = don't care
H = high voltage level
L = low voltage level
* = a non-occurring pattern of Byte Enables; either none are asserted, or the pattern has Byte Enables asserted for non-contiguous bytes



231630-8



231630-9



231630-10

Figure 5-7. Logic to Generate A1, BHE# and BLE# for 16-Bit Buses

Table 5-8. Transfer Bus Cycles for Bytes, Words and Dwords

	Byte-Length of Logical Operand							
	1	2				4		
Physical Byte Address in Memory (low-order bits)	xx	00	01	10	11	00	01	10
Transfer Cycles over 32-Bit Data Bus	b	w	w	w	hb,* lb	d	hb l3	hw, lw
Transfer Cycles over 16-Bit Data Bus	b	w	lb, hb	w	hb, lb	lw, hw	hb, lb, mw	hw, lw mw, hb, lb
Key:	b = byte transfer w = word transfer l = low-order portion m = mid-order portion x = don't care * = BS16# asserted causes second bus cycle							
	3 = 3-byte transfer d = Dword transfer h = high-order portion							

*For this case, 8086, 88, 186, 188, 286 transfer lb first, then hb.

The definition of each bus cycle is given by three definition signals: M/I/O#, W/R# and D/C#. At the same time, a valid address is present on the byte enable signals BE0#-BE3# and other address signals A2-A31. A status signal, ADS#, indicates when the 80386 issues a new bus cycle definition and address.

Collectively, the address bus, data bus and all associated control signals are referred to simply as "the bus".

When active, the bus performs one of the bus cycles below:

- 1) read from memory space
- 2) locked read from memory space
- 3) write to memory space
- 4) locked write to memory space
- 5) read from I/O space (or coprocessor)
- 6) write to I/O space (or coprocessor)
- 7) interrupt acknowledge
- 8) indicate halt, or indicate shutdown

Table 5-2 shows the encoding of the bus cycle definition signals for each bus cycle. See section 5.2.5 Bus Cycle Definition.

The data bus has a dynamic sizing feature supporting 32- and 16-bit bus size. Data bus size is indicated to the 80386 using its Bus Size 16 (BS16#) input. All bus functions can be performed with either data bus size.

When the 80386 bus is not performing one of the activities listed above, it is either Idle or in the Hold Acknowledge state, which may be detected by external circuitry. The idle state can be identified by the 80386, giving no further assertions on its address strobe output (ADS#) since the beginning of its most recent bus cycle, and the most recent bus cycle has been terminated. The hold acknowledge state is identified by the 80386 asserting its hold acknowledge (HLDA) output.

The shortest time unit of bus activity is a bus state. A bus state is one processor clock period (two CLK2 periods) in duration. A complete data transfer occurs during a bus cycle, composed of two or more bus states.

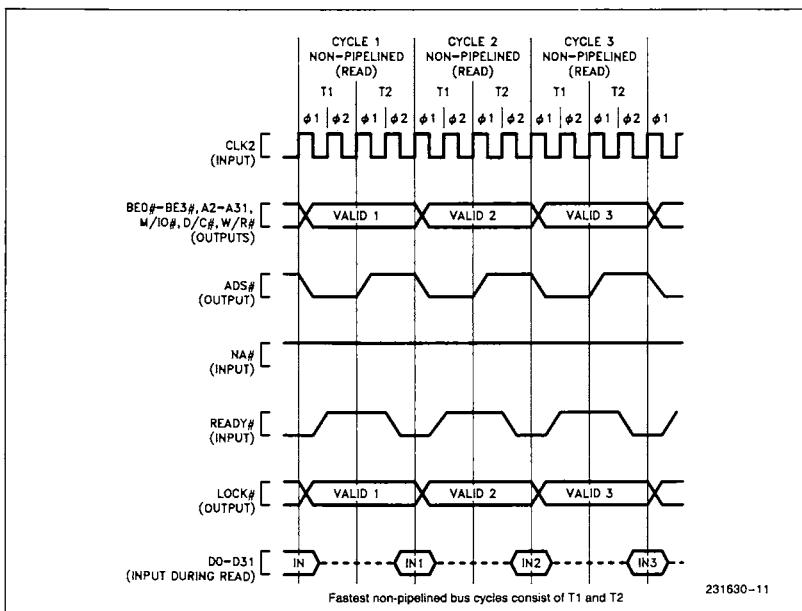


Figure 5-8. Fastest Read Cycles with Non-Pipelined Address Timing

The fastest 80386 bus cycle requires only two bus states. For example, three consecutive bus read cycles, each consisting of two bus states, are shown by Figure 5-8. The bus states in each cycle are named T1 and T2. Any memory or I/O address may be accessed by such a two-state bus cycle, if the external hardware is fast enough. The high-bandwidth, two-clock bus cycle realizes the full potential of fast main memory, or cache memory.

Every bus cycle continues until it is acknowledged by the external system hardware, using the 80386 READY# input. Acknowledging the bus cycle at the end of the first T2 results in the shortest bus cycle, requiring only T1 and T2. If READY# is not immediately asserted, however, T2 states are repeated indefinitely until the READY# input is sampled asserted.

5.4.2 Address Pipelining

The address pipelining option provides a choice of bus cycle timings. Pipelined or non-pipelined address timing is selectable on a cycle-by-cycle basis with the Next Address (NA#) input.

When address pipelining is not selected, the current address and bus cycle definition remain stable throughout the bus cycle.

When address pipelining is selected, the address (BE0#-BE3#, A2-A31) and definition (W/R#, D/C# and M/I/O#) of the next cycle are available before the end of the current cycle. To signal their availability, the 80386 address status output (ADS#) is also asserted. Figure 5-9 illustrates the fastest read cycles with pipelined address timing.

Note from Figure 5-9 the fastest bus cycles using pipelined address require only two bus states, named T1P and T2P. Therefore cycles with pipelined address timing allow the same data bandwidth as non-pipelined cycles, but address-to-data access time is increased compared to that of a non-pipelined cycle.

By increasing the address-to-data access time, pipelined address timing reduces wait state requirements. For example, if one wait state is required with non-pipelined address timing, no wait states would be required with pipelined address.

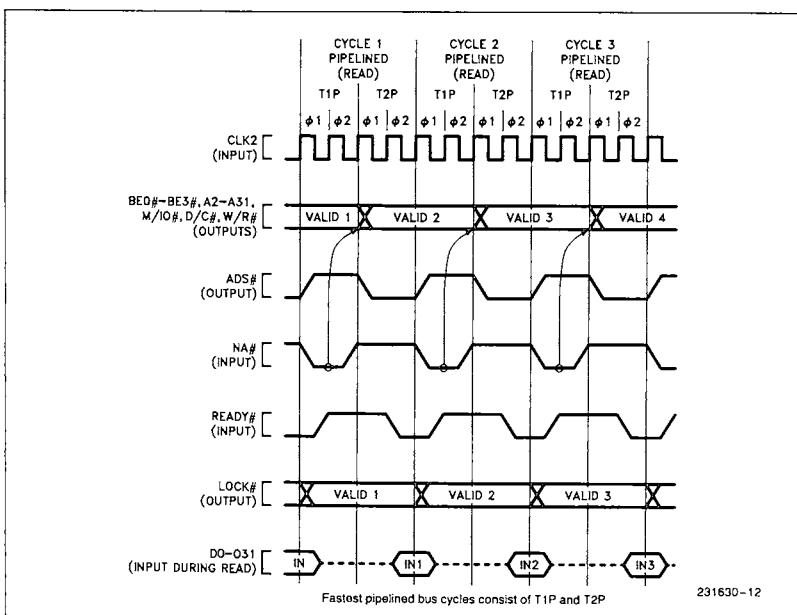


Figure 5-9. Fastest Read Cycles with Pipelined Address Timing

Pipelined address timing is useful in typical systems having address latches. In those systems, once an address has been latched, pipelined availability of the next address allows decoding circuitry to generate chip selects (and other necessary select signals) in advance, so selected devices are accessed immediately when the next cycle begins. In other words, the decode time for the next cycle can be overlapped with the end of the current cycle.

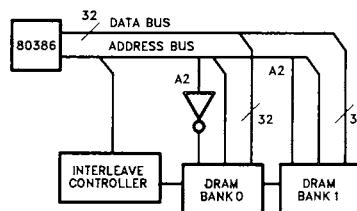
If a system contains a memory structure of two or more interleaved memory banks, pipelined address timing potentially allows even more overlap of activity. This is true when the interleaved memory controller is designed to allow the next memory operation

to begin in one memory bank while the current bus cycle is still activating another memory bank. Figure 5-10 shows the general structure of the 80386 with 2-bank and 4-bank interleaved memory. Note each memory bank of the interleaved memory has full data bus width (32-bit data width typically, unless 16-bit bus size is selected).

Further details of pipelined address timing are given in 5.4.3.4 Pipelined Address, 5.4.3.5 Initiating and Maintaining Pipelined Address, 5.4.3.6 Pipelined Address with Dynamic Bus Sizing, and 5.4.3.7 Maximum Pipelined Address Usage with 16-Bit Bus Size.

TWO-BANK INTERLEAVED MEMORY

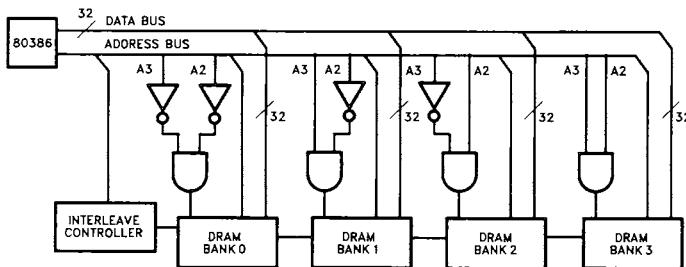
- a) Address signal A2 selects bank
- b) 32-bit datapath to each bank



231630-13

FOUR-BANK INTERLEAVED MEMORY

- a) Address signals A3 and A2 select bank
- b) 32-bit datapath to each bank



231630-14

Figure 5-10. 2-Bank and 4-Bank Interleaved Memory Structure

5.4.3 Read and Write Cycles

5.4.3.1 INTRODUCTION

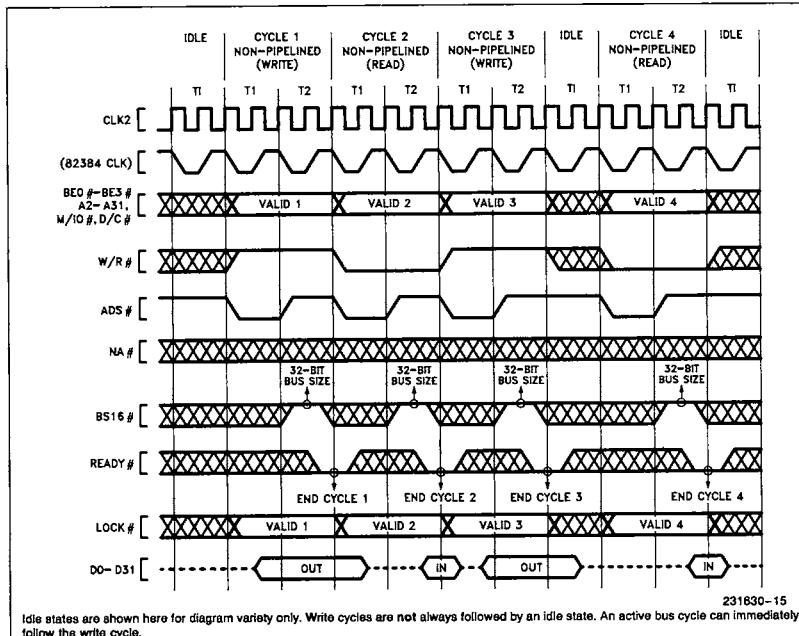
Data transfers occur as a result of bus cycles, classified as read or write cycles. During read cycles, data is transferred from an external device to the processor. During write cycles data is transferred in the other direction, from the processor to an external device.

Two choices of address timing are dynamically selectable: non-pipelined, or pipelined. After a bus idle state, the processor always uses non-pipelined address timing. However, the NA# (Next Address) input may be asserted to select pipelined address timing for the next bus cycle. When pipelining is selected and the 80386 has a bus request pending internally, the address and definition of the next cycle is made available even before the current bus cycle is acknowledged by READY#. Generally, the NA# input is sampled each bus cycle to select the desired address timing for the next bus cycle.

Two choices of physical data bus width are dynamically selectable: 32 bits, or 16 bits. Generally, the BS16# (Bus Size 16) input is sampled near the end of the bus cycle to confirm the physical data bus size applicable to the current cycle. Negation of BS16# indicates a 32-bit size, and assertion indicates a 16-bit bus size.

If 16-bit bus size is indicated, the 80386 automatically responds as required to complete the transfer on a 16-bit data bus. Depending on the size and alignment of the operand, another 16-bit bus cycle may be required. Table 5-7 provides all details. When necessary, the 80386 performs an additional 16-bit bus cycle, using D0-D15 in place of D16-D31.

Terminating a read cycle or write cycle, like any bus cycle, requires acknowledging the cycle by asserting the READY# input. Until acknowledged, the processor inserts wait states into the bus cycle, to allow adjustment for the speed of any external device. External hardware, which has decoded the address and bus cycle type asserts the READY# input at the appropriate time.



Idle states are shown here for diagram variety only. Write cycles are not always followed by an idle state. An active bus cycle can immediately follow the write cycle.

231630-15

Figure 5-11. Various Bus Cycles and Idle States with Non-Pipelined Address (zero wait states)

At the end of the second bus state within the bus cycle, READY# is sampled. At that time, if external hardware acknowledges the bus cycle by asserting READY#, the bus cycle terminates as shown in Figure 5-11. If READY# is negated as in Figure 5-12, the cycle continues another bus state (a wait state) and READY# is sampled again at the end of that state. This continues indefinitely until the cycle is acknowledged by READY# asserted.

When the current cycle is acknowledged, the 80386 terminates it. When a read cycle is acknowledged, the 80386 latches the information present at its data pins. When a write cycle is acknowledged, the 80386 write data remains valid throughout phase one of the next bus state, to provide write data hold time.

5.4.3.2 NON-PIPELINED ADDRESS

Any bus cycle may be performed with non-pipelined address timing. For example, Figure 5-11 shows a mixture of read and write cycles with non-pipelined address timing. Figure 5-11 shows the fastest possi-

ble cycles with non-pipelined address have two bus states per bus cycle. The states are named T1 and T2. In phase one of the T1, the address signals and bus cycle definition signals are driven valid, and to signal their availability, address status (ADS#) is simultaneously asserted.

During read or write cycles, the data bus behaves as follows. If the cycle is a read, the 80386 floats its data signals to allow driving by the external device being addressed. The 80386 requires that all data bus pins be at a valid logic state (high or low) at the end of each read cycle, when READY# is asserted. The system **MUST** be designed to meet this requirement. If the cycle is a write, data signals are driven by the 80386 beginning in phase two of T1 until phase one of the bus state following cycle acknowledgment.

Figure 5-12 illustrates non-pipelined bus cycles with one wait added to cycles 2 and 3. READY# is sampled negated at the end of the first T2 in cycles 2 and 3. Therefore cycles 2 and 3 have T2 repeated. At the end of the second T2, READY# is sampled asserted.

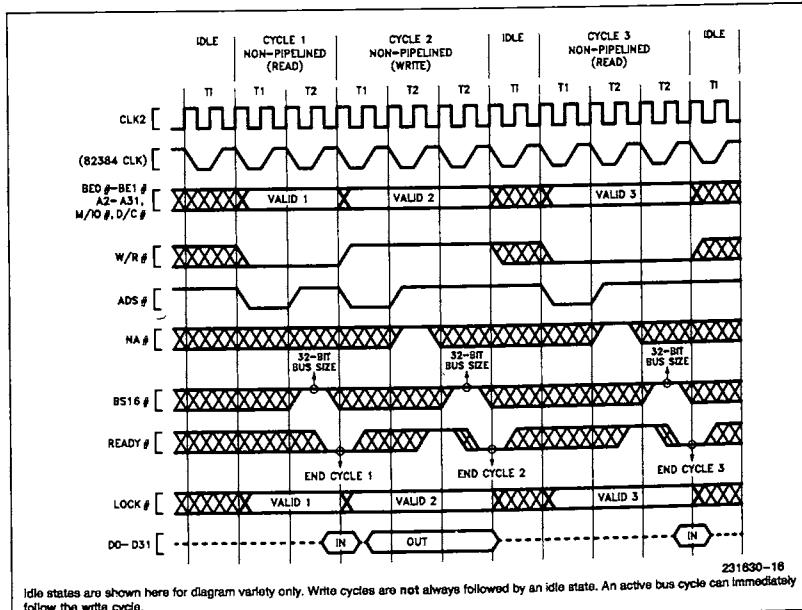


Figure 5-12. Various Bus Cycles and Idle States with Non-Pipelined Address
(various number of wait states)

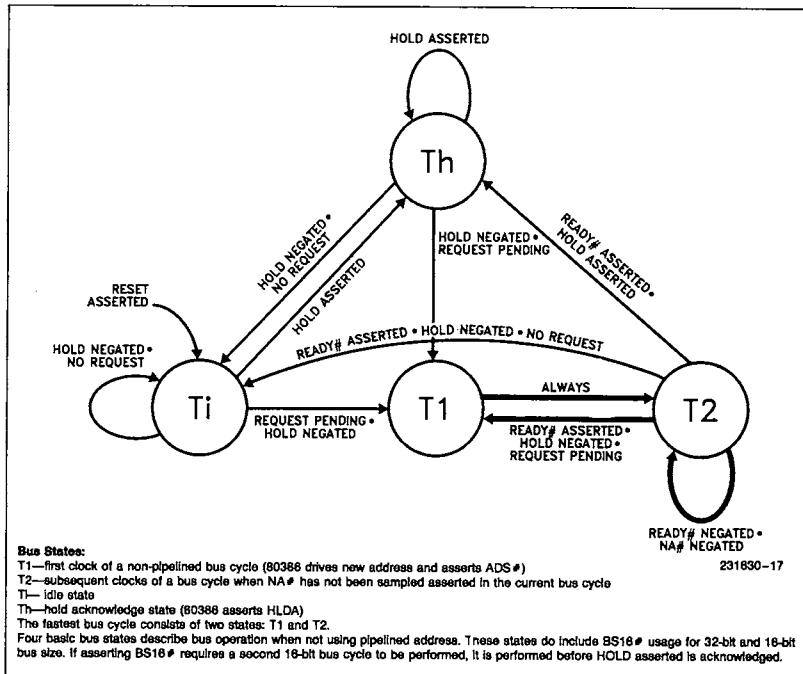


Figure 5-13. 80386 Bus States (not using pipelined address)

When address pipelining is not used, the address and bus cycle definition remain valid during all wait states. When wait states are added and you desire to maintain non-pipelined address timing, it is necessary to negate NA# during each T2 state except the last one, as shown in Figure 5-12 cycles 2 and 3. If NA# is sampled asserted during a T2 other than the last one, the next state would be T2I (for pipelined address) or T2P (for pipelined address) instead of another T2 (for non-pipelined address).

When address pipelining is not used, the bus states and transitions are completely illustrated by Figure 5-13. The bus transitions between four possible states: T1, T2, Ti, and Th. Bus cycles consist of T1 and T2, with T2 being repeated for wait states. Otherwise, the bus may be idle, in the Ti state, or in hold acknowledge, the Th state.

When address pipelining is not used, the bus state diagram is as shown in Figure 5-13. When the bus is

idle it is in state Ti. Bus cycles always begin with T1. T1 always leads to T2. If a bus cycle is not acknowledged during T2 and NA# is negated, T2 is repeated. When a cycle is acknowledged during T2, the following state will be T1 of the next bus cycle if a bus request is pending internally, or Ti if there is no bus request pending, or Th if the HOLD input is being asserted.

The bus state diagram in Figure 5-13 also applies to the use of BS16#. If the 80386 makes internal adjustments for 16-bit bus size, the adjustments do not affect the external bus states. If an additional 16-bit bus cycle is required to complete a transfer on a 16-bit bus, it also follows the state transitions shown in Figure 5-13.

Use of pipelined address allows the 80386 to enter three additional bus states not shown in Figure 5-13. Figure 5-20 in 5.4.3.4 Pipelined Address is the complete bus state diagram, including pipelined address cycles.

5.4.3.3 NON-PIPELINED ADDRESS WITH DYNAMIC DATA BUS SIZING

The physical data bus width for any non-pipelined bus cycle can be either 32-bits or 16-bits. At the beginning of the bus cycle, the processor behaves as if the data bus is 32-bits wide. When the bus cycle is acknowledged, by asserting READY# at the end of a T2 state, the most recent sampling of BS16# determines the data bus size for the cycle being acknowledged. If BS16# was most recently negated, the physical data bus size is defined as

32 bits. If BS16# was most recently asserted, the size is defined as 16 bits.

When BS16# is asserted and two 16-bit bus cycles are required to complete the transfer, BS16# must be asserted during the second cycle; 16-bit bus size is not assumed. Like any bus cycle, the second 16-bit cycle must be acknowledged by asserting READY#.

When a second 16-bit bus cycle is required to complete the transfer over a 16-bit bus, the addresses

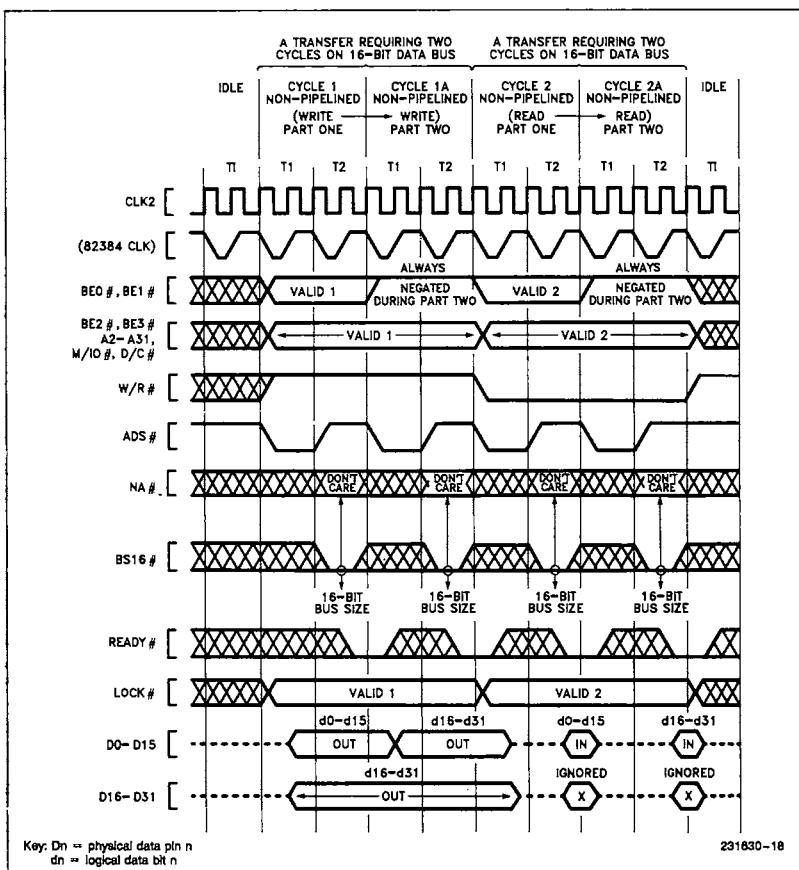


Figure 5-14. Asserting BS16# (zero wait states, non-pipelined address)

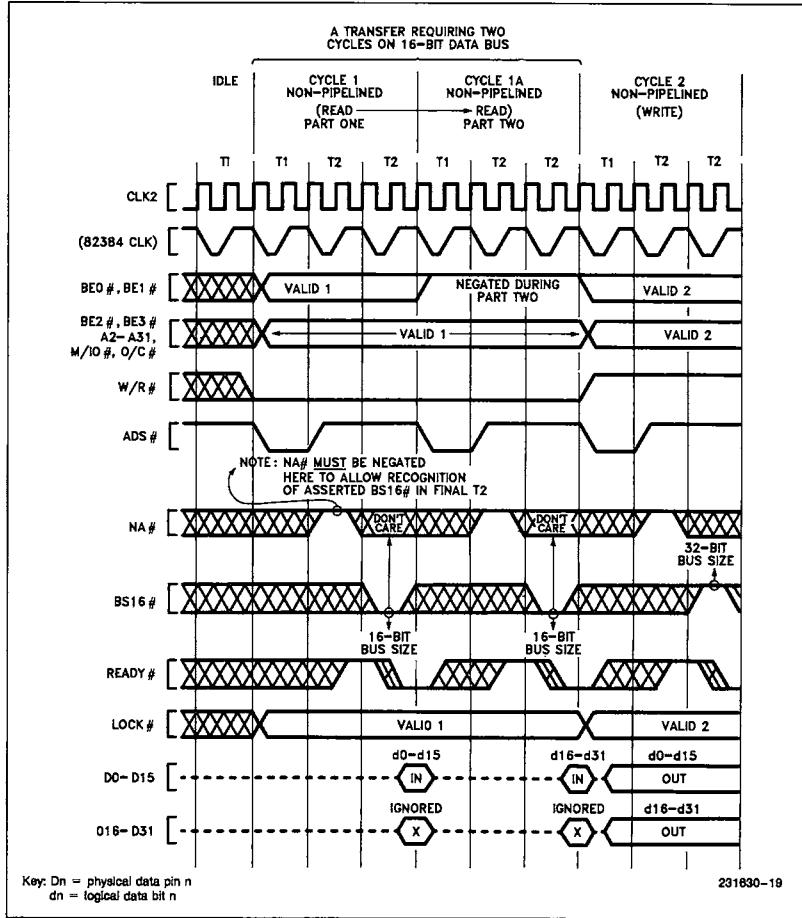


Figure 5-15. Asserting BS16# (one wait state, non-pipelined address)

generated for the two 16-bit bus cycles are closely related to each other. The addresses are the same except BE0# and BE1# are always negated for the second cycle. This is because data on D0-D15 was already transferred during the first 16-bit cycle.

Figures 5-14 and 5-15 show cases where assertion of BS16# requires a second 16-bit cycle for complete operand transfer. Figure 5-14 illustrates cycles

without wait states. Figure 5-15 illustrates cycles with one wait state. In Figure 5-15 cycle 1, the bus cycle during which BS16# is asserted, note that NA# must be negated in the T2 state(s) prior to the last T2 state. This is to allow the recognition of BS16# asserted in the final T2 state. The relation of NA# and BS16# is given fully in 5.4.3.4 Pipelined Address, but Figure 5-15 illustrates this only precaution you need to know when using BS16# with non-pipelined address.

5.4.3.4 PIPELINED ADDRESS

Address pipelining is the option of requesting the address and the bus cycle definition of the next, internally pending bus cycle before the current bus cycle is acknowledged with READY# asserted. ADS# is asserted by the 80386 when the next address is issued. The address pipelining option is controlled on a cycle-by-cycle basis with the NA# input signal.

Once a bus cycle is in progress and the current address has been valid for at least one entire bus state, the NA# input is sampled at the end of every phase one until the bus cycle is acknowledged. During non-pipelined bus cycles, therefore, NA# is sampled at the end of phase one of every T2. An example is Cycle 2 in Figure 5-16, during which NA# is sampled at the end of phase one of every T2 (it was asserted once during the first T2 and has no further effect during that bus cycle).

If NA# is sampled asserted, the 80386 is free to drive the address and bus cycle definition of the next bus cycle, and assert ADS#, as soon as it has a bus request internally pending. It may drive the next address as early as the next bus state, whether the current bus cycle is acknowledged at that time or not.

Regarding the details of address pipelining, the 80386 has the following characteristics:

- 1) For NA# to be sampled asserted, BS16# must be negated at that sampling window (see Figure 5-16 Cycles 3 and 4, and Figure 5-17 Cycles 2 through 4). If NA# and BS16# are both sampled asserted during the last T2 period of a bus cycle, BS16# asserted has priority. Therefore, if both are asserted, the current bus size is taken to be 16 bits and the next address is not pipelined. Conceptually, Figure 5-18 shows the internal 80386 logic providing these characteristics.

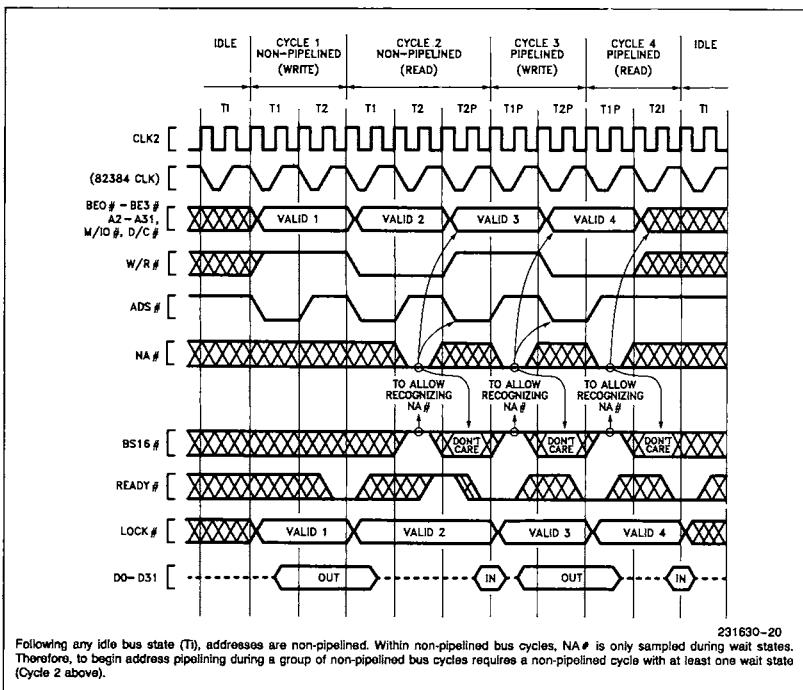


Figure 5-16. Transitioning to Pipelined Address During Burst of Bus Cycles

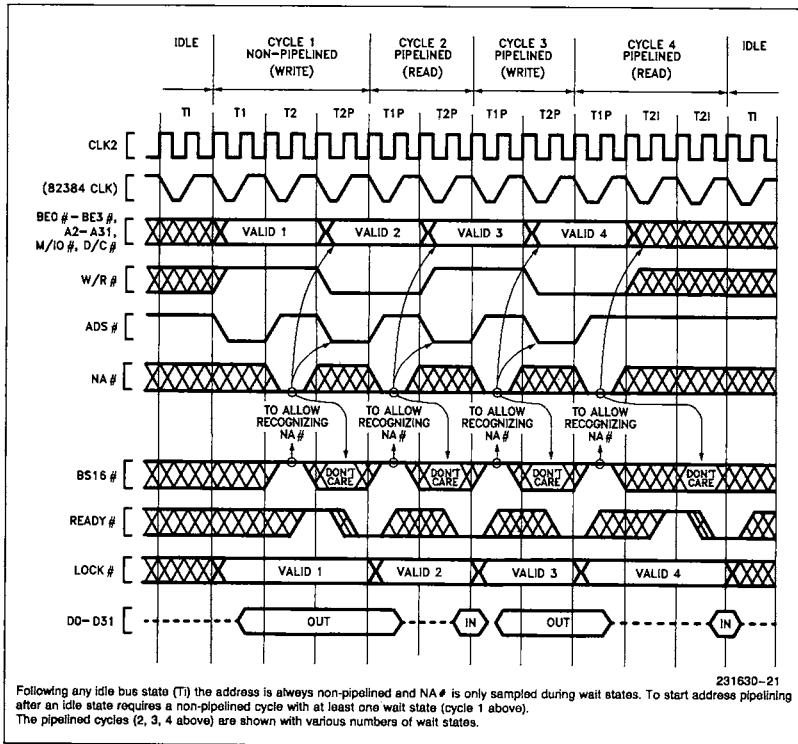


Figure 5-17. Fastest Transition to Pipelined Address Following Idle Bus State

- 2) The next address may appear as early as the bus state after NA# was sampled asserted (see Figures 5-16 or 5-17). In that case, state T2P is entered immediately. However, when there is not an internal bus request already pending, the next address will not be available immediately after NA# is asserted and T2I is entered instead of T2P (see Figure 5-19 Cycle 3). Provided the current bus cycle isn't yet acknowledged by READY# asserted, T2P will be entered as soon as the 80386 does drive the next address. External hardware should therefore observe the ADS# output as confirmation the next address is actually being driven on the bus.
 - 3) Once NA# is sampled asserted, the 80386 commits itself to the highest priority bus request that is pending internally. It can no longer perform another 16-bit transfer to the same address should
- BS16# be asserted externally, so thereafter must assume the current bus size is 32 bits. Therefore if NA# is sampled asserted within a bus cycle, BS16# is ignored thereafter in that bus cycle (see Figures 5-16, 5-17, 5-19). Consequently, do not assert NA# during bus cycles which must have BS16# driven asserted. See 5.4.3.6 Dynamic Bus-Sizing with Pipelined Address.
- 4) Any address which is validated by a pulse on the 80386 ADS# output will remain stable on the address pins for at least two processor clock periods. The 80386 cannot produce a new address more frequently than every two processor clock periods (see Figures 5-16, 5-17, 5-19).
 - 5) Only the address and bus cycle definition of the very next bus cycle is available. The pipelining capability cannot look further than one bus cycle ahead (see Figure 5-19 Cycle 1).

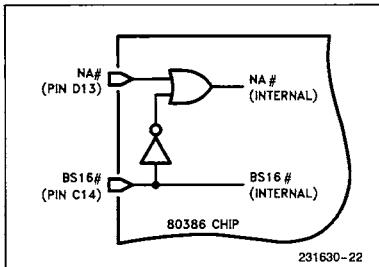


Figure 5-18. 80386 Internal Logic on NA# and BS16#

The complete bus state transition diagram, including operation with pipelined address is given by Figure 5-20. Note it is a superset of the diagram for non-pipelined address only, and the three additional bus states for pipelined address are drawn in bold.

The fastest bus cycle with pipelined address consists of just two bus states, T1P and T2P (recall for non-pipelined address it is T1 and T2). T1P is the first bus state of a pipelined cycle.

5.4.3.5 INITIATING AND MAINTAINING PIPELINED ADDRESS

Using the state diagram Figure 5-20, observe the transitions from an idle state, Ti, to the beginning of a pipelined bus cycle, T1P. From an idle state Ti, the first bus cycle must begin with T1, and there is therefore a non-pipelined bus cycle. The next bus cycle will be pipelined, however, provided NA# is asserted and the first bus cycle ends in a T2P state (the address for the next bus cycle is driven during T2P). The fastest path from an idle state to a bus cycle with pipelined address is shown in bold below:

Ti, Ti, Ti, **T1 - T2 - T2P,** **T1P - T2P,**
idle non-pipelined pipelined
states cycle cycle

T1-T2-T2P are the states of the bus cycle that establishes address pipelining for the next bus cycle, which begins with T1P. The same is true after a bus hold state, shown below:

Th, Th, Th, **T1 - T2 - T2P,** **T1P - T2P,**
hold non-pipelined pipelined
acknowledge cycle cycle
states

The transition to pipelined address is shown functionally by Figure 5-17 Cycle 1. Note that Cycle 1 is used to transition into pipelined address timing for the subsequent Cycles 2, 3 and 4, which are pipelined. The NA# input is asserted at the appropriate time to select address pipelining for Cycles 2, 3 and 4.

Once a bus cycle is in progress and the current address has been valid for one entire bus state, the NA# input is sampled at the end of every phase one until the bus cycle is acknowledged. During Figure 5-17 Cycle 1 therefore, sampling begins in T2. Once NA# is sampled asserted during the current cycle, the 80386 is free to drive a new address and bus cycle definition on the bus as early as the next bus state. In Figure 5-16 Cycle 1 for example, the next address is driven during state T2P. Thus Cycle 1 makes the transition to pipelined address timing, since it begins with T1 but ends with T2P. Because the address for Cycle 2 is available before Cycle 2 begins, Cycle 2 is called a pipelined bus cycle, and it begins with T1P. Cycle 2 begins as soon as READY# asserted terminates Cycle 1.

Example transition bus cycles are Figure 5-17 Cycle 1 and Figure 5-16 Cycle 2. Figure 5-17 shows transition during the very first cycle after an idle bus state, which is the fastest possible transition into address pipelining. Figure 5-16 Cycle 2 shows a transition cycle occurring during a burst of bus cycles. In any case, a transition cycle is the same whenever it occurs: it consists at least of T1, T2 (you assert NA# at that time), and T2P (provided the 80386 has an internal bus request already pending, which it almost always has). T2P states are repeated if wait states are added to the cycle.

Note three states (T1, T2 and T2P) are only required in a bus cycle performing a transition from non-pipelined address into pipelined address timing, for example Figure 5-17 Cycle 1. Figure 5-17 Cycles 2, 3 and 4 show that address pipelining can be maintained with two-state bus cycles consisting only of T1P and T2P.

Once a pipelined bus cycle is in progress, pipelined timing is maintained for the next cycle by asserting NA# and detecting that the 80386 enters T2P during the current bus cycle. The current bus cycle must end in state T2P for pipelining to be maintained in the next cycle. T2P is identified by the assertion of ADS#. Figures 5-16 and 5-17 however, each show pipelining ending after Cycle 4 because Cycle 4 ends in T2I. This indicates the 80386 didn't have an internal bus request prior to the acknowledgement of Cycle 4. If a cycle ends with a T2 or T2I, the next cycle will not be pipelined.

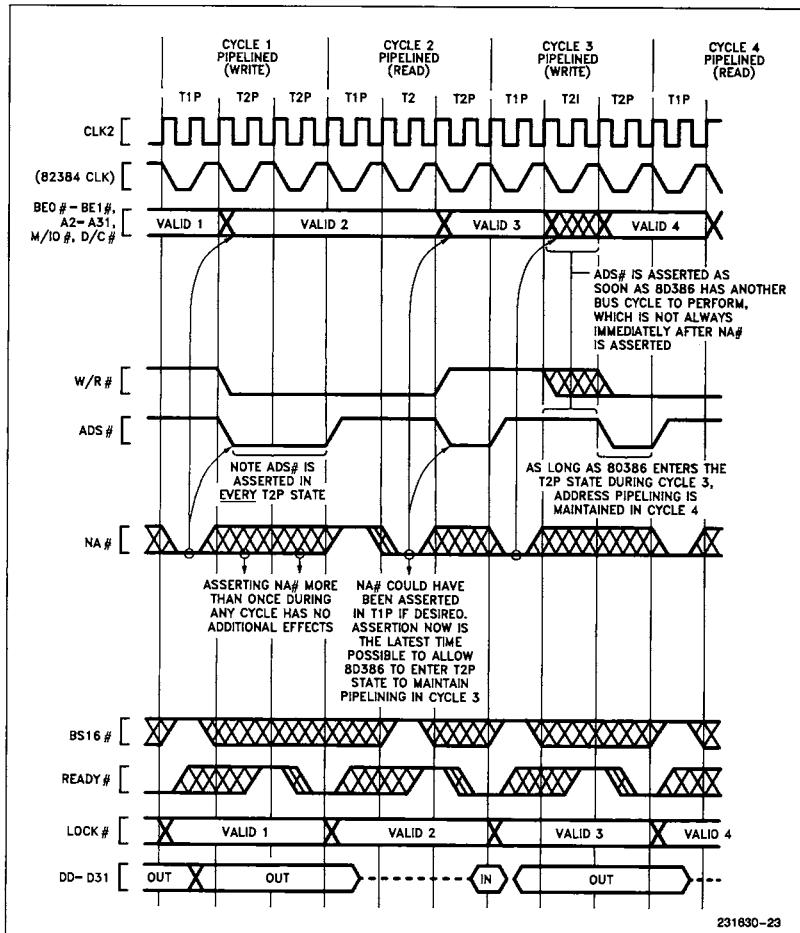


Figure 5-19. Details of Address Pipelining During Cycles with Wait States

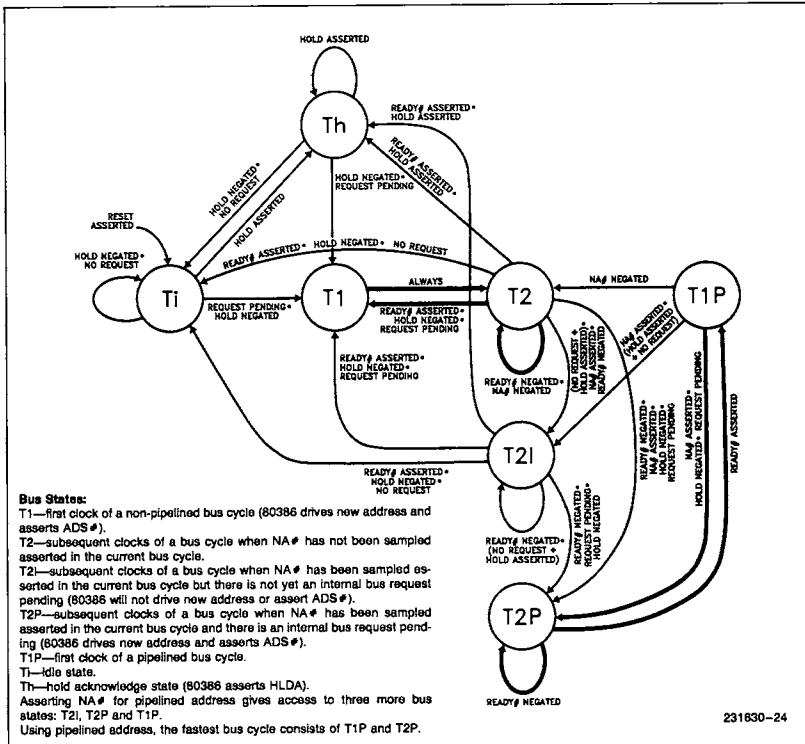


Figure 5-20. 80386 Complete Bus States (including pipelined address)

Realistically, address pipelining is almost always maintained as long as NA# is sampled asserted. This is so because in the absence of any other request, a code prefetch request is always internally pending until the instruction decoder and code prefetch queue are completely full. Therefore address pipelining is maintained for long bursts of bus cycles, if the bus is available (i.e., HOLD negated) and NA# is sampled asserted in each of the bus cycles.

5.4.3.6 PIPELINED ADDRESS WITH DYNAMIC DATA BUS SIZING

The BS16# feature allows easy interface to 16-bit data buses. When asserted, the 80386 bus interface

hardware performs appropriate action to make the transfer using a 16-bit data bus connected on D0-D15.

There is a degree of interaction, however, between the use of Address Pipelining and the use of Bus Size 16. The interaction results from the multiple bus cycles required when transferring 32-bit operands over a 16-bit bus. If the operand requires both 16-bit halves of the 32-bit bus, the appropriate 80386 action is a second bus cycle to complete the operand's transfer. It is this necessity that conflicts with NA# usage.

When NA# is sampled asserted, the 80386 commits itself to perform the next internally pending bus re-

quest, and is allowed to drive the next internally pending address onto the bus. Asserting NA# therefore makes it impossible for the next bus cycle to again access the current address on A2-A31, such as may be required when BS16# is asserted by the external hardware.

To avoid conflict, the 80386 is designed with following two provisions:

- 1) To avoid conflict, the 80386 is designed to ignore BS16# in the current bus cycle if NA# has already

been sampled asserted in the current cycle. If NA# is sampled asserted, the current data bus size is assumed to be 32 bits.

- 2) To also avoid conflict, if NA# and BS16# are both asserted during the same sampling window, BS16# asserted has priority and the 80386 acts as if NA# was negated at that time. Internal 80386 circuitry, shown conceptually in Figure 5-18, assures that BS16# is sampled asserted and NA# is sampled negated if both inputs are externally asserted at the same sampling window.

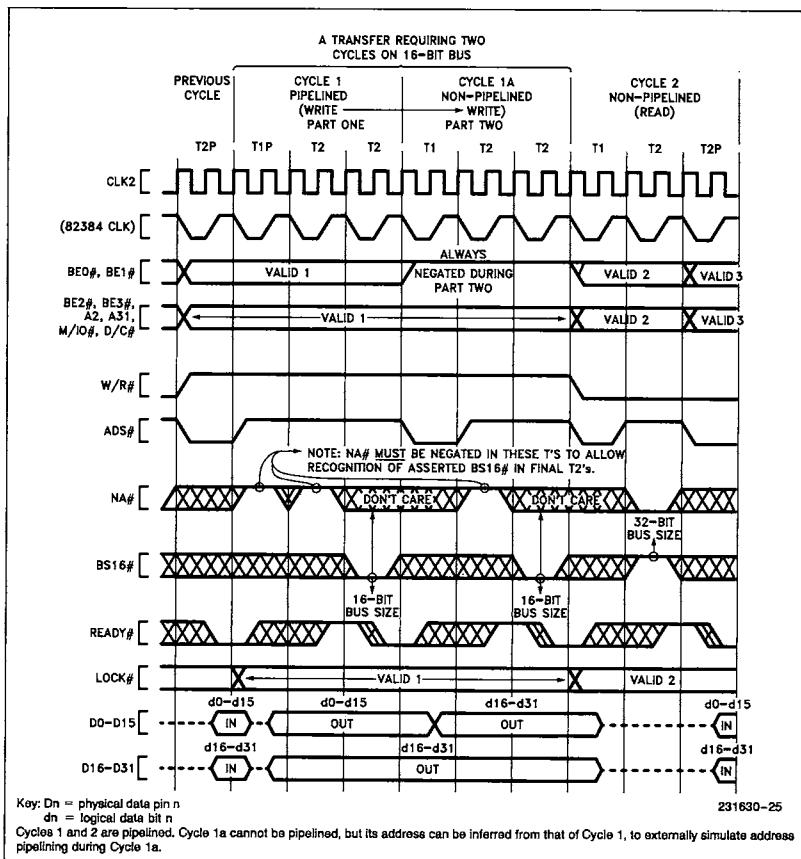


Figure 5-21. Using NA# and BS16#

Certain types of 16-bit or 8-bit operands require no adjustment for correct transfer on a 16-bit bus. Those are read or write operands using only the lower half of the data bus, and write operands using only the upper half of the bus since the 80386 simultaneously duplicates the write data on the lower half of the data bus. For these patterns of Byte Enables and the R/W# signals, BS16# need not be asserted at the 80386, allowing NA# to be asserted during the bus cycle if desired.

5.4.4 Interrupt Acknowledge (INTA) Cycles

In response to an interrupt request on the INTR input when interrupts are enabled, the 80386 performs

two interrupt acknowledge cycles. These bus cycles are similar to read cycles in that bus definition signals define the type of bus activity taking place, and each cycle continues until acknowledged by READY# sampled asserted.

The state of A2 distinguishes the first and second interrupt acknowledge cycles. The byte address driven during the first interrupt acknowledge cycle is 4 (A31-A3 low, A2 high, BE3#-BE1# high, and BE0# low). The address driven during the second interrupt acknowledge cycle is 0 (A31-A2 low, BE3#-BE1# high, BE0# low).

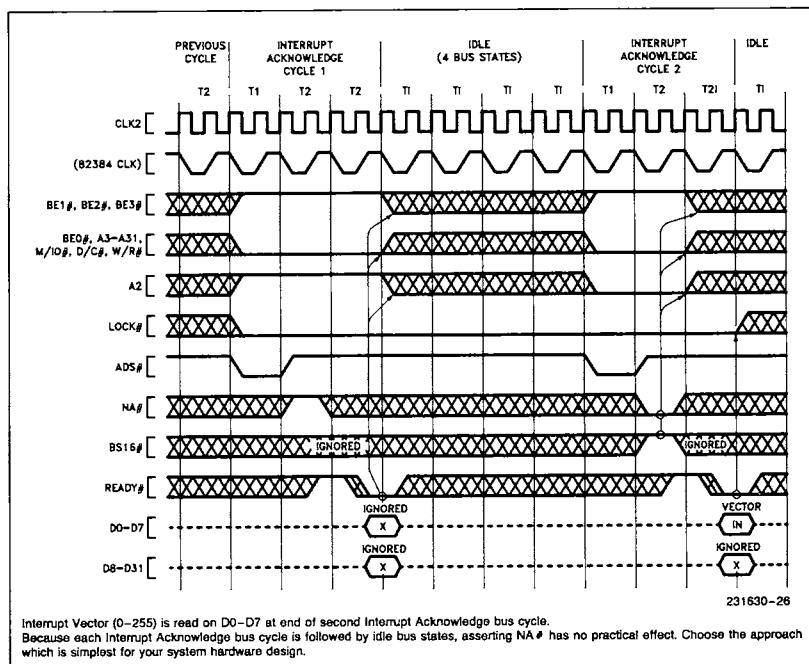


Figure 5-22. Interrupt Acknowledge Cycles

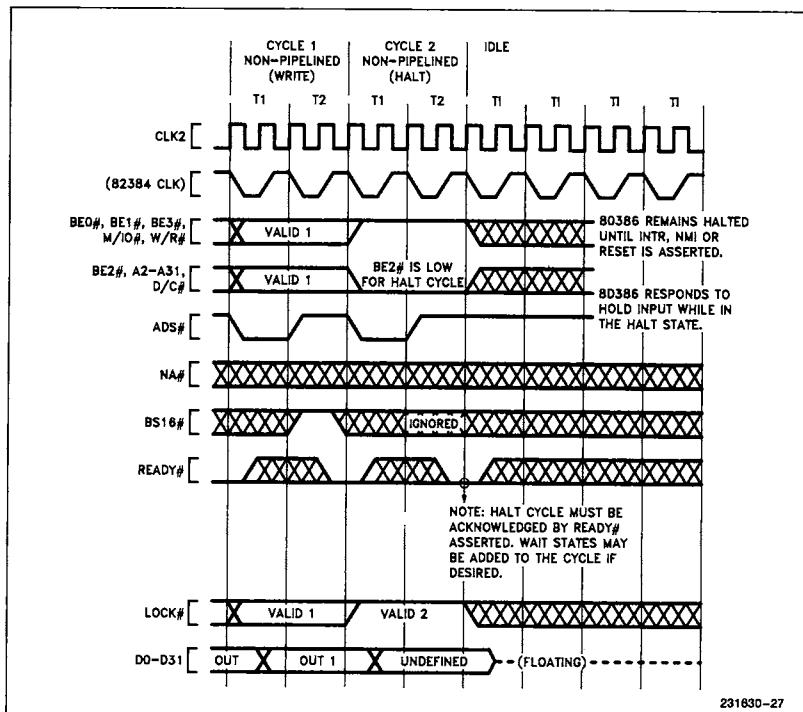


Figure 5-23. Halt Indication Cycle

The LOCK# output is asserted from the beginning of the first interrupt acknowledge cycle until the end of the second interrupt acknowledge cycle. Four idle bus states, TI, are inserted by the 80386 between the two interrupt acknowledge cycles, allowing at least 180 ns of locked idle time for future 80386 speed selections up to 24 MHz (CLK2 up to 48 MHz), for compatibility with spec TRHRL of the 8259A Interrupt Controller.

During both interrupt acknowledge cycles, D0-D31 float. No data is read at the end of the first interrupt acknowledge cycle. At the end of the second interrupt acknowledge cycle, the 80386 will read an external interrupt vector from D0-D7 of the data bus. The vector indicates the specific interrupt number (from 0-255) requiring service.

5.4.5 Halt Indication Cycle

The 80386 halts as a result of executing a HALT instruction. Signaling its entrance into the halt state, a halt indication cycle is performed. The halt indication cycle is identified by the state of the bus definition signals shown in 5.2.5 Bus Cycle Definition and a byte address of 2. BE0# and BE2# are the only signals distinguishing halt indication from shutdown indication, which drives an address of 0. During the halt cycle undefined data is driven on D0-D31. The halt indication cycle must be acknowledged by READY# asserted.

A halted 80386 resumes execution when INTR (if interrupts are enabled) or NMI or RESET is asserted.

5.4.6 Shutdown Indication Cycle

The 80386 shuts down as a result of a protection fault while attempting to process a double fault. Signaling its entrance into the shutdown state, a shutdown indication cycle is performed. The shutdown indication cycle is identified by the state of the bus definition signals shown in 5.2.5 Bus Cycle Definition and a byte address of 0. BE0# and BE2# are

the only signals distinguishing shutdown indication from halt indication, which drives an address of 2. During the shutdown cycle undefined data is driven on D0-D31. The shutdown indication cycle must be acknowledged by READY# asserted.

A shutdown 80386 resumes execution when NMI or RESET is asserted.

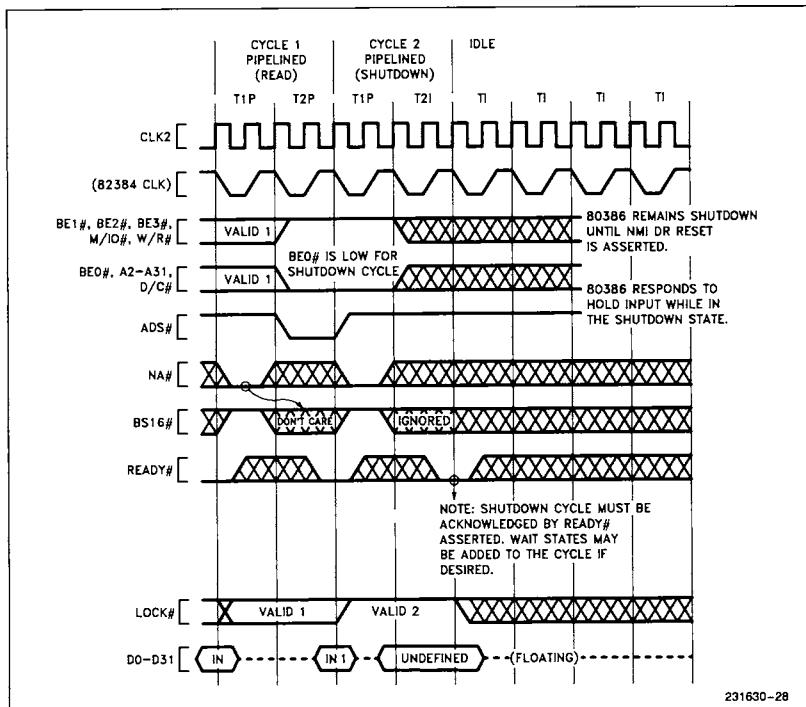


Figure 5-24. Shutdown Indication Cycle

231630-28

5.5 OTHER FUNCTIONAL DESCRIPTIONS

5.5.1 Entering and Exiting Hold Acknowledge

The bus hold acknowledge state, Th, is entered in response to the HOLD input being asserted. In the bus hold acknowledge state, the 80386 floats all output or bidirectional signals, except for HLDA. HLDA is asserted as long as the 80386 remains in the bus hold acknowledge state. In the bus hold acknowledge state, all inputs except HOLD and RESET are ignored (also up to one rising edge on NMI is remembered for processing when HOLD is no longer asserted).

is performed before HOLD is acknowledged, although the bus state diagrams in Figures 5-13 and 5-20 do not indicate that detail.

Th is exited in response to the HOLD input being negated. The following state will be T1 as in Figure 5-25 if no bus request is pending. The following bus state will be T1 if a bus request is internally pending, as in Figures 5-26 and 5-27.

Th is also exited in response to RESET being asserted.

If a rising edge occurs on the edge-triggered NMI input while in Th, the event is remembered as a non-maskable interrupt 2 and is serviced when Th is exited, unless of course, the 80386 is reset before Th is exited.

5.5.2 Reset During Hold Acknowledge

RESET being asserted takes priority over HOLD being asserted. Therefore, Th is exited in response to the RESET input being asserted. If RESET is asserted while HOLD remains asserted, the 80386 drives its pins to defined states during reset, as in Table 5-3 Pin State During Reset, and performs internal reset activity as usual.

If HOLD remains asserted when RESET is negated, the 80386 enters the hold acknowledge state before performing its first bus cycle, provided HOLD is still asserted when the 80386 would otherwise perform its first bus cycle. If HOLD remains asserted when RESET is negated, the BUSY# input is still sampled as usual to determine whether a self test is being requested, and ERROR# is still sampled as usual to determine whether an 80387 vs. an 80287 (or none) is present.

5.5.3 Bus Activity During and Following Reset

RESET is the highest priority input signal, capable of interrupting any processor activity when it is asserted. A bus cycle in progress can be aborted at any stage, or idle states or bus hold acknowledge states discontinued so that the reset state is established.

RESET should remain asserted for at least 15 CLK2 periods to ensure it is recognized throughout the 80386, and at least 78 CLK2 periods if 80386 self-test is going to be requested at the falling edge. RESET asserted pulses less than 15 CLK2 periods may not be recognized. RESET pulses less than 78 CLK2 periods followed by a self-test may cause the self-test to report a failure when no true failure exists. The additional RESET pulse width is required to clear additional state prior to a valid self-test.

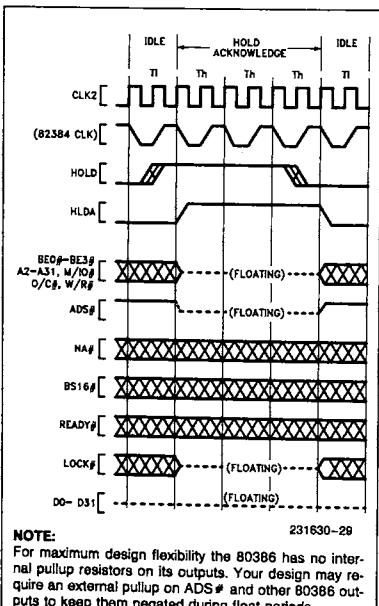


Figure 5-25. Requesting Hold from Idle Bus

Th may be entered from a bus idle state as in Figure 5-25 or after the acknowledgement of the current physical bus cycle if the LOCK# signal is not asserted, as in Figures 5-26 and 5-27. If HOLD is asserted during a locked bus cycle, the 80386 may execute one unlocked bus cycle before acknowledging HOLD. If asserting BS16# requires a second 16-bit bus cycle to complete a physical operand transfer, it

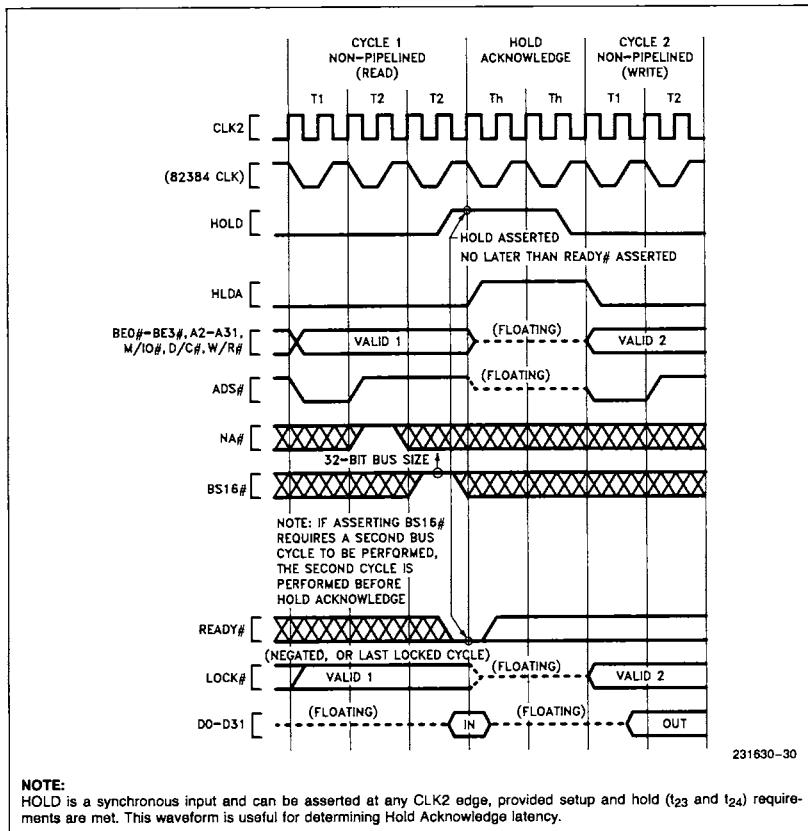
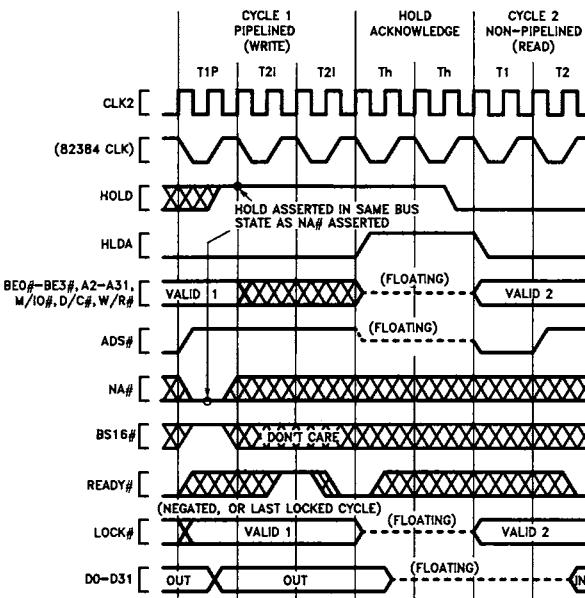


Figure 5-26. Requesting Hold from Active Bus (NA # negated)

Provided the RESET falling edge meets setup and hold times t_{25} and t_{26} , the internal processor clock phase is defined at that time, as illustrated by Figure 5-28 and Figure 7-7.

An 80386 self-test may be requested at the time RESET is negated by having the BUSY# input at a LOW level, as shown in Figure 5-28. The self-test requires $(220) + \text{approximately } 60$ CLK2 periods to complete. The self-test duration is not affected by the test results. Even if the self-test indicates a problem, the 80386 attempts to proceed with the reset sequence afterwards.

After the RESET falling edge (and after the self-test if it was requested) the 80386 performs an internal initialization sequence for approximately 350 to 450 CLK2 periods. Also during the initialization, between the 20th CLK2 period and the first bus cycle, the ERROR# input is sampled to determine the presence of an 80387 coprocessor versus the presence of an 80287 (or no coprocessor). During this time period, BUSY# must be HIGH. To distinguish between an 80287 being present and no coprocessor being present requires a software test.



231630-31

NOTE:

HOLD is a synchronous input and can be asserted at any CLK2 edge, provided setup and hold (t_{23} and t_{24}) requirements are met. This waveform is useful for determining Hold Acknowledge latency.

Figure 5-27. Requesting Hold from Active Bus (NA# asserted)

5.6 SELF-TEST SIGNATURE

Upon completion of self-test, (if self-test was requested by holding BUSY# LOW at least eight CLK2 periods before and after the falling edge of RESET), the EAX register will contain a signature of 00000000h indicating the 80386 passed its self-test of microcode and major PLA contents with no problems detected. The passing signature in EAX, 00000000h, applies to all 80386 revision levels. Any non-zero signature indicates the 80386 unit is faulty.

5.7 COMPONENT AND REVISION IDENTIFIERS

To assist 80386 users, the 80386 after reset holds a component identifier and a revision identifier in its DX

register. The upper 8 bits of DX hold 03h as identification of the 80386 component. The lower 8 bits of DX hold an 8-bit unsigned binary number related to the component revision level. The revision identifier begins chronologically with a value zero and is subject to change (typically it will be incremented) with component steppings intended to have certain improvements or distinctions from previous steppings.

These features are intended to assist 80386 users to a practical extent. However, the revision identifier value is not guaranteed to change with every stepping revision, or to follow a completely uniform numerical sequence, depending on the type or intention of revision, or manufacturing materials required to be changed. Intel has sole discretion over these characteristics of the component.

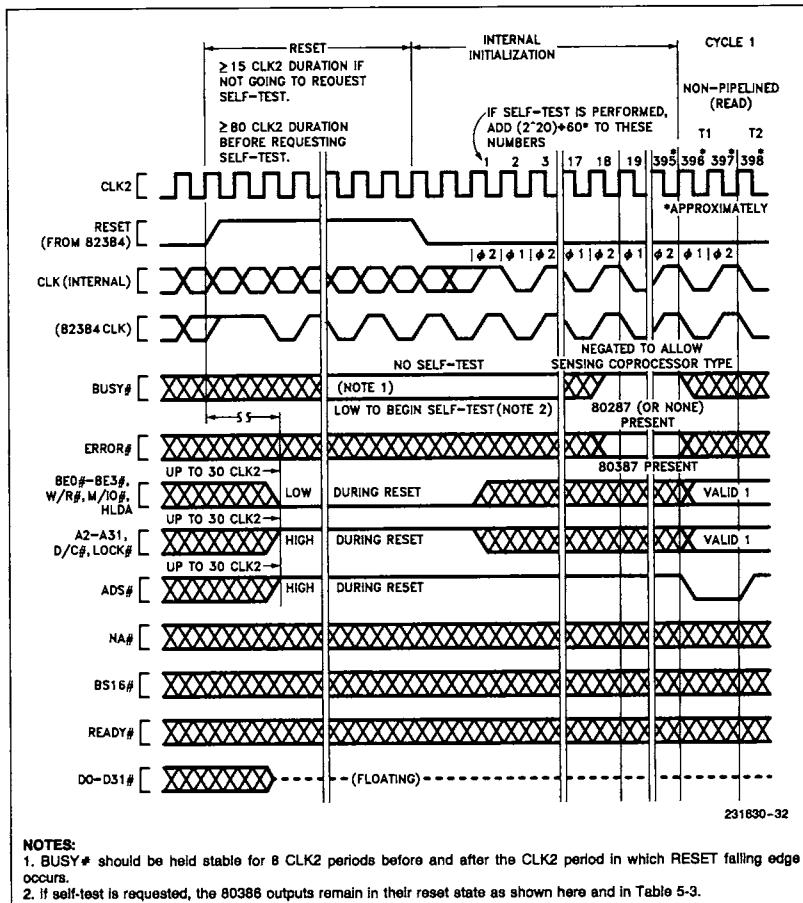


Figure 5-28. Bus Activity from Reset Until First Code Fetch

Table 5-10. Component and Revision Identifier History

80386 Stepping Name	Component Identifier	Revision Identifier	80386 Stepping Name	Component Identifier	Revision Identifier
B0	03	03			
B1	03	03			



5.8 COPROCESSOR INTERFACING

The 80386 provides an automatic interface for the Intel 80287 or 80387 numeric floating-point coprocessors. The 80287 and 80387 coprocessors use an I/O-mapped interface driven automatically by the 80386 and assisted by three dedicated signals: BUSY#, ERROR#, and PEREQ.

As the 80386 begins supporting a coprocessor instruction, it tests the BUSY# and ERROR# signals to determine if the coprocessor can accept its next instruction. Thus, the BUSY# and ERROR# inputs eliminate the need for any "preamble" bus cycles for communication between processor and coprocessor. The 80287 and 80387 can be given its command opcode immediately. The dedicated signals provide instruction synchronization, and eliminate the need of using the 80386 WAIT opcode (9Bh) for 80287/80387 instruction synchronization (the WAIT opcode was required when 8086 or 8088 was used with the 8087 coprocessor).

Custom coprocessors can be included in 80386-based systems, via memory-mapped or I/O-mapped interfaces. Such coprocessor interfaces allow a completely custom protocol, and are not limited to a set of coprocessor protocol "primitives". Instead, memory-mapped or I/O-mapped interfaces may use all applicable 80386 instructions for high-speed coprocessor communication. The BUSY# and ERROR# inputs of the 80386 may also be used for the custom coprocessor interface, if such hardware assist is desired. These signals can be tested by the 80386 WAIT opcode (9Bh). The WAIT instruction will wait until the BUSY# input is negated (interruptable by an NMI or enabled INTR input), but generates an exception 16 fault when the ERROR# pin is in the asserted state when the BUSY# goes (or is) negated. If the custom coprocessor interface is memory-mapped, protection of the addresses used for the interface can be provided with the 80386 on-chip paging or segmentation mechanisms. If the custom interface is I/O-mapped, protection of the interface can be provided with the 80386 IOPL (I/O Privilege Level) mechanism.

The 80287 and 80387 numeric coprocessor interfaces are I/O mapped as shown in Table 5-11. Note that the 80287/80387 coprocessor interface addresses are beyond the 0h-FFFFh range for programmed I/O. When the 80386 supports the 80287 or 80387 coprocessors, the 80386 automatically generates bus cycles to the coprocessor interface addresses.

Table 5-11. Numeric Coprocessor Port Addresses

Address in 80386 I/O Space	80287 Coprocessor Register	80387 Coprocessor Register
800000F8h	Opcode Register (16-bit port)	Opcode Register (32-bit port)
800000FCh	Operand Register (16-bit port)	Operand Register (32-bit port)

The 80287 coprocessor (16-bit) functions with either 80286 or 80386 processor. The 80387 coprocessor (32-bit) functions with the 80386 processor. To correctly map the 80287 and 80387 registers to the appropriate I/O addresses, connect the CMD0 and CMD1 lines of the 80287/80387 as listed in Table 5-12.

Table 5-12. Connections for CMD0 and CMD1
Inputs of 80287/80387

Coprocessor and Processor Configuration	Coprocessor CMD0 Connection	Coprocessor CMD1 Connection
80387 connected to 80386	connect to latched version of '386 A2 signal	None—80387 has no CMD1 pin
80287 connected to 80386	connect to latched version of '386 A2 signal	connect to ground
80287 connected to 80286	connect to latched version of '286 A1 signal	connect to latched version of '286 A2 signal

5.8.1 Software Testing for Coprocessor Presence

When software is used to test for coprocessor (80387 or 80287) presence, it should use only the following coprocessor opcodes: FINIT, FNINIT, FSTCW mem, FSTSW mem, FSTSW AX. To use other coprocessor opcodes when a coprocessor is known to be not present, first set EM = 1 in 80386 CR0.

6. MECHANICAL DATA

6.1 INTRODUCTION

In this section, the physical packaging and its connections are described in detail.

6.2 PIN ASSIGNMENT

The 80386 pinout as viewed from the top side of the component is shown by Figure 6-1. Its pinout as viewed from the Pin side of the component is Figure 6-2.

V_{CC} and GND connections must be made to multiple V_{CC} and V_{SS} (GND) pins. Each V_{CC} and V_{SS} must be connected to the appropriate voltage level. The circuit board should include V_{CC} and GND planes for power distribution and all V_{CC} and V_{SS} pins must be connected to the appropriate plane.

NOTE:

Pins identified as "N.C." should remain completely unconnected.

	P	N	M	L	K	J	H	G	F	E	D	C	B	A			
1															1		
2	A30	A27	A26	A23	A21	A20	A17	A16	A15	A14	A11	AB	VSS	VCC	2		
3	VCC	A31	A29	A24	A22	VSS	A18	VCC	VSS	A13	A10	A7	A5	VSS	3		
4															4		
5	D30	VSS	VCC	A28	A25	VSS	A19	VCC	VSS	A12	A9	A6	A4	A3	5		
6	D29	VCC	VSS									A2	NC	NC	6		
7				D27	D31							VCC	VSS	VCC	7		
8				VSS	D25	D28						NC	NC	VSS	8		
9				D24	VCC	VCC						NC	INTR	VCC	9		
10				VCC	D23	VSS						PEREQ	NMI	ERROR#	10		
11				D22	D21	D20						RESET	BUSY#	VSS	11		
12				D19	D17	VSS						LOCK#	W/R#	VCC	12		
13				D18	D16	D15						VSS	VSS	D/C#	13		
14				D14	D12	D10	VCC	D7	VSS	DO	VCC	CLK2	BE0#	VCC	14		
				D13	D11	VCC	D8	D5	VSS	D1	READY#	NC	NC	HA#	BE1#	BE2#	BE3#
				VSS	D9	HLDA	D6	D4	D3	D2	VCC	VSS	ADS#	HOLD	BS16#	VSS	VCC

Figure 6-1. 80386 PGA Pinout—View from Top Side

231630-33

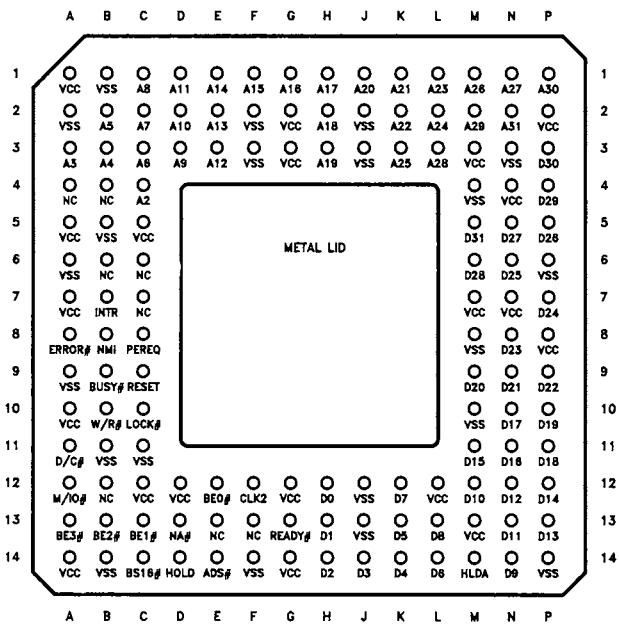


Figure 6-2. 80386 PGA Pinout—View from Pin Side

231630-34



80386

ADVANCE INFORMATION

Table 6-1. 80386 PGA Pinout—Functional Grouping

Pin / Signal	Pin / Signal	Pin / Signal	Pin / Signal
N2 A31	M5 D31	A1 V _{CC}	A2 V _{SS}
P1 A30	P3 D30	A5 V _{CC}	A6 V _{SS}
M2 A29	P4 D29	A7 V _{CC}	A9 V _{SS}
L3 A28	M6 D28	A10 V _{CC}	B1 V _{SS}
N1 A27	N5 D27	A14 V _{CC}	B5 V _{SS}
M1 A26	P5 D26	C5 V _{CC}	B11 V _{SS}
K3 A25	N6 D25	C12 V _{CC}	B14 V _{SS}
L2 A24	P7 D24	D12 V _{CC}	C11 V _{SS}
L1 A23	N8 D23	G2 V _{CC}	F2 V _{SS}
K2 A22	P9 D22	G3 V _{CC}	F3 V _{SS}
K1 A21	N9 D21	G12 V _{CC}	F14 V _{SS}
J1 A20	M9 D20	G14 V _{CC}	J2 V _{SS}
H3 A19	P10 D19	L12 V _{CC}	J3 V _{SS}
H2 A18	P11 D18	M3 V _{CC}	J12 V _{SS}
H1 A17	N10 D17	M7 V _{CC}	J13 V _{SS}
G1 A16	N11 D16	M13 V _{CC}	M4 V _{SS}
F1 A15	M11 D15	N4 V _{CC}	M8 V _{SS}
E1 A14	P12 D14	N7 V _{CC}	M10 V _{SS}
E2 A13	P13 D13	P2 V _{CC}	N3 V _{SS}
E3 A12	N12 D12	P8 V _{CC}	P6 V _{SS}
D1 A11	N13 D11		P14 V _{SS}
D2 A10	M12 D10		
D3 A9	N14 D9	F12 CLK2	A4 N.C.
C1 A8	L13 D8		B4 N.C.
C2 A7	K12 D7	E14 ADS#	B6 N.C.
C3 A6	L14 D6		B12 N.C.
B2 A5	K13 D5	B10 W/R#	C6 N.C.
B3 A4	K14 D4	A11 D/C#	C7 N.C.
A3 A3	J14 D3	A12 M/IO#	E13 N.C.
C4 A2	H14 D2	C10 LOCK#	F13 N.C.
A13 BE3#	H13 D1		
B13 BE2#	H12 D0	D13 NA#	C8 PEREQ#
C13 BE1#		C14 BS16#	B9 BUSY#
E12 BE0#		G13 READY#	A8 ERROR#
C9 RESET	D14 HOLD M14 HLDA	B7 INTR	B8 NMI

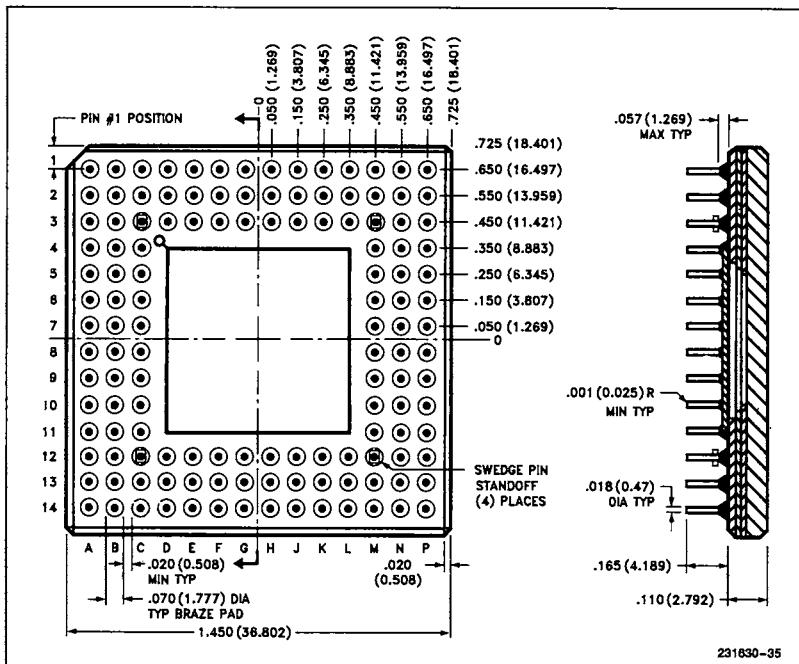


Figure 6-3. 132-Pin Ceramic PGA Package Dimensions

6.3 Package Dimensions and Mounting

The initial 80386 package is a 132-pin ceramic pin grid array (PGA). Pins of this package are arranged 0.100 inch (2.54mm) center-to-center, in a 14 x 14 matrix, three rows around.

A wide variety of available sockets allow low insertion force or zero insertion force mountings, and a choice of terminals such as soldertail, surface mount, or wire wrap. Several applicable sockets are listed in Table 6-2.

6.4 PACKAGE THERMAL SPECIFICATION

The 80386 is specified for operation when case temperature is within the range of 0°C–85°C. The case temperature may be measured in any environment,

to determine whether the 80386 is within specified operating range.

The PGA case temperature should be measured at the center of the top surface opposite the pins, as in Figure 6-4.

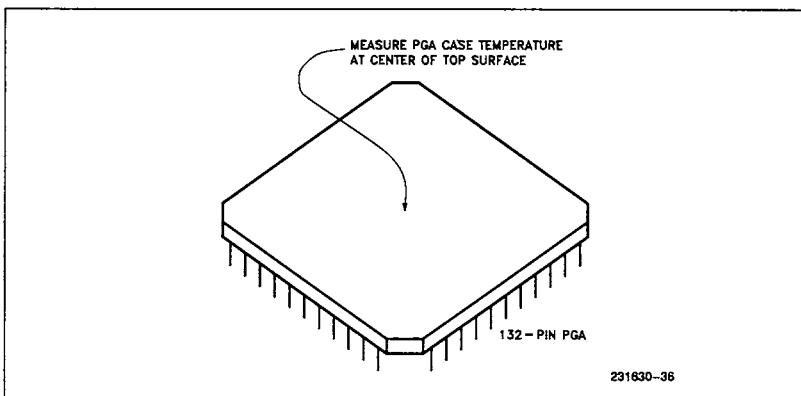


Figure 6-4. Measuring 80386 PGA Case Temperature

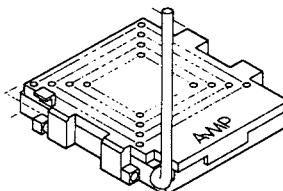
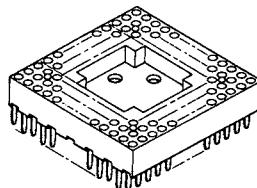
Table 6-2. Several Socket Options for 132-Pin PGA

- Low insertion force (LIF) soldertail
55274-1
- Amp tests indicate 50% reduction in insertion force compared to machined sockets

Other socket options

- Zero insertion force (ZIF) soldertail
55583-1
- Zero insertion force (ZIF) Burn-In version
55573-2

Amp Incorporated
(Harrisburg, PA 17105 U.S.A.
Phone 717-584-0100)



231630-45
Cam handle locks in low profile position when substrate is installed (handle UP for open and DOWN for closed positions)

courtesy Amp Incorporated

Table 6-2. Several Socket Options for 132-Pin PGA (Continued)

<p>Peel-A-Way™ Mylar and Kapton Socket Terminal Carriers</p> <ul style="list-style-type: none"> Low insertion force surface mount CS132-3TGG Low insertion force soldertail CS132-01TG Low insertion force wire-wrap CS132-02TG (two level) CS132-03TG (three-level) Low insertion force press-fit CS132-05TG <p>Advanced Interconnections (5 Division Street Warwick, RI 02818 U.S.A. Phone 401-885-0485)</p>	<p>Peel-A-Way Carrier No. 132: Kapton Carrier is KS132 Mylar Carrier is MS132</p> <p>Molded Plastic Body KS132 is shown below:</p> <p>FOOT PRINT NO. 132 1400 SQ. 100 THT 14 x 14 x 3 MM</p>	<table border="1"> <thead> <tr> <th>SOLDER TAIL -01</th> <th>LOW PROFILE -04</th> <th>PRESS FIT -05</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td>WIRE WRAP -02/-03</td> <td>SOLDER TAIL -03</td> <td>SURFACE MOUNTING -07</td> </tr> <tr> <td>PEEL-A-WAY</td> <td></td> <td></td> </tr> </tbody> </table>	SOLDER TAIL -01	LOW PROFILE -04	PRESS FIT -05				WIRE WRAP -02/-03	SOLDER TAIL -03	SURFACE MOUNTING -07	PEEL-A-WAY		
SOLDER TAIL -01	LOW PROFILE -04	PRESS FIT -05												
WIRE WRAP -02/-03	SOLDER TAIL -03	SURFACE MOUNTING -07												
PEEL-A-WAY														
		<p>courtesy Advanced Interconnections (Peel-A-Way Terminal Carriers U.S. Patent No. 4442938)</p> <p>231630-46</p>												

<ul style="list-style-type: none"> Low insertion force socket soldertail (for production use) 2XX-8578-00-3308 (new style) 2XX-8003-00-3302 (older style) Zero insertion force soldertail (for test and burn-in use) 2XX-6568-00-3302 <p>Textool Products Electronic Products Division/3M (1410 West Pioneer Drive Irving, Texas 75601 U.S.A. Phone 214-259-2676)</p>	

courtesy Textool Products/3M

231630-46

Table 6-3. 80386 PGA Package Thermal Characteristics

Parameter	Thermal Resistance — °C/Watt						
	0 (0)	50 (0.25)	100 (0.50)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)
θ Junction-to-Case (case measured at Fig. 6-4)	2	2	2	2	2	2	2
θ Case-to-Ambient (no heatsink)	19	18	17	15	12	10	9
θ Case-to-Ambient (with omnidirectional heatsink)	16	15	14	12	9	7	6
θ Case-to-Ambient (with unidirectional heatsink)	15	14	13	11	8	6	5

NOTES:

- Table 6-3 applies to 80386 PGA plugged into socket or soldered directly into board.
- $\theta_{JA} = \theta_{JC} + \theta_{CA}$

3. $\theta_{J-CAP} = 4^\circ\text{C/w}$ (approx.)
 $\theta_{J-PIN} = 4^\circ\text{C/w}$ (inner pins) (approx.)
 $\theta_{J-PIN} = 8^\circ\text{C/w}$ (outer pins) (approx.)



80386

ADVANCE INFORMATION

7. ELECTRICAL DATA

7.1 INTRODUCTION

The following sections describe recommended electrical connections for the 80386, and its electrical specifications.

7.2 POWER AND GROUNDING

7.2.1 Power Connections

The 80386 is implemented in CMOS III technology and has modest power requirements. However, its high clock frequency and 72 output buffers (address, data, control, and HLDA) can cause power surges as multiple output buffers drive new signal levels simultaneously. For clean on-chip distribution at high frequency, 20 V_{CC} and 21 V_{SS} pins separately feed functional units of the 80386.

Power and ground connections must be made to all external V_{CC} and GND pins of the 80386. On the circuit board, all V_{CC} pins must be connected on a V_{CC} plane. All V_{SS} pins must be likewise connected on a GND plane.

7.2.2 Power Decoupling Recommendations

Liberal decoupling capacitance should be placed near the 80386. The 80386 driving its 32-bit parallel address and data buses at high frequencies can cause transient power surges, particularly when driving large capacitive loads.

Low inductance capacitors and interconnects are recommended for best high frequency electrical performance. Inductance can be reduced by shortening circuit board traces between the 80386 and decou-

pling capacitors as much as possible. Capacitors specifically for PGA packages are also commercially available, for the lowest possible inductance.

7.2.3 Resistor Recommendations

The ERROR# and BUSY# inputs have resistor pull-ups of approximately 20 KΩ built-in to the 80386 to keep these signals negated when neither 80287 or 80387 are present in the system (or temporarily removed from its socket). The BS16# input also has an internal pullup resistor of approximately 20 KΩ, and the PEREQ input has an internal pulldown resistor of approximately 20 KΩ.

In typical designs, the external pullup resistors shown in Table 7-1 are recommended. However, a particular design may have reason to adjust the resistor values recommended here, or alter the use of pullup resistors in other ways.

7.2.4 Other Connection Recommendations

For reliable operation, always connect unused inputs to an appropriate signal level. N.C. pins should always remain unconnected.

Particularly when not using interrupts or bus hold, (as when first prototyping, perhaps) prevent any chance of spurious activity by connecting these associated inputs to GND:

Pin	Signal
B7	INTR
B8	NMI
D14	HOLD

If not using address pipelining, pullup D13 NA# to V_{CC}.

If not using 16-bit bus size, pullup C14 BS16# to V_{CC}.

Pullups in the range of 20 KΩ are recommended.

Table 7-1. Recommended Resistor Pullups to V_{CC}

Pin and Signal	Pullup Value	Purpose
E14 ADS#	20 KΩ ± 10%	Lightly Pull ADS# Negated During 80386 Hold Acknowledge States
C10 LOCK#	20 KΩ ± 10%	Lightly Pull LOCK# Negated During 80386 Hold Acknowledge States

7.3 MAXIMUM RATINGS

Table 7-2. Maximum Ratings

Parameter	80386-12 80386-16 Maximum Rating
Storage Temperature	-65°C to +150°C
Case Temperature Under Bias	-65°C to +110°C
Supply Voltage with Respect to V _{SS}	-0.5V to +6.5V
Voltage on Other Pins	-0.5V to V _{CC} + 0.5V

Table 7-2 is a stress rating only, and functional operation at the maximums is not guaranteed. Functional operating conditions are given in 7.4 D.C. Specifications and 7.5 A.C. Specifications.

Extended exposure to the Maximum Ratings may affect device reliability. Furthermore, although the 80386 contains protective circuitry to resist damage from static electric discharge, always take precautions to avoid high static voltages or electric fields.

7.4 D.C. SPECIFICATIONS

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

Table 7-3. 80386-20, 80386-16, D.C. Characteristics

Symbol	Parameter	80386-20 80386-16 Min	80386-20 80386-16 Max	Unit	Notes
V _{IL}	Input Low Voltage	-0.3	0.8	V	Note 1
V _{IH}	Input High Voltage	2.0	V _{CC} + 0.3	V	
V _{IICL}	CLK2 Input Low Voltage	-0.3	0.8	V	Note 1
V _{IICH}	CLK2 Input High Voltage	V _{CC} - 0.8	V _{CC} + 0.3	V	
V _{OL}	Output Low Voltage I _{OL} = 4 mA: A2-A31, D0-D31 I _{OL} = 5 mA: BEO# - BE3#, W/R#, D/C#, M/I/O#, LOCK#, ADS#, HLDA	0.45	0.8	V	
V _{OH}	Output High Voltage I _{OH} = -1 mA: A2-A31, D0-D31 I _{OH} = -0.9 mA: BEO# - BE3#, W/R#, D/C#, M/I/O#, LOCK#, ADS#, HLDA	2.4	V _{CC}	V	
I _{LI}	Input Leakage Current (for all pins except BS16#, PEREQ, BUSY#, and ERROR#)		±15	µA	0V ≤ V _{IN} ≤ V _{CC}
I _{JH}	Input Leakage Current (PEREQ only)		200	µA	V _{IH} = 2.4V (Note 2)
I _{JL}	Input Leakage Current (BS16#, BUSY#, and ERROR# only)		-400	µA	V _{IL} = 0.45V (Note 3)
I _{LO}	Output Leakage Current		±15	µA	0.45V ≤ V _{OUT} ≤ V _{CC}
I _{CC}	Supply Current CLK2 = 32 MHz with 80386-16 CLK2 = 40 MHz with 80386-20		460 550	mA mA	I _{CC} typ. = 370 mA I _{CC} typ. = 460 mA
C _{IN}	Input Capacitance		10	pF	F _c = 1 MHz (Note 4)
C _{OUT}	Output or I/O Capacitance		12	pF	F _c = 1 MHz (Note 4)
C _{CLK}	CLK2 Capacitance		20	pF	F _c = 1 MHz (Note 4)

NOTES:

1. The min value, -0.3, is not 100% tested.
2. PEREQ input has an internal pulldown resistor.
3. BS16#, BUSY# and ERROR# inputs each have an internal pullup resistor.
4. Not 100% tested.

7.5 A.C. SPECIFICATIONS

7.5.1 A.C. Spec Definitions

The A.C. specifications, given in Tables 7-4 and 7-5, consist of output delays, input setup requirements and input hold requirements. All A.C. specifications are relative to the CLK2 rising edge crossing the 2.0V level.

A.C. spec measurement is defined by Figure 7-1. Inputs must be driven to the voltage levels indicated by Figure 7-1 when A.C. specifications are measured. 80386 output delays are specified with minimum and maximum limits, measured as shown. The

minimum 80386 delay times are hold times provided to external circuitry. 80386 input setup and hold times are specified as minimums, defining the smallest acceptable sampling window. Within the sampling window, a synchronous input signal must be stable for correct 80386 operation.

Outputs NA#, W/R#, D/C#, M/IO#, LOCK#, BE0#-BE3#, A2-A31 and HLDA only change at the beginning of phase one. D0-D31 (write cycles) only change at the beginning of phase two. The READY#, HOLD, BUSY#, ERROR#, PEREQ and D0-D31 (read cycles) inputs are sampled at the beginning of phase one. The NA#, BS16#, INTR and NMI inputs are sampled at the beginning of phase two.

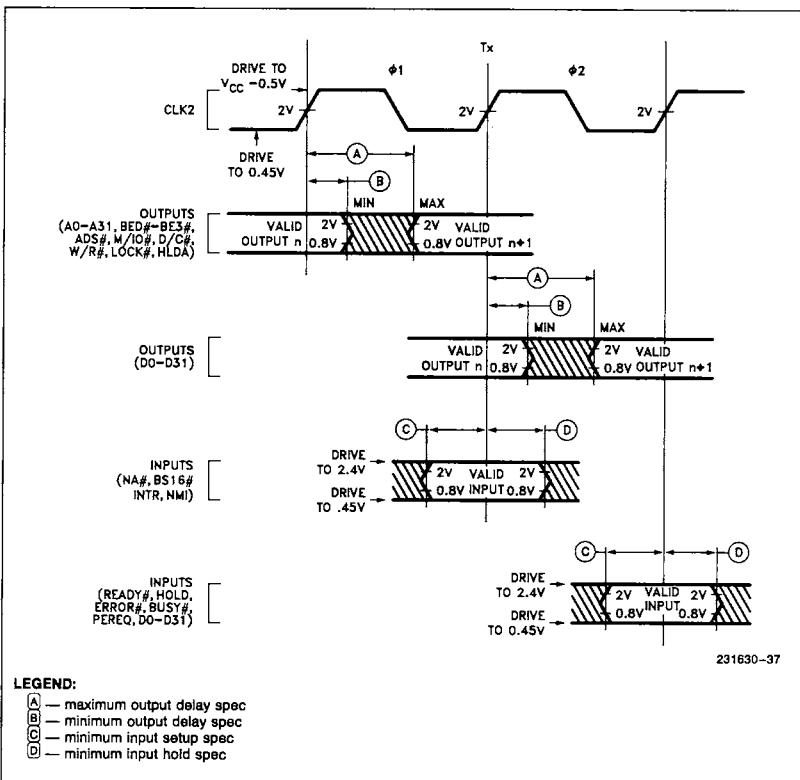


Figure 7-1. Drive Levels and Measurement Points for A.C. Specifications

7.5.2 A.C. Specification Tables

Functional Operating Range: $V_{CC} = 5V \pm 5\%$; $T_{CASE} = 0^\circ C$ to $85^\circ C$

Table 7-4. 80386-20 A.C. Characteristics

Symbol	Parameter	80386-20 Min	80386-20 Max	Unit	Ref. Figure	*	Notes
	Operating Frequency	4	20	MHz			Half of CLK2 Frequency
t_1	CLK2 Period	25	125	ns	7-3		
t_{2a}	CLK2 High Time	8		ns	7-3	at 2V	
t_{2b}	CLK2 High Time	5		ns		at ($V_{CC} - 0.8V$)	
t_{3a}	CLK2 Low Time	8		ns	7-3	at 2V	
t_{3b}	CLK2 Low Time	6		ns		at 0.8V	
t_4	CLK2 Fall Time						($V_{CC} - 0.8V$) to 0.8V
t_5	CLK2 Rise Time		8	ns	7-3	0.8V to ($V_{CC} - 0.8V$)	
t_6	A2-A31 Valid Delay	2		ns	7-5	$C_L = 120 pF$	
t_7	A2-A31 Float Delay			ns	7-6	(Note 1)	
t_8	BE0#-BE3#, LOCK# Valid Delay	2		ns	7-5	$C_L = 75 pF$	
t_9	BE0#-BE3#, LOCK# Float Delay		32	ns	7-6	(Note 1)	
t_{10}	W/R#, MREQ#, D/C# ADS# Valid Delay		30	ns	7-5	$C_L = 75 pF$	
t_{11}	W/R#, M/I/O#, D/C# ADS# Float Delay	4	30	ns	7-6	(Note 1)	
t_{12}	D0-D31 Write Data Valid Delay	2	40	ns	7-5	$C_L = 120 pF$	
t_{13}	D0-D31 Float Delay	2	27	ns	7-6	(Note 1)	
t_{14}	HLDA Valid Delay	4	30	ns	7-6	$C_L = 75 pF$	
t_{15}	NA# Setup Time	8		ns	7-4		
t_{16}	NA# Hold Time	20		ns	7-4		
t_{17}	BS16# Setup Time	12		ns	7-4		
t_{18}	BS16# Hold Time	20		ns	7-4		
t_{19}	READY# Setup Time	11		ns	7-4		
t_{20}	READY# Hold Time	3		ns	7-4		
t_{21}	D0-D31 Read Setup Time	10		ns	7-4		
t_{22}	D0-D31 Read Hold Time	5		ns	7-4		
t_{23}	HOLD Setup Time	16		ns	7-4		
t_{24}	HOLD Hold Time	4		ns	7-4		
t_{25}	RESET Setup Time	11		ns	7-7		

7.5.2 A.C. Specification Tables

Functional Operating Range: $V_{CC} = 5V \pm 5\%$; $T_{CASE} = 0^\circ C$ to $85^\circ C$ (Continued)

Table 7-4. 80386-20 A.C. Characteristics (Continued)

Symbol	Parameter	80386-20 Min	80386-20 Max	Unit	Ref. Figure	Notes
t_{26}	RESET Hold Time	3		ns	7-7	
t_{27}	NMI, INTR Setup Time	15		ns	7-4	(Note 2)
t_{28}	NMI, INTR Hold Time	15		ns	7-4	(Note 2)
t_{29}	PEREQ, ERROR #, BUSY # Setup Time	13		ns	7-4	(Note 2)
t_{30}	PEREQ, ERROR #, BUSY # Hold Time	4		ns	7-4	(Note 2)

NOTES:

1. Float condition occurs when maximum output current becomes less than I_{IO} in magnitude. Float delay is not 100% tested.
2. These inputs are allowed to be asynchronous to CLK2. The setup and hold specifications are given for testing purposes, to assure recognition within a specific CLK2 period.

Table 7-5. 80386-16 A.C. Characteristics

Symbol	Parameter	80386-16 Min	80386-16 Max	Unit	Ref. Figure	*	Notes
	Operating Frequency	4	16	MHz			Half of CLK2 Frequency
t_1	CLK2 Period	31	125	ns	7-3		
t_{2a}	CLK2 High Time	9		ns	7-3		at 2V
t_{2b}	CLK2 High Time	5		ns	7-3		at ($V_{CC} - 0.8V$)
t_{3a}	CLK2 Low Time	9		ns	7-3		at 2V
t_{3b}	CLK2 Low Time	7		ns	7-3		at 0.8V
t_4	CLK2 Fall Time	8		ns	7-3		($V_{CC} - 0.8V$) to 0.8V
t_5	CLK2 Rise Time	8		ns	7-3		0.8V to ($V_{CC} - 0.8V$)
t_6	A2-A31 Valid Delay	2		ns	7-5		$C_L = 120 \text{ pF}$
t_7	A2-A31 Float Delay	2		ns	7-6		(Note 1)
t_8	BE0#-BE3#, LOCK # Valid Delay	2	35	ns	7-5		$C_L = 75 \text{ pF}$
t_9	BE0#-BE3#, LOC Float Delay		40	ns	7-6		(Note 1)
t_{10}	W/R#, MREQ#, D/C# ADS# Valid Delay		35	ns	7-5		$C_L = 75 \text{ pF}$
t_{11}	W/R#, M/I/O#, M/C#, ADS# Float Delay	4	35	ns	7-6		(Note 1)
t_{12}	D0-D31 Write Data Valid Delay	2	50	ns	7-5		$C_L = 120 \text{ pF}$

7.5.2 A.C. Specification Tables

Functional Operating Range: $V_{CC} = 5V \pm 5\%$; $T_{CASE} = 0^\circ C$ to $85^\circ C$ (Continued)

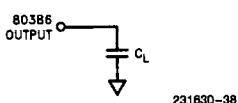
Table 7-5. 80386-16 A.C. Characteristics (Continued)

Symbol	Parameter	80386-16 Min	80386-16 Max	Unit	Ref. Figure	Notes
t_{13}	D0-D31 Write Data Float Delay	2	35	ns	7-6	(Note 1)
t_{14}	HLDA Valid Delay	4	35	ns	7-6	$C_L = 75 pF$
t_{15}	NA# Setup Time	10		ns	*7-4	
t_{16}	NA# Hold Time	20		ns	7-4	
t_{17}	BS16# Setup Time	12			7-4	
t_{18}	BS16# Hold Time	20			7-4	
t_{19}	READY# Setup Time	20		ns	7-4	
t_{20}	READY# Hold Time	3		ns	7-4	
t_{21}	D0-D31 Read Setup Time	10			7-4	
t_{22}	D0-D31 Read Hold Time	5			7-4	
t_{23}	HOLD Setup Time			ns	7-4	
t_{24}	HOLD Hold Time	4		ns	7-4	
t_{25}	RESET Setup Time	12		ns	7-7	
t_{26}	RESET Hold Time	3		ns	7-7	
t_{27}	NMI, INTR Setup Time	5		ns	7-4	(Note 2)
t_{28}	NMI, INTR Hold Time	5		ns	7-4	(Note 2)
t_{29}	PEREQ, EP#, BUSY# Setup Time			ns	7-4	(Note 2)
t_{30}	PEREQ, EP#, BUSY# Hold Time	4		ns	7-4	(Note 2)

NOTES:

1. Float condition occurs when maximum output current becomes less than I_{LO} in magnitude. Float delay is not 100% tested.
2. These inputs are allowed to be asynchronous to CLK2. The setup and hold specifications are given for testing purposes, to assure recognition within a specific CLK2 period.

7.5.3 A.C. Test Loads



$C_L = 120 pF$ on A2-A31, D0-D31
 $C_L = 75 pF$ on BE0#-BE3#, W/R#, M/I/O#, D/C#, ADS#,
LOCK#, HLDA
 C_L includes all parasitic capacitances.

Figure 7-2. A.C. Test Load

7.5.4 A.C. Timing Waveforms

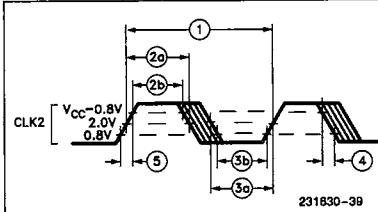


Figure 7-3. CLK2 Timing

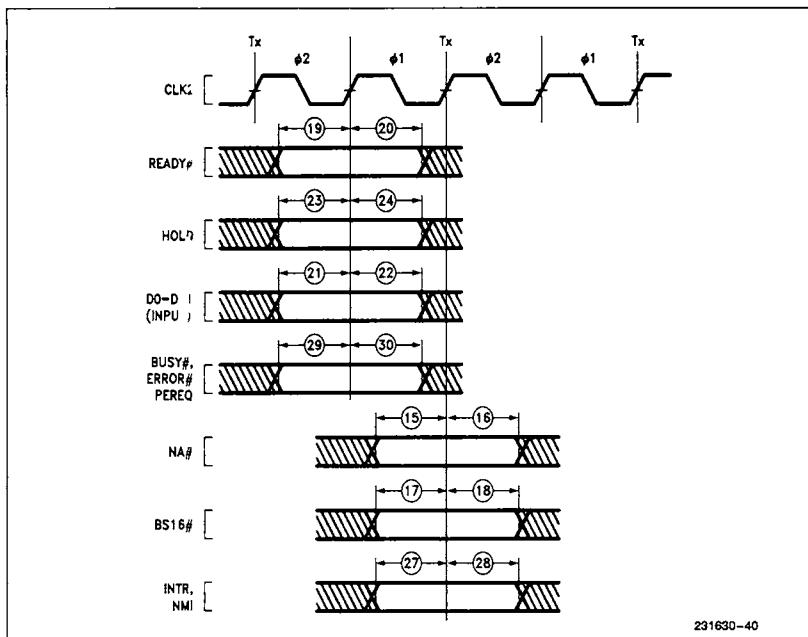


Figure 7-4. Input Setup and Hold Timing

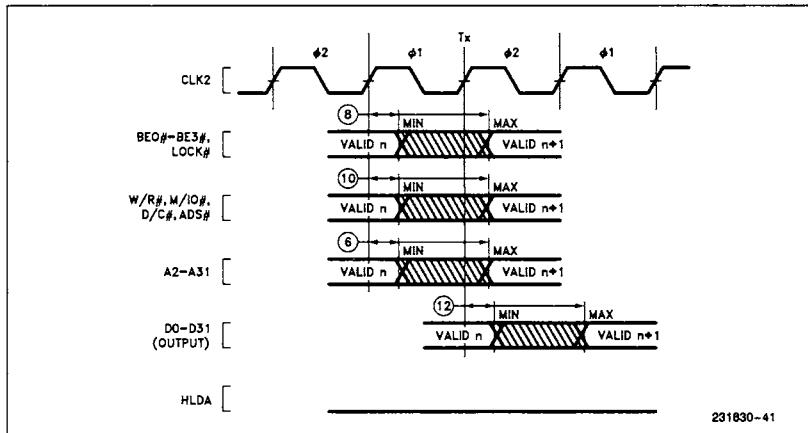


Figure 7-5. Output Valid Delay Timing

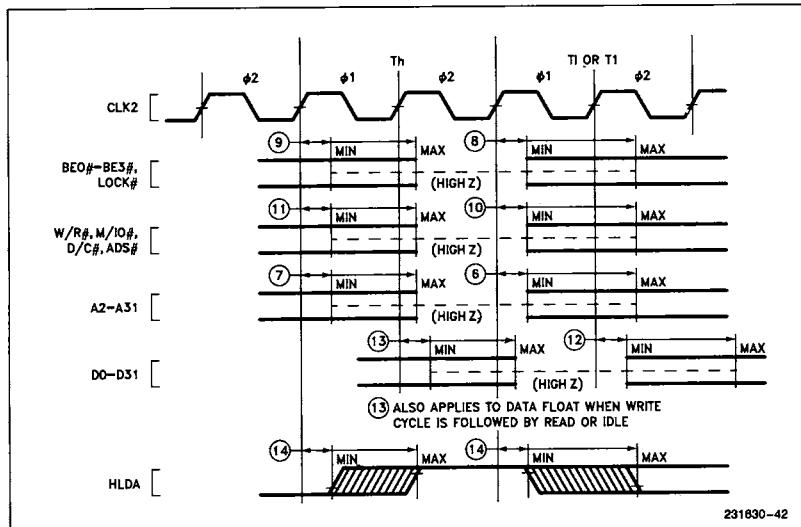


Figure 7-6. Output Float Delay and HLDA Valid Delay Timing

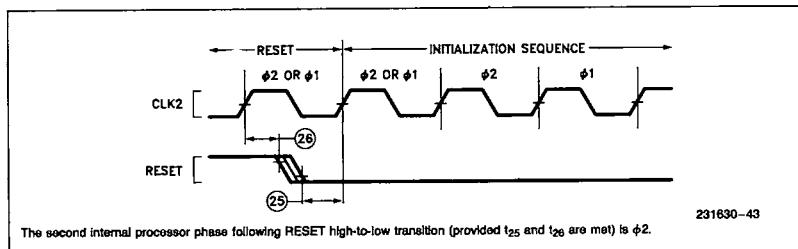


Figure 7-7. RESET Setup and Hold Timing, and Internal Phase

7.6 DESIGNING FOR ICE-386 USE

The 80386 in-circuit emulator product is ICE-386. Because of the high operating frequency of 80386 systems and ICE-386, there is no cable separating the ICE-386 probe module from the target system. The ICE-386 probe module has several electrical and mechanical characteristics that should be taken into consideration when designing the hardware.

Capacitive loading: ICE-386 adds up to 25 pF to each line.

Drive requirement: ICE-386 adds one standard TTL load on the CLK2 line, up to one advanced low-power Schottky TTL load per control signal line, and one advanced low-power Schottky TTL load per address, byte enable, and data line. These loads are within the probe module and are driven by the probe's 80386, which has standard drive and loading capability listed in Tables 7-3 and 7-4.

Power requirement: For noise immunity the ICE-386 probe is powered by the user system. The high-speed probe circuitry draws up to 0.7A plus the maximum 80386 I_{CC} from the user 80386 socket.

80386 location and orientation: The ICE-386 Processor Module (PM), and the Optional Isolation Board (OIB) used for extra electrical buffering of the

ICE initially, require clearance as illustrated in Figures 7-8 and 7-9, respectively. Figures 7-8 and 7-9 also illustrate the via holes in these modules for recommended orientation of a screw-actuated ZIF socket. Figure 7-10 illustrates the recommended orientation for a lever-actuated ZIF socket.

READY# drive: The ICE-386 system may be able to clear a user system READY# hang if the user's READY# driver is implemented with an open-collector or tri-state device.

Optional Interface Board (OIB) and CLK2 speed reduction: When the ICE-386 processor probe is first attached to an unverified user system, the OIB helps ICE-386 function in user systems with bus faults (shorted signals, etc.). After electrical verification it may be removed. Only when the OIB is installed, the user system must have a reduced CLK2 frequency of 16 MHz maximum.

Cache coherence: ICE-386 loads user memory by performing 80386 write cycles. Note that if the user system is not designed to update or invalidate its cache (if it has a cache) upon processor writes to memory, the cache could contain stale instruction code and/or data. For best use of ICE-386, the user should consider designing the cache (if any) to update itself automatically when processor writes occur, or find another method of maintaining cache data coherence with main user memory.

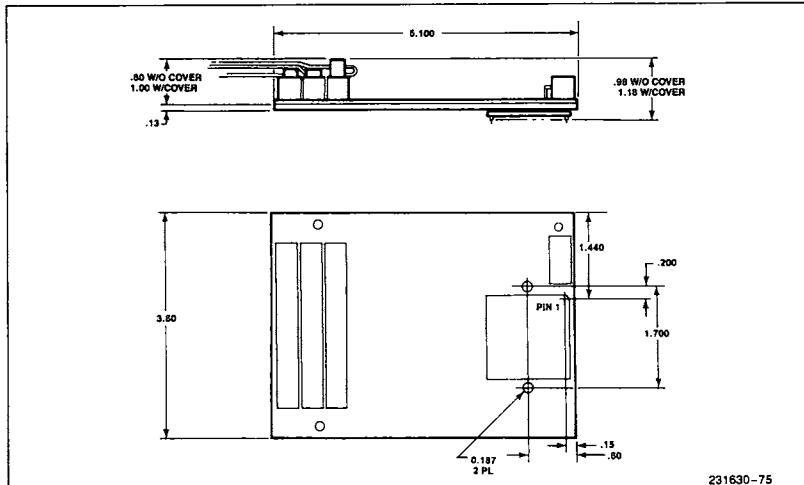


Figure 7-8. ICE-386 Processor Module Clearance Requirements (Inches)

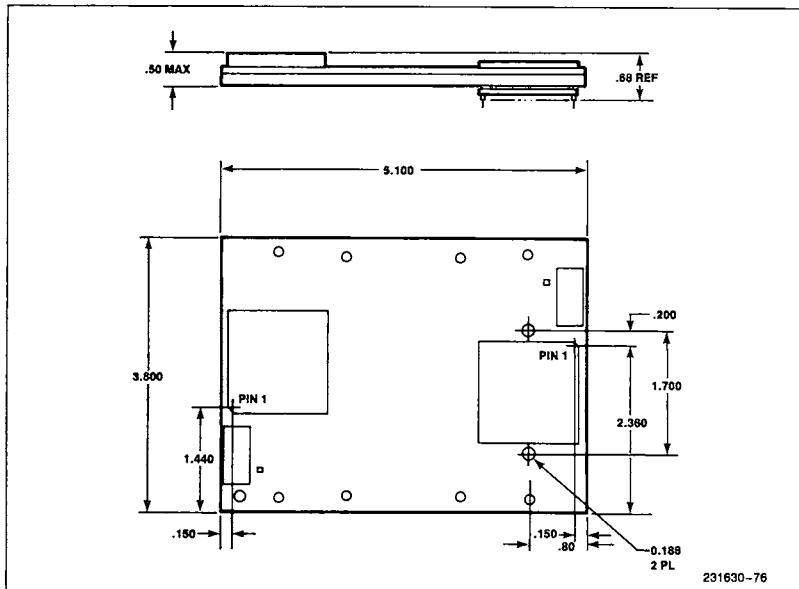


Figure 7-9. ICE-386 Optional Interface Module Clearance Requirements (Inches)

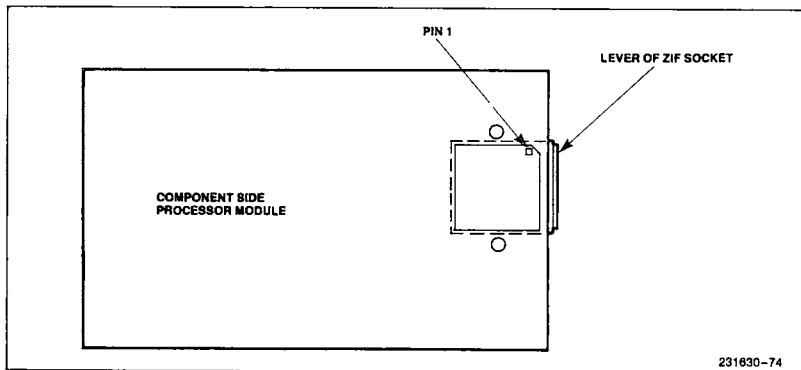


Figure 7-10. Recommended Orientation of Lever-Actuated ZIF Socket for ICE-386 Use



8. INSTRUCTION SET

This section describes the 80386 instruction set. A table lists all instructions along with instruction encoding diagrams and clock counts. Further details of the instruction encoding are then provided in the following sections, which completely describe the encoding structure and the definition of all fields occurring within 80386 instructions.

8.1 80386 INSTRUCTION ENCODING AND CLOCK COUNT SUMMARY

To calculate elapsed time for an instruction, multiply the instruction clock count, as listed in Table 8-1 below, by the processor clock period (e.g. 62.5 ns for an 80386-16 operating at 16 MHz (32 MHz CLK signal) and 50 ns for an 80386-20).

For more detailed information on the encodings of instructions refer to section 8.2 Instruction Encodings. Section 8.2 explains the general structure of instruction encodings, and defines exactly the encodings of all fields contained within the instruction.

Instruction Clock Count Assumptions

1. The instruction has been prefetched, decoded, and is ready for execution.
2. Bus cycles do not require wait states.
3. There are no local bus HOLD requests delaying processor access to the bus.
4. No exceptions are detected during instruction execution.
5. If an effective address is calculated, it does not use two general register components. One register, scaling and displacement can be used within the clock counts shown. However, if the effective address calculation uses two general register components, add 1 clock to the clock count shown.

Instruction Clock Count Notation

1. If two clock counts are given, the smaller refers to a register operand and the larger refers to a memory operand.
2. n = number of times repeated.
3. m = number of components in the next instruction executed, where the entire displacement (if any) counts as one component, the entire immediate data (if any) counts as one component, and each of the other bytes of the instruction and prefix(es) each count as one component.



80386

ADVANCE INFORMATION

Table 8-1. 80386 Instruction Set Clock Count Summary

INSTRUCTION	FORMAT	CLOCK COUNT		NOTES					
		Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode	Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode				
GENERAL DATA TRANSFER									
MOV = Move:									
Register to Register/Memory	<table border="1"><tr><td>10001100 w</td><td>mod reg</td><td>r/m</td></tr></table>	10001100 w	mod reg	r/m	2/2	2/2	b	h	
10001100 w	mod reg	r/m							
Register/Memory to Register	<table border="1"><tr><td>10001011 w</td><td>mod reg</td><td>r/m</td></tr></table>	10001011 w	mod reg	r/m	2/4	2/4	b	h	
10001011 w	mod reg	r/m							
Immediate to Register/Memory	<table border="1"><tr><td>11000111 w</td><td>mod 000</td><td>r/m</td></tr></table>	11000111 w	mod 000	r/m	2/2	2/2	b	h	
11000111 w	mod 000	r/m							
Immediate to Register (short form)	<table border="1"><tr><td>1011 w</td><td>reg</td><td>immediate data</td></tr></table>	1011 w	reg	immediate data	2	2			
1011 w	reg	immediate data							
Memory to Accumulator (short form)	<table border="1"><tr><td>10100000 w</td><td></td><td>full displacement</td></tr></table>	10100000 w		full displacement	4	4	b	h	
10100000 w		full displacement							
Accumulator to Memory (short form)	<table border="1"><tr><td>10100011 w</td><td></td><td>full displacement</td></tr></table>	10100011 w		full displacement	2	2	b	h	
10100011 w		full displacement							
Register Memory to Segment Register	<table border="1"><tr><td>10001110</td><td>mod sreg3</td><td>r/m</td></tr></table>	10001110	mod sreg3	r/m	2/5	18/19	b	h,i,j	
10001110	mod sreg3	r/m							
Segment Register to Register/Memory	<table border="1"><tr><td>10001100</td><td>mod sreg3</td><td>r/m</td></tr></table>	10001100	mod sreg3	r/m	2/2	2/2	b	h	
10001100	mod sreg3	r/m							
MOVBX == Move With Sign Extension									
Register From Register/Memory	<table border="1"><tr><td>00001111</td><td>10111111 w</td><td>mod reg</td><td>r/m</td></tr></table>	00001111	10111111 w	mod reg	r/m	3/8	3/8	b	h
00001111	10111111 w	mod reg	r/m						
MOVZX == Move With Zero Extension									
Register From Register/Memory	<table border="1"><tr><td>00001111</td><td>10110111 w</td><td>mod reg</td><td>r/m</td></tr></table>	00001111	10110111 w	mod reg	r/m	3/8	3/8	b	h
00001111	10110111 w	mod reg	r/m						
PUSH = Push:									
Register/Memory	<table border="1"><tr><td>11111111</td><td>mod 110</td><td>r/m</td></tr></table>	11111111	mod 110	r/m	5	5	b	h	
11111111	mod 110	r/m							
Register (short form)	<table border="1"><tr><td>01010</td><td>reg</td><td></td></tr></table>	01010	reg		2	2	b	h	
01010	reg								
Segment Register (ES, CS, SS or DS) (short form)	<table border="1"><tr><td>000sreg2110</td><td></td><td></td></tr></table>	000sreg2110			2	2	b	h	
000sreg2110									
Segment Register (ES, CS, SS or DS, FS or GS)	<table border="1"><tr><td>00001111</td><td>10 sreg30001</td><td></td></tr></table>	00001111	10 sreg30001		2	2	b	h	
00001111	10 sreg30001								
Immediate	<table border="1"><tr><td>011010s0</td><td></td><td>immediate data</td></tr></table>	011010s0		immediate data	2	2	b	h	
011010s0		immediate data							
PUSHA = Push All	<table border="1"><tr><td>01100000</td><td></td><td></td></tr></table>	01100000			18	18	b	h	
01100000									
POP = Pop									
Register/Memory	<table border="1"><tr><td>10001111</td><td>mod 000</td><td>r/m</td></tr></table>	10001111	mod 000	r/m	5	5	b	h	
10001111	mod 000	r/m							
Register (short form)	<table border="1"><tr><td>01011</td><td>reg</td><td></td></tr></table>	01011	reg		4	4	b	h	
01011	reg								
Segment Register (ES, CS, SS or DS) (short form)	<table border="1"><tr><td>000sreg2111</td><td></td><td></td></tr></table>	000sreg2111			7	21	b	h,i,j	
000sreg2111									
Segment Register (ES, CS, SS or DS, FS or GS)	<table border="1"><tr><td>00001111</td><td>10 sreg3001</td><td></td></tr></table>	00001111	10 sreg3001		7	21	b	h,i,j	
00001111	10 sreg3001								
POPA = Pop All	<table border="1"><tr><td>01100001</td><td></td><td></td></tr></table>	01100001			24	24	b	h	
01100001									
XCHG = Exchange									
Register/Memory With Register	<table border="1"><tr><td>10000111 w</td><td>mod reg</td><td>r/m</td></tr></table>	10000111 w	mod reg	r/m	3/5	3/5	b,f	f,h	
10000111 w	mod reg	r/m							
Register With Accumulator (short form)	<table border="1"><tr><td>10010</td><td>reg</td><td></td></tr></table>	10010	reg		3	3			
10010	reg								
IN = Input from:		Clk Count Virtual 8086 Mode							
Fixed Port	<table border="1"><tr><td>11100110 w</td><td>port number</td><td></td></tr></table>	11100110 w	port number		126	6*/28**		m	
11100110 w	port number								
Variable Port	<table border="1"><tr><td>1110110 w</td><td></td><td></td></tr></table>	1110110 w			127	7*/27**		m	
1110110 w									
OUT = Output to:									
Fixed Port	<table border="1"><tr><td>11100111 w</td><td>port number</td><td></td></tr></table>	11100111 w	port number		124	4*/24**		m	
11100111 w	port number								
Variable Port	<table border="1"><tr><td>1110111 w</td><td></td><td></td></tr></table>	1110111 w			125	5*/25**		m	
1110111 w									
LEA = Load EA to Register	<table border="1"><tr><td>10001101</td><td>mod reg</td><td>r/m</td></tr></table>	10001101	mod reg	r/m	2	2			
10001101	mod reg	r/m							

* If CPL ≤ IOPL

** If CPL > IOPL



80386

ADVANCE INFORMATION

Table 8-1. 80386 Instruction Set Clock Count Summary (Continued)

INSTRUCTION	FORMAT	CLOCK COUNT		NOTES					
		Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode	Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode				
SEGMENT CONTROL									
LDS = Load Pointer to DS	<table border="1"><tr><td>11000101</td><td>mod reg</td><td>r/m</td></tr></table>	11000101	mod reg	r/m	7	22	b	h, i, j	
11000101	mod reg	r/m							
LES = Load Pointer to ES	<table border="1"><tr><td>11000100</td><td>mod reg</td><td>r/m</td></tr></table>	11000100	mod reg	r/m	7	22	b	h, i, j	
11000100	mod reg	r/m							
LFS = Load Pointer to FS	<table border="1"><tr><td>00000111</td><td>10110100</td><td>mod reg</td><td>r/m</td></tr></table>	00000111	10110100	mod reg	r/m	7	25	b	h, i, j
00000111	10110100	mod reg	r/m						
LGS = Load Pointer to GS	<table border="1"><tr><td>00000111</td><td>10110101</td><td>mod reg</td><td>r/m</td></tr></table>	00000111	10110101	mod reg	r/m	7	25	b	h, i, j
00000111	10110101	mod reg	r/m						
LSS = Load Pointer to SS	<table border="1"><tr><td>00000111</td><td>10110010</td><td>mod reg</td><td>r/m</td></tr></table>	00000111	10110010	mod reg	r/m	7	22	b	h, i, j
00000111	10110010	mod reg	r/m						
FLAG CONTROL									
CLC = Clear Carry Flag	<table border="1"><tr><td>11111100</td></tr></table>	11111100	2	2					
11111100									
CLD = Clear Direction Flag	<table border="1"><tr><td>11111100</td></tr></table>	11111100	2	2					
11111100									
CLI = Clear Interrupt Enable Flag	<table border="1"><tr><td>11111010</td></tr></table>	11111010	8	8		m			
11111010									
CLTS = Clear Task Switched Flag	<table border="1"><tr><td>00000111</td><td>000000110</td></tr></table>	00000111	000000110	5	5	c	i		
00000111	000000110								
CMC = Complement Carry Flag	<table border="1"><tr><td>11110101</td></tr></table>	11110101	2	2					
11110101									
LAHF = Load AH Into Flag	<table border="1"><tr><td>10011111</td></tr></table>	10011111	2	2					
10011111									
POPF = Pop Flags	<table border="1"><tr><td>10011101</td></tr></table>	10011101	5	5	b	h, n			
10011101									
PUSHF = Push Flags	<table border="1"><tr><td>10011100</td></tr></table>	10011100	4	4	b	h			
10011100									
SAHF = Store AH Into Flags	<table border="1"><tr><td>10011110</td></tr></table>	10011110	3	3					
10011110									
STC = Set Carry Flag	<table border="1"><tr><td>11111001</td></tr></table>	11111001	2	2					
11111001									
STD = Set Direction Flag	<table border="1"><tr><td>11111001</td></tr></table>	11111001	2	2					
11111001									
STI = Set Interrupt Enable Flag	<table border="1"><tr><td>11111011</td></tr></table>	11111011	8	8		m			
11111011									
ARITHMETIC									
ADD = Add									
Register to Register	<table border="1"><tr><td>000000 dw</td><td>mod reg</td><td>r/m</td></tr></table>	000000 dw	mod reg	r/m	2	2			
000000 dw	mod reg	r/m							
Register to Memory	<table border="1"><tr><td>0000000 w</td><td>mod reg</td><td>r/m</td></tr></table>	0000000 w	mod reg	r/m	7	7	b	h	
0000000 w	mod reg	r/m							
Memory to Register	<table border="1"><tr><td>0000001 w</td><td>mod reg</td><td>r/m</td></tr></table>	0000001 w	mod reg	r/m	6	6	b	h	
0000001 w	mod reg	r/m							
Immediate to Register/Memory	<table border="1"><tr><td>1000008 w</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	1000008 w	mod 0 0 0	r/m	Immediate data	2/7	2/7	b	h
1000008 w	mod 0 0 0	r/m							
Immediate to Accumulator (short form)	<table border="1"><tr><td>0000010 w</td></tr></table>	0000010 w	immediate data	2	2				
0000010 w									
ADC = Add With Carry									
Register to Register	<table border="1"><tr><td>000100 dw</td><td>mod reg</td><td>r/m</td></tr></table>	000100 dw	mod reg	r/m	2	2			
000100 dw	mod reg	r/m							
Register to Memory	<table border="1"><tr><td>0001000 w</td><td>mod reg</td><td>r/m</td></tr></table>	0001000 w	mod reg	r/m	7	7	b	h	
0001000 w	mod reg	r/m							
Memory to Register	<table border="1"><tr><td>0001001 w</td><td>mod reg</td><td>r/m</td></tr></table>	0001001 w	mod reg	r/m	6	6	b	h	
0001001 w	mod reg	r/m							
Immediate to Register/Memory	<table border="1"><tr><td>1000008 w</td><td>mod 0 1 0</td><td>r/m</td></tr></table>	1000008 w	mod 0 1 0	r/m	immediate data	2/7	2/7	b	h
1000008 w	mod 0 1 0	r/m							
Immediate to Accumulator (short form)	<table border="1"><tr><td>0001010 w</td></tr></table>	0001010 w	immediate data	2	2				
0001010 w									
INC = Increment									
Register/Memory	<table border="1"><tr><td>1111111 w</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	1111111 w	mod 0 0 0	r/m	2/8	2/8	b	h	
1111111 w	mod 0 0 0	r/m							
Register (short form)	<table border="1"><tr><td>01000 reg</td></tr></table>	01000 reg	2	2					
01000 reg									
SUB = Subtract									
Register from Register	<table border="1"><tr><td>001010 dw</td><td>mod reg</td><td>r/m</td></tr></table>	001010 dw	mod reg	r/m	2	2			
001010 dw	mod reg	r/m							

Table 8-1. 80386 Instruction Set Clock Count Summary (Continued)

INSTRUCTION	FORMAT	CLOCK COUNT		NOTES		
		Real Address Mode or Virtual 80386 Mode	Protected Virtual Address Mode	Real Address Mode or Virtual 80386 Mode	Protected Virtual Address Mode	
ARITHMETIC (Continued)						
Register from Memory	0010100w mod reg r/m	7	7	b	h	
Memory from Register	0010101w mod reg r/m	8	8	b	h	
Immediate from Register/Memory	100000sw mod 101 r/m	Immediate data	2/7	2/7	b	h
Immediate from Accumulator (short form)	0010110w	Immediate data	2	2		
SBB = Subtract with Borrow						
Register from Register	000110dw mod reg r/m	2	2			
Register from Memory	0001100w mod reg r/m	7	7	b	h	
Memory from Register	0001101w mod reg r/m	6	6	b	h	
Immediate from Register/Memory	100000aw mod 011 r/m	Immediate data	2/7	2/7	b	h
Immediate from Accumulator (short form)	0001110w	Immediate data	2	2		
DEC = Decrement						
Register/Memory	1111111w reg 001 r/m	2/6	2/6	b	h	
Register (short form)	01001 reg	2	2			
CMP = Compare						
Register with Register	001110dw mod reg r/m	2	2			
Memory with Register	0011100w mod reg r/m	5	5	b	h	
Register with Memory	0011101w mod reg r/m	6	6	b	h	
Immediate with Register/Memory	100000aw mod 111 r/m	Immediate data	2/5	2/5	b	h
Immediate with Accumulator (short form)	0011110w	Immediate data	2	2		
NEG = Change Sign	1111011w mod 011 r/m	2/8	2/8	b	h	
AAC = ASCII Adjust for Add	00110111		4	4		
AAS = ASCII Adjust for Subtract	00111111		4	4		
DAA = Decimal Adjust for Add	00100111		4	4		
DAS = Decimal Adjust for Subtract	00101111		4	4		
MUL = Multiply (unsigned)						
Accumulator with Register/Memory	1111011w mod 100 r/m					
Multiplier-Byte						
-Word		12-17/15-20	12-17/15-20	b, d	d, h	
-Doubleword		12-25/15-28	12-25/15-28	b, d	d, h	
IMUL = Integer Multiply (signed)		12-41/15-44	12-41/15-44	b, d	d, h	
Accumulator with Register/Memory	1111011w mod 101 r/m					
Multiplier-Byte						
-Word		12-17/15-20	12-17/15-20	b, d	d, h	
-Doubleword		12-25/15-28	12-25/15-28	b, d	d, h	
Register with Register/Memory	00001111 10101111 mod reg r/m					
Multiplier-Byte						
-Word		12-17/15-20	12-17/15-20	b, d	d, h	
-Doubleword		12-25/15-28	12-25/15-28	b, d	d, h	
Register/Memory with Immediate to Register	011010s1 mod reg r/m	Immediate data				
-Word		13-26/14-27	13-26/14-27	b, d	d, h	
-Doubleword		13-42/14-43	13-42/14-43	b, d	d, h	



80386

ADVANCE INFORMATION

Table 8-1. 80386 Instruction Set Clock Count Summary (Continued)

INSTRUCTION	FORMAT	CLOCK COUNT		NOTES				
		Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode	Real Address Mode or Virtual 80386 Mode	Protected Virtual Address Mode			
ARITHMETIC (Continued)								
DIV = Divide (Unsigned)								
Accumulator by Register/Memory	<table border="1"><tr><td>11110111 w</td><td>mod 110 r/m</td></tr></table>	11110111 w	mod 110 r/m					
11110111 w	mod 110 r/m							
Divisor—Byte		14/17	14/17	b,e	e,h			
—Word		22/25	22/25	b,e	e,h			
—Doubleword		38/41	38/41	b,e	e,h			
IDIV = Integer Divide (Signed)								
Accumulator by Register/Memory	<table border="1"><tr><td>11110111 w</td><td>mod 111 r/m</td></tr></table>	11110111 w	mod 111 r/m					
11110111 w	mod 111 r/m							
Divisor—Byte		19/22	19/22	b,e	e,h			
—Word		27/30	27/30	b,e	e,h			
—Doubleword		43/48	43/48	b,e	e,h			
AAD = ASCII Adjust for Divide	<table border="1"><tr><td>11010101</td><td>00001010</td></tr></table>	11010101	00001010	19	19			
11010101	00001010							
AAM = ASCII Adjust for Multiply	<table border="1"><tr><td>11010100</td><td>00001010</td></tr></table>	11010100	00001010	17	17			
11010100	00001010							
CBW = Convert Byte to Word	<table border="1"><tr><td>10011000</td></tr></table>	10011000	3	3				
10011000								
CWD = Convert Word to Double Word	<table border="1"><tr><td>10011001</td></tr></table>	10011001	2	2				
10011001								
LOGIC								
Shift Rotate Instructions								
Not Through Carry (ROL, ROR, SAL, SAR, SHL, and SHR)								
Register/Memory by 1	<table border="1"><tr><td>1101000 w</td><td>mod TTT r/m</td></tr></table>	1101000 w	mod TTT r/m	3/7	3/7	b	h	
1101000 w	mod TTT r/m							
Register/Memory by CL	<table border="1"><tr><td>1101001 w</td><td>mod TTT r/m</td></tr></table>	1101001 w	mod TTT r/m	3/7	3/7	b	h	
1101001 w	mod TTT r/m							
Register/Memory by Immediate Count	<table border="1"><tr><td>1100000 w</td><td>mod TTT r/m</td></tr></table>	1100000 w	mod TTT r/m	3/7	3/7	b	h	
1100000 w	mod TTT r/m							
Immed 8-bit data								
Through Carry (RCL and RCR)								
Register/Memory by 1	<table border="1"><tr><td>1101000 w</td><td>mod TTT r/m</td></tr></table>	1101000 w	mod TTT r/m	9/10	9/10	b	h	
1101000 w	mod TTT r/m							
Register/Memory by CL	<table border="1"><tr><td>1101001 w</td><td>mod TTT r/m</td></tr></table>	1101001 w	mod TTT r/m	9/10	9/10	b	h	
1101001 w	mod TTT r/m							
Register/Memory by Immediate Count	<table border="1"><tr><td>1100000 w</td><td>mod TTT r/m</td></tr></table>	1100000 w	mod TTT r/m	9/10	9/10	b	h	
1100000 w	mod TTT r/m							
Immed 8-bit data								
TTT Instruction								
000 ROL								
001 ROR								
010 RCL								
011 RCR								
100 SHL/SAL								
101 SHR								
111 SAR								
SHLD = Shift Left Double								
Register/Memory by Immediate	<table border="1"><tr><td>00001111</td><td>10100100</td><td>mod reg r/m</td></tr></table>	00001111	10100100	mod reg r/m	3/7	3/7		
00001111	10100100	mod reg r/m						
Immed 8-bit data								
Register/Memory by CL	<table border="1"><tr><td>00001111</td><td>10100101</td><td>mod reg r/m</td></tr></table>	00001111	10100101	mod reg r/m	3/7	3/7		
00001111	10100101	mod reg r/m						
BHRD = Shift Right Double								
Register/Memory by Immediate	<table border="1"><tr><td>00001111</td><td>10101100</td><td>mod reg r/m</td></tr></table>	00001111	10101100	mod reg r/m	3/7	3/7		
00001111	10101100	mod reg r/m						
Immed 8-bit data								
Register/Memory by CL	<table border="1"><tr><td>00001111</td><td>10101101</td><td>mod reg r/m</td></tr></table>	00001111	10101101	mod reg r/m	3/7	3/7		
00001111	10101101	mod reg r/m						
AND = And								
Register to Register	<table border="1"><tr><td>001000 d w</td><td>mod reg r/m</td></tr></table>	001000 d w	mod reg r/m	2	2			
001000 d w	mod reg r/m							

Table 8-1. 80386 Instruction Set Clock Count Summary (Continued)

INSTRUCTION	FORMAT	CLOCK COUNT		NOTES				
		Real Address Mode or Virtual 8086 Mode	Protected Address Mode	Real Address Mode or Virtual 8086 Mode	Protected Address Mode			
LOGIC (Continued)								
Register to Memory	<table border="1"><tr><td>0010000w</td><td>mod reg</td><td>r/m</td></tr></table>	0010000w	mod reg	r/m	7	7	b	h
0010000w	mod reg	r/m						
Memory to Register	<table border="1"><tr><td>0010001w</td><td>mod reg</td><td>r/m</td></tr></table>	0010001w	mod reg	r/m	8	6	b	h
0010001w	mod reg	r/m						
Immediate to Register/Memory	<table border="1"><tr><td>1000000w</td><td>mod 100</td><td>r/m</td></tr></table>	1000000w	mod 100	r/m	2/7	2/7	b	h
1000000w	mod 100	r/m						
Immediate to Accumulator (Short Form)	<table border="1"><tr><td>0010010w</td><td></td><td>immediate data</td></tr></table>	0010010w		immediate data	2	2		
0010010w		immediate data						
TEST = And Function to Flags, No Result	<table border="1"><tr><td>1000010w</td><td>mod reg</td><td>r/m</td></tr></table>	1000010w	mod reg	r/m	2/5	2/5	b	h
1000010w	mod reg	r/m						
Register/Memory and Register	<table border="1"><tr><td>1000011w</td><td>mod reg</td><td>r/m</td></tr></table>	1000011w	mod reg	r/m	2/5	2/5	b	h
1000011w	mod reg	r/m						
Immediate Data and Register/Memory	<table border="1"><tr><td>1111011w</td><td>mod 000</td><td>r/m</td></tr></table>	1111011w	mod 000	r/m	2/5	2/5	b	h
1111011w	mod 000	r/m						
Immediate Data and Accumulator (Short Form)	<table border="1"><tr><td>1010100w</td><td></td><td>immediate data</td></tr></table>	1010100w		immediate data	2	2		
1010100w		immediate data						
OR = Or								
Register to Register	<table border="1"><tr><td>000010dw</td><td>mod reg</td><td>r/m</td></tr></table>	000010dw	mod reg	r/m	2	2		
000010dw	mod reg	r/m						
Register to Memory	<table border="1"><tr><td>0000100w</td><td>mod reg</td><td>r/m</td></tr></table>	0000100w	mod reg	r/m	7	7	b	h
0000100w	mod reg	r/m						
Memory to Register	<table border="1"><tr><td>0000101w</td><td>mod reg</td><td>r/m</td></tr></table>	0000101w	mod reg	r/m	6	8	b	h
0000101w	mod reg	r/m						
Immediate to Register/Memory	<table border="1"><tr><td>1000000w</td><td>mod 001</td><td>r/m</td></tr></table>	1000000w	mod 001	r/m	2/7	2/7	b	h
1000000w	mod 001	r/m						
Immediate to Accumulator (Short Form)	<table border="1"><tr><td>0000110w</td><td></td><td>immediate data</td></tr></table>	0000110w		immediate data	2	2		
0000110w		immediate data						
XOR = Exclusive Or								
Register to Register	<table border="1"><tr><td>001100dw</td><td>mod reg</td><td>r/m</td></tr></table>	001100dw	mod reg	r/m	2	2		
001100dw	mod reg	r/m						
Register to Memory	<table border="1"><tr><td>0011000w</td><td>mod reg</td><td>r/m</td></tr></table>	0011000w	mod reg	r/m	7	7	b	h
0011000w	mod reg	r/m						
Memory to Register	<table border="1"><tr><td>0011001w</td><td>mod reg</td><td>r/m</td></tr></table>	0011001w	mod reg	r/m	6	8	b	n
0011001w	mod reg	r/m						
Immediate to Register/Memory	<table border="1"><tr><td>1000000w</td><td>mod 110</td><td>r/m</td></tr></table>	1000000w	mod 110	r/m	2/7	2/7	b	h
1000000w	mod 110	r/m						
Immediate to Accumulator (Short Form)	<table border="1"><tr><td>0011010w</td><td></td><td>immediate data</td></tr></table>	0011010w		immediate data	2	2		
0011010w		immediate data						
NOT = Invert Register/Memory	<table border="1"><tr><td>1111011w</td><td>mod 010</td><td>r/m</td></tr></table>	1111011w	mod 010	r/m	2/8	2/8	b	h
1111011w	mod 010	r/m						
STRING MANIPULATION		Clik Count Virtual 8086 Mode						
CMPS = Compare Byte Word	<table border="1"><tr><td>1010011w</td></tr></table>	1010011w	10	10	b	h		
1010011w								
INS = Input Byte/Word from DX Port	<table border="1"><tr><td>0110110w</td></tr></table>	0110110w	120	9*/28**	b	h, m		
0110110w								
LODS = Load Byte/Word to AL/AX/EAX	<table border="1"><tr><td>1010110w</td></tr></table>	1010110w		5	5	b		
1010110w								
MOS = Move Byte Word	<table border="1"><tr><td>1010010w</td></tr></table>	1010010w		7	7	b		
1010010w								
OUTS = Output Byte/Word to DX Port	<table border="1"><tr><td>0110111w</td></tr></table>	0110111w	128	8*/28**	b	h, m		
0110111w								
SCAS = Scan Byte Word	<table border="1"><tr><td>1010111w</td></tr></table>	1010111w		7	7	b		
1010111w								
STOS = Store Byte/Word from AL/AX/EX	<table border="1"><tr><td>1010101w</td></tr></table>	1010101w		4	4	b		
1010101w								
XLAT = Translate String	<table border="1"><tr><td>11010111</td></tr></table>	11010111		5	5			
11010111								
REPEATED STRING MANIPULATION								
Repeated by Count in CX or ECX								
REPE CMPS = Compare String (Find Non-Match)	<table border="1"><tr><td>111100111</td><td>1010011w</td></tr></table>	111100111	1010011w		5+9n	5+9n	b	
111100111	1010011w							

* If CPL ≤ IOPL

** If CPL > IOPL



80386

ADVANCE INFORMATION

Table 8-1. 80386 Instruction Set Clock Count Summary (Continued)

INSTRUCTION	FORMAT	CLOCK COUNT		NOTES	
		Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode	Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode
REPEATED STRING MANIPULATION (Continued)					
REPNE CMPS = Compare String (Find Match)	11110010 1010011w	Clock Count Virtual 8086 Mode	5+8n	5+8n	b h
REP INS = Input String	11110010 0110110w	127+8n	13+8n	7+8n*/27+8n**	b h, m
REP LODS = Load String	11110010 1010110w		5+8n	5+8n	b h
REP MOVS = Move String	11110010 1010010w		7+4n	7+4n	b h
REP OUTS = Output String	11110010 0110111w		12+5n	8+5n*/28+5n**	b h, m
REPE SCAS = Scan String (Find Non-AL/AE/AEX)	11110011 1010111w		5+8n	5+8n	b h
REPNE SCAS = Scan String (Find AL/AE/AEX)	11110010 1010111w		5+8n	5+8n	b h
REP STOS = Store String	11110010 1010101w		5+5n	5+5n	b h
BIT MANIPULATION					
BSF = Scan Bit Forward	00001111 10111100 mod reg r/m		10+3n	10+3n	b h
BSR = Scan Bit Reverse	00001111 10111101 mod reg r/m		10+3n	10+3n	b h
BT = Test Bit					
Register/Memory, Immediate	00001111 10111010 mod 100 r/m immed 8-bit data		3/8	3/8	b h
Register/Memory, Register	00001111 10100011 mod reg r/m		3/12	3/12	b h
BTC = Test Bit and Complement					
Register/Memory, Immediate	00001111 10111010 mod 111 r/m immed 8-bit data		8/8	8/8	b h
Register/Memory, Register	00001111 10111011 mod reg r/m		8/13	8/13	b h
BTR = Test Bit and Reset					
Register/Memory, Immediate	00001111 10111010 mod 110 r/m immed 8-bit data		6/8	6/8	b h
Register/Memory, Register	00001111 10100011 mod reg r/m		8/13	6/13	b h
BTST = Test Bit and Set					
Register/Memory, Immediate	00001111 10111010 mod 101 r/m immed 8-bit data		6/8	6/8	b h
Register/Memory, Register	00001111 10101011 mod reg r/m		6/13	6/13	b h
CONTROL TRANSFER					
CALL = Call					
Direct Within Segment	11101000 full displacement		7+m	7+m	b r
Register/Memory					
Indirect Within Segment	11111111 mod 010 r/m		7+m/ 10+m	7+m/ 10+m	b h, r
Direct Intersegment	10011010 unsigned full offset, selector		17+m	34+m	b j,k,r

Notes:

† Clock count shown applies if I/O permission allows I/O to the port in virtual 8086 mode. If I/O bit map denies permission exception 13 fault occurs; refer to clock counts for INT 3 instruction.

* If CPL ≤ IOPL

** If CPL > IOPL



Table 8-1. 80386 Instruction Set Clock Count Summary (Continued)

INSTRUCTION	FORMAT	CLOCK COUNT		NOTES				
		Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode	Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode			
CONTROL TRANSFER (Continued)								
Protected Mode Only (Direct Intersegment)								
Via Call Gate to Same Privilege Level		52 + m			h,j,k,r			
Via Call Gate to Different Privilege Level, (No Parameters)		86 + m			h,j,k,r			
Via Call Gate to Different Privilege Level, (x Parameters)		94 + 4x + m			h,j,k,r			
From 286 Task to 286 TSS		273			h,j,k,r			
From 286 Task to 386 TSS		298			h,j,k,r			
From 286 Task to Virtual 8086 Task (386 TSS)		217			h,j,k,r			
From 386 Task to 286 TSS		273			h,j,k,r			
From 386 Task to 386 TSS		300			h,j,k,r			
From 386 Task to Virtual 8086 Task (386 TSS)		217			h,j,k,r			
Indirect Intersegment	<table border="1"><tr><td>11111111</td><td>mod 0 1 1</td><td>r/m</td></tr></table>	11111111	mod 0 1 1	r/m	22 + m	38 + m	b	h,j,k,r
11111111	mod 0 1 1	r/m						
Protected Mode Only (Indirect Intersegment)								
Via Call Gate to Same Privilege Level		56 + m			h,j,k,r			
Via Call Gate to Different Privilege Level, (No Parameters)		90 + m			h,j,k,r			
Via Call Gate to Different Privilege Level, (x Parameters)		98 + 4x + m			h,j,k,r			
From 286 Task to 286 TSS		278			h,j,k,r			
From 286 Task to 386 TSS		303			h,j,k,r			
From 286 Task to Virtual 8086 Task (386 TSS)		221			h,j,k,r			
From 386 Task to 286 TSS		278			h,j,k,r			
From 386 Task to 386 TSS		305			h,j,k,r			
From 386 Task to Virtual 8086 Task (386 TSS)		221			h,j,k,r			
JMP = Unconditional Jump								
Short	<table border="1"><tr><td>11101001</td><td>8-bit displacement</td></tr></table>	11101001	8-bit displacement	7 + m	7 + m		r	
11101001	8-bit displacement							
Direct within Segment	<table border="1"><tr><td>11101001</td><td>full displacement</td></tr></table>	11101001	full displacement	7 + m	7 + m		r	
11101001	full displacement							
Register/Memory Indirect within Segment	<table border="1"><tr><td>11111111</td><td>mod 1 0 0</td><td>r/m</td></tr></table>	11111111	mod 1 0 0	r/m	7 + m/ 10 + m	7 + m/ 10 + m	b	h,r
11111111	mod 1 0 0	r/m						
Direct Intersegment	<table border="1"><tr><td>11101010</td><td>unsigned full offset, selector</td></tr></table>	11101010	unsigned full offset, selector	12 + m	27 + m		j,k,r	
11101010	unsigned full offset, selector							
Protected Mode Only (Direct Intersegment)								
Via Call Gate to Same Privilege Level		45 + m			h,j,k,r			
From 286 Task to 286 TSS		274			h,j,k,r			
From 286 Task to 386 TSS		301			h,j,k,r			
From 286 Task to Virtual 8086 Task (386 TSS)		218			h,j,k,r			
From 386 Task to 286 TSS		270			h,j,k,r			
From 386 Task to 386 TSS		303			h,j,k,r			
From 386 Task to Virtual 8086 Task (386 TSS)		220			h,j,k,r			
Indirect Intersegment	<table border="1"><tr><td>11111111</td><td>mod 1 0 1</td><td>r/m</td></tr></table>	11111111	mod 1 0 1	r/m	17 + m	31 + m	b	h,j,k,r
11111111	mod 1 0 1	r/m						
Protected Mode Only (Indirect Intersegment)								
Via Call Gate to Same Privilege Level		49 + m			h,j,k,r			
From 286 Task to 286 TSS		279			h,j,k,r			
From 286 Task to 386 TSS		306			h,j,k,r			
From 286 Task to Virtual 8086 Task (386 TSS)		222			h,j,k,r			
From 386 Task to 286 TSS		275			h,j,k,r			
From 386 Task to 386 TSS		308			h,j,k,r			
From 386 Task to Virtual 8086 Task (386 TSS)		224			h,j,k,r			



80386

ADVANCE INFORMATION

Table 8-1. 80386 Instruction Set Clock Count Summary (Continued)

INSTRUCTION	FORMAT	CLOCK COUNT		NOTES			
		Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode	Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode		
CONTROL TRANSFER (Continued)							
RET = Return from CALL:							
Within Segment	<table border="1"><tr><td>11000011</td></tr></table>	11000011	10 + m	10 + m	b	g, h, r	
11000011							
Within Segment Adding Immediate to SP	<table border="1"><tr><td>11000010</td><td>16-bit displ</td></tr></table>	11000010	16-bit displ	10 + m	10 + m	b	g, h, r
11000010	16-bit displ						
Intersegment	<table border="1"><tr><td>11001011</td></tr></table>	11001011	18 + m	32+m	b	g, h, j, k, r	
11001011							
Intersegment Adding Immediate to SP	<table border="1"><tr><td>11001010</td><td>16-bit displ</td></tr></table>	11001010	16-bit displ	18 + m	32+m	b	g, h, j, k, r
11001010	16-bit displ						
Protected Mode Only (RET):							
to Different Privilege Level				68			
Intersegment				68			
Intersegment Adding Immediate to SP					h, j, k, r h, j, k, r		
CONDITIONAL JUMPS							
NOTE: Times Are Jump "Taken or Not Taken"							
JO = Jump on Overflow							
8-Bit Displacement	<table border="1"><tr><td>01110000</td><td>8-bit displ</td></tr></table>	01110000	8-bit displ	7 + m or 3	7 + m or 3		r
01110000	8-bit displ						
Full Displacement	<table border="1"><tr><td>00001111</td><td>10000000</td></tr></table>	00001111	10000000	full displacement	7 + m or 3	7 + m or 3	r
00001111	10000000						
JNO = Jump on Not Overflow							
8-Bit Displacement	<table border="1"><tr><td>01110001</td><td>8-bit displ</td></tr></table>	01110001	8-bit displ	7 + m or 3	7 + m or 3		r
01110001	8-bit displ						
Full Displacement	<table border="1"><tr><td>00001111</td><td>10000001</td></tr></table>	00001111	10000001	full displacement	7 + m or 3	7 + m or 3	r
00001111	10000001						
JB/JNAE = Jump on Below/Not Above or Equal							
8-Bit Displacement	<table border="1"><tr><td>01110010</td><td>8-bit displ</td></tr></table>	01110010	8-bit displ	7 + m or 3	7 + m or 3		r
01110010	8-bit displ						
Full Displacement	<table border="1"><tr><td>00001111</td><td>10000010</td></tr></table>	00001111	10000010	full displacement	7 + m or 3	7 + m or 3	r
00001111	10000010						
JNB/JAE = Jump on Not Below/Above or Equal							
8-Bit Displacement	<table border="1"><tr><td>01110011</td><td>8-bit displ</td></tr></table>	01110011	8-bit displ	7 + m or 3	7 + m or 3		r
01110011	8-bit displ						
Full Displacement	<table border="1"><tr><td>00001111</td><td>10000011</td></tr></table>	00001111	10000011	full displacement	7 + m or 3	7 + m or 3	r
00001111	10000011						
JE/JZ = Jump on Equal/Zero							
8-Bit Displacement	<table border="1"><tr><td>01110100</td><td>8-bit displ</td></tr></table>	01110100	8-bit displ	7 + m or 3	7 + m or 3		r
01110100	8-bit displ						
Full Displacement	<table border="1"><tr><td>00001111</td><td>10000100</td></tr></table>	00001111	10000100	full displacement	7 + m or 3	7 + m or 3	r
00001111	10000100						
JNE/JNZ = Jump on Not Equal/Not Zero							
8-Bit Displacement	<table border="1"><tr><td>01110101</td><td>8-bit displ</td></tr></table>	01110101	8-bit displ	7 + m or 3	7 + m or 3		r
01110101	8-bit displ						
Full Displacement	<table border="1"><tr><td>00001111</td><td>10000101</td></tr></table>	00001111	10000101	full displacement	7 + m or 3	7 + m or 3	r
00001111	10000101						
JBE/JNA = Jump on Below or Equal/Not Above							
8-Bit Displacement	<table border="1"><tr><td>01110110</td><td>8-bit displ</td></tr></table>	01110110	8-bit displ	7 + m or 3	7 + m or 3		r
01110110	8-bit displ						
Full Displacement	<table border="1"><tr><td>00001111</td><td>10000110</td></tr></table>	00001111	10000110	full displacement	7 + m or 3	7 + m or 3	r
00001111	10000110						
JS = Jump on Sign							
8-Bit Displacement	<table border="1"><tr><td>01111000</td><td>8-bit displ</td></tr></table>	01111000	8-bit displ	7 + m or 3	7 + m or 3		r
01111000	8-bit displ						
Full Displacement	<table border="1"><tr><td>00001111</td><td>10001000</td></tr></table>	00001111	10001000	full displacement	7 + m or 3	7 + m or 3	r
00001111	10001000						

Table 8-1. 80386 Instruction Set Clock Count Summary (Continued)

INSTRUCTION	FORMAT	CLOCK COUNT		NOTES				
		Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode	Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode			
CONDITIONAL JUMPS (Continued)								
JNS = Jump on Not Sign								
8-Bit Displacement	<table border="1"><tr><td>01111001</td><td>8-bit displ</td></tr></table>	01111001	8-bit displ	7 + m or 3	7 + m or 3		r	
01111001	8-bit displ							
Full Displacement	<table border="1"><tr><td>00001111</td><td>10001001</td></tr></table>	00001111	10001001	full displacement	7 + m or 3	7 + m or 3	r	
00001111	10001001							
JP/JPE = Jump on Parity/Parity Even								
8-Bit Displacement	<table border="1"><tr><td>01111010</td><td>8-bit displ</td></tr></table>	01111010	8-bit displ	7 + m or 3	7 + m or 3		r	
01111010	8-bit displ							
Full Displacement	<table border="1"><tr><td>00001111</td><td>10001010</td></tr></table>	00001111	10001010	full displacement	7 + m or 3	7 + m or 3	r	
00001111	10001010							
JNP/JPO = Jump on Not Parity/Parity Odd								
8-Bit Displacement	<table border="1"><tr><td>01111011</td><td>8-bit displ</td></tr></table>	01111011	8-bit displ	7 + m or 3	7 + m or 3		r	
01111011	8-bit displ							
Full Displacement	<table border="1"><tr><td>00001111</td><td>10001011</td></tr></table>	00001111	10001011	full displacement	7 + m or 3	7 + m or 3	r	
00001111	10001011							
JL/JNGE = Jump on Less/Not Greater or Equal								
8-Bit Displacement	<table border="1"><tr><td>01111100</td><td>8-bit displ</td></tr></table>	01111100	8-bit displ	7 + m or 3	7 + m or 3		r	
01111100	8-bit displ							
Full Displacement	<table border="1"><tr><td>00001111</td><td>10001100</td></tr></table>	00001111	10001100	full displacement	7 + m or 3	7 + m or 3	r	
00001111	10001100							
JNL/JQE = Jump on Not Less/Greater or Equal								
8-Bit Displacement	<table border="1"><tr><td>01111101</td><td>8-bit displ</td></tr></table>	01111101	8-bit displ	7 + m or 3	7 + m or 3		r	
01111101	8-bit displ							
Full Displacement	<table border="1"><tr><td>00001111</td><td>10001101</td></tr></table>	00001111	10001101	full displacement	7 + m or 3	7 + m or 3	r	
00001111	10001101							
JLE/JNG = Jump on Less or Equal/Not Greater								
8-Bit Displacement	<table border="1"><tr><td>01111110</td><td>8-bit displ</td></tr></table>	01111110	8-bit displ	7 + m or 3	7 + m or 3		r	
01111110	8-bit displ							
Full Displacement	<table border="1"><tr><td>00001111</td><td>10001110</td></tr></table>	00001111	10001110	full displacement	7 + m or 3	7 + m or 3	r	
00001111	10001110							
JNLE/JG = Jump on Not Less or Equal/Greater								
8-Bit Displacement	<table border="1"><tr><td>01111111</td><td>8-bit displ</td></tr></table>	01111111	8-bit displ	7 + m or 3	7 + m or 3		r	
01111111	8-bit displ							
Full Displacement	<table border="1"><tr><td>00001111</td><td>10001111</td></tr></table>	00001111	10001111	full displacement	7 + m or 3	7 + m or 3	r	
00001111	10001111							
JCXZ = Jump on CX Zero								
JECXZ = Jump on ECX Zero	<table border="1"><tr><td>11100011</td><td>8-bit displ</td></tr></table>	11100011	8-bit displ	8 + m or 5	8 + m or 5		r	
11100011	8-bit displ							
(Address Size Prefix Differentiates JCXZ from JECXZ)								
LOOP = Loop CX Times	<table border="1"><tr><td>11100010</td><td>8-bit displ</td></tr></table>	11100010	8-bit displ	11 + m	11 + m		r	
11100010	8-bit displ							
LOOPZ/LOOPE = Loop with Zero/Equal	<table border="1"><tr><td>11100001</td><td>8-bit displ</td></tr></table>	11100001	8-bit displ	11 + m	11 + m		r	
11100001	8-bit displ							
LOOPNZ/LOOPNE = Loop While Not Zero	<table border="1"><tr><td>11100000</td><td>8-bit displ</td></tr></table>	11100000	8-bit displ	11 + m	11 + m		r	
11100000	8-bit displ							
CONDITIONAL BYTE SET								
NOTE: Times Are Register/Memory								
SETO = Set Byte on Overflow								
To Register/Memory	<table border="1"><tr><td>00001111</td><td>10010000</td><td>mod 000 r/m</td></tr></table>	00001111	10010000	mod 000 r/m	4/5	4/5		h
00001111	10010000	mod 000 r/m						
SETNO = Set Byte on Not Overflow	<table border="1"><tr><td>00001111</td><td>10010001</td><td>mod 000 r/m</td></tr></table>	00001111	10010001	mod 000 r/m	4/5	4/5		h
00001111	10010001	mod 000 r/m						
SETB/SETNAE = Set Byte on Below/Not Above or Equal	<table border="1"><tr><td>00001111</td><td>10010010</td><td>mod 000 r/m</td></tr></table>	00001111	10010010	mod 000 r/m	4/5	4/5		h
00001111	10010010	mod 000 r/m						



80386

ADVANCE INFORMATION

Table 8-1. 80386 Instruction Set Clock Count Summary (Continued)

INSTRUCTION	FORMAT	CLOCK COUNT		NOTES					
		Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode	Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode				
CONDITIONAL BYTE SET (Continued)									
SETNB = Set Byte on Not Below/Above or Equal To Register/Memory	<table border="1"><tr><td>00001111</td><td>10010011</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	00001111	10010011	mod 0 0 0	r/m	4/5	4/5		h
00001111	10010011	mod 0 0 0	r/m						
SETE/SETZ = Set Byte on Equal/Zero To Register/Memory	<table border="1"><tr><td>00001111</td><td>10010100</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	00001111	10010100	mod 0 0 0	r/m	4/5	4/5		h
00001111	10010100	mod 0 0 0	r/m						
SETNE/SETNZ = Set Byte on Not Equal/Not Zero To Register/Memory	<table border="1"><tr><td>00001111</td><td>10010101</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	00001111	10010101	mod 0 0 0	r/m	4/5	4/5		h
00001111	10010101	mod 0 0 0	r/m						
SETBE/SETNA = Set Byte on Below or Equal/Not Above To Register/Memory	<table border="1"><tr><td>00001111</td><td>10010110</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	00001111	10010110	mod 0 0 0	r/m	4/5	4/5		h
00001111	10010110	mod 0 0 0	r/m						
SETNBE/SETA = Set Byte on Not Below or Equal/Above To Register/Memory	<table border="1"><tr><td>00001111</td><td>10010111</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	00001111	10010111	mod 0 0 0	r/m	4/5	4/5		h
00001111	10010111	mod 0 0 0	r/m						
SETS = Set Byte on Sign To Register/Memory	<table border="1"><tr><td>00001111</td><td>10011000</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	00001111	10011000	mod 0 0 0	r/m	4/5	4/5		h
00001111	10011000	mod 0 0 0	r/m						
SETNS = Set Byte on Not Sign To Register/Memory	<table border="1"><tr><td>00001111</td><td>10011001</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	00001111	10011001	mod 0 0 0	r/m	4/5	4/5		h
00001111	10011001	mod 0 0 0	r/m						
SETP/SETPE = Set Byte on Parity/Parity Even To Register/Memory	<table border="1"><tr><td>00001111</td><td>10011010</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	00001111	10011010	mod 0 0 0	r/m	4/5	4/5		h
00001111	10011010	mod 0 0 0	r/m						
SETNP/SETPO = Set Byte on Not Parity/Parity Odd To Register/Memory	<table border="1"><tr><td>00001111</td><td>10011011</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	00001111	10011011	mod 0 0 0	r/m	4/5	4/5		h
00001111	10011011	mod 0 0 0	r/m						
SETL/SETNGE = Set Byte on Less/Not Greater or Equal To Register/Memory	<table border="1"><tr><td>00001111</td><td>10011100</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	00001111	10011100	mod 0 0 0	r/m	4/5	4/5		h
00001111	10011100	mod 0 0 0	r/m						
SETNL/SETQE = Set Byte on Not Less/Greater or Equal To Register/Memory	<table border="1"><tr><td>00001111</td><td>01111101</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	00001111	01111101	mod 0 0 0	r/m	4/5	4/5		h
00001111	01111101	mod 0 0 0	r/m						
SETLE/SETNG = Set Byte on Less or Equal/Not Greater To Register/Memory	<table border="1"><tr><td>00001111</td><td>10011110</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	00001111	10011110	mod 0 0 0	r/m	4/5	4/5		h
00001111	10011110	mod 0 0 0	r/m						
SETNLE/SETG = Set Byte on Not Less or Equal/Greater To Register/Memory	<table border="1"><tr><td>00001111</td><td>10011111</td><td>mod 0 0 0</td><td>r/m</td></tr></table>	00001111	10011111	mod 0 0 0	r/m	4/5	4/5		h
00001111	10011111	mod 0 0 0	r/m						
ENTER = Enter Procedure	11001000 16-bit displacement, 8-bit level								
L = 0		10	10	b	h				
L = 1		12	12	b	h				
L > 1		15 + 4(n - 1)	15 + 4(n - 1)	b	h				
LEAVE = Leave Procedure	11001001	4	4	b	h				



80386

ADVANCE INFORMATION

Table 8-1. 80386 Instruction Set Clock Count Summary (Continued)

INSTRUCTION	FORMAT	CLOCK COUNT		NOTES			
		Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode	Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode		
INTERRUPT INSTRUCTIONS							
INT = Interrupt:							
Type Specified	<table border="1"><tr><td>11001101</td><td>type</td></tr></table>	11001101	type	37		b	
11001101	type						
Type 3	<table border="1"><tr><td>11001100</td></tr></table>	11001100	33		b		
11001100							
INTO = Interrupt 4 If Overflow Flag Set	<table border="1"><tr><td>11001110</td></tr></table>	11001110					
11001110							
If OF = 1		35		b, e			
If OF = 0		3	3	b, e			
Bound = Interrupt 5 If Detect Value Out of Range	<table border="1"><tr><td>01100010</td><td>mod reg r/m</td></tr></table>	01100010	mod reg r/m				
01100010	mod reg r/m						
If Out of Range		44		b, e	e, g, h, j, k, r		
If In Range		10	10	b, e	e, g, h, j, k, r		
Protected Mode Only (INT)							
INT: Type Specified							
Via Interrupt or Trap Gate to Same Privilege Level		59					
Via Interrupt or Trap Gate to Different Privilege Level							
From 286 Task to 286 TSS via Task Gate		99					
From 286 Task to 386 TSS via Task Gate		282					
From 286 Task to virt 8086 md via Task Gate		308					
From 386 Task to 286 TSS via Task Gate		226					
From 386 Task to 386 TSS via Task Gate		284					
From virt 8086 md to 286 TSS via Task Gate		311					
From virt 8086 md to 386 TSS via Task Gate		228					
From virt 8086 md to virt level 0 via Task Gate		289					
From virt 8086 md to 286 TSS via Task Gate		318					
From virt 8086 md to priv level 0 via Trap Gate or Interrupt Gate		119					
INT: TYPE 3							
Via Interrupt or Trap Gate to Same Privilege Level		59					
Via Interrupt or Trap Gate to Different Privilege Level							
From 286 Task to 286 TSS via Task Gate		99					
From 286 Task to 386 TSS via Task Gate		278					
From 286 Task to Virt 8086 md via Task Gate		305					
From 386 Task to 286 TSS via Task Gate		222					
From 386 Task to 386 TSS via Task Gate		280					
From virt 8086 md to 286 TSS via Task Gate		307					
From virt 8086 md to 386 TSS via Task Gate		224					
From virt 8086 md to virt level 0 via Task Gate		285					
From virt 8086 md to priv level 0 via Trap Gate or Interrupt Gate		312					
From virt 8086 md to priv level 0 via Trap Gate or Interrupt Gate		119					
INTO:							
Via Interrupt or Trap Gate to Same Privilege Level		59					
Via Interrupt or Trap Gate to Different Privilege Level							
From 286 Task to 286 TSS via Task Gate		99					
From 286 Task to 386 TSS via Task Gate		280					
From 286 Task to virt 8086 md via Task Gate		307					
From 386 Task to 286 TSS via Task Gate		224					
From 386 Task to 386 TSS via Task Gate		282					
From virt 8086 md to 286 TSS via Task Gate		309					
From virt 8086 md to 386 TSS via Task Gate		225					
From virt 8086 md to virt level 0 via Task Gate		287					
From virt 8086 md to priv level 0 via Trap Gate or Interrupt Gate		314					
From virt 8086 md to priv level 0 via Trap Gate or Interrupt Gate		119					



80386

ADVANCE INFORMATION

Table 8-1. 80386 Instruction Set Clock Count Summary (Continued)

INSTRUCTION	FORMAT	CLOCK COUNT		NOTES
		Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode	
INTERRUPT INSTRUCTIONS (Continued)				
VIA INTERRUPT OR TRAP GATE				
To Same Privilege Level		59		g, j, k, r
To Different Privilege Level		99		g, j, k, r
From 286 Task to 286 TSS via Task Gate		254		g, j, k, r
From 286 Task to 386 TSS via Task Gate		284		g, j, k, r
From 286 Task to virt 8086 Mode via Task Gate		231		g, j, k, r
From 386 Task to 286 TSS via Task Gate		264		g, j, k, r
From 386 Task to 386 TSS via Task Gate		294		g, j, k, r
From 386 Task to virt 8086 Mode via Task Gate		243		g, j, k, r
From virt 8086 Mode to 286 TSS via Task Gate		284		g, j, k, r
From virt 8086 Mode to 386 TSS via Task Gate		294		g, j, k, r
From virt 8086 md to priv level 0 via Trap Gate or Interrupt Gate		119		g, j, k, r
INTERRUPT RETURN				
IRET = Interrupt Return	11001111	22		g, h, j, k, r
Protected Mode Only (IRET)				
To the Same Privilege Level (within task)		38		g, h, j, k, r
To Different Privilege Level (within task)		82		g, h, j, k, r
From 286 Task to 286 TSS		232		h, j, k, r
From 286 Task to 386 TSS		265		h, j, k, r
From 286 Task to Virtual 8086 Task		214		h, j, k, r
From 286 Task to Virtual 8086 Mode (within task)		60		h, j, k, r
From 386 Task to 286 TSS		271		h, j, k, r
From 386 Task to 386 TSS		275		h, j, k, r
From 386 Task to Virtual 8086 Task		224		h, j, k, r
From 386 Task to Virtual 8086 Mode (within task)		80		h, j, k, r
PROCESSOR CONTROL				
HLT = HALT	11110100	5	5	1
MOV = Move to and From Control/Debug/Test Registers				
CRO/CR2/CR3 from register	00001111 00100010 11eee reg	10/4/5	10/4/5	1
Register From CRO-3	00001111 00100000 11eee reg	8	8	1
DR0-3 From Register	00001111 00100011 11eee reg	22	22	1
DR6-7 From Register	00001111 00100011 11eee reg	16	16	1
Register from DR6-7	00001111 00100001 11eee reg	14	14	1
Register from DR0-3	00001111 00100001 11eee reg	22	22	1
TR6-7 from Register	00001111 00100110 11eee reg	12	12	1
Register from TR6-7	00001111 00100100 11eee reg	12	12	1
NOP = No Operation	10010000	3	3	
WAIT = Wait until BUSY# pin is negated	10011011	8	8	



80386

ADVANCE INFORMATION

Table 8-1. 80386 Instruction Set Clock Count Summary (Continued)

INSTRUCTION	FORMAT	CLOCK COUNT		NOTES						
		Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode	Real Address Mode or Virtual 8086 Mode	Protected Virtual Address Mode					
PROCESSOR EXTENSION INSTRUCTIONS										
Processor Extension Escape	<table border="1"><tr><td>11011</td><td>TTT</td><td>mod</td><td>LLL</td><td>r/m</td></tr></table>	11011	TTT	mod	LLL	r/m	See 80287/80387 data sheets for clock counts			h
11011	TTT	mod	LLL	r/m						
PREFIX BYTES										
Address Size Prefix	<table border="1"><tr><td>01100</td><td>111</td></tr></table>	01100	111	0	0					
01100	111									
LOCK = Bus Lock Prefix	<table border="1"><tr><td>11110000</td></tr></table>	11110000	0	0		m				
11110000										
Operand Size Prefix	<table border="1"><tr><td>01100110</td></tr></table>	01100110	0	0						
01100110										
Segment Override Prefix										
CS:	<table border="1"><tr><td>00101110</td></tr></table>	00101110	0	0						
00101110										
DS:	<table border="1"><tr><td>00111110</td></tr></table>	00111110	0	0						
00111110										
ES:	<table border="1"><tr><td>00100110</td></tr></table>	00100110	0	0						
00100110										
FS:	<table border="1"><tr><td>01100100</td></tr></table>	01100100	0	0						
01100100										
GS:	<table border="1"><tr><td>01100101</td></tr></table>	01100101	0	0						
01100101										
SS:	<table border="1"><tr><td>00110110</td></tr></table>	00110110	0	0						
00110110										
PROTECTION CONTROL										
ARPL = Adjust Requested Privilege Level										
From Register/Memory	<table border="1"><tr><td>01100011</td><td>mod</td><td>reg</td><td>r/m</td></tr></table>	01100011	mod	reg	r/m	N/A	20/21	a	h	
01100011	mod	reg	r/m							
LAR = Load Access Rights										
From Register/Memory	<table border="1"><tr><td>00001111</td><td>00000010</td><td>mod</td><td>reg</td><td>r/m</td></tr></table>	00001111	00000010	mod	reg	r/m	N/A	15/18	a	g, h, j, p
00001111	00000010	mod	reg	r/m						
LGDT = Load Global Descriptor Table Register	<table border="1"><tr><td>00001111</td><td>00000001</td><td>mod</td><td>010</td><td>r/m</td></tr></table>	00001111	00000001	mod	010	r/m	11	11	b, c	h, i
00001111	00000001	mod	010	r/m						
LDIT = Load Interrupt Descriptor Table Register	<table border="1"><tr><td>00001111</td><td>00000001</td><td>mod</td><td>011</td><td>r/m</td></tr></table>	00001111	00000001	mod	011	r/m	11	11	b, c	h, i
00001111	00000001	mod	011	r/m						
LLDT = Load Local Descriptor Table Register to Register/Memory	<table border="1"><tr><td>00001111</td><td>00000000</td><td>mod</td><td>010</td><td>r/m</td></tr></table>	00001111	00000000	mod	010	r/m	N/A	20/24	a	g, h, j, l
00001111	00000000	mod	010	r/m						
LMSW = Load Machine Status Word										
From Register/Memory	<table border="1"><tr><td>00001111</td><td>00000001</td><td>mod</td><td>110</td><td>r/m</td></tr></table>	00001111	00000001	mod	110	r/m	10/13	10/13	b, c	h, i
00001111	00000001	mod	110	r/m						
LSL = Load Segment Limit										
From Register/Memory	<table border="1"><tr><td>00001111</td><td>00000011</td><td>mod</td><td>reg</td><td>r/m</td></tr></table>	00001111	00000011	mod	reg	r/m	N/A	20/21	a	g, h, j, p
00001111	00000011	mod	reg	r/m						
Byte-Granular Limit										
Page-Granular Limit										
LTR = Load Task Register										
From Register/Memory	<table border="1"><tr><td>00001111</td><td>00000000</td><td>mod</td><td>001</td><td>r/m</td></tr></table>	00001111	00000000	mod	001	r/m	N/A	23/27	a	g, h, j, l
00001111	00000000	mod	001	r/m						
SGDT = Store Global Descriptor Table Register										
Table Register	<table border="1"><tr><td>00001111</td><td>00000001</td><td>mod</td><td>000</td><td>r/m</td></tr></table>	00001111	00000001	mod	000	r/m	9	9	b, c	h
00001111	00000001	mod	000	r/m						
SIDT = Store Interrupt Descriptor Table Register										
Table Register	<table border="1"><tr><td>00001111</td><td>00000001</td><td>mod</td><td>001</td><td>r/m</td></tr></table>	00001111	00000001	mod	001	r/m	9	9	b, c	h
00001111	00000001	mod	001	r/m						
SLDT = Store Local Descriptor Table Register To Register/Memory	<table border="1"><tr><td>00001111</td><td>00000000</td><td>mod</td><td>000</td><td>r/m</td></tr></table>	00001111	00000000	mod	000	r/m	N/A	2/2	a	h
00001111	00000000	mod	000	r/m						



80386

ADVANCE INFORMATION

Table 8-1. 80386 Instruction Set Clock Count Summary (Continued)

INSTRUCTION	FORMAT	CLOCK COUNT		NOTES			
		Real Address Mode or Virtual 8088 Mode	Protected Virtual Address Mode	Real Address Mode or Virtual 8088 Mode	Protected Virtual Address Mode		
SMSW	= Store Machine Status Word	[00001111] [00000001] mod 100 r/m		10/13	10/13	b, c	h, i
STR	= Store Task Register To Register/Memory	[00001111] [00000000] mod 001 r/m		N/A	2/2	a	h
VERR	= Verify Read Accesses Register/Memory	[00001111] [00000000] mod 100 r/m		N/A	10/11	a	g, h, j, p
VERW	= Verify Write Accesses	[00001111] [00000000] mod 101 r/m		N/A	15/16	a	g, h, i, p

INSTRUCTION NOTES FOR TABLE 8-1

Notes a through c apply to 80386 Real Address Mode only:

- a. This is a Protected Mode instruction. Attempted execution in Real Mode will result in exception 6 (invalid opcode).
 b. Exception 13 fault (general protection) will occur in Real Mode if an operand reference is made that partially or fully extends beyond the maximum CS, DS, ES, FS or GS limit, FFFFH. Exception 12 fault (stack segment limit violation or not present) will occur in Real Mode if an operand reference is made that partially or fully extends beyond the maximum SS limit.
 c. This instruction may be executed in Real Mode. In Real Mode, its purpose is primarily to initialize the CPU for Protected Mode.

Notes d through g apply to 80386 Real Address Mode and 80386 Protected Virtual Address Mode:

- d. The 80386 uses an early-out multiply algorithm. The actual number of clocks depends on the position of the most significant bit in the operand (multiplier).

Clock counts given are minimum to maximum. To calculate actual clocks use the following formula:

$$\text{Actual Clock} = \begin{cases} m < > 0 & \max([log_2(m)], 3) + b \text{ clocks} \\ m = 0 & 3 + b \text{ clocks} \end{cases}$$

In this formula, m is the multiplier, and

b = 9 for register to register,

b = 12 for memory to register,

b = 10 for register with immediate to register,

b = 11 for memory with immediate to register.

- e. An exception may occur, depending on the value of the operand.

- f. LOCK# is automatically asserted, regardless of the presence or absence of the LOCK# prefix.

- g. LOCK# is asserted during descriptor table accesses.

Notes h through r apply to 80386 Protected Virtual Address Mode only:

- h. Exception 13 fault (general protection violation) will occur if the memory operand in CS, DS, ES, FS or GS cannot be used due to either a segment limit violation or access rights violation. If a stack limit is violated, an exception 12 (stack segment limit violation or not present) occurs.

- i. For segment load operations, the CPL, RPL, and DPL must agree with the privilege rules to avoid an exception 13 fault (general protection violation). The segment's descriptor must indicate "present" or exception 11 (CS, DS, ES, FS, GS not present). If the SS register is loaded and a stack segment not present is detected, an exception 12 (stack segment limit violation or not present) occurs.

- j. All segment descriptor accesses in the GDT or LDT made by this instruction will automatically assert LOCK# to maintain descriptor integrity in multiprocessor systems.

- K. JMP, CALL, INT, RET and IRET instructions referring to another code segment will cause an exception 13 (general protection violation) if an applicable privilege rule is violated.

- l. An exception 13 fault occurs if CPL is greater than 0 (0 is the most privileged level).

- m. An exception 13 fault occurs if IOPL is greater than IOPL.

- n. The IF bit of the flag register is not updated if CPL is greater than IOPL. The IOPL and VM fields of the flag register are updated only if CPL = 0.

- o. The PE bit of the MSW (CR0) cannot be reset by this instruction. Use MOV into CR0 if desiring to reset the PE bit.

- p. Any violation of privilege rules as applied to the selector operand does not cause a protection exception; rather, the zero flag is cleared.

- q. If the coprocessor's memory operand violates a segment limit or segment access rights, an exception 13 fault (general protection exception) will occur before the ESC instruction is executed. An exception 12 fault (stack segment limit violation or not present) will occur if the stack limit is violated by the operand's starting address.

- r. The destination of a JMP, CALL, INT, RET or IRET must be in the defined limit of a code segment or an exception 13 fault (general protection violation) will occur.

8.2 INSTRUCTION ENCODING

8.2.1 Overview

All instruction encodings are subsets of the general instruction format shown in Figure 8-1. Instructions consist of one or two primary opcode bytes, possibly an address specifier consisting of the "mod r/m" byte and "scaled index" byte, a displacement if required, and an immediate data field if required.

Within the primary opcode or opcodes, smaller encoding fields may be defined. These fields vary according to the class of operation. The fields define such information as direction of the operation, size of the displacements, register encoding, or sign extension.

Almost all instructions referring to an operand in memory have an addressing mode byte following the primary opcode byte(s). This byte, the mod r/m byte, specifies the address mode to be used. Certain

encodings of the mod r/m byte indicate a second addressing byte, the scale-index-base byte, follows the mod r/m byte to fully specify the addressing mode.

Addressing modes can include a displacement immediately following the mod r/m byte, or scaled index byte. If a displacement is present, the possible sizes are 8, 16 or 32 bits.

If the instruction specifies an immediate operand, the immediate operand follows any displacement bytes. The immediate operand, if specified, is always the last field of the instruction.

Figure 8-1 illustrates several of the fields that can appear in an instruction, such as the mod field and the r/m field, but the Figure does not show all fields. Several smaller fields also appear in certain instructions, sometimes within the opcode bytes themselves. Table 8-2 is a complete list of all fields appearing in the 80386 instruction set. Further ahead, following Table 8-2, are detailed tables for each field.

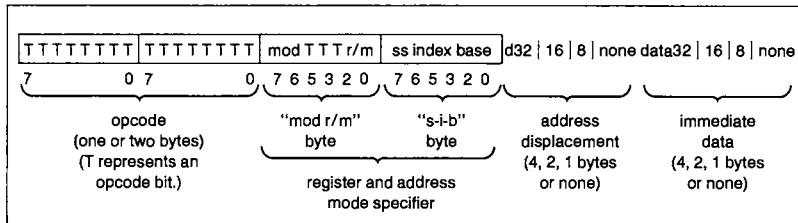


Figure 8-1. General Instruction Format

Table 8-2. Fields within 80386 Instructions

Field Name	Description	Number of Bits
w	Specifies if Data is Byte or Full Size (Full Size is either 16 or 32 Bits)	1
d	Specifies Direction of Data Operation	1
s	Specifies if an Immediate Data Field Must be Sign-Extended	1
reg	General Register Specifier	3
mod r/m	Address Mode Specifier (Effective Address can be a General Register)	2 for mod; 3 for r/m
ss	Scale Factor for Scaled Index Address Mode	2
index	General Register to be used as Index Register	3
base	General Register to be used as Base Register	3
sreg2	Segment Register Specifier for CS, SS, DS, ES	2
sreg3	Segment Register Specifier for CS, SS, DS, ES, FS, GS	3
ttn	For Conditional Instructions, Specifies a Condition Asserted or a Condition Negated	4

Note: Table 8-1 shows encoding of individual instructions.

8.2.2 32-Bit Extensions of the Instruction Set

With the 80386, the 86/186/286 instruction set is extended in two orthogonal directions: 32-bit forms of all 16-bit instructions are added to support the 32-bit data types, and 32-bit addressing modes are made available for all instructions referencing memory. This orthogonal instruction set extension is accomplished having a Default (D) bit in the code segment descriptor, and by having 2 prefixes to the instruction set.

Whether the instruction defaults to operations of 16 bits or 32 bits depends on the setting of the D bit in the code segment descriptor, which gives the default length (either 32 bits or 16 bits) for both operands and effective addresses when executing that code segment. In the Real Address Mode or Virtual 8086 Mode, no code segment descriptors are used, but a D value of 0 is assumed internally by the 80386 when operating in those modes (for 16-bit default sizes compatible with the 8086/80186/80286).

Two prefixes, the Operand Size Prefix and the Effective Address Size Prefix, allow overriding individually the Default selection of operand size and effective address size. These prefixes may precede any opcode bytes and affect only the instruction they precede. If necessary, one or both of the prefixes may be placed before the opcode bytes. The presence of the Operand Size Prefix and the Effective Address Prefix will toggle the operand size or the effective address size, respectively, to the value "opposite" from the Default setting. For example, if the default operand size is for 32-bit data operations, then presence of the Operand Size Prefix toggles the instruction to 16-bit data operation. As another example, if the default effective address size is 16 bits, presence of the Effective Address Size prefix toggles the instruction to use 32-bit effective address computations.

These 32-bit extensions are available in all 80386 modes, including the Real Address Mode or the Virtual 8086 Mode. In these modes the default is always 16 bits, so prefixes are needed to specify 32-bit operands or addresses. For instructions with more than one prefix, the order of prefixes is unimportant.

Unless specified otherwise, instructions with 8-bit and 16-bit operands do not affect the contents of the high-order bits of the extended registers.

8.2.3 Encoding of Instruction Fields

Within the instruction are several fields indicating register selection, addressing mode and so on. The exact encodings of these fields are defined immediately ahead.

8.2.3.1 ENCODING OF OPERAND LENGTH (w) FIELD

For any given instruction performing a data operation, the instruction is executing as a 32-bit operation or a 16-bit operation. Within the constraints of the operation size, the w field encodes the operand size as either one byte or the full operation size, as shown in the table below.

w Field	Operand Size During 16-Bit Data Operations	Operand Size During 32-Bit Data Operations
0	8 Bits	8 Bits
1	16 Bits	32 Bits

8.2.3.2 ENCODING OF THE GENERAL REGISTER (reg) FIELD

The general register is specified by the reg field, which may appear in the primary opcode bytes, or as the reg field of the "mod r/m" byte, or as the r/m field of the "mod r/m" byte.

Encoding of reg Field When w Field is not Present in Instruction

reg Field	Register Selected During 16-Bit Data Operations	Register Selected During 32-Bit Data Operations
000	AX	EAX
001	CX	ECX
010	DX	EDX
011	BX	EBX
100	SP	ESP
101	BP	EBP
101	SI	ESI
101	DI	EDI

Encoding of reg Field When w Field is Present in Instruction

reg	Function of w Field	
	(when w = 0)	(when w = 1)
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX
100	AH	SP
101	CH	BP
110	DH	SI
111	BH	DI

Register Specified by reg Field During 32-Bit Data Operations		
reg	Function of w Field	
	(when w = 0)	(when w = 1)
000	AL	EAX
001	CL	ECX
010	DL	EDX
011	BL	EBX
100	AH	ESP
101	CH	EBP
110	DH	ESI
111	BH	EDI

8.2.3.3 ENCODING OF THE SEGMENT REGISTER (sreg) FIELD

The sreg field in certain instructions is a 2-bit field allowing one of the four 80286 segment registers to be specified. The sreg field in other instructions is a 3-bit field, allowing the 80386 FS and GS segment registers to be specified.

2-Bit sreg2 Field

2-Bit sreg2 Field	Segment Register Selected
00	ES
01	CS
10	SS
11	DS

3-Bit sreg3 Field

3-Bit sreg3 Field	Segment Register Selected
000	ES
001	CS
010	SS
011	DS
100	FS
101	GS
110	do not use
111	do not use

8.2.3.4 ENCODING OF ADDRESS MODE

Except for special instructions, such as PUSH or POP, where the addressing mode is pre-determined, the addressing mode for the current instruction is specified by addressing bytes following the primary opcode. The primary addressing byte is the "mod r/m" byte, and a second byte of addressing information, the "s-i-b" (scale-index-base) byte, can be specified.

The s-i-b byte (scale-index-base byte) is specified when using 32-bit addressing mode and the "mod r/m" byte has $r/m = 100$ and mod = 00, 01 or 10. When the sib byte is present, the 32-bit addressing mode is a function of the mod, ss, index, and base fields.

The primary addressing byte, the "mod r/m" byte, also contains three bits (shown as TTT in Figure 8-1) sometimes used as an extension of the primary opcode. The three bits, however, may also be used as a register field (reg).

When calculating an effective address, either 16-bit addressing or 32-bit addressing is used. 16-bit addressing uses 16-bit address components to calculate the effective address while 32-bit addressing uses 32-bit address components to calculate the effective address. When 16-bit addressing is used, the "mod r/m" byte is interpreted as a 16-bit addressing mode specifier. When 32-bit addressing is used, the "mod r/m" byte is interpreted as a 32-bit addressing mode specifier.

Tables on the following three pages define all encodings of all 16-bit addressing modes and 32-bit addressing modes.



80386

ADVANCE INFORMATION

Encoding of 16-bit Address Mode with "mod r/m" Byte

mod r/m	Effective Address
00 000	DS:[BX + SI]
00 001	DS:[BX + DI]
00 010	SS:[BP + SI]
00 011	SS:[BP + DI]
00 100	DS:[SI]
00 101	DS:[DI]
00 110	DS:d16
00 111	DS:[BX]
01 000	DS:[BX + SI + d8]
01 001	DS:[BX + DI + d8]
01 010	SS:[BP + SI + d8]
01 011	SS:[BP + DI + d8]
01 100	DS:[SI + d8]
01 101	DS:[DI + d8]
01 110	SS:[BP + d8]
01 111	DS:[BX + d8]

mod r/m	Effective Address
10 000	DS:[BX + SI + d16]
10 001	DS:[BX + DI + d16]
10 010	SS:[BP + SI + d16]
10 011	SS:[BP + DI + d16]
10 100	DS:[SI + d16]
10 101	DS:[DI + d16]
10 110	SS:[BP + d16]
10 111	DS:[BX + d16]
11 000	register—see below
11 001	register—see below
11 010	register—see below
11 011	register—see below
11 100	register—see below
11 101	register—see below
11 110	register—see below
11 111	register—see below

Register Specified by r/m During 16-Bit Data Operations		
mod r/m	Function of w Field	
	(when w = 0)	(when w = 1)
11 000	AL	AX
11 001	CL	CX
11 010	DL	DX
11 011	BL	BX
11 100	AH	SP
11 101	CH	BP
11 110	DH	SI
11 111	BH	DI

Register Specified by r/m During 32-Bit Data Operations		
mod r/m	Function of w Field	
	(when w = 0)	(when w = 1)
11 000	AL	EAX
11 001	CL	ECX
11 010	DL	EDX
11 011	BL	EBX
11 100	AH	ESP
11 101	CH	EBP
11 110	DH	ESI
11 111	BH	EDI



80386

ADVANCE INFORMATION

Encoding of 32-bit Address Mode with "mod r/m" byte (no "s-i-b" byte present):

mod r/m	Effective Address
00 000	DS:[EAX]
00 001	DS:[ECX]
00 010	DS:[EDX]
00 011	DS:[EBX]
00 100	s-i-b is present
00 101	DS:d32
00 110	DS:[ESI]
00 111	DS:[EDI]
01 000	DS:[EAX + d8]
01 001	DS:[ECX + d8]
01 010	DS:[EDX + d8]
01 011	DS:[EBX + d8]
01 100	s-i-b is present
01 101	SS:[EBP + d8]
01 110	DS:[ESI + d8]
01 111	DS:[EDI + d8]

mod r/m	Effective Address
10 000	DS:[EAX + d32]
10 001	DS:[ECX + d32]
10 010	DS:[EDX + d32]
10 011	DS:[EBX + d32]
10 100	s-i-b is present
10 101	SS:[EBP + d32]
10 110	DS:[ESI + d32]
10 111	DS:[EDI + d32]
11 000	register—see below
11 001	register—see below
11 010	register—see below
11 011	register—see below
11 100	register—see below
11 101	register—see below
11 110	register—see below
11 111	register—see below

Register Specified by reg or r/m during 16-Bit Data Operations:

mod r/m	function of w field	
	(when w = 0)	(when w = 1)
11 000	AL	AX
11 001	CL	CX
11 010	DL	DX
11 011	BL	BX
11 100	AH	SP
11 101	CH	BP
11 110	DH	SI
11 111	BH	DI

Register Specified by reg or r/m during 32-Bit Data Operations:

mod r/m	function of w field	
	(when w = 0)	(when w = 1)
11 000	AL	EAX
11 001	CL	ECX
11 010	DL	EDX
11 011	BL	EBX
11 100	AH	ESP
11 101	CH	EBP
11 110	DH	ESI
11 111	BH	EDI



80386

ADVANCE INFORMATION

Encoding of 32-bit Address Mode ("mod r/m" byte and "s-i-b" byte present):

mod base	Effective Address
00 000	DS:[EAX + (scaled index)]
00 001	DS:[ECX + (scaled index)]
00 010	DS:[EDX + (scaled index)]
00 011	DS:[EBX + (scaled index)]
00 100	SS:[ESP + (scaled index)]
00 101	DS:[d32 + (scaled index)]
00 110	DS:[ESI + (scaled index)]
00 111	DS:[EDI + (scaled index)]
01 000	DS:[EAX + (scaled index) + d8]
01 001	DS:[ECX + (scaled index) + d8]
01 010	DS:[EDX + (scaled index) + d8]
01 011	DS:[EBX + (scaled index) + d8]
01 100	SS:[ESP + (scaled index) + d8]
01 101	SS:[EBP + (scaled index) + d8]
01 110	DS:[ESI + (scaled index) + d8]
01 111	DS:[EDI + (scaled index) + d8]
10 000	DS:[EAX + (scaled index) + d32]
10 001	DS:[ECX + (scaled index) + d32]
10 010	DS:[EDX + (scaled index) + d32]
10 011	DS:[EBX + (scaled index) + d32]
10 100	SS:[ESP + (scaled index) + d32]
10 101	SS:[EBP + (scaled index) + d32]
10 110	DS:[ESI + (scaled index) + d32]
10 111	DS:[EDI + (scaled index) + d32]

ss	Scale Factor
00	x1
01	x2
10	x4
11	x8

Index	Index Register
000	EAX
001	ECX
010	EDX
011	EBX
100	no index reg**
101	EBP
110	ESI
111	EDI

**IMPORTANT NOTE:

When index field is 100, indicating "no index register," then ss field MUST equal 00. If index is 100 and ss does not equal 00, the effective address is undefined.

NOTE:

Mod field in "mod r/m" byte; ss, index, base fields in "s-i-b" byte.

8.2.3.5 ENCODING OF OPERATION DIRECTION (d) FIELD

In many two-operand instructions the d field is present to indicate which operand is considered the source and which is the destination.

d	Direction of Operation
0	Register/Memory <- Register "reg" Field Indicates Source Operand; "mod r/m" or "mod ss index base" Indicates Destination Operand
1	Register <- Register/Memory "reg" Field Indicates Destination Operand; "mod r/m" or "mod ss index base" Indicates Source Operand

8.2.3.6 ENCODING OF SIGN-EXTEND (s) FIELD

The s field occurs primarily to instructions with immediate data fields. The s field has an effect only if the size of the immediate data is 8 bits and is being placed in a 16-bit or 32-bit destination.

s	Effect on Immediate Data8	Effect on Immediate Data 16 32
0	None	None
1	Sign-Extend Data8 to Fill 16-Bit or 32-Bit Destination	None

8.2.3.7 ENCODING OF CONDITIONAL TEST (tttn) FIELD

For the conditional instructions (conditional jumps and set on condition), tttn is encoded with n indicating to use the condition (n = 0) or its negation (n = 1), and tt giving the condition to test.

Mnemonic	Condition	tttn
O	Overflow	0000
NO	No Overflow	0001
B/NAE	Below/Not Above or Equal	0010
NB/AE	Not Below/Above or Equal	0011
E/Z	Equal/Zero	0100
NE/NZ	Not Equal/Not Zero	0101
BE/NA	Below or Equal/Not Above	0110
NBE/A	Not Below or Equal/Above	0111
S	Sign	1000
NS	Not Sign	1001
P/PE	Parity/Parity Even	1010
NP/PO	Not Parity/Parity Odd	1011
L/NGE	Less Than/Not Greater or Equal	1100
NL/GE	Not Less Than/Greater or Equal	1101
LE/NG	Less Than or Equal/Greater Than	1110
NLE/G	Not Less or Equal/Greater Than	1111

8.2.3.8 ENCODING OF CONTROL OR DEBUG OR TEST REGISTER (eee) FIELD

For the loading and storing of the Control, Debug and Test registers.

When Interpreted as Control Register Field

eee Code	Reg Name
000	CR0
010	CR2
011	CR3

Do not use any other encoding

When Interpreted as Debug Register Field

eee Code	Reg Name
000	DR0
001	DR1
010	DR2
011	DR3
110	DR6
111	DR7

Do not use any other encoding

When Interpreted as Test Register Field

eee Code	Reg Name
110	TR6
111	TR7

Do not use any other encoding



**National
Semiconductor**

February 1985

NS16450/INS8250A/NS16C450/INS82C50A Asynchronous Communications Element

General Description

The NS16450 is an improved specification version of the INS8250A Asynchronous Communications Element (ACE). The improved specifications ensure compatibility with the NS32016 and other state-of-the-art CPUs. Functionally, the NS16450 is equivalent to the INS8250A. The INS8250A is available in both 5V - 5% and 5V ± 10% operating ranges. See ordering instructions on the last page. The ACE is fabricated using National Semiconductor's advanced scaled N-channel silicon-gate MOS process, X莫斯.

The NS16C450 and NS82C50A are functionally equivalent to their X莫斯 counterparts, except that they are CMOS parts (The CMOS parts will be available after June 1985). It functions as a serial data input/output interface in a microcomputer system. The functional configuration of the ACE is programmed by the system software via a TRI-STATE® 8-bit bidirectional data bus; this includes the on-board baud rate generator.

The ACE performs serial-to-parallel conversion on data characters received from a peripheral device or a MODEM, and parallel-to-serial conversion on data characters received from the CPU. The CPU can read the complete status of the ACE at any time during the functional operation. Status information reported includes the type and condition of the transfer operations being performed by the ACE, as well as any error conditions (parity, overrun, framing, or break interrupt).

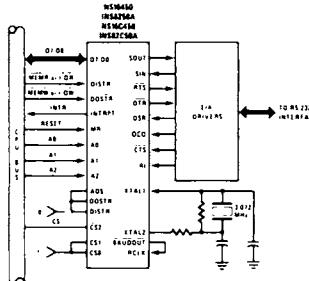
The ACE includes a programmable baud generator that is capable of dividing the timing reference clock input by divisors of 1 to $(2^{16} - 1)$, and producing a $16 \times$ clock for driving the internal transmitter logic. Provisions are also included to use this $16 \times$ clock to drive the receiver logic. Also included in the ACE is a complete MODEM-control capability, and a processor-interrupt system that may be software tailored to the user's requirements to minimize the computing required to handle the communications link.

Features

- Easily interfaces to most popular microprocessors.
- Adds or deletes standard asynchronous communication bits (start, stop, and parity) to or from serial data stream.
- Full double buffering eliminates need for precise synchronization.
- Independently controlled transmit, receive, line status, and data set interrupts.
- Programmable baud generator allows division of any input clock by 1 to $(2^{16} - 1)$ and generates the internal $16 \times$ clock.
- Independent receiver clock input.
- MODEM control functions (CTS, RTS, DSR, DTR, RI, and DCD).
- Fully programmable serial-interface characteristics:
 - 5-, 6-, 7-, or 8-bit characters
 - Even, odd, or no-parity bit generation and detection
 - 1, 1½, or 2-stop bit generation
 - Baud generation (DC to 56k baud).
- False start bit detection.
- Complete status reporting capabilities.
- TRI-STATE TTL drive capabilities for bidirectional data bus and control bus.
- Line break generation and detection.
- Internal diagnostic capabilities:
 - Loopback controls for communications link fault isolation
 - Break, parity, overrun, framing error simulation.
- Full prioritized interrupt system controls.

TRI-STATE® is a registered trademark of National Semiconductor Corp.

Basic Configuration



TLC/B401-1

Absolute Maximum Ratings

Temperature Under Bias	0°C to + 70°C
Storage Temperature	- 65°C to + 150°C
All Input or Output Voltages with Respect to V _{SS}	- 0.5V to + 7.0V
Power Dissipation	700 mW

Note: Maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended and should be limited to those conditions specified under DC electrical characteristics.

DC Electrical Characteristics

T_A = 0°C to + 70°C, V_{CC} = + 5V ± 5%, V_{SS} = 0V, unless otherwise specified.

Symbol	Parameter	Conditions	NS16450 NS16C450 (Note 1)		INS8250A, AWN INS82C50A (Note 1)		Units
			Min	Max	Min	Max	
V _{ILX}	Clock Input Low Voltage	V _{CC} = 5.25V, T _A = 25°C No Loads on output SIN, DSR, RLSD, CTS, RI = 2.0V All other inputs = 0.8V	- 0.5	0.8	- 0.5	0.8	V
V _{IHX}	Clock Input High Voltage		2.0	V _{CC}	2.0	V _{CC}	V
V _{IL}	Input Low Voltage		- 0.5	0.8	- 0.5	0.8	V
V _{IH}	Input High Voltage		2.0	V _{CC}	2.0	V _{CC}	V
V _{OL}	Output Low Voltage	I _{OL} = 1.6 mA on all *		0.4		0.4	V
V _{OH}	Output High Voltage	I _{OH} = - 1.0 mA *	2.4		2.4		V
I _{CC} (AV)	Avg. Power Supply Current (V _{CC})	V _{CC} = 5.25V, T _A = 25°C No Loads on output SIN, DSR, RLSD, CTS, RI = 2.0V All other inputs = 0.8V			120	95	mA
I _{CC} (AV)	Avg. Power Supply Current (V _{CC}) CMOS Parts Only				10	10	mA
I _{IL}	Input Leakage	V _{CC} = 5.25V, V _{SS} = 0V		± 10		± 10	µA
I _{CL}	Clock Leakage	All other pins floating. V _{IN} = 0V, 5.25V		± 10		± 10	µA
I _{OZ}	TRI-STATE Leakage	V _{CC} = 5.25V, V _{SS} = 0V V _{OUT} = 0V, 5.25V 1) Chip deselected 2) WRITE mode, chip selected		± 20		± 20	µA
V _{ILMR}	MR Schmitt V _{IL}				0.8	0.8	V
V _{IMHR}	MR Schmitt V _{IH}		2.0		2.0		V

* Does not apply to XTAL?

Capacitance T_A = 25°C, V_{CC} = V_{SS} = 0V

Symbol	Parameter	Conditions	Min	Typ	Max	Units
C _{XTAL2}	Clock Input Capacitance	f _c = 1 MHz		15	20	pF
C _{XTAL1}	Clock Output Capacitance			20	30	pF
C _{IN}	Input Capacitance			6	10	pF
C _{OUT}	Output Capacitance			10	20	pF

Note 1: All specifications for CMOS parts are preliminary

AC Electrical Characteristics $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $V_{CC} = +5V \pm 5\%$

Symbol	Parameter	Conditions	NS16450		INS8250A, AWN		Units
			Min	Max	Min	Max	
t_{AW}	Address Strobe Width		60		90		ns
t_{AS}	Address Setup Time		60		90		ns
t_{AH}	Address Hold Time		0		0		ns
t_{CS}	Chip Select Setup Time		60		90		ns
t_{CH}	Chip Select Hold Time		0		0		ns
t_{DW}	DISTR/DISTR Strobe Width		125		175		ns
t_{RC}	Ready Cycle Delay		175		500		ns
t_{RC}	Ready Cycle $t_{AR} + t_{DW} + t_{RC}$		360		755		ns
t_{DD}	DISTR/DISTR to Driver Disable Delay	@100 pF loading***	60		75		ns
t_{DDD}	Delay from DISTR/DISTR to Data	@100 pF loading	125		175		ns
t_{HZ}	DISTR/DISTR to Floating Data Delay	@100 pF loading***	0	100	100		ns
t_{DOW}	DOSTR/DOSTR Strobe Width		100		175		ns
t_{WC}	Write Cycle Delay		200		500		ns
WC	Write Cycle $t_{AW} + t_{DOW} + t_{WC}$		360		755		ns
t_{DS}	Data Setup Time		40		90		ns
t_{DH}	Data Hold Time		40		60		ns
t_{CSC^*}	Chip Select Output Delay from Select	@100 pF loading		100		125	ns
t_{RA^*}	Address Hold Time from DISTR/DISTR		20		20		ns
t_{RCS^*}	Chip Select Hold Time from DISTR/DISTR		20		20		ns
t_{AR^*}	DISTR/DISTR Delay from Address		60		80		ns
t_{CSR^*}	DISTR/DISTR Delay from Chip Select		50		80		ns
t_{WA^*}	Address Hold Time from DOSTR/DOSTR		20		20		ns
t_{WCS^*}	Chip Select Hold Time from DOSTR/DOSTR		20		20		ns
t_{AW^*}	DOSTR/DOSTR Delay from Address		60		80		ns
t_{CSW^*}	DOSTR/DOSTR Delay from Select		50		80		ns
t_{MRW}	Master Reset Pulse Width		5		10		μs
t_{XH}	Duration of Clock High Pulse	External Clock (3.1 MHz Max.)	140		140		ns
t_{XL}	Duration of Clock Low Pulse	External Clock (3.1 MHz Max.)	140		140		ns
Baud Generator							
N	Baud Divisor		1	$2^{16}-1$	1	$2^{16}-1$	
t_{BLD}	Baud Output Negative Edge Delay	100 pF Load		125		250	ns
t_{BHD}	Baud Output Positive Edge Delay	100 pF Load		125		250	ns
t_{LW}	Baud Output Down Time	$f_X = 2 \text{ MHz}, -2, 100 \text{ pF Load}$	425		425		ns
t_{HW}	Baud Output Up Time	$f_X = 3 \text{ MHz}, -3, 100 \text{ pF Load}$	330		330		ns
Receiver							
t_{SCO}	Delay from RCLK to Sample Time		2		2		μs
t_{SINT}	Delay from Stop to Set Interrupt		1	1	1	1	RCLK** Cycles
t_{RINT}	Delay from DISTR/DISTR (RD RBR/RDLSR) to Reset Interrupt	100 pF Load		1		1	μs

*Applicable only when ADS is held low

**RCLK is equal to t_{AR} and t_{RC}

***Charge and discharge time is determined by V_{OL} , V_{OH} , and the external loading

Note 1: All specifications for CMOS parts are preliminary

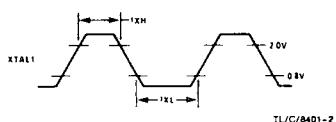
AC Electrical Characteristics (Continued)

Symbol	Parameter	Conditions	NS16450		INS8250A, AWN INS82C50A (Note 1)		Units
			Min	Max	Min	Max	
Transmitter							
t _{HR}	Delay from DOSTR/DOSTR (WR THR) to Reset Interrupt	100 pF Load		175		1000	ns
t _{PS}	Delay from Initial INTR Reset to Transmit Start		8	24	8	24	RCLK Cycles
t _{SI}	Delay from Initial Write to Interrupt		16	32	16	32	RCLK Cycles
t _{STI}	Delay from Stop to Interrupt (THRE)		8	8	8	8	RCLK Cycles
t _{IR}	Delay from DISTR/DISTR (RD IIR) to Reset Interrupt (THRE)	100 pF Load		250		1000	ns
Modem Control							
t _{MDO}	Delay from DOSTR/DOSTR (WR MCR) to Output	100 pF Load		200		1000	ns
t _{SIM}	Delay to Set Interrupt from MODEM Input	100 pF Load				1000	ns
t _{RIM}	Delay to Reset Interrupt from DISTR/DISTR (RD MSR)	100 pF Load		250		1000	ns

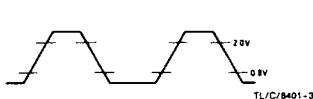
Note 1: All specifications for CMOS parts are preliminary.

Timing Waveforms (All timings are referenced to valid 0 and valid 1)

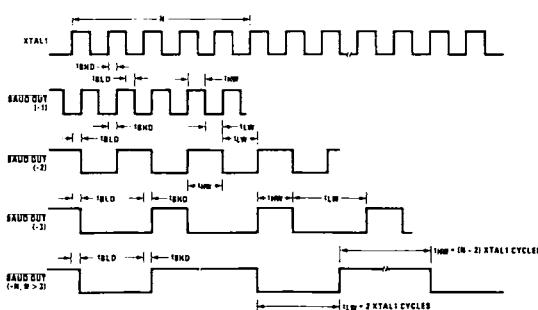
External Clock Input (3.1 MHz Max.)



AC Test Points



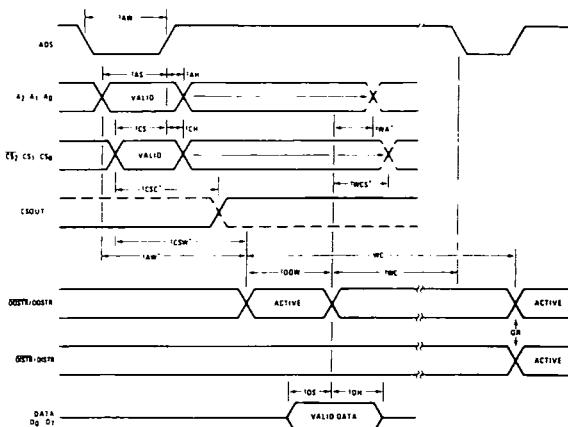
BAUDOUT Timing



TL/C/8401-4

Timing Waveforms (Continued)

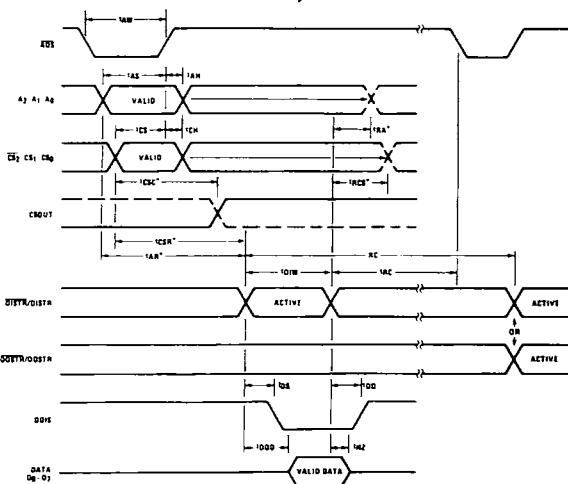
Write Cycle



*Applicable Only When ADS is Tied Low

TL/C/8401-5

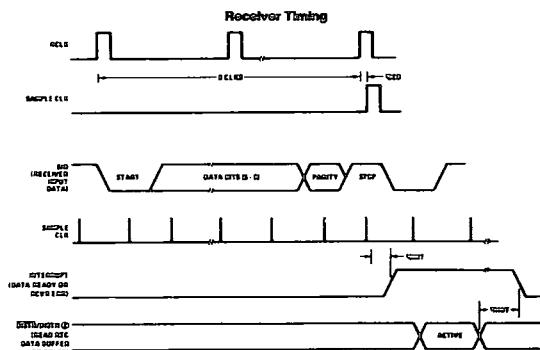
Read Cycle



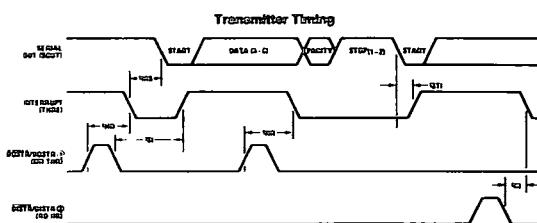
*Applicable Only When ADS is Tied Low.

TL/C/8401-6

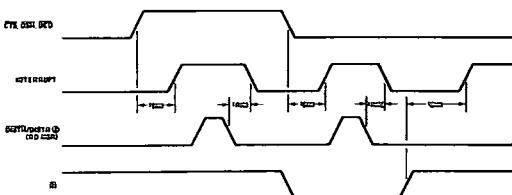
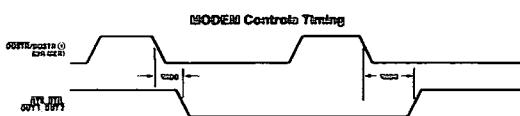
Timing Waveforms (Continued)



TL/C/0401-7



TL/C/0401-8

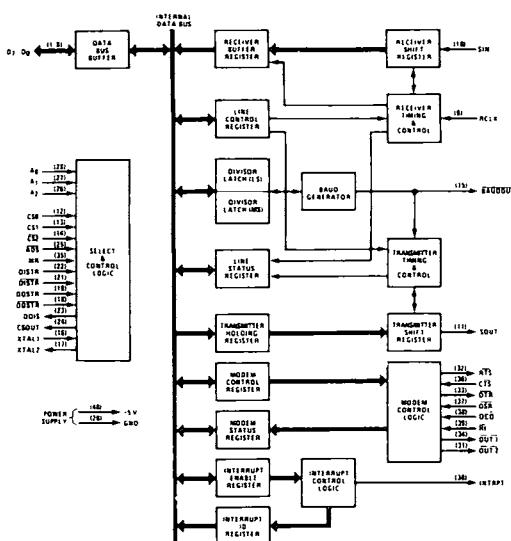


TL/C/0401-9

Note 1: See Write Cycle Timing

Note 2: See Read Cycle Timing

Block Diagram



TL/C/8401-10

Note: Applicable pinout numbers are included within parenthesis.

Functional Pin Description

The following describes the function of all NS16450, NS16C450 and INS8250A, INS82C50A input and output pins. Some of these descriptions reference internal circuits.

Note: In the following descriptions, a low represents a logic 0 (0V nominal) and a high represents a logic 1 (+2.4V nominal).

INPUT SIGNALS

Chip Select (CS0, CS1, CS2), Pins 12–14: When CS0 and CS1 are high and CS2 is low, the chip is selected. Chip selection is complete when the decoded chip select signal is latched with an active (low) Address Strobe (ADS) input. This enables communication between the ACE and the CPU.

Data Input Strobe (DISTR, DISTRI), Pins 22 and 21: When DISTR is high or DISTRI is low while the chip is selected, it allows the CPU to read status information or data from a selected register of the ACE.

Note: Only an active DISTR or DISTRI input is required to transfer data from the ACE during a read operation. Therefore, be either the DISTR input permanently low or the DISTRI input permanently high, if not used.

Data Output Strobe (DOSTR, DOSTR), Pins 19 and 18: When DOSTR is high or DOSTR is low while the chip is selected, allows the CPU to write data or control words into a selected register of the ACE.

Note: Only an active DOSTR or DOSTR input is required to transfer data to the ACE during a write operation. Therefore, be either the DOSTR input permanently low or the DOSTR input permanently high, if not used.

Address Strobe (ADS), Pin 25: When low, provides latching for the Register Select (A0, A1, A2) and Chip Select (CS0, CS1, CS2) signals.

Note: An active ADS input is required when the Register Select (A0, A1, A2) signals are not stable for the duration of a read or write operation. If not required, tie the ADS input permanently low.

Register Select (A0, A1, A2), Pins 26–28: These three inputs are used during a read or write operation to select an ACE register to read from or write into as indicated in the table below. Note that the state of the Divisor Latch Access Bit (DLAB), which is the most significant bit of the Line Control Register, affects the selection of certain INS8250A registers. The DLAB must be set high by the system software to access the Baud Generator Divisor Latches.

Functional Pin Description (Continued)

DLAB	A ₂	A ₁	A ₀	Register
0	0	0	0	Receiver Buffer (read), Transmitter Holding Register (write)
0	0	0	1	Interrupt Enable
X	0	1	0	Interrupt Identification (read only)
X	0	1	1	Line Control
X	1	0	0	MODEM Control
X	1	0	1	Line Status
X	1	1	0	MODEM Status
X	1	1	1	Scratch
1	0	0	0	Divisor Latch (least significant byte)
1	0	0	1	Divisor Latch (most significant byte)

Master Reset (MR), Pin 35: This input is buffered with a TTL-compatible Schmitt Trigger with 0.5V typical hysteresis. When high, it clears all the registers (except the Receiver Buffer, Transmitter Holding, and Divisor Latches), and the control logic of the ACE. Also, the state of various output signals (SOUT, INTAPT, OUT 1, OUT 2, RTS, DTR) are affected by an active MR input. (Refer to Table I.)

Receiver Clock (RCLK), Pin 9: This input is the 16 × baud rate clock for the receiver section of the chip.

Serial Input (SIN), Pin 10: Serial data input from the communications link (peripheral device, MODEM, or data set).

Clear to Send (CTS), Pin 36: The CTS signal is a MODEM control function input whose conditions can be tested by the CPU by reading bit 4 (CTS) of the MODEM Status Register. Bit 0 (DCTS) of the MODEM Status Register indicates whether the CTS input has changed state since the previous reading of the MODEM Status Register. CTS has no effect on the Transmitter.

Note: Whenever the CTS bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Data Set Ready (DSR), Pin 37: When low, this indicates that the MODEM or data set is ready to establish the communications link and transfer data with the ACE. The DSR signal is a MODEM-control function input whose condition can be tested by the CPU by reading bit 5 (DSR) of the MODEM Status Register. Bit 1 (DDSR) of the MODEM Status Register indicates whether the DSR input has changed state since the previous reading of the MODEM Status Register.

Note: Whenever the DSR bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Data Carrier Detect (DCD), Pin 38: When low, indicates that the data carrier has been detected by the MODEM or data set. The DCD signal is a MODEM-control function input whose condition can be tested by the CPU by reading bit 7 (DCD) of the MODEM Status Register. Bit 3 (DDCD) of the MODEM Status Register indicates whether the DCD input has changed state since the previous reading of the MODEM Status Register. DCD has no effect on the receiver.

Note: Whenever the DCD bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Ring Indicator (RI), Pin 39: When low, indicates that a telephone ringing signal has been received by the MODEM or data set. The RI signal is a MODEM-control function input whose condition can be tested by the CPU by reading bit 6 (RI) of the MODEM Status Register. Bit 2 (TERI) of the MODEM Status Register indicates whether the RI input has changed from a low to a high state since the previous reading of the MODEM Status Register.

Note: Whenever the RI bit of the MODEM Status Register changes from a high to a low state, an interrupt is generated if the MODEM Status Register is enabled.

V_{CC}, Pin 40: +5V supply.

V_{GND}, Pin 20: Ground (0V) reference.

OUTPUT SIGNALS

Data Terminal Ready (DTR), Pin 33: When low, informs the MODEM or data set that the ACE is ready to communicate. The DTR output signal can be set to an active low by programming bit 0 (DTR) of the MODEM Control Register to a high level. The DTR signal is set high upon a Master Reset operation. The DTR signal is forced to its inactive state (high) during loop mode operation.

Request to Send (RTS), Pin 32: When low, informs the MODEM or data set that the ACE is ready to transmit data. The RTS output signal can be set to an active low by programming bit 1 (RTS) of the MODEM Control Register. The RTS signal is set high upon a Master Reset operation. The RTS signal is forced to its inactive state (high) during loop mode operation.

Output 1 (OUT 1), Pin 34: User-designated output that can be set to an active low by programming bit 2 (OUT 1) of the MODEM Control Register to a high level. The OUT 1 signal is set high upon a Master Reset Operation. The OUT 1 signal is forced to its inactive state (high) during loop mode operation.

Output 2 (OUT 2), Pin 31: User-designated output that can be set to an active low by programming bit 3 (OUT 2) of the MODEM Control Register to a high level. The OUT 2 signal is set high upon a Master Reset Operation. The OUT 2 signal is forced to its inactive state (high) during loop mode operation.

Chip Select Out (CSOUT), Pin 24: When high, indicates that the chip has been selected by active CS0, CS1, and CS2 inputs. No data transfer can be initiated until the CSOUT signal is a logic 1. CSOUT goes low when chip is deselected.

Driver Disable (DDIS), Pin 23: Goes low whenever the CPU is reading data from the ACE. A high-level DDIS output can be used to disable an external transceiver (if used between the CPU and ACE on the D₇-D₀ Data Bus) at all times, except when the CPU is reading data.

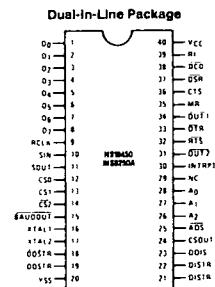
Baud Out (BAUDOUT), Pin 15: 16 × clock signal for the transmitter section of the ACE. The clock rate is equal to the main reference oscillator frequency divided by the specified divisor in the Baud Generator Divisor Latches. The BAUDOUT may also be used for the receiver section by tying this output to the RCLK input of the chip.

Functional Pin Description (Continued)

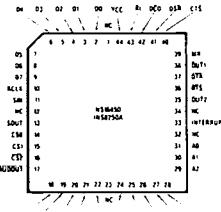
Interrupt (INTRPT), Pin 30: Goes high whenever any one of the following interrupt types has an active high condition and is enabled via the IER. Receiver Error Flag, Received Data Available, Transmitter Holding Register Empty, and MODEM Status. The INTRPT signal is reset low upon the appropriate interrupt service or a Master Reset operation.

Serial Output (SOUT), Pin 11: Composite serial data output to the communications link (peripheral, MODEM or data set). The SOUT signal is set to the Marking (logic 1) state upon a Master Reset operation.

Connection Diagrams



TL/C/8401-11



TL/C/8401-18

TABLE I. ACE Reset Functions

Register/Signal	Reset Control	Reset State
Interrupt Enable Register	Master Reset	All Bits Low (0-3 forced and 4-7 permanent)
Interrupt Identification Register	Master Reset	Bit 0 is High, Bits 1 and 2 Low Bits 3-7 are Permanently Low
Line Control Register	Master Reset	All Bits Low
MODEM Control Register	Master Reset	All Bits Low
Line Status Register	Master Reset	All Bits Low, Except Bits 5 and 6 are High
MODEM Status Register	Master Reset	Bits 0-3 Low Bits 4-7—Input Signal
SOUT	Master Reset	High
INTRPT (RCVR Errs)	Read LSR/MR	Low
INTRPT (RCVR Data Ready)	Read RBR/MR	Low
INTRPT (THRE)	Read IIR/Write THR/MR	Low
INTRPT (Modem Status Changes)	Read MSR/MR	Low
DOUT 2	Master Reset	High
RTS	Master Reset	High
DTR	Master Reset	High
DOUT 1	Master Reset	High

Comm1 = 3F8h Base

Accessible Registers

The system programmer may access or control any of the ACE registers summarized in Table II via the CPU. These registers are used to control ACE operations and to transmit and receive data.

LINE CONTROL REGISTER

The system programmer specifies the format of the asynchronous data communications exchange via the Line Control Register. In addition to controlling the format, the programmer may retrieve the contents of the Line Control Register for inspection. This feature simplifies system programming and eliminates the need for separate storage in system memory of the line characteristics. The contents of the Line Control Register are indicated in Table II and are described below.

Table II. Summary of Accessible Registers

Bit No.	Register Address											
	0 DLAB = 0	0 DLAB = 0	1 DLAB = 0	2	3	4	5	6	7	0 DLAB = 1	1 DLAB = 1	
	Receiver Buffer Register (Read Only)	Transmitter Holding Register (Write Only)	Interrupt Enable Register	Interrupt Ident. Register (Read Only)	Line Control Register	MODEM Control Register	Line Status Register	MODEM Status Register	Scratch Register	Divisor Latch (LS)	Latch (MS)	
	RBR	THR	IER	IIR	LCR	MCR	LSR	MSR	SCR	DLL	DLM	
0	Data Bit 0*	Data Bit 0	Enable Received Data Available Interrupt (ERBI)	"0" if Interrupt Pending	Word Length Select Bit 0 (WLS0)	Data Terminal Ready (DTR)	Data Ready (DR)	Delta Clear to Send (DCTS)	Bit 0	Bit 0	Bit 8	
1	Data Bit 1	Data Bit 1	Enable Transmitter Holding Register Empty interrupt (ETBEI)	Interrupt ID Bit (0)	Word Length Select Bit 1 (WLS1)	Request to Send (RTS)	Overrun Error (OE)	Delta Data Set Ready (DDSR)	Bit 1	Bit 1	Bit 9	
2	Data Bit 2	Data Bit 2	Enable Receiver Line Status Interrupt (ELSI)	Interrupt ID Bit (1)	Number of Stop Bits (STB)	Out 1	Parity Error (PE)	Training Edge Ring Indicator (TERI)	Bit 2	Bit 2	Bit 10	
3	Data Bit 3	Data Bit 3	Enable MODEM Status Interrupt (EDSSI)	0	Parity Enable (PEN)	Out 2	Framing Error (FE)	Delta Data Carrier Detect (DDCD)	Bit 3	Bit 3	Bit 11	
4	Data Bit 4	Data Bit 4	0	0	Even Parity Select (EPS)	Loop	Break Interrupt (BI)	Clear to Send (CTS)	Bit 4	Bit 4	Bit 12	
5	Data Bit 5	Data Bit 5	0	0	Stick Parity	0	Transmitter Holding Register (THRE)	Data Set Ready (DSR)	Bit 5	Bit 5	Bit 13	
6	Data Bit 6	Data Bit 6	0	0	Set Break	0	Transmitter Empty (TEM)	Ring Indicator (RI)	Bit 6	Bit 6	Bit 14	
7	Data Bit 7	Data Bit 7	0	0	Divisor Latch Access Bit (DLAB)	0	0	Data Carrier Detect (DCD)	Bit 7	Bit 7	Bit 15	

*Bit 0 is the least significant bit. It is the first bit serially transmitted or received.

Accessible Registers (Continued)

half Stop bits are generated. If bit 2 is a logic 1 when either a 6-, 7-, or 8-bit word length is selected, two Stop bits are generated. The Receiver checks the first Stop-bit only, regardless of the number of Stop bits selected.

Bit 3: This bit is the Parity Enable bit. When bit 3 is a logic 1, a Parity bit is generated (transmit data) or checked (receive data) between the last data word bit and Stop bit of the serial data. (The Parity bit is used to produce an even or odd number of 1s when the data word bits and the Parity bit are summed.)

Bit 4: This bit is the Even Parity Select bit. When bit 3 is a logic 1 and bit 4 is a logic 0, an odd number of logic 1s is transmitted or checked in the data word bits and Parity bit. When bit 3 is a logic 1 and bit 4 is a logic 1, an even number of logic 1s is transmitted or checked.

Bit 5: This bit is the Stick Parity bit. When bits 3, 4 and 5 are logic 1 the Parity bit is transmitted and checked by the receiver as a logic 0. If bits 3 and 5 are 1 and bit 4 is a logic 0 then the Parity bit is transmitted as a 0.

Bit 6: This bit is the Break Control bit. When it is set to a logic 1, the serial output (SOUT) is forced to the Spacing (logic 0) state. The break is disabled by setting bit 6 to a logic 0. The Break Control bit acts only on SOUT and has no effect on the transmitter logic.

Note: This feature enables the CPU to alert a terminal in a computer communications system if the following sequence is followed; no erroneous or extraneous characters will be transmitted because of the break.

2. Load an all 0s, pad character, in response to THRE
2. Set break after the next THRE
3. Wait for the transmitter to be idle, ($\text{TEM7} = 1$), and clear break when normal transmission has to be restored

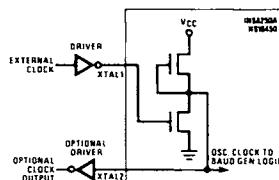
During the break, the Transmitter can be used as a character timer to accurately establish the break duration.

Bit 7: This bit is the Divisor Latch Access Bit (DLAB). It must be set high (logic 1) to access the Divisor Latches of the Baud Generator during a Read or Write operation. It must be set low (logic 0) to access the Receiver Buffer, the Transmitter Holding Register, or the Interrupt Enable Register.

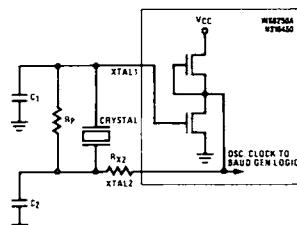
Table III. Baud Rates Using 1.8432 MHz Crystal

Desired Baud Rate	Divisor Used to Generate 16 x Clock	Percent Error Difference Between Desired and Actual
50	2304	—
75	1536	—
110	1047	0.026
134.5	857	0.058
150	768	—
300	384	—
600	192	—
1200	96	—
1800	64	—
2000	58	0.69
2400	48	—
3600	32	—
4800	24	—
7200	16	—
9600	12	—
19200	6	—
38400	3	—
56000	2	2.86

Typical Clock Circuits



TL/C/8401-12



TL/C/8401-13

Typical Crystal Oscillator Network

CRYSTAL	R _P	R _{X2}	C ₁	C ₂
3.1 MHz	1 MΩ	1.5k	10-30 pF	40-60 pF
1.8 MHz	1 MΩ	1.5k	10-30 pF	40-60 pF

Table IV. Baud Rates Using 3.072 MHz Crystal

Desired Baud Rate	Divisor Used to Generate 16 x Clock	Percent Error Difference Between Desired and Actual
50	3840	—
75	2560	—
110	1745	0.026
134.5	1428	0.034
150	1280	—
300	640	—
600	320	—
1200	160	—
1800	107	0.312
2000	96	—
2400	80	—
3600	53	0.628
4800	40	—
7200	27	1.23
9600	20	—
19200	10	—
38400	5	—

Accessible Registers (Continued)

PROGRAMMABLE BAUD GENERATOR

The ACE contains a programmable Baud Generator that is capable of taking any clock input (DC to 3.1 MHz) and dividing it by any divisor from 1 to $2^{16} - 1$. The output frequency of the Baud Generator is $16 \times$ the Baud [divisor # = (frequency input) \div (baud rate \times 16)]. Two 8-bit latches store the divisor in a 16-bit binary format. These Divisor Latches must be loaded during initialization in order to ensure desired operation of the Baud Generator. Upon loading either of the Divisor Latches, a 16-bit Baud counter is immediately loaded. This prevents long counts on initial load.

Tables III and IV illustrate the use of the Baud Generator with crystal frequencies of 1.8432 MHz and 3.072 MHz respectively. For baud rates of 38400 and below, the error obtained is minimal. The accuracy of the desired baud rate is dependent on the crystal frequency chosen.

Note: The maximum operating frequency of the Baud Generator is 3.1 MHz. However, when using divisors of 3 and below, the maximum frequency is equal to the divisor in MHz. For example, if the divisor is 1, then the maximum frequency is 1 MHz. In no case should the date rate be greater than 56k baud.

LINE STATUS REGISTER

This 8-bit register provides status information to the CPU concerning the data transfer. The contents of the Line Status Register are indicated in Table 2 and are described below.

Bit 0: This bit is the receiver Data Ready (DR) indicator. Bit 0 is set to a logic 1 whenever a complete incoming character has been received and transferred into the Receiver Buffer Register. Bit 0 is reset to a logic 0 by reading the data in the Receiver Buffer Register.

Bit 1: This bit is the Overrun Error (OE) indicator. Bit 1 indicates that data in the Receiver Buffer Register was not read by the CPU before the next character was transferred into the Receiver Buffer Register, thereby destroying the previous character. The OE indicator is reset whenever the CPU reads the contents of the Line Status Register.

Bit 2: This bit is the Parity Error (PE) indicator. Bit 2 indicates that the received data character does not have the correct even or odd parity, as selected by the even-parity-select bit. The PE bit is set to a logic 1 upon detection of a parity error and is reset to a logic 0 whenever the CPU reads the contents of the Line Status Register.

Bit 3: This bit is the Framing Error (FE) indicator. Bit 3 indicates that the received character did not have a valid Stop bit. Bit 3 is set to a logic 1 whenever the Stop bit following the last data bit or parity bit is detected as a zero bit (Spacing level). The FE indicator is reset whenever the CPU reads the contents of the Line Status indicator.

Bit 4: This bit is the Break Interrupt (BI) indicator. Bit 4 is set to a logic 1 whenever the received data input is held in the Spacing (logic 0) state for longer than a full word transmission time (that is, the total time of Start bit + data bits + Parity + Stop bits). The BI indicator is reset whenever the CPU reads the contents of the Line Status indicator.

Note: Bits 1 through 4 are the error conditions that produce a Receiver Line Status interrupt whenever any of the corresponding conditions are detected.

Bit 5: The bit is the Transmitter Holding Register Empty (THRE) indicator. Bit 5 indicates that the ACE is ready to accept a new character for transmission. In addition, this bit causes the ACE to issue an interrupt to the CPU when the Transmitter Holding Register Empty interrupt enable is set high. The THRE bit is set to a logic 1 when a character is transferred from the Transmitter Holding Register into the Transmitter Shift Register. The bit is reset to logic 0 concurrently with the loading of the Transmitter Holding Register by the CPU.

Bit 6: This bit is the Transmitter Empty (TEM) indicator. Bit 6 is set to a logic 1 whenever the Transmitter Holding Register (THR) and the Transmitter Shift Register (TSR) are both empty. It is reset to a logic 0 whenever either the THR or TSR contains a data character.

Bit 7: This bit is permanently set to logic 0.

Note: The Line Status Register is intended for read operations only. Writing to this register is not recommended as this operation is used for factory testing.

TABLE V. Interrupt Control Functions

Interrupt Identification Register			Interrupt Set and Reset Functions			
Bit 2	Bit 1	Bit 0	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	0	1	—	None	None	—
1	1	0	Highest	Receiver Line Status	Overrun Error or Parity Error or Framing Error or Break Interrupt	Reading the Line Status Register
1	0	0	Second	Received Data Available	Receiver Data Available	Reading the Receiver Buffer Register
0	1	0	Third	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR Register (if source of interrupt) or Writing into the Transmitter Holding Register
0	0	0	Fourth	MODEM Status	Clear to Send or Data Set Ready or Ring Indicator or Data Carrier Detect	Reading the MODEM Status Register

Accessible Registers (Continued)

INTERRUPT IDENTIFICATION REGISTER

The ACE has an on-chip interrupt capability that allows for flexibility in interfacing popular microprocessors presently available. In order to provide minimum software overhead during data character transfers, the ACE prioritizes interrupts into four levels. The four levels of interrupt conditions are as follows: Receiver Line Status (priority 1); Received Data Ready (priority 2); Transmitter Holding Register Empty (priority 3), and MODEM Status (priority 4).

Information indicating that a prioritized interrupt is pending and the type of that interrupt are stored in the Interrupt Identification Register (IIR). When addressed during chip-select time, the IIR freezes the highest priority interrupt pending and no other interrupts are acknowledged until the particular interrupt is serviced by the CPU. The contents of the IIR are indicated in Table II and are described below.

Bit 0: This bit can be used in either a hardwired prioritized or polled environment to indicate whether an interrupt is pending. When bit 0 is a logic 0, an interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt service routine. When bit 0 is a logic 1, no interrupt is pending and polling (if used) continues.

Bits 1 and 2: These two bits of the IIR are used to identify the highest priority interrupt pending as indicated in Table V.

Bits 3 through 7: These five bits of the IIR are always logic 0.

INTERRUPT ENABLE REGISTER

This 8-bit register enables the four types of interrupts of the ACE to separately activate the chip interrupt (INTRPT) output signal. It is possible to totally disable the interrupt system by resetting bits 0 through 3 of the Interrupt Enable Register. Similarly, by setting the appropriate bits of this register to a logic 1, selected interrupts can be enabled. Disabling the interrupt system inhibits the Interrupt Identification Register and the active (high) INTRPT output from the chip. All other system functions operate in their normal manner, including the setting of the Line Status and MODEM Status Registers. The contents of the Interrupt Enable Register are indicated in Table II and are described below.

Bit 0: This bit enables the Received Data Available Interrupt when set to logic 1.

Bit 1: This bit enables the Transmitter Holding Register Empty Interrupt when set to logic 1.

Bit 2: This bit enables the Receiver Line Status Interrupt when set to logic 1.

Bit 3: This bit enables the MODEM Status Interrupt when set to logic 1.

Bits 4 through 7: These four bits are always logic 0.

MODEM CONTROL REGISTER

This 8-bit register controls the interface with the MODEM or data set (or a peripheral device emulating a MODEM). The contents of the MODEM Control Register are indicated in Table II and are described below.

Bit 0: This bit controls the Data Terminal Ready (DTR) output. When bit 0 is set to a logic 1, the DTR output is forced to a logic 0. When bit 0 is reset to a logic 0, the DTR output is forced to a logic 1.

Note: The DTR output of the ACE may be applied to an EIA inverting line driver (such as the DS1488) to obtain the proper polarity input at the succeeding MODEM or data set.

Bit 1: This bit controls the Request to Send (RTS) output. Bit 1 affects the RTS output in a manner identical to that described above for bit 0.

Bit 2: This bit controls the Output 1 (OUT 1) signal, which is an auxiliary user-designated output. Bit 2 affects the OUT 1 output in a manner identical to that described above for bit 0.

Bit 3: This bit controls the Output 2 (OUT 2) signal, which is an auxiliary user-designated output. Bit 3 affects the OUT 2 output in a manner identical to that described above for bit 0.

Bit 4: This bit provides a local loopback feature for diagnostic testing of the ACE. When bit 4 is set to logic 1, the following occur: the transmitter Serial Output (SOUT 1) is set to the Marking (logic 1) state; the receiver Serial Input (SIN) is disconnected; the output of the Transmitter Shift Register is "looped back" into the Receiver Shift Register input; the four MODEM Control inputs (CTS, DSR, DCD, and RI) are disconnected; and the four MODEM Control outputs (DTR, RTS, OUT 1, and OUT 2) are internally connected to the four MODEM Control inputs, and the MODEM Control output pins are forced to their inactive state (high). In the diagnostic mode, data that is transmitted is immediately received. This feature allows the processor to verify the transmit-and-receive-data paths of the ACE.

In the diagnostic mode, the receiver and transmitter interrupts are fully operational. The MODEM Control Interrupts are also operational, but the interrupts' sources are now the lower four bits of the MODEM Control Register instead of the four MODEM Control inputs. The interrupts are still controlled by the Interrupt Enable Register.

Bits 5 through 7: These bits are permanently set to logic 0.

MODEM STATUS REGISTER

This 8-bit register provides the current state of the control lines from the MODEM (or peripheral device) to the CPU. In addition to this current-state information, four bits of the MODEM Status Register provide change information. These bits are set to a logic 1 whenever a control input from the MODEM changes state. They are reset to logic 0 whenever the CPU reads the MODEM Status Register.

Accessible Registers (Continued)

The contents of the MODEM Status Register are indicated in Table II and are described below.

BIT 0: This bit is the Delta Clear to Send (DCTS) indicator. Bit 0 indicates that the CTS input to the chip has changed state since the last time it was read by the CPU.

BIT 1: This bit is the Delta Data Set Ready (DDSR) indicator. Bit 1 indicates that the DSR input to the chip has changed state since the last time it was read by the CPU.

BIT 2: This bit is the Trailing Edge of Ring Indicator (TERI) detector. Bit 2 indicates that the RI input to the chip has changed from a low to a high state.

BIT 3: This bit is the Delta Data Carrier Detect (DDCD) indicator. Bit 3 indicates that the DCD input to the chip has changed state.

Note: Whenever bit 0, 1, 2, or 3 is set to logic 1, a MODEM Status Interrupt is generated.

BIT 4: This bit is the complement of the Clear to Send (CTS) input. If bit 4 (loop) of the MCR is set to a 1, this bit is equivalent to RTS in the MCR.

BIT 5: This bit is the complement of the Data Set Ready (DSR) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to DTR in the MCR.

BIT 6: This bit is the complement of the Ring Indicator (RI) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to OUT 1 in the MCR.

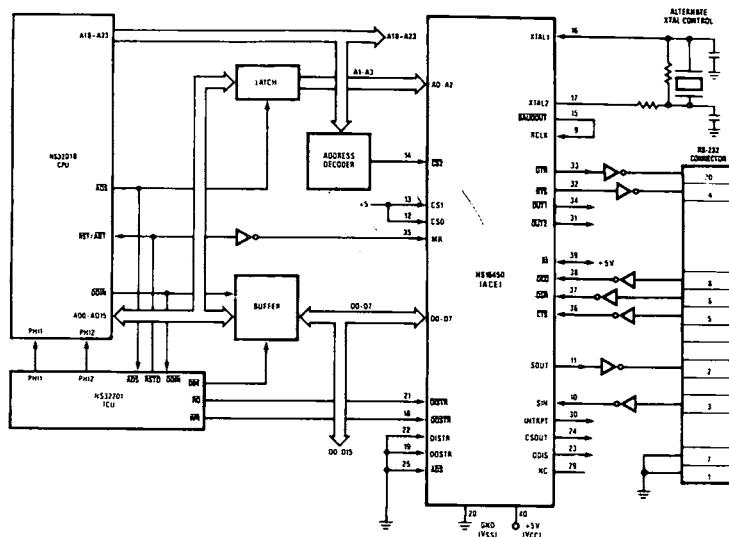
BIT 7: This bit is the complement of the Data Carrier Detect (DCD) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to OUT 2 of the MCR.

SCRATCHPAD REGISTER

This 8-bit Read/Write Register does not control the ACE in any way. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.

Typical Applications

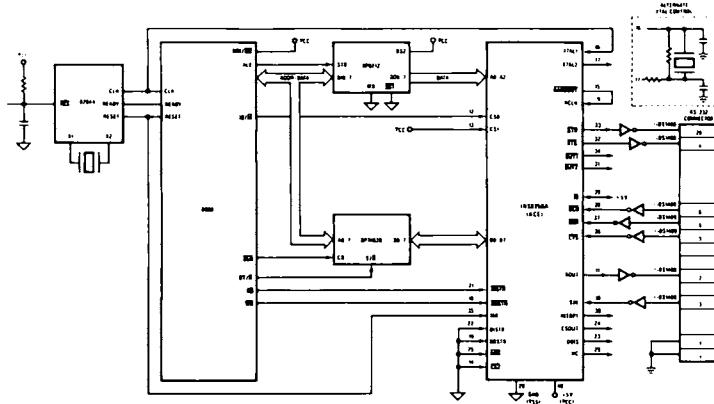
This shows the basic connections of an NS16450 to an NS32016 CPU



TL/C/8401-14

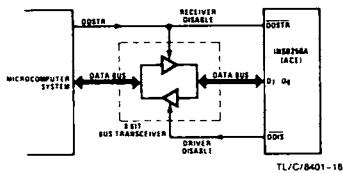
Typical Applications (Continued)

This shows the basic connections of an INS8250A to an 8088 CPU



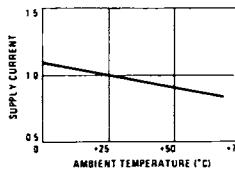
TL/C/8401-15

Typical Interface for a High-Capacity Data Bus



TL/C/8401-16

Typical Supply Current vs. Temperature, Normalized

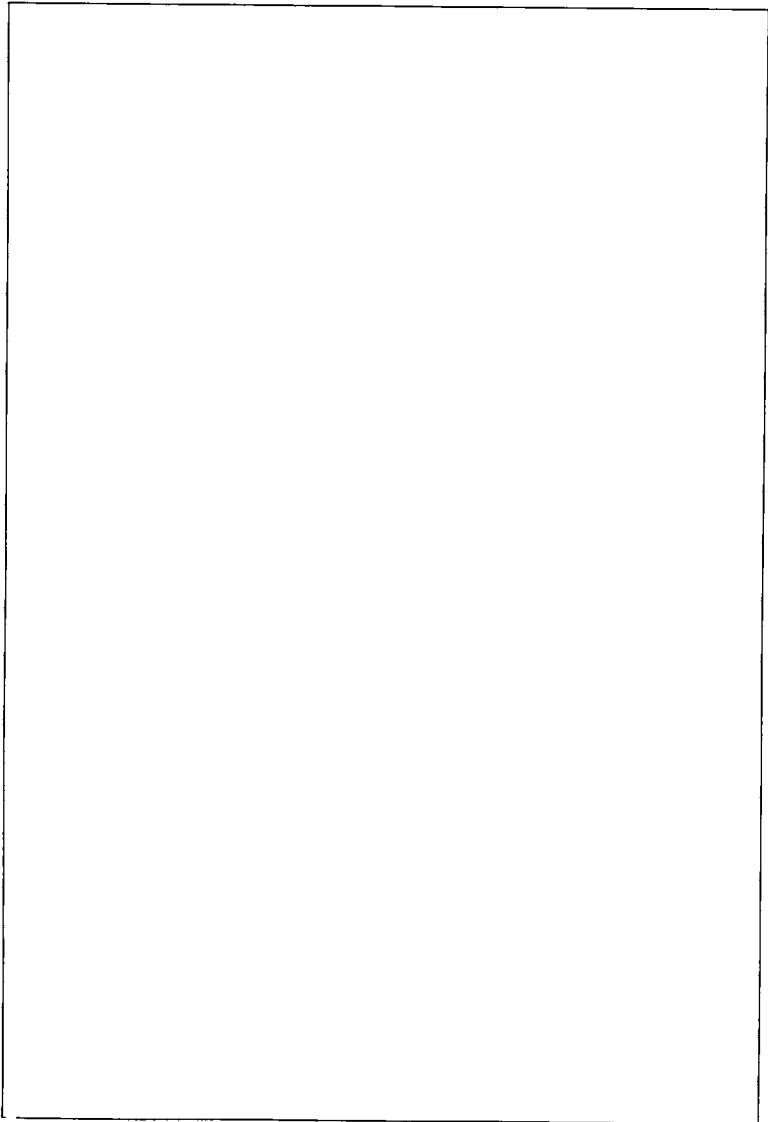


TL/C/8401-17

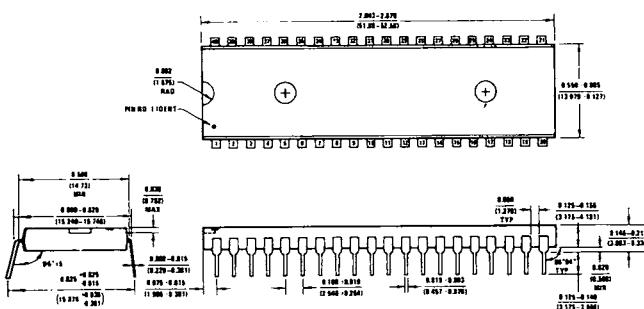
Ordering Information

Order Number	Description
Plastic Dip Package NS16450N or NS-16450N	high speed part
INS8250AN INS8250AWN NS16C450N* INS82C50AN*	$V_{CC} = 5V \pm 5\%$ $V_{CC} = 5V \pm 10\%$ CMOS high speed part CMOS $V_{CC} = 5V \pm 5\%$
Plastic Chip Carrier Package NS16450V or NS-16450V	high speed part
INS8250A NS16C450V* INS82C50AV*	$V_{CC} = 5V \pm 5\%$ CMOS high speed part CMOS $V_{CC} = 5V \pm 5\%$

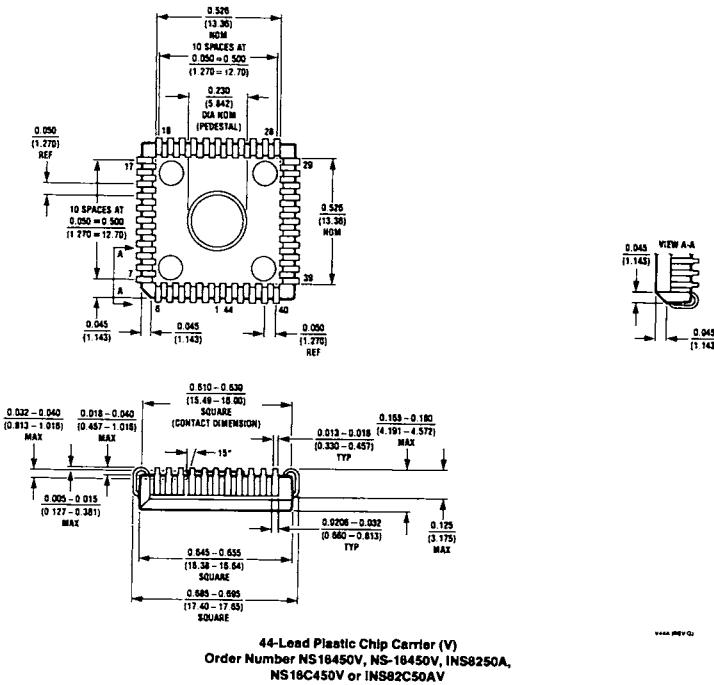
*The CMOS parts will be available after 6/85



Physical Dimensions inches (millimeters)



Plastic Dual-In-Line Package (N)
Order Number NS16450N, NS-16450N, INS8250AN,
INS8250AWN, NS16C450N, or INS82C50AN
NS Package Number N40A

Physical Dimensions inches (millimeters) (Continued)**LIFE SUPPORT POLICY**

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



National Semiconductor
Corporation
2901 Semiconductor Drive
P.O. Box 5060
Santa Clara, CA 95052-9060
Tel: (408) 721-5000
Fax: (916) 239-9240

National Semiconductor
GmbH
Friedrich-Ebert-Strasse Nr. 5
D-8000 Muenchen 21
West Germany
Tel: (089) 5 00 12 0
Tele: 522772

NS Japan K.K.
4-402 Suidobashi,
Togoshi-cho, Chiyoda-ku,
Tokyo 101, Japan
Tel: (03) 585-2151
FAX: 011-61-3-988-1700

N.S.M.E. Ltd.
33, Adele Street, N.W.E., Belgrave
Austin Tower, 14th Floor
22-26 Belgrave Avenue
Tunbridge Wells, Kent, TN1 1JL
Hong Kong
Tel: 3-713-2001, 3-713-0845
Cable: NSSEAMKT
Telex: 520908 NSSEA HK

National Semiconductor
On Broadgate
8 Anderston
22-26 Belgrave Avenue
Tunbridge Wells, Kent, TN1 1JL
Tel: (0171) 215-5000
Telex: 391-1131801 NSDR BR

National Semiconductor
(Australia) Pty. Ltd.
21/3 High Street
Box 1000, Potts Point, NSW 2021
Tel: (013) 511-215-5000
Telex: 391-1131801
Fax: 02-2042000

M5M4256J-10, -12, -15

262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM

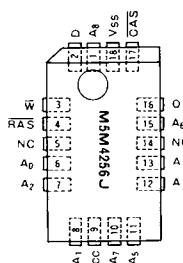
DESCRIPTION

This is a family of 262 144-word by 1-bit dynamic RAMs, fabricated with the high performance N-channel silicon gate MOS process, and is ideal for large-capacity memory systems where high speed, low power dissipation, and low costs are essential. The use of double-layer polysilicon process combined with silicide technology and a single-transistor dynamic storage cell provide high circuit density at reduced costs, and the use of dynamic circuitry including sense amplifiers assures low power dissipation. Multiplexed address inputs permit both a reduction in pins to the 18-pin plastic lead chip carrier configuration and an increase in system densities. In addition to the RAS only refresh mode, the Hidden refresh mode and CAS before RAS refresh mode are available.

FEATURES

Type name	Access time (max) (ns)	Cycle time (min) (ns)	Power dissipation (typ) (mW)
M5M4256J-10	100	200	300
M5M4256J-12	120	230	260
M5M4256J-15	150	260	230

- 18-pin plastic lead chip carrier
- $5V \pm 10\%$ supply
- Low standby power dissipation: 25mW (max)
- Low operating power dissipation:
M5M4256J-10 385mW (max)
M5M4256J-12 360mW (max)
M5M4256J-15 330mW (max)
- Unlatched output enables two-dimensional chip selection and extended page boundary.

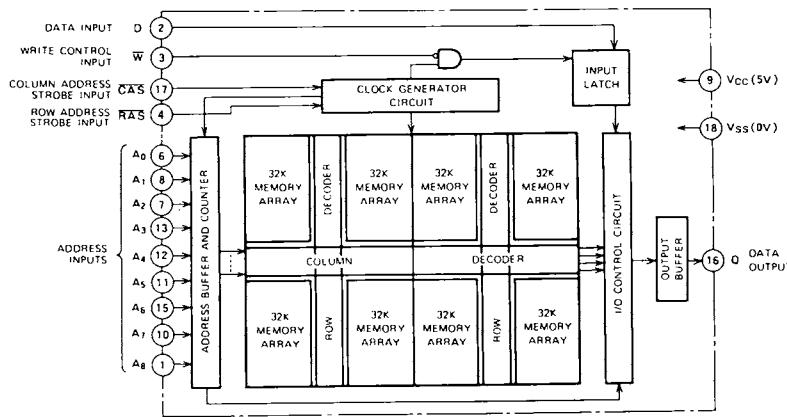
PIN CONFIGURATION (TOP VIEW)

Outline 18P0A NC : NO CONNECTION

- Early-write operation gives common I/O capability
- Read-modify-write, RAS-only-refresh, Page-mode capabilities
- CAS before RAS refresh mode capability
- All input terminals have low input capacitance and are directly TTL-compatible
- Output is three-state and directly TTL-compatible
- 256 refresh cycles every 4ms
- CAS controlled output allows hidden refresh
- Output data can be held infinitely by CAS.

APPLICATION

Main memory unit for computers, Microcomputer memory

BLOCK DIAGRAM

MITSUBISHI LSIs
M5M4256J-10, -12, -15

262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM

FUNCTION

The M5M4256J provides, in addition to normal read, write, and read-modify-write operations, a number of other functions, e.g., page mode, RAS-only refresh, and delayed-write. The input conditions for each are shown in Table 1.

Table 1 Input conditions for each mode

Operation	Inputs						Output Q	Refresh
	RAS	DAS	W	D	Row address	Column address		
Read	ACT	ACT	NAC	DNC	APD	APD	VLD	YES
Write	ACT	ACT	ACT	VLD	APD	APD	OPN	YES
Read-modify-write	ACT	ACT	ACT	VLD	APD	APD	VLD	YES
RAS only refresh	ACT	NAC	DNC	DNC	APD	DNC	OPN	YES
Hidden refresh	ACT	ACT	DNC	DNC	DNC	DNC	VLD	YES
CAS before RAS refresh	ACT	ACT	DNC	DNC	DNC	DNC	OPN	YES
Standby	NAC	DNC	DNC	DNC	DNC	DNC	OPN	NO

Note ACT active, NAC nonactive, DNC don't care, VLD valid, APD applied, OPN open

SUMMARY OF OPERATIONS

Addressing

To select one of the 262 144 memory cells in the M5M4256J the 18-bit address signal must be multiplexed into 9 address signals, which are then latched into the on-chip latch by two externally-applied clock pulses. First, the negative-going edge of the row-address-strobe pulse (RAS) latches the 9 row-address bits; next, the negative-going edge of the column-address-strobe pulse (CAS) latches the 9 column-address bits. Timing of the RAS and CAS clocks can be selected by either of the following two methods:

1. The delay time from RAS to CAS ($t_{(RAS-CAS)}$) is set between the minimum and maximum values of the limits. In this case, the internal CAS control signals are inhibited almost until $t_{(RAS-CAS)} \max$ ('gated CAS' operation). The external CAS signal can be applied with a margin not affecting the on-chip circuit operations, e.g. access time, and the address inputs can be easily changed from row address to column address.
2. The delay time $t_{(RAS-CAS)}$ is set larger than the maximum value of the limits. In this case the internal inhibition of CAS has already been released, so that the internal CAS control signals are controlled by the externally applied CAS, which also controls the access time.

Data Input

Data to be written into a selected cell is strobed by the later of the two negative transitions of W input and CAS input. Thus when the W input makes its negative transition prior to CAS input (early write), the data input is strobed by CAS, and the negative transition of CAS is set as the

reference point for set-up and hold times. In the read-write or read-modify-write cycles, however, when the W input makes its negative transition after CAS, the W negative transition is set as the reference point for setup and hold times.

Data Output Control

The output of the M5M4256J is in the high-impedance state when CAS is high. When the memory cycle in progress is a read, read-modify-write, or a delayed-write cycle, the data output will go from the high-impedance state to the active condition, and the data in the selected cell will be read. This data output will have the same polarity as the input data. Once the output has entered the active condition, this condition will be maintained until CAS goes high, irrespective of the condition of RAS.

The output will remain in the high-impedance state throughout the entire cycle in an early-write cycle.

These output conditions, of the M5M4256J which can readily be changed by controlling the timing of the write pulse in a write cycle, and the width of the CAS pulse in a read cycle, offer capabilities for a number of applications, as follows.

1. Common I/O Operation

If all write operations are performed in the early-write mode, input and output can be connected directly to give a common I/O data bus.

2. Data Output Hold

The data output can be held between read cycles, without lengthening the cycle time. This enables extremely flexible clock-timing settings for RAS and CAS.

MITSUBISHI LSIs
M5M4256J-10, -12, -15

262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM

3. Two Methods of Chip Selection

Since the output is not latched, $\overline{\text{CAS}}$ is not required to keep the outputs of selected chips in the matrix in a high-impedance state. This means that $\overline{\text{CAS}}$ and/or $\overline{\text{RAS}}$ can both be decoded for chip selection.

4. Extended-Page Boundary

By decoding $\overline{\text{CAS}}$, the page boundary can be extended beyond the 512 column locations in a single chip. In this case, $\overline{\text{RAS}}$ must be applied to all devices.

Page-Mode Operation

This operation allows for multiple-column addressing at the same row address, and eliminates the power dissipation associated with the negative-going edge of $\overline{\text{RAS}}$, because once the row address has been strobed, $\overline{\text{RAS}}$ is maintained. Also, the time required to strobe in the row address for the second and subsequent cycles is eliminated, thereby decreasing the access and cycle times.

Refresh

Each of the 256 rows ($A_0 \sim A_7$) of the M5M4256J must be refreshed every 4 ms to maintain data. The methods of refreshing for the M5M4256J are as follows.

1. Normal Refresh

Read cycle and Write cycle (early write, delayed write or read-modify-write) refresh the selected row as defined by the low order ($\overline{\text{RAS}}$) addresses. Any write cycle, of course, may change the state of the selected cell. Using a read, write, or read-modify-write cycle for refresh is not recommended for systems which utilize "wired-OR" outputs since output bus contention will occur.

2. RAS Only Refresh

In this refresh method, the $\overline{\text{CAS}}$ clock should be at a V_{IH} level and the system must perform $\overline{\text{RAS}}$ Only cycle on all 256 row address every 4 ms. The sequential row addresses from an external counter are latched with the $\overline{\text{RAS}}$ clock and associated internal row locations are refreshed. A $\overline{\text{RAS}}$ Only Refresh cycle maintains the output in the high impedance state with a typical power reduction of 20% over a read or write cycle.

3. CAS before RAS Refresh

If $\overline{\text{CAS}}$ falls $t_{SUH}(\overline{\text{CAS}}, \overline{\text{RAS}})$ earlier than $\overline{\text{RAS}}$ and if $\overline{\text{CAS}}$ is kept low by $t_{HR}(\overline{\text{RAS}}, \overline{\text{CAS}})$ after $\overline{\text{RAS}}$ falls, $\overline{\text{CAS}}$ before $\overline{\text{RAS}}$ Refresh is initiated. The external address is ignored and the refresh address generated by the internal 8-bit counter is put into the address buffer to refresh the corresponding row. The output will stay in the high impedance state.

If $\overline{\text{CAS}}$ is kept low after the above operation, $\overline{\text{RAS}}$ cycle initiates $\overline{\text{RAS}}$ Only Refresh with internally generated refresh address (Automatic refresh). The output will again stay in the high impedance state.

Bringing $\overline{\text{RAS}}$ high and then low while $\overline{\text{CAS}}$ remains high initiates the normal $\overline{\text{RAS}}$ Only Refresh using the external address.

If $\overline{\text{CAS}}$ is kept low after the normal read/write cycle, $\overline{\text{RAS}}$ cycle initiates the $\overline{\text{RAS}}$ Only Refresh using the internal refresh address and especially after the normal read cycle, it becomes Hidden Refresh with internal address. The output is available until $\overline{\text{CAS}}$ is brought high.

4. Hidden Refresh

A feature of the M5M4256J is that refresh cycles may be performed while maintaining valid data at the output pin by extending the $\overline{\text{CAS}}$ active time from a previous memory read cycle. This feature is referred to as hidden refresh.

Hidden refresh is performed by holding $\overline{\text{CAS}}$ at V_{IL} and taking $\overline{\text{RAS}}$ high and after a specified precharge period, executing a $\overline{\text{RAS}}$ -only cycling, but with $\overline{\text{CAS}}$ held low.

The advantage of this refresh mode is that data can be held valid at the output data port indefinitely by leaving the $\overline{\text{CAS}}$ asserted. In many applications this eliminates the need for off-chip latches.

Power Dissipation

Most of the circuitry in the M5M4256J is dynamic, and most of the power is dissipated when addresses are strobed. Both $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ are decoded and applied to the M5M4256J as chip-select in the memory system, but if $\overline{\text{RAS}}$ is decoded, all unselected devices go into stand-by independent of the $\overline{\text{CAS}}$ condition, minimizing system power dissipation.

Power Supplies

The M5M4256J operates on a single 5V power supply.

A wait of some 500 μ s and eight or more dummy cycles is necessary after power is applied to the device before memory operation is achieved.

MITSUBISHI LSIs
M5M4256J-10, -12, -15

262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage		1 ~ 7	V
V_I	Input voltage		1 ~ 7	V
V_O	Output voltage		- 1 ~ 7	V
I_O	Output current		50	mA
P_d	Power dissipation	$T_a = 25^\circ\text{C}$	1000	mW
T_{opr}	Operating temperature		0 ~ 70	°C
T_{stg}	Storage temperature		- 65 ~ 150	°C

RECOMMENDED OPERATING CONDITIONS ($T_a = 0 \sim 70^\circ\text{C}$, unless otherwise noted) (Note 1)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.5	5	5.5	V
V_{SS}	Supply voltage	0	0	0	V
V_{IH}	High level input voltage, all inputs	2.4		6.5	V
V_{IL}	Low level input voltage, all inputs	- 2		0.8	V

Note 1. All voltage values are with respect to V_{SS}

ELECTRICAL CHARACTERISTICS ($T_a = 0 \sim 70^\circ\text{C}$, $V_{CC} = 5\text{V} \pm 10\%$, $V_{SS} = 0\text{V}$, unless otherwise noted) (Note 2)

Symbol	Parameter	Test conditions			Unit
		Min	Typ	Max	
V_{OH}	High-level output voltage	$I_{OH} = - 5\text{mA}$	2.4		V _{CC}
V_{OL}	Low-level output voltage	$I_{OL} = 4.2\text{mA}$	0	0.4	V
I_{OZ}	Off-state output current	Q floating	$0 \leq V_{OUT} \leq 5.5\text{V}$	- 10	μA
I_1	Input current	$0 \leq V_{IN} \leq V_{CC}$, Other input pins = 0V	- 10	10	μA
$I_{CC1(AV)}$	Average supply current from V_{CC} , operating (Note 3, 4)	M5M4256J-10 M5M4256J-12 M5M4256J-15	RAS, CAS cycling $t_{CR} = t_{CW} = \text{min. output open}$	70 65 60	mA
	Supply current from V_{CC} , standby		RAS = CAS = V_{IH} output open	4.5	mA
				60 55	mA
$I_{CC3(AV)}$	Average supply current from V_{CC} , refreshing (Note 3)	M5M4256J-10 M5M4256J-12 M5M4256J-15	RAS cycling $CAS = V_{IH}$ $t_{CR(RAS)} = \text{min. output open}$	50 55 50	mA
	Average supply current from V_{CC} , page mode (Note 3, 4)	M5M4256J-10 M5M4256J-12 M5M4256J-15	RAS = V_{IH} , CAS cycling $t_{CPG} = \text{min. output open}$	55 50 45	mA
				65 60	mA
$I_{CC4(AV)}$	Average supply current from V_{CC} , page mode (Note 3, 4)	M5M4256J-10 M5M4256J-12 M5M4256J-15	RAS = V_{IH} , CAS before RAS refresh cycling $t_{CPG} = \text{min. output open}$	55 50 45	mA
	Average supply current from V_{CC} , CAS before RAS refresh mode (Note 3)	M5M4256J-10 M5M4256J-12 M5M4256J-15	CAS before RAS refresh cycling $t_{CR(RAS)} = \text{min. Output open}$	65 60 55	mA
				5	pF
$C_I(A)$	Input capacitance, address inputs			5	pF
$C_I(D)$	Input capacitance, data input			5	pF
$C_I(W)$	Input capacitance, write control input			7	pF
$C_I(RAS)$	Input capacitance, RAS input			10	pF
$C_I(CAS)$	Input capacitance, CAS input			10	pF
C_O	Output capacitance	$V_O = V_{SS}$, $f = 1\text{MHz}$, $V_I = 25\text{mVrms}$		7	pF

Note 2. Current flowing into an IC is positive, out is negative.

Note 3. $I_{CC1(AV)}$, $I_{CC3(AV)}$, $I_{CC4(AV)}$ and $I_{CC6(AV)}$ are dependent on cycle rate. Maximum current is measured at the fastest cycle rate.

Note 4. $I_{CC1(AV)}$ and $I_{CC4(AV)}$ are dependent on output loading. Specified values are obtained with the output open.

262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM**TIMING REQUIREMENTS (For Read, Write, Read-Modify-Write, Refresh, and Page-Mode Cycle)**
(Ta = 0 ~ 70°C, Vcc = 5V ± 10%, Vss = 0V, unless otherwise noted. See notes 5, 6 and 7)

Symbol	Parameter	Alternative Symbol	Limits						Unit
			M5M4256J-10		M5M4256J-12		M5M4256J-15		
Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
tREF	Refresh cycle time	tREF	—	4	—	4	—	4	ns
tW(RASH)	RAS high pulse width	tRP	90	—	100	—	100	—	ns
tW(RASL)	RAS low pulse width	tRAS	100	: 10000	120	: 10000	150	: 10000	ns
tW(CASL)	CAS low pulse width	tCAS	50	—	60	—	75	—	ns
tW(CASH)	CAS high pulse width	tCPH	25	—	30	—	35	—	ns
tH(RAS-CAS)	CAS hold time after RAS	tCSH	100	—	120	—	150	—	ns
tH(CAS-RAS)	RAS hold time after CAS	tRSH	50	—	60	—	75	—	ns
tD(CAS-RAS)	Delay time, CAS to RAS	tCRP	20	—	30	—	30	—	ns
tD(RAS-CAS)	Delay time, RAS to CAS	tCSR	15	50	20	60	25	75	ns
tSU(RA-RAS)	Row address setup time before RAS	tASR	0	—	0	—	0	—	ns
tSU(CA-CAS)	Column address setup time before CAS	tASC	-5	—	-5	—	-5	—	ns
tH(RAS-RA)	Row address hold time after RAS	tRAH	10	—	15	—	20	—	ns
tH(CAS-CA)	Column address hold time after CAS	tCAH	15	—	20	—	25	—	ns
tH(RAS-CA)	Column address hold time after RAS	tAR	65	—	80	—	100	—	ns
tTHL	Transition time	tT	3	50	3	50	3	50	ns
tTLH	—	tT	3	50	3	50	3	50	ns

Note 5 An initial pause of 500μs is required after power up followed by any eight RAS or RAS/CAS cycles before proper device operation is achieved.

6 The switching characteristics are defined as $t_{THL} = t_{TLH}$ = 5ns.7 Reference levels of input signals are $V_{IH\ min}$ and $V_{IL\ max}$. Reference levels for transition time are also between V_{IH} and V_{IL} .

8 Except for page mode.

9 $t_d(RAS-CAS)$ requirement is applicable for all RAS/CAS cycles.10 Operation within the $t_d(RAS-CAS)$ max limit insures that $t_a(RAS)$ max can be met. $t_d(RAS-CAS)$ max is specified reference point only. If $t_d(RAS-CAS)$ is greater than the specified $t_d(RAS-CAS)$ max limit, then access time is controlled exclusively by $t_a(RAS)$. $t_d(RAS-CAS)\ min = t_h(RAS-RA)\ min + 2(t_{THL(t_{TLH})} + t_{SU(CA-CAS)})\ min$.**SWITCHING CHARACTERISTICS (Ta = 0 ~ 70°C, Vcc = 5V ± 10%, Vss = 0V, unless otherwise noted)****Read Cycle**

Symbol	Parameter	Alternative Symbol	Limits						Unit
			M5M4256J-10		M5M4256J-12		M5M4256J-15		
Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
tRC	Read cycle time	tRC	200	—	230	—	260	—	ns
tSU(R-A)	Read setup time before CAS	tRCS	0	—	0	—	0	—	ns
tH(CAS-R)	Read hold time after CAS	tRCH	0	—	0	—	0	—	ns
tH(RAS-R)	Read hold time after RAS	tRHR	20	—	20	—	20	—	ns
tOFF(CAS)	Output disable time	tOFF	0	25	0	35	0	40	ns
tAC(RAS)	CAS access time	tCAC	50	—	60	—	75	—	ns
tAC(RAS)	RAS access time	tRAC	100	—	120	—	150	—	ns

Note 11 Either $t_h(RAS-R)$ or $t_h(CAS-R)$ must be satisfied for a read cycle.12 $t_d(RAS-CAS)$ max defines the time at which the output achieves the open circuit condition and is not reference to V_{OH} or V_{OL} .13 This is the value when $t_d(RAS-CAS) \geq t_d(RAS-CAS)$ max. Test conditions, Load = 2TTL, $C_L = 100\text{pF}$.14 This is the value when $t_d(RAS-CAS) < t_d(RAS-CAS)$ max. When $t_d(RAS-CAS) \geq t_d(RAS-CAS)$ max, $t_a(RAS)$ will increase by the amount that $t_d(RAS-CAS)$ exceeds the value shown. Test conditions, Load = 2TTL, $C_L = 100\text{pF}$.**Write Cycle**

Symbol	Parameter	Alternative Symbol	Limits						Unit
			M5M4256J-10		M5M4256J-12		M5M4256J-15		
Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
tCW	Write cycle time	tRC	200	—	230	—	260	—	ns
tSU(W-CAS)	Write setup time before CAS	tWCS	-10	—	10	—	-10	—	ns
tH(CAS-W)	Write hold time after CAS	tWCH	35	—	40	—	45	—	ns
tH(RAS-W)	Write hold time after RAS	tWCR	85	—	100	—	120	—	ns
tHW(RAS)	RAS hold time after write	tRWL	35	—	40	—	45	—	ns
tHW(CAS)	CAS hold time after write	tCWL	35	—	40	—	45	—	ns
tW(W)	Write pulse width	tWP	35	—	40	—	45	—	ns
tSU(D-CAS)	Data in setup time before CAS	tDS	0	—	0	—	0	—	ns
tH(CAS-D)	Data in hold time after CAS	tDH	25	—	30	—	35	—	ns
tH(RAS-D)	Data in hold time after RAS	tDHR	70	—	90	—	110	—	ns

MITSUBISHI LSIs
MSM4256J-10, -12, -15

262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM

Read, Write and Read-Modify-Write Cycles

Symbol	Parameter	Alternative Symbol	Limits						Unit
			M5M4256J-10		M5M4256J-12		M5M4256J-15		
Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
t_{CRW}	Read-write cycle time (Note 15)	t_{RWC}	225		260		295		ns
t_{CRMW}	Read-modify write cycle time (Note 16)	t_{RMWC}	235		275		310		ns
$t_{h(w-RAS)}$	RAS hold time after write	t_{RWL}	35		40		45		ns
$t_{h(w-CAS)}$	CAS hold time after write	t_{CWL}	35		40		45		ns
$t_{w(w)}$	Write pulse width	t_{WP}	35		40		45		ns
$t_{SU(w-CAS)}$	Read setup time before CAS	t_{RCS}	0		0		0		ns
$t_{d(RAS-w)}$	Delay time, RAS to write (Note 17)	t_{RWD}	90		110		135		ns
$t_{d(CAS-w)}$	Delay time, CAS to write (Note 17)	t_{CWD}	40		50		60		ns
$t_{SU(D-W)}$	Data in setup time before write	t_{DS}	0		0		0		ns
$t_{h(w-D)}$	Data in hold time after write	t_{DH}	35		40		45		ns
$t_{dS(CAS)}$	Output disable time	t_{DF}	0	25	0	35	0	40	ns
$t_a(CAS)$	CAS access time (Note 18)	t_{CAC}	50		60		75		ns
$t_a(RAS)$	RAS access time (Note 14)	t_{RAC}	100		120		150		ns

Note 15. t_{CRW} min is defined as t_{CRW} min = $t_d(RAS-CAS)$ max + $t_d(CAS-w)$ min + $t_h(w-RAS)$ + $t_w(RASH)$ + 3t $T_{LM}(t_{HL})$

16. t_{CRMW} min is defined as t_{CRMW} min = $t_a(RAS)$ max + $t_h(w-RAS)$ + $t_w(RASH)$ + 3t $T_{LM}(t_{HL})$

17. $t_{h(w-CAS)}$, $t_d(RAS-w)$, and $t_d(CAS-w)$ do not define the limits of operation, but are included as electrical characteristics only.

When $t_{SU(w-CAS)} \geq t_{SU(w-CAS)}$ min, an early write cycle is performed, and the data output keeps the high-impedance state.

When $t_d(RAS-w) \geq t_d(RAS-w)$ min, and $t_d(CAS-w) \geq t_{SU(w-CAS)}$ min a read-write cycle is performed, and the data of the selected address will be read out on the data output.

For all conditions other than those described above, the condition of data output lat access time and until CAS goes back to V_{IH} is not defined.

Page-Mode Cycle

Symbol	Parameter	Alternative Symbol	Limits						Unit
			M5M4256J-10		M5M4256J-12		M5M4256J-15		
Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
t_{CPG}	Page mode cycle time	t_{PC}	100		125		145		ns
$t_w(CASH)$	CAS high pulse width	t_{CP}	40		55		60		ns
t_{CPGRW}	Page-mode read-write cycle time	t_{PCRW}	140		160		180		ns

CAS before RAS Refresh Cycle (Note 18)

Symbol	Parameter	Alternative Symbol	Limits						Unit
			M5M4256J-10		M5M4256J-12		M5M4256J-15		
Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
$t_{SR(RAS-RAS)}$	CAS setup time for auto refresh	t_{CSR}	25		30		30		ns
$t_{HR(RAS-CAS)}$	CAS hold time for auto refresh	t_{CHR}	40		50		50		ns
$t_{DR(RAS-CAS)}$	Precharge to CAS active time	t_{RPC}	0		0		0		ns

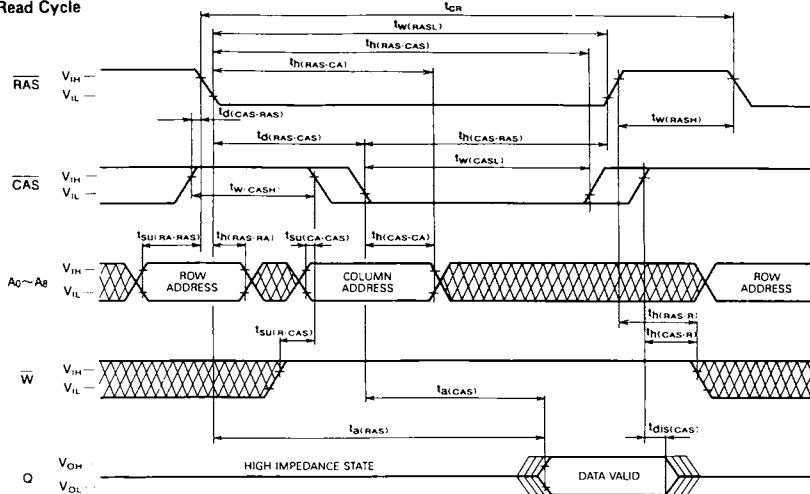
Note 18 Eight or more CAS before RAS cycles is necessary for proper operation of CAS before RAS refresh mode

MITSUBISHI LSIs
M5M4256J-10, -12, -15

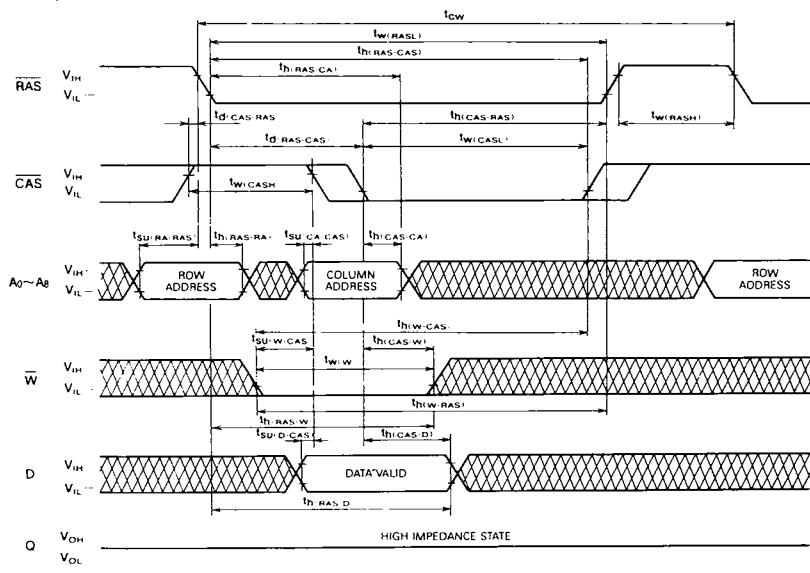
262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM

TIMING DIAGRAMS (Note 19)

Read Cycle



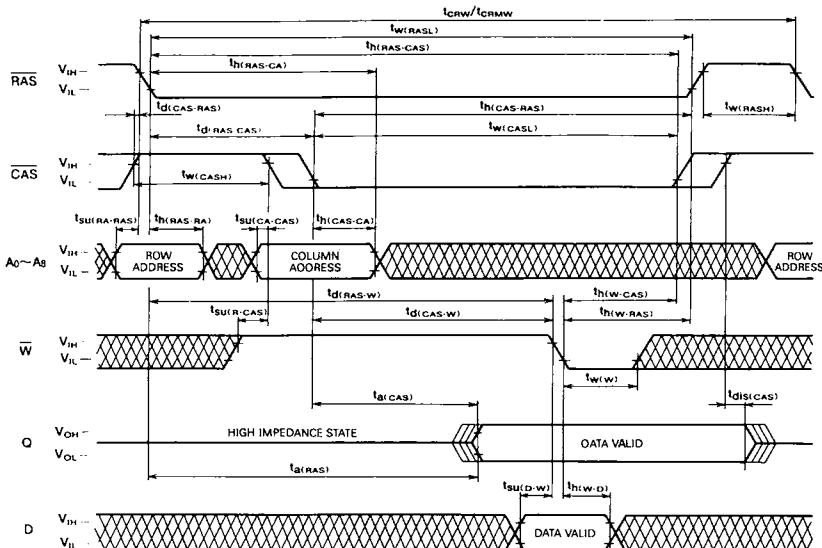
Write Cycle (Early Write)



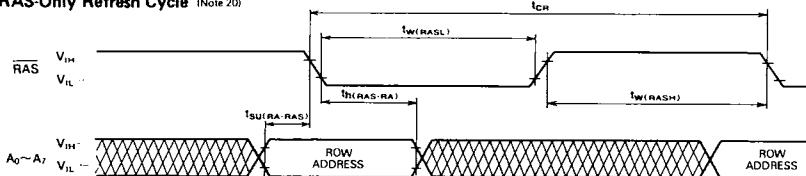
MITSUBISHI LSIs
M5M4256J-10, -12, -15

262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM

Read-Write and Read-Modify-Write Cycles



RAS-Only Refresh Cycle (Note 20)



Note 19. Indicates the don't care input.

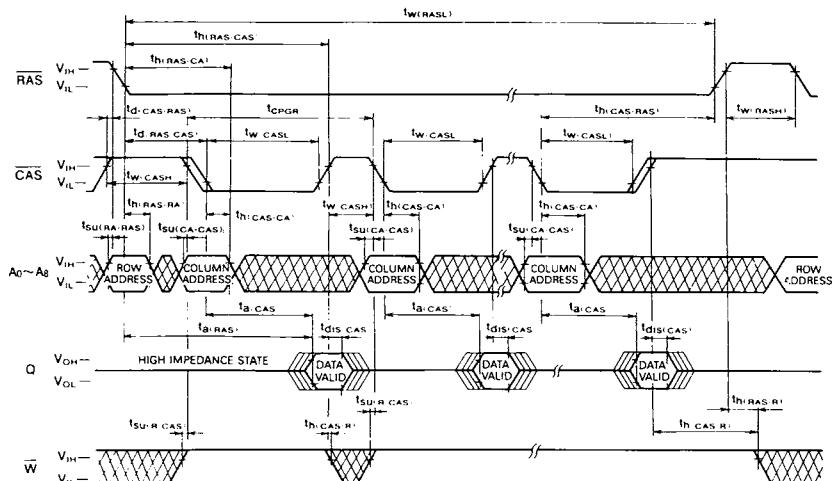
Note 20. $\overline{CAS} = V_{IH}$, $\overline{W}, D = \text{don't care}$.
 A_8 may be V_{IH} or V_{IL} .

The center-line indicates the high-impedance state.

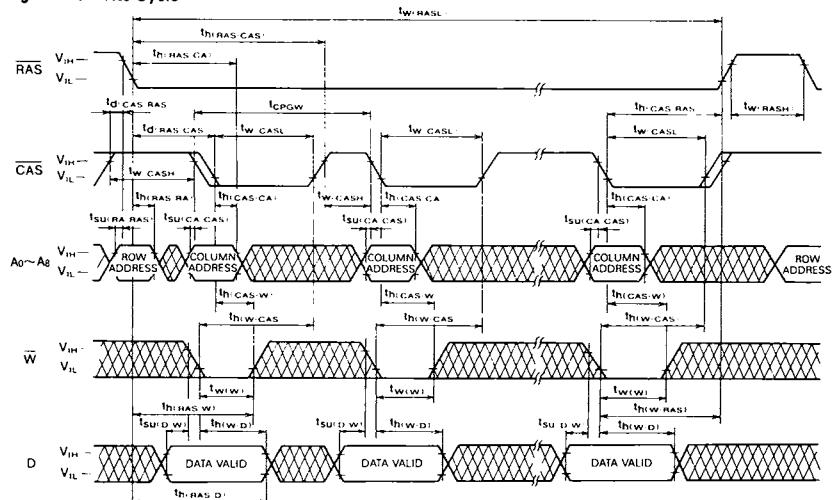
MITSUBISHI LSIs
M5M4256J-10, -12, -15

262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM

Page-Mode Read Cycle



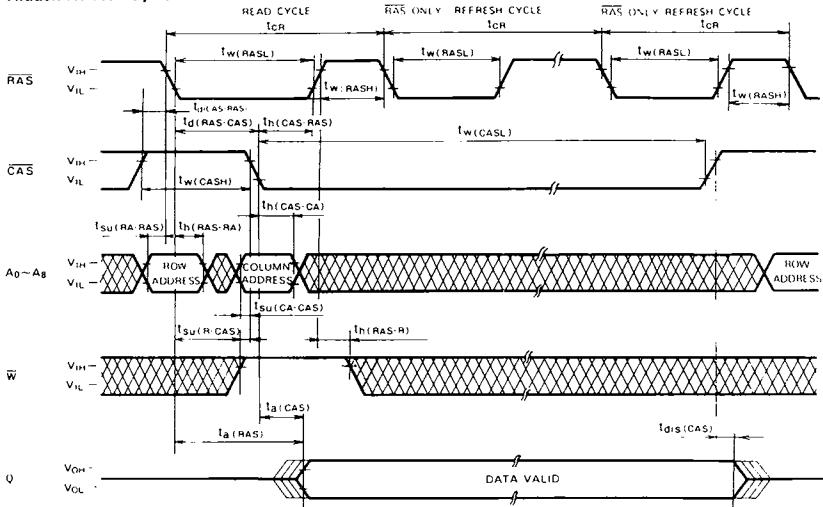
Page-Mode Write Cycle



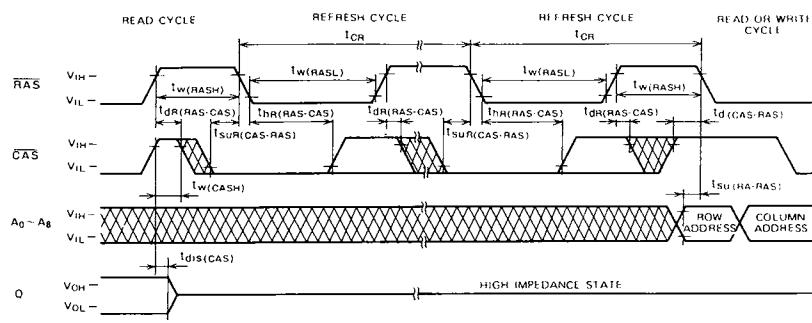
MITSUBISHI LSIs
M5M4256J-10, -12, -15

262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM

Hidden Refresh Cycle



CAS before RAS Refresh (Note 21)



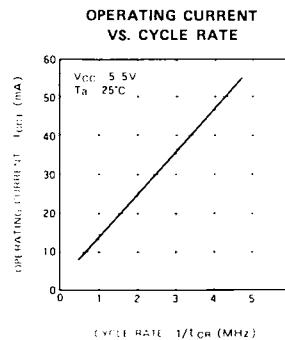
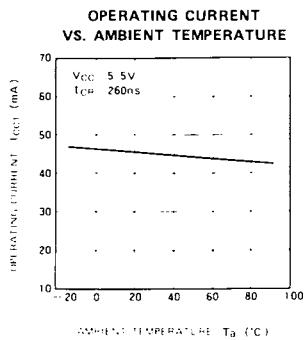
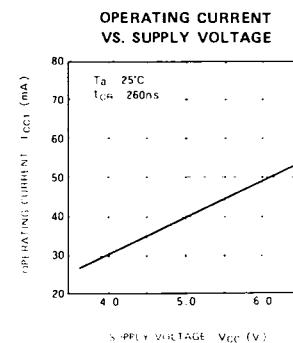
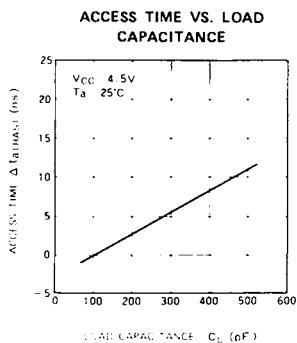
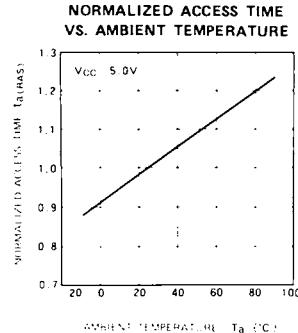
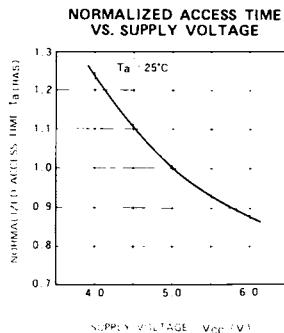
Note 21 W, D - don't care

MITSUBISHI LSIs

M5M4256J-10, -12, -15

262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM

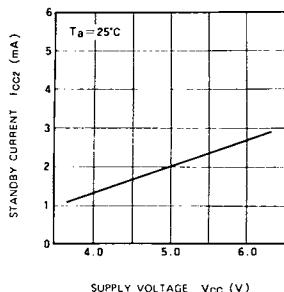
TYPICAL CHARACTERISTICS



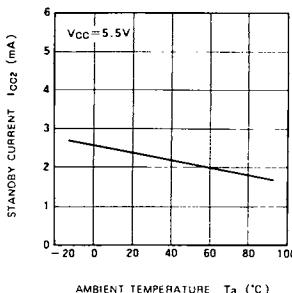
MITSUBISHI LSIs
M5M4256J-10, -12, -15

262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM

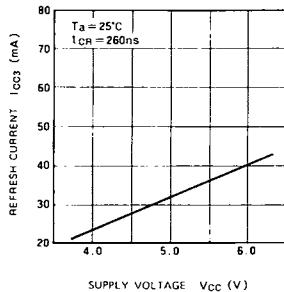
**STANDBY CURRENT
VS. SUPPLY VOLTAGE**



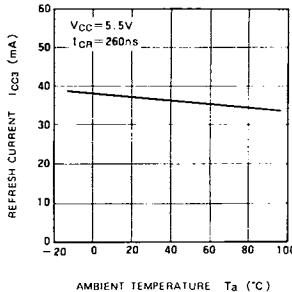
**STANDBY CURRENT
VS. AMBIENT TEMPERATURE**



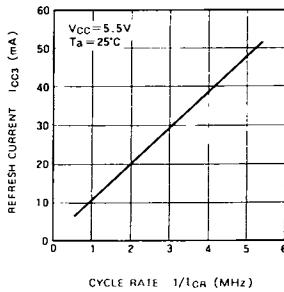
**REFRESH CURRENT
VS. SUPPLY VOLTAGE**



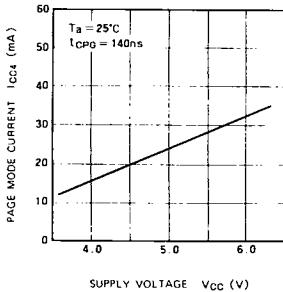
**REFRESH CURRENT
VS. AMBIENT TEMPERATURE**



**REFRESH CURRENT
VS. CYCLE RATE**



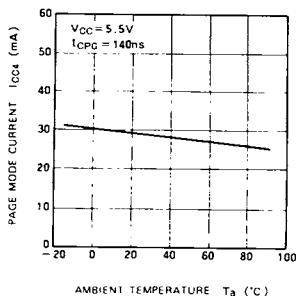
**PAGE MODE CURRENT
VS. SUPPLY VOLTAGE**



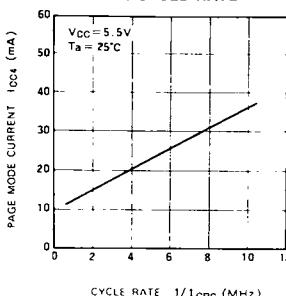
MITSUBISHI LSIs
MSM4256J-10, -12, -15

262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM

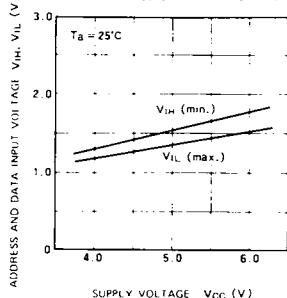
PAGE MODE CURRENT
VS. AMBIENT TEMPERATURE



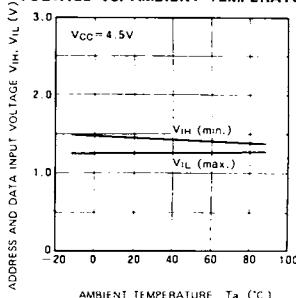
PAGE MODE CURRENT
VS. CYCLE RATE



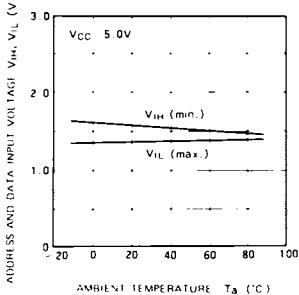
ADDRESS AND DATA INPUT
VOLTAGE VS. SUPPLY VOLTAGE



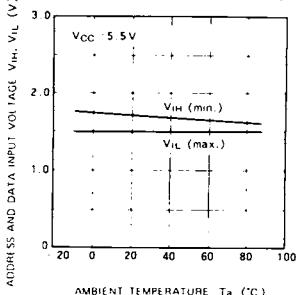
ADDRESS AND DATA INPUT
VOLTAGE VS. AMBIENT TEMPERATURE



ADDRESS AND DATA INPUT
VOLTAGE VS. AMBIENT TEMPERATURE

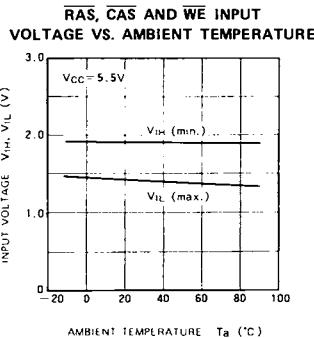
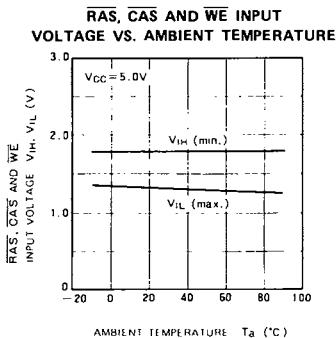
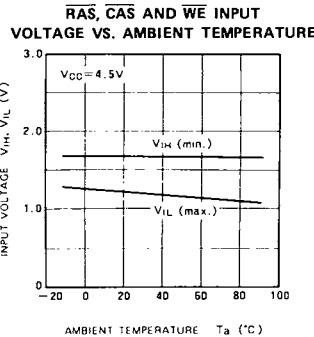
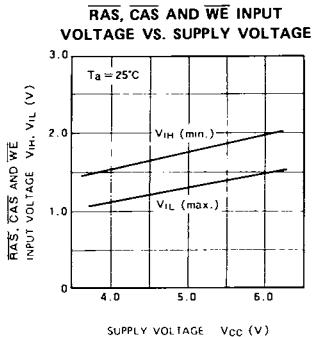


ADDRESS AND DATA INPUT
VOLTAGE VS. AMBIENT TEMPERATURE



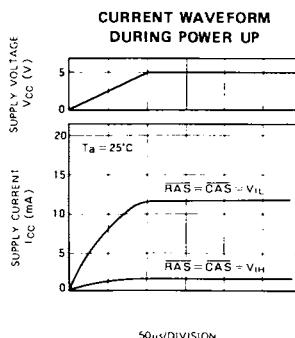
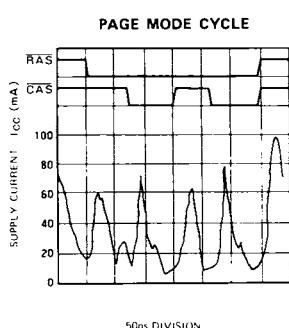
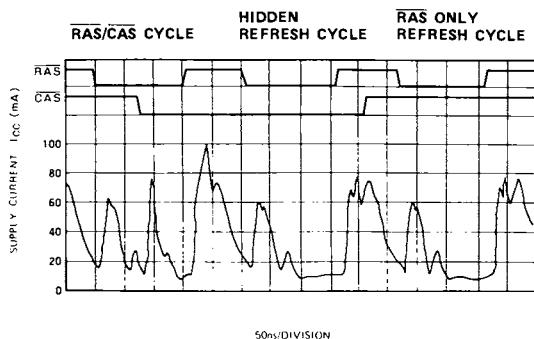
MITSUBISHI LSIs
MSM4256J-10, -12, -15

262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM



MITSUBISHI LSIs
M5M4256J-10, -12, -15

262144-BIT (262144-WORD BY 1-BIT) DYNAMIC RAM







MOTOROLA

MC14066B

QUAD ANALOG SWITCH/QUAD MULTIPLEXER

The MC14066B consists of four independent switches capable of controlling either digital or analog signals. This quad bilateral switch is useful in signal gating, chopper, modulator, demodulator and CMOS logic implementation.

The MC14066B is designed to be pin-for-pin compatible with the MC14016B, but has much lower ON resistance. Input voltage swings as large as the full supply voltage can be controlled via each independent control input.

- High On/Off Output Voltage Ratio - 65 dB typical
- Quiescent Current = 0.5 nA/package typical @ 5 Vdc
- Low Crosstalk Between Switches -50 dB typical @ 8 MHz
- Diode Protection on All Inputs
- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Transmits Frequencies Up to 65 MHz @ 10 Vdc
- Linearized Transfer Characteristics, $\Delta R_{ON} < 60 \Omega$ for $V_{IN} = V_{DD}$ to V_{SS} (at 15V)
- Low Noise - 12 nV/ $\sqrt{\text{Cycle}}$, f ≥ 1 kHz typical
- Pin-for-Pin Replacement for CD4016, CD4066, MC14016

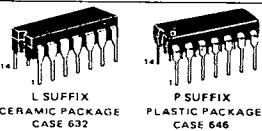
MAXIMUM RATINGS (Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
DC Supply Voltage	V_{DD}	-0.5 to +18	Vdc
Input Voltage, All Inputs	V_{IN}	-0.5 to $V_{DD} + 0.5$	Vdc
Through Current	I	25	mA/dc
Operating Temperature Range - AL Device CL/CP Device	T_A	-55 to +125 -40 to +85	°C
Storage Temperature Range	T_{STG}	-65 to +150	°C

CMOS SSI

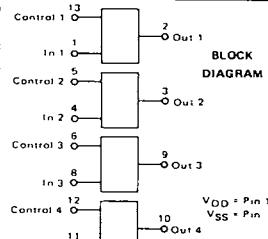
(LOW POWER COMPLEMENTARY MOS)

QUAD ANALOG SWITCH QUAD MULTIPLEXER

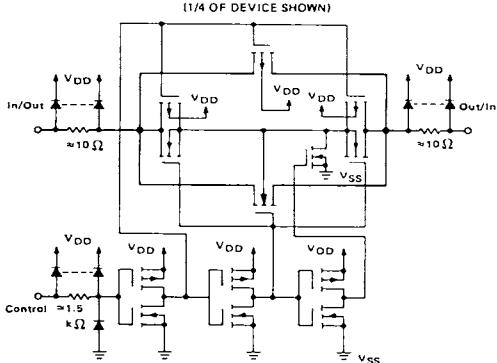


ORDERING INFORMATION

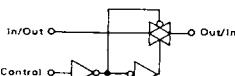
MC14XX	Suffix Denotes
L	Ceramic Package
P	Plastic Package
A	Extended Operating Temperature Range
C	Limited Operating Temperature Range



CIRCUIT SCHEMATIC (1/4 OF DEVICE SHOWN)



LOGIC DIAGRAM AND TRUTH TABLE (1/4 OF DEVICE SHOWN)



Control	Switch	Logic Diagram Restrictions
0	OFF	$V_{SS} \leq V_{IN} \leq V_{DD}$
1	ON	$V_{SS} \leq V_{OUT} \leq V_{DD}$

$V_{Control}$	V_{IN} to V_{OUT} Resistance
V_{SS}	$> 10^9 \text{ Ohms typ}$
V_{DD}	$3 \times 10^2 \text{ Ohms typ}$

MC14066B

ELECTRICAL CHARACTERISTICS

Characteristic	Symbol	V _{DD} Vdc	T _{low} *		25°C			T _{high} *		Unit
			Min	Max	Min	Typ	Max	Min	Max	
Input Voltage (Control) "0" Level (V _O = 4.5 or 9.0 Vdc)	V _{IL}	5.0	-	1.5	-	2.25	1.5	-	1.5	Vdc
		10	-	3.0	-	4.50	3.0	-	3.0	
		15	-	3.75	-	6.75	3.75	-	3.75	
"1" Level (V _O = 0.5 or 4.5 Vdc) (V _O = 1.0 or 9.0 Vdc) (V _O = 1.5 or 13.5 Vdc)	V _{IH}	5.0	3.5	-	3.5	2.75	-	3.5	-	Vdc
		10	7.0	-	7.0	5.50	-	7.0	-	
		15	11.25	-	11.25	8.75	-	11.25	-	
Input Current (AL Device) Control	I _{IN}	15	-	±0.1	-	±0.00001	±0.1	-	±1.0	μA/dc
Input Current (CL/CP Device) Control	I _{IN}	15	-	±0.3	-	±0.00001	±0.3	-	±1.0	μA/dc
Input Capacitance (V _{in} = 0) Control Input Switch Inputs	C _{IN}	-	-	-	-	5.0	7.5	-	-	pF
Output Capacitance	C _{OUT}	10	-	-	-	8.0	-	-	-	pF
Leadthrough Capacitance	C _{IN} + C _{OUT}	10	-	-	-	0.5	-	-	-	pF
Quiescent Current (AL Device) (Per Package)	I _Q	5.0	-	0.25	-	0.0005	0.25	-	7.5	μA/dc
		10	-	0.50	-	0.0010	0.50	-	15	
		15	-	1.00	-	0.0015	1.00	-	30	
Quiescent Current (CL/CP Device) (Per Package)	I _Q	5.0	-	1.0	-	0.0005	1.0	-	7.5	μA/dc
		10	-	2.0	-	0.0010	2.0	-	15	
		15	-	4.0	-	0.0015	4.0	-	30	
ON Resistance (AL Device)	R _{ON}	5.0	-	800	-	250	1050	-	1200	Ω
		10	-	400	-	120	500	-	520	
		15	-	270	-	80	280	-	300	
ON Resistance (CL/CP Device)	R _{ON}	5.0	-	880	-	250	1050	-	1300	Ω
		10	-	450	-	120	500	-	550	
		15	-	250	-	80	280	-	320	
Δ ON Resistance Between Any Two of Four Switches	ΔR _{ON}	5.0	-	-	-	25	-	-	-	Ω
		10	-	-	-	10	-	-	-	
		15	-	-	-	5.0	-	-	-	
Input/Output Leakage Current Switch OFF (AL Device)	-	15	-	±100	-	±0.01	±100	-	±1000	nA/dc
Input/Output Leakage Current Switch OFF (CL/CP Device)	-	15	-	±300	-	±0.01	±300	-	±1000	nA/dc

*The formulas given are for the typical characteristics only.

T_{low} = -55°C for AL Device, -40°C for CL/CP Device.

T_{high} = +125°C for AL Device, +85°C for CL/CP Device.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range V_{SS} < V_{in} or V_{out} < V_{DD}.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

MC14066B

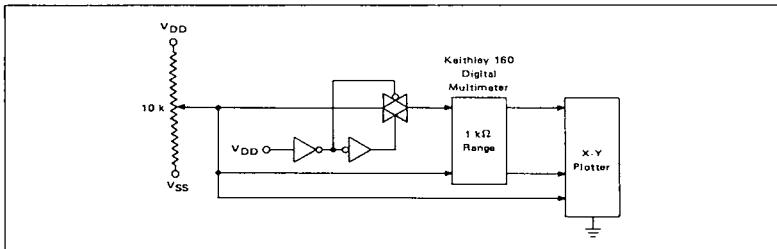
SWITCHING CHARACTERISTICS* ($C_L = 50 \text{ pF}$, $T_A = 25^\circ\text{C}$ unless otherwise noted.)

Characteristic	Symbol	V_{DD} Vdc	Min	Typ	Max	Unit
Propagation Delay Times $V_{SS} = 0 \text{ Vdc}$						
Input to Output ($R_L = 10 \text{ k}\Omega$) $t_{PLH}, t_{PHL} = [0.17 \text{ ns}/\mu\text{F}] C_L + 15.5 \text{ ns}$ $t_{PLH}, t_{PHL} = [0.08 \text{ ns}/\mu\text{F}] C_L + 6.0 \text{ ns}$ $t_{PLH}, t_{PHL} = [0.06 \text{ ns}/\mu\text{F}] C_L + 4.0 \text{ ns}$	t_{PLH}, t_{PHL}	5.0 10 15	— — —	20 10 7.0	40 20 15	ns
Control to Output ($R_L = 1 \text{ k}\Omega$) Output "1" to High Impedance	t_{PHZ}	5.0 10 15	— — —	40 35 30	80 70 60	ns
Output "0" to High Impedance	t_{PLZ}	5.0 10 15	— — —	40 35 30	80 70 60	ns
High Impedance to Output "1"	t_{PZH}	5.0 10 15	— — —	60 20 15	120 40 30	ns
High Impedance to Output "0"	t_{PZL}	5.0 10 15	— — —	60 20 15	120 40 30	ns
Sine Wave Distortion $V_{SS} = -5 \text{ Vdc}$ $(V_{in} = 1.77 \text{ Vdc, RMS Centered @ } 0.0 \text{ Vdc,}$ $R_L = 10 \text{ k}\Omega, I = 1.0 \text{ kHz})$	--	5.0	--	0.1	—	%
Frequency Response (Switch ON) $V_{SS} = -5 \text{ Vdc}$ $(R_L = 1 \text{ k}\Omega, 20 \log_{10} \frac{V_{out}}{V_{in}} = -3 \text{ dB})$	--	5.0	—	65	—	MHz
Feedthrough Attenuation (Switch OFF) $V_{SS} = -5 \text{ Vdc}$ $(R_L = 1 \text{ k}\Omega, 20 \log_{10} \frac{V_{out}}{V_{in}} = -50 \text{ dB})$	--	5.0	—	1.0	—	MHz
Crosstalk Between Any Two Switches $V_{SS} = -5 \text{ Vdc}$ $(R_L = 1 \text{ k}\Omega, 20 \log_{10} \frac{V_{out(B)}}{V_{in(A)}} \approx -50 \text{ dB,}$ $\text{Switch A ON, Switch B OFF})$	--	5.0	—	8.0	—	MHz
Crosstalk, Control Input to Signal Output $V_{SS} = -5 \text{ Vdc}$ Maximum Control Input Frequency $V_{SS} = 0 \text{ Vdc}$ $(20 \log_{10} \frac{V_{out}}{V_{in}} = -6 \text{ dB})$	-- --	5.0 5.0 10 15	— — — —	300 6.0 8.0 8.5	— — — —	mV MHz

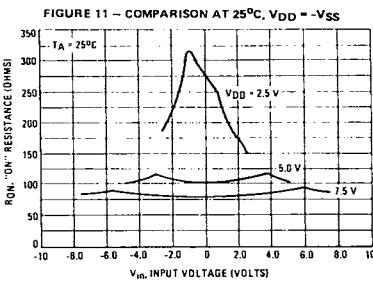
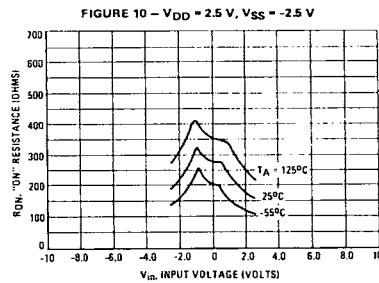
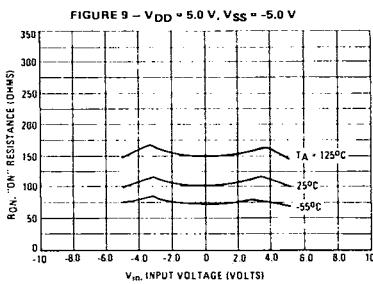
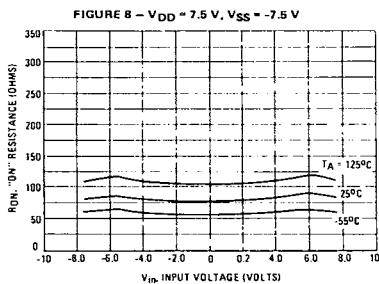
*The formulas given are for the typical characteristics only.

MC14066B

FIGURE 7 – CHANNEL RESISTANCE (R_{ON}) TEST CIRCUIT



TYPICAL RESISTANCE CHARACTERISTICS



MC14066B

TEST CIRCUITS

FIGURE 1 – INPUT VOLTAGE

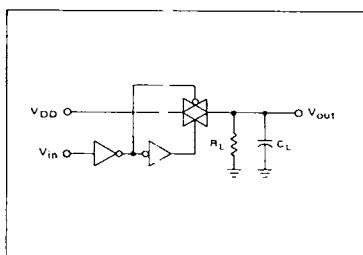


FIGURE 2 – PROPAGATION DELAY TIME,
CONTROL TO OUTPUT

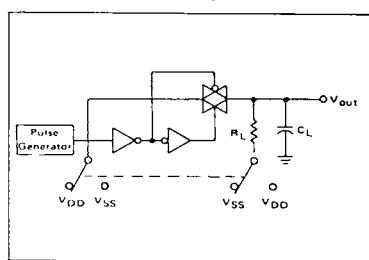


FIGURE 3 – BANDWIDTH AND
FEEDTHROUGH ATTENUATION

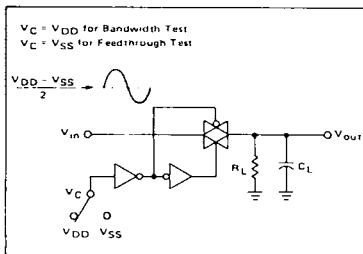


FIGURE 4 – CROSSTALK BETWEEN
ANY TWO SWITCHES

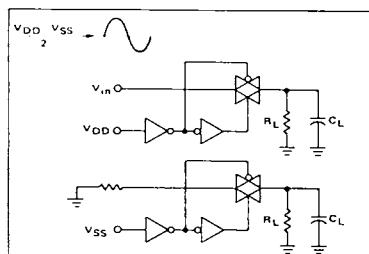


FIGURE 5 – CROSSTALK,
CONTROL TO OUTPUT

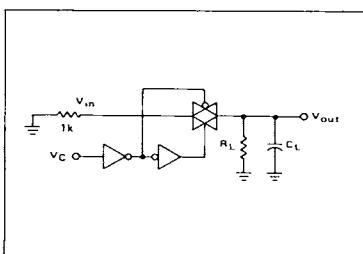
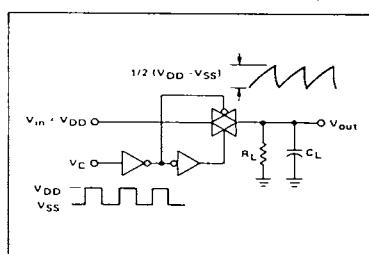


FIGURE 6 – MAXIMUM CONTROL FREQUENCY







MOTOROLA

MC14069UB

HEX INVERTER

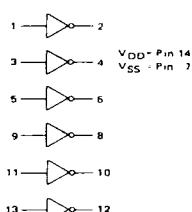
The MC14069UB hex inverter is constructed with MOS P-channel and N-channel enhancement mode devices in a single monolithic structure. These inverters find primary use where low power dissipation and/or high noise immunity is desired. Each of the six inverters is a single stage to minimize propagation delays.

- Quiescent Current = 0.5 nA typ/pkg @ 5 Vdc
- Noise Immunity = 45% of V_{DD} typ
- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Capable of Driving Two Low-Power TTL Loads, One Low-Power Schottky TTL Load or Two HTL Loads Over the Rated Temperature Range
- Double Diode Protection on All Inputs
- Pin-for-Pin Replacement for CD4069UB
- Meets JEDEC UB Specifications

MAXIMUM RATINGS (Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
DC Supply Voltage	V _{DD}	-0.5 to +18	Vdc
Input Voltage, All Inputs	V _{in}	0.5 to V _{DD} < 0.5	Vdc
DC Current Draw per Pin	I	10	mA/dc
Operating Temperature Range AL Device CL/CP Device	T _A	-55 to +125 -40 to +85	°C
Storage Temperature Range	T _{stg}	-65 to +150	°C

LOGIC DIAGRAM



CIRCUIT SCHEMATIC (1/6 OF CIRCUIT SHOWN)

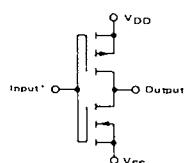
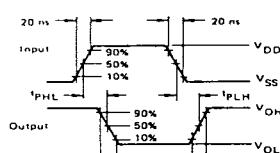
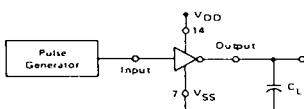


FIGURE 1 - SWITCHING TIME TEST CIRCUIT AND WAVEFORMS



CMOS SSI

(LOW POWER COMPLEMENTARY MOS)

HEX INVERTER



L SUFFIX
CLASIC PACKAGE
CASE 632



P SUFFIX
PLASTIC PACKAGE
CASE 646

ORDERING INFORMATION

MC14XXXUB	Suffix Denotes
L	Ceramic Package
P	Plastic Package
A	Extended Operating Temperature Range
C	Limited Operating Temperature Range

This device contains circuitry to protect the inputs against damage due to high static voltages or electrical fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltage to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range V_{SS} ≤ V_{in} or V_{out} ≤ V_{DD}.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

MC14069UB

ELECTRICAL CHARACTERISTICS

Characteristic	Symbol	V _{DD} Vdc	T _{low} *		25°C				T _{high} *		Unit
			Min	Max	Min	Typ	Max	Min	Max	Unit	
Output Voltage "0" Level V _{in} = V _{DD} or 0	V _{DL}	5.0	-	0.05	-	0	0.05	-	0.05	-	Vdc
	V _{DL}	10	-	0.05	-	0	0.05	-	0.05	-	
	V _{DL}	15	-	0.05	-	0	0.05	-	0.05	-	
	V _{DH}	5.0	4.95	-	4.95	5.0	-	4.95	-	-	
	V _{DH}	10	9.95	-	9.95	10	-	9.95	-	-	
	V _{DH}	15	14.95	-	14.95	15	-	14.95	-	-	
Input Voltage ^{**} (V _O = 4.5 Vdc) (V _O = 9.0 Vdc) (V _O = 13.5 Vdc)	V _{IL}	5.0	-	1.0	-	2.25	1.0	-	1.0	-	Vdc
	V _{IL}	10	-	2.0	-	4.50	2.0	-	2.0	-	
	V _{IL}	15	-	2.5	-	6.75	2.5	-	2.5	-	
	V _{IH}	5.0	4.0	-	4.0	2.75	-	4.0	-	-	
	V _{IH}	10	8.0	-	8.0	5.50	-	8.0	-	-	
	V _{IH}	15	12.5	-	12.5	8.25	-	12.5	-	-	
Output Drive Current (AL Device) (V _{OH} = 2.5 Vdc) Source (V _{OH} = 4.6 Vdc) (V _{OH} = 9.5 Vdc) (V _{OH} = 13.5 Vdc)	I _{OH}	5.0	-3.0	-	-2.4	-4.2	-	-1.7	-	-	mAdc
	I _{OH}	5.0	-0.64	-	-0.51	-0.88	-	-0.36	-	-	
	I _{OH}	10	-1.6	-	-1.3	-2.25	-	-0.9	-	-	
	I _{OH}	15	-4.2	-	-3.4	-8.8	-	-2.4	-	-	
	I _{OL}	5.0	0.64	-	0.51	0.88	-	0.36	-	-	
	I _{OL}	10	1.6	-	1.3	2.25	-	0.9	-	-	
Output Drive Current (CL/CP Device) (V _{OH} = 2.5 Vdc) Source (V _{OH} = 4.6 Vdc) (V _{OH} = 9.5 Vdc) (V _{OH} = 13.5 Vdc)	I _{OH}	5.0	-2.5	-	-2.1	-4.2	-	-1.7	-	-	mAdc
	I _{OH}	5.0	-0.52	-	-0.44	-0.88	-	-0.36	-	-	
	I _{OH}	10	-1.3	-	-1.1	-2.25	-	-0.9	-	-	
	I _{OH}	15	-3.6	-	-3.0	-8.8	-	-2.4	-	-	
	I _{OL}	5.0	0.52	-	0.44	0.88	-	0.36	-	-	
	I _{OL}	10	1.3	-	1.1	2.25	-	0.9	-	-	
Input Current (AL Device) I _{in}	I _{in}	15	-	:0.1	-	+0.00001	:0.1	-	+1.0	-	μAdc
	I _{in}	15	-	:0.3	-	+0.00001	:0.3	-	+1.0	-	
	C _{in}	-	-	-	-	5.0	7.5	-	-	-	
Quiescent Current (AL Device) (Per Package)	I _{DD}	5.0	-	0.25	-	0.0005	0.25	-	7.5	-	μAdc
	I _{DD}	10	-	0.50	-	0.0010	0.50	-	15	-	
	I _{DD}	15	-	1.00	-	0.0015	1.00	-	30	-	
Quiescent Current (CL/CP Device) (Per Package)	I _{DD}	5.0	-	1.0	-	0.0005	1.0	-	7.5	-	μAdc
	I _{DD}	10	-	2.0	-	0.0010	2.0	-	15	-	
	I _{DD}	15	-	4.0	-	0.0015	4.0	-	30	-	
Total Supply Current*** (Dynamic plus Quiescent, Per Gate) (C _L = 50 pF)	I _T	5.0	-	-	-	I _T = [0.3 μA/kHz] f + I _{DD} /6	-	-	-	-	μAdc
	I _T	10	-	-	-	I _T = [0.6 μA/kHz] f + I _{DD} /6	-	-	-	-	
	I _T	15	-	-	-	I _T = [0.9 μA/kHz] f + I _{DD} /6	-	-	-	-	
Output Rise and Fall Times** (C _L = 50 pF) t _{TLH} , t _{THL} = (1.35 ns/pF) C _L + 33 ns t _{TLH} , t _{THL} = (0.60 ns/pF) C _L + 20 ns t _{TLH} , t _{THL} = (0.40 ns/pF) C _L + 20 ns	t _{TLH} , t _{THL}	5.0	-	-	-	100	200	-	-	-	ns
	t _{TLH} , t _{THL}	10	-	-	-	50	100	-	-	-	
	t _{TLH} , t _{THL}	15	-	-	-	40	80	-	-	-	
Propagation Delay Times** (C _L = 50 pF) t _{PLH} , t _{PHL} = (0.90 ns/pF) C _L + 20 ns t _{PLH} , t _{PHL} = (0.36 ns/pF) C _L + 22 ns t _{PLH} , t _{PHL} = (0.26 ns/pF) C _L + 17 ns	t _{PLH} , t _{PHL}	5.0	-	-	-	65	125	-	-	-	ns
	t _{PLH} , t _{PHL}	10	-	-	-	40	75	-	-	-	
	t _{PLH} , t _{PHL}	15	-	-	-	30	55	-	-	-	

* T_{low} = -55°C for AL Device, -40°C for CL/CP Device.

** T_{high} = +125°C for AL Device, +85°C for CL/CP Device.

*** To calculate total supply current at loads other than 50 pF:

$$I_T(C_L) = I_T(50 \text{ pF}) + 6 \times 10^{-3}(C_L - 50) V_{DD}$$

where: I_T is in μA (per package), C_L in pF, V_{DD} in Vdc, and f kHz is input frequency.

** The formulas given are for the typical characteristics only at 25°C.

#Noise immunity specified for worst case input combination.

Noise Margin for both "1" and "0" level -

$$0.5 \text{ Vdc min} @ V_{DD} = 5.0 \text{ Vdc}$$

$$1.0 \text{ Vdc min} @ V_{DD} = 10 \text{ Vdc}$$

$$1.0 \text{ Vdc min} @ V_{DD} = 15 \text{ Vdc}$$

CUPL Version 2.10a Serial# 2-00003-187
Copyright (c) 1983,84,85,86 Personal CAD Systems, Inc.
CREATED Thu May 07 16:34:29 1987

LISTING FOR LOGIC DESCRIPTION FILE: u7.pld

```
1:          PARTNO      U7;
2:          NAME        U7;
3:          DATE        05/04/87 ;
4:          REV         01 ;
5:          DESIGNER    ;
6:          COMPANY     Tandy ;
7:          ASSEMBLY   XXXX ;
8:          LOCATION    YYYY ;
9:
10:         ****
11:/* This device performs buffer control for the system data bus */
12:/* on the Model 4000. */
13:/*
14:/*
15:/*
16:/*
17:/* Allowable Target Device Types: 82S153A */
18:/*
19:/*
20:/** Inputs */
21:
22:PIN 1      = LCSROM    ;      /* Rom Select           */
23:PIN 2      = SDIR       ;      /* System Bus Direction */
24:PIN 3      = 287CS     ;      /* 287 Select           */
25:PIN 4      = HLDAL      ;      /* Hold Acknowledge    */
26:PIN 5      = ATEN       ;      /* AT Enable             */
27:PIN 6      = AEN8       ;      /* 8 Bit Enable          */
28:PIN 7      = XBHE       ;      /* Bus High Enable      */
29:PIN 8      = MEMCS16   ;      /* 16 Bit Enable          */
30:PIN 9      = XIOR       ;      /* I/O READ              */
31:
32:/** Outputs */
33:
34:PIN 17     = ENHLB      ;      /* Low Byte Enable      */
35:PIN 18     = SBUSEN     ;      /* System Bus Enable    */
36:PIN 19     = DIR245     ;      /* System Buffer Direction */
37:
38:
39:/** Logic Equations */
40:
41:DIR245    =      (SDIR # !LCSROM # (!287CS & !XIOR))
42:          & (AEN8 # XBHE # MEMCS16) ;
43:
44:SBUSEN   =
45:          ATEN & !HLDAL ;
46:
47:ENHLB    =
48:          AEN8 # XBHE # MEMCS16 ;
49:
```

Jedec Fuse Checksum (5332)
Jedec Transmit Checksum (0219)

CUPL Version 2.10a Serial# 2-00003-187
Copyright (c) 1983,84,85,86 Personal CAD Systems, Inc.
CREATED Wed May 20 14:20:24 1987

LISTING FOR LOGIC DESCRIPTION FILE: U9.PLD.pld

```
1:          PARTNO      U9;
2:          NAME        U9;
3:          DATE        04/16/87 ;
4:          REV         A ;
5:          DESIGNER    Tandy ;
6:          COMPANY    XXXX ;
7:          ASSEMBLY   XXXX ;
8:          LOCATION   YYYY ;
9:
10:         ****
11:/* This device performs buffer control between the X data bus
12:   and the S data bus on the Tandy 4000. Also this provides
13:   initial FDC port decoding.
14:   Allowable Target Device Types: PLS153A
15:   ****
16:   ****
17:   ****
18:   ****
19:
20:/** Inputs */
21:
22:PIN 1      = !DACK2 ;      /* DMA ACK FDC           */
23:PIN 2      = AEN ;        /* Acknowledge           */
24:PIN 3      = IXIOR ;     /*                         */
25:PIN 4      = !XMEMR ;    /*                         */
26:PIN 5      = XDIR ;      /* Output of 82A303       */
27:PIN 7      = !FDCSEL2 ;  /* FDC2 SELECT           */
28:PIN 8      = !FDCSELL ; /* FDC1 SELECT           */
29:PIN 9      = XA4 ;        /*                         */
30:PIN 11     = XA5 ;        /*                         */
31:PIN 12     = XA6 ;        /*                         */
32:PIN 13     = XA7 ;        /*                         */
33:PIN 14     = XA8 ;        /*                         */
34:PIN 15     = XA9 ;        /*                         */
35:
36:/** Outputs */
37:
38:PIN 17     = !FDCSEL ;   /* FDC ADDRESS DECODE    */
39:PIN 18     = !XBUSEN ;   /* X DATA BUS ENABLE      */
40:PIN 19     = XBUSDIR ;  /* X DATA BUS DIRECTION   */
41:
42:/** Declarations and Intermediate Variable Definitions */
43:
44:/** Logic Equations */
45:
46:FDCSEL =    !AEN & XA9 & XA8 & XA7 & XA6 & XA5 & XA4 & FDCSELL1 & !FDCSEL2
47:          # !AEN & XA9 & XA8 & !XA7 & XA6 & XA5 & XA4 & !FDCSELL1 & FDCSEL2 ;
48:
```

49:XBUSDIR = XDIR ;
50:
51:XBUSEN = !AEN ;
52:
53:
54:

Jedec Fuse Checksum (222D)
Jedec Transmit Checksum (9F35)

CUPL Version 2.10a Serial# 2-00003-187
Copyright (c) 1983,84,85,86 Personal CAD Systems, Inc.
CREATED Wed May 20 11:55:13 1987

LISTING FOR LOGIC DESCRIPTION FILE: u10.pld

```
1:          PARTNO    U10;
2:          NAME      U10;
3:          DATE      04/16/87 ;
4:          REV       A ;
5:          DESIGNER   ;
6:          COMPANY    Tandy ;
7:          ASSEMBLY   XXXX ;
8:          LOCATION    YYYY ;
9:
10:
11: ****
12: /*
13: /* This device performs address decoding for the FDC circuitry. */
14: /* on the TANDY 4000 . */
15: /*
16: ****
17: /* Allowable Target Device Types: PLS153A */
18: ****
19:
20:/** Inputs */
21:
22:PIN 1      = XA0      ;      /* */
23:PIN 2      = XA1      ;      /* */
24:PIN 3      = XA2      ;      /* */
25:PIN 4      = XA3      ;      /* */
26:PIN 7      = ALE301   ;      /* ALE FROM 82C301 */
27:PIN 8      = !REF     ;      /* REFRESH */
28:PIN 9      = !XIOR    ;      /* */
29:PIN 19     = !FDCSEL  ;      /* FDC ADDRESS DECODE */
30:PIN 16     = DCHNG    ;      /* DISK CHANGE STATUS */
31:
32:/** Outputs */
33:
34:PIN 11     = ALE306   ;      /* ALE OUTPUT TO 82C306 */
35:PIN 12     = SDB07    ;      /* */
36:PIN 15     = !LDCR    ;      /* LOAD CONTROL REGISTER @ 3F7h */
37:PIN 14     = !LDOR    ;      /* LOAD OPERATION REGISTER @ 3F2h */
38:PIN 13     = !CS      ;      /* WD37C65 CHIP SELECT */
39:PIN 18     = !OEN     ;      /* DATA BIT 7 ENABLE @ 3F7h */
40:
```

```
41:/** Declarations and Intermediate Variable Definitions */
42:
43:FIELD PORT    = [XA3..XA0] ;
44:
45:/** Logic Equations **/
46:
47:CS =  PORT:[4..5] & FDCSEL ;
48:
49:LDOR = PORT:[2] & FDCSEL ;
50:
51:LDCR = PORT:[7] & FDCSEL ;
52:
53:SDB07 = !DCHNG ;
54:
55:OEN =  PORT:[7] & FDCSEL & XIOR ;
56:
57:SDB07.OE = OEN ;
58:
59:ALE306 = ALE301 # REF ;
60:
61:
62:
```

Jedec Fuse Checksum (3FB3)
Jedec Transmit Checksum (E3C6)

CUPL Version 2.10a Serial# 2-00003-187
Copyright (c) 1983,84,85,86 Personal CAD Systems, Inc.
CREATED Fri Jun 05 11:45:47 1987

LISTING FOR LOGIC DESCRIPTION FILE: u22a.pld

```
1:  
2:          PARTNO      U22;  
3:          NAME        U22a;  
4:          DATE        06/05/87 ;  
5:          REV         02 ;  
6:          DESIGNER    ;  
7:          COMPANY     Tandy ;  
8:          ASSEMBLY   XXXX ;  
9:          LOCATION    YYYY ;  
10:  
11:/*************************************************************/*/  
12:/*  
13:/* This device performs 80287 co-processor control for the  
14:/* Tandy 4000.  
15:/*  
16:/*************************************************************/*/  
17:/* Allowable Target Device Types: PLS173  
18:/*************************************************************/*/  
19:  
20:/** Inputs **/  
21:  
22:PIN 1      = XA00      ;      /*  
23:PIN 2      = XA03      ;      /*  
24:PIN 3      = 287CS    ;      /* 287 DECODE  
25:PIN 4      = RESET4   ;      /* PERIPHERAL RESET  
26:PIN 5      = XIOW      ;      /*  
27:PIN 6      = COPEN     ;      /* 287 ENABLED  
28:PIN 7      = !MCS      ;      /* WEITEK CHIP SELECT  
29:PIN 8      = INTR      ;      /* WEITEK INTERRUPT  
30:PIN 9      = !PRES     ;      /* WEITEK PRESENT  
31:PIN 10     = 287ERR    ;      /* 287 ERROR  
32:PIN 11     = 287BSY    ;      /* 287 BUSY  
33:PIN 13     = 287PER    ;      /* 287 PEREQ  
34:  
35:/** Outputs **/  
36:  
37:PIN 14     = 287RESET  ;      /* 287 RESET  
38:PIN 15     = !NPCS     ;      /* 287 CHIP SELECT  
39:PIN 16     = !AF32EN   ;      /* INTERNAL AF32 ENABLE  
40:PIN 17     = IRQ13    ;      /* INTERRUPT REQUEST 13  
41:PIN 18     = !386BSY   ;      /* 80386 BUSY  
42:PIN 19     = !IOCS16   ;      /*  
43:PIN 20     = 386PER    ;      /* 80386 PEREQ  
44:PIN 21     = ERR1      ;      /* INTERNAL  
45:PIN 22     = DELERR    ;      /* INTERNAL  
46:PIN 23     = !AF32    ;      /* 32 BIT DEVICE  
47:  
48:
```

```
49:  
50:  
51:/** Logic Equations **/  
52:  
53:ERR1      =      1287ERR & !COPEN ;  
54:  
55:DELERR    =      1287ERR & ERR1 & !COPEN ;  
56:  
57:386PER    =      287PER & !COPEN  
58:           # IRQ13 ;  
59:  
60:287RESET=   XA00 & !XA03 & !287CS & !XIOW  
61:           # RESET4 ;  
62:  
63:NPCS      =      1287CS & XA03 & !IRQ13 ;  
64:  
65:386BSY    =      1287BSY & !COPEN  
66:           # IRQ13 & !COPEN  
67:           # NPCS & COPEN ;  
68:  
69:IOCS16    =      1287CS & XA03 & !IRQ13 ;  
70:  
71:IOCS16.OE =      NPCS ;  
72:  
73:IRQ13     =      1287BSY & !1287ERR & !DELERR & XA03 & !RESET4 & !COPEN  
74:           # 1287BSY & !1287ERR & !DELERR & XIOW & !RESET4 & !COPEN  
75:           # 1287BSY & !1287ERR & !DELERR & 287CS & !RESET4 & !COPEN  
76:           # IRQ13 & !RESET4 & XA03 & !COPEN  
77:           # IRQ13 & !RESET4 & XIOW & !COPEN  
78:           # IRQ13 & !RESET4 & 287CS & !COPEN ;  
79:  
80:  
81:  
82:
```

Jedec Fuse Checksum (8E81)
Jedec Transmit Checksum (80BD)

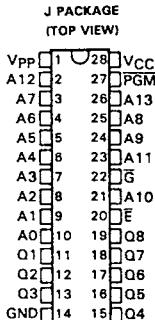
TMS27C128

131,072-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY

OCTOBER 1984 — REVISED NOVEMBER 1985

- Organization . . . 16K × 8
- Single 5-V Power Supply
- Pin Compatible with Existing 64K and 128K EPROMs
- All Inputs/Outputs Fully TTL Compatible
- Max Access/Min Cycle Time

'27C128-1, '27C128-15	150 ns
'27C128-2, '27C128-20	200 ns
'27C128, '27C128-25	250 ns
'27C128-3, '27C128-30	300 ns
'27C128-4, '27C128-45	450 ns
- HVC MOS Technology
- 3-State Output Buffers
- 400 mV Guaranteed DC Noise Immunity with Standard TTL Loads
- Low Power Dissipation ($V_{CC} = 5.25$ V)
 - Active . . . 210 mW Worst Case
 - Standby . . . 1.4 mW Worst Case (CMOS-Input Levels)



PIN NOMENCLATURE	
A0-A13	Address Inputs
E	Chip Enable/Power Down
G	Output Enable
GND	Ground
PGM	Program
Q1-Q8	Outputs
VCC	5-V Power Supply
V _{PP}	12.5-V Power Supply

6.

EPROMs/PROMs

description

The TMS27C128 series are 131,072-bit, ultraviolet-light erasable, electrically programmable read-only memories. These devices are fabricated using HVC MOS technology for high speed and simple interface with MOS and bipolar circuits. All inputs (including program data inputs) can be driven by Series 74 TTL circuits without the use of external pull-up resistors, and each output can drive one Series 74 TTL circuit without external resistors. The data outputs are three state for connecting multiple devices to a common bus. The TMS27C128 is pin compatible with existing 28-pin ROMs and EPROMs. It is offered in a dual-in-line ceramic package (J suffix) rated for operation from 0°C to 70°C.

Since these EPROMs operate from a single 5-V supply (in the read mode), they are ideal for use in microprocessor-based systems. One other (12.5 V) supply is needed for programming, but all programming signals are TTL level. For programming outside the system, existing EPROM programmers can be used. Locations may be programmed singly, in blocks, or at random.

operation

There are seven modes of operation for the TMS27C128 listed on the following page. Read mode requires a single 5-V supply. All inputs are TTL level except for V_{PP} during programming (12.5 V) and 12 V on A9 for signature mode.

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS
INSTRUMENTS
POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

Copyright © 1985, Texas Instruments Incorporated

TMS27C128

131,072-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY

FUNCTION (PINS)	MODE						
	Read	Output Disable	Standby	Programming	Verify	Program Inhibit	Signature Mode
\bar{E} (20)	V_{IL}	V_{IL}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{IL}
\bar{G} (22)	V_{IL}	V_{IH}	X [†]	V_{IH}	V_{IL}	X	V_{IL}
\bar{PGM} (27)	V_{IH}	V_{IH}	X	V_{IL}	V_{IH}	X	V_{IH}
V_{PP} (1)	V_{CC}	V_{CC}	V_{CC}	V_{PP}	V_{PP}	V_{PP}	V_{CC}
V_{CC} (28)	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
A_9 (24)	X	X	X	X	X	X	V_H^{\ddagger} V_H^{\ddagger}
A_0 (10)	X	X	X	X	X	X	V_{IL} V_{IH}
Q1-Q8 (11-13, 15-19)	D _{OUT}	HI-Z	HI-Z	D _{IN}	D _{OUT}	HI-Z	CODE MFG DEVICE 97 83

[†]X can be V_{IL} or V_{IH} .

[‡] $V_H = 12 \text{ V} \pm 0.5 \text{ V}$.

6

read/output disable

When the outputs of two or more TMS27C128's are connected in parallel on the same bus, the output of any particular device in the circuit can be read with no interference from the competing outputs of the other devices. To read the output of the TMS27C128, a low-level signal is applied to the \bar{E} and \bar{G} pins. All other devices in the circuit should have their outputs disabled by applying a high-level signal to one of these pins. Output data is accessed at pins Q1 to Q8.

power down

Active I_{CC} current can be reduced from 40 mA to 500 μA (TTL-level inputs) or 250 μA (CMOS-level inputs) by applying a high TTL signal to the \bar{E} pin. In this mode all outputs are in the high-impedance state.

erasure

Before programming, the TMS27C128 is erased by exposing the chip through the transparent lid to high intensity ultraviolet light (wavelength 2537 angstroms). The recommended minimum exposure dose (UV intensity \times exposure time) is fifteen watt-seconds per square centimeter. A typical 12 milliwatt per square centimeter, filterless UV lamp will erase the device in 21 minutes. The lamp should be located about 2.5 centimeters above the chip during erasure. After erasure, all bits are in the high state. It should be noted that normal ambient light contains the correct wavelength for erasure. Therefore, when using the TMS27C128, the window should be covered with an opaque label.

fast programming

After erasure (all bits are in logic '1' state), logic '0's are programmed into the desired locations. A programmed '0' can only be erased by ultraviolet light. Data is presented in parallel (eight bits) on pins Q1 to Q8. Once addresses and data are stable, PGM is pulsed. The programming mode is achieved when $V_{PP} = 12.5 \text{ V}$, $\bar{PGM} = V_{IL}$, $V_{CC} = 6.0 \text{ V}$, $\bar{G} = V_{IH}$, and $\bar{E} = V_{IL}$. More than one TMS27C128 can be programmed when the devices are connected in parallel. Locations can be programmed in any order.

TMS27C128
131,072-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY

Programming uses two types of programming pulses: Prime and Final. The length of the Prime pulse is 1 millisecond; this pulse is applied X times. After each Prime pulse, the byte being programmed is verified. If the correct data is read, the Final programming pulse is applied; if correct data is not read, an additional 1 millisecond pulse is applied up to a maximum X of 25. The Final programming pulse is 3X long. This sequence of programming and verification is performed at $V_{CC} = 6.0$ V and $V_{PP} = 12.5$ V. When the full fast programming routine is complete, all bits are verified with $V_{CC} = V_{PP} = 5$ V (see Figure 1).

program inhibit

Programming may be inhibited by maintaining a high level input on the \bar{E} pin or \overline{PGM} pin.

program verify

Programmed bits may be verified with $V_{PP} = 12.5$ V when $\bar{G} = V_{IL}$, $\bar{E} = V_{IL}$, and $\overline{PGM} = V_{IH}$.

signature mode

The signature mode provides access to a binary code identifying the manufacturer and type. This mode is activated when A9 (pin 24) is forced to $12\text{V} \pm 0.5$ V. Two identifier bytes are accessed by A0 (pin 10) i.e., A0 = V_{IL} – manufacturer; A0 = V_{IH} – device. All other addresses must be held at V_{IL} . Each byte possesses odd parity on bit Q8. The manufacturer code for this device is 97, and the device code is 83.

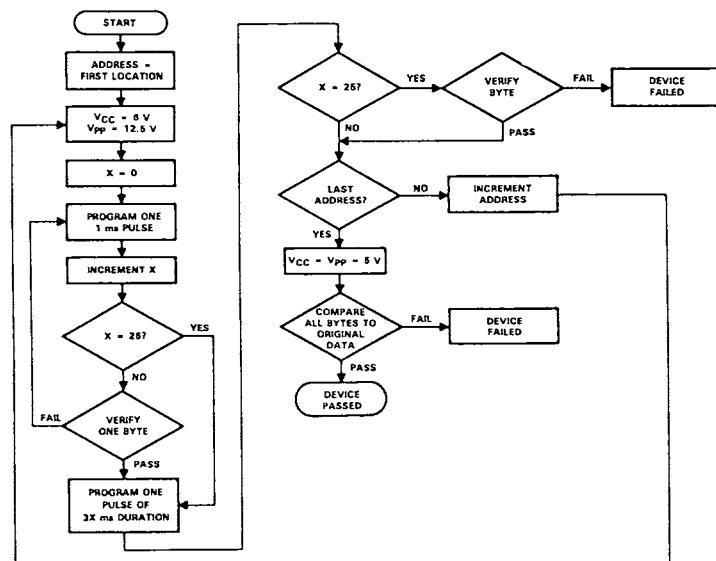
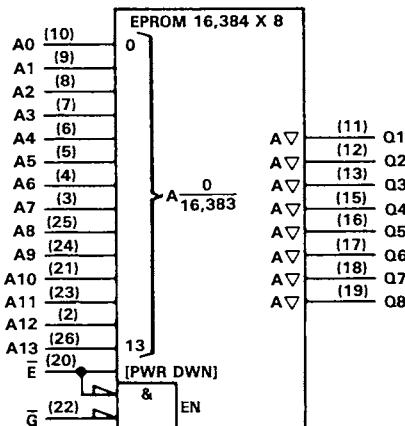


FIGURE 1. FAST PROGRAMMING FLOWCHART

TMS27C128
131,072-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY

logic symbol†



6

†This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage range, V _{CC} (see Note 1).....	-0.6 V to 7 V
Supply voltage range, V _{PP} (see Note 1).....	-0.6 V to 14 V
Input voltage range (see Note 1): All inputs except A9.....	-0.6 V to 6.5 V
A9.....	-0.6 V to 13.5 V
Output voltage range (see Note 1).....	-0.6 V to V _{CC} + 1 V
Operating free-air temperature range.....	0°C to 70°C
Storage temperature range.....	-65°C to 150°C

†Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: Under absolute maximum ratings, voltage values are with respect to GND.

TMS27C128

131,072-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY

recommended operating conditions

			TMS27C128-1	TMS27C128-15	UNIT	
			TMS27C128-2	TMS27C128-20		
			TMS27C128	TMS27C128-25		
			TMS27C128-3	TMS27C128-30		
			TMS27C128-4	TMS27C128-45		
			MIN	NOM	MAX	UNIT
V _{CC}	Supply voltage (see Note 2)		4.75	5	5.25	
V _{PP}	Supply voltage (see Note 3)			V _{CC}		
V _{IH}	High-level input voltage	TTL	2	V _{CC} +1	2	
		CMOS	V _{CC} -0.2	V _{CC} +0.2	V _{CC} -0.2	
V _{IL}	Low-level input voltage	TTL	-0.5	0.8	-0.5	
		CMOS	GND-0.2	GND+0.2	GND-0.2	
T _A	Operating free-air temperature		0	70	0	°C

NOTES: 2. V_{CC} must be applied before or at the same time as V_{PP} and removed after or at the same time as V_{PP}. The device must not be inserted into or removed from the board when V_{PP} or V_{CC} is applied.

3. V_{PP} can be connected to V_{CC} directly (except in the program mode). V_{CC} supply current in this case would be I_{CC}+I_{PP}. During programming, V_{PP} must be maintained at 12.5 V ($\pm 0.5\text{V}$).

electrical characteristics over full ranges of recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP [†]	MAX	UNIT
V _{OH}	High-level output voltage	I _{OH} = -400 μA	2.4			V
V _{OL}	Low-level output voltage	I _{OL} = 2.1 mA		0.4		V
I _I	Input current (leakage)	V _I = 0 V to 5.5 V		± 10		μA
I _O	Output current (leakage)	V _O = 0 V to V _{CC}		± 10		μA
I _{PP1}	V _{PP} supply current	V _{PP} = V _{CC} = 5.5 V		100		μA
I _{PP2}	V _{PP} supply current (during program pulse)	V _{PP} = 13 V		30	50	mA
I _{CC1}	V _{CC} supply current (standby)	V _{CC} = 5.5 V, E = V _{IH}		500		μA
		V _{CC} = 5.5 V, E = V _{CC}		250		μA
I _{CC2}	V _{CC} supply current (active)	V _{CC} = 5.5 V, E = V _{IL} , t _{cycle} = minimum cycle time, outputs open		30	40	mA

[†]Typical values are at T_A = 25°C and nominal voltages.

capacitance over recommended supply voltage range and operating free-air temperature range, f = 1 MHz[†]

PARAMETER		TEST CONDITIONS	MIN	TYP [‡]	MAX	UNIT
C _i	Input capacitance	V _I = 0 V, f = 1 MHz		6	9	pF
C _o	Output capacitance	V _O = 0 V, f = 1 MHz		8	12	pF

[†]Capacitance measurements are made on sample basis only.

[‡]Typical values are at T_A = 25°C and nominal voltages.

TMS27C128

131,072-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY

switching characteristics over full ranges of recommended operating conditions (see Note 4)

PARAMETER	TEST CONDITIONS (SEE NOTES 4 AND 5)	'27C128-1		'27C128-2		'27C128		UNIT	
		'27C128-15		'27C128-20		'27C128-25			
		MIN	MAX	MIN	MAX	MIN	MAX		
$t_{\text{a}}(\text{A})$	Access time from address		150		200		250	ns	
$t_{\text{a}}(\text{E})$	Access time from chip enable		150		200		250	ns	
$t_{\text{en}}(\text{G})$	Output enable time from $\overline{\text{G}}$		75		75		100	ns	
t_{dis}	Output disable time from $\overline{\text{G}}$ or E , whichever occurs first [†]		0	60	0	60	0	ns	
$t_{\text{v(A)}}$	Output data valid time after change of address, E , or $\overline{\text{G}}$, whichever occurs first [†]		0		0		0	ns	

PARAMETER	TEST CONDITIONS (SEE NOTES 4 AND 5)	'27C128-3		'27C128-4		UNIT	
		'27C128-30		'27C128-45			
		MIN	MAX	MIN	MAX		
$t_{\text{a}}(\text{A})$	Access time from address		300		450	ns	
$t_{\text{a}}(\text{E})$	Access time from chip enable		300		450	ns	
$t_{\text{en}}(\text{G})$	Output enable time from $\overline{\text{G}}$		120		150	ns	
t_{dis}	Output disable time from $\overline{\text{G}}$ or E , whichever occurs first [†]		0	105	0	130	ns
$t_{\text{v(A)}}$	Output data valid time after change of address, E , or $\overline{\text{G}}$, whichever occurs first [†]		0		0	ns	

[†]Value calculated from 0.5 V delta to measured output level. This parameter is only sampled and not 100% tested.

recommended timing requirements for programming, $T_A = 25^\circ\text{C}$, $V_{CC} = 6\text{ V}$, $V_{PP} = 12.5\text{ V}$ (see Note 4)

		MIN	NOM	MAX	UNIT	
$t_{\text{w}}(\text{IPGM})$	Initial program pulse duration		0.95	1	1.05	ms
$t_{\text{w}}(\text{FGPM})$	Final pulse duration		2.85		78.75	ms
$t_{\text{su}}(\text{A})$	Address setup time		2		μs	
$t_{\text{su}}(\text{E})$	E setup time		2		μs	
$t_{\text{su}}(\text{G})$	$\overline{\text{G}}$ setup time		2		μs	
$t_{\text{dis}}(\text{G})$	Output disable time from $\overline{\text{G}}$		0	130	ns	
$t_{\text{en}}(\text{G})$	Output enable time from $\overline{\text{G}}$			150	ns	
$t_{\text{su}}(\text{D})$	Data setup time		2		μs	
$t_{\text{su}}(\text{VPP})$	V_{PP} setup time		2		μs	
$t_{\text{su}}(\text{VCC})$	V_{CC} setup time		2		μs	
$t_{\text{h}}(\text{A})$	Address hold time		0		μs	
$t_{\text{h}}(\text{D})$	Data hold time		2		μs	

NOTES: 4. For all switching characteristics and timing measurements, input pulse levels are 0.40 V to 2.4 V and $V_{PP} = 12.5\text{ V} \pm 0.5\text{ V}$ during programming.

5. Common test conditions apply for $t_{\text{dis}}(\text{G})$ except during programming.

TMS27C128

131,072-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY

PARAMETER MEASUREMENT INFORMATION

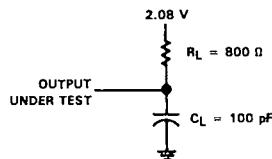
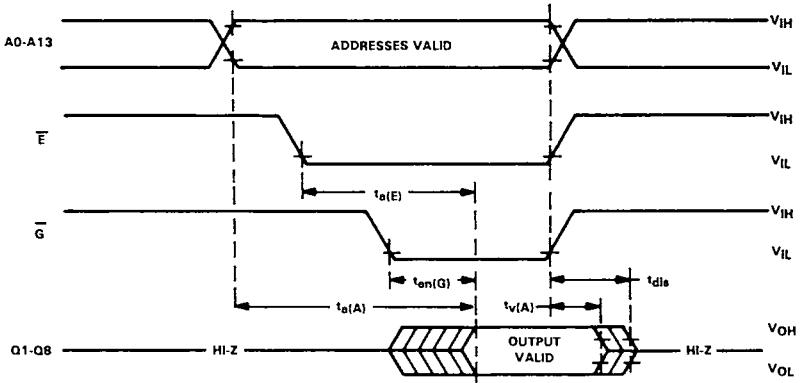


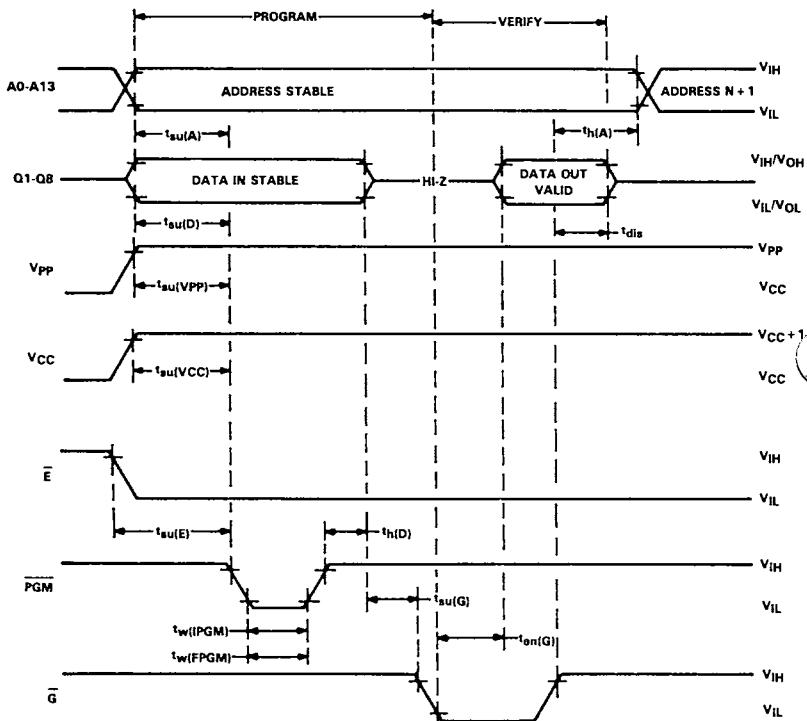
FIGURE 2. OUTPUT LOAD CIRCUIT

read cycle timing



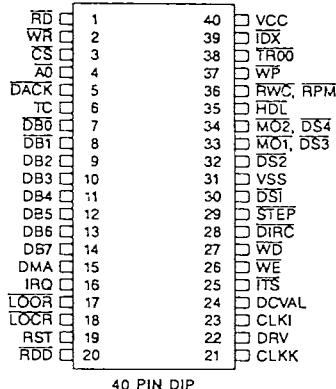
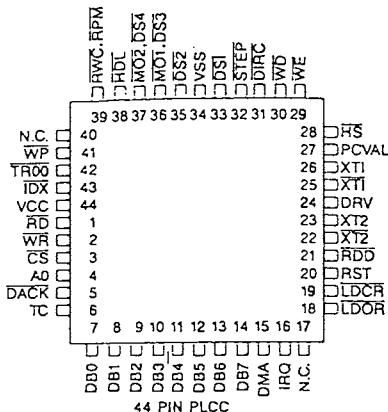
TMS27C128
131,072-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY

program cycle timing



WESTERN DIGITAL CORPORATION

WD37C65 FLOPPY DISK SUBSYSTEM CONTROLLER



FEATURES

- IBM PC/AT COMPATIBLE FORMAT (SINGLE AND DOUBLE DENSITY)
 - FLOPPY CONTROL AND OPERATIONS ON CHIP
 - IN PC/AT MODE, PROVIDES REQUIRED SIGNAL QUALIFICATION TO DMA CHANNEL
 - BIOS COMPATIBLE
 - DUAL SPEED SPINDLE DRIVE SUPPORT
- ADDRESS MARK DETECTION CIRCUITRY INTERNAL TO FLOPPY DISK CONTROLLER
- MULTISECTOR AND MULTITRACK TRANSFER CAPABILITY
- DIRECT FLOPPY DISK DRIVE INTERFACE WITH NO BUFFERS NEEDED
 - 48mA SINK OUTPUT DRIVERS
 - SCHMITT TRIGGER LINE RECEIVERS
- COMPATIBLE WITH PD8080/85, PD8086, AND PD780 (Z80 (TM)) MICROPROCESSORS
- ON CHIP CLOCK GENERATION
- TWO TTL CLOCK INPUTS FOR 40 PIN DIP
- TWO XTAL OSCILLATOR CIRCUITS FOR 44 PIN PLCC
- AUTOMATIC WRITE PRECOMPENSATION
- INNER TRACK VALUE OF 125 OR 167NS PIN SELECTABLE
- ENHANCED HOST INTERFACE
 - READWRITE ACCESSES COMPATIBLE REGISTERS WITH 8 OR 12 MHz 286 MICROPROCESSOR WITH 0 WAIT STATES
 - 20 LS-TTL OUTPUT DRIVE CAPABILITY
 - INPUTS ARE TTL LEVEL SCHMITT TRIGGER (EXCEPT DATA BUS)
 - DMA TIMING CORRECTED
- USER PROGRAMMABLE TRACK STEPPING RATE AND HEAD LOAD/UNLOAD TIME
- DRIVES UP TO FOUR FLOPPY OR MICRO FLOPPY DISK (TM) DRIVES
- DATA TRANSFER IN DMA OR NON-DMA MODE
- PARALLEL SEEK OPERATIONS
 - ON UP TO FOUR DRIVES
- HIGH PERFORMANCE, CLASSICAL 2ND ORDER, TYPE 2, PHASE LOCKED LOOP DIGITAL DATA SEPARATOR
 - 125, 250, 300, 500 KBITS/SEC DATA RATES
 - CMOS LOW POWER 125mW
 - +5VDC POWER SUPPLY

DESCRIPTION

The WD37C65 Floppy Disk Subsystem Controller is an LSI device that provides all the needed functionality between the Host processor peripheral bus and the cable connector to the floppy disk drive. This 'superchip' integrates: Formatter/Controller, Data Separation, Write Precompensation, Data Rate Selection, Clock Generation, Drive Interface Drivers and Receivers.

On the disk drive interface, the WD37C65 includes Data Separation that has been designed to address high performance error rates on floppy disk drives, and contains all the necessary logic to achieve classical 2nd order, type 2, phase locked loop performance. Write Precompensation is included, in addition to the usual Formatting, Encoding/Decoding, Stepper Motor Control, and Status Sensing Functions. All inputs are TTL compatible Schmitt Trigger line receivers, and outputs are high current, open drain, with the drivers meeting the 48 mA ANSI Specification.

The Host interface has been improved for speed operation supporting 8 or 12 MHz, 286 microprocessor bus without the use of wait states, the inputs are Schmitt Triggers (except the data bus). Output drive capability is 20 LS-TTL loads, allowing direct interconnection to bus structures without the use of buffers or transceivers. For PC and

PC/AT applications, qualification of interrupt request and DMA request is provided.

Traditionally, Data Rate Selection, Drive Selection, and Stepper Motor Control have been output ports of the Host processor architecture. In the PC/AT these functions are latched into registers addressed within the I/O mapping of the system. The Control Register provides support logic that latches the two LSBs used to select the desired data rate that controls internal clock generation. The Operations Register replaces the standard latched port used in floppy subsystems. These registers are incorporated into the WD37C65.

All Clock Generation: SCLK – Sampling Clock, WCLK – Write Clock, and MCLK – Master Clock, are included in the WD37C65. XTAL oscillator circuits provide the necessary signals for internal timing when using the 44 pin PLCC. If the 40 pin DIP is used, the TTL level clock inputs must be provided. There are two oscillators in the WD37C65; one at 16 MHz that handles all standard data rates (500, 250, and 125 Kb/Sec). The other oscillator is at 9.6 MHz to support the 300 Kb/Sec data rate used in PC/AT designs. Some AT compatibles use two-speed disk drives. If two-speed disk drives are used, the drive type input pin is grounded and only the 16 MHz clock is required. A RPM output signal will become active when a spindle speed rate of 300 RPM is desired.

PIN DESCRIPTIONS

D/P PIN NUMBER	MNEMONIC	SIGNAL NAME	I/O	FUNCTION
1/1	RD	READ	I	Control signal for transfer of data or status onto the data bus by the WD37C65.
2/2	WR	WRITE	I	Control signal for latching data from the bus into the WD37C65 Buffer Register.
3/3	CS	CHIP SELECT	I	Selected when 0 (low) allowing RD or WR operation from the Host.
4/4	A0	ADDRESS LINE	I	Address line selecting data (-1) or status (-0) information. (A0 = logic 0 during WR is illegal).
5/5	DACK	DMA ACKNOWLEDGE	I	Used by the DMA controller to transfer data from the WD37C65 onto the bus. Logical equivalent to CS and A0-1. In Special or PC/AT Mode, this signal is qualified by DMAEN from the Operations Register.
6/6	TC	TERMINAL COUNT	I	This signal indicates to WD37C65 that data transfer is complete. If DMA operational mode is selected for command execution, TC will be qualified by DACK, but not in the programmed I/O execution. In PC/AT or Special Mode, qualification by DACK requires the Operations Register signal DMAEN to be logically true. Note also that in PC/AT Mode, TC will be qualified by DACK, whether in DMA or non-DMA Host operation. Programmed I/O in PC/AT Mode will cause an abnormal termination error at the completion of a command.
7-14 7-14	DB0 thru DB7	DATA BUS 0 thru DATA BUS 7	I/O	8-Bit, bi-directional, tri-state, data bus. D0 is the least significant bit (LSB). D7 is the most significant bit (MSB).

PIN DESCRIPTIONS (cont.)

D/P PIN NUMBER	MNEMONIC	SIGNAL NAME	I/O	FUNCTION
15/15	DMA	DIRECT MEMORY ACCESS	O	DMA request for byte transfers of data. In Special or PC/AT mode, this pin is tri-stated, enabled by the DMAEN signal from the Operation Register. This pin is driven in the Base Mode.
16/16	IRQ	INTERRUPT	O	Interrupt request indicating the completion of command execution or data transfer requests (in non-DMA mode). Normally driven in base mode. In Special or PC/AT Mode, this pin is tri-stated, enabled by the DMAEN signal from the Operations Register.
17				Not connected in the 44 pin PLCC.
17/18	LDOR	LOAD OPERATIONS REGISTER	I	Address decode which enables the loading of the Operations Register. Internally gated with WR creates the strobe which latches the data bus into the Operations Register.
18/19	LCR	LOAD CONTROL REGISTER	I	Address decode which enables loading of the Control Register. Internally gated with WR creates the strobe which latches the two LSBs from the data bus into the Control Register.
19/20	RST	RESET	I	Resets controller, placing microsequencer in idle. Resets device outputs. Puts device in Base Mode, not PC/AT or Special Mode.
20/21	RDD	READ DISK DATA	I	This is the raw serial bit stream from the disk drive. Each falling edge of the pulses represents a flux transition of the encoded data.
21/	CLK2	CLOCK2	I	TTL level clock input used for non-standard data rates; is 9.6MHz for 300 Kb/s, and can only be selected from the Control Register.
/22	XT2	XTAL2	O	XTAL oscillator drive output.
/23	XT2	XTAL2	I	XTAL oscillator input used for non-standard data rates. It may be driven with a TTL level signal.
22/24	DRV	DRIVE TYPE	I	Drive type input indicates to the device that a two-speed spindle motor is used if logic is 0. In that case, the second clock input will never be selected and must be grounded.
23/	CLK1	CLOCK1	I	TTL level clock input is used to generate all internal timings for standard data rates. Frequency must be 16MHz \pm 0.1%, and may have 40/60 or 60/40 duty cycle.
/25	XT1	XTAL1	O	XTAL oscillator drive output.
/26	XT1	XTAL1	I	XTAL oscillator input requiring 16MHz crystal. This oscillator is used for all standard data rates, and may be driven with a TTL level signal.
24/27	PCVAL	PRECOMPEN- SATION VALUE	I	PRECOMPENSATION VALUE select input. This pin determines the amount of write precompensation used on the inner tracks of the diskette. Logic 1 - 125ns, Logic 0 - 187ns.
25/28	HS	HEAD SELECT	O	High current driver (HCD) output selects the head (side) of the floppy disk that is being read or written. Logic 1 - side 0, Logic 0 - side 1.
26/29	WE	WRITE ENABLE	O	This HCD output becomes true, active low, just prior to writing on the diskette. This allows current to flow through the write head.

PIN DESCRIPTIONS (cont.)

D/P PIN NUMBER	MNEMONIC	SIGNAL NAME	I/O	FUNCTION
27/30	WD	WRITE DATA	O	This HCD output is WRITE DATA. Each falling edge of the encoded data pulse stream causes a flux transition on the media.
28/31	DIRC	DIRECTION	O	This HCD output determines the direction of the head stepper motor. Logic 1 - outward motion. Logic 0 - inward motion.
29/32	STEP	STEP PULSE	O	This HCD output issues an active low pulse for each track to track movement of the head.
30/33	DS1	DRIVE SELECT 1	O	This HCD output, when active low is DRIVE SELECT 1, in PC/AT Mode, enabling the interface in this disk drive. This signal comes from the Operations Register. In Base, or Special Mode, this output is #1 of the four decoded Unit Selects, as specified in the device command syntax.
31/34	VSS	GROUND	O	Ground.
32/35	DS2	DRIVE SELECT 2	O	This HCD output when active low is DRIVE SELECT 2, in PC/AT Mode, enabling the interface in this disk drive. This signal comes from the Operations Register. In Base or the Special Mode, this output is #2 of the four decoded Unit Selects as specified in the device command syntax.
33/36	MOT1, DS3	MOTOR ON 1, DRIVE SELECT 3	O	This HCD output when active low is MOTOR ON enable for disk drive #1, in PC/AT Mode. This signal comes from the Operations Register. In the Base or Special Mode, this output is #3 of the four decoded Unit Selects as specified in the device command syntax.
34/37	MOT2, DS4	MOTOR ON 2, DRIVE SELECT 4	O	This HCD output when active low is MOTOR ON enable for disk drive #2, in PC/AT mode. This signal comes from the Operations Register. In the Base or Special Mode, this output is #4 of the four decoded Unit Selects as specified in the device command syntax.
35/38	HDL	HEAD LOADED	O	This HCD output when active low causes the head to be loaded against the media in the selected drive.
36/39	RWC, RPM	REDUCED WRITE CURRENT REVOLUTIONS PER MINUTE	O	This HCD output when active low causes a REDUCED WRITE CURRENT when bit density is increased toward the inner tracks, becoming active when tracks > 28 are accessed. This condition is valid for Base or Special Mode, and is indicative of when write precompensation is necessary. In the PC/AT mode, this signal can be used on two-speed disk drives to select 300 RPM spindle rate, active low, when 250 MFM or 125 FM KB/s is selected.
40				Not connected in the 44 pin PLCC.
37/41	WP	WRITE PROTECTED	I	This Schmitt Trigger (ST) input senses status from the disk drive indicating active low, when a diskette is WRITE PROTECTED.
38/42	TR00	TRACK 00		This ST input senses status from disk drive indicating active low, when the head is positioned over the outermost track, TRACK 00.
39/43	IDX	INDEX	I	This ST input senses status from the disk drive indicating active low, when the head is positioned over the beginning of a track marked by an index hole.
40/44	VCC	+SVDC		Input power supply.

ARCHITECTURE

The WD37C65 Floppy Disk Subsystem Controller is an LSI device that provides all the needed functionality between the Host processor peripheral bus and the cable connector to the floppy disk drive. This "superchip" integrates: Formatter/Controller, Data Separation, Write

Precompensation, Data Rate Selection, Clock Generation, Drive Interface Drivers and Receivers.

Figure 1 illustrates a block diagram of the WD37C65 Floppy Disk Subsystem Controller.

Figure 2 illustrates a typical WD37C65 system.

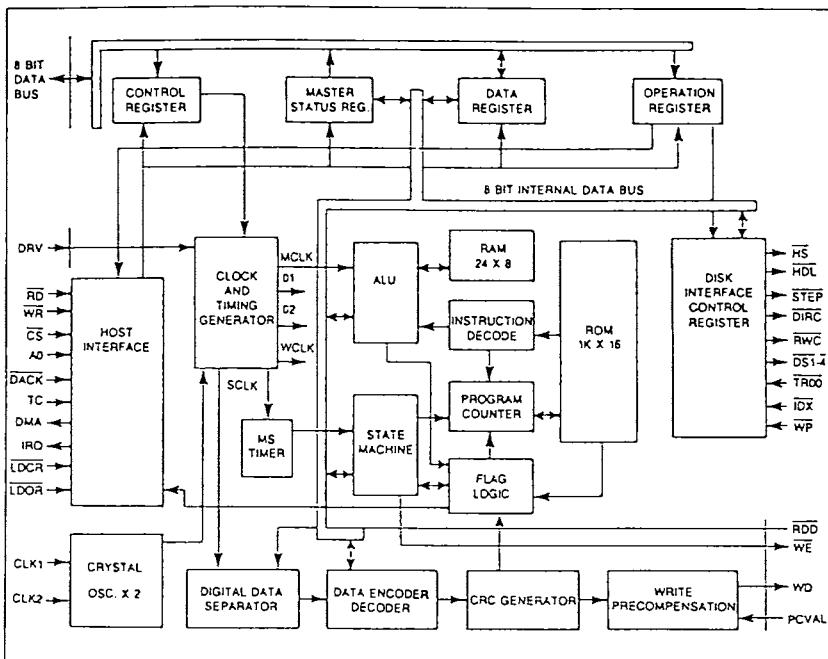


FIGURE 1. WD37C65 BLOCK DIAGRAM

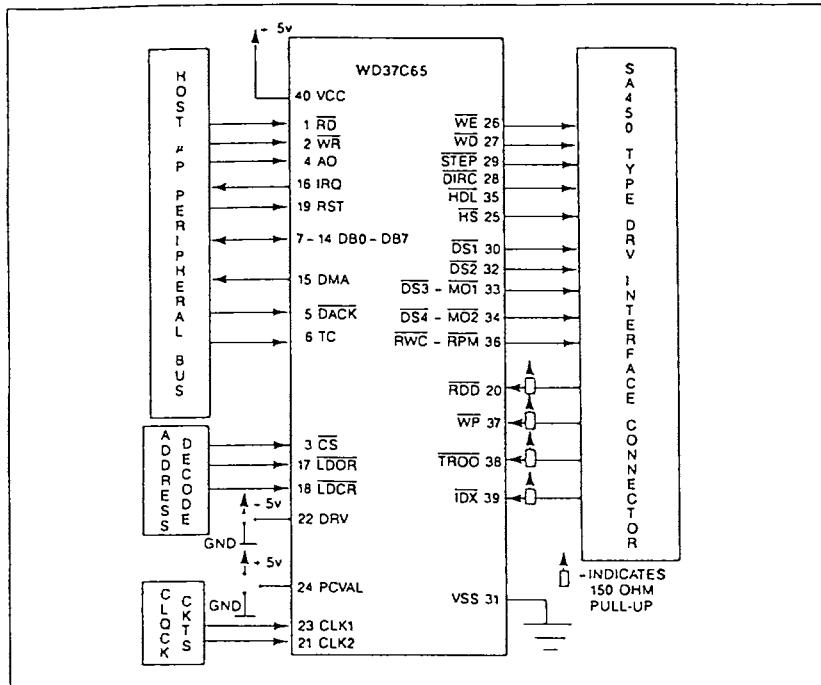


FIGURE 2. TYPICAL WD37C65 SYSTEM

HOST INTERFACE

The Host Interface is the Host Microprocessor Peripheral Bus. This bus is composed of eight control signals and eight data signals. In the Special or PC/AT Modes, IRO and DMA request are tri-stated and qualified by DMA enable, internally provided by the Operations Register. The data bus, DMA, and IRO outputs are designed to handle 20 LSSTL loading. Inputs, except the data bus, are Schmitt Trigger receivers and can be hooked up to a bus or backplane without any additional buffering. During the Command or Result phases the Main Status Register must be read by the processor before each byte of information is written into or read from the Data Register. After each byte of data is read from or written into the Data Register, the CPU should wait for 12 μ s before reading the Main Status Register. Bits D6 and D7 in the Main Status Register must be in a 0 and 1 state, respectively, before each byte of the command word may be written into the WD37C65. Many of the commands require multiple bytes, as result the Main Status Register must

be read prior to each byte transfer to the WD37C65. During the Result phase, Bits D6 and D7 in the Main Status Register must both be 1's (D6=1 and D7=1) before reading each byte from the Data Register. Note that this reading of the Main Status Register before each byte transfer to the WD37C65 is required only in the Command and Result phases, and not during the Execution phase. During the Execution phase, the Main Status Register need not be read. If the WD37C65 is in the non-DMA Mode, then the receipt of each data byte (WD37C65 is reading data from the FDD) is indicated by an interrupt signal on pin 16 (IRO-1). The generation of a Read signal (RD-0) or Write signal (WR-0) will clear the interrupt as well as output the data onto the data bus. If the processor cannot handle interrupts fast enough (every 13 μ s for the MFM Mode and 27 μ s for the FM Mode), then it may poll the Main Status Register and bit D7 (ROM) functions as the Interrupt signal. If a Write Command is in process then the WR signal performs the reset to the Interrupt signal.

All timings mentioned above double for mini floppy data rates.

Note that in the non-DMA Mode it is necessary to examine the Main Status Register to determine the cause of the interrupt, since it could be a data interrupt or a command termination interrupt, either normal or abnormal. If the WD37C65 is in the DMA Mode, no Interrupt signals are generated during the Execution phase. The WD37C65 generates DMA's (DMA Requests) when each byte of data is available. The DMA Controller responds to this request with both DACK=0 (DMA Acknowledge) and an RD=0 (Read signal). When the D/A Acknowledge signal goes low (DACK=0), then the DMA Request is cleared (DMA=0). If a Write Command has been issued then a WR signal will appear instead of RD. After the Execution phase has been completed (terminal Count has occurred) or the EOT sector read/written, then an Interrupt will occur (IRQ =1). This signifies the beginning of the Result phase. When the first byte of data is read during the Result phase, the Interrupt is automatically cleared (IRQ =0).

It should be noted that in PC/AT usage, non-DMA Host transfers are not the normal procedure. If the user chooses to do so, the WD37C65 will successfully complete commands, but will always give abnormal termination error status since TC is qualified by an inactive DACK.

The RD or WR signals should be asserted while DACK is true. The CS signal is used in conjunction with RD and WR as a gating function during programmed I/O operations. CS has no effect during DMA operations. If the non-DMA Mode is chosen, the DACK signal should be pulled up to Vcc. It is important to note that during the Result phase all bytes shown in the Command Table must be read. The Read Data Command for example, has several bytes of data in the Result phase. All seven bytes must be read in order to successfully complete the Head Data Command. The WD37C65 will not accept a

new command until all seven bytes have been read. Other commands may require fewer bytes to be read during the Result phase. The WD37C65 contains five Status Registers. The Main Status Register mentioned may be read by the processor at any time. The other four Status Registers (STD, ST1, ST2, and ST3) are available only during the Result phase and may be read only after completing a command. The particular command that has been executed determines how many of the Status Registers will be read.

The bytes of data which are sent to the WD37C65 to form the Command phase, and are read out of the WD37C65 in the Result phase, must occur in the order shown in the Command Table. The Command Code must be sent first and the other bytes sent in the prescribed sequence. No foreshortening of the Command or Result phases is allowed. After the last byte of data in the Command phase is sent to the WD37C65, the Execution phase automatically starts. In a similar fashion, when the last byte of data is read out in the Result phase, the command is automatically ended and the WD37C65 is ready for a new command.

CONTROL REGISTER

The Control Register provides support logic that latches the two LSBs of the data bus upon receiving LDCR and WR. These bits are used to select the desired data rate which in turn controls the internal clock generation. Clock switchover is internally "deglitched", allowing continuous operation after changing data rates. If the Control Register is not used, the data rate is governed by the supplied clock or crystal. The frequency must be 64X the desired MFM data rate, up to a maximum frequency of 16 MHz. This implies a maximum data rate of 250 Kb/s unless the Control Register is used. Switching this clock must be "glitchless" or the device will need to be reset. Table 1 presents the Control Register.

TABLE 1. CONTROL REGISTER

CR1	CR0	DRV	DATA RATE	COMMENTS	RPM (IN PC/AT MODE)
0	0	X	500 K	MFM	1
0	0	X	250 K	FM	1
0	1	0	250 K	MFM	0
0	1	1	300 K	MFM	0
1	0	X	250 K	MFM, RST Default	1
1	0	X	125 K	FM, RST Default	1
1	1	X	125 K	FM	0

MASTER STATUS REGISTER

The Master Status Register is an 8-bit register that contains the status information of the FDC, and may be accessed at any time. Only the Master Status Register may be read and used to facilitate the transfer of data between the processor and WD37C65. The DIO and ROM bits in the Master Status Register indicate when data is ready and in which direction data will be transferred on the data bus. The maximum time between the last RD

or WR during a command or result phase and DIO and ROM getting set is 12μs if 500 Kb/s MFM data rate is selected. (If 250 Kb/s MFM is selected, the delay is 24μs.) For this reason everytime the Master Status Register is read the CPU should wait 12μs. The maximum time from the trailing edge of the last RD in the result phase to when DB4 (FDC busy) goes low is 12μs.

The bits in the Master Status Register are listed in Table 2.

TABLE 2. MASTER STATUS REGISTER BITS

BIT			DESCRIPTION
NO.	NAME	SYMBOL	
DB0	FDD 0 BUSY	D0B	FDD number is 0 in the Seek Mode. If any of the bits is set, FDC will not accept READ or WRITE Commands.
DB1	FDD 1 BUSY	D1B	FDD number 1 is in the Seek Mode. If any of the bits is set, FDC will not accept READ or WRITE Commands.
DB2	FDD 2 BUSY	D2B	FDD number 2 is in the Seek Mode. If any of the bits is set, FDC will not accept READ or WRITE Commands.
DB3	FDD 3 BUSY	D3B	FDD number 3 is in the Seek Mode. If any of the bits is set, FDC will not accept READ or WRITE Commands.
DB4	FDC BUSY	CB	A READ or WRITE Command is in progress. FDC will not accept any other command.
DB5	EXECUTION MODE	EXM	This bit is set only during Execution Phase in non-DMA Mode. When DB5 goes low Execution Phase has ended and Results Phase has started. It operates only during non-DMA Mode of operation.
DB6	DATA INPUT	DIO	Indicates direction of data transfer between FDC and Data Register. If DIO=1, then transfer is from Data Register to the processor. If DIO=0, then transfer is from the processor to Data Register.
DB7	REQUEST FOR MASTER	RQM	Indicates Data Register is ready to send or receive data to or from the processor. Both bits DIO and RQM should be used to perform the handshaking functions of "ready" and "direction" to the processor.

The bits in Status Register 0 are listed in Table 3.

TABLE 3. STATUS REGISTER 0 BITS

BIT			DESCRIPTION
NO.	NAME	SYMBOL	
D7	INTERRUPT CODE	IC	D7=0 and D6=0. Normal Termination of command was completed and properly executed. D7=0 and D6=1. Abnormal Termination of command, (AT). Execution of command was started but was not successfully completed.
D6			D7=1 and D6=0. Invalid Command issue, (IC). Command which was issued was never started. D7=1 and D6=1. Abnormal Termination because during command execution the ready signal from FDD changed state.
D5	SEEK END	SE	When the FDC completes the SEEK Command this flag is set to 1 (high).
D4	EQUIPMENT CHECK	EC	If a fault signal is received from the FDD, or if the Track 0 signal fails to occur after 255 step pulses (Recalibrate Command) then this flag is set.
D3	NOT READY	NR	Since drive Ready is always presumed true, this will always be a logic 0.
D2	HEAD SELECT	HS	This flag is used to indicate the state of the head at interrupt.
D1	UNIT SELECT 1	US1	This flag is used to indicate a Drive Unit Number at interrupt.
D0	UNIT SELECT 0	US0	This flag is used to indicate a Drive Unit Number at interrupt.

The bits in Status Register 1 are listed in Table 4.

TABLE 4. STATUS REGISTER 1 BITS

BIT			DESCRIPTION
NO.	NAME	SYMBOL	
D7	END OF CYLINDER	EN	When the FDC tries to access a sector beyond the final sector of a cylinder, this flag is set.
D6			Not used. This bit is always 0 (low).
D5	DATA ERROR	DE	When the FDC detects a *CRC error in either the ID field or the data field, this flag is set.
D4	OVERRUN	OR	If the FDC is not serviced by the Host system during data transfers within a certain time interval, this flag is set.
D3			Not used. This bit is always 0 (low).
D2	NO DATA	ND	During execution of READ DATA, WRITE DELETED DATA, or SCAN Command, if the FDC cannot find the sector specified in the * * IDR Register, this flag is set. During execution of the READ ID Command if the FDC cannot read the ID field without an error, then this flag is set.
			During execution of the READ A TRACK Command, if the starting sector cannot be found, then this flag is set.
D1	NOT WRITABLE	NW	During execution of WRITE DATA, WRITE DELETED DATA or FORMAT A TRACK Command, if the FDC detects a WP signal from the FDD, then this flag is set.
D0	MISSING ADDRESS MARK	MA	If the FDC cannot detect the ID Address Mark after encountering the index hole twice, then this flag is set. If the FDC cannot detect the Data Address Mark or Deleted Data Address Mark, this flag is set. At the same time the MD (Missing Address Mark in data field) of Status Register 2 is set.

The bits in Status Register 2 are listed in Table 5.

TABLE 5. STATUS REGISTER 2 BITS

BIT			DESCRIPTION
NO.	NAME	SYMBOL	
D7			Not Used. This bit is always 0 (low).
D6	CONTROL MARK	CM	During execution of the READ DATA or SCAN Command, if the FDC encounters a sector which contains a Deleted Data Address Mark, this flag is set.
D5	DATA ERROR	DD	If the FDC detects a CRC error in the data field then this flag is set.
D4	WRONG CYLINDER	WC	This bit is related to the ND bit, and the contents of * * * C on the medium is different from that stored in the IDR, this flag is set.
D3	SCAN EQUAL	SH	During execution of the SCAN Command if the condition of "equal" is satisfied, this flag is set.
D2	SCAN NOT	SN	During execution of the SCAN Command, if the FDC cannot find a sector on the cylinder which meets the condition, then this flag is set.
D1	BAD CYLINDER	BC	This bit is related to the ND bit, and when the contents of C on the medium is different from that stored in the IDR and the contents of C is FF, then this flag is set.
D0	MISSING ADDRESS MARK IN DATA FIELD	MD	When data is read from the medium, if the FDC cannot find a Data Address Mark or Deleted Data Address Mark, then this flag is set.

The bits in Status Register 3 are listed in Table 6.

TABLE 6. STATUS REGISTER 3 BITS

BIT			DESCRIPTION
NO.	NAME	SYMBOL	
D7		-	Not used. Will always be logic 0.
D6	WRITE PROTECTED	WP	This bit is used to indicate the status of the WRITE PROTECTED signal from the FDD.
D5	READY	RY	This bit will always be a logic 1. Drive is presumed to be ready.
D4	TRACK 0	TO	This bit is used to indicate the status of the Track 0 signal from the FDD.
D3	WRITE PROTECTED	WP	This bit is used to indicate the status of the WRITE PROTECTED signal from the FDD.
D2	HEAD SELECT	HS	This bit is used to indicate the status of the Side Select signal to the FDD.
D1	UNIT SELECT 1	US1	This bit is used to indicate the status of the Unit Select 1 signal to the FDD.
D0	UNIT SELECT 2	US0	This bit is used to indicate the status of the Unit Select 0 signal to the FDD.

* CRC - Cyclic Redundancy Check

** IDR - Internal Data Register

*** C - Cylinder

DATA REGISTER

The 8-bit Data Register stores data, commands, parameters, and FDD status information. Data bytes are read out of, or written into, the Data Register in order to program or obtain the results after a particular command.

The relationship between the Master Status Register and the Data Register and the signals RD, WR, and A0 are shown in Table 7.

TABLE 7. MASTER STATUS AND DATA REGISTERS RELATIONSHIP

A0	RD	WR	FUNCTION
0	0	1	Read Main Status Register
0	1	0	Illegal
0	0	0	Illegal
1	0	0	Illegal
1	0	1	Read from Data Register
1	1	0	Write into Data Register

OPERATIONS REGISTER

The Operations Register provides support logic that latches the data bus upon receiving LDOR and WR. The Operations Register replaces the typical latched port

found in floppy subsystems used to control disk drive spindle motors and to select the desired disk drive. Table 8 represents the Operations Register.

TABLE 8. OPERATIONS REGISTER

OR0	DSEL	: Drive Select, if low and MOEN1 = 1, then DS1 is active. If high and MOEN2 = 1, then DS2 is active, but only in the PC/AT Mode.
OR1	(X)	: Has no defined function. A spare.
OR2	SRST	: Soft reset, active low.
OR3	DMAEN	: DMA enable, active in Special and PC/AT Mode. Qualifies DMA and IRQ outputs and DACK input.
OR4	MOEN1	: Motor On enable, inverted output M01 is active only in PC/AT Mode.
OR5	MOEN2	: Motor On enable, inverted output M02 is active only in PC/AT Mode.
OR6	(X)	: Has no defined function. A spare.
OR7	(X)	: Mode Select. During a soft reset condition, may be used to select between Special Mode and PC/AT Mode.

BASE, SPECIAL, AND PC/AT MODES

Base, Special, and PC/AT Modes allow subtle differences which the user may find desirable. The Control Register may be used in any mode without altering functionality.

Base Mode

After a hardware reset, RST active, the WD37C65 will be held in soft reset, SRST active, with the normally driven signals, DMA request and IRQ request outputs tri-stated. Base Mode may be initiated at this time by a chip access by the Host. Although this may be any read or write, it is strongly recommended that the Base Mode user's first chip access be a read of the Master Status Register. Once Base Mode is entered the soft reset is released, and IRQ and DMA are driven. Base Mode prohibits the use of the Operations Register, hence there can be no qualifying by DMAEN and no soft resets. The Drive Select outputs, DS1 to DS4, offer a 1 of 4 decoding of the Unit Select bits resident in the command structure. Pin RWC represents Reduce Write Current and is indicative of when write precompensation is necessary.

Special Mode

Special Mode allows use of the Operations Register for the DMAEN signal as a qualifier and to do a software driven device reset SRST. To enter Special Mode, the Operations Register is loaded with (1 X 0 0 X 0 X), setting Mode Select to a logic 1 disabling MOEN1 and MOEN2 and causing SRST to be active. Then a read of the Control Register address, LDCR and RD, will set the device into Special Mode. The DS1 through DS4 is again offered in this mode, as is RWC.

PC/AT Mode

For PC/AT compatibility users will write to the Operations Register, LDOR and WR; this action performed after a

hardware reset, or in the Base Mode, initiates PC/AT Mode. PC/AT Mode can also be entered from Special Mode by loading the Operations Register with (0 X 0 0 X 0 X), setting Mode Select to a logic 0, disabling MOEN1 and MOEN2, and causing SRST to be active. Then a read of the Control Register address sets the device into PC/AT Mode. The DS outputs are now replaced with the DSEL and MOEN signals buffered from the Operations Register. DMAEN and SRST are supported and compatible with the current BIOS. RWC pin function is now RPM so that users with two-speed drives may reduce spindle speed from a nominal 360 revolutions per minute, to 300 revolutions per minute when active low, or used to reduce write current when a slower data rate is selected for a given drive.

POLLING ROUTINE

After any reset the WD37C65, a hard RST or soft SRST, US0 and US1, will automatically go into a Polling Routine. In between commands (and between step pulses in the SEEK Command) the WD37C65 polls all four FDDs looking for a change in the Ready line from any of the drives. Since the drive is always presumed Ready, an interrupt will only be generated following a reset. This occurs because a reset forces Not Ready status, which then promptly becomes Ready. Note that in Special or PC/AT Modes, if DMAEN is not valid prior to 1ms after reset goes inactive, then IRO may be already set and pending when finally enabled onto the bus. The polling of the Ready line by the WD37C65 occurs continuously between commands. Each drive is polled every 1.024ms except during the READ/WRITE Commands. For mini-floppies, the polling rate is 2.048ms. The drive polling sequence is 1-2-4-3.

Figure 2 illustrates the Drive Select Polling Timing.

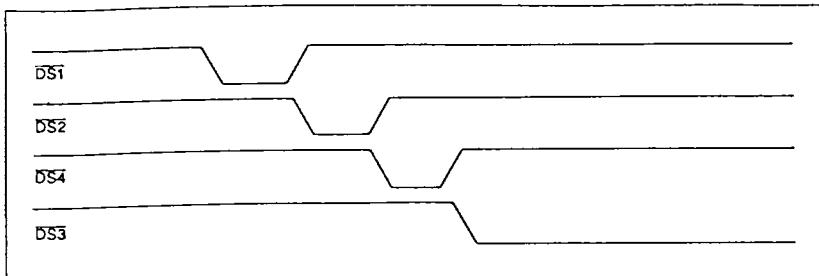


FIGURE 2. DRIVE SELECT POLLING TIMING

DEVICE RESETS

The WD37C65 supports both hardware reset (RST) pin (19), and a software reset (SRST), through use of the Operations Register. The RST pin will cause a device reset for the active duration. RST causes a default to Base Mode, and default selects 250K MFM (or 125K FM, code dependent) as the data rate (16 MHz input clock). SRST will reset the microcontroller as did the RST, but will not affect the current data rate selection or the mode. Both RST and SRST, when active, will disable the high current driver outputs to the disk drive.

If the XTAL oscillators are used, instead of the TTL driven clock inputs, the hardware RST active time requirement will be extended. The oscillator circuit is designed so that RST will bootstrap the circuit into guaranteed oscillation

in a fixed amount of time. The extended reset time allows the growth of the oscillation to produce stable internal clock timing.

DATA SEPARATOR

The Data Separator is a WD92C32 Digital Phase Lock Loop Floppy Disk Data Separator (DPLL). It was designed to address high performance error rates on floppy disk drives, and to provide a superior performance in terms of available bit jitter tolerance. It contains the necessary logic to achieve classical 2nd order, type 2, phase locked loop performance. Figure 1 illustrates the WD92C32 used as the Data Separator in the WD37C65 system. Figure 3 illustrates the WD92C32 simplified block diagram.

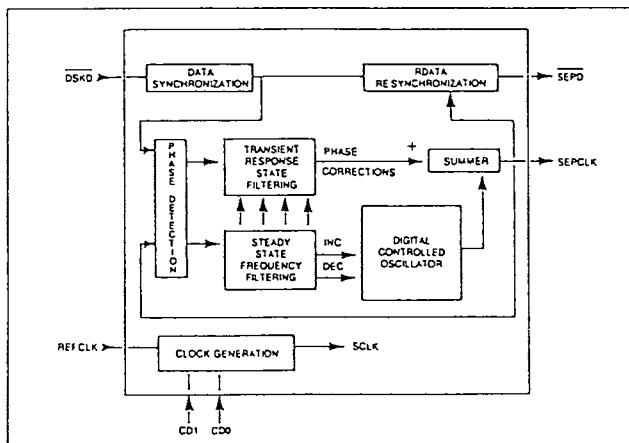


FIGURE 3. WD92C32 SIMPLIFIED BLOCK DIAGRAM

WRITE PRECOMPENSATION

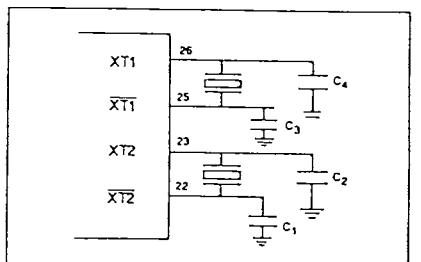
The WD37C65 maintains the standard first level algorithm to determine when Write Precompensation should be applied. The EARLY and LATE signals are used internally to select the appropriate delay in the write data pulse stream. The encoded WRITE DATA signal is synchronized to the 16 MHz clock, if this is the frequency on CLK1 pin (23), and clocked through a shift register. Signals EARLY, NOM, and LATE determine the amount of delay through the shift register before a multiplexed gates the chosen bit to the output. The output data pulse width has a 25% duty cycle, i.e., $\frac{1}{4}$ of the bit cell period, and equal to $\frac{1}{4}$ the WCLK period.

When PCVAL pin (24) = 1 all data will be precompensated by $\pm 125\text{ns}$ regardless of track number and data rate, however, this is only for MFM encoding. There is no Write Precompensation for FM. If PCVAL = 0, and if a track inside number 28 is accessed, then $\pm 187\text{ns}$ precompensation will be generated. For frequencies other than 16 MHz on the CLK1 pin, the precompensation values will be two and three clock cycles respectively.

TABLE 9. CLOCK DATA RATE

DATA RATE	CODE	SCLK	MCLK	WCLK
500 Kb/S	MFM	16.0 MHz	4.0 MHz	1.0 MHz
250 Kb/S	FM	8.0 MHz	4.0 MHz	500 KHz
250 Kb/S	MFM	8.0 MHz	2.0 MHz	500 KHz
125 Kb/S	FM	4.0 MHz	2.0 MHz	250 KHz
300 Kb/S	MFM	9.6 MHz	2.4 MHz	600 KHz

Figure 4 illustrates the XTAL oscillator circuits for the 44 pin PLCC configuration.



9.6 MHz \pm 100 ppm	16.0 MHz \pm 100 ppm
R series = 30 ohm max	R series = 30 ohm max
C shunt = 10 pf max	C shunt = 10 pf max
C1 = 68 pf 5% mica	C3 = 47 pf 5% mica
C2 = 56 pf 5% mica	C4 = 15 pf 5% mica

FIGURE 4. XTAL OSCILLATOR CIRCUITS FOR THE 44 PIN PLCC

When the non-standard data rate using CLK2 is chosen, the MFM precompensation will always be 2 clock cycles. For 9.6 MHz this is $\pm 208\text{ns}$. In this case, the PCVAL function is disabled.

CLOCK GENERATION

This logical block provides all the clocks needed by the WD37C65. They are: Sampling Clock (SCLK), Write Clock (WCLK), and the Master Clock (MCLK).

SCLK drives the WD92C32 Data Separator used during data recovery. This clock's frequency is always 32X the selected data rate.

WCLK is used by the encoder logic to place MFM or FM on the serial WD-stream to the disk. WCLK always has a frequency 2X the selected data rate.

MCLK is used by the microsequencer. MCLK and $\overline{\text{MCLK}}$ clock all latches in a two-phase scheme. One microinstruction cycle is four MCLK cycles. MCLK has a frequency equal to 8X the selected MFM data rate or 16X the FM data rate. Table 9 presents the Clock Data Rate.

COMMANDS

The WD37C65 is capable of performing 15 different commands. Each command is initiated by a multibyte transfer from the processor. The results after execution of the command may also be a multibyte transfer back to the processor. The commands consist of three phases: Command Phase, Execution Phase, and the Result Phase.

Command Phase – The Floppy Disk Controller (FDC) receives all information required to perform a particular operation from the processor.

Execution Phase – The FDC performs the operation it was instructed to do.

Result Phase – After completion of the operation, status and other housekeeping information are made available to the processor.

Table 10 lists the 15 WD37C65 commands.

TABLE 10. WD37C65 COMMANDS

READ DATA
READ DELETED DATA
WRITE DATA
WRITE DELETED DATA
READ A TRACK
READ ID
FORMAT A TRACK
SCAN EQUAL
SCAN LOW OR EQUAL
SCAN HIGH OR EQUAL
RECALIBRATE
SENSE INTERRUPT STATUS
SPECIFY
SENSE DRIVE STATUS
SEEK

Tables 11 through 25 are presented to show the required parameters and results for each command. Most commands require nine command bytes and return seven

bytes during the result phase. The "W" to the left of each byte indicates a command phase byte to be written. An "R" indicates a result byte.

TABLE 11. READ DATA

PHASE	R/W	D7	D6	D5	D4	D3	D2	D1	D0	REMARKS	
COMMAND	W	MT	MF	SK	0	0	1	1	0	Command Codes	
	W	X	X	X	X	X	HS	US1	US0		
	W	C									
	W	H									
	W	R									
	W	N									
	W	EOT									
	W	GPL									
	W	DTL									
EXECUTION										Data transfer between FDD and main system.	
RESULTS	R					ST0				Status information after command execution.	
	R					ST1					
	R					ST2					
	R					C				Sector ID information after command execution.	
	R					H					
	R					R					
	R					N					

TABLE 12. READ DELETED DATA

PHASE	R/W	D7	D6	D5	D4		D3	D2	D1	D0	REMARKS
COMMAND	W	MT	MF	SK	0		1	1	0	0	Command Codes
	W	X	X	X	X		X	HS	US1	US0	Sector ID information prior to command execution. The four bytes are compared against header on floppy disk.
	W	←	C	→							
	W	←	H	→							
	W	←	R	→							
	W	←	N	→							
	W	←	EOT	→							
	W	←	GPL	→							
	W	←	DTL	→							
EXECUTION											
RESULTS	R	←	ST0	→							Status information after command execution.
	R	←	ST1	→							
	R	←	ST2	→							
	R	←	C	→							
	R	←	H	→							
	R	←	R	→							
	R	←	N	→							
	R	←	DTL	→							

TABLE 13. WRITE DATA

PHASE	R/W	D7	D6	D5	D4		D3	D2	D1	D0	REMARKS
COMMAND	W	MT	MF	0	0		0	1	0	1	Command Codes
	W	X	X	X	X		X	HS	US1	US0	Sector ID information prior to command execution. The four bytes are compared against header on floppy disk.
	W	←	C	→							
	W	←	H	→							
	W	←	R	→							
	W	←	N	→							
	W	←	EOT	→							
	W	←	GPL	→							
	W	←	DTL	→							
EXECUTION											
RESULTS	R	←	ST0	→							Status information after command execution.
	R	←	ST1	→							
	R	←	ST2	→							
	R	←	C	→							
	R	←	H	→							
	R	←	R	→							
	R	←	N	→							
	R	←	DTL	→							

TABLE 14. WRITE DELETED DATA

PHASE	R/W	D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
COMMAND	W	MT	MF	0	0	1	0	0	1	Command Codes
	W	X	X	X	X	X	HS	US1	US0	Sector ID information prior to command execution. The four bytes are compared against header on floppy disk.
	W	←	C	→						
	W	←	H	→						
	W	←	R	→						
	W	←	N	→						
	W	←	EOT	→						
	W	←	GPL	→						
	W	←	DTL	→						
EXECUTION										Data transfer between FDD and main system.
RESULTS	R	←	ST0	→						Status information after command execution.
	R	←	ST1	→						
	R	←	ST2	→						
	R	←	C	→						
	R	←	H	→						
	R	←	R	→						
	R	←	N	→						

TABLE 15. READ A TRACK

PHASE	R/W	D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
COMMAND	W	0	MF	SK	0	0	0	1	0	Command Codes
	W	X	X	X	X	X	HS	US1	US0	Sector ID information prior to command execution.
	W	←	C	→						
	W	←	H	→						
	W	←	R	→						
	W	←	N	→						
	W	←	EOT	→						
	W	←	GPL	→						
	W	←	DTL	→						
EXECUTION										Data transfer between FDD and main system. FDD reads all data fields from index hole to EOT.
RESULTS	R	←	ST0	→						Status information after command execution.
	R	←	ST1	→						
	R	←	ST2	→						
	R	←	C	→						
	R	←	H	→						
	R	←	R	→						
	R	←	N	→						

TABLE 16. READ ID

PHASE	R/W	D7	D6	D5	D4		D3	D2	D1	D0	REMARKS
COMMAND	W	0	MF	0	0		1	0	1	0	Command Codes
	W	X	X	X	X		X	HS	US1	US0	
EXECUTION											The first correct ID information on the cylinder is stored in Data Register.
RESULTS	R					STD					Status information after command execution.
	R					ST1					
	R					ST2					
	R					C					Sector ID information read during Execution Phase from floppy disk.
	R					H					
	R					R					
	R					N					

TABLE 17. FORMAT A TRACK

PHASE	R/W	D7	D6	D5	D4		D3	D2	D1	D0	REMARKS
COMMAND	W	0	MF	0	0		1	1	0	1	Command Codes
	W	X	X	X	X		X	HS	US1	US0	
	W					N					Bytes/Sector
	W					SC					Sectors/Track
	W					GPL					Gap 3
	W					D					Filler Byte
EXECUTION											Floppy Disk Controller (FDC) formats an entire track.
RESULTS	R					STD					Status information after command execution.
	R					ST1					
	R					ST2					
	R					C					In this case, the ID information has no meaning.
	R					H					
	R					R					
	R					N					

TABLE 18. SCAN EQUAL

PHASE	R/W	D7	D6	D5	D4		D3	D2	D1	D0	REMARKS
COMMAND	W	MT	MF	SK	1		0	0	0	1	Command Codes
	W	X	X	X	X		X	HS	US1	US0	
	W					C					Sector ID information prior to command execution.
	W					H					
	W					R					
	W					N					
	W					EOT					
	W					GPL					
	W					STP					
EXECUTION											Data compared between the FDD and main system.
RESULTS	R					STD					Status information after command execution.
	R					ST1					
	R					ST2					
	R					C					Sector ID information after command execution.
	R					H					
	R					R					
	R					N					

TABLE 19. SCAN LOW OR EQUAL

PHASE	R/W	D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
COMMAND	W	MT	MF	SK	1	1	0	0	1	Command Codes
	W	X	X	X	X	X	HS	US1	US0	Sector ID information prior to command execution.
	W	←	—	—	C	→	—	—	—	
	W	←	—	—	H	→	—	—	—	
	W	←	—	—	R	→	—	—	—	
	W	←	—	—	N	→	—	—	—	
	W	←	—	—	EOT	→	—	—	—	
	W	←	—	—	GPL	→	—	—	—	
EXECUTION	W	←	—	—	STP	→	—	—	—	Data compared between the FDD and main system.
	R	←	—	—	ST0	→	—	—	—	Status information after command execution.
	R	←	—	—	ST1	→	—	—	—	
	R	←	—	—	ST2	→	—	—	—	
	R	←	—	—	C	→	—	—	—	
	R	←	—	—	H	→	—	—	—	
	R	←	—	—	R	→	—	—	—	
	R	←	—	—	N	→	—	—	—	
RESULTS	R	←	—	—	—	—	—	—	—	Sector ID information after command execution.

TABLE 20. SCAN HIGH OR EQUAL

PHASE	R/W	D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
COMMAND	W	MT	MF	SK	1	1	1	0	1	Command Codes
	W	X	X	X	X	X	HS	US1	US0	Sector ID information prior to command execution.
	W	←	—	—	C	→	—	—	—	
	W	←	—	—	H	→	—	—	—	
	W	←	—	—	R	→	—	—	—	
	W	←	—	—	N	→	—	—	—	
	W	←	—	—	EOT	→	—	—	—	
	W	←	—	—	GPL	→	—	—	—	
EXECUTION	W	←	—	—	STP	→	—	—	—	Data compared between the FDD and main system.
	R	←	—	—	ST0	→	—	—	—	Status information after command execution.
	R	←	—	—	ST1	→	—	—	—	
	R	←	—	—	ST2	→	—	—	—	
	R	←	—	—	C	→	—	—	—	
	R	←	—	—	H	→	—	—	—	
	R	←	—	—	R	→	—	—	—	
	R	←	—	—	N	→	—	—	—	
RESULTS	R	←	—	—	—	—	—	—	—	Sector ID information after command execution.

TABLE 21. RECALIBRATE

PHASE	R/W	D7	D6	D5	D4		D3	D2	D1	D0	REMARKS
COMMAND	W	0	0	0	0		0	1	1	1	Command Codes
	W	X	X	X	X		X	0	US1	US0	

TABLE 22. SENSE INTERRUPT STATUS

PHASE	R/W	D7	D6	D5	D4		D3	D2	D1	D0	REMARKS
COMMAND	W	0	0	0	0		1	0	0	0	Command Codes
RESULTS	R	←	STD	→							Status information about the FDC at the end of seek operation.
	R	←	PCN	→							

TABLE 23. SPECIFY

PHASE	R/W	D7	D6	D5	D4		D3	D2	D1	D0	REMARKS
COMMAND	W	0	0	0	0		0	0	1	1	Command Codes
	W	←	SRT	→	←	HUT	→				
	W	←	HLT	→							ND

TABLE 24. SENSE DRIVE STATUS

PHASE	R/W	D7	D6	D5	D4		D3	D2	D1	D0	REMARKS
COMMAND	W	0	0	0	0		0	1	0	0	Command Codes
	W	X	X	X	X		X	HS	US1	US0	

RESULTS R ← ST3 → Status information about the FDC.

TABLE 25. SEEK

PHASE	R/W	D7	D6	D5	D4		D3	D2	D1	D0	REMARKS
COMMAND	W	0	0	0	0		1	1	1	1	Command Codes
	W	X	X	X	X		X	HS	US1	US0	
EXECUTION		←	NCN	→							Head is positioned over proper cylinder on the diskette.

Table 26 defines, in alphabetical order, the symbols used in Command Tables 11 through 25.

TABLE 26. COMMAND SYMBOL DESCRIPTIONS

SYMBOL	NAME	DESCRIPTION
A0	ADDRESS LINE 0	A0 controls selection of Main Status Register (A0=0) or Data Register (A0=1).
C	CYLINDER NUMBER	C stands for the current/selected cylinder (track) numbers 0 through 255 of the medium.
D	DATA	D stands for the data pattern which is going to be written into a sector.
D7 - D0	DATA BUS	8-bit DATA BUS, where D7 stands for a most significant bit, and D0 stands for a least significant bit.
DTL	DATA LENGTH	When N is defined as 00, DTL stands for the DATA LENGTH which users are going to read out or write into the sector.

TABLE 26. COMMAND SYMBOL DESCRIPTIONS (cont.)

SYMBOL	NAME	DESCRIPTION
EOT	END OF TRACK	EOT stands for the final sector number on a cylinder. During Read or Write operations, FDC will stop data transfer after a sector number equal to EOT.
GPL	GAP LENGTH	GPL stands for the length of Gap 3. During the FORMAT Command it determines the size of Gap 3.
H	HEAD ADDRESS	H stands for head number 0 or 1, as specified in the ID field.
HLT	HEAD LOAD TIME	HLT stands for the HEAD LOAD TIME in FDD (2 to 254ms in 2ms increments).
HS	HEAD SELECT	HS stands for a selected head number 0 or 1 and controls the polarity of pin 25 (in 40 pin DIP) or pin 28 (in 44 pin PLCC).
HUT	HEAD UNLOAD TIME	HUT stands for the HEAD UNLOAD TIME after a Read or Write operation has occurred (16 to 240ms in 16ms increments).
MF	FM or MFM	If MF is low, FM mode is selected, and if it is high, MFM mode is selected.
MT	MULTITRACK	If MT is high, a MULTITRACK operation is performed. If MT=1 after finishing Read/Write operation on side 0, FDC will automatically start searching for sector 1 on side 1.
N	NUMBER	N stands for the NUMBER of data bytes written in a sector.
NCN	NEW CYLINDER NUMBER	NCN stands for a NEW CYLINDER NUMBER which is going to be reached as a result of the Seek operation. Desired position of head.
ND	NON-DMA MODE	ND stands for operation in the NON-DMA MODE.
PCN	PRESENT CYLINDER	PCN stands for the cylinder number at the completion of the SENSE INTERRUPT STATUS Command. Position of head at present time.
R	RECORD	R stands for the sector number which will be read or written.
R/W	READ/WRITE	R/W stands for either READ or WRITE signal.
SC	SECTOR	SC indicates the number of Sectors per cylinder.
SK	SKIP	SK stands for SKIP Deleted Data Address mark.
SRT	STEP RATE TIME	SRT stands for the Stepping Rate for the FDD (1 to 16ms in 1ms increments). Stepping Rate applies to all drives. In 2's complement format, F(Hex)-1ms, E(Hex)-2ms, etc.
ST0 ST1 ST2 ST3	STATUS 0 STATUS 1 STATUS 2 STATUS 3	ST0 - 3 stands for one of four registers which store the STATUS information after a command has been executed. This information is available during the result phase after command execution. These registers should not be confused with the Main Status Register (selected by A0=0). ST0 - 3 may be read only after a command has been executed and contains information relevant to that particular command.
STP		During a SCAN operation, if STP=1, the data in contiguous sectors is compared byte by byte with data sent from the processor (or DMA); if STP=2, then alternate sectors are read and compared.
US0,US1	UNIT SELECT	US stands for a selected drive; binary encoded, 1 of 4.

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Operating Temperature	0°C (32°F) to 70°C (158°F)
Storage Temperature	-55°C (-67°F) to +125°C (257°F)
Voltage on any pin with respect to ground	-0.3V to VCC +0.3V
Supply Voltage with respect to ground	7V

NOTE

Maximum limits indicate where permanent device damage occurs. Continuous operation at these limits is not intended and should be limited to those conditions specified in the DC Operating Characteristics.

DC Operating Characteristics

TA = 0°C (32°F) to 70°C (158°F); VCC = +5V ± 10%

SYMBOL	PARAMETER	MIN	MAX	UNITS
VCC	+5VDC Power Supply	4.5	5.5	V
VIL	Input Low Voltage - Data Bus & XTOSC		0.8	V
VIH	Input High Volt - Data Bus & XTOSC	2.0		V
VILT	Input Low Threshold - Schmitt Trigger	0.9	1.1	V
VIHT	Input High Threshold - Schmitt Trigger	1.7	1.9	V
VOL	Output Low - DBx,IRQ,DMA.; Io = 12.0mA		0.4	V
VOH	Output High - DBx,IRQ,DMA.; Io = -5.0mA	2.8		V
VOLHC	Output Low - High Current; Io = 48.0mA		0.4	V
ILUL	Latch Up Current Low	40.0		mA
ILUH	Latch Up Current High	-40.0		mA
ILL	Leakage Current Low		10.0	uA
ILH	Leakage Current High		-10.0	uA
ICC	Supply Current - 100uA Source Loads		35.0	mA
ICCHL	Supply Current - 5.0mA Source Loads		85.0	mA
PD	Power Dissipation - ICC Max *		375.0	mW
PDHL	Power Dissipation - ICCHL Max *		525.0	mW

* Includes open drain high current drivers at Vol = 0.4V.

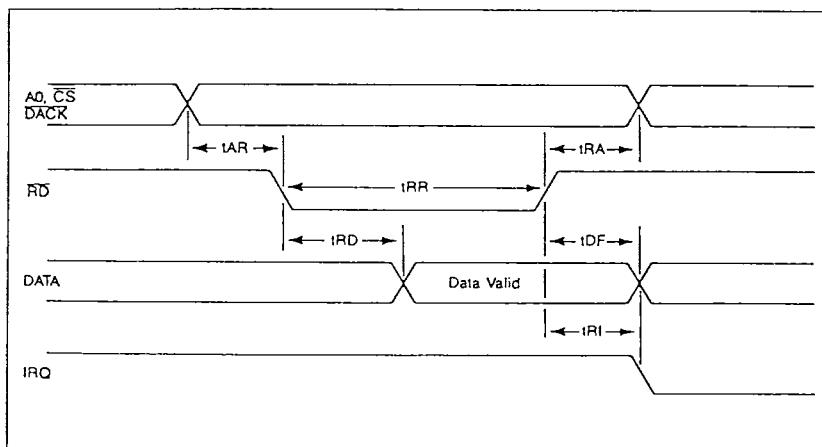


FIGURE 5. READ TIMING

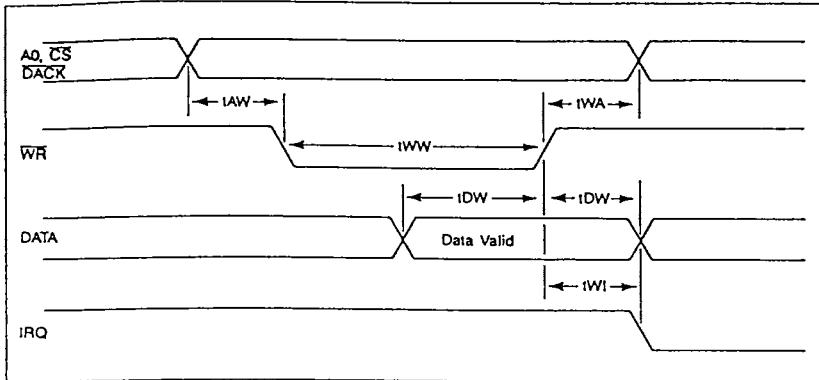


FIGURE 6. WRITE TIMING

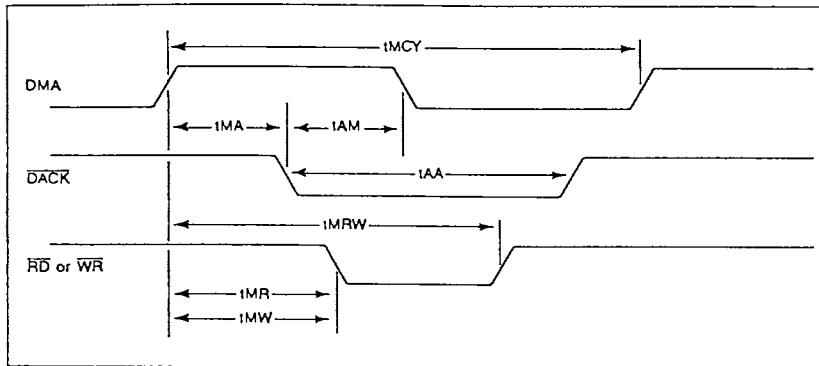


FIGURE 7. DMA TIMING

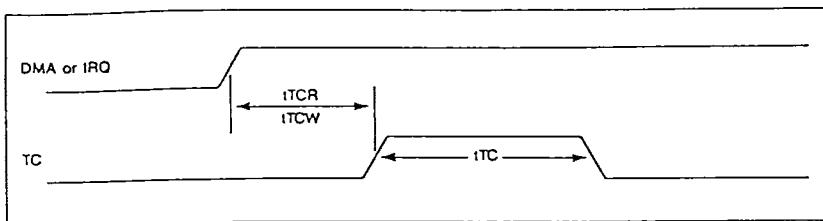


FIGURE 8. TERMINAL COUNT TIMING

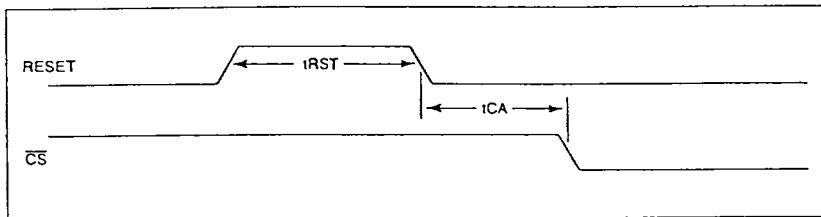


FIGURE 9. RESET TIMING

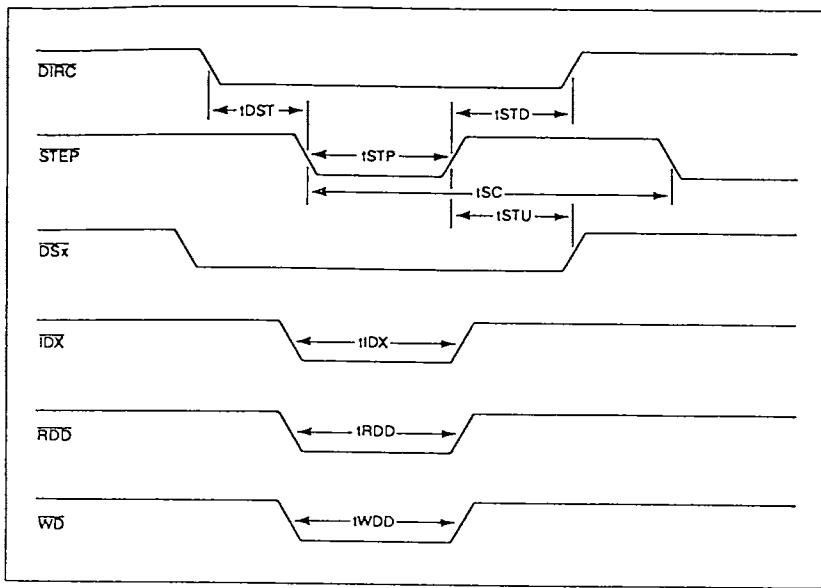


FIGURE 10. DISK DRIVE TIMING

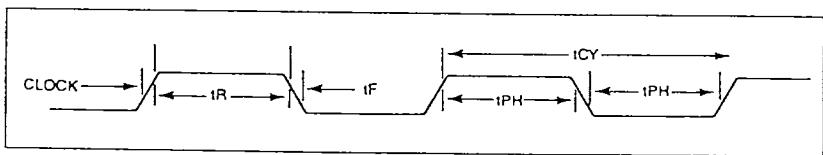


FIGURE 11. CLOCK TIMING

AC Operating Characteristics

TA = 0°C (32°F) to 70°C (158°F); VCC = +5V ± 10%

CL = 100pf

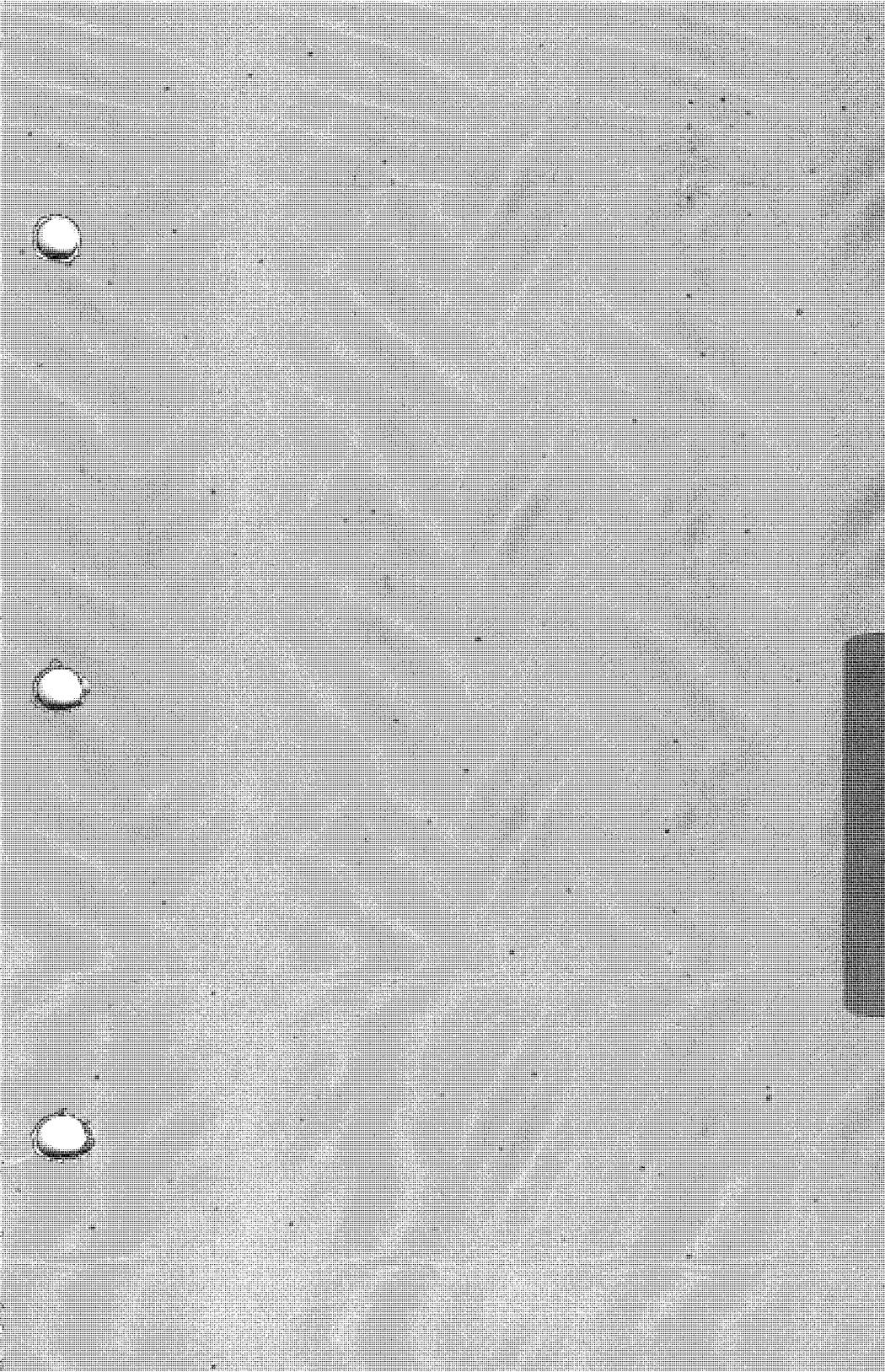
SYMBOL	PARAMETER	MIN	MAX	UNITS
tCY	Clock Period	60		nS
tPH	Clock Active (High or Low)	25		nS
tR	Clock Rise Time (Vin 0.8 to 2.0)		5	nS
tF	Clock Fall Time (Vin 2.0 to 0.8)		5	nS
tAR	AO,CS,DACK Set Up Time to RD Low	0		nS
tRA	AO,CS,DACK Hold Time to RD High	0		nS
tRR	RD Width	90		nS
tRD	Data Access Time From RD Low		90	nS
tDF	DB To Float Delay From RD High	10	65	nS
tAW	AO,CS,DACK,LDCR,LDDR, Set Up Time To WR Low	0		nS
tWA	AO,CS,DACK,LDCR,LDDR, Hold Time From WR High	0		nS
tWW	WR Width	60		nS
tDW	Data Set Up Time To WR High	80		nS
tWD	Data Hold Time From WR High	0		nS
tRI	IRQ Reset Delay Time From RD High		1MCY +150nS	
tWI	IRQ Reset Delay Time From WR High		1MCY +150nS	
tMCY	DMA Cycle Time	52		MCY
tAM	DMA Reset Delay Time From DACK Low		140	nS
tMA	DACK Delay Time From DMA High	0		nS
tAA	DACK Width	2		MCY
tTC	TC Width	60		nS
tIRST	Reset Width - TTL Driven CLK1	224		CY
tSRST	Reset Width - Software Reset	5		MCY
tRDD	RDD Active Time Low	40		nS
tWDD	WD Write Data Width Low		1/2 (TYP)	WCY
tDST	DIRC Hold & Set Up To STEP Low	4		MCY
tSTU	DSX Hold Time From STEP Low	20		MCY
tSTP	STEP Active Time Low	24		MCY
tSC	STEP Cycle Time	132		MCY
tSTD	DIRC Hold Time After STEP	96		MCY
tIDX	IDX Index Pulse Width	2		MCY
tMR	RD Delay From DMA	0		nS
tMW	WR Delay From DMA	0		nS
tMRW	RD Or WR Response From DMA High		48	MCY
tCA	Chip Access Delay From RST Low - TTL	32		MCY
tCAS	Chip Access Delay From tSRST Low	40		MCY
tXCA	Chip Access Delay from RST - OSC XT1 At 16 MHz	500		uS
tXTS	XT2 Access Delay After RST 9.6 MHz	1000		uS
tTCR	TC Delay From Last DMA Or IRO, RD	0	192	MCY
tTCW	TC Delay From Last DMA Or IRO, WR	0	384	MCY

CY Specifies CLK1 or XT1 period

MCY Specifies MCLK period, dependent on selected data rate

WCY Specifies WCLK period, dependent on selected data rate







TANDY COMPUTER PRODUCTS

4000 Power Supplies



TANDY COMPUTER PRODUCTS

4000 Astec Power Supply



TANDY COMPUTER PRODUCTS

**4000 Astec Power Supply
Contents**

Section	Page
Introduction	1
Theory of Operation	1
Switch Setting	5



AA13265 THEORY OF OPERATION

I. Introduction

The Astec P/N AA13265 is a feed forward mode off-line switching power supply accepting either 115VAC or 230VAC nominal input and delivering four regulated DC outputs at a total of 192 watts.

II. General Theory

The AC voltage from the power is routed through the EMI filter, rectified, and develops approximately 300 VDC across the capacitive input filters. By turning on the switching transistor this 300 VDC is applied across the primary winding of the power transformer. Due to the phasing of the power transformer windings, the output rectifiers are forward biased during the primary conduction time. Energy is transferred to the output filters and then to the load.

The control circuit senses the +5 VDC and the +12 VDC outputs. It determines the point at which the turn-off circuitry disables the switching transistor.

The protection circuitry senses over and under voltage conditions on all four outputs, over current on the +/-12 VDC rails, and also generates a power good signal. Should the protection circuitry sense a fault, it shuts down the power converter until the power is recycled.

A. EMI Filter

The EMI Filter consists of a common mode choke (T201), differential mode chokes (L201, L202), line to line caps (C201, C202), and line and neutral to ground caps (C203, C204). The purpose of this circuit is to suppress conducted Electro Magnetic Interference which is being fed back into the AC mains by the power supply and the logic systems connected to the power supply.

B. Surge Protection

Because the AA13265 has a capacitive input filter, input surges can be very high due to the fact that the input capacitors act like a short circuit when power is first applied. Thermistors are designed into the input circuit to limit the turn-on surge. As current passes thru the thermistors (R202, R203, R204, R205) they heat up and consequently their resistance drops to near zero.

C. Capacitive Input Filter and Voltage Select

C205 and C206 make up the energy storage from which the power stage draws to deliver energy to the loads. These two capacitors are wired in series which allows us to tap off from the center point. When the voltage select is in the 230VAC position this center cap is left

unconnected. In this situation, the diode bridge is connected as a full wave bridge rectifier and produces a DC voltage equivalent to 1.414 times the input voltage (approximately 300VDC). When the voltage select is in the 115VAC position the center tap is connected to the neutral line. This converts the input capacitor array into a capacitive doubler, charging C206 down approximately 150VDC during the negative half-cycle of the AC waveform and charging C205 up 150VDC during the positive half-cycle of the AC waveform. This results in B+ voltage across the full capacitor array of approximately 300VDC. Consequently, the primary winding and switching transistor see the same B+ operating voltage at both voltage select positions.

D. Power Conversion

As explained in Section II, the power transistor (Q11) turns on, drawing current through the primary winding to return, developing a magnetic field with a linear current ramp. Please notice the dot location on this winding. When Q11 is on, the polarity is such that the dot and of the winding is positive. The secondary windings have the same polarity. With dots marking the positive end of the windings at this instant of the cycle, the output rectifiers (D12, D13) are forward biased and conduct. Thus the energy being stored in the input caps is transferred to the output filters and the load.

A common core output choke (L2) is placed between the output rectifiers (D12,D13) and the output filters. This choke serve two purposes. One, it stores energy delivered from T1 in order to help compensate for load surges on the +12V and -5V outputs. Two, it acts as a transformer sourcing energy for the -12V and -5V outputs. The -12V and -5V outputs are derived from separate windings on L2.

The output filters are of Pi filter configuration with inductors preceded and followed by capacitors. For example, on the +5V output the filter consists of C19 followed by L4 followed by C26. The +12V and -12V outputs have similar filters. The -5V output is via a 3T regulator (IC3).

E. Control Circuit

The control circuit consists of an Astec developed chip (IC2), a pulse transformer (T2), and associated timing and reference generating components. IC2 compares the output with a generated reference voltage. When the output starts to rise beyond a preset limit, IC2 feeds a pulse thru the primary of T2. This pulse is transferred to the turn-off circuitry in the base control of switch transistor (Q11). Consequently, Q11 is quickly turned off preventing further energy transfer to the secondary of T1 on that cycle.

F. Turn-on/Turn-off Circuits

This section of the circuit consists of seven different parts.

1. The clock consists of Q6 and Q7 along with associated components. The clock timing is established by R12 and C4.

2. The clock is inhibited during Q11 on time by Q5. It is inhibited during Q11 recovery time by Q4.
3. When Q11 is turned on, positive feedback for base drive is supplied thru T3.
4. The normal method of turn-off for Q11 is via a pulsed signal fed back from the output thru T2 to the base of Q9. Q9 turns on and quickly robs Q11 of base drive, thereby turning Q11 off.
5. Q11 can also be turned off via the primary volt/second limiter. (R17, C6, Q8) or primary current limiter. When Q11 is conducting, a voltage is developed across R35. The voltage is directly proportional to the collector current of Q11. Should the current becomes too great, due to an overload on the secondary side of T1, the voltage across R35 will turn on Q8. This transistor then turns on Q10 and Q9, which turns off Q11.
6. If for some reason, control of Q11 can not be maintained, R23 which is a fusible resistor will open the collector current path preventing further damage to the power supply.
7. The shut down latch can also control Q11 turn-off by inhibiting the clock pulse. The latch is initially triggered from the secondary protection circuitry thru opto coupler (IC1). Q3 provides a regenerative latch in conjunction with the transistor side of IC1. R1, R2 and Z1 provide a regulated voltage source to keep the latch on until prime power is recycled. In addition, Q2 inhibits false triggering of the latch upon power supply startup.

G. Protection Circuit

As stated in the General Theory section, the protection circuit fulfills numerous functions.

1. +/-12V over current protection - +12V output current is drawn via transformer (T4). The secondary of T4 provides a voltage proportional to the output current that is rectified and then fed to the base of transistor (Q104). When Q104 turns on, it turns off Q105 which turns Q12 and the LED portion of IC1 on. The -12V current is sensed as a voltage potential across resistor (R48). This voltage is then applied to the base of Q104 in the same manner as the +12V sense voltage. Capacitor (C103) is used to provide surge ride thru capability on the two outputs.
2. Overvoltage protection (OVP) and undervoltage protection (UVP) are provided via Q101, Q102, Q103, Q104, Q105 and associated components. The precision reference for this circuit is generated by programmable zener IC101. Should any of the outputs be overvoltage (or be +5V undervoltage) this circuit will activate, Q12 and the LED portion of IC1 are turned on.

3. Power good is indicated by IC102 and its associated components. The power good signal is tied to the +5V output thru R141. This circuit supervises the AC input voltage and the four DC output voltages. When transistor (Q106) is turned on, the power good signal is low and vice versa. The power good is low when AC input voltage or any one of the four output voltages are lower than predetermined value.

H. Fan Control

A 12VDC fan is operated from the -12VDC output. This circuit contains a fan speed control consisting of a normally open thermal switch in parallel with a ten ohm resistor. Under normal operation, the switch is open and the fan operates at a low speed. Should the power supply become too warm, the switch closes, shorting out the resistor, and the fan operates at full speed. When the power supply cools down, the switch opens again, slowing down the fan. In this manner a nearly constant temperature is maintained within the power supply.

Astec Power Supply Switch Setting

Check to be sure that switch, S201,(on the power supplies' EMI/FILTER board) is set to the proper voltage for the location in which the system is to be installed.



TANDY COMPUTER PRODUCTS

4000 Tamura Power Supply



TANDY COMPUTER PRODUCTS

**4000 Tamura Power Supply
Contents**

Section	Page
Electrical Specifications	1
Mechanical Specifications	3
Block Diagram	5
Theory of Operation	7
Schematic	9



I. ELECTRICAL SPECIFICATION

1. INPUT VOLTAGE:

90 to 135 VAC, 47 to 63 Hz or
198 to 264 VAC, 47 to 63 Hz

2. INPUT SURGE CURRENT:

Limit ----- 50A max. (Cold start)

3. EFFICIENCY:

At full rated load with 120/240 VAC input at 50/60Hz.
Nominal ----- 75%
Limit ----- 60% min.

4. OUTPUT VOLTAGE:

	NOMINAL VOLTAGE	REGULATION TOLERANCE	LIMITS
Vo1	+5.10 VDC	+/- 5%	4.85 to 5.35 VDC
Vo2	-5.10 VDC	+/- 10%	-5.61 to -4.59 VDC
Vo3	+12.0 VDC	+/- 5%	11.4 to 12.6 VDC
Vo4	-12.0 VDC	+/- 10%	-13.2 to -10.8 VDC
Vo5	+12 V (FAN)	+/- 10%	10.8 to 13.2 VDC

5. OUTPUT RIPPLE AND NOISE VOLTAGE:

OUTPUT	RIPPLE AND NOISE LIMIT
Vo1	50mV P-P
Vo2	100mV P-P
Vo3	100mV P-P
Vo4	150mV P-P
Vo5	500mV P-P

Note: Ripple is defined as a composite of power line frequency component plus a high frequency component due to the power oscillator. Common mode noise which may be observed due to oscilloscope connections will be ignored.

6. OUTPUT OVER VOLTAGE PROTECTION:

The +5.1V output shall be protected from overvoltage fault conditions by crowbar circuitry that is set to trip in the range of 5.8 to 6.6V.

7. OUTPUT STABILITY:

The power supply must remain stable for any step load current change during any combination of input line voltage and output current loading between and including rated minimum and maximum values. The output loads may include capacitance of up to 1000 microfarads on the +5.1V output, up to 500 microfarads on either 12V output, and up to 50 microfarads on the -5.1V output.

8. **OUTPUT TRANSIENT RESPONSE**
 For a step load current change of the positive twelve volt output between 0.6A and 6.55A the maximum voltage excursion of the positive twelve volt output shall be 500 millivolts and of the positive five volt output shall be 150 millivolts.
9. **OUTPUT HOLDUP TIME**
 Nominal Line ----- 16 mSec. min.
 Low line ----- 10 mSec. min.
10. **OUTPUT CURRENT**
- | OUTPUT | MINIMUM LOAD | MAXIMUM LOAD |
|----------------------------|--------------|---|
| V _{o1} +5.1V | 5.0A | 19.8A |
| V _{o2} -5.1V | 0.0A | 0.3A |
| V _{o3} +12V | 0.6A | 6.55A(9.0A surge for 15 seconds) |
| V _{o4} -12V | 0.0A | 0.3A |
| V _{o5} +12V (FAN) | 0.3A | 0.75A (Thermostat Terminal shorted)
(Thermostat Terminal Open) |
11. **OUTPUT CURRENT LIMITING**
 Over current protection will prevent damage to the power supply when any output is short circuited continuously with 100 milliohms or less. Each output will be internally limited to the following levels:
- | OUTPUT | CURRENT LIMIT LEVEL |
|-----------------------|---------------------|
| V _{o1} +5.1V | 27.5A |
| V _{o2} -5.1V | 8A |
| V _{o3} +12V | 12A |
| V _{o4} -12V | 8A |
12. **ENVIRONMENTAL REQUIREMENTS**
- | | |
|-----------------------------|--------------------------------|
| Operating Temperature Range | 0 degrees C to +50 degrees C |
| Storage Temperature Range | -40 degrees C to +85 degrees C |
| Operating Humidity | 85% RH at 35 degrees C |
| Storage Humidity | 95% RH at 55 degrees C |
13. **SAFETY REQUIREMENTS**
 The P.S.U. is complied with U.L. standard 114 and complied with CSA standard C22.2 No. 154-M1983 for 120/240 VAC or 120 VAC only operation.
 The P.S.U. is complied with VDE 0806/8.81 (IEC 380) for Class I equipment for 120/240 VAC or 240 VAC only operation.
14. **HI-POT TEST**
 The high potential test shall be conducted in accordance with IEC 380, subclauses 16.1 and 16.3. To perform the HI-POT test, connect all secondary outputs and secondary ground together. Also connect the input line and neutral together. Apply the voltages indicated below between the indicated points for a period of one minute.* The HI-POT failure indicator should be set to trip at 500 micro-amperes.

Input to Output ----- 1250V RMS, 50/60Hz or 1750 VDC
 Input to Earth Ground ----- 1250V RMS, 50/60Hz or 1750 VDC

* HI-POT TEST in production line.

Input to Output & Earth Ground -- 1500V RMS, 50/60Hz or 2120 VDC
 for 2 sec.

15. LINE CONDUCTED EMI
The power supply must exceed the VDE 0871/6, 7B limit B for HF equipment and DP equipment with 3 dB margin at 10 KHz, increasing linearly to 8 dB at 0.10 MHz and with 8 dB margin for 0.10 to 30 MHz. Line conducted noise is measured at 120/240 VAC 9/60Hz input for all output loads from minimum to maximum. Line conducted EMI shall be measured in a configuration representative of the intended application.

In the case of 120 VAC only operation, the power supply must exceed the FCC part 15J for Class B computing device with 3 dB margin at 450 KHz, increasing linearly to 8 dB margin from 1.0 to 30 MHz.
16. LINE TRANSIENT
The power supply shall meet the line transient requirements of IEE 472-1974 for Common Mode and Differential Mode operation.
17. OUTPUT SIGNAL
The power supply will provide a combination "ACLOW" and "RESET" signal to be called "DCOK". DCOK will be a TTL compatible signal, high is more than 2.4 VDC, low is less than 0.4 VDC, sink is more than 1.6 milliamperes when low.
The DCOK signal will go low whenever the input line voltage drops to less than 82.5 VAC (+/-2.5 VAC) for nominal input voltage of 120 VAC or less than 165 VAC (+/-5.0 VAC) for nominal input voltage of 240 VAC.
The DCOK signal will not go high until 100 to 500 milliseconds after the +5 Volt output stabilizes at its operating value when the unit is turned on.

II. CIRCUIT DIAGRAM

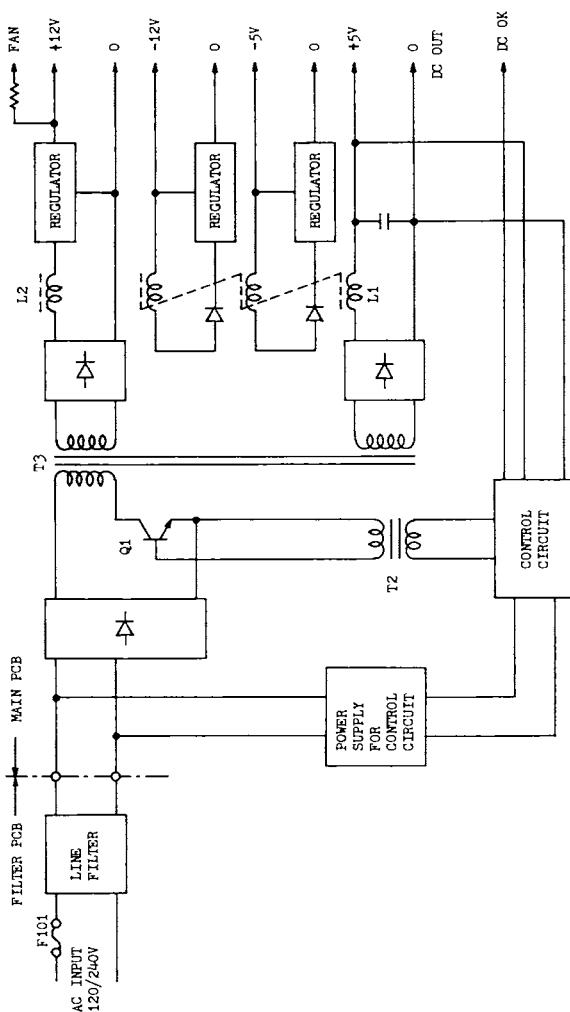
As per attached DWG. No. 3P-M1-0373, 4P-M1-0375

III. MECHANICAL SPECIFICATIONS

1. DIMENSION : As per attached DWG. No. 3P-K1-0282
2. WEIGHT : Approx. 1.3 kg



POWER SUPPLY BLOCK DIAGRAM





A. THEORY OF OPERATION

The power supply model 8790095 is a 192 watts switching power supply. This power supply is composed of a filter PCB and main PCB.

1. Filter PCB

The filter PCB is connected to the AC line input, and a fuse is on this PCB.

The EMI Filter circuit reduces noise that leaks from the power source to the AC line.

2. Main PCB

The silicon bridge D1 rectifies the AC line to DC. The thermistors reduce the charging current of capacitors C1 and C2 when power is on.

The power converter circuit is generally called the "Forward converter". The main switching transistor Q1 chops the DC voltage of C1 and C2 at 50KHz. The chopped DC voltage is then transferred by transformer T3 to the isolated voltages and rectified to the required voltage.

+12V voltage is regulated by series dropper Q4.

-5V and -12V voltages are supplied from the flyback energy of L1, and are adjusted by the voltage regulators IC1 and IC2 (package of TO-220).

In the case of over current fault, the increased voltage of the current transformer CT1 or CT2 is fed to the control PCB.

- Control PCB (sub board on the main PCB)

The control PCB contains the output voltage control circuit, +5V over voltage protection circuit, and the power good signal (to be called "DC-OK") circuit.

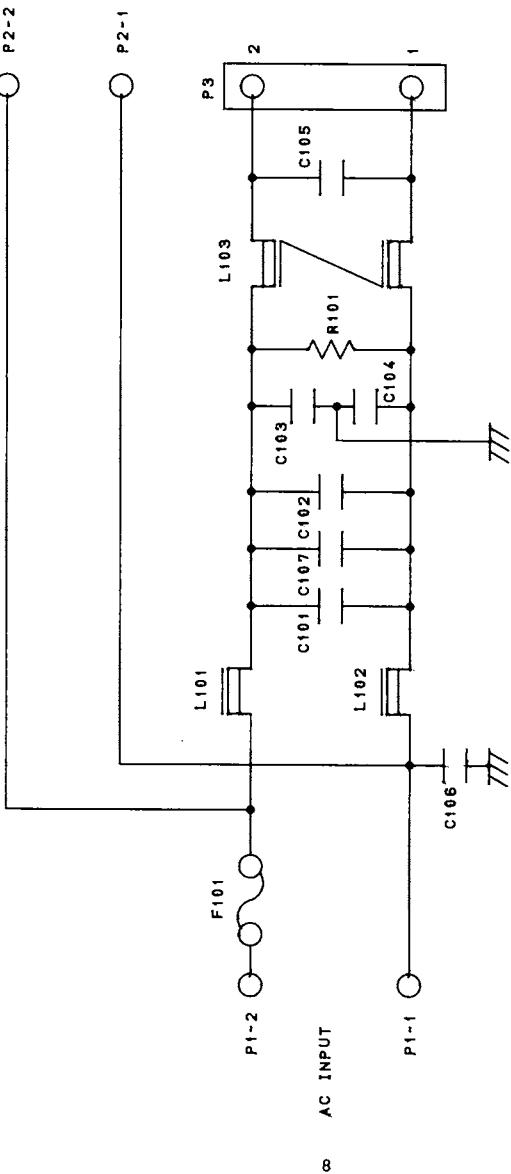
IC1 is a 16-pin DIP package which incorporates the functions of a pulse-width-modulation control circuit and is provided the voltage variation of +5V output.

The main switching transistor (on the main PCB) is controlled by the pulse from IC1.

In the case of over voltage fault, the pulse oscillator in IC1 is stopped by PUT1 under the control of zener diode D4.

IC2 is a voltage comparator IC for the power good signal circuit.

AC MONITOR OUTPUT



Model no. 8790095 LINE FILTER





TANDY COMPUTER PRODUCTS

4000 Keyboard



Keyboard Specification 20425

KEYBOARD SPECIFICATION

20425

FOR THE "101" KEYBOARD (E03370051)

James E. Katona

Engineer

L. M. Nelson

Approved

3-10-87

Date

3/11/87

Date

Keyboard Specification 20425

Keyboard Specification 20425

CONTENTS

	<u>PAGE</u>
1.0 GENERAL	5
2.0 SCOPE	5
3.0 APPLICABLE DOCUMENTS	5
4.0 MECHANICAL REQUIREMENTS	5
4.1 Switch Profiles	5
4.2 Key Layout, Legends, and Colors	5
4.3 Keystroke	9
4.3.1 Total Travel	9
4.3.2 Pre Travel	9
4.3.3 Force	9
4.3.4 Breakover Feedback	9
5.0 FUNCTIONAL REQUIREMENTS	9
5.1 Scan Codes	9
5.2 Protocol	23
5.2.1 Communication Mode 1 (PC/XT)	23
5.2.2 Communication Mode 2 (AT)	24
5.2.3 Commands From The Keyboard To The System .	25
5.2.4 Commands From The System To The Keyboard .	27
5.3 Key Rollover	31
5.4 Power-Up and Self Test	31
5.5 Autorepeat	31
5.6 Buffering	31
5.7 Mode Switch Settings	32
5.8 Auto-Discrimination	32
5.9 LED Indicators	33
5.10 Software "Watch Dog"	33
5.0 ELECTRICAL REQUIREMENTS	33
6.1 Signal Connector	34
6.2 Chassis Ground	34
6.3 Power Requirement	34

Keyboard Specification 20425

	<u>PAGE</u>
7.0 ENVIRONMENTAL REQUIREMENTS	34
7.1 Temperature	34
7.2 Relative Humidity	34
7.3 Shock	34
7.4 Vibration	34
7.5 Altitude	34
7.6 Electrostatic Immunity (ESD)	35
7.7 EMI/RFI	35
7.8 Safety	35
8.0 RELIABILITY	35
8.1 Mean Time Before Failure (MTBF)	35
8.2 Switch Life	35
8.3 Maintainability	35

Keyboard Specification 20425

1.0 GENERAL

The "101" keyboard (EO3370051) shall be a direct, plug compatible replacement for the 101 IBM* PC, XT, AT, and compatible personal computer keyboards. No software modification or special interfaces shall be needed by the user.

2.0 SCOPE

This specification defines the functional, mechanical, electrical, environmental, and reliability characteristics of the EO3370051 keyboard.

Specifically, the keyboard is encoded such that all keys produce a unique output code upon switch depression and a similar but unique output code upon switch release. The communication with the host system is via a synchronous serial link. The keytop layout, switch encoding, and serial communication are all compatible with, but not identical to, the IBM PC, XT, and AT keyboards.

3.0 APPLICABLE DOCUMENTS

0115562 Factory Test Procedure
16-02464 ESD Test Procedure, Part 1B

4.0 MECHANICAL REQUIREMENTS

4.1 Switch Profiles

Figure 1 shows the keyboard profile.

4.2 Keytop Layout, Legends And Colors

Figure 2 illustrates the keytop layout and appropriate legends.

Table 1 specifies the color for each keytop. The Key Tronic color codes used are:

WA - Fog (Marbon P/N 2500)
GE - Sea Mist (Marbon P/N 20779).

* IBM PC, XT, and AT are trademarks of the International Business Machine Corporation.

Keyboard Specification 20425

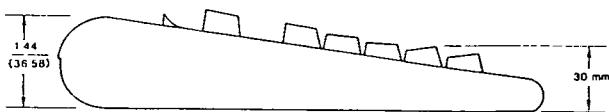


Figure 1: Keyboard Profile

Keyboard Specification 20425

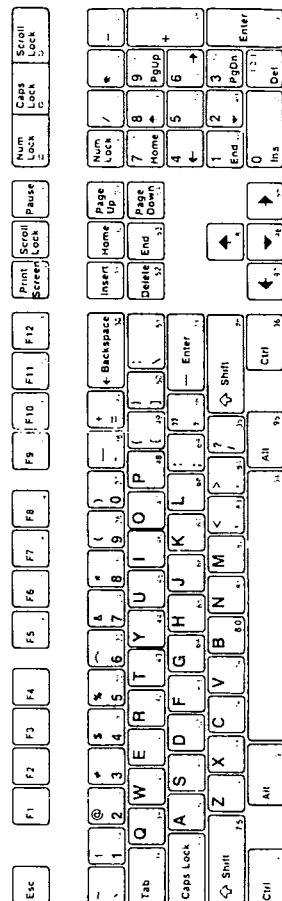


Figure 2: Keypad Layout and Legends

Keyboard Specification 20425

TABLE 1

KEYTOP COLOR CODES

<u>Key</u>	<u>Station</u>	<u>Color</u>	<u>Key</u>	<u>Station</u>	<u>Color</u>	<u>Key</u>	<u>Station</u>	<u>Color</u>
1		GE	34		GE	67		WA
2		WA	35		GE	68		WA
3		WA	36		GE	69		WA
4		WA	37		GE	70		WA
5		WA	38		GE	71		GE
6		GE	39		WA	72		WA
7		GE	40		WA	73		WA
8		GE	41		WA	74		WA
9		GE	42		WA	75		GE
10		WA	43		WA	76		WA
11		WA	44		WA	77		WA
12		WA	45		WA	78		WA
13		WA	46		WA	79		WA
14		GE	47		WA	80		WA
15		GE	48		WA	81		WA
16		GE	49		WA	82		WA
17		WA	50		WA	83		WA
18		WA	51		WA	84		WA
19		WA	52		GE	85		WA
20		WA	53		GE	86		GE
21		WA	54		GE	87		GE
22		WA	55		WA	88		WA
23		WA	56		WA	89		WA
24		WA	57		WA	90		WA
25		WA	58		GE	91		GE
26		WA	59		GE	92		GE
27		WA	60		WA	93		GE
28		WA	61		WA	94		WA
29		WA	62		WA	95		GE
30		GE	63		WA	96		GE
31		GE	64		WA	97		GE
32		GE	65		WA	98		GE
33		GE	66		WA	99		GE
						100		WA
						101		WA

Keyboard Specification 20425

4.3 Keystitch

4.3.1 Total Travel = 0.150" \pm .020"

4.3.2 Pre Travel = 0.075" \pm 0.015"

4.3.3 Force - All keyswitches shall utilize a 2.0 oz. rubber sleeve.

4.3.4 Breakover Feedback - The keyswitches shall utilize a sensory feedback to assure the operator that the key has been depressed beyond its operating point.

5.0 FUNCTIONAL REQUIREMENTS

5.1 Scan Codes

The keyboard shall generate a unique Hex scan code for each and every keystitch including codes for both depression (make) and release (break). For the AT, the break code is the same as the make code preceded by "FO". The keystitch to scan code assignments are illustrated and listed on the following pages:

- Figure 3 shows the PC/XT scan codes and a detailed listing is provided in Table 2 (Scan Set 1).
- Figure 4 shows the AT scan codes and detailed listings are provided in Table 3 (Scan Set 2) and Table 4 (Scan Set 3).

Keyboard Specification 20425

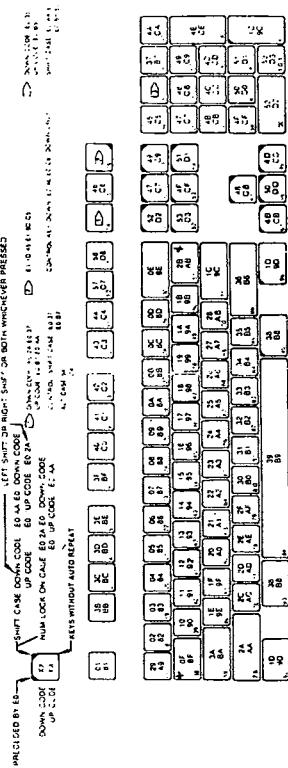


Figure 3: PC/XT Scan Codes

Keyboard Specification 20425

TABLE 2
SCAN SET 1, XT SCAN CODES

KEY NUMBER	MAKE CODE	BREAK CODE
1	01	81
2	3B	BB
3	3C	BC
4	3D	BD
5	3E	BE
6	3F	BF
7	40	CO
8	41	C1
9	42	C2
10	43	C3
11	44	C4
12	57	D7
13	58	D8
15	46	C6
17	29	A9
18	02	82
19	03	83
20	04	84
21	05	85
22	06	86
23	07	87
24	08	88
25	09	89
26	0A	8A
27	0B	8B
28	0C	8C
29	0D	8D
30	0E	8E
34	45	C5
36	37	B7
37	4A	CA
38	0F	8F
39	10	90
40	11	91
41	12	92
42	13	93
43	14	94
44	15	95
45	16	96
46	17	97
47	18	98
48	19	99
49	1A	9A
50	1B	9B

TABLE 2 (Continued)

SCAN SET 1, XT SCAN CODES

<u>KEY NUMBER</u>	<u>MAKE CODE</u>	<u>BREAK CODE</u>
51	2B	AB
55	47	C7
56	48	C8
57	49	C9
58	4E	CE
59	3A	BA
60	1E	9E
61	1F	9F
62	20	A0
63	21	A1
64	22	A2
65	23	A3
66	24	A4
67	25	A5
68	26	A6
69	27	A7
70	28	A8
71	1C	9C
72	4B	Cb
73	4C	CC
74	4D	CD
75	2A	AA
76	2C	AC
77	2D	AD
78	2E	AE
79	2F	AF
80	30	B0
81	31	B1
82	32	B2
83	33	B3
84	34	B4
85	35	B5
86	36	B6
88	4F	CF
89	50	D0
90	51	D1
91	E0 1C	E0 9C
92	1D	9D
93	38	B8
94	39	B9
95	E0 38	E0 B8
96	E0 1D	E0 9D
100	52	D2
101	53	D3

Keyboard Specification 20425

TABLE 2 (Continued)
SCAN SET 1, XT SCAN CODES

KEY NUMBER	BASE CASE, OR SHIFT+NUM LOCK MAKE/BREAK	SHIFT CASE MAKE/BREAK	NUM LOCK ON MAKE/BREAK
31	E0 52 /E0 D2	E0 AA E0 52 /E0 D2 E0 2A	E0 2A E0 52 /E0 D2 E0 AA
32	E0 47 /E0 C7	E0 AA E0 47 /E0 C7 E0 2A	E0 2A E0 47 /E0 C7 E0 AA
33	E0 49 /E0 C9	E0 AA E0 49 /E0 C9 E0 2A	E0 2A E0 49 /E0 C9 E0 AA
52	E0 53 /E0 D3	E0 AA E0 53 /E0 D3 E0 2A	E0 2A E0 53 /E0 D3 E0 AA
53	E0 4F /E0 CF	E0 AA E0 4F /E0 CF E0 2A	E0 2A E0 4F /E0 CF E0 AA
54	E0 51 /E0 D1	E0 AA E0 51 /E0 D1 E0 2A	E0 2A E0 51 /E0 D1 E0 AA
87	E0 48 /E0 C8	E0 AA E0 48 /E0 C8 E0 2A	E0 2A E0 48 /E0 C8 E0 AA
97	E0 4B /E0 CB	E0 AA E0 4B /E0 CB E0 2A	E0 2A E0 4B /E0 CB E0 AA
98	E0 50 /E0 D0	E0 AA E0 50 /E0 D0 E0 2A	E0 2A E0 50 /E0 D0 E0 AA
99	E0 4D /E0 CD	E0 AA E0 4D /E0 CD E0 2A	E0 2A E0 4D /E0 CD E0 AA

If the left Shift key is held down, the AA / 2A shift break/make codes are sent with the other scan codes. If the right Shift key is held down, B6 / 36 is sent. If both Shift keys are down, both sets of codes are sent with the other scan code.

Keyboard Specification 20425

TABLE 2 (continued)
SCAN SET 1, XT SCAN CODES

KEY NUMBER	SCAN CODE MAKE / BREAK	SHIFT CASE MAKE / BREAK
35	EO 35/EO B5	EO AA EO 35 / EO B5 EO 2A

If the left Shift key is held down, the AA / 2A shift make/break codes are sent with the other scan codes. If the right Shift key is held down, B6 / 36 is sent. If both Shift keys are down, both sets of codes are sent with the other scan codes.

KEY NUMBER	SCAN CODE MAKE / BREAK	CTRL CASE, SHIFT CASE MAKE / BREAK	ALT CASE MAKE / BREAK
14	EO 2A EO 37 /EO B7 EO AA	EO 37 / EO E7	S4 / D4

KEY NUMBER	MAKE CODE	CTRL KEY PRESSED
16	E1 1D 45 E1 9D C5	EO 46 EO C6

This key is not typematic. All associated scan codes occur on the make of the key.

Keyboard Specification 20425

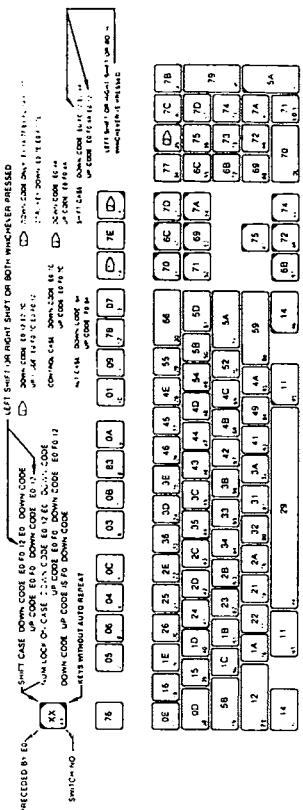


Figure 4: AT Scan Codes

Keyboard Specification 20425

TABLE 3
SCAN SET 2, AT SCAN CODES

KEY NUMBER	MAKE CODE	BREAK CODE
1	76	F0 76
2	05	F0 05
3	06	F0 06
4	04	F0 04
5	0C	F0 0C
6	03	F0 03
7	0B	F0 0B
8	83	F0 83
9	0A	F0 0A
10	01	F0 01
11	09	F0 09
12	78	F0 78
13	07	F0 07
15	7E	F0 7E
17	0E	F0 0E
18	16	F0 16
19	1E	F0 1E
20	26	F0 26
21	25	F0 25
22	2E	F0 2E
23	36	F0 36
24	3D	F0 3D
25	3E	F0 3E
26	46	F0 46
27	45	F0 45
28	4E	F0 4E
29	55	F0 55
30	66	F0 66
34	77	F0 77
36	7C	F0 7C
37	7B	F0 7B
38	0D	F0 0D
39	15	F0 15
40	1D	F0 1D
41	24	F0 24
42	2D	F0 2D
43	2C	F0 2C
44	35	F0 35
45	3C	F0 3C
46	43	F0 43
47	44	F0 44
48	40	F0 40
49	54	F0 54
50	5B	F0 5B

Keyboard Specification 20425

TABLE 3
SCAN SET 2, AT SCAN CODES

<u>KEY NUMBER</u>	<u>MAKE CODE</u>	<u>BREAK CODE</u>
51	5D	F0 5D
55	6C	F0 6C
56	75	F0 75
57	7D	F0 7D
58	79	F0 79
59	58	F0 58
60	1C	F0 1C
61	1B	F0 1B
62	23	F0 23
63	2B	F0 2B
64	34	F0 34
65	33	F0 33
66	3B	F0 3B
67	42	F0 42
68	4B	F0 4B
69	4C	F0 4C
70	52	F0 52
71	5A	F0 5A
72	6B	F0 6B
73	73	F0 73
74	74	F0 74
75	12	F0 12
76	1A	F0 1A
77	22	F0 22
78	21	F0 21
79	2A	F0 2A
80	32	F0 32
81	31	F0 31
82	3A	F0 3A
83	41	F0 41
84	49	F0 49
85	4A	F0 4A
86	59	F0 59
88	69	F0 69
89	72	F0 72
90	7A	F0 7A
91	E0 5A	E0 F0 5A
92	14	F0 14
93	11	F0 11
94	29	F0 29
95	E0 11	E0 F0 11
96	E0 14	E0 F0 14
100	70	F0 70
101	71	F0 71

Keyboard Specification 20425

TABLE 3 (Continued)
SCAN SET 2, AT SCAN CODES

KEY NUMBER	BASE CASE, OR SHIFT+NUM LOCK MAKE/BREAK	SHIFT CASE MAKE/BREAK	NUM LOCK ON MAKE/BREAK
31	EO 70 / EO FO 70	EO FO 12 EO 70 /EO FO 70 EO 12	EO 12 EO 70 /EO FO 70 EO FO 12
32	EO 6C / EO FO 6C	EO FO 12 EO 6C /EO FO 6C EO 12	EO 12 EO 6C /EO FO 6C EO FO 12
33	EO 7D / EO FO 7D	EO FO 12 EO 7D /EO FO 7D EO 12	EO 12 EO 7D /EO FO 7D EO FO 12
52	EO 71 / EO FO 71	EO FO 12 EO 71 /EO FO 71 EO 12	EO 12 EO 71 /EO FO 71 EO FO 12
53	EO 69 / EO FO 69	EO FO 12 EO 69 /EO FO 69 EO 12	EO 12 EO 69 /EO FO 69 EO FO 12
54	EO 7A / EO FO 7A	EO FO 12 EO 7A /EO FO 7A EO 12	EO 12 EO 7A /EO FO 7A EO FO 12
87	EO 75 / EO FO 75	EO FO 12 EO 75 /EO FO 75 EO 12	EO 12 EO 75 /EO FO 75 EO FO 12
97	EO 6B / EO FO 6B	EO FO 12 EO 6B /30 FO 6B EO 12	EO 12 EO 6B /EO FO 6B EO FO 12
98	EO 72 / EO FO 72	EO FO 12 EO 72 /EO FO 72 EO 12	EO 12 EO 72 /EO FO 72 EO FO 12
99	EO 74 / EO FO 74	EO FO 12 EO 74 /EO FO 74 EO 12	EO 12 EO 74 /EO FO 74 EO FO 12

If the left Shift key is held down, the FO 12/12 shift make/break codes are sent with the other scan codes. If the right Shift key is held down, FO 59/59 is sent. If both Shift keys are down, both sets of codes are sent with the other scan code.

Keyboard Specification 20425

TABLE 3 (Continued)

SCAN SET 2, AT SCAN CODES

KEY NUMBER	SCAN CODE MAKE / BREAK	SHIFT CASE MAKE / BREAK
35	EO 4A/EO FO 4A	EO FO 12 EO 4A / EO FO 4A EO 12

If the left Shift key is held down, the FO 12/12 shift make/break codes are sent with the other scan codes. If the right Shift key is held down, FO 59/59 is sent. If both Shift keys are down, both sets of codes are sent with the other scan codes.

KEY NUMBER	SCAN CODE MAKE / BREAK	CTRL CASE, SHIFT CASE MAKE / BREAK	ALT CASE MAKE / BREAK
14	EO 12 EO 7C /EO FO 7C EO FO 12	EO 7C / EO FO 7C	84/FO 84

KEY NUMBER	MAKE CODE	CTRL KEY PRESSED
16	E1 14 77 E1 FO 14 FO 77	EO 7E EO FO 7E

This key is not typematic. All associated scan codes occur on the make of the key.

TABLE 4
SCAN SET 3, AT DEFAULT KEY STATE

KEY NUMBER	MAKE CODE	BREAK CODE	DEFAULT KEY STATE
1	08	F0 08	Make only
2	07	F0 07	Make only
3	0F	F0 0F	Make only
4	17	F0 17	Make only
5	1F	F0 1F	Make only
6	27	F0 27	Make only
7	2F	F0 2F	Make only
8	37	F0 37	Make only
9	3F	F0 3F	Make only
10	47	F0 47	Make only
11	4F	F0 4F	Make only
12	56	F0 56	Make only
13	5E	F0 5E	Make only
14	57	F0 57	Make only
15	5F	F0 5F	Make only
16	62	F0 62	Make only
17	0E	F0 0E	Typematic
18	16	F0 16	Typematic
19	1E	F0 1E	Typematic
20	26	F0 26	Typematic
21	25	F0 25	Typematic
22	2E	F0 2E	Typematic
23	36	F0 36	Typematic
24	3D	F0 3D	Typematic
25	3E	F0 3E	Typematic
26	46	F0 46	Typematic
27	45	F0 45	Typematic
28	4E	F0 4E	Typematic
29	55	F0 55	Typematic
30	66	F0 66	Typematic
31	67	F0 67	Make only
32	6E	F0 6E	Make only
33	6F	F0 6F	Make only
34	76	F0 76	Make only
35	77	F0 77	Make only
36	7E	F0 7E	Make only
37	84	F0 84	Make only
38	0D	F0 0D	Typematic
39	15	F0 15	Typematic
40	1D	F0 1D	Typematic
41	24	F0 24	Typematic
42	2D	F0 2D	Typematic
43	2C	F0 2C	Typematic
44	35	F0 35	Typematic

Keyboard Specification 20425

TABLE 4 (Continued)

SCAN SET 3, AT DEFAULT KEY STATE

<u>KEY NUMBER</u>	<u>MAKE CODE</u>	<u>BREAK CODE</u>	<u>DEFAULT KEY STATE</u>
45	3C	F0 3C	Typematic
46	43	F0 43	Typematic
47	44	F0 44	Typematic
48	4D	F0 4D	Typematic
49	54	F0 54	Typematic
50	5B	F0 5B	Typematic
51	5C	F0 5C	Typematic
52	64	F0 64	Typematic
53	65	F0 65	Make only
54	6D	F0 6D	Make only
55	6C	F0 6C	Make only
56	75	F0 75	Make only
57	7D	F0 7D	Make only
58	7C	F0 7C	Typematic
59	14	F0 14	Make/Break
60	1C	F0 1C	Typematic
61	1B	F0 1B	Typematic
62	23	F0 23	Typematic
63	2B	F0 2B	Typematic
64	34	F0 34	Typematic
65	33	F0 33	Typematic
66	3B	F0 3B	Typematic
67	42	F0 42	Typematic
68	4B	F0 4B	Typematic
69	4C	F0 4C	Typematic
70	52	F0 52	Typematic
71	5A	F0 5A	Typematic
72	6B	F0 6B	Make only
73	73	F0 73	Make only
74	74	F0 74	Make only
75	12	F0 12	Make/Break
76	1A	F0 1A	Typematic
77	22	F0 22	Typematic
78	21	F0 21	Typematic
79	2A	F0 2A	Typematic
80	32	F0 32	Typematic
81	31	F0 31	Typematic
82	3A	F0 3A	Typematic
83	41	F0 41	Typematic
84	49	F0 49	Typematic
85	4A	F0 4A	Typematic
86	59	F0 59	Make/Break
87	63	F0 63	Typematic
88	69	F0 69	Make only

Keyboard Specification 20425

TABLE 4 (Continued)

SCAN SET 3, AT DEFAULT KEY STATE

<u>KEY NUMBER</u>	<u>MAKE CODE</u>	<u>BREAK CODE</u>	<u>DEFAULT KEY STATE</u>
89	72	F0 72	Make only
90	7A	F0 7A	Make only
91	79	F0 79	Make only
92	11	F0 11	Make/Break
93	19	F0 19	Make/Break
94	29	F0 29	Typematic
95	39	F0 39	Make only
96	58	F0 58	Make only
97	61	F0 61	Typematic
98	60	F0 60	Typematic
99	6A	F0 6A	Typematic
100	70	F0 70	Make only
101	71	F0 71	Make only

Keyboard Specification 20425

5.2 Protocol

5.2.1 Communication Mode 1 (PC/XT)

The keyboard shall communicate with the system using synchronous serial protocol. When no communication is in process, the keyboard holds the data line low and the clock line high. Transmissions consist of a 10 bit data word shown as follows (terminology assumes positive log c.):

0	1	x	x	x	x	x	x	x	x
Keyboard	Start	B0	B1	B2	B3	B4	B5	B6	B7

The system may hold the clock line low for a minimum of 12.5 milliseconds to initiate a keyboard reset (see Power-Up Sequence). The keyboard shall not attempt to transmit data while the clock line is being held by the system.

Before initiating a transmission, the keyboard lowers the clock line as a Request To Send (RTS). The keyboard then checks the state of the data line. If the system is holding the data line low, then the keyboard interface is inhibited. The keyboard shall retain the keycode in the buffer, return the clock and data lines to the idle state, and resume scanning until the interface is enabled.

If the interface is enabled, the keyboard shall transmit its data in the previously described format. Data is valid during the time that the clock is high and for a minimum of 2.5 microseconds after the falling edge of the clock. See Figure 5 for the timing diagram.

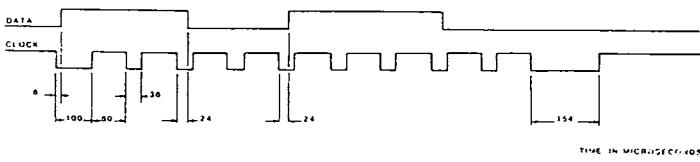


Figure 5: PC/XT Timing Diagram

Keyboard Specification 20425

5.2.2 Communication Mode 2 (AT)

The keyboard shall communicate with the system using synchronous serial protocol. This communication is bidirectional, with the keyboard clocking all data transfers (the terminology assumes positive logic).

When there is no communication in process, the clock and data lines shall idle in the high state. Transmissions consist of an 11 bit data word shown below and in Figure 6.

0	x	x	x	x	x	x	x	x	P	1	
Start	B0	B1	B2	B3	B4	B5	B6	B7	Parity	Stop	
(Low)	(LSB)								(MSB)	(Odd)	(High)

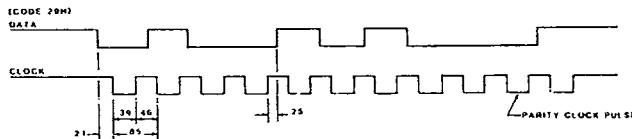


Figure 8A: Keyboard to System

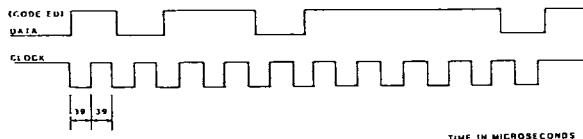


Figure 68 System to Keyboard

Figure 6: AT Timing Diagram

Keyboard Specification 20425

5.2.3 Commands From The Keyboard To The System

Prior to initiating a transmission of a code, the keyboard shall first check the state of the clock and data lines. If the clock line is low, then the keyboard interface is inhibited, and the code shall be kept in the keyboard buffer until the inhibit is removed. If the clock line is high and the data line is low, then the host is issuing an RTS, and the keyboard shall keep the code in the keyboard buffer and prepare to receive data.

If the keyboard interface is enabled and no RTS is detected, then the keyboard shall initiate a transmission by sending a low start bit, followed by the rest of the code. Keyboard data shall be valid prior to the falling edge of the clock and after the rising edge of the clock. See Figure 6A for the timing diagram.

During the transmission of a code, the keyboard shall check the state of the clock line at intervals of not less than 60 microseconds. If the clock line is detected low prior to the rising edge of the parity bit clock, then a data collision occurs. The keyboard will stop transmission, place the code back in the keyboard buffer, and prepare to respond to the next action by the system.

The keyboard shall send commands to the system. The commands and their function are listed as follows:

00 hex - Keyboard Buffer Overrun, Scan Sets 2 or 3

If the keyboard buffer overflows, and the keyboard is currently in Mode 2, then it will issue a hex 00 to indicate this condition.

AA hex - Self test passed

This command is issued after successful completion of the keyboard self test. The self test may be initiated by a Reset command from the system, or by a Power On Reset.

EE hex - Echo

The Echo command is sent in response to an Echo command from the system.

FA hex - Acknowledge

The keyboard sends an Acknowledge in response to any valid command from the system except for Resend and Echo.

Keyboard Specification 20425

FC hex - Self Test Failure

This response is issued by the keyboard in place of the AA hex if the keyboard detects a problem during its self test. Refer to Power-Up and Self Test, Paragraph 5.4.

FE hex - Resend

The keyboard issues a Resend command in response to inputs which have parity or framing errors, or if the input is invalid.

FF hex - Keyboard Buffer Overrun, Scan Set 1

If the keyboard is in Communication Mode 1 and the keycode buffer overflows, then it indicates this situation to the system by transmitting FF hex to the host.

83AB hex - Keyboard ID

The keyboard ID is a 2 byte number issued in response to a Read ID command from the system. The keyboard responds to a Read ID with an Acknowledge, followed by the ID bytes. After the output of the ID bytes, the keyboard begins scanning.

Keyboard Specification 20425

5.2.4 Commands From The System To The Keyboard

Before the system can transmit data to the keyboard, it must first check to see if the keyboard is sending data. If the keyboard is sending, and the data is past the parity clock pulse, then the system must accept that data prior to initiating its own transmission.

If the keyboard data string has not yet reached the tenth clock pulse, or it is not currently transmitting data, then the system takes control by lowering the clock line for at least 60 microseconds, then releasing it after clamping the data line low for a start bit. The keyboard will respond to the RTS within 5 milliseconds by clocking the start bit in. The keyboard will continue clocking data as shown in the timing diagram (Figure 6B). The system should insure that the data is valid before the rising edge of the keyboard clock pulse and after the falling edge.

After the parity bit the system should raise the data line for a stop bit. The keyboard shall check for the "1" stop bit, then clamp the data line low prior to clocking the stop bit. This action signals the system that the keyboard has received its data. If the system has not raised the data line, then the keyboard receives a framing error and shall continue to clock until the data line is raised by the system. After receiving either a framing or parity error the keyboard will respond by issuing a RESEND.

All commands from the system require some sort of a response from the keyboard. This keyboard response shall occur within 20 milliseconds of the receipt of the command.

The following commands may be sent to the keyboard at any time, following the protocol described in Paragraph 5.2.3. These commands are only valid in Communication Mode 2. The keyboard shall issue a response within 20 milliseconds of the receipt of any of these commands except for Reset. The commands and their function are listed as follows:

ED hex - Set Status Indicators

The keyboard shall respond to this command with an Acknowledge, stop scanning and wait for the status byte. The status byte has a bit for each of the LED's on the keyboard. If the bit is set (1) then the LED is on, if it is clear (0) then the corresponding LED is off. Default disable and Set disable commands do not affect status indicators.

Keyboard Specification 20425

Bit 0 corresponds to the Scroll Lock indicator, bit 1 is Num Lock and bit 2 is Caps Lock. The rest of the byte must be zeros.

After responding to the command the keyboard shall resume scanning if it was previously enabled. If, instead of a status byte, the system follows the Set Status Indicators command with a valid command, no change to the LED's occur and the keyboard executes the new command instead.

EE hex - Echo

This is provided for diagnostic purposes. The keyboard shall respond with EE instead of Acknowledge.

F0 hex - Select Scan Set

This is used to select one of the three Scan Sets or tell the system which Scan Set is currently in use. After receiving this command, the keyboard shall transmit an Acknowledge to the system. The system then sends the request byte and the keyboard transmits an Acknowledge. A request byte value of 00 hex will cause the keyboard to respond with the Scan Set currently in use. A request byte value of 01 hex selects Scan Set 1, 02 hex selects Scan Set 2, 03 hex selects Scan Set 3.

The keyboard returns to the scanning state it was in before the select Scan Set.

F2 hex - Read ID

This command causes the keyboard to send two identification bytes (83AB). The keyboard shall send an Acknowledge and stop scanning. Then the two ID bytes are sent. The second byte follows the first byte within 500 microseconds. After both identification bytes are sent the keyboard begins scanning.

F3 hex - Set Typematic Values

This is a two-byte command to change the typematic rate and delay, with each byte answered by the keyboard with an Acknowledge. The second byte is the delay byte and defines both the delay before typematic action as well as the typematic rate. Bits 0 (LSB), 1, 2, 3 and 4 define the typematic rate by the following equation:

$$\text{Period} = (8+B0) * (2^{**B1}) * 0.00417$$

** indicates exponentiation

B0 = Binary value of bits 2, 1, and 0
B1 = Binary value of bits 4 and 3

Keyboard Specification 20425

The delay is defined as follows:

$$T = (B3+1) * 250 \text{ milliseconds}$$

B3 = Binary value of bits 6 and 5
Bit 7 (MSB) is always 0

F4 hex - Enable

This command enables the keyboard. The keyboard shall respond with an Acknowledge, clear the output buffer, clear the last repeating key, and begin scanning.

F5 hex - Default Disable

The keyboard stops scanning and resets to default status. The keyboard transmits an Acknowledge, clears the output buffer, sets the default key types for Scan Set 3, sets default repeat rate/delay, and clears last repeating key.

F6 hex - Set Default

This command resets the keyboard to Power-Up state. The keyboard responds with an Acknowledge, clears the output buffer, set default key types for Scan Set 3, sets default repeat rate/delay, and clears the last typematic key.

F7 hex - Set All Keys - Typematic

F8 hex - Set All Keys - Make/Break

F9 hex - Set All Keys - Make

FA hex - Set All Keys - Typematic/make/Break

See common explanation below.

The keyboard shall send an Acknowledge, clear the output buffer, set all keys to the type requested by the command, and continue scanning if it was previously enabled. These commands can be sent using any scan set but affect Scan Set 3 operation only.

FB hex - Set Key Type - Typematic

FC hex - Set Key Type - Make/Break

FD hex - Set Key Type - Make

See common explanation below.

The keyboard shall send an Acknowledge, clear the output buffer, and wait for the key scan code. The key scan code identifies which key will be set to the function requested by the command. Only Scan Set

Keyboard Specification 20425

3 values are valid. Only Scan Set 3 operation will be affected, but command can be sent using any scan set.

FF hex - Resend

After receiving this command, the keyboard shall transmit the last byte sent. If the last byte was Resend, the keyboard shall transmit the previous byte sent before the Resend command.

FF hex - Reset

After receiving this command, the keyboard shall transmit an Acknowledge to the system. The keyboard shall wait till the system accepts the Acknowledge response. The system will accept the Acknowledge by raising the CLOCK and DATA lines for a minimum of 500 microseconds. The keyboard shall be disabled until system accepts Acknowledge or a new command is sent.

The keyboard than executes the self test routine. (See Power-Up and Self Test, Paragraph 5.4. After the power up routines, the keyboard shall be in the default state.

EF hex - Invalid Command

F1 hex - Invalid Command

} See common explanation below.

After receiving the EF or F1 command, the keyboard shall transmit Resend.

Keyboard Specification 20425

5.3 Key Rollover

The keyboard shall incorporate N-Key Rollover (NKRO) to avoid loss of keystroke data during high speed entry. NKRO is defined as all key depressions and releases correctly detected in any sequence regardless of how many keys are being held depressed.

5.4 Power-Up and Self Test

The keyboard shall contain reset circuitry. The duration of the keyboard Power-On Reset (POR) shall be greater than 150 and less than 2000 milliseconds from power-up.

After executing POR the keyboard shall execute a self test. The self test shall consist of a ROM checksum test, a RAM test and a test for stuck key switches. After completion of the test, the keyboard shall send the result of the diagnostic to the host. The code will be AA hex to indicate successful completion of the self test or FC hex to indicate failure of some portion of the test. If the keyboard fails the self test, it shall be disabled and wait for a command from the system.

Results of the self test are transmitted between 450 and 2500 milliseconds after POR and between 300 and 500 milliseconds after the self test is initiated.

5.5 Autorepeat

The power-on default condition shall cause the last key depressed to repeat at 10 characters per second after a 500 millisecond delay. This may be changed by the system when the keyboard is using Communication Mode 2.

5.6 Buffering

The keyboard shall be capable of storing 16 scan codes in a first in - first out (FIFO) circular buffer. If buffer overflow occurs, the last code in the buffer is replaced by a hex 00 in Communication Mode 2, and hex FF in Communication Mode 1.

5.7 Mode Switch Settings

The keyboard shall use the mode switch settings shown in Table 5.

Keyboard Specification 20425

TABLE 5
MODE SWITCH SETTINGS

<u>Host</u>	<u>Communication Mode</u>	<u>Settings*</u>	<u>Scan Set Used</u>
Enhanced XT/AT	1 or 2	#1 OFF #2 OFF	1, 2, or 3
PC or XT	1	#1 ON #2 OFF	1
AT	2	#1 ON #2 ON (See Note 1)	2

* Switch positions 3 and 4 are not used.

5.8 Auto-Discrimination

If Switch #1 is OFF, the keyboard shall power-up in Communication Mode 2 with Scan Set 2 as the active scan set. If the system clamps the data line on the last clock of power-up completion (code "AA"), for at least 45 microseconds, the keyboard shall be in Communication Mode 1 with Scan Set 1. See Figure 7.

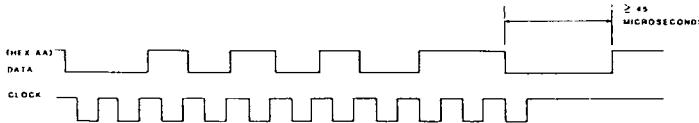


Figure 7: Power-Up Auto-Discrimination Timing

Keyboard Specification 20425

5.9 LED Indicators

The keyboard shall include three LED indicators, namely, Num Lock, Caps Lock, and Scroll Lock.

The indicators shall be located at the right hand end of the Function key row, and directly above the Numeric Keypad. (Refer to the F0 hex command in Paragraph 5.2.4.)

The keyboard shall power-up with all LED's OFF.

5.10 Software "Watch Dog"

The keyboard software shall contain a "watch dog" timer which attempts to keep the microprocessor running properly at all times (in spite of possible noise "glitches" on the power supply or ESD events).

The watch dog sets certain RAM locations during the scanning routine. Then, during the internal timer interrupt routine, these locations are tested. If the expected values are found it is assumed that the microprocessor is still scanning properly. The RAM locations are reset to some other values for the next test and normal execution continues. If the reset values are not in the RAM locations (or any other value besides the one set during scanning are found) it is assumed that a problem has occurred and a complete reset is performed by the microprocessor.

The results of this reset are identical to the response to a reset command from the host.

6.0 Electrical Requirements

The interface shall consist of two bidirectional lines, clock and data, which can be controlled by open collector drivers on either side. The keyboard side shall be terminated by 3300 ohm resistors. Voltage levels shall be TTL compatible, and the keyboard drivers capable of sinking a minimum of 20 mA including the current sourced by the pullup resistors on the keyboard).

Keyboard Specification Z0425

6.1 Signal Connector

The signal connector shall be a 5 pin DIN connector, Manufacturing Switch Craft (PN 05BL5M), or equivalent. Connections shall be as follows:

<u>Pin</u>	<u>Signal</u>	
1	Clock	
2	Data	
3	No Connection	
4	Logic Ground	
5	+ 5 VDC	

Connector on the
cable of keyboard
(looking in).

6.2 Chassis Ground

The chassis ground shall be isolated from logic ground.

6.3 Power Requirement

The keyboard shall require 5 volts +/-5% at 400 millamps maximum.

7.0 ENVIRONMENTAL REQUIREMENTS

7.1 Temperature

Operating 0 to 55 degrees C
Non-operating -40 to 70 degrees C

7.2 Relative Humidity

20% to 95% non-condensing

7.3 Shock

Operating. . . 10G 1/2 sine wave, 10ms duration, any axis
Non-operating . 100G 1/2 sine wave, 10ms duration, any axis

7.4 Vibration

Operating. . . 0.4 inch double amplitude 5 to 50 Hz
Non-operating . 0.4 inch double amplitude 2 to 10Hz
(In original cartons)

7.5 Altitude

Operating. . . -1,000 ft to +12,000 ft
-1,000 ft to +12,000 ft

Keyboard Specification 20425

7.6 Electrostatic Immunity

The keyboard shall meet the electrostatic immunity requirements described in Key Tronic document 36-02464, Part 18.

7.7 EMI/RFI

The keyboard shall be certified to comply with FCC rules, Subpart J of Part 15 for class B equipment when used in conjunction with an IBM PC,XT, or AT. The keyboard shall not hinder any other properly designed product from obtaining certification.

7.8 Safety

The keyboard shall not be manufactured from any material that will prevent or obstruct the obtaining Underwriter's Laboratories (UL) recognition.

Materials shall conform to the following UL requirements:

- Printed Circuit Board UL94V-0
- Keytops and Enclosure UL94HB
- Cables UL2960 and 20197

8.0 RELIABILITY

8.1 Mean Time Before Failure (MTBF)

The MTBF of the keyboard is calculated in excess of 100,000 hours based on failure rates determined through either actual life testing or MIL-HDBK-217C calculations.

8.2 Switch Life

Switch life shall be a minimum of 100 million cycles based on life test data.

8.3 Maintainability

The mean time to repair the keyboard by a qualified repair technician shall be 15 minutes.





APPLICATION NOTE

CAPACITANCE MATRIX INTERROGATION CHIP

PART NUMBER 22-00958-000

KEY TRONIC DOCUMENT NUMBER 36-1829





REVISIONS

Revision	Description	Date	By
---	Original Draft	08/1/84	RLN <i>RJA</i>
A	Production Release	08/15/84	RLN <i>RJA</i>
B	I ₁₁₁ was ~400uA, is -900uA I ₁₁₂ was 5mA, is 3mA t _{SS} max. was: 16.9us is: 18us	03/05/85	RLN <i>GTM 20 June 85</i>



1.0 GENERAL DESCRIPTION:

The 22-00958-000 is a full-custom, CMOS, integrated circuit, which provides for a low power matrix scanning and detection capability to be used on encoded capacitance keyboards. The device will accommodate 128 keys with the capacitance matrix configured in a 16 X 8 matrix (16 drive lines and 8 sense lines). The device will operate from a 5VDC + 10% power supply and is available in a plastic or ceramic 40-pin dual-in-line package.

1.1 APPLICATION:

The unique capacitance sensing circuitry (which is particularly insensitive to keyboard PCB loading effects) makes this device very versatile for use in encoded capacitance keyboard applications.

One primary application will be where low power operation is required. This device is also used in the emulation of the single chip keyboard controller 20-90049-XXX. A typical keyboard configuration is shown in Figure 1. The device pinout is shown in figure 2. The device will typically be used with the 8048 family of single chip microcomputers such as the 8048, 8049, 8035, 8039, etc., and their CMOS equivalents.

2.0 DEVICE INTERFACES

2.1 The microcomputer interfaces are

- A. **A0-A3:** A four-bit matrix address, which typically is provided by the lower order PORT 2 pins of the microcomputer. The input structures are a Schmitt trigger with an internal resistive pullup.
- B. **RST:** A reset pulse typically provided by the PROG output of the microcomputer, which latches the matrix address, initializes the Y line counter and the comparator sensing circuitry. The input structure is a Schmitt trigger with an internal resistive pullup.
- C. **STB:** A strobe signal provided by a PORT 1 line, which initiates the actual scanning operation by internally enabling the timing and control logic of the device on the falling edge of the signal. This signal is generated by the microcomputer for each key location within the matrix. The input structure is a Schmitt trigger with an internal resistive pullup.

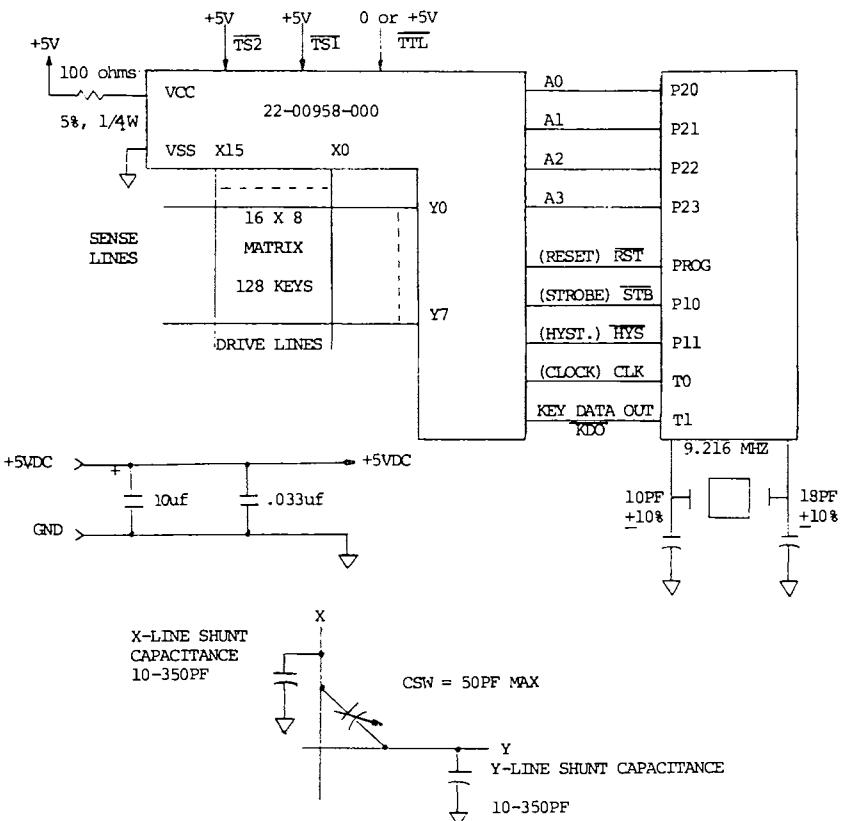


FIGURE 1: TYPICAL KEYBOARD CONFIGURATION



Y7	1	40	Vcc
Y6	2	39	Key Data Out
Y5	3	38	TS2
Y4	4	37	TSI
Y3	5	36	N/C
Y2	6	35	N/C
Y1	7	34	TTL
Y0	8	33	AO
X15	9	32	A1
X14	10	31	A2
X13	11	30	A3
X12	12	29	Clock
X11	13	28	Hysteresis
X10	14	27	Strobe
X9	15	26	Reset
X8	16	25	X0
X7	17	24	X1
X6	18	23	X2
X5	19	22	X3
Vss	20	21	X4

22-00958-000

FIGURE 2: DEVICE PINOUT



- D. CLK: A clock line from the T0 pin of the 8048 microcomputer family with a frequency of approximately 1/3 the timing crystal frequency. The timing parameters specified herein are based upon a clock frequency range of 2.67 MHz to 3.67 MHz. However, the device will operate with a clock frequency down to 1 MHz if required. All internal timing and control is derived from this signal. The input structure is LSTTL compatible.
- E. HYS: A hysteresis select signal from the controlling electronics, usually a PORT 1 pin, which alters the threshold of the key detection circuitry by disabling the smaller of two internal reference capacitors. The signal is normally high. When this signal is toggled to a low level, it increases the detection sensitivity. The input structure is a Schmitt trigger with an internal resistive pullup.
- F. KDO: The key data out signal represents the latched serial data output to the microcomputer which is driven to a low state for each detected key closure.

2.2 The capacitive keyboard interfaces are:

- A. X0-X15: Sixteen normally low matrix drive lines (referred to as X lines) which are sequentially pulsed high at the rising edge of the strobe pulse for each switch location within the matrix.
- B. Y0-Y7: Eight matrix sense lines (referred to as Y lines) which connect to a single analog charge difference detector through an 8 to 1 multiplexing function within the device. The detector senses the effective charge present at the Y input, and outputs a digital signal to a data latch. When selected, each of the eight matrix sense lines is biased nominally at +2.0V with Vcc = 5.0V. Unselected Y lines are terminated through a pull down resistance of <1K ohms. The input impedance at each input is 19.2K ohm \pm 30% when selected.

2.3 Drive Select Pin:

TTL: Pin #34 is provided to allow selection of either standard LSTTL or CMOS interface capability. The pin is to be connected to VSS when the device is being driven by TTL outputs and is to be connected to VCC when the device is being driven by CMOS outputs. When TTL is low, pullup resistors are connected to AO-A3, HYS, STB, and RST inputs.



2.4 Two pins are used to invoke test modes for the device:

TS1: When pin 37 is connected to VSS, test mode 1 is enabled, which presents five internal timing signals to pins 9 through 13. The normal X11 through X15 outputs are disabled in this mode. In normal operation, TS1 is connected to VCC.

TS2: When pin 38 is connected to VSS, test mode 2 is enabled. This test mode disables two internal bias resistors on the Y lines and allows measurement of the prebias voltage, which is used during device qualification and production screening. In normal operation, TS2 is connected to VCC.

3.0 CAPACITANCE THRESHOLD

The capacitance threshold of the device is $8.1 \text{ pF} \pm 21\%$ ($6.4 \text{ pF} < C_{th} < 9.8 \text{ pF}$). This threshold limit applies over the supply range of $5v \pm 10\%$, temperature range of 0 degrees C to 70 degrees C and Y line shunt capacitance of 10 pF to 350 pF and a Y line input impedance of $19.2k \text{ ohm} \pm 30\%$.

If a key capacitance, connected between an X drive line and a Y sense line, is equal to or less than 6.4 pF , KDO will be high, indicating an open key. Similarly, if a key capacitance is equal to or greater than 9.8 pF , KDO will be low, indicating a closed key.

4.0 HYSTERESIS CONTROL

Under software control, the capability exists to internally alter the threshold of the device. The hysteresis control is active when the microcomputer detects a key closure at any one key address. Upon such detection, the effective capacitance threshold is decreased by approximately 20%. For example, if the capacitance threshold was at a nominal 8.1 pF initially, upon software control, it is alterable to 6.5 pF .

ESD LATCHUP PROTECTION

5.0 Since all keyboard installations are inherently susceptible to electrostatic discharges (ESD) due to constant human interaction, a 100 ohm to $1K \text{ ohm}$, 5% , $1/4W$ resistor should be installed in series with the +5VDC connection to this device. This precaution will ensure that the device (in varying degrees, all CMOS devices are inherently susceptible to SCR latch up phenomena) will withstand a minimum ESD level of 15KV applied directly to the keyboard.



6.0 DC SPECIFICATION (Ta = 0 to 70 degrees C, VCC = 5V \pm 10%)

PARAMETER	TEST CONDITIONS	SIGNAL NAME	LIMITS		UNITS
			MIN	MAX	
Icc	$\overline{\text{TTL}} = \text{VSS}$; Static	Supply Current Drain		3	mA
Vih1	$\overline{\text{TTL}} = \text{VSS}$ $I_{ih1} = -40\mu\text{A}^{**}$	$\overline{\text{STB}}$, $\overline{\text{RST}}$, HYS, AO-A3, $\overline{\text{TS1}}$, $\overline{\text{TS2}}$	3.8		V
Vil1	$I_{il1} = -900\mu\text{A}^{**}$ $\overline{\text{TTL}} = \text{VSS}$	$\overline{\text{STB}}$, $\overline{\text{RST}}$, HYS, AO-A3, $\overline{\text{TS1}}$, $\overline{\text{TS2}}$.6	V
Vil2	Y inputs unselected $I_{il2} = 3\text{mA}$	Y0 through Y7		.4	V
Vih3 Vil3		CLK, $\overline{\text{TTL}}$ CLK, $\overline{\text{TTL}}$	2	.8	V
Voh1	$I_{oh1} = -800\mu\text{A}$	X0-X15	2.4		V
Vol1	$I_{ol1} = 5\text{mA}$	X0-X15		.45	V
Voh2	$I_{oh2} = -1.6 \text{ mA}$	$\overline{\text{KDO}}$	2.4		V
Vol2	$I_{ol2} = 1.6 \text{ mA}$	$\overline{\text{KDO}}$.45	V
Vt+ Positive Going Threshold Voltage		$\overline{\text{RST}}$, $\overline{\text{STB}}$, HYS, AO-A3	1.8	3.0	V
VT- Negative Going Threshold Voltage		$\overline{\text{RST}}$, $\overline{\text{STB}}$, HYS, AO-A3	1.3	2.1	V
Hysteresis [(Vt+) - (Vt-)]		$\overline{\text{RST}}$, $\overline{\text{STB}}$, HYS, AO-A3	0.5		V
Cin	$f_C = 1\text{MHz}$	Pin Input Capacitance		10	pF

** An internal resistor pullup is included to allow these pins to be driven by LSTTL levels.



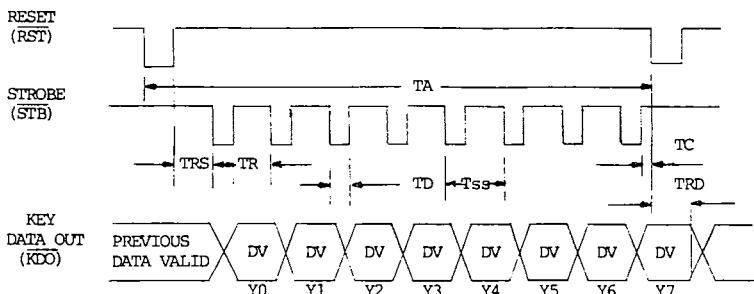
7. TIMING REQUIREMENTS:

The following AC specifications refer to the timing diagram of Figure 3. The values apply over a temperature range of 0 to 70 degrees C, Vcc = 5V + 10% and clock frequency of 2.67 to 3.67 MHz. Load capacitance for KDO = 50 pF.

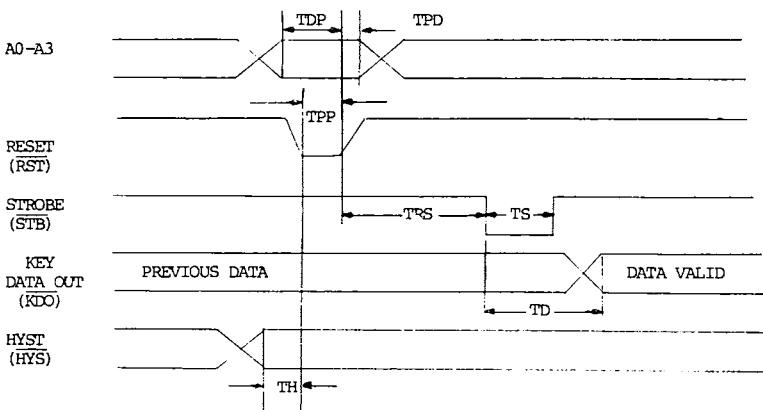
SYMBOL	PARAMETERS	VALUE		UNITS
		MIN.	MAX.	
tcy	Clock period	273	375	ns
tch	Clock high	91	125	ns
tcl	Clock low	182	250	ns
tpp	RST pulse width	500	---	ns
tdp	AO-A3 stable to rising edge of <u>RST</u>	300	---	ns
tpd	AO-A3 hold time from rising <u>RST</u>	30	---	ns
ta	Time for full column scan (8 keys)	100.7	---	us
th	HYS stable to falling edge of <u>RST</u>	-0-	---	ns
trs	RST rising edge to STB falling edge	9.3	12.8	us
tss	STB period	12.3	18.	us
ts	STB low	2.73	---	us
tr	STB high	9.56	---	us
tc	STB rising edge to next <u>RST</u> falling edge	2.46	---	us
trd	RST falling edge to KDO data invalid	---	5.0	us
td	STB falling edge to valid data on KDO	2.7	---	us



SCAN OF 8 KEYS (1 X LINE)



SINGLE KEY SCAN



CLOCK TIMMING

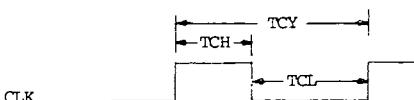


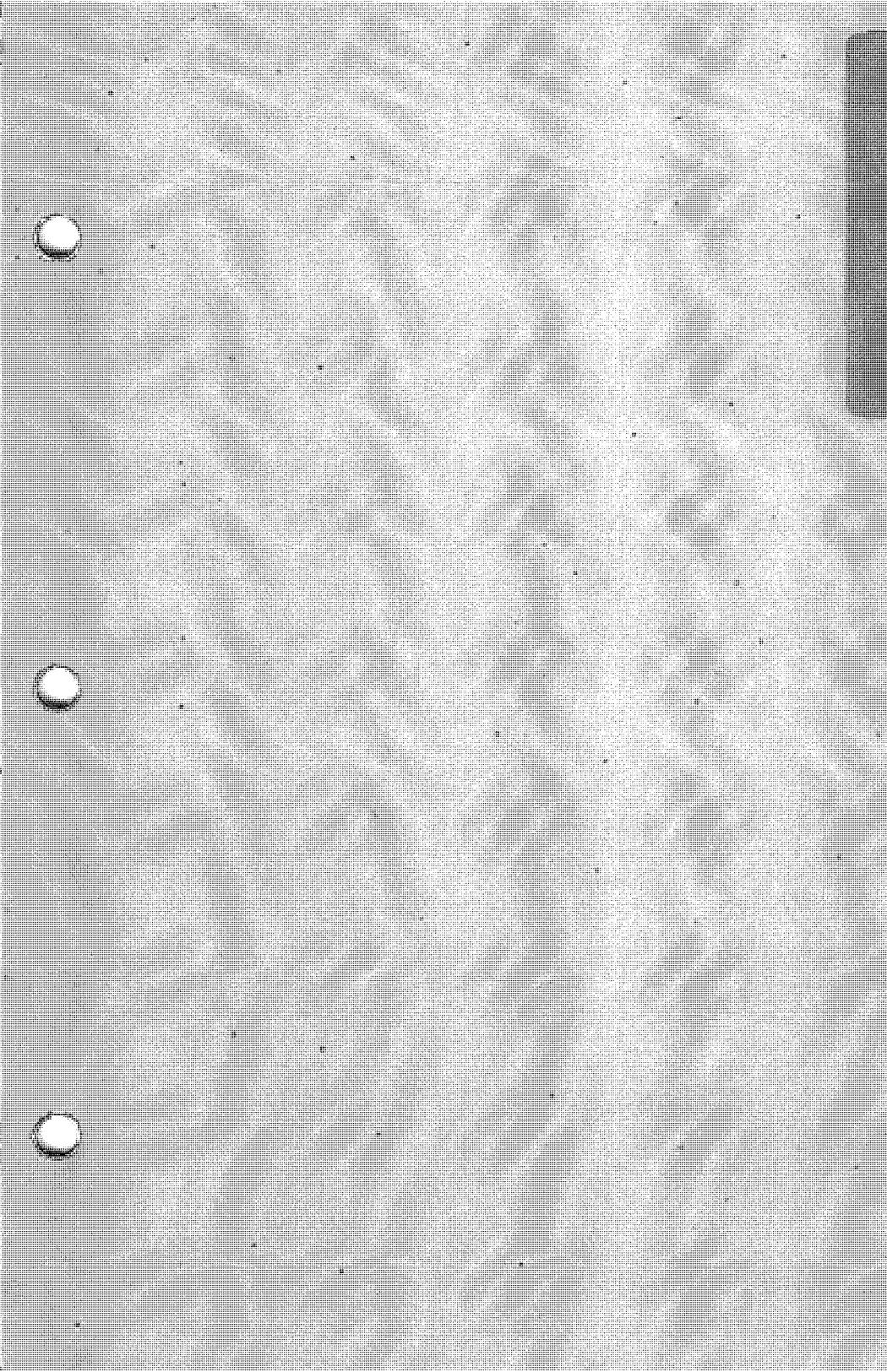
FIGURE 3



7.1 DEVICE TIMING AND OPERATION:

Figure 3 shows the timing for the scan of eight keys on one drive line. The reset pulse is issued prior to any key scanning sequence. After the generation of a reset pulse, eight strobe signals will be sent by the processor, as shown. The switch matrix is scanned eight keys at a time. The matrix is interrogated by addressing one of the drive lines and checking all of the sense lines to determine the up/down condition of the keys at the drive/sense intersections of that particular X line. The following command sequence is an example of how the above timing relationships can be derived using the 8049 microcomputer. The example assumes that a reset pulse is generated from the PROG output of the 8049, the strobe signal is derived from PORT 10, the hysteresis signal is derived from PORT 11 and the clock signal from the T0 output.

INTRGT:	MOV A,SRADDR	GET X-LINE ADDRESS
	MOVD P7,A	OUTPUT THE X-LINE, CLEAR LATCH
	MOV DLYCNT,#8	LOAD Y LINE COUNTER
	CLR A	
	ORL P1,#001H	LOAD P10 WITH A ONE (STROBE)
EXTY:	ANL P1,#0FEH	BRING P10 LOW (INTERROGATE FIRST Y LINE)
	ORL P1,#001	BRING P10 HIGH AGAIN
	JT1 \$+3	CHECK KDO FOR KEY CLOSURE
	INC A	
	RR A	
	DJNZ DLYCNT,NEXTY	IF 8 STROBES NOT COMPLETE, GENERATE ANOTHER ONE
...	^ @SRADDR	COMPARE TO PREVIOUS ...





TANDY COMPUTER PRODUCTS

4000 Disk Drive



**Specifications
of
MP-F73W-70D**

Double Sided 80 Tracks
Recording Capacity 2MBytes
Transfer Rate 500 Kbits/sec at 2MB mode
250 Kbits/sec at 1MB mode

VALID for MP-F73W-70D,
with the following serial numbers :

30,000,001 _____
_____ _____ _____
_____ _____ _____
_____ _____ _____

SONY CORPORATION

MFD TECHNICAL INFORMATION 00-0060 REV. 11-87 July 28, '87



*** Contents ***

1. Introduction
2. Specifications
 - 2.1 Configuration
 - 2.2 Physical Dimensions
 - 2.3 Performance
 - 2.3.1 Capacity
 - 2.3.2 Transfer Rate
 - 2.3.3 Access Time
 - 2.3.4 Functional
 - 2.3.5 Reliability
 - 2.4 Input Power Requirements
 - 2.4.1 Power Consumption
 - 2.4.2 Supply Voltages
 - 2.5 Environmental Limits and Orientation
 - 2.5.1 Temperature Range
 - 2.5.2 Humidity Range
 - 2.5.3 Vibration
 - 2.5.4 Shock
 - 2.5.5 Orientation
3. Signal Interface
 - 3.1 Pin Assignment
 - 3.1.1 Signal Connector
 - 3.1.2 Signal Connector Pin Assignment
 - 3.1.3 Power Supply Connector
 - 3.1.4 Power Supply Connector Pin Assignment
 - 3.2 DC Characteristics of Interface Signals
 - 3.2.1 Output Signals from Drive
 - 3.2.2 Inputs Signal to Drive
 - 3.2.3 Recommended Circuit for Signal Interface

3.3 Signal Definitions

- 3.3.1 DRIVE SELECT 0,1,2,3
- 3.3.2 MOTOR ON
- 3.3.3 STEP
- 3.3.4 DIRECTION
- 3.3.5 HEAD SELECT
- 3.3.6 WRITE GATE
- 3.3.7 WRITE DATA
- 3.3.8 INDEX
- 3.3.9 TRACK 00
- 3.3.10 WRITE PROTECT
- 3.3.11 READ DATA
- 3.3.12 DISK CHANGE

3.4 Timing Requirements

- 3.4.1 Head Access
- 3.4.2 TRACK 00 Signal
- 3.4.3 Write Data Timing
- 3.4.4 Read Data Timing
- 3.4.5 Index Pulse
- 3.4.6 Disk Change

3.5 Power on and Power off Requirements

- 3.5.1 Data Protection
- 3.5.2 Power Supply Sequencing
- 3.5.3 Power-On Reset Timing

3.6 Disk Motor Rotation and Disk Insertion.

4. Safety

5. Power On Initialization

1. Introduction

This document describes the specifications of the MP-F73W-70D which has the capability to read and write up to 2MB data (unformatted) on a HD disk as well as read and write up to 1MB (unformatted) data on a 1MB disk. The main features of the MP-F73W-70D are: low power consumption, low height, and high reliability with a simple mechanism and electric circuit.

This drive maintains the 300rpm of the disk motor rotational speed constantly, therefore,

- a. When a 2MF disk is inserted into the MP-F73W-70D, both read and write operations can be carried out and the data transfer rate is 500 kbytes/sec.
- b. When a 1MB disk (proposed ANSI standard) is inserted into the MP-F73W-7 D, the data transfer rate becomes 250 kbytes/second.

2. Specifications

2.1 Configuration

The drive consists of Read/Write heads, head positioning mechanism, disk motor, interface logic circuit and Read/Write circuit.

2.2 Physical Dimensions

The detailed physical dimensions are shown in Figure 2.1.
The main dimensions are:

- 1) Height : 30 mm (1.18 in.)
- 2) Width : 101.6 mm (4.00 in.)
- 3) Depth : 150 mm (5.91 in.)
- 4) Weight : 480g (1.06 pounds) max.

2.3 Performance

2.3.1 Recording Capacity (unformatted, MFM)

<u>2MB mode</u>	<u>1MB mode</u>
2.0 Mbytes/disk	1.0 Mbytes / disk
1.0 Mbytes/surface	0.5 Mbytes / surface
12.5 Kbytes/track	6.25 Kbytes / track

2.3.2 Transfer Rate

Burst transfer rate : 500 Kbytes/sec for MFM in a 2MB mode
250 Kbytes/sec for MFM in a 1MB mode



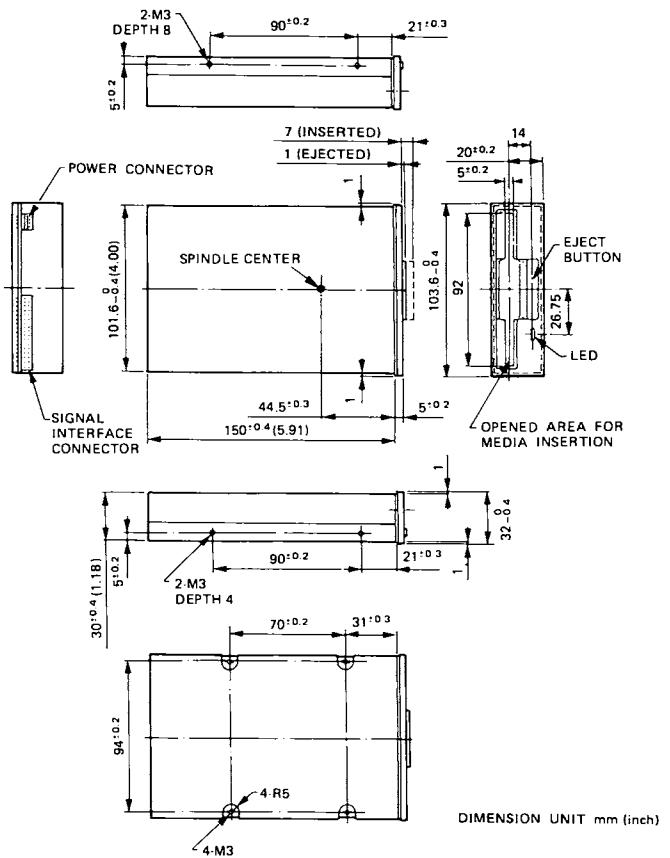


Fig. 2.1 Physical Dimensions

2.3.3 Access Time

a. Track to Track Slew Rate : 3 msec max.

b. Head Settling Time : 15 msec max.

The value of 15msec is the time necessary to stabilize the head within 0.035mm of its absolute position.

c. Motor Start Time : 500 msec max.
(700 msec max.*)

Motor start time is defined as the time period necessary to stabilise the Motor Rotational Speed variance to less than +/-1.5% after turning the MOTOR ON signal on.

NB. When a disk is inserted in the drive, the Motor Start Time will be 700 msec at maximum, but, after that it will be 500 msec max. as long as the disk is kept inserted.

2.3.4 Functional

a. Rotation Speed : 300 rpm
The continuous speed variation is within +/-1.5%.
The instantanous speed variation is within +/-1.5%.

b. Recording Density : 17434 BPI (Side 1, Track 79) in a 2MB mode.
8717 BPI (Side 1, Track 79) in a 1MB mode.

c. Track Density : 135 TPI

d. Cylinders : 80

e. Tracks : 160

f. R/W Heads : 2

2.3.5 Reliability

a. Mean Time Between Failures (MTBF) : 10,000 POH

b. Mean Time to Repair (MTTR) : 30 minutes

c. Preventive Maintenance (PM) : Not Required .

d. Components life : 5 years or 15000 POH

e. Error Rate :

1. Soft Read Error : Less than 1 per 10^9 bits read

2. Hard Read Error : Less than 1 per 10^{12} bits read

3. Seek Error : Less than 1 per 10^6 seeks

2.4 Input Power Requirements

2.4.1 Power Consumption

TTL Interface

Standby	255 mW
Operation (read/write mode)	2.8 W

2.4.2 Supply Voltages

<u>Voltage</u>	<u>Max. Ripple</u>	<u>Current</u>
+12.0V +/-5%	0.1Vpp	Standby 0.3 mA Average 130 mA (Read) Peak 500 mA (Motor Start) Peak 450 mA (stepping during Motor On)
+5.0V +/-5%	0.1Vpp	Standby 50 mA Operating 240 mA

2.5 Environmental Limits

2.5.1 Temperature Range

Operating	: 5°C to 50°C ambient (40°F to 122°F)
Transportation	: -40°C to 60°C (-40°F to 140°F)
Storage	: -20°C to 60°C (-20°F to 140°F)

2.5.2 Humidity Range

Operating	: 8% to 80% relative humidity with a wet bulb temperature of 29°C (85°F) and no condensation.
Transportation and Storage	: 5% to 95% relative humidity and no condensation

2.5.3 Vibration

Operating	: The unit can perform Read/Write operations without an error rate beyond that specified while withstanding continuous vibrations at a frequency of 10 to 500 Hz with an acceleration of no more than 0.5G along each of the three mutually perpendicular axes.
-----------	---

Transportation and Storage	: The unit can withstand continuous vibrations from 10 to 300 Hz with a maximum acceleration of 2.0G along each of the three mutually perpendicular axes without any degradation of any characteristics below the performance specifications.
----------------------------	---

2.5.4 Shock

Operating : The unit can withstand a 5.0G shock for 11 msec with a 1/2 sine wave shape in each of the three mutually perpendicular axes while performing normal Read/Write functions without damage or any loss of data.

Transportation and Storage : The unit when unpacked can withstand an 11 msec with a 1/2 sine wave shock of 60G on any of the three mutually perpendicular axes.

2.5.5 Orientation

The drive does not necessarily need to be horizontally positioned. In fact, as seen in figure 2-3, there are many other possible orientations.

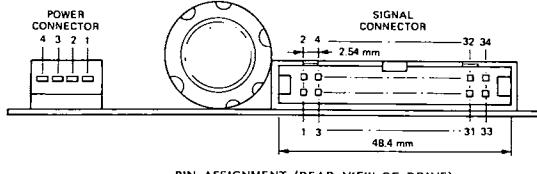
3. Signal Interface

3.1 Connector and Pin Assignments

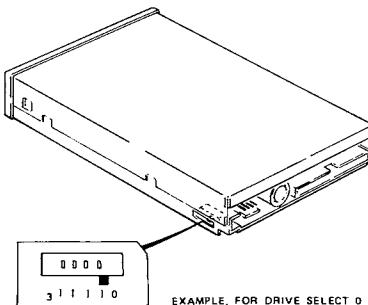
3.1.1 Signal connector

Receptacle : 3M 3414-6500xx or Equivalent

Cable : 3M 3365/34 or Equivalent



PIN ASSIGNMENT (REAR VIEW OF DRIVE)



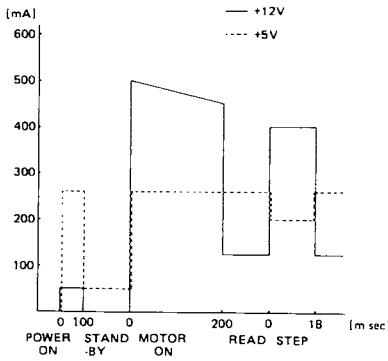


Fig. 2.2 DC Current Profile (typical)

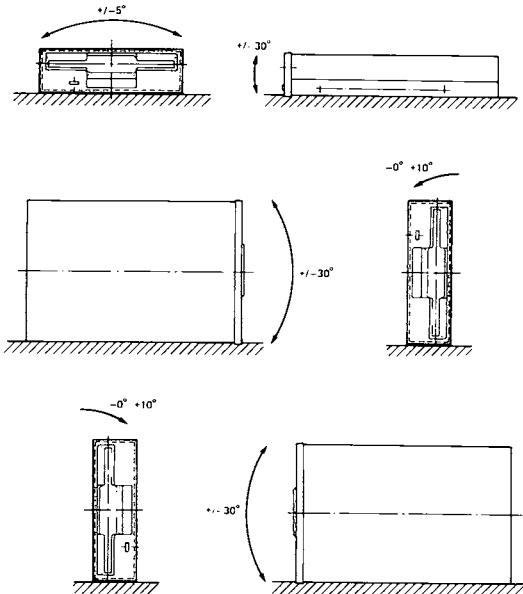


Fig. 2.3 Orientations

3.1.2 Signal Connector Pin Assignment

PIN	SIGNAL DESCRIPTION	PIN	SIGNAL DESCRIPTION
1	N.C.	2	N.C.
3	+5V	4	N.C.
5	+5V	6	DRIVE SELECT 3
7	+5V	8	INDEX
9	+5V	10	DRIVE SELECT 0
11	+5V	12	DRIVE SELECT 1
13	RETURN	14	DRIVE SELECT 2
15	RETURN	16	MOTOR ON
17	RETURN	18	DIRECTION
19	RETURN	20	STEP
21	RETURN	22	WRITE DATA
23	RETURN	24	WRITE GATE
25	RETURN	26	TRACK 00
27	RETURN	28	WRITE PROTECT
29	+12V	30	READ DATA
31	+12V	32	HEAD SELECT
33	+12V	34	DISK CHANGE

3.1.3 Power Supply Connector

Receptacle : AMP 171822-4 or Equivalent

Contact : AMP 170262-1 or Equivalent

Wire : AWG 20

3.1.4 Power Supply Connector Pin Assignment

PIN	SIGNAL DESCRIPTION
1	+5V
2	GND (+5V Return)
3	GND (+12V Return)
4	+12V

3.2 DC Characteristics of Interface Signals

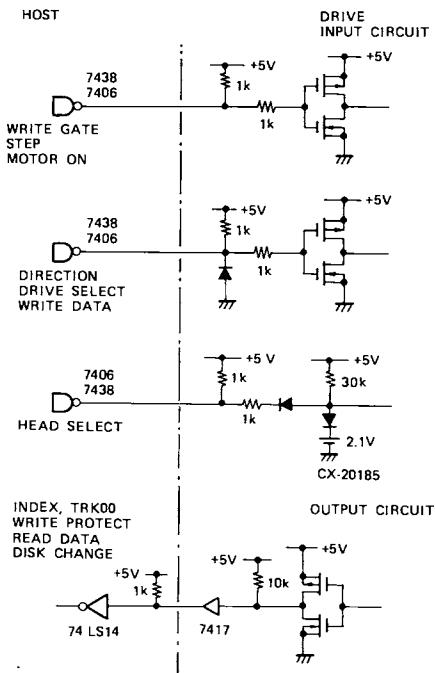
3.2.1 Output Signal from Drive

Name	Output IOH(mA)	Current IOL(mA)	Output VOH(V)	Voltage VOL(V)
TTL interface	0.25	40		0.7
All outputs				

3.2.2 Input Signal to Drive

Name	Input Current VIN=0.4V IIL(mA)	Input Voltage Threshold VIH(V) VIL(V)
TTL interface	-5.0	2.2 0.8
All inputs		

3.2.3 Recommended Circuit for Signal Interface



1 K ohm pull-up resister is recommended to be implemented on the output line from the drive.

The cable length must be less than 1.5m.

Recommended driver IC : 7406, 7438

3.3 Signal Definitions

3.3.1 DRIVE SELECT 0,1,2,3

The SELECT lines are used to enable or disable all other interface lines except a MOTOR ON line. When the SELECT line is true (low), the drive is enabled and considered active. When the SELECT line is false (high), all controller inputs except the MOTOR ON line are ignored and all output lines are disabled.

NB. IN USE (LED) LAMP:

When a drive is selected, the IN USE lamp on the selected drive is turned on, and when a drive is not selected, it is turned off.

3.3.2 MOTOR ON

When this input is true (low) and a disk is inserted, the spindle motor starts to run. When this line is made false (high) or a disk is ejected, the spindle motor decelerates and stop.

However, if the MOTOR ON signal becomes false(high) during either a write or erase operation, the disk motor does not stop rotating until both the ERASE GATE signal and the WRITE GATE signal become high (false).

3.3.3 STEP

When a drive is selected, a true (low) pulse on this line causes the Read/Write head to move to the adjacent track. The direction of the head movement is determined by the status of the DIRECTION input at the trailing edge of the pulse.

The step operation can be performed even if there is no disk inserted in the drive.

3.3.4 DIRECTION

When a drive is selected, a false (high) level on this input causes a STEP input to move the Read/Write head away from the disk spindle. A true (low) level causes a STEP pulse input to move the Read/Write head toward the drive spindle.

3.3.5 HEAD SELECT

When a drive is selected, a true (low) level on this input causes Head 1 (upper) to be selected. A false (high) level on this input will cause Head 0 (lower) to be selected.

If the HEAD SELECT signal changes during either write or erase operation, the head will not be changed until both ERASE GATE and WRITE GATE signal becomes high (false).

3.3.6 WRITE GATE

When this line is made true (low) while a drive is selected, the write current circuits are enabled and information may be written under control of the WRITE DATA input.

3.3.7 WRITE DATA

If the WRITE GATE is true (low), a true pulse (low) on the WRITE DATA line signal causes a bit to be written on the disk. Pulses on this line is neglected when WRITE GATE signal is false (high). No pre-compensation is required.

3.3.8 INDEX

When the drive is selected, a true (low) pulse is generated on this line by each revolution of the spindle.

3.3.9 TRACK 00

This line is true (low) when the drive is selected and the Read/Write head is positioned on track 00.

3.3.10 WRITE PROTECT

If a write-protect disk is inserted while a drive is selected, this line becomes true (low) and the drive is not able to write data. At all other times, except when a disk is ejected while the drive is selected, this line becomes false (high).

3.3.11 READ DATA

When the drive is selected, a true (low) pulse is generated on this line every time a bit is detected.

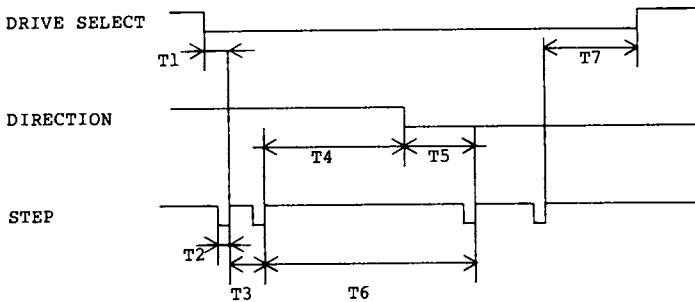
3.3.12 DISK CHANGE

This line is true (low) whenever a disk is removed from the drive. The line will remain true (low) until both the following conditions have been met:

1. A disk is inserted,
and
2. A STEP pulse has been received when the drive is selected.

3.4 Timing Requirements

3.4.1 Head Access



T1 : 0.5 us min.

T2 : 1.3 us min.

T3 : 3.0 ms min.

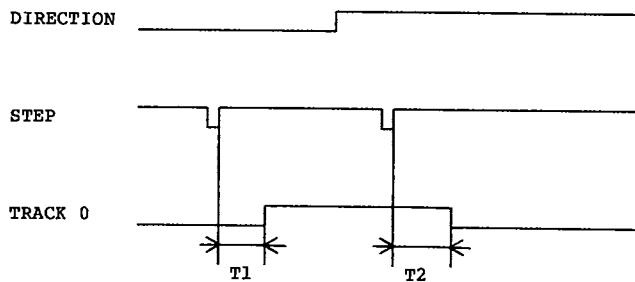
T4 : 2.4 us min.

T5 : 0.5 us min.

T6 : 18 ms min.

T7 : 2.5 us min.

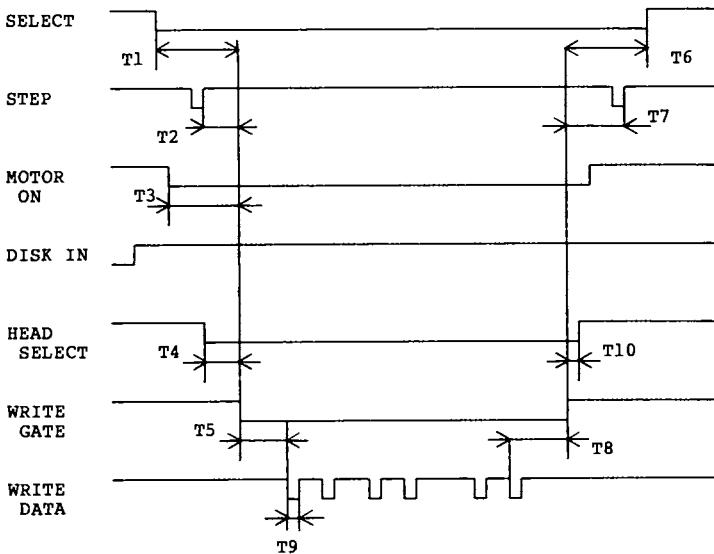
3.4.2 TRACK 00 Signal



T1 : 2.9 msec max.

T2 : 2.9 msec max.

3.4.3 Write Data Timing



T1 : 0.5 us min.

T6 : 0.5 us min.

T2 : 18 ms min.

T7 : 585 us min. (1020 us min.)

T3 : 500 ms max.**

T8 : 4 us min. (8 us min.)

T4 : 100 us min.

T9 : 150 ns min., 1000 ns max.
(150 ns min., 2000 ns max.)

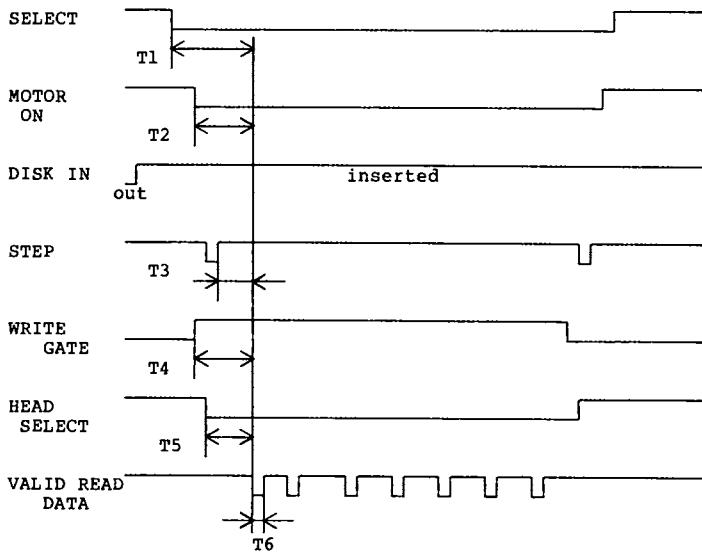
T5 : 4 us max.
(8 us max.)

T10 : See 3.3.4

***NB.** DISK IN, the disk-in sensor signal inside the drive, is high when a disk is inserted in the drive.

****NB.** When a disk is inserted in the drive, the Motor Start Time will be 700 msec at maximum, but, after that it will be 500 msec max. as long as the disk is kept inserted.

3.4.4 Read Data Timing



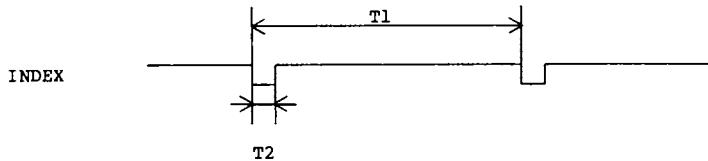
T1 : 0.5 us max. T4 : 615 us max. (1050 us max.)

T2 : 500 ms max.* T7 : 100 us max

T3 : 18 ms max. T8 : 350 ns min., 550 ns max.

NB. When a disk is inserted in the drive, the T2, is 700 msec at maximum, but, after that T2 is 500 msec max as long as the disk is kept inserted.

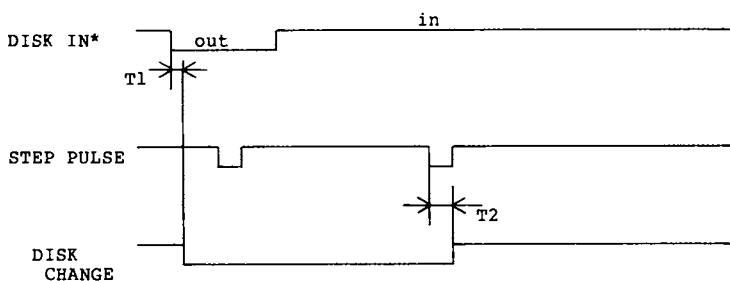
3.4.5 Index Pulse



T1* : 197 ms min., 203 ms max.

T2 : 1.25 ms min., 1.45 ms max.

3.4.6 Disk Change



T1 : 0.5 us max. T2 : 1.6 us max

*DISK IN, the disk-in sensor signal inside the drive, is high when a disk is inserted in the drive.

3.5 Power On and Power off Requirements

3.5.1 Data Protection

Turning power on or off will not cause any damage to recorded data on the disk as long as the drive is not in the midst of writing when the power is shut off or supplied.

3.5.2 Power Supply Sequencing

When the power is turned on, no special power supply sequencing is required. When the power is turned off, although there are no sequencing or timing requirements, both power supplies must fall monotonically to zero volts.

3.5.3 Power-On Reset Timing

Because it takes up to 200 msec to reset the control IC after the power has been turned on, the MP-F73W-70D cannot correctly perform any operations for this period of time after Power-On.

3.6 Disk motor rotation and Disk Insertion.

Even if the MOTOR ON signal is low (true), the disk motor will not rotate until a disk is inserted.

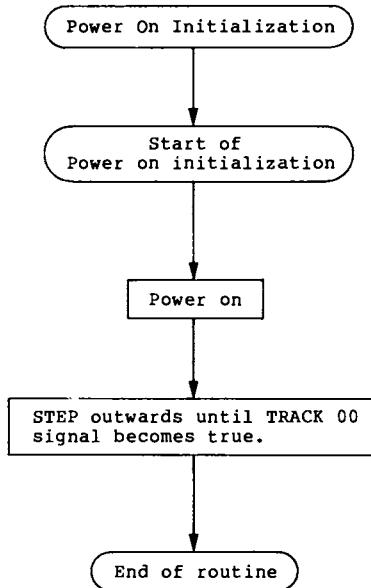
4. Safety Standards

MP-F73W-70D will meet the following product safety regulations:

U.L. 478
C.S.A. C.22.2, No.154
U.L. 94V-0 for Front Bezel

5. Power On Initialization

In order to reduce the peak current requirement when used in a daisy chain, the MP-F73W-70D has been designed not to seek track 00 automatically. If all the drives connected in the daisy chain sought track 00 simultaneously, this would place a significant power drain on the host system. Thus, the host system must perform the following routine just after power on in order to reset the track counter inside the drive.



Setting the Chip Selector Switch

Set the chip selector switch, S5, according to the first character, "K" or "L", which is printed on the head carriage. See Figures 1 through 3.

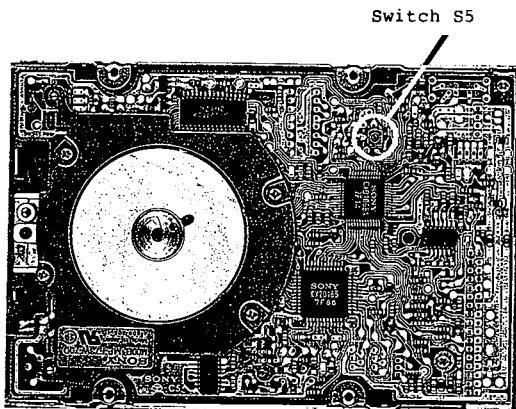


Fig. 1. Bottom view of MP-F73W-700

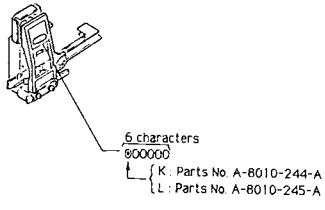
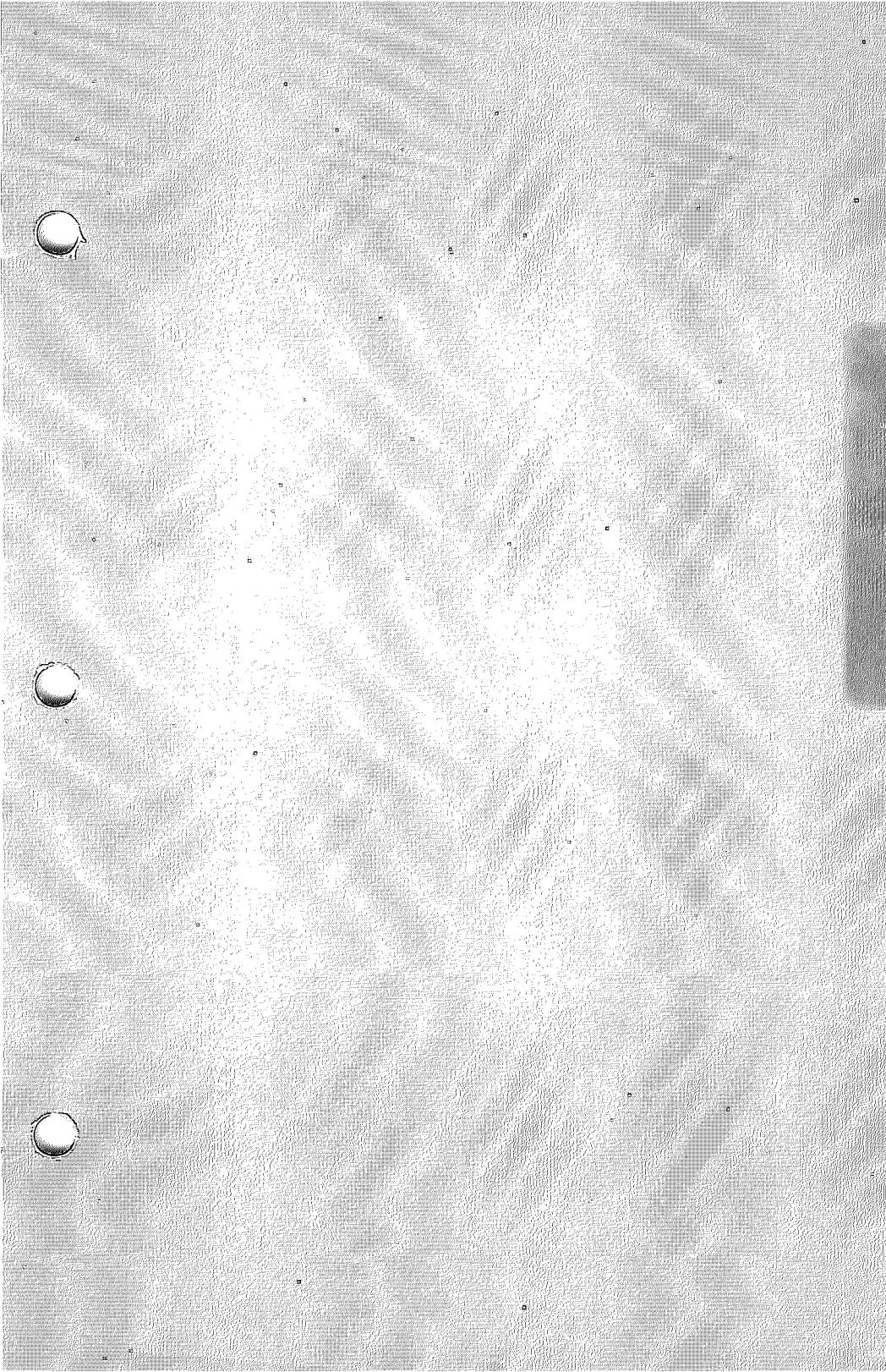


Fig. 2. Head Carriage Ass'y



Fig. 3. setting position

Set the position of switch S5 according to the first character, K or L.

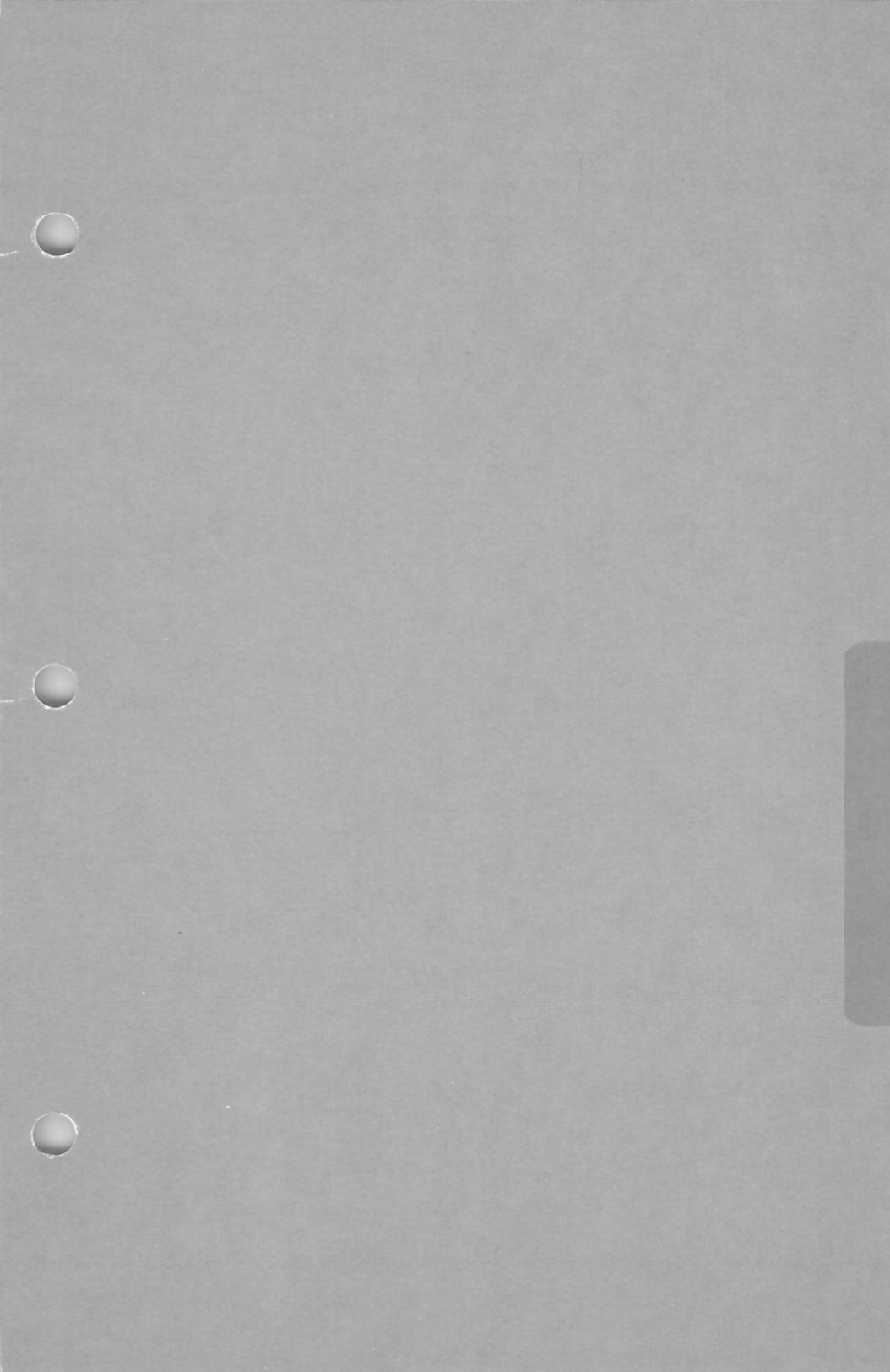




TANDY COMPUTER PRODUCTS

4000 Options







SOFTWARE



Software Contents

BIOS Services

Device I/O Services	1
Video Display	2
Function Descriptions	4
Equipment	14
Memory Size	15
Floppy Diskette I/O Support	16
Hard Disk Support	22
Serial Communications	31
Multi-Tasking Support	34
Keyboard	39
Line Printer	43
System Clock	45

Keyboard ASCII and Scan Codes

Keyboard ASCII and Scan Codes	49
-------------------------------------	----

MS-DOS Memory Map

High-Level Memory Map	53
Interrupt Vector Table	54
ROM BIOS Data Area	56
MS-DOS and BASIC Data Area	61



Tandy 3000/4000 BIOS Services

Device I/O Services

The BIOS (Basic Input/Output System) is the lowest-level interface between other software (application programs and the operating system itself) and the hardware. The BIOS routines provide various device input/output services, as well as boot strap, and print screen, and other services. Some of the services that BIOS provides, such as the graphics routines, are not available through the operating system,

All calls to the BIOS are made through software interrupts (that is, by means of assembly-language "INT x" instructions). Each I/O device is provided with a software interrupt, that transfers execution to the routine.

Entry parameters to BIOS routines are normally passed in CPU registers. Similarly, exit parameters are generally returned from these routines to the caller in CPU registers. To ensure BIOS compatibility with other machines, the register usage and conventions are, for the most part, identical.

The following pages describe the entry and exit requirements for each BIOS routine. To execute a BIOS call, load the registers as indicated under the ".EN" (entry condition). (Register AH contains the function number in cases where a single interrupt can perform more than one operation.) Then issue the interrupt given for the call. For example, the following can be used to read a character from the keyboard:

```
MOV AH,0  
INT 16H
```

Upon return, AL contains the ASCII character and AH contains the keyboard scan code.

Note: All registers except those used to return parameters to the caller are saved and restored by the BIOS routines.

Following is a quick reference list of software interrupts for all device I/O and system status services.

Service	Software Interrupts
Video Display	10 hex (16 dec)
Equipment	11 hex (17 dec)
Memory Size	12 hex (18 dec)
Floppy Disk	13 hex (19 dec)
Hard Disk	13 hex (19 dec)
Serial Communications	14 hex (20 dec)
Multi-Tasking Support	15 hex (21 dec)
Joystick	15 hex (21 dec)
Microsecond Delay	15 hex (21 dec)
Extended Memory	15 hex (21 dec)
Virtual Mode	15 hex (21 dec)
Keyboard	16 hex (22 dec)
Line Printer	17 hex (23 dec)
System Clock	1A hex (26 dec)

Video Display

These routines provide an interface to the video display, which is the half of the console (CON) device that is for output. MS-DOS considers the video display to be the default standard output (STDOUT) device.

Software Interrupt: 10 hex (16 dec)

Function Summary:

Control Routines:

AH = 0	Set CRT mode
AH = 1	Set cursor type
AH = 2	Set cursor position
AH = 3	Get cursor position
AH = 4	Read light pen position
AH = 5	Select active page
AH = 6	Scroll active page up
AH = 7	Scroll active page down

Text Routines:

AH = 8	Read attribute/character
AH = 9	Write attribute/character
AH = 10	Write character only

Graphics Routines:

AH = 11	Set color palette
AH = 12	Write dot
AH = 13	Read dot

Other Routines:

AH = 14	Write TTY* to active page
AH = 15	Get CRT mode
AH = 19	Write string

*Screen width is determined by the mode previously set. Some control characters (ASCII 00H-1FH) perform the usual special terminal function. These include (but are not limited to) BEL (07H), BS (08H), LF (0AH), and CR (0DH).

Note that AX is modified on all video calls.

Function Descriptions:

Set CRT Mode

Set CRT mode.



Entry Conditions

AH = 0

AL = mode value, as follows:

Alpha Modes

AL = 0	40 x 25 monochrome
--------	--------------------

AL = 1	40 x 25 color
--------	---------------

AL = 2	80 x 25 monochrome
--------	--------------------

AL = 3	80 x 25 color
--------	---------------

Graphics Modes

AL = 4	320 x 200 color graphics
--------	--------------------------

AL = 5	320 x 200 monochrome graphics
--------	-------------------------------

AL = 6	640 x 200 monochrome graphics
--------	-------------------------------

AL = 7	80 x 25 monochrome with mono text adapter
--------	--



Set Cursor Type

Set the cursor type and attribute.

Entry Conditions

AH = 1

CH = bit values:

Bits 5-6 cause an invisible or erratically blinking cursor

Bits 4-0 point to the start line for cursor within character cell

CL = bit values:

Bits 4-0 point to the end line for cursor within character cell

Set Cursor Position

Write (set) cursor position.

Entry Conditions:

AH = 2

BH = page number (must be 0 for graphics modes or when used with mono text adapter)

DH = row (0 = top row)

DL = column (0 = leftmost position)

Get Cursor Position

Read (get) cursor position.



Entry Conditions:

AH = 3

BH = page number (must be 0 for graphics modes or when used with mono text adapter)

Exit Conditions:

DH = row of current cursor position (0 represents top row)

DL = column of current cursor position (0 represents left-most column)

CH = cursor type currently set

Bits 4-0 point to the start line for cursor within character cell

CL = bit values:

Bits 4-0 point to the end line for cursor within character cell



Read Light Pen Position

Reads light pen position.

Entry Conditions:

AH = 4

Exit Conditions:

AH 0 = light pen switch not activated
1 = light pen values in registers

DH = row of current light pen position

DH = column of current light pen position

CH = raster line (0-199)

BX = pixel column (0-319 or 0-639)



Select Active Page

Select active display page (valid with color graphics adapter in alpha mode).

Entry Conditions:

AH = 5

AL = 0 through 7 new page value for modes 0, 1

AL = 0 through 3 new page value for modes 2, 3

Exit Conditions:

BH = contents of CRT page register

BL = contents of CPU page register

Scroll Up

Scroll active page up

Entry Conditions:

AH = 6

AL = number of lines to scroll; this number of lines will be blank at the bottom of the window. (0 means blank entire window)

CH = row of upper left corner of scroll window

CL = column of upper left corner of scroll window

DH = row of lower right corner of scroll window

DL = column of lower right corner of scroll window

BH = attribute (alpha modes) or color (graphics modes) to be used on blank line.

Attributes:

Alpha modes (0, 1, 2, 3) and monochrome mode (7):

Bits 2, 1, 0 = foreground color (alpha)

In mono, 0 = black, 1 = underline, 7 = white.

Bits 3 = foreground intensity

Bits 6, 5, 4 = background color

In mono, 7 = white, other values = black.

Bit 7 = blink

Select foreground color 0 - 15 and background color 0 - 7 from background color list of Set Color Palette (AH = 11).

Sample Attributes

Attribute	Alpha Modes	Mono Mode
07H	White on Black	Normal Text
0FH	Bright White on Black	High Intensity Text
87H	Blinking White on Black	Blinking Text
8FH	Blinking Bright White on Black	High Intensity Blinking Text
70H	Black on White	Reverse Video
F0H	Black on Blinking White	Blinking Reverse Video
01H	Blue on Black	Underline Text
09H	Bright Blue on Black	High Intensity Underline Text
00H	Black on Black	Invisible

Color (Graphics Mode):

BH = color palette index

The color is selected from the current palette based on the color palette index. In Modes 4 and 5, BH can be 0-3. In Mode 6, BH can be 0 or 1. All unused bits are zero. See default palettes of Set Color Palette (AH = 11).

[1000] In Modes 8 and 9, BH can be 0-15. In Mode A, BH can be 0-3.

Scroll Down

Scroll active page down.

Entry Conditions:

AH	=	7
AL	=	number of lines to scroll (0 means blank entire window)
CH	=	row of upper left corner of scroll window
CL	=	column of upper left corner of scroll window
DH	=	row of lower right corner of scroll window
DL	=	column of lower right corner of scroll window
BH	=	attribute (alpha modes) or color (graphics modes) to be used on a blank line.

See Scroll Up (AH = 6) for attribute values and Set Color Palette (A_I = 11) for color values.

Read Attribute or Color/Character

Read a character and its attribute or color at the current cursor position.

Entry Conditions:

AH	=	8
BH	=	display page number (not used in graphics modes or with mono text adapter)

Exit Conditions:

AL	=	character read
AH	=	attribute of character (alpha modes only)

Write Attribute Or Color/Character

Write a character and its attribute or color at the current cursor position.

Entry Conditions:

AH	=	9
BH	=	display page number (not used in graphics modes)
CX	=	number of characters to write
AL	=	character to write
BL	=	attribute of character (for alpha modes) or color of character (for graphics modes). If Bit 7 of BL is set, the color of the character is XOR'ed with the currently displayed color value for the character. See Scroll Up (AH = 6) for attribute values and Set Color Palette (AH = 11) for color values.

Write Character Only

Write character only at current cursor position.

Entry Conditions:

AH	=	10
BH	=	display page number (valid for alpha modes only)
CX	=	number of characters to write
AL	=	character to write
BL	=	color of character (graphics modes only)

Set Color Palette

Select the color palette.

Entry Conditions:

AH = 11
BH = 0

Set background color to color value in BL. Color values are:

0 = black	8 = dark gray
1 = blue	9 = light blue
2 = green	10 = light green
3 = cyan	11 = light cyan
4 = red	12 = light red
5 = magenta	13 = light magenta
6 = yellow	14 = yellow
7 = gray	15 = white

BH = 1 Set default palette to the number (0 or 1) in BL.

In black and white modes:

BL = 0: 1 for white
BL = 1: 1 for black

In 4 color graphics modes:

BL = 0 (1 = green/2 = red/3 = yellow)
BL = 1 (1 = cyan/2 = magenta/3 = white)

In 16 color graphics modes:

(1 = blue/2 = green/3 = cyan/4 = red/5 = magenta/6 = yellow/
7 = light gray/8 = dark gray/9 = light blue/10 = light green/
11 = light cyan/12 = light red/13 = light magenta/14 = yellow/
15 = white)

Note: For alpha modes, Palette Entry 0 indicates the border color. For graphics mode Palette Entry 0 indicates the border and the background colors.

Note: Some colors may not appear the same on all monitors (e.g., yellow can appear brown).



Write Dot

Write a pixel (dot).

Entry Conditions:

AH	=	12
DX	=	row number
CX	=	column number
AL	=	color value (When Bit 7 of AL is set, the resultant color value of the dot is the exclusive OR of the current dot color value and the value in AL.)

Read Dot

Read a pixel (dot).



Entry Conditions:

AH	=	13
DX	=	row number
CX	=	column number

Exit Conditions:

AL = color value of dot read

Write TTY

Write a character in teletype fashion. (Control characters are interpreted in the normal manner.)

Entry Conditions:

AH	=	14
AL	=	character to write
BL	=	foreground color (graphics mode)



Get CRT Mode

Get the current video mode.

Entry Conditions:

AH = 15

Exit Conditions:

AL = current video mode. See Set CRT Mode (AH = 0) for values
AH = number of columns on screen
BH = current active display page

Write String

Write a string of characters.

Entry Conditions:

AH = 19
ES:BP = pointer to the string of characters to be written
CX = number of characters in the string to display
DX = starting cursor position (DH = row, DL = column)
BH = page number (for alpha modes)
BL = attribute for characters (If attributes are not in the string, see below.)
AL 0 = Cursor is not moved, string is characters only, attribute is in BL.
1 = Cursor is moved to the next character position after the last character written, string is characters only, attribute is in BL.
2 = Cursor is not moved, string has alternating characters and attributes.
3 = Cursor is moved to the next character position after the last character written, string has alternating characters and attributes.

Equipment

This service returns the equipment flag (hardware configuration of the computer system) in the AX register.

Software Interrupt: 11 hex (17 dec)

The equipment flag returned in the AX register has the following meanings for each bit:

Reset =	the indicated equipment is not in the system
Set =	the indicated equipment is in the system
Bit 0	diskette installed
Bit 1	math coprocessor
Bit 3,2	not used
Bit 5,4	initial video mode 01 = 40 x 25 BW on color graphics adapter 10 = 80 x 25 BW on color graphics adapter 11 = 80 x 25 BW on monochrome text adapter
Bit 7,6	number of diskette drives (only if Bit 0 = 1) 00 = 1 01 = 2
Bit 8	not used
Bit 11,10,9	number of RS232 cards
Bit 12	not used
Bit 13	not used
Bit 15,14	number of printers

Memory Size

This service returns the total number of kilobytes of RAM in the computer system (contiguous starting from address 0) in the AX register. The maximum value returned is 640 (280 hex).

Software Interrupt: 12 hex (18 dec)

Floppy Diskette Drive I/O Support

These routines provide a means to control diskette drive operation.

Software Interrupt: 13 hex (19 dec)

Function Summary:

AH	=	0	Reset all disks
AH	=	1	Return status of last disk operation
AH	=	2	Read sectors from disk
AH	=	3	Write sectors to disk
AH	=	4	Verify sectors on disk
AH	=	5	Format track on disk
AH	=	21	Read disk type
AH	=	22	Get diskette change line status
AH	=	23	Set diskette type for format

Function Descriptions:

Reset All Disks

Resets the floppy diskette system, reset associated hardware, and recalibrate all drives.

Entry Conditions:

AH = 0

Exit Conditions:

See the following section, "Exits From All Floppy Disk Calls."

Return Status of Last Diskette Operation

Return the status of the last diskette operation in AH.

Entry Conditions:

AH = 1
DL < 80H

Exit Conditions:

AH = status of the last operation. See the following section, "Exits From All Floppy Disk Calls," for values.

Read Sectors From Disk

Read the specified sectors from disk into RAM.

Entry Conditions:

AH	=	2
DL	=	drive number (0-1)
DH	=	head number (0-1)
CH	=	track number (0-79)
CL	=	sector number (1-15)
AL	=	number of sectors to read (1-18)
ES:BX	=	pointer to disk buffer

Exit Conditions:

See the following section, "Exits From All Floppy Disk Calls."

Write Sectors To Disk

Write the specified sectors from RAM to disk.

Entry Conditions:

AH	=	3
DL	=	drive number (0-1)
DH	=	head number (0-1)
CH	=	track number (0-79)
CL	=	sector number (1-15)
AL	=	number of sectors to write (1-18)
ES:BX	=	pointer to disk buffer

Exit Conditions:

See the following section, "Exits From All Floppy Disk Calls."

Verify Sectors On Disk

Verify the specified sectors.



Entry Conditions:

AH	=	4
DL	=	drive number (0-1)
DH	=	head number (0-1)
CH	=	track number (0-79)
CL	=	sector number (1-18)
AL	=	number of sectors to verify (1-18)

Exit Conditions:

See the following section, "Exits From All Floppy Disk Calls."

Format Track On Disk

Format the specified track.

Note: Function Call 23 must be performed before using this routine.

To format 360K diskettes, the gap length for format and end of track parameters of the disk parameter table (INT 1EH) must be changed. These parameters should be restored when formatting is complete.



Entry Conditions:

AH	=	5
DL	=	drive number (0-1)
DH	=	head number (0-1)
CH	=	track number (0-79)
AL	=	number of sectors per track (used only for DMA bound check)
ES:BX	=	pointer to sector format table consisting of four bytes per sector.

The first byte is the track number.

The second byte is the head number.

The third byte is the sector number.

The fourth byte is the sector size code:

00 = 128 bytes/sector

01 = 256 bytes/sector

02 = 512 bytes/sector

03 = 1024 bytes/sector

For Track 0, Head 1, on a nine-sector/track diskette formatted with an interleave of 1 (256 bytes/sector) the table would be:

00H,01H,01H,02H, 00H,01H,02H,02H, 00H,01H,03H,02H
00H,01H,04H,02H, 00H,01H,05H,02H, 00H,01H,06H,02H
00H,01H,07H,02H, 00H,01H,08H,02H, 00H,01H,09H,02H

Exit Conditions:

See the following section, "Exits From All Floppy Disk Calls."

Read Disk Type

Return the disk type.

Entry Conditions:

AH	=	21
DL	=	drive number (0-1)

Exit Conditions:

AH	0	=	not present
	1	=	diskette with no change line
	2	=	diskette with change line
	3	=	hard disk

Get Disk Change Line Status

Test to see if diskette has changed.

Entry Conditions:

AH	=	22
DL	=	drive number (0-1)

Exit Conditions:

AH	00	=	[NC] diskette has not changed
	06	=	[C] diskette has changed
Also see the following section called "Exits From All Floppy Disk Calls."			

Set Disk Type For Format

Set the diskette type for the next FORMAT command.

Entry Conditions:

AH	=	23
AL	=	00 = not used
		01 = 360K media in a 360K drive
		02 = 360K media in a 1.2M drive
		03 = 1.2M media in a 1.2M drive
		1.44M media in a 1.44M drive
		04 = 720K disk in 720K drive
DL	=	drive number (0-1)

Exit Conditions:

See the following section, "Exits From All Floppy Disk Calls."

Exits From All Floppy Disk Calls

[NC]	=	operation was successful (AH = 0)
[C]	=	operation failed
AH		
	01H	Bad command or parameter
	02H	Address mark not found
	03H	Write protected
	04H	Sector not found
	06H	Diskette has changed
	08H	DMA overrun error
	09H	DMA boundary error
	10H	Bad CRC on disk read
	20H	Controller failed
	40H	Seek failed
	80H	Device timeout

Hard Disk Support

The following routines provide access to Hard Disk operations.

Software Interrupt: 13 hex (19 dec)

Function Summary:

AH	=	0	Reset all disks
AH	=	1	Return status of last disk operation
AH	=	2	Read sectors from disk
AH	=	3	Write sectors to disk
AH	=	4	Verify sectors on disk
AH	=	5	Format track on disk
AH	=	6	Unused
AH	=	7	Unused
AH	=	8	Return current hard disk parameters
AH	=	9	Initialize hard disk parameters
AH	=	10	Read long
AH	=	11	Write long
AH	=	12	Perform seek on hard disk
AH	=	13	Reset hard disk
AH	=	14	Unused
AH	=	15	Unused
AH	=	16	Test for hard disk ready
AH	=	17	Recalibrate hard disk
AH	=	18	Unused
AH	=	19	Unused
AH	=	20	Controller internal diagnostic
AH	=	21	Read disk type

Reset Hard Disks

Reset the hard disk system. Reset associated hardware, and recalibrate all drives.

Entry Conditions:

AH	=	0
DL	=	drive number (80H-81H)

Exit Conditions:

See the following section, "Exits From All Hard Disk Calls."

Return Status of Last Hard Disk Operation

Return the status of the last hard disk operation in AL.

Entry Conditions:

AH	=	1
DL	>=	80H

Exit Conditions:

AL	=	status of the last operation. See the following section, "Exits From All Hard Disk Calls," for values.
----	---	--

Read Sectors From Disk

Read the specified sectors from disk into RAM.

Entry Conditions:

AH	=	2
DL	=	drive number (80H-81)
DH	=	head number (0-15)
CH	=	least significant part of cylinder number (0-1023) (See CL Bits 7, 6)
CL Bits 7, 6 =		most significant part of cylinder number
CL Bits 5-0 =		sector number (1-17)
AL	=	number of sectors to read (1-80)
ES:BX	=	pointer to disk buffer

Exit Conditions:

See the following section named "Exits From All Hard Disk Calls."

Write Sectors To Disk

Write the specified sectors from RAM to disk.

Entry Conditions:

AH =	3
DL =	drive number (80H-81H)
DH =	head number (0-15)
CH =	least significant part of cylinder number (0-1023) (See CL Bits 7, 6)
CL Bits 7, 6 =	most significant part of cylinder number
CL Bits 5-0 =	sector number (1-17)
AL =	number of sectors to write (1-80 hex)
ES:BX =	pointer to disk buffer

Exit Conditions:

See the following section, "Exits From All Hard Disk Calls."

Verify Sectors On Disk

Verify the specified sectors.

Entry Conditions:

AH =	4
DL =	drive number (80H-81H)
DH =	head number (0-15)
CH =	least significant part of cylinder number (0-1023) (See CL Bits 7, 6)
CL Bits 7, 6 =	most significant part of cylinder number
CL Bits 5-0 =	sector number (1-17)
AL =	number of sectors to verify (1-80 hex)

Exit Conditions:

See the following section, "Exits From All Hard Disk Calls."

Format Track On Disk

Format the specified track

Entry Conditions:

AH	=	5
DL	=	drive number (80H-81H)
DH	=	head number (0-15)
CH	=	least significant part of cylinder number (01023) (See CL Bits 7, 6)
CL Bits 7, 6 =		most significant part of cylinder number
ES:BX	=	pointer to sector format table consisting of two bytes for each sector. The first byte is 00 for a good sector, and 80H for a bad sector. The second byte is the sec- tor number. For a 17-sector track with an interleave of 2, the table is:

00H,01H, 00H,0AH, 00H,02H, 00H,0BH, 00H,03H, 00H,0CH,
00H,04H, 00H,0DH, 00H,05H, 00H,0EH, 00H,06H, 00H,0FH,
00H,07H, 00H,10H, 00H,08H, 00H,11H, 00H,09H

Exit Conditions:

See the following section, "Exits From All Hard Disk Calls."

Return Current Hard Disk Parameters

Return the current parameter values for the specified hard disk.

Entry Conditions:

AH	=	8
DL	=	drive number (80H-81H)

Exit Conditions:

DL = number of hard disk drives attached
DH = maximum usable value for head number
CH = least significant part of maximum usable value for cylinder number
CL
Bits 7, 6 = most significant part of maximum usable value for cylinder number
CL
Bits 5-0 = maximum usable value for sector number

See also the following section, "Exits From All Hard Disk Calls."

Initialize Hard Disk Parameters

(Use this function only with the Tandy 3000)

Initialize the hard disk parameters for the specified hard disk. The values are taken from the current drive parameter table pointed to by INT 41H for Drive 80H, and INT 46H for Drive 81H.

Entry Conditions:

AH = 9
DL = drive number (80H-81H)

Exit Conditions:

See the following section, "Exits From All Hard Disk Calls."

Read Long

(Use this function only with the Tandy 3000)
Read specified sectors and four ECC bytes per sector from disk
to RAM.

Entry Conditions:

AH	=	10
DL	=	drive number (80H-81H)
DH	=	head number (0-15)
CH	=	least significant part of cylinder number (0-1023). (See CL Bits 7, 6)
CL		
Bits 7, 6	=	most significant part of cylinder number
CL		
Bits 5-0	=	sector number (1 -17)
AL	=	number of sectors to read (1-79 hex)
ES:BX	=	pointer to disk buffer

Exit Conditions:

See the following section, "Exits From All Hard Disk Calls."

Write Long

Write specified sectors plus four ECC bytes per sector from RAM
to disk.

Entry Conditions:

AH	=	11
DL	=	drive number (80H-81H)
DH	=	head number (0-15)
CH	=	least significant part of cylinder number (0-1023). (See CL Bits 7, 6)
CL		
Bits 7, 6	=	most significant part of cylinder number
CL		
Bits 5-0	=	sector number (1 -17)
AL	=	number of sectors to write (1-79 hex)
ES:BX	=	pointer to disk buffer

Exit Conditions:

See the following section, “Exits From All Hard Disk Calls.”

Perform Seek On Hard Disk

Seek to the specified hard disk track.

Entry Conditions

AH	=	12
DL	=	drive number (80H-81H)
DH	=	head number (0-15)
CH	=	least significant part of cylinder number (0-1023). (See CL Bits 7, 6)
CL		
Bits 7, 6	=	most significant part of cylinder number
ES:BX	=	pointer to disk buffer

Exit Conditions:

See the following section, “Exits From All Hard Disk Calls.”

Reset Hard Disk

Reset the hard disk system. Reset associated hardware, and recalibrate all hard disk drives.

Entry Conditions:

AH	=	13
DL	=	drive number (80H-81H)

Exit Conditions:

See the following section, “Exits From All Hard Disk Calls.”

Test For Hard Disk Ready

Check to see if the specified hard disk drive is ready.

Entry Conditions:

AH = 16
DL = drive number (80H-81H)

Exit Conditions:

See the following section, "Exits From All Hard Disk Calls."

Recalibrate Hard Disk

Recalibrate the specified hard disk drive.

Entry Conditions:

AH = 17
DL = drive number (80H-81H)

Exit Conditions:

See the following section, "Exits From All Hard Disk Calls."

Controller Internal Diagnostic

Perform internal diagnostic on the hard disk controller.

Entry Conditions:

AH = 20
DL = drive number (80H-81H)

Exit Conditions:

See the following section, "Exits From All Hard Disk Calls."

Read Disk Type

Return the disk type and the number of sectors available on the hard disk.

Entry Conditions:

AH	=	21
DL	=	drive number (80H-81H)

Exit Conditions:

AH	0	= Not present
	3	= Hard disk drive
CX:DX	=	Number of sectors available on the hard disk. (Assumes 512 byte sectors.)

Exits From All Hard Disk Calls

[NC]	=	operation was successful (AH = 0)
[C]	=	operation failed
AH	01H	= Bad command or parameter
	02H	= Address mark not found
	04H	= Sector not found
	05H	= Reset failed
	07H	= Drive parameter command failed
	09H	= DMA boundary error
	0AH	= Bad sector flag detected
	10H	= Bad ECC on disk read
	11H	= Data corrected during read
	20H	= Controller failed
	40H	= Seek failed
	80H	= Device timeout
	AAH	= Drive not ready
	BBH	= Undefined error occurred
	CCH	= Write error

Serial Communications

These routines provide asynchronous byte stream I/O to and from the RS-232C serial communications port.

Software Interrupt: 14 hex (20 dec)

Function Summary:

AH	=	0	Initialize comm port
AH	=	1	Transmit character
AH	=	2	Receive character
AH	=	3	Get current comm status
DX	=		communication port number (0 or 1)

Function Descriptions:

Initialize Comm Port

Initialize the communication port according to the parameters in AL and DX.

Entry Conditions:

AH	=	0
AL	=	RS-232C parameters, as follows:
DX	=	Communications port number (0 or 1)

7 6 5	4 3	2	1 0
Baud Rate	Parity	Stop Bits	Word Length

000 = 110 baud	x0 = none	0 = 1 bit	10 = 7 bits
001 = 150 baud	01 = odd	1 = 2 bits	11 = 8 bits
010 = 300 baud	11 = even		
011 = 600 baud			
100 = 1200 baud			
101 = 2400 baud			
110 = 4800 baud			
111 = 9600 baud			



Exit Conditions:

AX = RS-232 status; see "Get Current Comm Status" (AH = 3)

Transmit Character

Transmit (output) the character in AL

Entry Conditions:

AH = 1
AL = character to transmit
DX = port number (0 or 1)

Exit Conditions:

AH = RS-232 status; see "Get Current Comm Status (AH = 3). If Bit 7 is set, the routine is unable to transmit the character because of a timeout error.

AL is preserved.

Receive Character

Receive (input) a character in AL (wait for a character, if necessary). On exit, AH contains the RS-232 status, except that only the error bits (1, 2, 3, 4, 7) can be set; the timeout Bit (7), if set, indicates that data set ready was not received. Thus, AH is non-zero only when an error occurs. If bit 7 is set, the other bit values are not meaningful.

Entry Conditions:

AH = 2
DX = port number (0 or 1)

Exit Conditions:

AL = character received
AH = RS-232 status; see "Get Current Comm Status" (AH = 3)



Get Current Comm Status

Read the communication status in AX.

Entry Conditions:

AH = 3
DX = port number (0 or 1)

Exit Conditions:

AH = RS-232 status, as follows (set = true):
Bit 0 = data ready
Bit 1 = overrun error
Bit 2 = parity error
Bit 3 = framing error
Bit 4 = break detect
Bit 5 = transmitter holding register empty
Bit 6 = transmitter shift register empty
Bit 7 = timeout occurred

AL = modem status, as follows (set = true):
Bit 0 = delta clear to send
Bit 1 = delta data set ready
Bit 2 = trailing edge ring detector
Bit 3 = delta receive line signal detect
Bit 4 = clear to send
Bit 5 = data set ready
Bit 6 = ring indicator
Bit 7 = receive line signal detect

Multi-Tasking Support

Hooks, Joystick Support, Microsecond Delay, 80286 Virtual Mode
Support, Extended Memory Support

Software Interrupt: 15 hex (21 dec)

Function Summary:

Device Open **Entry Conditions:**

AH = 80H
BX = Device ID
CX = Process ID

Device Close **Exit Conditions:** None

Entry Conditions:

AH = 81H
BX = Device ID
CX = Process ID

Exit Conditions: None

Program Terminate **Entry Conditions:**

AH = 82H
BX = Device ID

Exit Conditions: None

Event Wait

Entry Conditions:

AH = 83H	
AL = 0	Set interval
ES:BX	= Pointer to a byte in caller's memory that has the sign bit set after the time interval expires.
CX,DX	= The number of microseconds to elapse before the high order byte in the caller's memory pointed to by ES:BX is set.
AL = 1	Cancel event wait

Exit Conditions: None

Joystick Support

Entry Conditions:

AH = 84H	
DX = 0	Read the current joystick switch setting. On return, AL contains the switch settings in Bits 4-7.
DX = 1	Read the resistive inputs of the game adapter. The following values are returned: AX = Joystick A x value BX = Joystick A y value CX = Joystick B x value DX = Joystick B y value

Entry Conditions:

AH = 85H	
AL = 0	Sys Req key pressed
DX = 1	Sys Req key released

Exit Conditions: None

Wait

Wait indicated number of microseconds

Entry Conditions:

AH = 86H Wait indicated number of microseconds
CX,DX = Number of microseconds to wait before returning.

Exit Conditions: None

Move Block of Memory

Transfers the contents of a block of memory from memory addressed above 1024K to memory addressed below 1024K. This uses the 80286 processor "protected" mode.

Entry Conditions:

AH = 87H
ES:SI = A pointer to a Global Descriptor Table (GDT) as defined in the following table. The calling program must have previously set up this table.
CX = Number of 16-bit words to be transferred. The source and destination global descriptors must contain a length field of at least (2 * CX - 1)

GDT:	Descriptor	Usage (each entry is 8 bytes)
	0	Dummy
	1	Segment address of this GDT
	2	Pointer to area from which to move data
	3	Pointer to area to receive the data
	4	BIOS code segment descriptor
	5	BIOS stack segment descriptor

Exit Conditions:

AH = 0 Operation successful
AH = 1 RAM parity error
AH = 2 Exception interrupt occurred
AH = 3 Gating of address line 20 failed
Carry flag set if error
Zero flag set if successful

Extended Memory Size

Returns the amount of extended memory starting at address 1024K.

Entry Conditions:

AH = 88H

Exit Conditions:

AX = Amount of contiguous available memory in kilobytes starting at address 1024K.

Processor to Virtual Mode

Set the 80286 processor into protected virtual memory mode.

Entry Conditions:

AH = 89H
ES:SI = Pointer to a Global Descriptor Table (GDT) set up by the calling program as defined in the following table.
BH = Offset within the Interrupt Descriptor Table (IDT) that contains the first eight interrupts
BL = Offset within the Interrupt Descriptor Table that contains the second eight hardware interrupts.

GDT:	Descriptor	Usage (each entry is 8 bytes)
	0	Dummy
	1	Segment address of this GDT
	2	Pointer to the IDT
	3	Pointer to user's data segment
	4	Pointer to user's extra segment
	5	Pointer to user's stack segment
	6	Pointer to user's code segment
	7	Pointer to BIOS code segment

Exit Conditions:

AH = 0 Operation successful

Device Busy Wait Loop

Entry Conditions:

AH	=	90H
AL	=	Type code
00H - 7FH = Non re-entrant devices		
80H - BFH = Re-entrant devices. ES:BX		
points to data block		
C0H - FFH = Wait only		

Exit Conditions: None

Interrupt Complete

Entry Conditions:

AH	=	
AL	=	Type code (See function AH = 90H)

Exit Conditions: None

Keyboard

The following routines provide access to keyboard functions.

Software Interrupt: 16 hex (22 dec)

Function Summary:

AH = 0	Read Keyboard Input
AH = 1	Read Keyboard Status
AH = 2	Read Shift Status
AH = 3	Set Typematic Rate
AH = 5	Stuff Keyboard Buffer
AH = 10	Read Extended Keyboard Input
AH = 11	Read Extended Keyboard Status
AH = 12	Read Extended Shift Status

Function Descriptions:

Read Keyboard

Read the next character typed at the keyboard. Return the ASCII value of the character and the keyboard scan code, removing the entry from the keyboard buffer (destructive read).

Entry Conditions:

AH = 0

Exit Conditions:

AL = ASCII value of character
AH = keyboard scan code

Scan Keyboard

Set up the zero flag (Z flag) to indicate whether a character is available to be read from the keyboard or not. If a character is available, return the ASCII value of the character and the keyboard scan code. The entry remains in the keyboard buffer (non-destructive read).

Entry Conditions:

AH = 1

Exit Conditions:

Z = 1 no character is available
Z = 0 a character is available, in which case:
 AL = ASCII value of character
 AH = keyboard scan code

Get Shift Status

Return the current shift status.

Entry Conditions:

AH = 2

Exit Conditions:

AL = current shift status (bit settings: set = true, reset = false)
 Bit 0 = RIGHT SHIFT key pressed
 Bit 1 = LEFT SHIFT key pressed
 Bit 2 = CTRL (control) key pressed
 Bit 3 = ALT (alternate mode) key pressed
 Bit 4 = SCROLL state active
 Bit 5 = NUMBER lock engaged
 Bit 6 = CAPS lock engaged
 Bit 7 = INSERT state active

Set Typematic Rate

Entry Conditions:

AH = 3
AL = 5
BL = Typematic rate (00 - 1F hex)
BH = Typematic delay (00 - 3 hex)

Exit Conditions: None

Stuff Keyboard Buffer

Entry Conditions:

AH = 5
CH = Scan Code to Place in Keyboard Buffer
CL = ASCII Character to Place in Keyboard Buffer

Exit Conditions:

AL = 0 Store is Successful
AL = 1 Store is Unsuccessful
CY = 1 Error

Read Extended Keyboard Input

Entry Conditions:

AH = 10

Exit Conditions:

AL = ASCII Character
AH = Scan Code

Read Extended Keyboard Status

Includes extended read interface for extended keyboard.

Entry Conditions:

AH = 11

Exit Conditions:

Z = 1 No Character Available
Z = 0 Character Available, in which case
 AL = ASCII Character
 AH = Scan Code

Read Extended Shift Status

Entry Conditions:

AH = 12

Exit Conditions:

AL = Shift Status Byte — Bit = true, (on or depressed)
— Bit = false (or off)

Bit 0 = Right Shift key
Bit 1 = Left Shift key
Bit 2 = Ctrl (control) key
Bit 3 = Alt (alternate mode) key
Bit 4 = Scroll Lock
Bit 5 = Num Lock
Bit 6 = Caps Lock
Bit 7 = Ins(ert)

AH = Extended Shift Status —
 Bit = 1 — currently depressed
 Bit = 0 — not currently depressed

Bit 0 = Left Ctrl key
Bit 1 = Left Alt key
Bit 2 = Right Ctrl key
Bit 3 = Right Alt key
Bit 4 = Scroll Lock key
Bit 5 = Num Lock key
Bit 6 = Caps Lock key
Bit 7 = SysRq key

Line Printer

These routines provide an interface to the parallel line printer.

Software Interrupt: 17 hex (23 dec)

Function Summary:

AH	=	0	Print character
AH	=	1	Initialize printer port
AH	=	2	Get current printer status

Function Descriptions:

Print Character

Print a character.

Entry Conditions:

AH	=	0
AL	=	character to be printed
DX	=	printer to be used (0-2)

Exit Conditions:

AH = printer status; see the following section
"Get Current Printer Status (AH = 2). If
Bit 0 is set, the character could not be
printed because of a timeout error."

Initialize Printer Port

Initialize the printer port.

Entry Conditions:

AH	=	1
DX	=	printer to be used (0-2)

Exit Conditions:

AH = printer status; (See the following section
"Get Current Printer Status" (AH = 2)

Get Current Printer Status

Read the printer status in AH.

Entry Conditions:

AH = 2

Exit Conditions:

AH = printer status, as follows (set = true):

Bit 0 = timeout occurred
Bit 1 = [unused]
Bit 2 = [unused]
Bit 3 = I/O error
Bit 4 = selected
Bit 5 = out of paper
Bit 6 = acknowledge
Bit 7 = not busy

(Note: If printer port specified by DX does not exist, the status value is not meaningful.)

System Clock

These routines provide the means of reading and setting the system clock tick counter, the CMOS read time clock, and the system alarm function.

Software Interrupt: 1A hex (26 dec)

Function Summary:

AH = 0	Read system clock
AH = 1	Set system clock
AH = 2	Read CMOS time of day
AH = 3	Set CMOS time of day
AH = 4	Read CMOS date
AH = 5	Set CMOS date
AH = 6	Set the alarm
AH = 7	Reset the alarm
AL = 80H	Set up sound multiplexer

Function Descriptions:

Read System Clock

Read the system clock value kept in RAM.

Note: The clock runs at a rate of 1,193,180/65,536 per second. (About 18.2 ticks per second.) 24 hours equals 1800B0h ticks.

Entry Conditions:

AH = 0

Exit Conditions:

AL =	If the timer has not exceeded 24 hours since the last time this function was called, AL contains 0, otherwise it contains 1.
CX =	high (most significant) portion of clock count.
DX =	low (least significant) portion of clock count.

Set System Clock

Set the system clock value kept in RAM.

Entry Conditions:

AH	=	1
CX	=	high (most significant) portion of clock count
DX	=	low (least significant) portion of clock count

Read CMOS Time Of Day

Read the time of day kept in CMOS.

Entry Conditions:

AH	=	2
----	---	---

Exit Conditions:

CH	=	hours in BCD
CL	=	minutes in BCD
DH	=	seconds in BCD
DL	=	daylight savings time (1 = yes, 0 = no)

Set CMOS Time Of Day

Set the time of day kept in CMOS.

Entry Conditions:

AH	=	3
CH	=	hours in BCD
CL	=	minutes in BCD
DH	=	seconds in BCD
DL	=	If daylight savings time, 1. Otherwise 0

Read CMOS Date

Read the date kept in CMOS

Entry Conditions:

AH = 4

Exit Conditions:

CH = century in BCD
CL = year in BCD
DH = month in BCD
DL = day in BCD

Set CMOS Date

Set the date kept in CMOS

Entry Conditions:

AH = 5
CH = century in BCD
CL = year in BCD
DH = month in BCD
DL = day in BCD

Set The Alarm

Set the alarm function to generate an INT 4AH at the specified time. The user must provide an alarm routine and initialize the vector for interrupt 4AH to point to this routine.

Entry Conditions:

AH = 6
CH = hours in BCD
CL = minutes in BCD
DH = seconds in BCD



Exit Conditions:

- [NC] = CMOS clock operating correctly, and no other alarm is currently in process.
- [C] = An alarm is currently in progress, or the CMOS clock is not operating correctly.

Reset The Alarm

Reset the alarm function set by Function 6.

Entry Conditions:

- AH = 7



Keyboard ASCII and Scan Codes

The table in this appendix lists the keys on the Tandy 3000 & 4000 keyboard in key number order, along with the ASCII codes they generate. For each key, the following entries are given:

Scan Code — A value which uniquely identifies the physical key on the keyboard that is pressed.

Keyboard Legend — The physical marking(s) on the key. If there is more than one marking, the upper one is listed first.

ASCII Code — The ASCII codes associated with the key. The four modes are:

Normal — The normal ASCII value (returned when only the indicated key is depressed).

SHIFT — The shifted ASCII value (returned when SHIFT is also depressed).

CTRL — The control ASCII value (returned when CTRL is also depressed).

ALT — The alternate ASCII value (returned when ALT is also depressed).

Remarks — Any remarks or special functions.

The following special symbols appear in the table:

- x Values preceded by "x" are extended ASCII codes (codes preceded by an ASCII NUL, 00).
- No ASCII code is generated but the special function described in the Remarks column is performed. If no comment is included, the key does not generate a code and no function is performed.

Note: All numeric values in the table are expressed in hexadecimal.

Keyboard Scan Codes

		AT Mode System										
AT Mode	KB Scan	Scan Code and XT Mode KB Scan		Make Break	Code	Code	NORM	SFT	CTL	ALT	KEY #	Description
76	F076	01	81	1B	1B	1B	—	—	—	—	1	Esc
05	F005	3B	BB	3B	54	5E	x068	—	—	—	2	F1
06	F006	3C	BC	3C	55	5F	X069	—	—	—	3	F2
04	F004	3D	BD	3D	56	60	X06A	—	—	—	4	F3
0C	F00C	3E	BE	3E	57	61	X06B	—	—	—	5	F4
03	F003	3F	BF	3F	58	62	X06C	—	—	—	6	F5
0B	F00B	40	C0	40	59	63	X06D	—	—	—	7	F6
83	F083	41	C1	41	5A	64	X06E	—	—	—	8	F7
0A	F00A	42	C2	42	5B	65	X06F	—	—	—	9	F8
01	F001	43	C3	43	5C	66	X070	—	—	—	10	F9
09	F009	44	C4	44	5D	67	X071	—	—	—	11	F10
78	F078	57	D7	57	76	7F	X08A	—	—	—	12	F11
07	F007	58	D8	58	77	80	X08B	—	—	—	13	F12
E0,12	E0,F0,14											
E0,76	F0,77,7737	B7	2A	—	72	—	—	—	—	—	14	Print
7E	F07E	46	C6	—	—	—	—	—	—	—	15	Scan Scroll Lock
E1,14,77												
E1,F0,		E1,	E1									
14,40,		10,	9D									
77,77,		45	C5	—	—	—	—	—	—	—	16	Pause Break
OE	F00E	29	A9	60	7E	—	—	—	—	—	17	- or /
16	F016	02	82	31	21	—	X078	—	—	—	18	! or 1
1E	F01E	03	83	32	40	—	X079	—	—	—	19	@ or 2
26	F026	04	84	33	23	—	X07A	—	—	—	20	# or 3
25	F025	05	85	34	24	—	X07A	—	—	—	21	\$ or 4
2E	F02E	06	86	35	25	—	X07C	—	—	—	22	% or 5
36	F036	07	87	36	5E	—	X07D	—	—	—	23	' or 6
3D	F03D	08	88	37	26	—	X07E	—	—	—	24	& or 7
3E	F03E	09	89	38	2A	—	X07F	—	—	—	25	* or 8
46	F046	0A	8A	39	28	—	X080	—	—	—	26	(or 9
45	F045	0B	8B	40	29	—	X081	—	—	—	27) or 0
4E	F04E	0C	8C	2D	5F	1F	X082	—	—	—	28	_or -
55	F055	0D	8D	3D	2B	—	X083	—	—	—	29	+ or =
66	F066	0E	8E	08	08	7F	—	—	—	—	30	Backspace
70	F070	52	D2	52	—	—	—	—	—	—	31	Insert

AT Mode		AT Mode System							KEY #	Description		
KB Scan Code	Break Code	Scan Code and XT Mode KB Scan		NORM	SFT	ASCII	CTL	ALT				
		Make	Break									
6C	F06C	47	C7	47	—	X077	—	—	32	Home/Clr Scn		
7D	F07D	49	C9	X049	—	X084	—	—	33	Pg Up/Top of Text		
77	F077	45	C5	—	—	—	—	—	34	Num lock		
4A	F04A	35	B5	2F	—	—	—	—	35	/		
7C	F07C	37	B7	2A	—	72	—	—	36	*/Prt Scrn/ CPSCRN		
7B	F07B	4A	CA	2D	—	—	—	—	37	—		
0D	F00D	0F	8F	09	00F	—	—	—	38	Tab		
15	F015	10	90	71	51	11	X010	—	39	Q or q		
1D	F01D	11	91	77	57	17	X011	—	40	W or w		
24	F024	12	92	65	45	05	X012	—	41	E or e		
2D	F02D	13	93	72	52	12	X013	—	42	R or r		
2C	F02C	14	94	74	54	14	X014	—	43	T or t		
35	F035	15	95	79	59	19	X015	—	44	Y or y		
3C	F03C	16	96	75	55	15	X016	—	45	U or u		
43	F043	17	97	69	49	09	X017	—	46	I or i		
44	F044	18	98	6F	4F	0F	X018	—	47	O or o		
4D	F04D	19	99	70	50	10	X019	—	48	P or p		
54	F054	1A	9A	5B	7B	1B	—	—	49	{ or [
5B	F05B	1B	9B	5D	7D	1D	—	—	50	} or]		
5D	F05D	2B	AB	5C	7C	1C	—	—	51	or \		
71	F071	53	D3	X053	—	—	—	—	52	Delete		
69	F069	4F	CF	X04F	—	X075	—	—	53	End		
7A	F07A	51	D1	X051	—	X076	—	—	54	Pg Dn		
6C	F06C	47	C7	X047	37	X077	—	—	55	7, Home		
75	F075	48	C8	X048	38	—	—	—	56	8 or ▲		
7D	F07D	49	C9	X049	39	X084	—	—	57	9 or PgUp		
79	F079	4E	CE	2B	—	—	—	—	58	+		
58	F058	3A	BA	—	—	—	—	—	59	Caps Lock		
1C	F01C	1E	9E	61	41	01	X01E	—	60	A or a		
1B	F01B	1F	9F	73	53	13	X01F	—	61	S or s		
23	F023	20	A0	64	44	04	X020	—	62	D or d		
2B	F02B	21	A1	66	46	06	X021	—	63	F or f		
34	F034	22	A2	67	47	07	X022	—	64	G or g		
33	F033	23	A3	68	48	08	X023	—	65	H or h		
3B	F03B	24	A4	6A	4A	0A	X024	—	66	J or j		
42	F042	25	A5	6B	4B	0B	X025	—	67	K or k		
4B	F04B	26	A6	6C	4C	0C	X026	—	68	L or l		
4C	F04C	27	A7	3B	3A	—	—	—	69	: or ;		



AT Mode System

Scan Code and

XT Mode KB Scan

AT Mode KB Scan	Make Code	Break Code	Make Code	Break Code	NORM	SFT	ASCII	KEY #	Description
52 F052	28	A8	27	22	—	—	—	70	" or '
5A F05A	1C	9C	0D	0D	0A	—	—	71	Enter
6B F06B	4B	CB	X04B	34	X073	—	—	72	4 or ▲
73 F073	4C	CC	—	35	—	—	—	73	5
74 F074	4D	CD	X04D	36	X074	—	—	74	6 or ▼
12 F012	3A	AA	—	—	—	—	—	76	Shift
1A F01A	2C	AC	7A	5A	1A	X02C	77	Z or z	
22 F022	2D	AD	78	58	18	X020	78	X or x	
21 F021	2E	AE	63	43	03	X02E	79	C or c	
2A F02A	2F	AF	76	56	16	X02F	80	V or v	
32 F032	30	B0	62	42	02	X030	81	B or b	
31 F031	31	B1	6E	4E	0E	X031	82	N or n	
3A F03A	32	B2	6D	4D	0D	X032	83	M or m	
41 F041	33	B3	2C	3C	—	—	84	< or ,	
49 F049	34	B4	2E	3E	—	—	85	> or .	
4A F04A	35	B5	2F	3F	—	—	86	? or /	
59 F059	36	B6	—	—	—	—	87	Shift	
75 F075	48	C8	X048	—	—	—	88	▲	
69 F069	4F	CF	X04F	31	X075	—	89	1 or End	
72	50						98	2 or ▼	
7A F07A	51	D1	X051	33	X076	—	91	3 or Pg Dn	
14 F014	1D	9D	—	—	—	—	92	Ctrl	
11 F011	38	B8	—	—	—	—	93	Alt	
29 F029	39	B9	20	20	20	X020	94	Space	
11 F011	E0,	E0,							
38		B8	—	—	—	—	95	Alt	
14 F014	1D	9D	—	—	—	—	96	Ctrl	
6B F06B	4B	CB	X04B	—	X073	—	97	▲	
72 F072	50	D0	X050	—	X075	—	98	▼	
74 F074	E0,	E0,							
		4D	CD	X04D	—	X074	—	99	►
70 F070	52	D2	X052	30	—	—	100	0 or Ins	
71 F071	53	D3	X053	2E	—	—	101	. or Del	
5A F05A	1C	9C	0D	0D	0A	—	102	Enter	

Memory Map

High-Level Memory Map

HEX ADDRESS 0-16 MEG	DESCRIPTION
000000	Interrupt Vector Table (INTs 00h - FFh)
000400	ROM BIOS Data Area
000500	MS-DOS & BASIC Data Area
000600	RESERVED
000700	MS-DOS interface drivers to ROM BIOS I/O routines (IBMBIO.COM)
Note 1	MS-DOS interrupt handler service routines (INT 21h functions)
Note 1	MS-DOS structures, buffers, stack space and installable device drivers
Note 1	Resident portion of command interpreter (COMMAND.COM), Interrupt handlers for INT 22h (terminate), INT 23h (Control-Break), INT 24h (Critical Error) and code to reload the transient portion of COMMAND.COM.
Note 1	Transient Program Area. Used by external commands and utilities.
Note 1	Transient portion of COMMAND.COM Command interpreter, internal commands, batch processor, external command loader.
Note 2	End of conventional RAM
0A0000	RESERVED for EGA type display buffer
0B0000	RESERVED for MDPA type display buffer
0B8000	RESERVED for CGA type display buffer
0C0000	RESERVED for ROM BIOS extensions on I/O adapters
0C8000	ROM BIOS Initialization Diagnostic Routines
100000	Extended (upper Meg) RAM
FF8000	Duplicated code assignment of addresses 0F8000- 0FFFFF

Notes:

- 1 These addresses are not absolute as they depend on the hardware installed and/or the software driver configuration. But the relative order of these items is preserved.
- 2 The Transient Program Area extends to all of the conventional RAM installed. Maximum of 640K.

Interrupt Vector Table

HEX ADDRESS SEQ. OFFSET	INTERRUPT NUMBER	DESCRIPTION
0:0	00	Divide by zero (CPU generated)
0:4	01	Single Step (CPU generated)
0:8	02	Non-Maskable Interrupt (H/W generated)
0:C	03	Software Break Point Interrupt
0:10	04	Arithmetic Overflow (CPU generated)
0:14	05	Print Screen
0:18	06	Invalid Op-code Exception (CPU generated)
0:1C	07	Processor Extension not available Exception (CPU generated)
0:20	08	Timer Clock Tick (H/W generated IRQ0)
0:24	09	Keyboard (H/W generated IRQ1)
0:28	0A	I/O Channel available (H/W generated IRQ9) re-directed from Slave Interrupt Controller
0:30	0B	I/O Channel available, usually secondary serial communications (H/W generated IRQ3)
0:34	0C	I/O Channel available, usually primary serial communications (H/W generated IRQ4)
0:2C	0D	I/O Channel available (H/W generated IRQ5)
0:38	0E	Floppy Disk (H/W generated IRQ6)
0:3C	0F	Printer (H/W generated IRQ7)
0:40	10	Video BIOS I/O Call
0:44	11	BIOS Equipment Check I/O Call
0:48	12	BIOS Memory Size I/O Call
0:4C	13	Diskette Disk BIOS I/O Call
0:50	14	RS-232 Communications BIOS I/O Call
0:54	15	I/O Systems BIOS Extension Call
0:58	16	Keyboard BIOS I/O Call
0:5C	17	Printer BIOS I/O Call
0:60	18	Boot Stap Loader Failure Hook
0:64	19	Boot Strap Loader
0:68	1A	Time Date BIOS I/O Call
0:6C	1B	Keyboard Control-Break Hook
0:70	1C	Timer Clock Tick Hook
0:74	1D	Video Display Parameters Pointer
0:78	1E	Floppy Diskette Parameters Pointer
0:7C	1F	Video Graphics Characters Pointer (Characters 128-255 only)

Interrupt Vector Table

HEX ADDRESS SEQ. OFFSET	INTERRUPT NUMBER	DESCRIPTION
0:80	20	MS-DOS Program Terminate
0:84	21	MS-DOS Functions I/O Call
0:88	22	MS-DOS Programs' Terminate Routine
0:8C	23	MS-DOS Programs' Control-Break Exit Routine
0:90	24	MS-DOS Programs' Critical Error Routine
0:94	25	MS-DOS Absolute Disk Read Call
0:98	26	MS-DOS Absolute Disk Write Call
0:9C	27	MS-DOS Terminate and Stay Resident Call
0:A0	28-3F	RESERVED for MS-DOS
0:100	40	Re-vectoried Diskette BIOS INT 13h I/O Call when Fixed Disk is Present
0:104	41	Fixed Disk Drive #1 parameters pointer
0:108	42	Re-vectoried Video BIOS INT 10h I/O Call when optional video ROM, is present.
0:10C	43	EGA Video Display parameters
0:110	44-45	RESERVED for BIOS
0:118	46	Fixed Disk Drive #2 parameters pointer
0:11C	47-5F	RESERVED
0:180	60-66	RESERVED
0:19C	67	RESERVED for Lotus/Intel/Microsoft Expanded Memory Manager Support
0:1A0	68-6F	Not Used
0:1C0	70	Real Time Clock (H/W generated IRQ8)
0:1C4	71	Re-direction of old PC style INT 0Ah (IRQ2)
0:1C8	72	I/O Channel Available (H/W generated IRQ 10)
0:1CC	73	I/O Channel Available (H/W generated IRQ 11)
0:1D0	74	I/O Channel Available (H/W generated IRQ 12)
0:1D4	75	Numeric Co-Processor (H/W generated IRQ 13)
0:1D8	76	Fixed Disk (H/W generated IRQ 14)
0:1DC	77	I/O Channel Available (H/W generated IRQ 15)
0:1E0	78-7F	Not Used
0:200	80-85	RESERVED by BASIC
0:218	86-F0	Used by BASIC
0:3C4	F1-FF	Not Used

ROM BIOS Data Area

STARTING ADDRESS	LENGTH IN BYTES	DESCRIPTION																		
40:0	8	RS-232 Communications Adapter Installed Base Address Table (max. of 4 entries)																		
40:8	8	Parallel Printer Adapter Installed Base Address Table (max. of 4 entries)																		
40:10	2	Installed Equipment Flag (same as INT 11h return word)																		
40:12	1	Initialization/Diagnostic scratch byte																		
40:13	2	System Memory Size in kilobytes (640K max.)																		
40:15	2	Initialization/Diagnostic scratch word																		
40:17	1	Primary Keyboard Shift Status																		
		<table border="1"> <thead> <tr> <th>Bit</th><th>Description</th></tr> </thead> <tbody> <tr> <td>7</td><td>1 = Insert state is on</td></tr> <tr> <td>6</td><td>1 = Caps Lock state is on</td></tr> <tr> <td>5</td><td>1 = Num Lock state is on</td></tr> <tr> <td>4</td><td>1 = Scroll Lock state is on</td></tr> <tr> <td>3</td><td>1 = Alt key is depressed</td></tr> <tr> <td>2</td><td>1 = Ctrl key is depressed</td></tr> <tr> <td>1</td><td>1 = Left Shift key is depressed</td></tr> <tr> <td>0</td><td>1 = Right Shift key is depressed</td></tr> </tbody> </table>	Bit	Description	7	1 = Insert state is on	6	1 = Caps Lock state is on	5	1 = Num Lock state is on	4	1 = Scroll Lock state is on	3	1 = Alt key is depressed	2	1 = Ctrl key is depressed	1	1 = Left Shift key is depressed	0	1 = Right Shift key is depressed
Bit	Description																			
7	1 = Insert state is on																			
6	1 = Caps Lock state is on																			
5	1 = Num Lock state is on																			
4	1 = Scroll Lock state is on																			
3	1 = Alt key is depressed																			
2	1 = Ctrl key is depressed																			
1	1 = Left Shift key is depressed																			
0	1 = Right Shift key is depressed																			
40:18	1	Secondary Keyboard Shift Status																		
		<table border="1"> <thead> <tr> <th>Bit</th><th>Description</th></tr> </thead> <tbody> <tr> <td>7</td><td>1 = Insert key is depressed</td></tr> <tr> <td>6</td><td>1 = Caps Lock key is depressed</td></tr> <tr> <td>5</td><td>1 = Num Lock key is depressed</td></tr> <tr> <td>4</td><td>1 = Scroll Lock key is depressed</td></tr> <tr> <td>3</td><td>1 = Suspend Mode is active</td></tr> <tr> <td>2</td><td>1 = Sys Req key is depressed</td></tr> <tr> <td>1</td><td>1 = Left Alt key is depressed</td></tr> <tr> <td>0</td><td>1 = Left Ctrl key is depressed</td></tr> </tbody> </table>	Bit	Description	7	1 = Insert key is depressed	6	1 = Caps Lock key is depressed	5	1 = Num Lock key is depressed	4	1 = Scroll Lock key is depressed	3	1 = Suspend Mode is active	2	1 = Sys Req key is depressed	1	1 = Left Alt key is depressed	0	1 = Left Ctrl key is depressed
Bit	Description																			
7	1 = Insert key is depressed																			
6	1 = Caps Lock key is depressed																			
5	1 = Num Lock key is depressed																			
4	1 = Scroll Lock key is depressed																			
3	1 = Suspend Mode is active																			
2	1 = Sys Req key is depressed																			
1	1 = Left Alt key is depressed																			
0	1 = Left Ctrl key is depressed																			
40:19	1	Alt-numpad accumulator																		
40:1A	2	Keyboard Buffer Head pointer																		

ROM BIOS Data Area

STARTING ADDRESS	LENGTH IN BYTES	DESCRIPTION
40:1C	2	Keyboard Buffer Tail pointer
40:1E	20h	15 entry Keyboard Buffer
40:3E	1	Floppy Disk Recalibration Status
		Bit Description
		7 1 = Floppy disk operation completed
		6-2 Not Used
		1 0 = Floppy Drive 1 needs re-calibration
		0 0 = Floppy Drive 0 needs re-calibration
40:3F	1	Floppy Disk Motor On Status
		Bit Description
		7-2 Not Used
		1 1 = Floppy Disk Drive 1 is currently on
		0 1 = Floppy Disk Drive 0 is currently on
40:40	1	Floppy disk motor On Time-out Value-Timer Clock ticks left until diskette motors are turned off
40:41	7	Floppy Disk S/W Status of last floppy disk INT 13h I/O operation
40:42	7	Floppy Disk Controller Status Byte Area
40:49	1	Current BIOS video mode
40:4A	2	Current BIOS video screen column width
40:4C	2	Current BIOS video screen buffer length in bytes
40:4E	2	Current BIOS video display page starting offset address
40:50	10h	Row/Column cursor coordinates for each of up to 8 display pages
40:60	2	Current BIOS video cursor mode setting
40:62	1	Current BIOS video display page
40:63	2	Base I/O port address of the active display card
40:65	1	Copy of value written to Mode Selected Register
40:66	1	Copy of value written to Color Palette Register

ROM BIOS Data Area

STARTING ADDRESS	LENGTH IN BYTES	DESCRIPTION
40:67	4	ROM initialization code optional ROM offset/ segment address storage
40:6B	1	Last unexpected hardware interrupt that occurred or FFh if it was an unexpected software interrupt
40:6C	2	Least significant word of Timer Clock Tick count
40:6F	2	Most significant word of Timer Clock Tick count
40:70	1	24-hour Timer Clock Tick rollover counter.
40:71	1	Keyboard Control-Break flag
		Bit Description
		7 1 = Keyboard Break
		6-0 Not Used
40:72	2	Warm Restart flag =1234h, if it's a warm start =anything else if it's a cold boot
40:74	1	Hard Disk S/W status of last INT 13h I/O operation
40:75	1	# of Hard Disk attached to system
40:76	1	Copy of Hard Disk control byte
40:77	1	Hard Disk port offset
40:78	4	Parallel Printer time-out table (max. of 4 entries)
40:7C	4	RS-232 Communications time-out table (max. of 4 entries)
40:80	2	Keyboard buffer start offset location pointer (offset from 40:0)
40:82	2	Keyboard buffer end offset location pointer (offset from 40:0)
40:84	1	EGA # of CRT Rows — 1
40:85	2	EGA # of bytes/character in EGA Font Table

ROM BIOS Data Area

STARTING ADDRESS	LENGTH IN BYTES	DESCRIPTION
40:87	1	EGA — Miscellaneous Information Byte 1
		Bit Description
	7	High bit of video mode, 1 = don't clear buffer display
	6,5	EGA memory size in 64K chunks 00=64K, 01=128K, 10=192K, 11=256K
	4	RESERVED
	3	0 = EGA active, 1 = not active
	2	0 = Write to CRT anytime 1 = Wait for display enable
	1	0 = Color monitor attached to EGA card 1 = monochrome monitor attached to EGA card
	0	0 = emulate cursor type
40:88	1	EGA Miscellaneous Information Byte 2
		Bit Description
	7-4	EGA Feature Bits
	30	EGA DIP switches
40:89	2	RESERVED
40:8B	1	Last floppy disk rate selected
40:8C	1	Hard disk status register copy
40:8D	1	Hard disk error register copy
40:8E	1	Hard disk interrupt flag
40:8F	1	Multi-rate Floppy disk controller Flag
		Bit Description
	7-1	RESERVED
	0	Always = 1
40:90	2	Floppy drive 0/1 Media State Table
40:92	2	Floppy drive 0/1 Operation State Table
40:94	2	Floppy drive 0/1 Current Track Table

ROM BIOS Data Area

STARTING ADDRESS	LENGTH IN BYTES	DESCRIPTION																		
		Bit Description																		
40:96	1	Enhanced Keyboard Flags																		
		<table border="1"> <thead> <tr> <th>Bit</th><th>Description</th></tr> </thead> <tbody> <tr> <td>7</td><td>1 = Read ID bytes command in progress</td></tr> <tr> <td>6</td><td>1 = First of ID bytes was last</td></tr> <tr> <td>5</td><td>1 = Force Num Lock if 101-keyboard</td></tr> <tr> <td>4</td><td>1 = 101-key keyboard is attached</td></tr> <tr> <td>3</td><td>1 = Right ALT key is depressed</td></tr> <tr> <td>2</td><td>1 = Right Ctrl key is depressed</td></tr> <tr> <td>1</td><td>1 = Keyboard controller E0 command was last</td></tr> <tr> <td>0</td><td>1 = Keyboard controller E1 command was last</td></tr> </tbody> </table>	Bit	Description	7	1 = Read ID bytes command in progress	6	1 = First of ID bytes was last	5	1 = Force Num Lock if 101-keyboard	4	1 = 101-key keyboard is attached	3	1 = Right ALT key is depressed	2	1 = Right Ctrl key is depressed	1	1 = Keyboard controller E0 command was last	0	1 = Keyboard controller E1 command was last
Bit	Description																			
7	1 = Read ID bytes command in progress																			
6	1 = First of ID bytes was last																			
5	1 = Force Num Lock if 101-keyboard																			
4	1 = 101-key keyboard is attached																			
3	1 = Right ALT key is depressed																			
2	1 = Right Ctrl key is depressed																			
1	1 = Keyboard controller E0 command was last																			
0	1 = Keyboard controller E1 command was last																			
40:97	1	Keyboard LED and Flags data area																		
		<table border="1"> <thead> <tr> <th>Bit</th><th>Description</th></tr> </thead> <tbody> <tr> <td>7</td><td>1 = Indicates 3 failures of sending data to keyboard</td></tr> <tr> <td>6</td><td>1 = LED Update in progress</td></tr> <tr> <td>5</td><td>1 = re-send received from keyboard</td></tr> <tr> <td>4</td><td>1 = acknowledge received from keyboard</td></tr> <tr> <td>3</td><td>1 = Not Used</td></tr> <tr> <td>2</td><td>1 = Caps Lock LED is on</td></tr> <tr> <td>1</td><td>1 = Num Lock LED is on</td></tr> <tr> <td>0</td><td>1 = Scroll Lock LED is on</td></tr> </tbody> </table>	Bit	Description	7	1 = Indicates 3 failures of sending data to keyboard	6	1 = LED Update in progress	5	1 = re-send received from keyboard	4	1 = acknowledge received from keyboard	3	1 = Not Used	2	1 = Caps Lock LED is on	1	1 = Num Lock LED is on	0	1 = Scroll Lock LED is on
Bit	Description																			
7	1 = Indicates 3 failures of sending data to keyboard																			
6	1 = LED Update in progress																			
5	1 = re-send received from keyboard																			
4	1 = acknowledge received from keyboard																			
3	1 = Not Used																			
2	1 = Caps Lock LED is on																			
1	1 = Num Lock LED is on																			
0	1 = Scroll Lock LED is on																			
40:98	2	Real-Time Clock user wait flag offset address																		
40:9A	2	Real-Time Clock user wait flag segment address																		
40:9C	2	Real-Time Clock wait count low word																		
40:9E	2	Real-Time Clock wait count high word																		
40:A0	1	Real-Time Clock wait active flag																		
40:A1	7	RESERVED																		
40:A8	4	EGA-pointer to table of EGA pointers																		
40:AC	9	RESERVED																		
40:B5	1	Tandy Information Flag																		
		<table border="1"> <thead> <tr> <th>Bit</th><th>Description</th></tr> </thead> <tbody> <tr> <td>7-1</td><td>RESERVED</td></tr> <tr> <td>0</td><td>Floppy Swap Drive Flag = 1, swap drives = 0, don't swap drives</td></tr> </tbody> </table>	Bit	Description	7-1	RESERVED	0	Floppy Swap Drive Flag = 1, swap drives = 0, don't swap drives												
Bit	Description																			
7-1	RESERVED																			
0	Floppy Swap Drive Flag = 1, swap drives = 0, don't swap drives																			
40:B6	4Ah	Note: Valid for 2 floppy drive systems only RESERVED																		

MS-DOS & BASIC Data Area

<u>STARTING ADDRESS</u>	<u>LENGTH IN BYTES</u>	<u>DESCRIPTION</u>
50:0	1	Print Screen Status Flag = 0, Print Screen not active or successful = 1, Print Screen in progress = FFh, Print Screen encountered error
50:1	3	RESERVED
50:4	1	MS-DOS Single Floppy Drive Mode Status Byte = 0, logical drive A: last accessed = 1, logical drive B: last accessed
50:5	FBh	RESERVED



**TANDY 4000
Appendix A
80387 Math CoProcessor**



Tandy 4000 80387 Support Version

Differences in Rev. A and Rev. C Main Logic Boards

1. 80287 Socket U25 has been removed.
2. Oscillator Clock Y5 has been removed.
3. PAL U22 has been removed.
4. PAL U53 has been added for 80387 select and decode.
5. PAL U49 has been added to generate 80386 handshake and control signals.
6. Jumper E3-E4 has been added to select 80387 as installed.

Note: The BIOS for Rev. C is not interchangeable with the BIOS for Rev. A.

80387 Numeric Processor

The optional 80387 numeric processor is a 68-pin IC that can be used in the available socket (U15) on the main logic board. Socket U15 is a 121 PGA superset of the 68-pin socket and can also be used for the Weitek 1167 Co-Processor. Chip select and address decoding for the Numeric Processor and the Weitek are provided by PAL U53. PAL U49 is used to generate the READY, BUSY, ERROR, PEREQ, and IRQ13 signals needed to interface the 80387 to the system. The numeric processor executes mathematical calculations independently, allowing the CPU to perform other tasks. It runs synchronously to the 80386, interfaces through the local CPU bus, and uses the same clock. Error reporting is interfaced through IRQ13 via the 82C206 IPC. A detailed description of the 82C206 IPC is provided in the "I/O Decode" and "Devices" sections of this manual. A detailed description of the 80387 can be found in "Appendix A" in this manual.



```

1:          PARTNO    U53;
2:          NAME      U53;
3:          DATE     12/28/87 ;
4:          REV      C ;
5:          DESIGNER R. THOMPSON ;
6:          COMPANY   Tandy ;
7:          ASSEMBLY XXXX ;
8:          LOCATION  YYYY ;
9:
10:
11:/*********************************************************/
12:/*
13:/* This device performs address decoding for the 80387 circuitry */
14:/* on the TANDY 4000. */
15:/*
16:/*********************************************************/
17:/* Allowable Target Device Types: 16L8B */
18:/*********************************************************/
19:
20:/** Inputs  */
21:
22:PIN 1      = A31      ;      /* */
23:PIN 2      = MALE     ;      /* */
24:PIN 3      = RESET3   ;      /* */
25:PIN 4      = XIOW     ;      /* */
26:PIN 5      = XA04     ;      /* */
27:PIN 6      = XA03     ;      /* */
28:PIN 7      = XA02     ;      /* */
29:PIN 8      = XA01     ;      /* */
30:PIN 9      = XA00     ;      /* */
31:PIN 11     = 287CS    ;      /* */
32:PIN 13     = NPSEL    ;      /* */
33:PIN 18     = MIO      ;      /* */
34:
35:/** Outputs */
36:
37:PIN 14     = !NPCS    ;      /* NUMERIC PROCESSOR SELECT */
38:PIN 16     = !AF32FF  ;      /* OUTPUT TO AF32 FLIP-FLOP */
39:
40:/** Declarations and Intermediate Variable Definitions */
41:
42:
43:/** Logic Equations */
44:
45:NPCS = !XIOW & !287CS & XA04 & !XA03 & !XA02 & !XA01 ;
46:
47:AF32FF = !MALE & A31 & !MIO & !NPSEL & !RESET3 ;
48:

```

Jedec Fuse Checksum (0EBC)
Jedec Transmit Checksum (6DB1)



```

1:
2:          PARTNO    U49;
3:          NAME      U49;
4:          DATE      11/16/87 ;
5:          REV       B ;
6:          DESIGNER  R. THOMPSON ;
7:          COMPANY   Tandy ;
8:          ASSEMBLY  XXXX ;
9:          LOCATION  YYYY ;
0:

11:***** */
12:/*          */
13:/* This device generates READY, BUSY, ERROR, PEREQ and IRQ13 */
14:/* on the TAND 4000 . */
15:/*          */
16:***** */
17:/* Allowable Target Device Types: 16L8B */
18:***** */
19:
20:/** Inputs */
21:
22:PIN 1      = 387ERROR ; /*          */
23:PIN 2      = 387BUSY  ; /*          */
24:PIN 3      = 387PERQ  ; /*          */
25:PIN 5      = INTA    ; /*          */
26:PIN 7      = RESET3  ; /*          */
27:PIN 8      = 386RDY  ; /* 386 SYSTEM READY */
28:PIN 9      = NPCS    ; /* NUMERIC PROCESSOR SELECT */
29:PIN 11     = INT13FF ; /* LATCHED IRQ13 FROM FLIP-FLOP */
30:
31:/** Outputs */
32:
33:PIN 12     = !386BSY ; /* BUSY TO 80386 */
34:PIN 13     = PEREQ   ; /* PEREQ TO 80386 */
35:PIN 15     = !MSKERR  ; /*          */
36:PIN 16     = BUSYFF  ; /*          */
37:PIN 17     = !ERRFF  ; /*          */
38:PIN 18     = !FB13   ; /*          */
39:PIN 19     = !386ERR  ; /*          */
40:
41:/** Declarations and Intermediate Variable Definitions */
42:
43:/** Logic Equations */
44:
45:386BSY = !387BUSY # INT13FF ;
46:
47:386ERR = !387ERROR & !MSKERR ;
48:
49:BUSYFF = !387BUSY ;
50:
51:PEREQ = 387PERQ # 387BUSY & INT13FF ;
52:
53:MSKERR = !RESET3 # 386RDY & MSKERR ;
54:
55:ERRFF = 387ERROR ;
56:
57:FB13 = !RESET3 # INTA & !NPCS ;
58:

Jedec Fuse Checksum      (462A)
Jedec Transmit Checksum (DBBA)

```



80387 80-BIT CHMOS III NUMERIC PROCESSOR EXTENSION

- High Performance 80-Bit Internal Architecture
- Implements ANSI/IEEE Standard 754-1985 for Binary Floating-Point Arithmetic
- Five to Six Times 8087/80287 Performance
- Upward Object-Code Compatible from 8087 and 80287
- Expands 80386 Data Types to Include 32-, 64-, 80-Bit Floating Point, 32-, 64-Bit Integers and 18-Digit BCD Operands
- Directly Extends 80386 Instruction Set to Include Trigonometric, Logarithmic, Exponential and Arithmetic Instructions for All Data Types
- Full-Range Transcendental Operations for SINE, COSINE, TANGENT, ARCTANGENT and LOGARITHM
- Built-In Exception Handling
- Operates Independently of Real, Protected and Virtual-8086 Modes of the 80386
- Eight 80-Bit Numeric Registers, Usable as Individually Addressable General Registers or as a Register Stack
- Available in 68-Pin PGA Package
(See Packaging Spec: Order #231369)

The Intel 80387 is a high-performance numerics processor extension that extends the 80386 architecture with floating point, extended integer and BCD data types. The 80386/80387 computing system fully conforms to the ANSI/IEEE floating-point standard. Using a numerics oriented architecture, the 80387 adds over seventy mnemonics to the 80386/80387 instruction set, making the 80386/80387 a complete solution for high-performance numerics processing. The 80387 is implemented with 1.5 micron, high-speed CHMOS III technology and packaged in a 68-pin ceramic pin grid array (PGA) package. The 80386/80387 is upward object-code compatible from the 80386/80287, 80286/80287 and 8086/8087 computing systems.

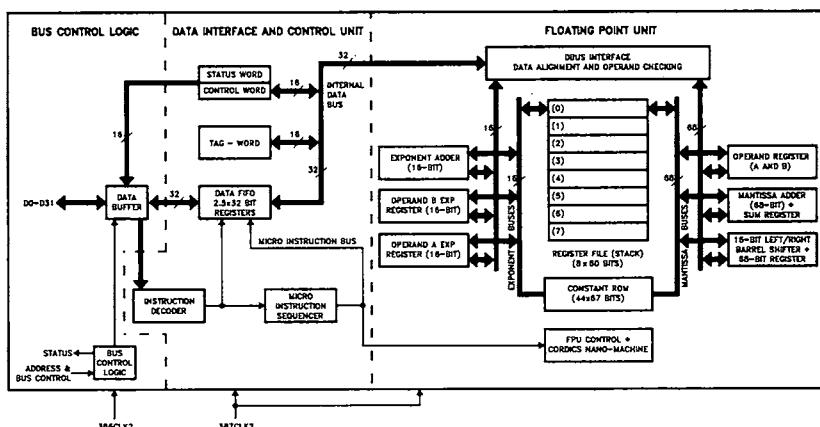


Figure 0.1. 80387 Block Diagram

231920-1



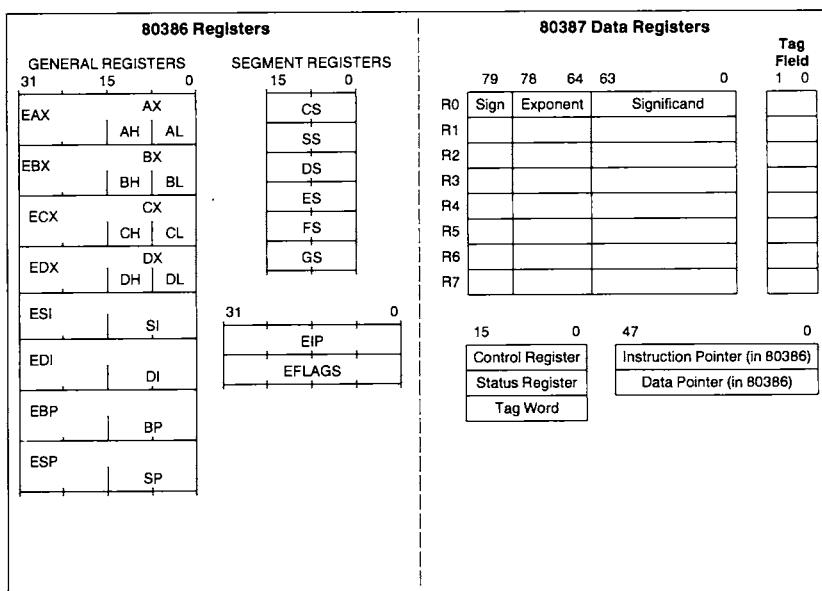


Figure 1.1. 80386/80387 Register Set

1.0 FUNCTIONAL DESCRIPTION

The 80387 Numeric Processor Extension (NPX) provides arithmetic instructions for a variety of numeric data types in 80386/80387 systems. It also executes numerous built-in transcendental functions (e.g. tangent, sine, cosine, and log functions). The 80387 effectively extends the register and instruction set of an 80386 system for existing data types and adds several new data types as well. Figure 1.1 shows the model of registers visible to 80386/80387 programs. Essentially, the 80387 can be treated as an additional resource or an extension to the 80386. The 80386 together with an 80387 can be used as a single unified system, the 80386/80387.

The 80387 works the same whether the 80386 is executing in real-address mode, protected mode, or virtual-8086 mode. All memory access is handled by the 80386; the 80387 merely operates on instructions and values passed to it by the 80386. Therefore, the 80387 is not sensitive to the processing mode of the 80386.

In real-address mode and virtual-8086 mode, the 80386/80387 is completely upward compatible with software for 8086/8087, 80286/80287 real-address mode, and 80386/80287 real-address mode systems.

In protected mode, the 80386/80387 is completely upward compatible with software for 80286/80287 protected mode, and 80386/80287 protected mode systems.

The only differences of operation that may appear when 8086/8087 programs are ported to a protected-mode 80386/80387 system (*not* using virtual-8086 mode), is in the format of operands for the administrative instructions FLDENV, FSTENV, FRSTOR and FSAVE. These instructions are normally used only by exception handlers and operating systems, not by applications programs.

The 80387 contains three functional units that can operate in parallel to increase system performance. The 80386 can be transferring commands and data to the 80387 *bus control logic* for the next instruction while the 80387 *floating-point unit* is performing the current numeric instruction.



2.0 PROGRAMMING INTERFACE

The 80387 adds to an 80386 system additional data types, registers, instructions, and interrupts specifically designed to facilitate high-speed numerics processing. To use the 80387 requires no special programming tools, because all new instructions and data types are directly supported by the 80386 assembler and compilers for high-level languages. All 8086/6088 development tools that support the 8087 can also be used to develop software for the 80386/80387 in real-address mode or virtual-8086 mode. All 80286 development tools that support the 80287 can also be used to develop software for the 80386/80387.

All communication between the 80386 and the 80387 is transparent to applications software. The CPU automatically controls the 80387 whenever a numerics instruction is executed. All physical memory and virtual memory of the CPU are available for storage of the instructions and operands of programs that use the 80387. All memory addressing modes, including use of displacement, base register, index register, and scaling, are available for addressing numerics operands.

Section 6 at the end of this data sheet lists by class the instructions that the 80387 adds to the instruction set of an 80386 system.

2.1 Data Types

Table 2.1 lists the seven data types that the 80387 supports and presents the format for each type. Operands are stored in memory with the least significant digit at the lowest memory address. Programs retrieve these values by generating the lowest address. For maximum system performance, all operands should start at physical-memory addresses evenly divisible by four (doubleword boundaries); operands may begin at any other addresses, but will require extra memory cycles to access the entire operand.

Internally, the 80387 holds all numbers in the extended-precision real format. Instructions that load operands from memory automatically convert operands represented in memory as 16-, 32-, or 64-bit integers, 32- or 64-bit floating-point numbers, or 18-digit packed BCD numbers into extended-precision real format. Instructions that store operands in memory perform the inverse type conversion.

2.2 Numeric Operands

A typical NPX instruction accepts one or two operands and produces a single result. In two-operand instructions, one operand is the contents of an NPX register, while the other may be a memory location. The operands of some instructions are predefined; for example FSQRT always takes the square root of the number in the top stack element.

Table 2.1. 80387 Data Type Representation In Memory

Data Formats	Range	Precision	Most Significant Byte								HIGHEST ADDRESSED BYTE							
			7	0	7	0	7	0	7	0	7	0	7	0	7	0	7	0
Word Integer	10^4	16 Bits																
Short Integer	10^9	32 Bits																
Long Integer	10^{19}	64 Bits																
Packed BCD	10^{18}	18 Digits	S	X	d_{17}	d_{16}	d_{15}	d_{14}	d_{13}	d_{12}	d_{11}	d_{10}	d_{9}	d_8	d_7	d_6	d_5	d_4
Single Precision	$10^{\pm 38}$	24 Bits	S		BIASED EXPONENT		SIGNIFICAND											
Double Precision	$10^{\pm 308}$	53 Bits	S		BIASED EXPONENT		SIGNIFICAND											
Extended Precision	$10^{\pm 4932}$	64 Bits	S		BIASED EXPONENT		1		SIGNIFICAND									

231920-2

NOTES:

- (1) S = Sign bit (0 = positive, 1 = negative)
- (2) d_n = Decimal digit (two per byte)
- (3) X = Bits have no significance; 80387 ignores when loading, zeros when storing
- (4)* = Position of implicit binary point
- (5) I = Integer bit of significand; stored in temporary real, implicit in single and double precision
- (6) Exponent Bias (normalized values):
 - Single: 127 (7FH)
 - Double: 1023 (3FFH)
 - Extended Real: 16383 (3FFFH)
- (7) Packed BCD: $(-1)^S (D_{17} \dots D_0)$
- (8) Real: $(-1)^S (2E-BIAS) (F_0 F_1 \dots)$

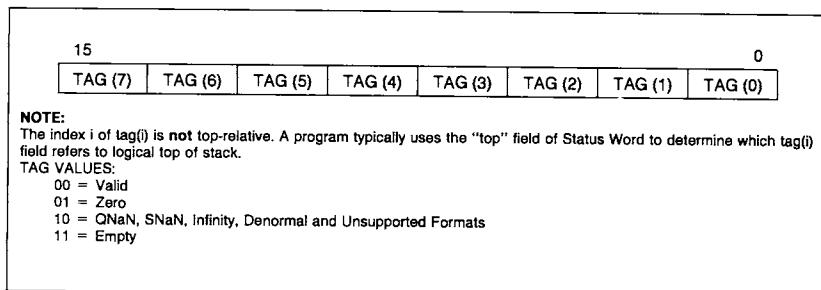


Figure 2.1. 80387 Tag Word

2.3 Register Set

Figure 1.1 shows the 80387 register set. When an 80387 is present in a system, programmers may use these registers in addition to the registers normally available on the 80386.

2.3.1 DATA REGISTERS

80387 computations use the 80387's data registers. These eight 80-bit registers provide the equivalent capacity of twenty 32-bit registers. Each of the eight data registers in the 80387 is 80 bits wide and is divided into "fields" corresponding to the NPXs extended-precision real data type.

The 80387 register set can be accessed either as a stack, with instructions operating on the top one or two stack elements, or as a fixed register set, with instructions operating on explicitly designated registers. The TOP field in the status word identifies the current top-of-stack register. A "push" operation decrements TOP by one and loads a value into the new top register. A "pop" operation stores the value from the current top register and then increments

TOP by one. Like 80386 stacks in memory, the 80387 register stack grows "down" toward lower-addressed registers.

Instructions may address the data registers either implicitly or explicitly. Many instructions operate on the register at the TOP of the stack. These instructions implicitly address the register at which TOP points. Other instructions allow the programmer to explicitly specify which register to user. This explicit register addressing is also relative to TOP.

2.3.2 TAG WORD

The tag word marks the content of each numeric data register, as Figure 2.1 shows. Each two-bit tag represents one of the eight numerics registers. The principal function of the tag word is to optimize the NPXs performance and stack handling by making it possible to distinguish between empty and nonempty register locations. It also enables exception handlers to check the contents of a stack location without the need to perform complex decoding of the actual data.

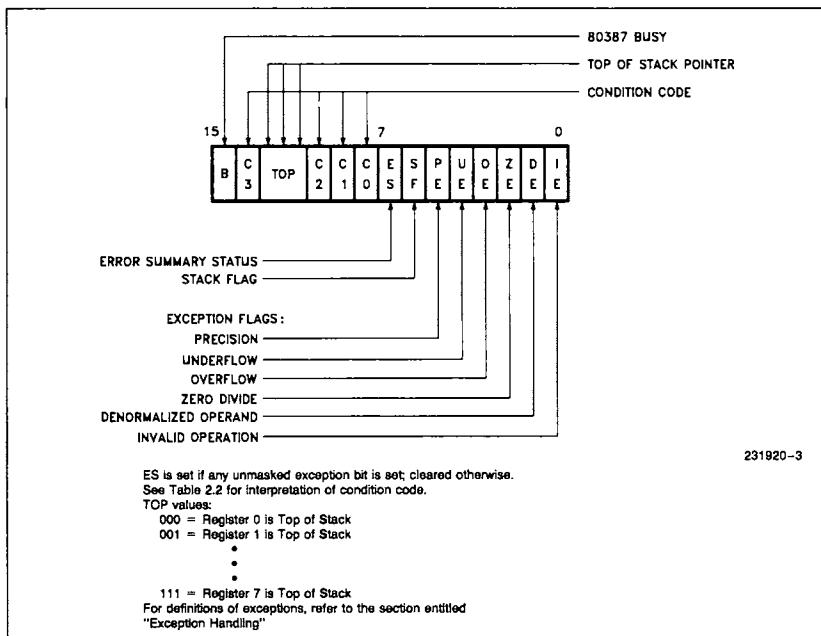


Figure 2.2. 80387 Status Word

2.3.3 STATUS WORD

The 16-bit status word (in the status register) shown in Figure 2.2 reflects the overall state of the 80387. It may be read and inspected by CPU code.

Bit 15, the B-bit (busy bit) is included for 8087 compatibility only. It reflects the contents of the ES bit (bit 7 of the status word), not the status of the BUSY# output of 80387/80287.

Bits 13-11 (TOP) point to the 80387 register that is the current top-of-stack.

The four numeric condition code bits (C_3-C_0) are similar to the flags in a CPU; instructions that perform arithmetic operations update these bits to reflect the outcome. The effects of these instructions on the condition code are summarized in Tables 2.2 through 2.5.

Bit 7 is the error summary (ES) status bit. This bit is set if any unmasked exception bit is set; it is clear otherwise. If this bit is set, the ERROR# signal is asserted.

Bit 6 is the stack flag (SF). This bit is used to distinguish invalid operations due to stack overflow or underflow from other kinds of invalid operations. When SF is set, bit 9 (C_1) distinguishes between stack overflow ($C_1 = 1$) and underflow ($C_1 = 0$).

Figure 2.2 shows the six exception flags in bits 5-0 of the status word. Bits 5-0 are set to indicate that the 80387 has detected an exception while executing an instruction. A later section entitled "Exception Handling" explains how they are set and used.

Note that when a new value is loaded into the status word by the FLDEN# or FRSTOR instruction, the value of ES (bit 7) and its reflection in the B-bit (bit 15) are not derived from the values loaded from memory but rather are dependent upon the values of the exception flags (bits 5-0) in the status word and their corresponding masks in the control word. If ES is set in such a case, the ERROR# output of the 80387 is activated immediately.

Table 2.2. Condition Code Interpretation

Instruction	C0 (S)	C3 (Z)	C1 (A)	C2 (C)
FPREM, FPREM1 (see Table 2.3)	Three least significant bits of quotient Q2	Q0	Q1 or O/U #	Reduction 0 = complete 1 = incomplete
FCOM, FCOMP, FCOMPP, FTST, FUCOM, FUCOMP, FUCOMPP, FICOM, FICOMP	Result of comparison (see Table 2.4)		Zero or O/U #	Operand is not comparable (Table 2.4)
FXAM	Operand class (see Table 2.5)		Sign or O/U #	Operand class (Table 2.5)
FCHS, FABS, FXCH, FINCTOP, FDECSTOP, Constant loads, FXTRACT, FLD, FILD, FBLD, FSTP (ext real)	UNDEFINED		Zero or O/U #	UNDEFINED
FIST, FBSTP, FRNDINT, FST, FSTP, FADD, FMUL, FDIV, FDIVR, FSUB, FSUBR, FSCALE, FSQRT, FPATAN, F2XM1, FYLP2X, FYLP2XP1	UNDEFINED		Roundup or O/U #	UNDEFINED
FPTAN, FSIN FCOS, FSINCOS	UNDEFINED		Roundup or O/U #, undefined if C2 = 1	Reduction 0 = complete 1 = incomplete
FLDENW, FRSTOR	Each bit loaded from memory			
FLDCW, FSTENV, FSTCW, FSTSW, FCLEX, FINIT, FSAVE	UNDEFINED			
O/U #	When both IE and SF bits of status word are set, indicating a stack exception, this bit distinguishes between stack overflow (C1 = 1) and underflow (C1 = 0).			
Reduction	If FPREM or FPREM1 produces a remainder that is less than the modulus, reduction is complete. When reduction is incomplete the value at the top of the stack is a partial remainder, which can be used as input to further reduction. For FPTAN, FSIN, FCOS, and FSINCOS, the reduction bit is set if the operand at the top of the stack is too large. In this case the original operand remains at the top of the stack.			
Roundup	When the PE bit of the status word is set, this bit indicates whether the last rounding in the instruction was upward.			
UNDEFINED	Do not rely on finding any specific value in these bits.			

Table 2.3. Condition Code Interpretation after FPREM and FPREM1 Instructions

Condition Code				Interpretation after FPREM and FPREM1	
C2	C3	C1	C0		
1	X	X	X	Incomplete Reduction: further iteration required for complete reduction	
0	Q1	Q0	Q2	Q MOD8	Complete Reduction: C0, C3, C1 contain three least significant bits of quotient
	0	0	0	0	
	0	1	0	1	
	1	0	0	2	
	1	1	0	3	
	0	0	1	4	
	0	1	1	5	
	1	0	1	6	
	1	1	1	7	

Table 2.4. Condition Code Resulting from Comparison

Order	C3	C2	C0
TOP > Operand	0	0	0
TOP < Operand	0	0	1
TOP = Operand	1	0	0
Unordered	1	1	1

Table 2.5. Condition Code Defining Operand Class

C3	C2	C1	C0	Value at TOP
0	0	0	0	+ Unsupported
0	0	0	1	+ NaN
0	0	1	0	- Unsupported
0	0	1	1	- NaN
0	1	0	0	+ Normal
0	1	0	1	+ Infinity
0	1	1	0	- Normal
0	1	1	1	- Infinity
1	0	0	0	+ 0
1	0	0	1	+ Empty
1	0	1	0	- 0
1	0	1	1	- Empty
1	1	0	0	+ Denormal
1	1	1	0	- Denormal

2.3.4 INSTRUCTION AND DATA POINTERS

Because the NPX operates in parallel with the CPU, any errors detected by the NPX may be reported after the CPU has executed the ESC instruction which caused it. To allow identification of the failing numeric instruction, the 80386/80387 contains two pointer registers that supply the address of the failing numeric instruction and the address of its numeric memory operand (if appropriate).

The instruction and data pointers are provided for user-written error handlers. These registers are actually located in the 80386, but appear to be located in the 80387 because they are accessed by the ESC instructions FLDENV, FSTENV, FSAVE, and FRSTOR. (In the 8086/8087 and 80286/80287, these registers are located in the NPX.) Whenever the 80386 decodes a new ESC instruction, it saves

the address of the instruction (including any prefixes that may be present), the address of the operand (if present), and the opcode.

The instruction and data pointers appear in one of four formats depending on the operating mode of the 80386 (protected mode or real-address mode) and depending on the operand-size attribute in effect (32-bit operand or 16-bit operand). When the 80386 is in virtual-8086 mode, the real-address mode formats are used. (See Figures 2.3 through 2.6.) The ESC instructions FLDENV, FSTENV, FSAVE, and FRSTOR are used to transfer these values between the 80386 registers and memory. Note that the value of the data pointer is *undefined* if the prior ESC instruction did not have a memory operand.

32-BIT PROTECTED MODE FORMAT			
31	23	15	7
RESERVED		CONTROL WORD	0
RESERVED		STATUS WORD	4
RESERVED		TAG WORD	8
IP OFFSET			C
RESERVED		CS SELECTOR	10
DATA OPERAND OFFSET			14
RESERVED		OPERAND SELECTOR	18

Figure 2.3. Protected Mode 80387 Instruction and Data Pointer Image in Memory, 32-Bit Format

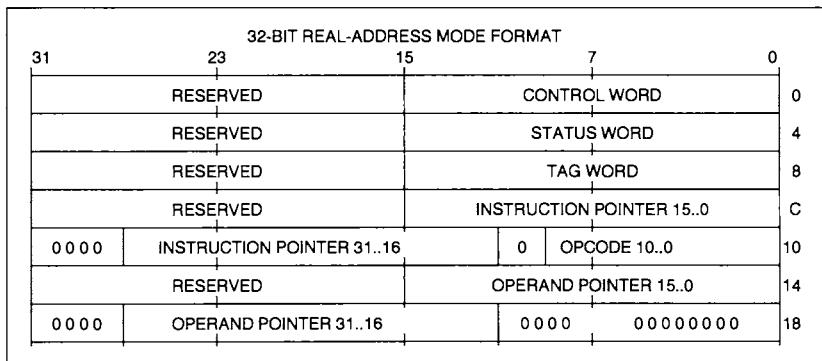


Figure 2.4. Real Mode 80387 Instruction and Data Pointer Image in Memory, 32-Bit Format

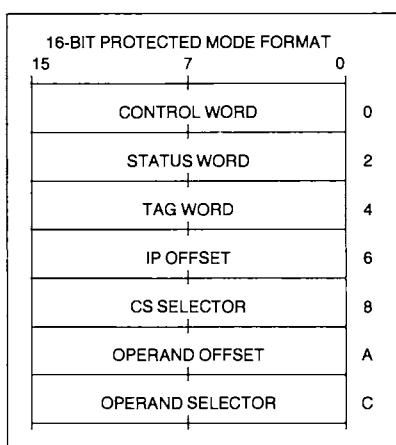


Figure 2.5. Protected Mode 80387
Instruction and Data Pointer
Image in Memory, 16-Bit Format

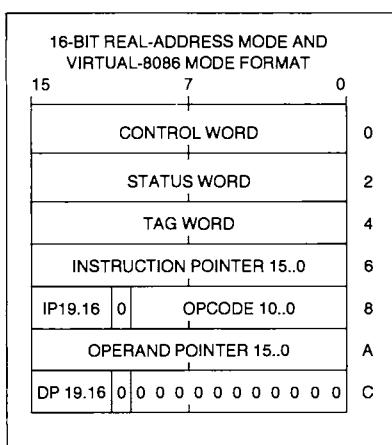


Figure 2.6. Real Mode 80387
Instruction and Data Pointer
Image in Memory, 16-Bit Format

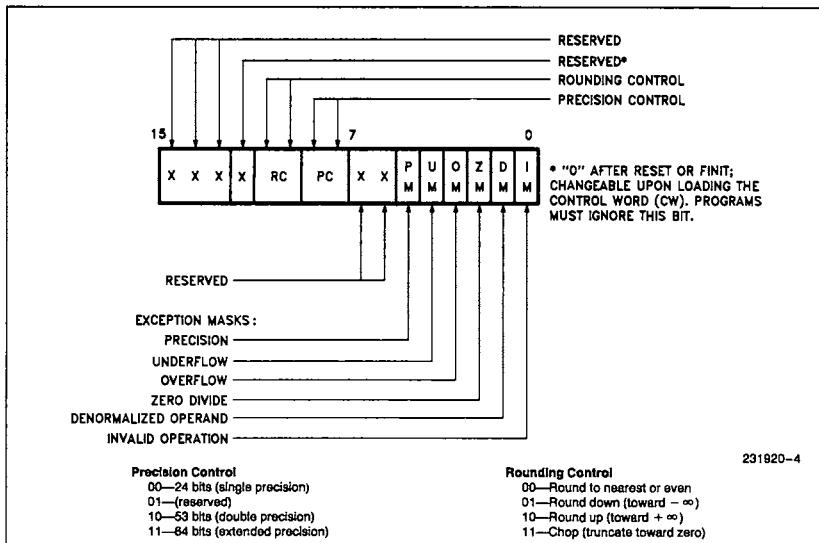


Figure 2.7. 80387 Control Word

2.3.5 CONTROL WORD

The NPX provides several processing options that are selected by loading a control word from memory into the control register. Figure 2.7 shows the format and encoding of fields in the control word.

The low-order byte of this control word configures the 80387 error and exception masking. Bits 5–0 of the control word contain individual masks for each of the six exceptions that the 80387 recognizes.

The high-order byte of the control word configures the 80387 operating mode, including precision and rounding.

- Bit 12 no longer defines infinity control and is a reserved bit. Only affine closure is supported for infinity arithmetic. The bit is initialized to zero after RESET or FINIT and is changeable upon loading the CW. Programs must ignore this bit.
- The rounding control (RC) bits (bits 11–10) provide for directed rounding and true chop, as well as the unbiased round to nearest even mode specified in the IEEE standard. Rounding control

affects only those instructions that perform rounding at the end of the operation (and thus can generate a precision exception); namely, FST, FSTP, FIST, all arithmetic instructions (except FPREM, FPREM1, FXTRACT, FABS, and FCHS), and all transcendental instructions.

- The precision control (PC) bits (bits 9–8) can be used to set the 80387 internal operating precision of the significand at less than the default of 64 bits (extended precision). This can be useful in providing compatibility with early generation arithmetic processors of smaller precision. PC affects only the instructions ADD, SUB, DIV, MUL, and SQRT. For all other instructions, either the precision is determined by the opcode or extended precision is used.

2.4 Interrupt Description

Several interrupts of the 80386 are used to report exceptional conditions while executing numeric programs in either real or protected mode. Table 2.6 shows these interrupts and their causes.



Table 2.6. 80388 Interrupt Vectors Reserved for NPX

Interrupt Number	Cause of Interrupt
7	An ESC instruction was encountered when EM or TS of 80386 control register zero (CR0) was set. EM = 1 indicates that software emulation of the instruction is required. When TS is set, either an ESC or WAIT instruction causes interrupt 7. This indicates that the current NPX context may not belong to the current task.
9	An operand of a coprocessor instruction wrapped around an addressing limit (0FFFFH for small segments, 0FFFFFFFH for big segments, zero for expand-down segments) and spanned inaccessible addresses ^a . The failing numerics instruction is not restartable. The address of the failing numerics instruction and data operand may be lost; an FSTENV does not return reliable addresses. As with the 80286/80287, the segment overrun exception should be handled by executing an FNINIT instruction (i.e. an FINIT without a preceding WAIT). The return address on the stack does not necessarily point to the failing instruction nor to the following instruction. The interrupt can be avoided by never allowing numeric data to start within 108 bytes of the end of a segment.
13	The first word or doubleword of a numeric operand is not entirely within the limit of its segment. The return address pushed onto the stack of the exception handler points at the ESC instruction that caused the exception, including any prefixes. The 80387 has not executed this instruction; the instruction pointer and data pointer register refer to a previous, correctly executed instruction.
16	The previous numerics instruction caused an unmasked exception. The address of the faulty instruction and the address of its operand are stored in the instruction pointer and data pointer registers. Only ESC and WAIT instructions can cause this interrupt. The 80386 return address pushed onto the stack of the exception handler points to a WAIT or ESC instruction (including prefixes). This instruction can be restarted after clearing the exception condition in the NPX. FNINIT, FNCLEX, FNSTSW, FNSTENV, and FNSAVE cannot cause this interrupt.

a. An operand may wrap around an addressing limit when the segment limit is near an addressing limit and the operand is near the largest valid address in the segment. Because of the wrap-around, the beginning and ending addresses of such an operand will be at opposite ends of the segment. There are two ways that such an operand may also span inaccessible addresses: 1) if the segment limit is not equal to the addressing limit (e.g. addressing limit is FFFFH and segment limit is FFFDH) the operand will span addresses that are not within the segment (e.g. an 8-byte operand that starts at valid offset FFFC will span addresses FFFC-FFFF and 0000-0003; however addresses FFFE and FFFF are not valid, because they exceed the limit); 2) if the operand begins and ends in present and accessible pages but intermediate bytes of the operand fall in a not-present page or a page to which the procedure does not have access rights.

2.5 Exception Handling

The 80387 detects six different exception conditions that can occur during instruction execution. Table 2.7 lists the exception conditions in order of precedence, showing for each the cause and the default action taken by the 80387 if the exception is masked by its corresponding mask bit in the control word.

Any exception that is not masked by the control word sets the corresponding exception flag of the status word, sets the ES bit of the status word, and asserts the ERROR# signal. When the CPU attempts to execute another ESC instruction or WAIT, exception 7 occurs. The exception condition must be resolved via an interrupt service routine. The 80386/80387 saves the address of the floating-point instruction that caused the exception and the address of any memory operand required by that instruction.

2.6 Initialization

80387 initialization software must execute an FNINIT instruction (i.e. an FINIT without a preceding WAIT) to clear ERROR#. The FNINIT is not required for the 80287, though Intel documentation recommends its use (refer to the Numerics Supplement to the *iAPX 286 Programmer's Reference Manual*). After a hardware RESET, the ERROR# output is asserted to indicate that an 80387 is present. To accomplish this, the IE and ES bits of the status word are set, and the IM bit in the control word is reset. After FNINIT, the status word and the control word have the same values as in an 80287 after RESET.

2.7 8087 and 80287 Compatibility

This section summarizes the differences between the 80387 and the 80287. Any migration from the 8087 directly to the 80387 must also take into account the differences between the 8087 and the 80287 as listed in Appendix A.

Many changes have been designed into the 80387 to directly support the IEEE standard in hardware. These changes result in increased performance by eliminating the need for software that supports the standard.

2.7.1 GENERAL DIFFERENCES

The 80387 supports only affine closure for infinity arithmetic, not projective closure. Bit 12 of the Control Word (CW) no longer defines infinity control. It is a reserved bit; but it is initialized to zero after RESET or FINIT and is changeable upon loading the CW. Programs must ignore this bit.

Operands for FSCALE and FPATAN are no longer restricted in range (except for $\pm \infty$); F2XM1 and FPATAN accept a wider range of operands.

The results of transcendental operations may be slightly different from those computed by 80287.

In the case of FPATAN, the 80387 supplies a true tangent result in ST(1), and (always) a floating point 1 in ST.

Rounding control is in effect for FLD *constant*.

Software cannot change entries of the tag word to values (other than empty) that do not reflect the actual register contents.

After reset, FINIT, and incomplete FPREM, the 80387 resets to zero the condition code bits C₃-C₀ of the status word.

In conformance with the IEEE standard, the 80387 does not support the special data formats: pseudozero, pseudo-NaN, pseudoinfinity, and unnormal.

Table 2.7. Exceptions

Exception	Cause	Default Action (if exception is masked)
Invalid Operation	Operation on a signaling NaN, unsupported format, indeterminate form ($0^* \infty$, $0/0$, $(+\infty) + (-\infty)$, etc.), or stack overflow/underflow (SF is also set).	Result is a quiet NaN, integer indefinite, or BCD indefinite
Denormalized Operand	At least one of the operands is denormalized, i.e. it has the smallest exponent but a nonzero significand.	Normal processing continues
Zero Divisor	The divisor is zero while the dividend is a noninfinite, nonzero number.	Result is ∞
Overflow	The result is too large in magnitude to fit in the specified format.	Result is largest finite value or ∞
Underflow	The true result is nonzero but too small to be represented in the specified format, and, if underflow exception is masked, denormalization causes loss of accuracy.	Result is denormalized or zero
Inexact Result (Precision)	The true result is not exactly representable in the specified format (e.g. 1/3); the result is rounded according to the rounding mode.	Normal processing continues

2.7.2 EXCEPTIONS

When the overflow or underflow exception is masked, one difference from the 80287 is in rounding when overflow or underflow occurs. The 80387 produces results that are consistent with the rounding mode. The other difference is that the 80387 sets its underflow flag only if there is also a loss of accuracy during denormalization.

A number of differences exist due to changes in the IEEE standard and to functional improvements to the architecture of the 80387:

1. Fewer invalid-operation exceptions due to denormal operands, because the instructions FSQRT, FDIV, FPREM and conversions to BCD or to integer normalize denormal operands before proceeding.
2. The FSQRT, FBSTP, and FPREM instructions may cause underflow, because they support denormal operands.
3. The denormal exception can occur during the transcendental instructions and the FXTRACT instruction.
4. The denormal exception no longer takes precedence over all other exceptions.
5. When the operand is zero, the FXTRACT instruction reports a zero-divide exception and leaves $- \infty$ in ST(1).
6. The status word has a new bit (SF) that signals when invalid-operation exceptions are due to stack underflow or overflow.
7. FLD *extended precision* no longer reports denormal exceptions, because the instruction is not numeric.
8. FLD *single/double precision* when the operand is denormal converts the number to extended precision and signals the denormalized operand exception. When loading a signaling NaN, FLD *single/double precision* signals an invalid-operation exception.
9. The 80387 only generates quiet NaNs (as on the 80287); however, the 80387 distinguishes between quiet NaNs and signaling NaNs. Signaling NaNs trigger exceptions when they are used as operands; quiet NaNs do not (except for FCOM, FIST, and FBSTP which also raise IE for quiet NaNs).

3.0 HARDWARE INTERFACE

In the following description of hardware interface, the # symbol at the end of a signal name indicates that the active or asserted state occurs when the

signal is at a low voltage. When no # is present after the signal name, the signal is asserted when at the high voltage level.

3.1 Signal Description

In the following signal descriptions, the 80387 pins are grouped by function as follows:

1. Execution control—386CLK2, 387CLK2, CKM, RESETIN
2. NPX handshake—PEREQ, BUSY#, ERROR#
3. Bus interface pins—D31-D0, W/R#, ADS#, READY#, READYO#
4. Chip/Port Select—STEN, NPS1#, NPS2, CMDO#
5. Power supplies—V_{CC}, V_{SS}

Table 3.1 lists every pin by its identifier, gives a brief description of its function, and lists some of its characteristics. All output signals are tristate; they leave floating state only when STEN is active. The output buffers of the bidirectional data pins D31-D0 are also tristate; they leave floating state only in read cycles when the 80387 is selected (i.e. when STEN, NPS1#, and NPS2 are all active).

Figure 3.1 and Table 3.2 together show the location of every pin in the pin grid array.

3.1.1 80386 CLOCK 2 (386CLK2)

This input uses the 80386 CLK2 signal to time the bus control logic. Several other 80387 signals are referenced to the rising edge of this signal. When CKM = 1 (synchronous mode) this pin also clocks the data interface and control unit and the floating-point unit of the 80387. This pin requires MOSI-level input. The signal on this pin is divided by two to produce the internal clock signal CLK.

3.1.2 80387 CLOCK 2 (387CLK2)

When CKM = 0 (asynchronous mode) this pin provides the clock for the data interface and control unit and the floating-point unit of the 80387. In this case, the ratio of the frequency of 387CLK2 to the frequency of 386CLK2 must lie within the range 10:16 to 16:10. When CKM = 1 (synchronous mode) this pin is ignored; 386CLK2 is used instead for the data interface and control unit and the floating-point unit. This pin requires TTL-level input.



80387

ADVANCE INFORMATION

Table 3.1. 80387 Pin Summary

Pin Name	Function	Active State	Input/Output	Referenced To
386CLK2 387CLK2 CKM RESETIN	80386 ClocK 2 80387 ClocK 2 80387 CLocking Mode System reset	High	I I I	386CLK2
PEREQ	Processor Extension REQuest	High	O	386CLK2/STEN
BUSY # ERROR #	Busy status Error status	Low Low	O O	386CLK2/STEN 387CLK2/STEN
D31-D0 W/R # ADS # READY # READYO #	Data pins Write/Read bus cycle ADdress Strobe Bus ready input Ready output	High Hi/Lo Low Low Low	I/O I I I O	386CLK2 386CLK2 386CLK2 386CLK2 386CLK2/STEN
STEN NPS1 # NPS2 CMD0 #	STatus ENable NPX select #1 NPX select #2 CoMmanD	High Low High Low	I I I I	386CLK2 386CLK2 386CLK2 386CLK2
VCC VSS			I	

NOTE:

STEN is referenced to only when getting the output pins into or out of tristate mode.

Table 3.2. 80387 Pin Cross-Reference

A2	—	D9	C11	—	VSS	J10	—	Vss
A3	—	D11	D1	—	D5	J11	—	CKM
A4	—	D12	D2	—	D4	K1	—	PEREQ
A5	—	D14	D10	—	D24	K2	—	BUSY #
A6	—	Vcc	D11	—	D25	K3	—	Tie High
A7	—	D16	E1	—	Vcc	K5	—	W/R #
A8	—	D18	E2	—	Vss	K5	—	Vcc
A9	—	Vcc	E10	—	D26	K6	—	NPS2
A10	—	D21	E11	—	D27	K7	—	ADS #
B1	—	D8	F1	—	Vcc	K8	—	READY #
B2	—	Vss	F2	—	Vss	K9	—	No Connect
B3	—	D10	F10	—	Vcc	K10	—	386CLK2
B4	—	Vcc	F11	—	Vss	K11	—	387CLK2
B5	—	D13	G1	—	D3	L2	—	ERROR #
B6	—	D15	G2	—	D2	L3	—	READYO #
B7	—	Vss	G10	—	D28	L5	—	STEN
B8	—	D17	G11	—	D29	L5	—	Vss
B9	—	D19	H1	—	D1	L6	—	NPS1 #
B10	—	D20	H2	—	D0	L7	—	Vcc
B11	—	D22	H10	—	D30	L8	—	CMD0 #
C1	—	D7	H11	—	D31	L9	—	Tie High
C2	—	D6	J1	—	Vss	L10	—	RESETIN
C10	—	D23	J2	—	Vcc			

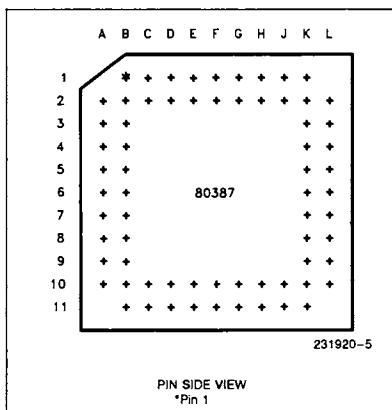


Figure 3.1. 80387 Pin Configuration

3.1.3 80387 CLOCKING MODE (CKM)

This pin is a strapping option. When it is strapped to V_{CC}, the 80387 operates in synchronous mode; when strapped to V_{SS}, the 80387 operates in asynchronous mode. These modes relate to clocking of the data interface and control unit and the floating-point unit only; the bus control logic always operates synchronously with respect to the 80386.

3.1.4 SYSTEM RESET (RESETIN)

A LOW to HIGH transition on this pin causes the 80387 to terminate its present activity and to enter a dormant state. RESETIN must remain HIGH for at least 78 387CLK2 periods. The HIGH to LOW transitions of RESETIN must be synchronous with 386CLK2, so that the phase of the internal clock of the bus control logic (which is the 386CLK2 divided by 2) is the same as the phase of the internal clock of the 80386. After RESETIN goes LOW, at least 50 387CLK2 periods must pass before the first NPX instruction is written into the 80387. This pin should be connected to the 80386 RESET pin. Table 3.3 shows the status of other pins after a reset.

Table 3.3. Output Pin Status after Reset

Pin Value	Pin Name
HIGH	READYO #, BUSY #
LOW	PEREQ, ERROR #
Tri-State OFF	D31-D0

3.1.5 PROCESSOR EXTENSION REQUEST (PEREQ)

When active, this pin signals to the 80386 CPU that the 80387 is ready for data transfer to/from its data FIFO. When all data is written to or read from the data FIFO, PEREQ is deactivated. This signal always goes inactive before BUSY # goes inactive. This signal is referenced to 386CLK2. It should be connected to the 80386 PREQ input. Refer to Figure 3.7 for the timing relationships between this and the BUSY # and ERROR # pins.

3.1.6 BUSY STATUS (BUSY #)

When active, this pin signals to the 80386 CPU that the 80387 is currently executing an instruction. This signal is referenced to 386CLK2. It should be connected to the 80386 BUSY # pin. Refer to Figure 3.7 for the timing relationships between this and the PREQ and ERROR # pins.

3.1.7 ERROR STATUS (ERROR #)

This pin reflects the ES bits of the status register. When active, it indicates that an unmasked exception has occurred (except that, immediately after a reset, it indicates to the 80386 that an 80387 is present in the system). This signal can be changed to inactive state only by the following instructions (without a preceding WAIT): FNINIT, FNCLEX, FNSTENV, and FNSAVE. This signal is referenced to 387CLK2. It should be connected to the 80386 ERROR # pin. Refer to Figure 3.7 for the timing relationships between this and the PREQ and BUSY # pins.

3.1.8 DATA PINS (D31-D0)

These bidirectional pins are used to transfer data and opcodes between the 80386 and 80387. They are normally connected directly to the corresponding 80386 data pins. HIGH state indicates a value of one. D0 is the least significant data bit. Timings are referenced to 386CLK2.

3.1.9 WRITE/READ BUS CYCLE (W/R #)

This signal indicates to the 80387 whether the 80386 bus cycle in progress is a read or a write cycle. This pin should be connected directly to the 80386 W/R # pin. HIGH indicates a write cycle; LOW, a read cycle. This input is ignored if any of the signals STEN, NPS1#, or NPS2 is inactive. Setup and hold times are referenced to 386CLK2.

3.1.10 ADDRESS STROBE (ADS#)

This input, in conjunction with the READY# input indicates when the 80387 bus-control logic may sample W/R# and the chip-select signals. Setup and hold times are referenced to 386CLK2. This pin should be connected to the 80386 ADS# pin.

3.1.11 BUS READY INPUT (READY#)

This input indicates to the 80387 when an 80386 bus cycle is to be terminated. It is used by the bus-control logic to trace bus activities. Bus cycles can be extended indefinitely until terminated by READY#. This input should be connected to the same signal that drives the 80386 READ# input. Setup and hold times are referenced to 386CLK2.

3.1.12 READY OUTPUT (READYO#)

This pin is activated at such a time that write cycles are terminated after two clocks and read cycles after three clocks. In configurations where no extra wait states are required, it can be used to directly drive the 80386 READY# input. Refer to section 3.4 "Bus Operation" for details. This pin is activated only during bus cycles that select the 80387. This signal is referenced to 386CLK2.

3.1.13 STATUS ENABLE (STEN)

This pin serves as a chip select for the 80387. When inactive, this pin forces BUSY#, PEREQ, ERROR#, and READYO# outputs into floating state. D31-D0 are normally floating and leave floating state only if STEN is active and additional conditions are met. STEN also causes the chip to recognize its other chip-select inputs. STEN makes it easier to do on-board testing (using the overdrive method) of other chips in systems containing the 80387. STEN should be pulled up with a resistor so that it can be pulled down when testing. In boards that do not use on-board testing, STEN should be connected to Vcc. Setup and hold times are relative to 386CLK2. Note that STEN must maintain the same setup and hold times as NPS1#, NPS2, and CMD0# (i.e. if STEN changes state during an 80387 bus cycle, it should change state during the same CLK period as the NPS1#, NPS2, and CMD0# signals).

3.1.14 NPX Select #1 (NPS1#)

When active (along with STEN and NPS2) in the first period of an 80386 bus cycle, this signal indicates that the purpose of the bus cycle is to communicate with the 80387. This pin should be connected directly to the 80386 M/IO# pin, so that the 80387 is selected only when the 80386 performs I/O cycles. Setup and hold times are referenced to 386CLK2.

3.1.15 NPX SELECT #2 (NPS2)

When active (along with STEN and NPS1#) in the first period of an 80386 bus cycle, this signal indicates that the purpose of the bus cycle is to communicate with the 80387. This pin should be connected directly to the 80386 A31 pin, so that the 80387 is selected only when the 80386 uses one of the I/O addresses reserved for the 80387 (800000F8 or 800000FC). Setup and hold times are referenced to 386CLK2.

3.1.16 COMMAND (CMD0#)

During a write cycle, this signal indicates whether an opcode (CMD0# active) or data (CMD0# inactive) is being sent to the 80387. During a read cycle, it indicates whether the control or status register (CMD0# active) or a data register (CMD0# inactive) is being read. CMD0# should be connected directly to the A2 output of the 80386. Setup and hold times are referenced to 386CLK2.

3.2 Processor Architecture

As shown by the block diagram on the front page, the NPX is internally divided into three sections: the bus control logic (BCL), the data interface and control unit, and the floating point unit (FPU). The FPU (with the support of the control unit which contains the sequencer and other support units) executes all numerics instructions. The data interface and control unit is responsible for the data flow to and from the FPU and the control registers, for receiving the instructions, decoding them, and sequencing the microinstructions, and for handling some of the administrative instructions. The BCL is responsible for 80386 bus tracking and interface. The BCL is the only unit in the 80387 that must run synchronously with the 80386; the rest of the 80387 can run asynchronously with respect to the 80386.

3.2.1 BUS CONTROL LOGIC

The BCL communicates solely with the CPU using I/O bus cycles. The BCL appears to the CPU as a special peripheral device. It is special in two respects: the CPU initiates I/O automatically when it encounters ESC instructions, and the CPU uses reserved I/O addresses to communicate with the BCL. The BCL does not communicate directly with memory. The CPU performs all memory access, transferring input operands from memory to the 80387 and transferring outputs from the 80387 to memory.

3.2.2 DATA INTERFACE AND CONTROL UNIT

The data interface and control unit latches the data and, subject to BCL control, directs the data to the FIFO or the instruction decoder. The instruction decoder decodes the ESC instructions sent to it by the CPU and generates controls that direct the data flow in the FIFO. It also triggers the microinstruction sequencer that controls execution of each instruction. If the ESC instruction is FINIT, FCLEX, FSTSW, FSTSW AX, or FSTCW, the control executes it independently of the FPU and the sequencer. The data interface and control unit is the one that generates the BUSY#, PEREQ and ERROR# signals that synchronize 80387 activities with the 80386. It also supports the FPU in all operations that it cannot perform alone (e.g. exceptions handling, transcendental operations, etc.).

3.2.3 FLOATING POINT UNIT

The FPU executes all instructions that involve the register stack, including arithmetic, logical, transcen-

dental, constant, and data transfer instructions. The data path in the FPU is 84 bits wide (68 significant bits, 15 exponent bits, and a sign bit) which allows internal operand transfers to be performed at very high speeds.

3.3 System Configuration

As an extension to the 80386, the 80387 can be connected to the CPU as shown by Figure 3.2. A dedicated communication protocol makes possible high-speed transfer of opcodes and operands between the 80386 and 80387. The 80387 is designed so that no additional components are required for interface with the 80386. The 80387 shares the 32-bit wide local bus of the 80386 and most control pins of the 80387 are connected directly to pins of the 80386.

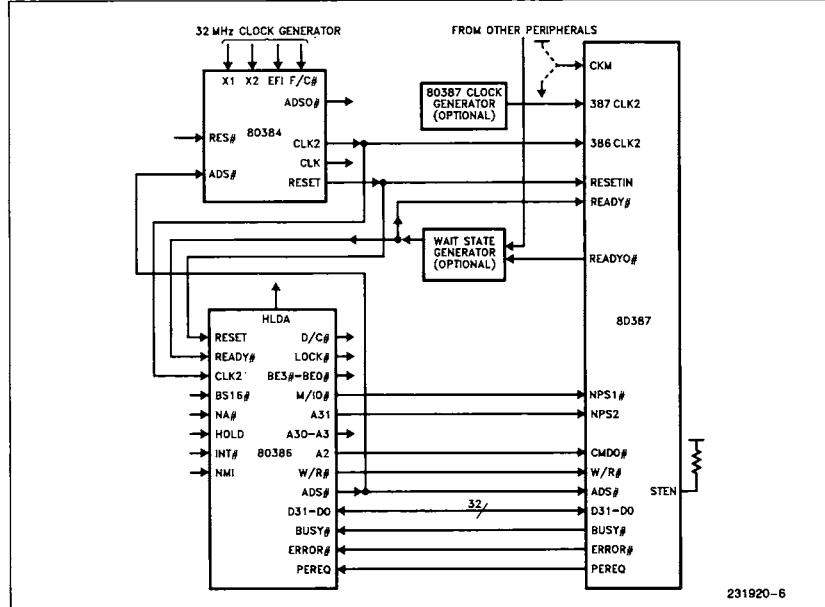


Figure 3.2. 80386/80387 System Configuration

Table 3.4. Bus Cycles Definition

STEN	NPS1 #	NPS2	CMD0 #	W/R #	Bus Cycle Type
0	x	x	x	x	80387 not selected and all outputs in floating state
1	1	x	x	x	80387 not selected
1	x	0	x	x	80387 not selected
1	0	1	0	0	CW or SW read from 80387
1	0	1	0	1	Opcode write to 80387
1	0	1	1	0	Data read from 80387
1	0	1	1	1	Data write to 80387

3.3.1 BUS CYCLE TRACKING

The ADS# and READY# signals allow the 80387 to track the beginning and end of 80386 bus cycles, respectively. When ADS# is asserted at the same time as the 80387 chip-select inputs, the bus cycle is intended for the 80387. To signal the end of a bus cycle for the 80387, READY# may be asserted directly or indirectly by the 80387 or by other bus-control logic. Refer to Table 3.4 for definition of the types of 80387 bus cycles.

3.3.2 80387 ADDRESSING

The NPS1 #, NPS2 and STEN signals allow the NPX to identify which bus cycles are intended for the NPX. The NPX responds only to I/O cycles when bit 31 of the I/O address is set. In other words, the NPX acts as an I/O device in a reserved I/O address space.

Because A₃₁ is used to select the 80387 for data transfers, it is not possible for a program running on the 80386 to address the 80387 with an I/O instruction. Only ESC instructions cause the 80386 to communicate with the 80387. The 80386 BS16# input must be inactive during I/O cycles when A₃₁ is active.

3.3.3 FUNCTION SELECT

The CMD0# and W/R# signals identify the four kinds of bus cycle: control or status register read, data read, opcode write, data write.

3.3.4 CPU/NPX Synchronization

The pin pairs BUSY#, PEREQ, and ERROR# are used for various aspects of synchronization between the CPU and the NPX.

BUSY# is used to synchronize instruction transfer from the 80386 to the 80387. When the 80387 recognizes an ESC instruction, it asserts BUSY#. For most ESC instructions, the 80386 waits for the 80387 to deassert BUSY# before sending the new opcode.

The NPX uses the PEREQ pin of the 80386 CPU to signal that the NPX is ready for data transfer to or from its data FIFO. The NPX does not directly access memory; rather, the 80386 provides memory access services for the NPX. Thus, memory access on behalf of the NPX always obeys the rules applicable to the mode of the 80386, whether the 80386 be in real-address mode or protected mode.

Once the 80386 initiates an 80387 instruction that has operands, the 80386 waits for PEREQ signals that indicate when the 80387 is ready for operand transfer. Once all operands have been transferred (or if the instruction has no operands) the 80386 continues program execution while the 80387 executes the ESC instruction.

In 8086/8087 systems, WAIT instructions may be required to achieve synchronization of both commands and operands. In 80286/80287 and 80386/80387 systems, WAIT instructions are required only for operand synchronization; namely, after NPX stores to memory (except FSTSW and FSTCW) or loads from memory. Used this way, WAIT ensures that the value has already been written or read by the NPX before the CPU reads or changes the value.

Once it has started to execute a numerics instruction and has transferred the operands from the 80386, the 80387 can process the instruction in parallel with and independent of the host CPU. When the NPX detects an exception, it asserts the ERROR# signal, which causes an 80386 interrupt.

3.3.5 SYNCHRONOUS OR ASYNCHRONOUS MODES

The internal logic of the 80387 (the FPU) can either operate directly from the CPU clock (synchronous mode) or from a separate clock (asynchronous mode). The two configurations are distinguished by the CKM pin. In either case, the bus control logic (BCL) of the 80387 is synchronized with the CPU clock. Use of asynchronous mode allows the 80386 and the FPU section of the 80387 to run at different speeds. In this case, the ratio of the frequency of

387CLK2 to the frequency of 386CLK2 must lie within the range 10:16 to 16:10. Use of synchronous mode eliminates one clock generator from the board design.

3.3.6 AUTOMATIC BUS CYCLE TERMINATION

In configurations where no extra wait states are required, READYO# can be used to drive the 80386 READY# input. If this pin is used, it should be connected to the logic that ORs all READY outputs from peripherals on the 80386 bus. READYO# is asserted by the 80387 only during I/O cycles that select the 80387. Refer to section 3.4 "Bus Operation" for details.

3.4 Bus Operation

With respect to the bus interface, the 80387 is fully synchronous with the 80386. Both operate at the same rate, because each generates its internal CLK signal by dividing 386CLK2 by two.

The 80386 initiates a new bus cycle by activating ADS#. The 80387 recognizes a bus cycle, if, during the cycle in which ADS# is activated, STEN, NPS1#, and NPS2 are all activated. Proper operation is achieved if NPS1# is connected to the M/I/O# output of the 80386, and NPS2 to the A31 output. The 80386's A31 output is guaranteed to be inactive in all bus cycles that do not address the 80387 (i.e. I/O cycles to other devices, interrupt acknowledgement, and reserved types of bus cycles). System logic must not signal a 16-bit bus cycle via the 80386 BS16# input during I/O cycles when A31 is active.

During the CLK period in which ADS# is activated, the 80387 also examines the W/R# input signal to determine whether the cycle is a read or a write cycle and examines the CMD0# input to determine whether an opcode, operand, or control/status register transfer is to occur.

The 80387 supports both pipelined and nonpipelined bus cycles. A nonpipelined cycle is one for which the 80386 asserts ADS# when no other 80387 bus cycle is in progress. A pipelined bus cycle is one for which the 80386 asserts ADS# and provides valid next-address and control signals as soon as in the second CLK period after the ADS# assertion for the previous 80386 bus cycle. Pipelining increases the availability of the bus by at least one CLK period. The 80387 supports pipelined bus cycles in order to optimize address pipelining by the 80386 for memory cycles.

Bus operation is described in terms of an abstract *state machine*. Figure 3.3 illustrates the states and state transitions for 80387 bus cycles:

- T_1 is the idle state. This is the state of the bus logic after RESET, the state to which bus logic returns after every nonpipelined bus cycle, and the state to which bus logic returns after a series of pipelined cycles.
- T_{RS} is the READY# sensitive state. Different types of bus cycle may require a minimum of one or two successive T_{RS} states. The bus logic remains in T_{RS} state until READY# is sensed, at which point the bus cycle terminates. Any number of wait states may be implemented by delaying READY#, thereby causing additional successive T_{RS} states.
- T_p is the first state for every pipelined bus cycle.

The READYO# output of the 80387 indicates when a bus cycle for the 80387 may be terminated if no extra wait states are required. For all write cycles (except those for the instructions FLDENV and FRSTOR), READYO# is always asserted in the first T_{RS} state, regardless of the number of wait states. For all read cycles and write cycles for FLDENV and FRSTOR, READYO# is always asserted in the second T_{RS} state, regardless of the number of wait states. These rules apply to both pipelined and nonpipelined cycles. Systems designers may use READYO# in one of three ways:

1. Leave it disconnected and use external logic to generate READY# signals. When choosing this option, 80387 requirements for wait states in read cycles and write cycles of FLDENV and FRSTOR must be obeyed.
2. Connect it (directly or through logic that ORs READY signals from other devices) to the READY# inputs of the 80386 and 80387.
3. Use it as one input to a wait-state generator.

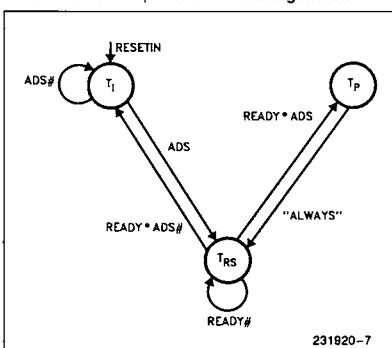


Figure 3.3. Bus State Diagram

The following sections illustrate different types of 80387 bus cycles.

Because different instructions have different amounts of overhead before, between, and after operand transfer cycles, it is not possible to represent in a few diagrams all of the combinations of successive operand transfer cycles. The following bus-cycle diagrams show memory cycles between 80387 operand-transfer cycles. Note however that, during the instructions FLDENV, FSTENV, FSAVE, and FRSTOR, some consecutive accesses to the NPX do not have intervening memory accesses. For the timing relationship between operand transfer cycles and opcode write or other overhead activities, see Figure 3.7.

3.4.1 NONPIPELINED BUS CYCLES

Figure 3.4 illustrates bus activity for consecutive nonpipelined bus cycles.

3.4.1.1 Write Cycle

At the second clock of the bus cycle, the 80387 enters the T_{RS} (READY#-sensitive) state. During this state, the 80387 samples the READY# input and stays in this state as long as READY# is inactive.

In write cycles, the 80387 drives the READYO# signal for one CLK period beginning with the second CLK of the bus cycle; therefore, the fastest write cycle takes two CLK cycles (see cycle 2 of Figure 3.4). For the instructions FLDENV and FRSTOR, however, the 80387 forces a wait state by delaying the activation of READYO# to the second T_{RS} cycle (not shown in Figure 3.4).

When READY# is asserted the 80387 returns to the idle state, in which ADS# could be asserted again by the 80386 for the next cycle.

3.4.1.2 Read Cycle

At the second clock of the bus cycle, the 80387 enters the T_{RS} state. See Figure 3.4. In this state, the 80387 samples the READY# input and stays in this state as long as READY# is inactive.

At the rising edge of CLK in the second clock period of the cycle, the 80387 starts to drive the D31-D0 outputs and continues to drive them as long as it stays in T_{RS} state.

In read cycles that address the 80387, at least one wait state must be inserted to insure that the 80386 latches the correct data. Since the 80387 starts driving the system data bus only at the rising edge of CLK in the second clock period of the bus cycle, not enough time is left for the data signals to propagate and be latched by the 80386 at the falling edge of the same clock period. The 80387 drives the READYO# signal for one CLK period in the third CLK of the bus cycle. Therefore, if the READYO# output is used to drive the 80386 READY# input, one wait state is inserted automatically.

Because one wait state is required for 80387 reads, the minimum is three CLK cycles per read, as cycle 3 of Figure 3.4 shows.

When READY# is asserted the 80387 returns to the idle state, in which ADS# could be asserted again by the 80386 for the next cycle. The transition from T_{RS} state to idle state causes the 80387 to put the tristate D31-D0 outputs into the floating state, allowing another device to drive the system data bus.

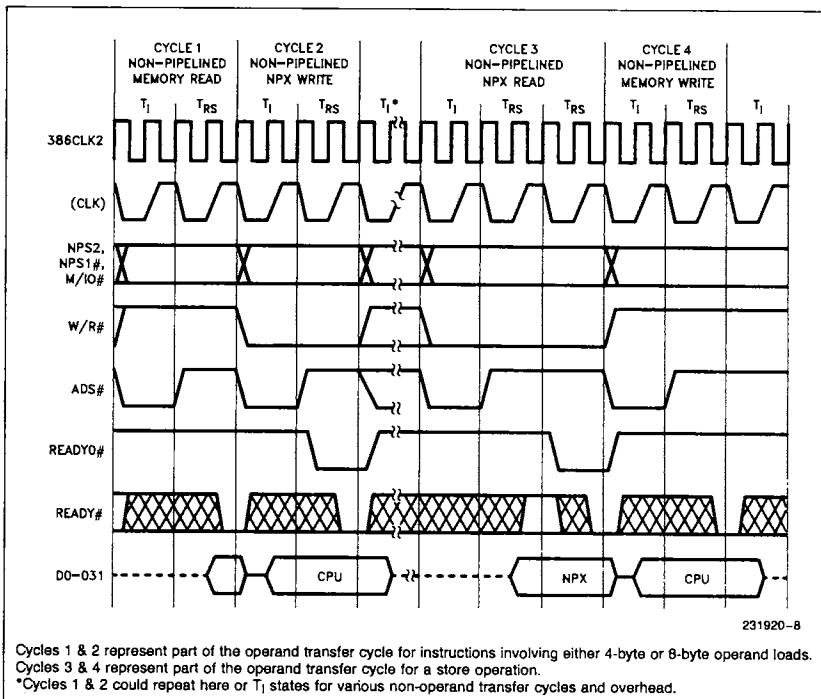


Figure 3.4. Nonpipelined Read and Write Cycles

3.4.2 PIPELINED BUS CYCLES

Because all the activities of the 80387 bus interface occur either during the T_{RS} state or during the transitions to or from that state, the only difference between a pipelined and a nonpipelined cycle is the manner of changing from one state to another. The exact activities in each state are detailed in the previous section "Nonpipelined Bus Cycles".

When the 80386 asserts ADS# before the end of a bus cycle, both ADS# and READY# are active during a T_{RS} state. This condition causes the 80387 to change to a different state named T_P. The 80387 activities in the transition from a T_{RS} state to a T_P state are exactly the same as those in the transition from a T_{RS} state to a T_I state in nonpipelined cycles.

T_P state is metastable; therefore, one clock period later the 80387 returns to T_{RS} state. In consecutive pipelined cycles, the 80387 bus logic uses only T_{RS} and T_P states.

Figure 3.5 shows the fastest transition into and out of the pipelined bus cycles. Cycle 1 in this figure represents a nonpipelined cycle. (Nonpipelined write cycles with only one T_{RS} state (i.e. no wait states) are always followed by another nonpipelined cycle, because READY# is asserted before the earliest possible assertion of ADS# for the next cycle.)

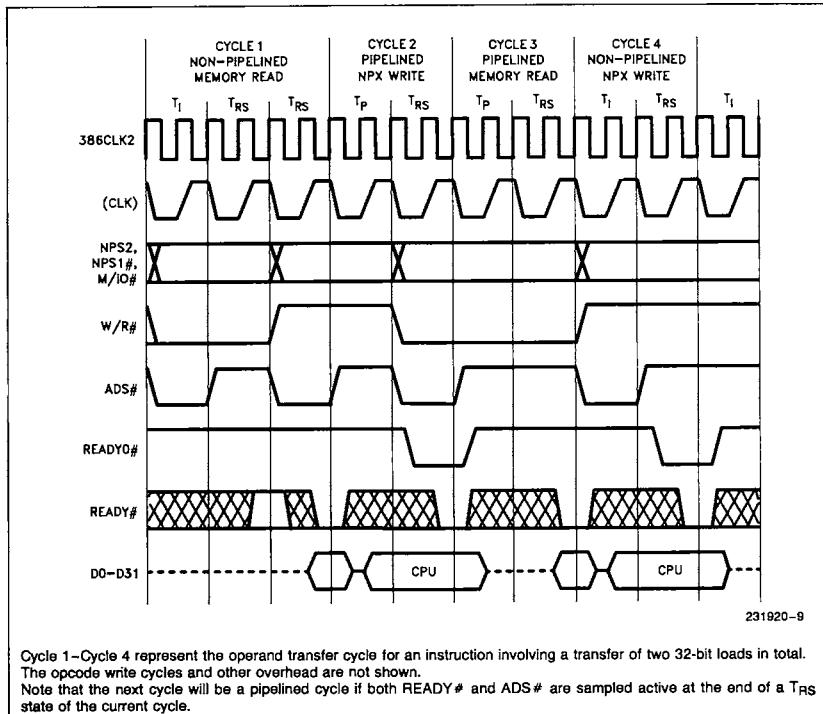
Figure 3.6 shows the pipelined write and read cycles with one additional T_{RS} states beyond the minimum required. To delay the assertion of READY# requires external logic.

3.4.3 BUS CYCLES OF MIXED TYPE

When the 80387 bus logic is in the T_{RS} state, it distinguishes between nonpipelined and pipelined cycles according to the behavior of $ADS\#$ and $READY\#$. In a nonpipelined cycle, only $READY\#$ is activated, and the transition is from T_{RS} to idle state. In a pipelined cycle, both $READY\#$ and $ADS\#$ are active and the transition is first from T_{RS} state to T_P state then, after one clock period, back to T_{RS} state.

3.4.4 BUSY# AND PEREQ TIMING RELATIONSHIP

Figure 3.7 shows the activation of $BUSY\#$ at the beginning of instruction execution and its deactivation after execution of the instruction is complete. $PEREQ$ is activated in this interval. If $ERROR\#$ (not shown in the diagram) is ever asserted, it would occur at least six 386CLK2 periods after the deactivation of $PEREQ$ and at least six 386CLK2 periods before the deactivation of $BUSY\#$. Figure 3.7 shows also that $STEN$ is activated at the beginning of a bus cycle.



Cycle 1–Cycle 4 represent the operand transfer cycle for an instruction involving a transfer of two 32-bit loads in total. The opcode write cycles and other overhead are not shown.
Note that the next cycle will be a pipelined cycle if both $READY\#$ and $ADS\#$ are sampled active at the end of a T_{RS} state of the current cycle.

Figure 3.5. Fastest Transitions to and from Pipelined Cycles

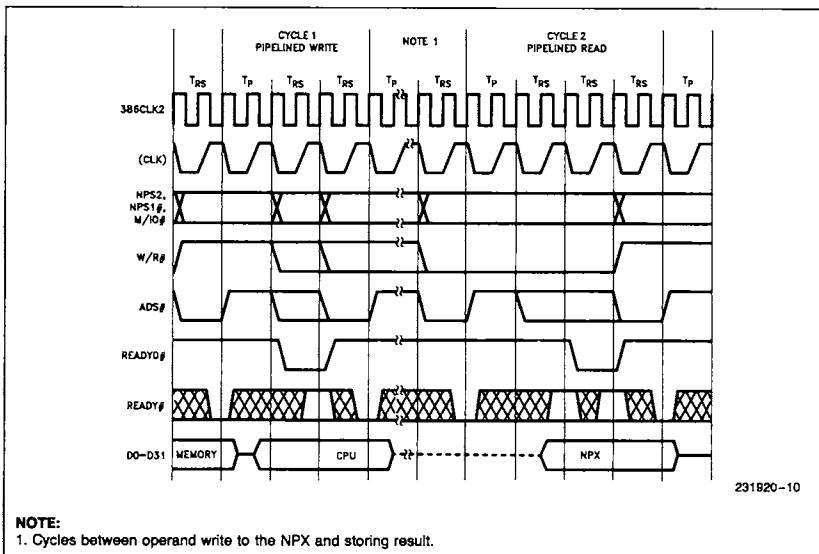


Figure 3.6. Pipelined Cycles with Wait States

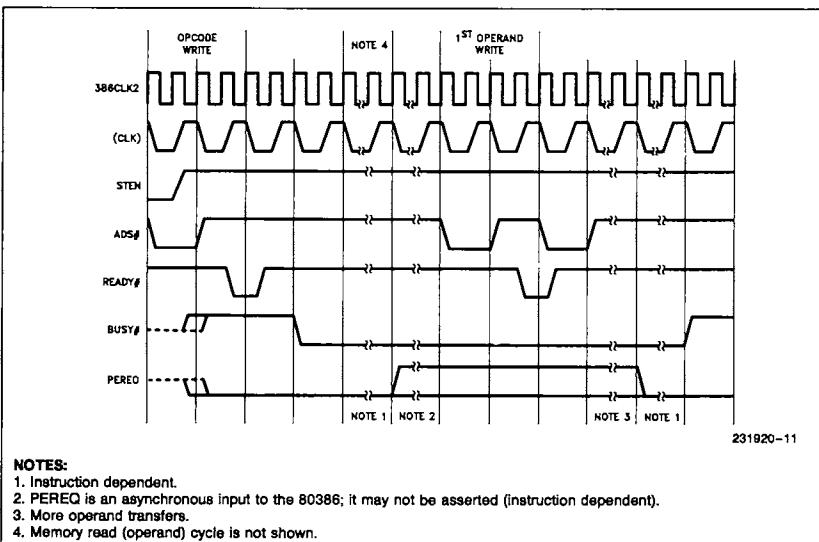
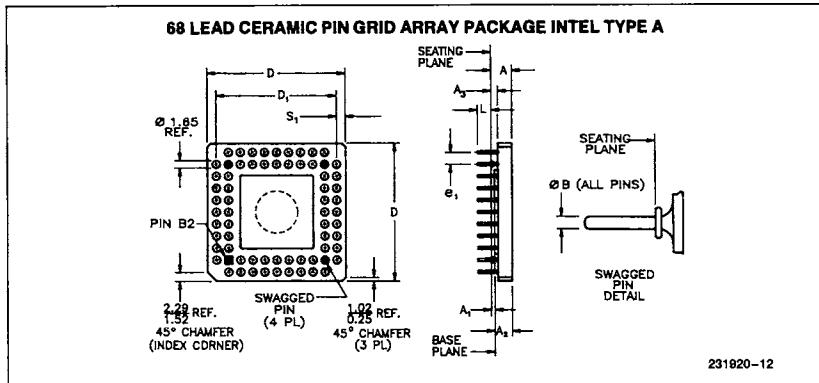


Figure 3.7. STEN, BUSY# and PEREQ Timing Relationship

4.0 MECHANICAL DATA



Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
A	3.56	4.57		0.140	0.180	
A ₁	0.76	1.27	Solid Lid	0.030	0.050	Solid Lid
A ₁		0.41	EPROM Lid		0.016	EPROM Lid
A ₂	2.72	3.43	Solid Lid	0.107	0.135	Solid Lid
A ₂	3.43	4.32	EPROM Lid	0.135	0.170	EPROM Lid
A ₃	1.14	1.40		0.045	0.055	
B	0.43	0.51		0.017	0.020	
D	28.83	29.59		1.135	1.165	
D ₁	25.27	25.53		0.995	1.005	
e ₁	2.29	2.79		0.090	0.110	
L	2.29	3.30		0.090	0.130	
N	68			68		
S ₁	1.27	2.54		0.050	0.100	
ISSUE	IWS REV 7 3/26/86					

Figure 4.1. Package Description

5.0 ELECTRICAL DATA

5.1 Absolute Maximum Ratings*

Case Temperature T_C

Under Bias -65°C to $+110^{\circ}\text{C}$

Storage Temperature -65°C to $+150^{\circ}\text{C}$

Voltage on Any Pin with

Respect to Ground -0.5 to $V_{CC} + 0.5\text{V}$

Power Dissipation 1.5W

*Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

NOTICE: Specifications contained within the following tables are subject to change.

5.2 D.C. Characteristics

Table 5.1. DC Specifications $T_C = 0^{\circ}$ to 85°C , $V_{CC} = 5\text{V} \pm 5\%$

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input LO Voltage	-0.3	+0.8	V	(Note 1)
V_{IH}	Input HI Voltage	2.0	$V_{CC} - 0.3$	V	(Note 1)
V_{CL}	386CLK2 Input LO Voltage	-0.3	+0.8	V	
V_{CH}	386CLK2 Input HI Voltage	3.7	$V_{CC} - 0.3$	V	
V_{OL}	Output LO Voltage	2.0	+0.8	V	(Note 2)
V_{OH}	Output HI Voltage	2.0	$V_{CC} - 0.5$	V	(Note 3)
I_{CC}	Power Supply Current	150 (typical)	250	mA	$387\text{CLK2} = 32\text{ MHz}^{(4)}$
I_{CC}	Power Supply Current	190 (typical)	310	mA	$387\text{CLK2} = 40\text{ MHz}^{(4)}$
I_{LI}	Input Leakage Current		± 15	μA	$0\text{V} \leq V_{IN} \leq V_{CC}$
I_{LO}	I/O Leakage Current		± 15	μA	$0.45\text{V} \leq V_O \leq V_{CC}$
C_{IN}	Input Capacitance		10	pF	$f_c = 1\text{ MHz}$
C_O	I/O Output Capacitance		12	pF	$f_c = 1\text{ MHz}$
C_{CLK}	Clock Capacitance		20	pF	$f_c = 1\text{ MHz}$

NOTES:

- This parameter is for all inputs, including 387CLK2 but excluding 386CLK2.
- This parameter is measured at I_{OL} as follows:
 $\text{data} = 4.0\text{ mA}$
 $\text{READYO\#} = 2.5\text{ mA}$
 $\text{ERROR\#, BUSY\#, PEREQ} = 2.5\text{ mA}$
- This parameter is measured at I_{OH} as follows:
 $\text{data} = 1.0\text{ mA}$
 $\text{READYO\#} = 0.6\text{ mA}$
 $\text{ERROR\#, BUSY\#, PEREQ} = 0.6\text{ mA}$
- I_{CC} is measured at steady state, maximum capacitive loading on the outputs, and worst-case DC level at the inputs; 386CLK2 at the same frequency as 387CLK2.

5.3 A.C. Characteristics

Table 5.2. Timing Requirements $T_C = 0^\circ \text{ to } 85^\circ\text{C}$, $V_{CC} = 5V \pm 5\%$

Pin	Symbol	Parameter	16 MHz		20 MHz		Test Conditions	Figure Reference
			Min (ns)	Max (ns)	Min (ns)	Max (ns)		
386CLK2	t1	period	31.25	125	25	125	2.0V	5.1
386CLK2	t2a	high time	9		8		2.0V	
386CLK2	t2b	high time	5		5		3.7V	
386CLK2	t3a	low time	9		8		2.0	
386CLK2	t3b	low time	7		6		0.8V	
386CLK2	t4	fall time		8		8	3.7V to 0.8V	
386CLK2	t5	rise time		8		8	0.8V to 3.7V	
387CLK2	t1	period	31.25	125	25	125	1.4V	5.1
387CLK2	t2a	high time	9		8		1.4V	
387CLK2	t2b	high time	5		5		2.0V	
387CLK2	t3a	low time	9		8		3.7V	
387CLK2	t3b	low time	7		6		0.8V	
387CLK2	t4	fall time		8		8	2.0V to 0.8V	
387CLK2	t5	rise time		8		8	0.8V to 3.7V	
386CL2/387CLK2		ratio	10/16	16/10	10/16	16/10		
READYO#	t7	out delay	3	35			$C_L = 75 \text{ pF}$	5.2
READYO#	t7	out delay	3	32			$C_L = 25 \text{ pF}$	
PEREQ	t7	out delay	3	3			$C_L = 75 \text{ pF}$	
BUSY#	t7	out delay	3	35			$C_L = 75 \text{ pF}$	
ERROR#	t7	out delay	3	35			$C_L = 75 \text{ pF}$	
D31-D0	t8	out delay	0	55			$C_L = 120 \text{ pF}$	5.3
D31-D0	t10	setup time	10		10			
D31-D0	t11	hold time	0		0			
D31-D0	t12*	float time	5		0			
PEREQ	t13*	float time	1	60			$C_L = 75 \text{ pF}$	5.5
BUSY#	t13*	float time	1	60			$C_L = 75 \text{ pF}$	
ERROR#	t13*	float time	1	60			$C_L = 75 \text{ pF}$	
READYO#	t13*	float time	1	60			$C_L = 75 \text{ pF}$	
ADS#	t14	setup time	25		20			5.3
ADS#	t14	hold time	25		4			
W/R#	t14	setup time	25		20			
W/R#	t15	hold time	4		4			
READY#	t16	setup time	20		12			5.3
READY#	t17	hold time	3		3			
CMD0#	t16	setup time	20		18			
CMD0#	t17	hold time	1		1			
NPS1#, NPS2	t16	setup time	20		18			
NPS1#, NPS2	t17	hold time	1		1			
STEN	t16	setup time	20		20			
STEN	t17	hold time	1		1			
RESETIN	t18	setup time	12		10			5.4
RESETIN	t19	hold time	3		3			

*Float condition occurs when maximum output current becomes less than I_{LO} in magnitude. Float delay is not tested.

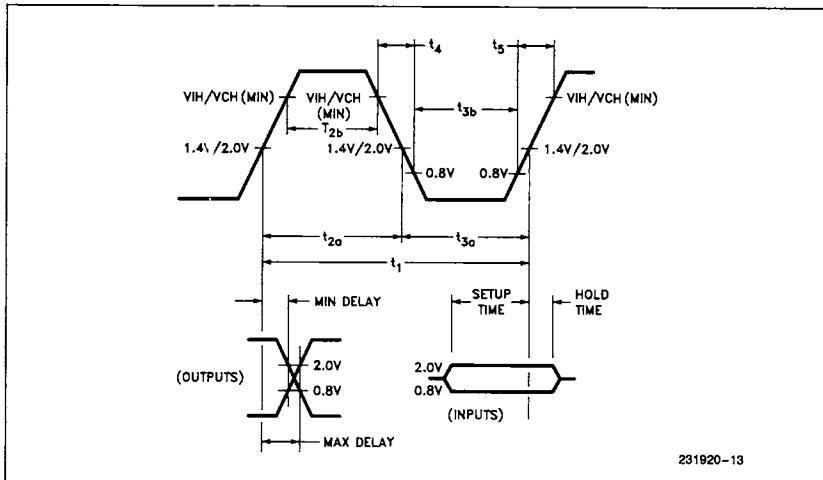


Figure 5.1. 386CLK2/387CLK2 Waveform

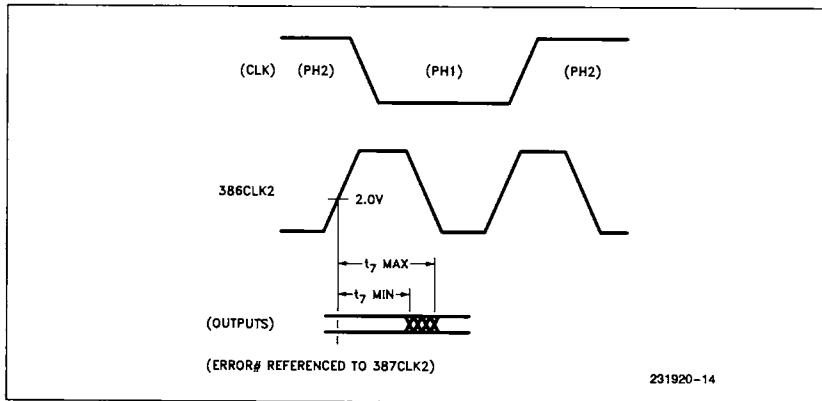


Figure 5.2. Output Signals

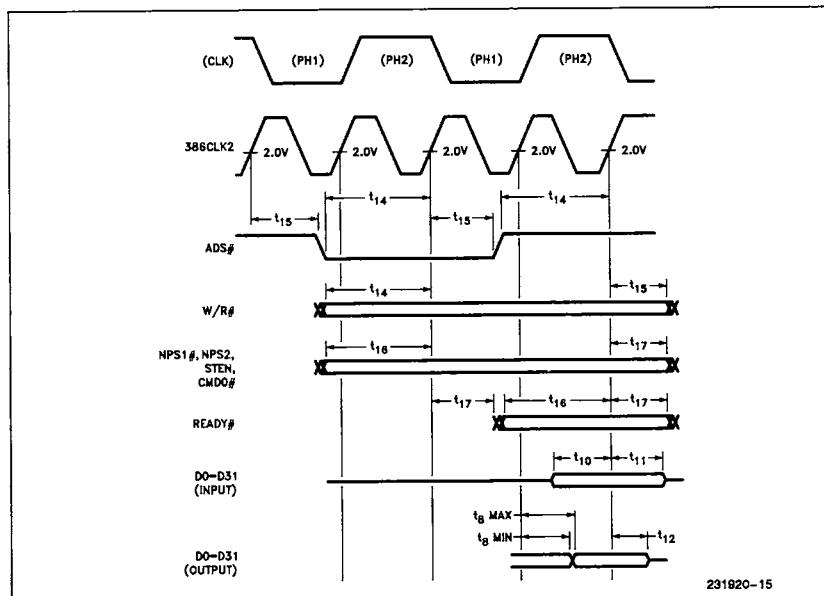


Figure 5.3. Input and I/O Signals

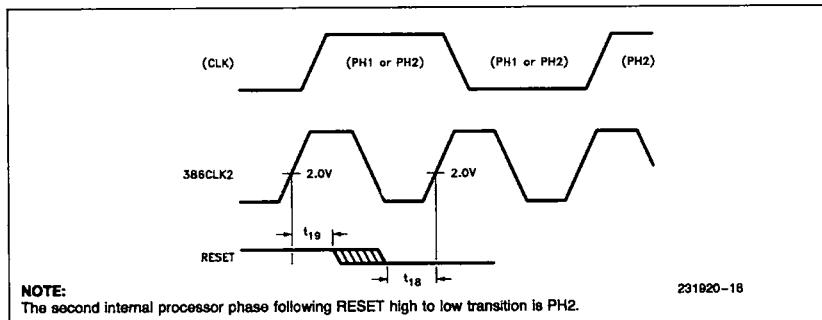


Figure 5.4. RESET Signal

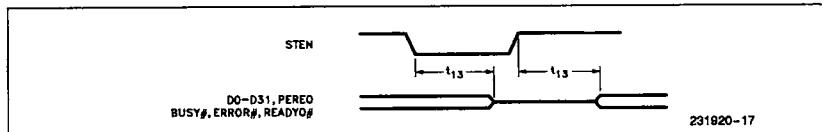


Figure 5.5. Float from STEN

Table 5.3. Other Parameters

Pin	Symbol	Parameter *	Min	Max	Units
RESETIN	t30	Duration	78		387CLK2
RESETIN	t31	RESETIN inactive to 1st opcode write	100		387CLK2
BUSY#	t32	Duration	6		386CLK2
BUSY#, ERROR#	t33	ERROR# (in) active to BUSY# inactive	6		386CLK2
PEREQ, ERROR#	t34	PEREQ inactive to ERROR# active	6		386CLK2
READY#, BUSY#	t35	READY# active to BUSY# active	4	4	386CLK2
READY#	t36	Minimum time from opcode write to Decoder operand write	6		386CLK2
READY#	t37	Minimum time from operand write to Brand write	8		386CLK2

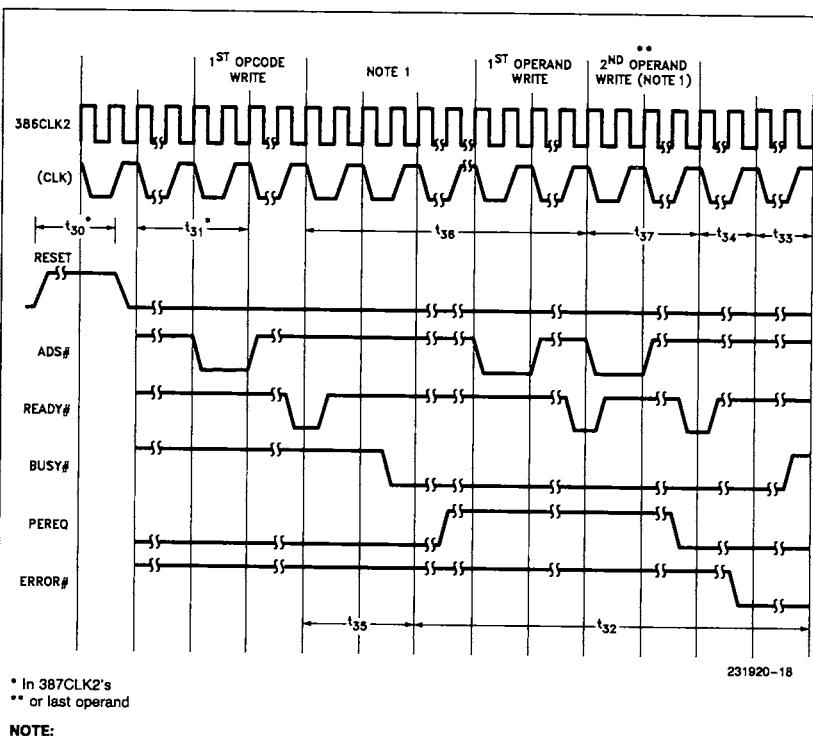


Figure 5.6. Other Parameters

Instruction										Optional Fields	
First Byte				Second Byte							
1	11011	OPA		1	MOD		1	OPB	R/M	SIB	DISP
2	11011	MF		OPA	MOD		OPB		R/M	SIB	DISP
3	11011	d	P	OPA	1	1	OPB		ST(i)		
4	11011	0	0	1	1	1	OP				
5	11011	0	1	1	1	1	OP				

15-11 10 9 8 7 6 5 4 3 2 1 0

6.0 80387 EXTENSIONS TO THE 80386 INSTRUCTION SET

Instructions for the 80387 assume one of the five forms shown in the following table. In all cases, instructions are at least two bytes long and begin with the bit pattern 11011B, which identifies the ESCAPE class of instruction. Instructions that refer to memory operands specify addresses using the 80386 addressing modes.

OP = Instruction opcode, possible split into two fields OPA and OPB

MF = Memory Format
 00—32-bit real
 01—32-bit integer
 10—64-bit real
 11—16-bit integer

P = Pop
 0—Do not pop stack
 1—Pop stack after operation

ESC = 11011

d = Destination
 0—Destination is ST(0)
 1—Destination is ST(i)

R XOR d = 0—Destination (op) Source
 R XOR d = 1—Source (op) Destination

ST(i) = Register stack element *i*

000 = Stack top

001 = Second stack element

•

•

•

111 = Eighth stack element

MOD (Mode field) and R/M (Register/Memory specifier) have the same interpretation as the corresponding fields of 80386 instructions (refer to *80386 Programmer's Reference Manual*)

SIB (Scale Index Base) byte and DISP (displacement) are optionally present in instructions that have MOD and R/M fields. Their presence depends on the values of MOD and R/M, as for 80386 instructions.

The instruction summaries that follow assume that the instruction has been prefetched, decoded, and is ready for execution; that bus cycles do not require wait states; that there are no local bus HOLD requests delaying processor access to the bus; and that no exceptions are detected during instruction execution. If the instruction has MOD and R/M fields that call for both base and index registers, add one clock.



80387 Extensions to the 80386 Instruction Set

Instruction	Encoding			Clock Count Range			
	Byte 0	Byte 1	Optional Bytes 2-6	32-Bit Real	32-Bit Integer	64-Bit Real	16-Bit Integer
DATA TRANSFER							
FLD = Load ^d							
Integer/real memory to ST(0)	ESC MF 1	MOD 000 R/M	SIB/DISP	20	45-52	25	61-65
Long integer memory to ST(0)	ESC 111	MOD 101 R/M	SIB/DISP		56-67		
Extended real memory to ST(0)	ESC 011	MOD 101 R/M	SIB/DISP		44		
BCD memory to ST(0)	ESC 111	MOD 100 R/M	SIB/DISP		266-275		
ST(i) to ST(0)	ESC 001	11000 ST(i)			14		
FST = Store							
ST(0) to integer/real memory	ESC MF 1	MOD 010 R/M	SIB/DISP	44	79-93	45	82-95
ST(0) to ST(i)	ESC 101	11010 ST(i)			11		
FSTP = Store and Pop							
ST(0) to integer/real memory	ESC MF 1	MOD 011 R/M	SIB/DISP	44	79-93	45	82-95
ST(0) to long integer memory	ESC 111	MOD 111 R/M	SIB/DISP		80-97		
ST(0) to extended real	ESC 011	MOD 111 R/M	SIB/DISP		53		
ST(0) to BCD memory	ESC 111	MOD 110 R/M	SIB/DISP		512-534		
ST(0) to ST(i)	ESC 101	11001 ST(i)			12		
FXCH = Exchange							
ST(i) and ST(0)	ESC 001	11001 ST(i)			18		
COMPARISON							
FCOM = Compare							
Integer/real memory to ST(0)	ESC MF 0	MOD 010 R/M	SIB/DISP	26	56-63	31	71-75
ST(i) to ST(0)	ESC 000	11010 ST(i)			24		
FCOMP = Compare and pop							
Integer/real memory to ST	ESC MF 0	MOD 011 R/M	SIB/DISP	26	56-63	31	71-75
ST(i) to ST(0)	ESC 000	11011 ST(i)			26		
FCOMPP = Compare and pop twice							
ST(1) to ST(0)	ESC 110	1101 1001			26		
FTST = Test ST(0)							
	ESC 001	1110 0100			28		
FUCOMP = Unordered compare and pop							
	ESC 101	11100 ST(i)			24		
FUCOMP = Unordered compare and pop twice							
	ESC 101	11101 ST(i)			26		
FUCCOMP = Unordered compare and pop twice							
	ESC 010	1110 1001			26		
FXAM = Examine ST(0)						30-38	
	ESC 001	11100101					
CONSTANTS							
FLDZ = Load +0.0 into ST(0)	ESC 001	1110 1110			20		
FLD1 = Load +1.0 into ST(0)	ESC 001	1110 1000			24		
LDPI = Load pi into ST(0)	ESC 001	1110 1011			40		
LDL2T = Load log ₂ (10) into ST(0)	ESC 001	1110 1001			40		

Shaded areas indicate instructions not available in 8087/80287.

NOTE:

- a. When loading single- or double-precision zero from memory, add 5 clocks.

80387 Extensions to the 80386 Instruction Set (Continued)

Instruction	Encoding			Clock Count Range			
	Byte 0	Byte 1	Optional Bytes 2-6	32-Bit Real	32-Bit Integer	64-Bit Real	16-Bit Integer
CONSTANTS (Continued)							
FLDL2E = Load log ₂ (e) into ST(0)	ESC 001	1110 1010				40	
FLDLG2 = Load log ₁₀ (2) into ST(0)	ESC 001	1110 1100				41	
FLDLN2 = Load log _e (2) into ST(0)	ESC 001	1110 1101				41	
ARITHMETIC							
FADD = Add							
Integer/real memory with ST(0)	ESC MF 0	MOD 000 R/M	SIB/DISP				
ST(i) and ST(0)	ESC d P 0	11000 ST(i)				24-32	57-72 29-37 71-85
FSUB = Subtract							23-31 ^b
Integer/real memory with ST(0)	ESC MF 0	MOD 10 R R/M	SIB/DISP				
ST(i) and ST(0)	ESC d P 0	1110 R R/M				24-32	57-82 28-36 71-83 ^c
FMUL = Multiply							
Integer/real memory with ST(0)	ESC MF 0	MOD 001 R/M	SIB/DISP				
ST(i) and ST(0)	ESC d P 0	1100 1 R/M				27-35	81-82 32-57 76-87
FDIV = Divide							
Integer/real memory with ST(0)	ESC MF 0	MOD 11 R R/M	SIB/DISP				
ST(i) and ST(0)	ESC d P 0	1111 R R/M				88	120-127 ^f 84 136-140 ^e
FSQRT ^d = Square root							88 ^g
FSCALE = Scale ST(0) by ST(1)	ESC 001	1111 1010					122-129
FPREM = Partial remainder							
FPREM1 = Partial remainder (IEEE)	ESC 001	1111 0100					67-86
FRNDINT = Round ST(0) to integer	ESC 001	1111 1100					74-155
FXTRACT = Extract components of ST(0)							
FABS = Absolute value of ST(0)	ESC 001	1111 0100					70-76
FCHS = Change sign of ST(0)	ESC 001	1110 0001					22
	ESC 001	1110 0000					24-25

Shaded areas indicate instructions not available in 8087/80287.

NOTES:

- b. Add 3 clocks to the range when d = 1.
- c. Add 1 clock to each range when R = 1.
- d. Add 3 clocks to the range when d = 0.
- e. typical = 52 (When d = 0, 46-54, typical = 49).
- f. Add 1 clock to the range when R = 1.
- g. 135-141 when R = 1.
- h. Add 3 clocks to the range when d = 1.
- i. $-0 \leq ST(0) \leq +\infty$.



80387

ADVANCE INFORMATION

80387 Extensions to the 80386 Instruction Set (Continued)

Instruction	Encoding			Clock Count Range
	Byte 0	Byte 1	Optional Bytes 2-5	
TRANSCENDENTAL				
FCOS^k = Cosine of ST(0)	ESC 001	1111 1111		123-772
FPTAN^k = Partial tangent of ST(0)	ESC 001	1111 0010		191-4971
FPATAN = Partial arctangent	ESC 001	1111 0011		314-487
FSIN^k = Sine of ST(0)	ESC 001	1111 1110		122-771
FSINCOS^k = Sine and cosine of ST(0)	ESC 001	1111 1011		194-809
F2XM1^l = $2^{ST(0)} - 1$	ESC 001	1111 0000		211-476
FYL2X^m = $ST(1) * \log_2(ST(0))$	ESC 001	1111 0001		120-538
FYL2XP1ⁿ = $ST(1) * \log_2(ST(0)) + 1.0$	ESC 001	1111 1001		257-547
PROCESSOR CONTROL				
FINIT = Initialize NPX	ESC 011	1110 0011		33
FSTSW AX = Store status word	ESC 111	1110 0000		13
FLDCW = Load control word	ESC 001	MOD 101 R/M	SIB/DISP	19
FSTCW = Store control word	ESC 101	MOD 111 R/M	SIB/DISP	15
FSTSW = Store status word	ESC 101	MOD 111 R/M	SIB/DISP	15
FCLEX = Clear exceptions	ESC 011	1110 0010		11
FSTENV = Store environment	ESC 001	MOD 110 R/M	SIB/DISP	103-104
FLDENV = Load environment	ESC 001	MOD 100 R/M	SIB/DISP	71
FSAVE = Save state	ESC 101	MOD 110 R/M	SIB/DISP	375-376
FRSTOR = Restore state	ESC 101	MOD 100 R/M	SIB/DISP	308
FINCSTP = Increment stack pointer	ESC 001	1111 0111		21
FDECSTP = Decrement stack pointer	ESC 001	1111 0110		22
FFREE = Free ST(i)	ESC 101	1100 0 ST(i)		18
FNOP = No operations	ESC 001	1101 0000		12

Shaded areas indicate instructions not available in 8087/80287.

NOTES:

- j. These timings hold for operands in the range $|x| < \pi/4$. For operands not in this range, up to 76 additional clocks may be needed to reduce the operand.
- k. $0 \leq |ST(0)| < 2^{63}$.
- l. $-0.5 \leq ST(0) \leq 0.5$.
- m. $0 \leq ST(0) < \infty, -\infty < ST(1) < +\infty$.
- n. $0 \leq |ST(0)| < (2 - SQRT(2))/2, -\infty < ST(1) < +\infty$.

APPENDIX A COMPATIBILITY BETWEEN THE 80287 AND THE 8087

The 80286/80287 operating in Real-Address mode will execute 8086/8087 programs without major modification. However, because of differences in the handling of numeric exceptions by the 80287 NPX and the 8087 NPX, exception-handling routines *may* need to be changed.

This appendix summarizes the differences between the 80287 NPX and the 8087 NPX, and provides details showing how 8086/8087 programs can be ported to the 80286/80287.

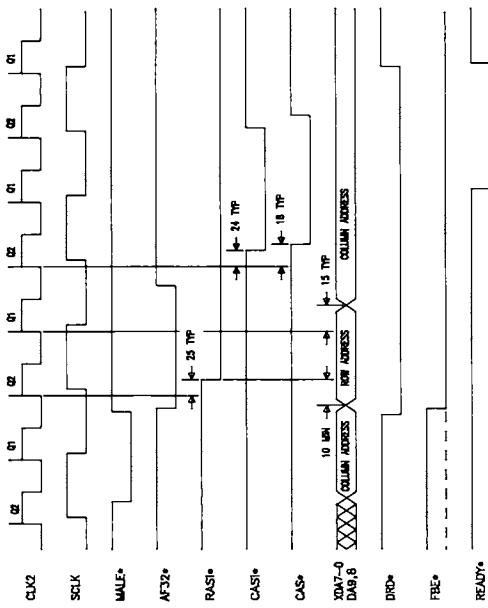
1. The NPX signals exceptions through a dedicated ERROR line to the 80286. The NPX error signal does not pass through an interrupt controller (the 8087 INT signal does). Therefore, any interrupt-controller-oriented instructions in numeric exception handlers for the 8086/8087 should be deleted.
2. The 8087 instructions FENI/FNENI and FDISI/FNDISI perform no useful function in the 80287. If the 80287 encounters one of these opcodes in its instruction stream, the instruction will effectively be ignored—none of the 80287 internal states will be updated. While 8086/8087 containing these instructions may be executed on the 80286/80287, it is unlikely that the exception-handling routines containing these instructions will be completely portable to the 80287.
3. Interrupt vector 16 must point to the numeric exception handling routine.
4. The ESC instruction address saved in the 80287 includes any leading prefixes before the ESC opcode. The corresponding address saved in the 8087 does not include leading prefixes.
5. In Protected-Address mode, the format of the 80287's saved instruction and address pointers is different than for the 8087. The instruction opcode is not saved in Protected mode—exception handlers will have to retrieve the opcode from memory if needed.
6. Interrupt 7 will occur in the 80286 when executing ESC instructions with either TS (task switched) or EM (emulation) of the 80286 MSW set (TS = 1 or EM = 1). If TS is set, then a WAIT instruction will also cause interrupt 7. An exception handler should be included in 80286/80287 code to handle these situations.
7. Interrupt 9 will occur if the second or subsequent words of a floating-point operand fall outside a segment's size. Interrupt 13 will occur if the starting address of a numeric operand falls outside a segment's size. An exception handler should be included in 80286/80287 code to report these programming errors.
8. Except for the processor control instructions, all of the 80287 numeric instructions are automatically synchronized by the 80286 CPU—the 80286 automatically tests the BUSY line from the 80287 to ensure that the 80287 has completed its previous instruction before executing the next ESC instruction. No explicit WAIT instructions are required to assure this synchronization. For the 8087 used with 8086 and 8088 processors, explicit WAITS are required before each numeric instruction to ensure synchronization. Although 8086/8087 programs having explicit WAIT instructions will execute perfectly on the 80286/80287 without reassembly, these WAIT instructions are unnecessary.
9. Since the 80287 does not require WAIT instructions before each numeric instruction, the ASM86 assembler does not automatically generate these WAIT instructions. The ASM86 assembler, however, automatically precedes every ESC instruction with a WAIT instruction. Although numeric routines generated using the ASM86 assembler will generally execute correctly on the 80286/80287, reassembly using ASM86 may result in a more compact code image.

The processor control instructions for the 80287 may be coded using either a WAIT or No-WAIT form of mnemonic. The WAIT forms of these instructions cause ASM86 to precede the ESC instruction with a CPU WAIT instruction, in the identical manner as does ASM86.

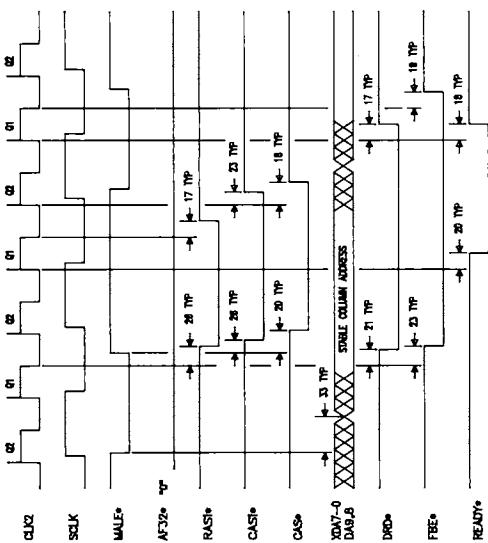


RADIO SHACK
A Division of Tandy Corporation
Fort Worth, Texas 76102

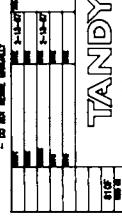
82C302 TIMING DIAGRAM (CPU TO LOCAL MEMORY)
READ CYCLE WITH RAS BEING INACTIVE



82C302 TIMING DIAGRAM (CPU TO LOCAL MEMORY)
READ HIT, 0 WAIT STATE



THIS IS A CD-REWRITABLE DOCUMENT

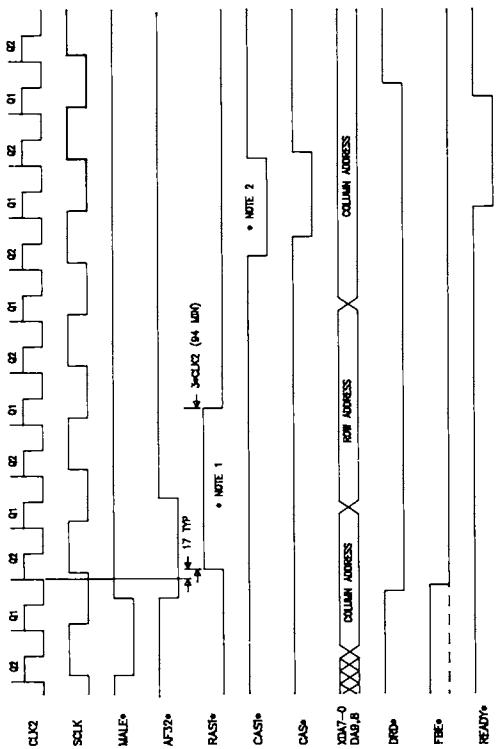


TANDY

D 1 Of



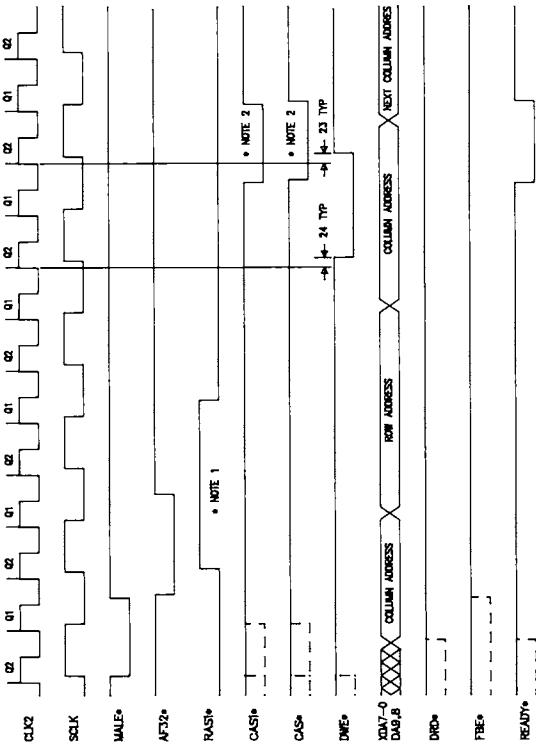
82C302 TIMING DIAGRAM (CPU TO LOCAL MEMORY)
READ MISS CYCLE



NOTES

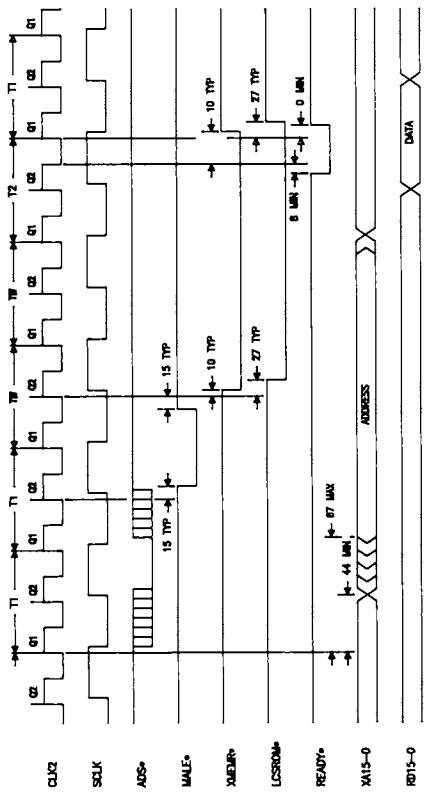
1. RAS# PREDENCE TIME WILL BE INCREASED BY 2 CLOCK CYCLES IF BIT 7 OF REGISTER 11 (OR 15) IS PROGRAMMED TO "1".
2. CAS# & CAS# PULSE WIDTHS WILL BE INCREASED BY 2 CLOCK CYCLES IF BIT 6 OF REGISTER 11 (OR 15) IS PROGRAMMED TO "1".

82C302 TIMING DIAGRAM (CPU TO LOCAL MEMORY)
WRITE MISS CYCLE





ROM READ CYCLE

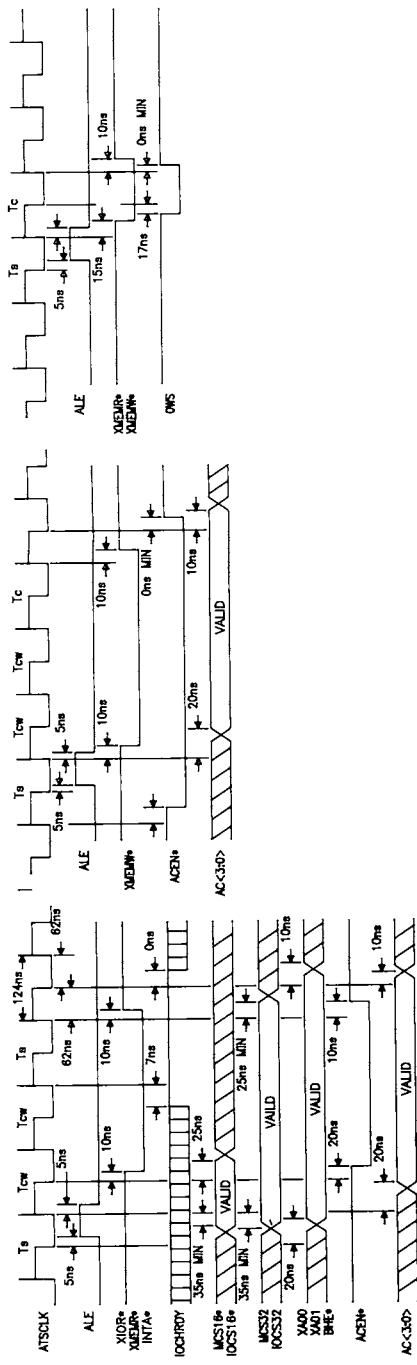


NOTES

1. ALL UNITS IN NANOSECONDS.



82C301 I/O TIMING DIAGRAMS

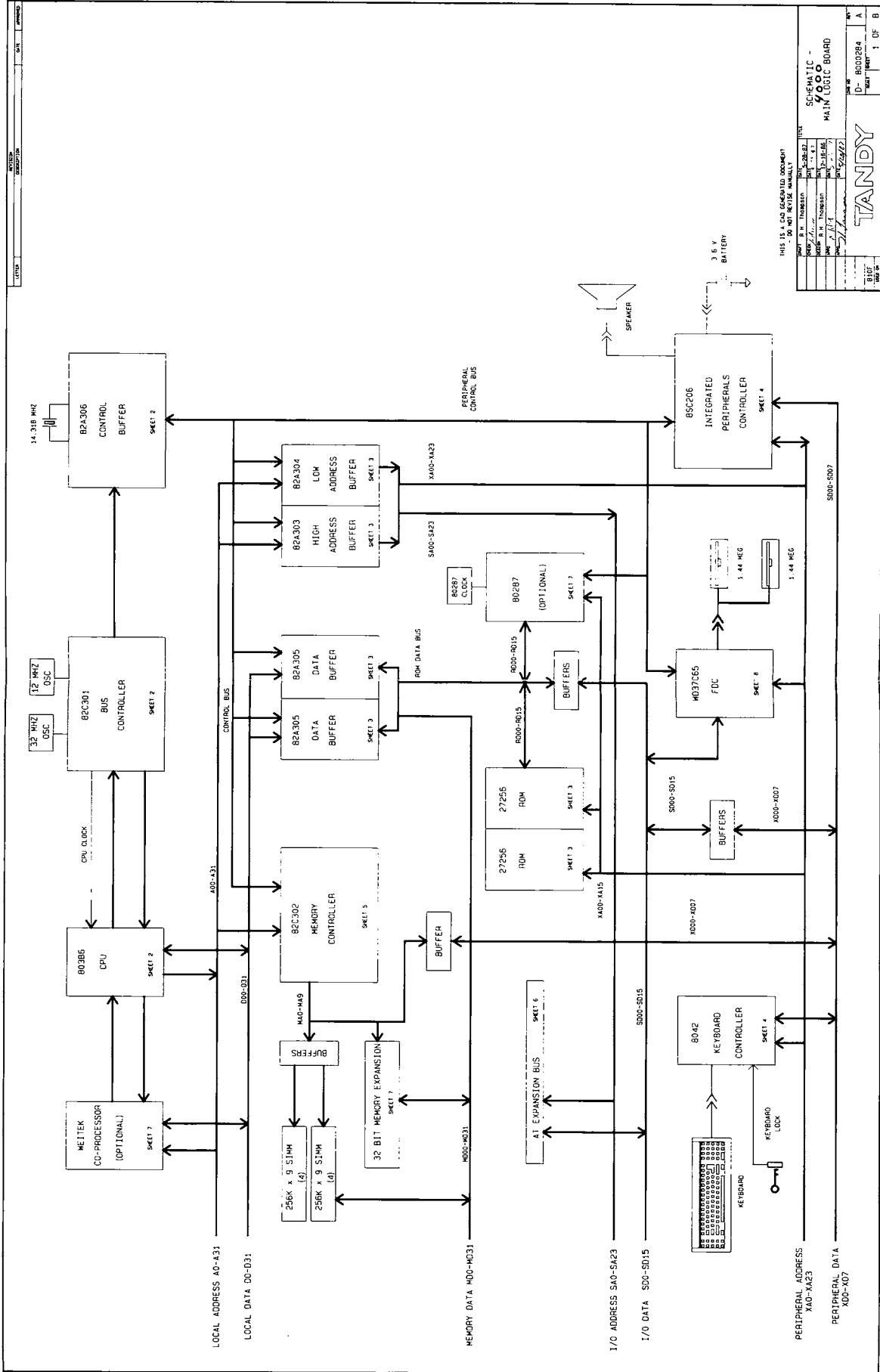


TANDY

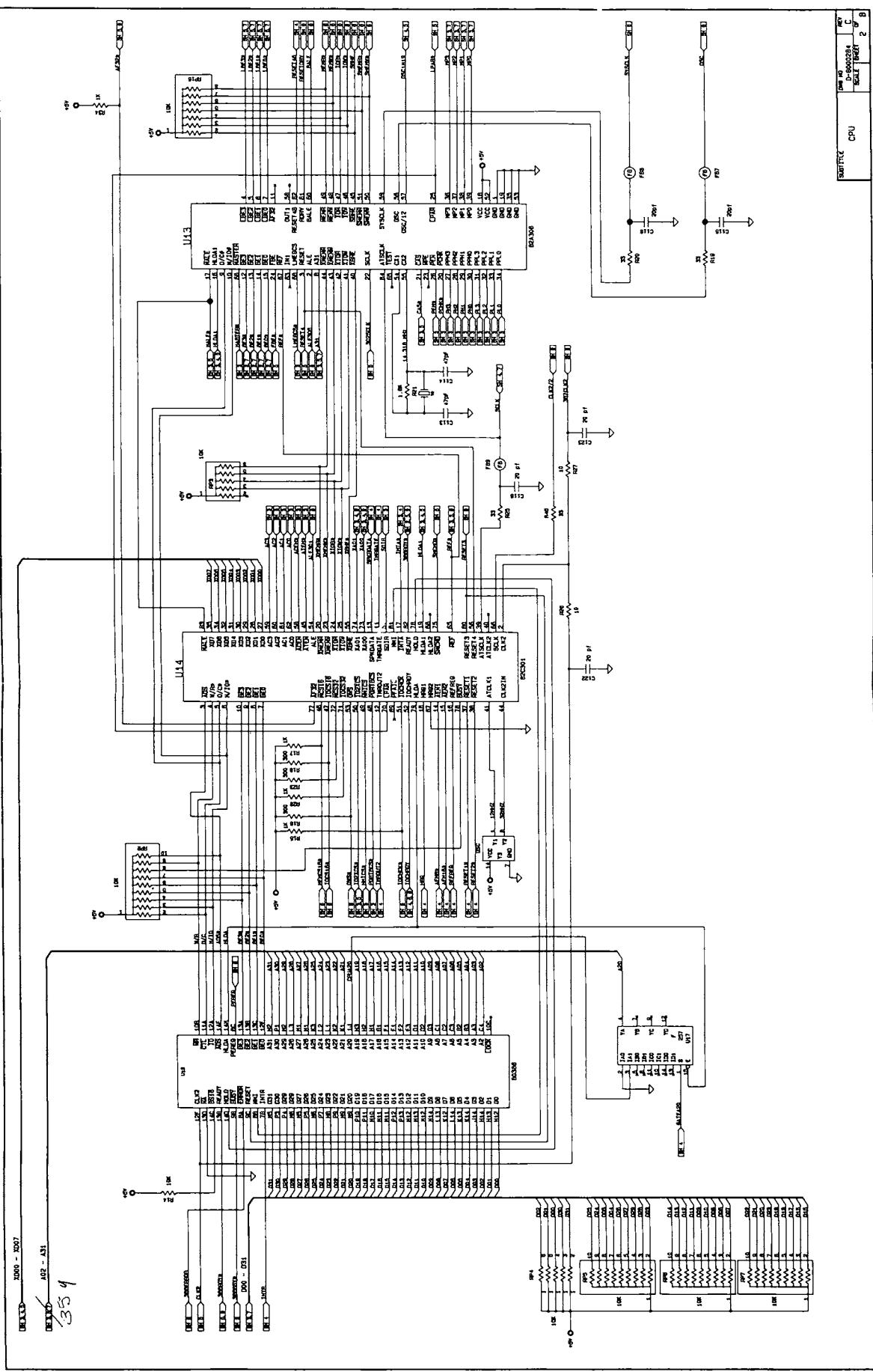
1 of

TANDY

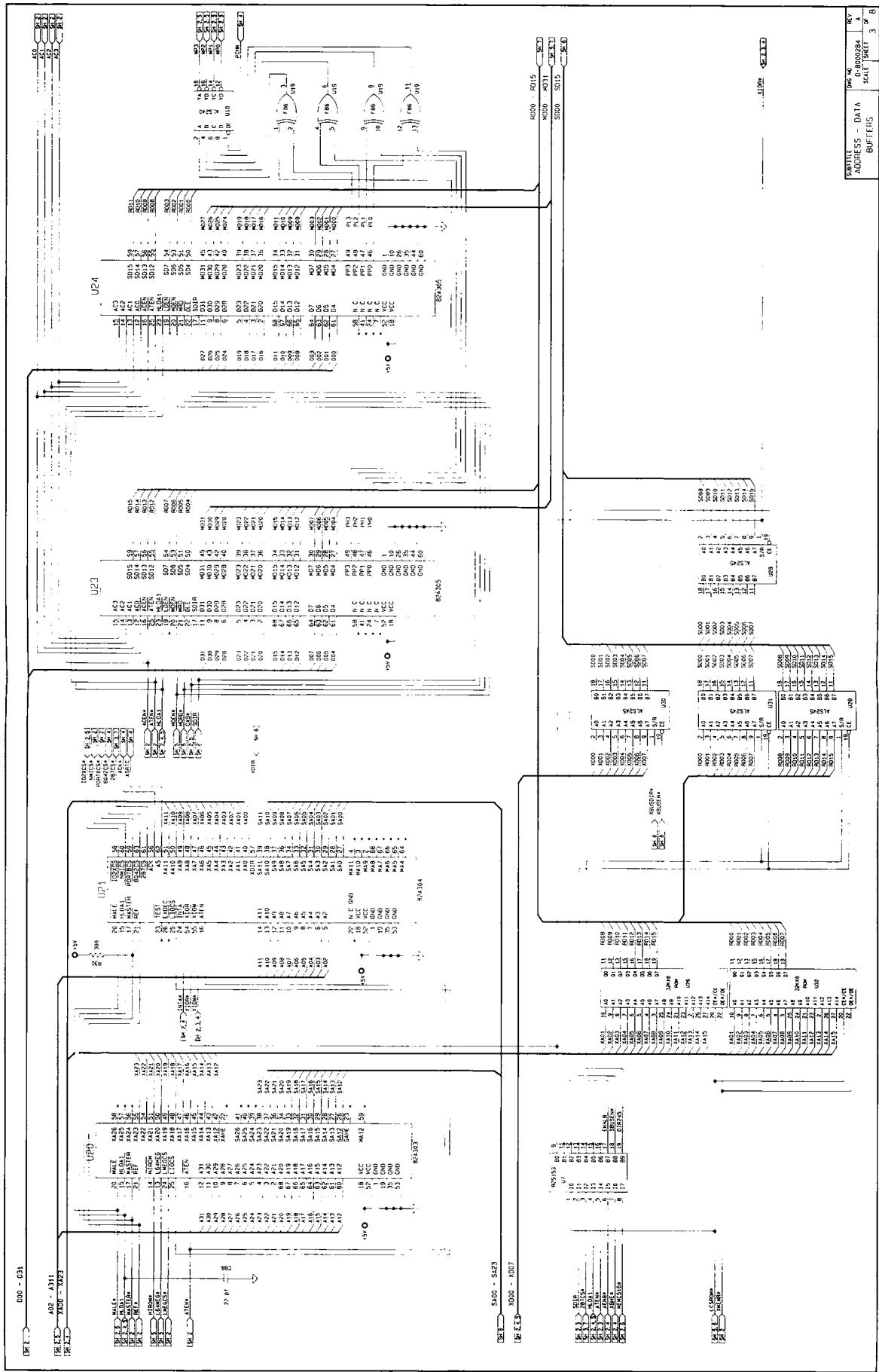




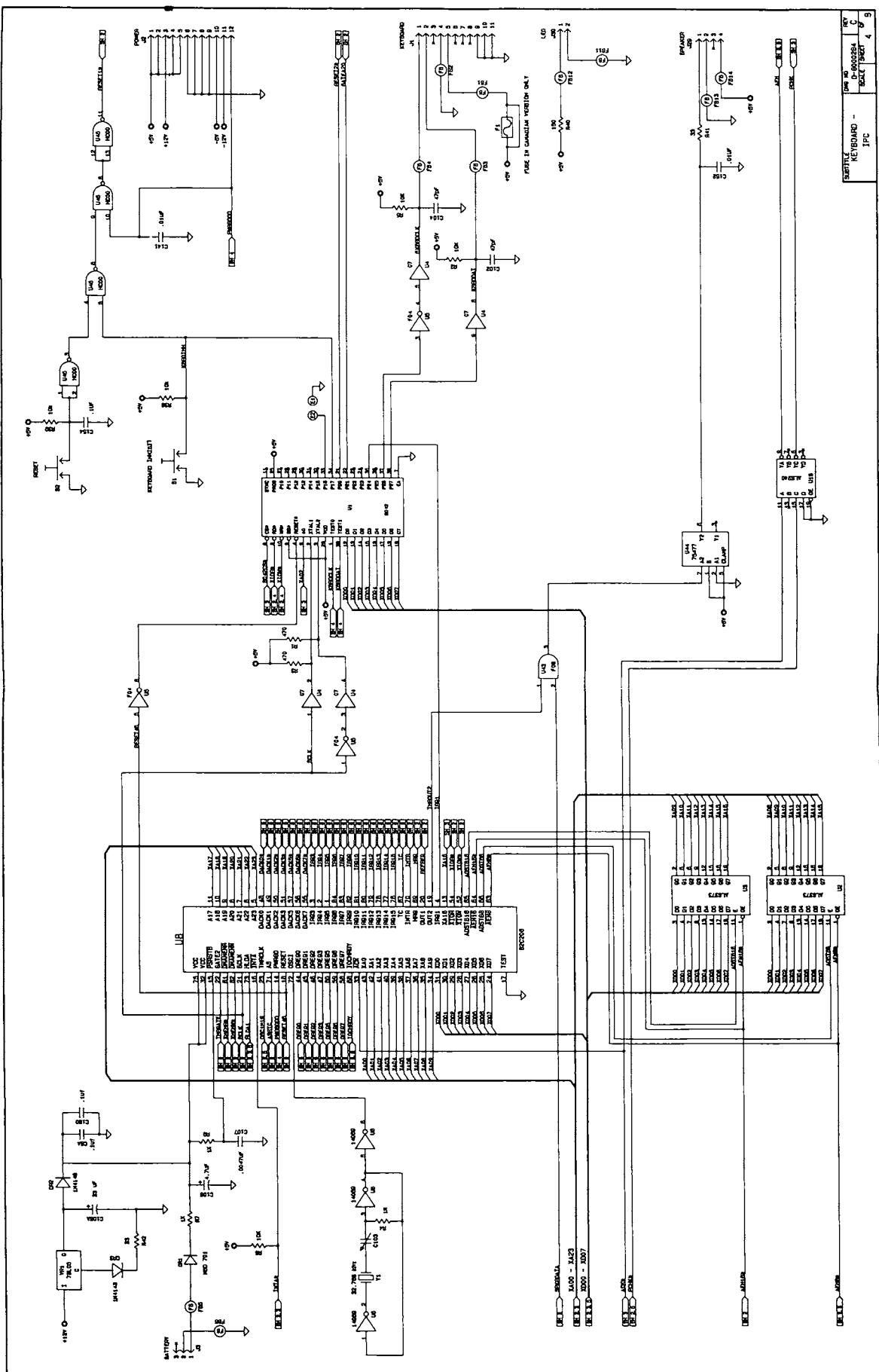




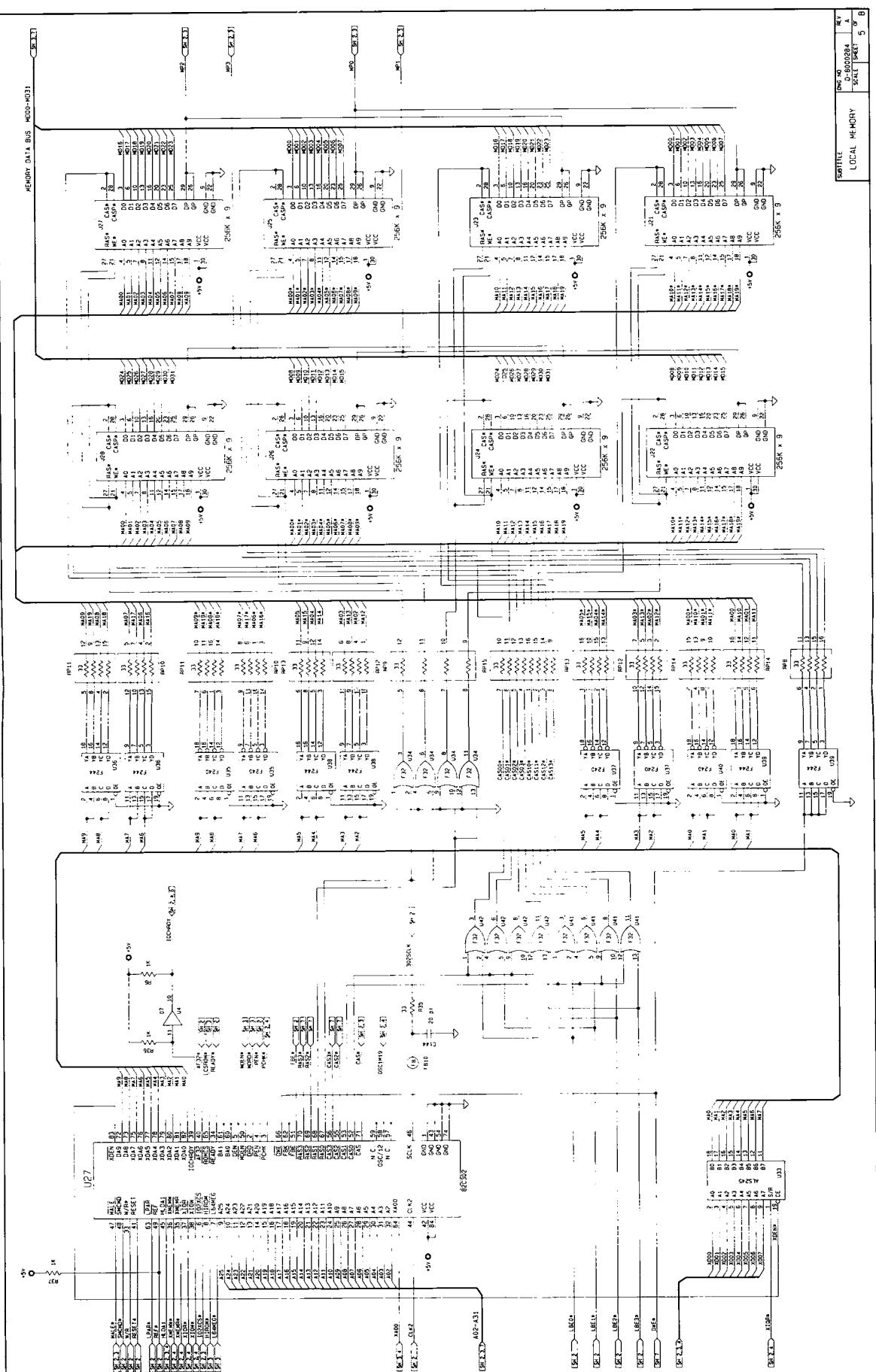








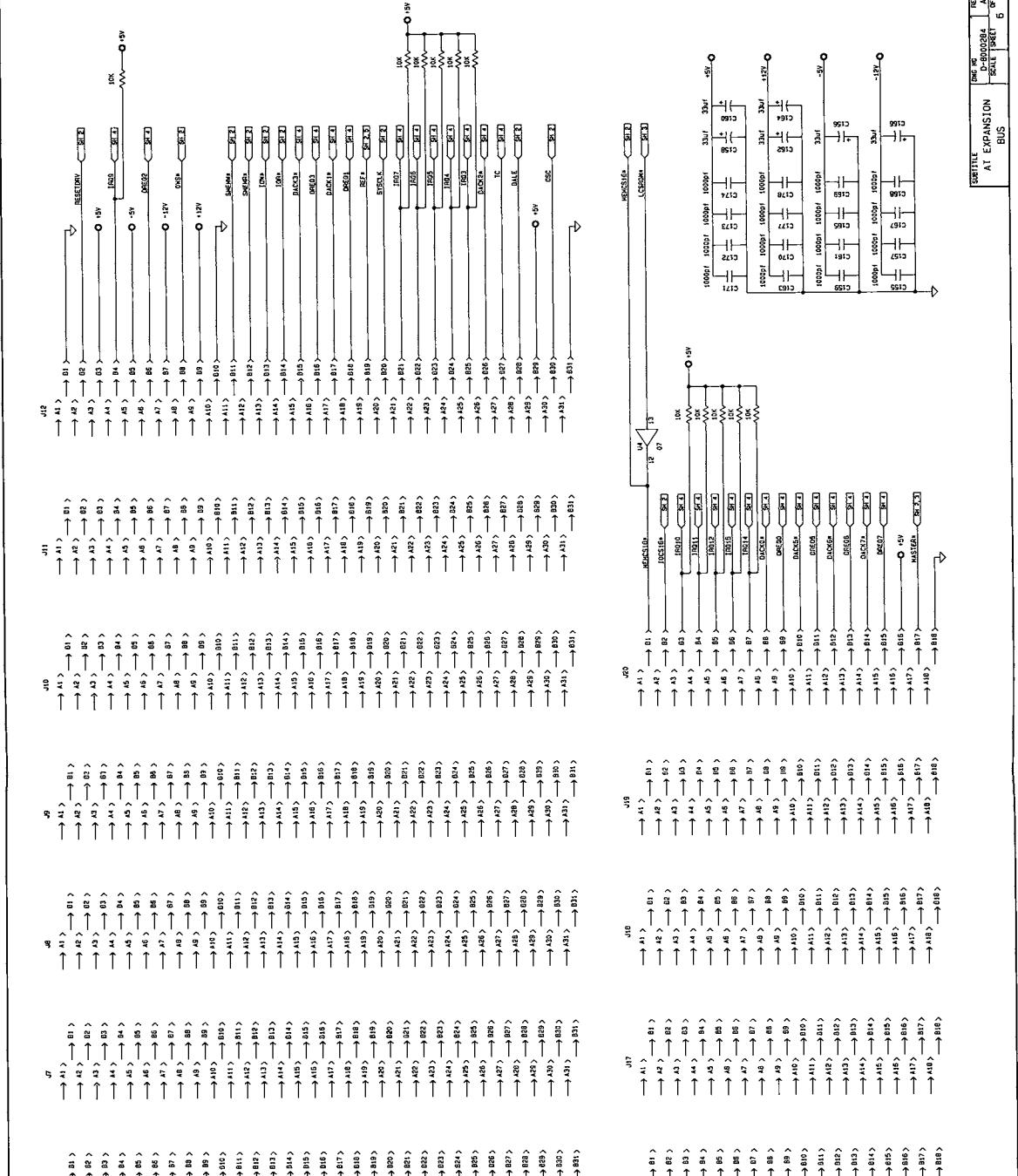




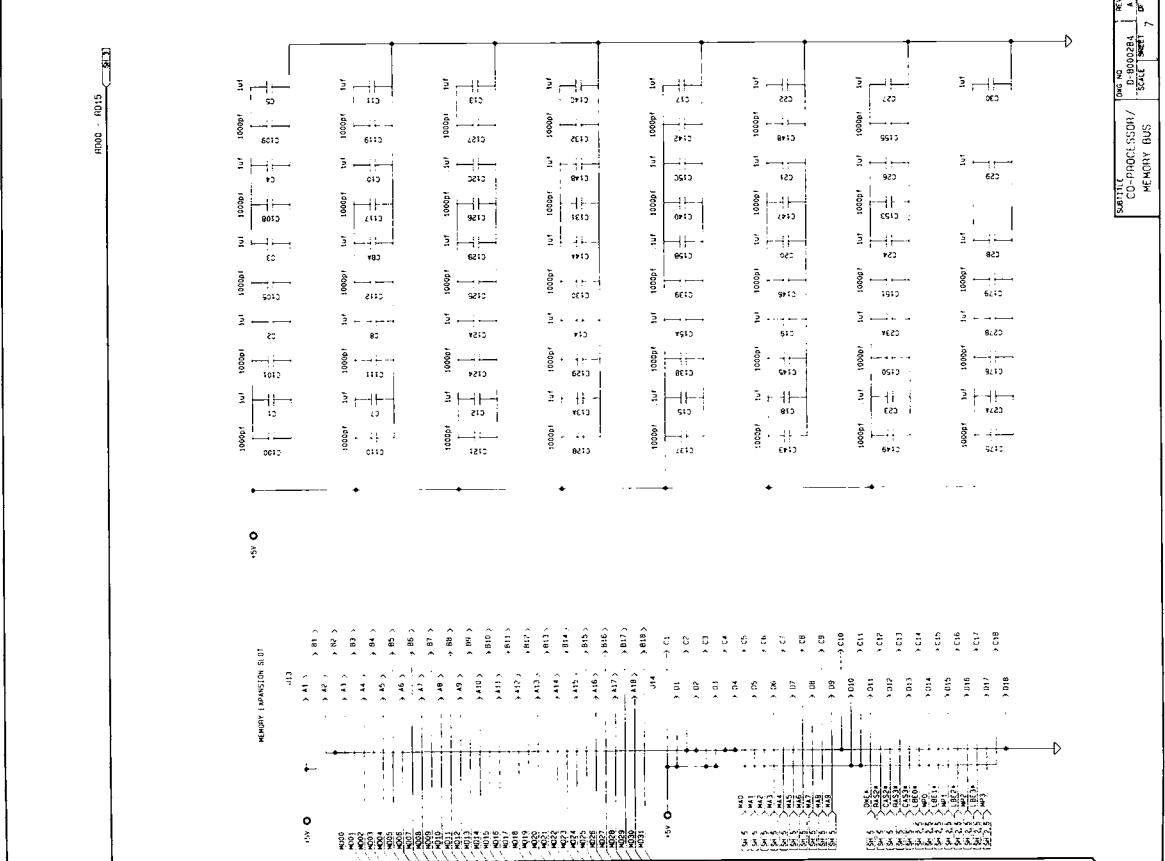
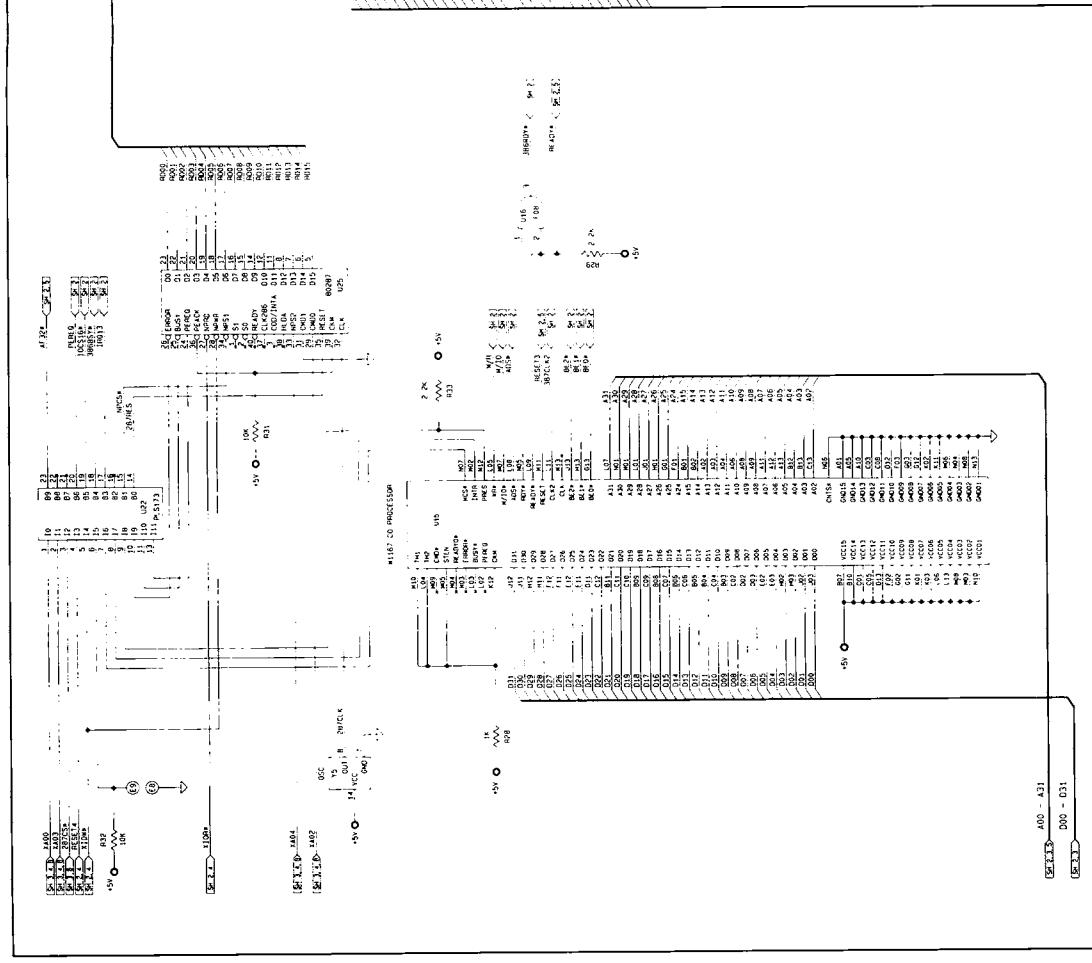
SOURCE D-602028-A
LOCAL MEMORY -
RECALL SET 5 OF

SOURCE D-602028-A
LOCAL MEMORY -
RECALL SET 5 OF

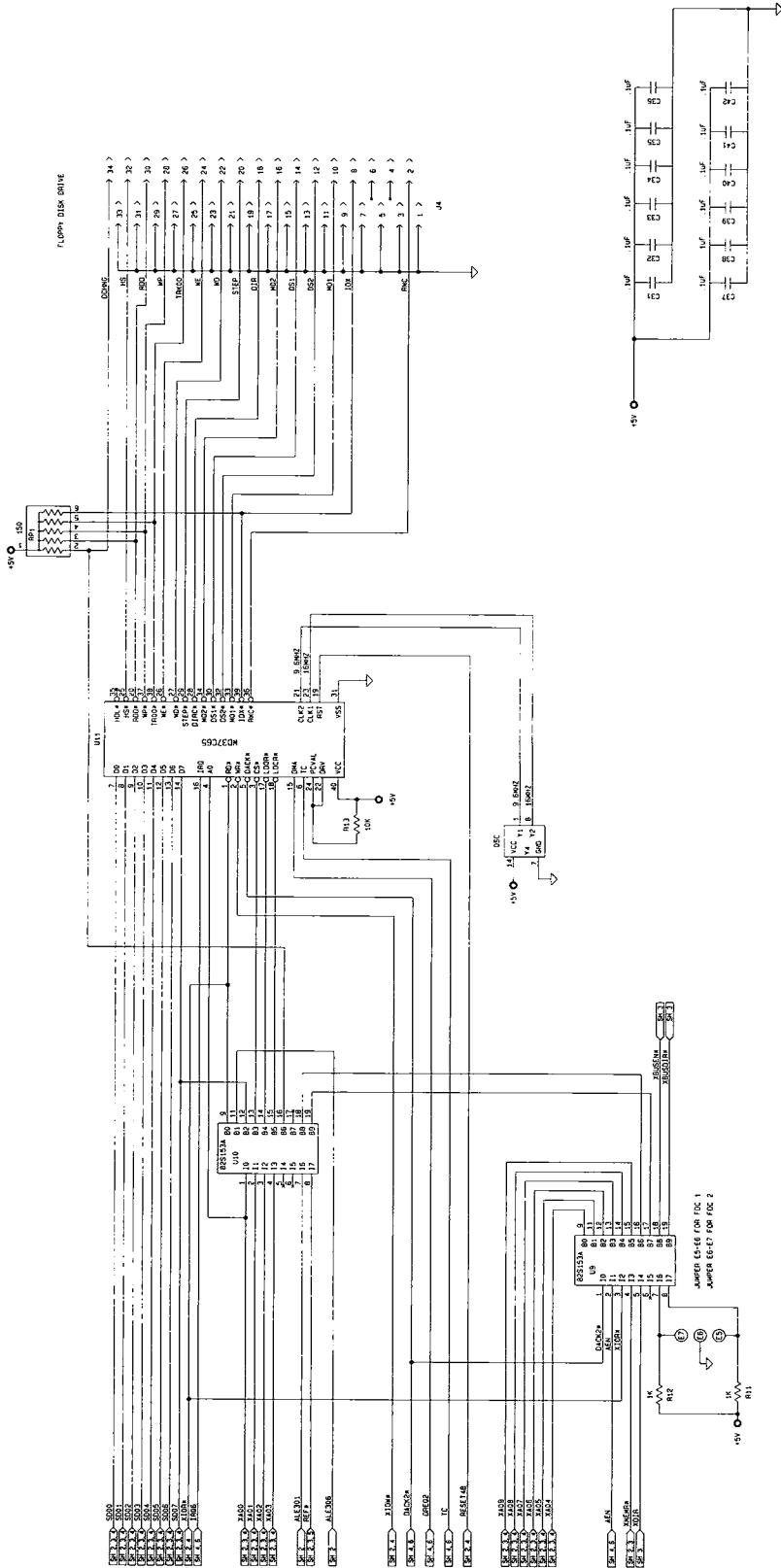


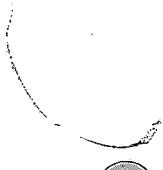




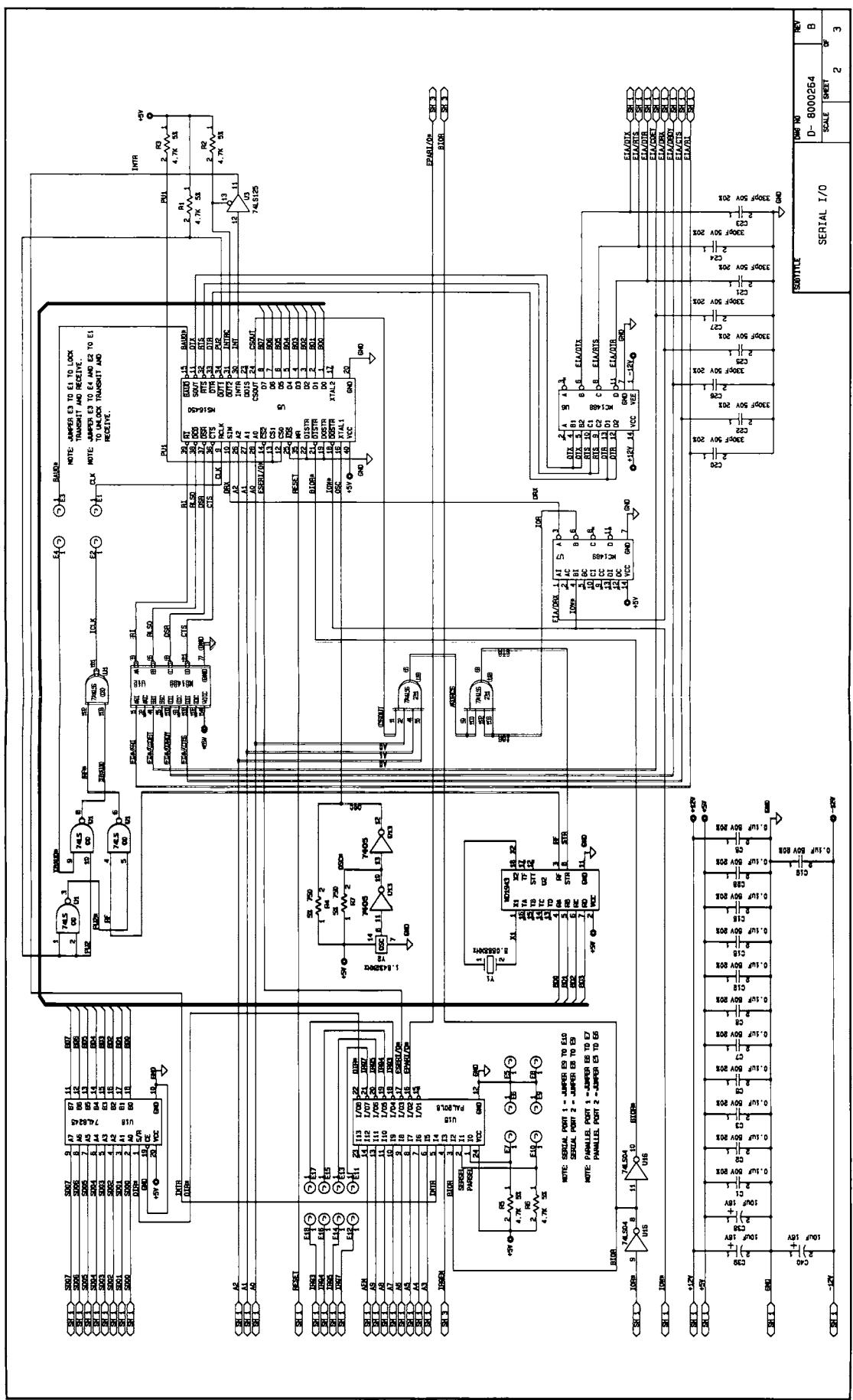




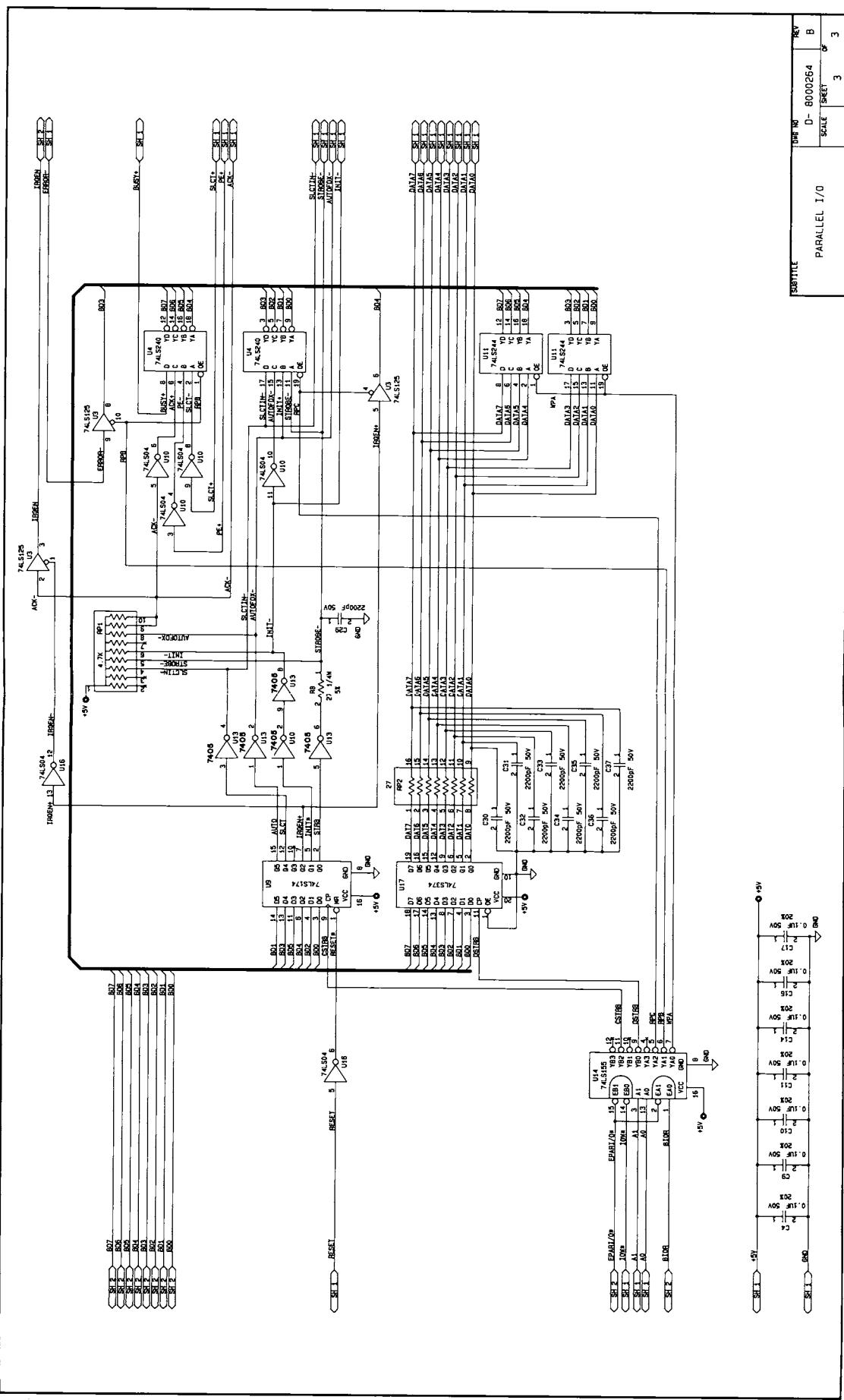




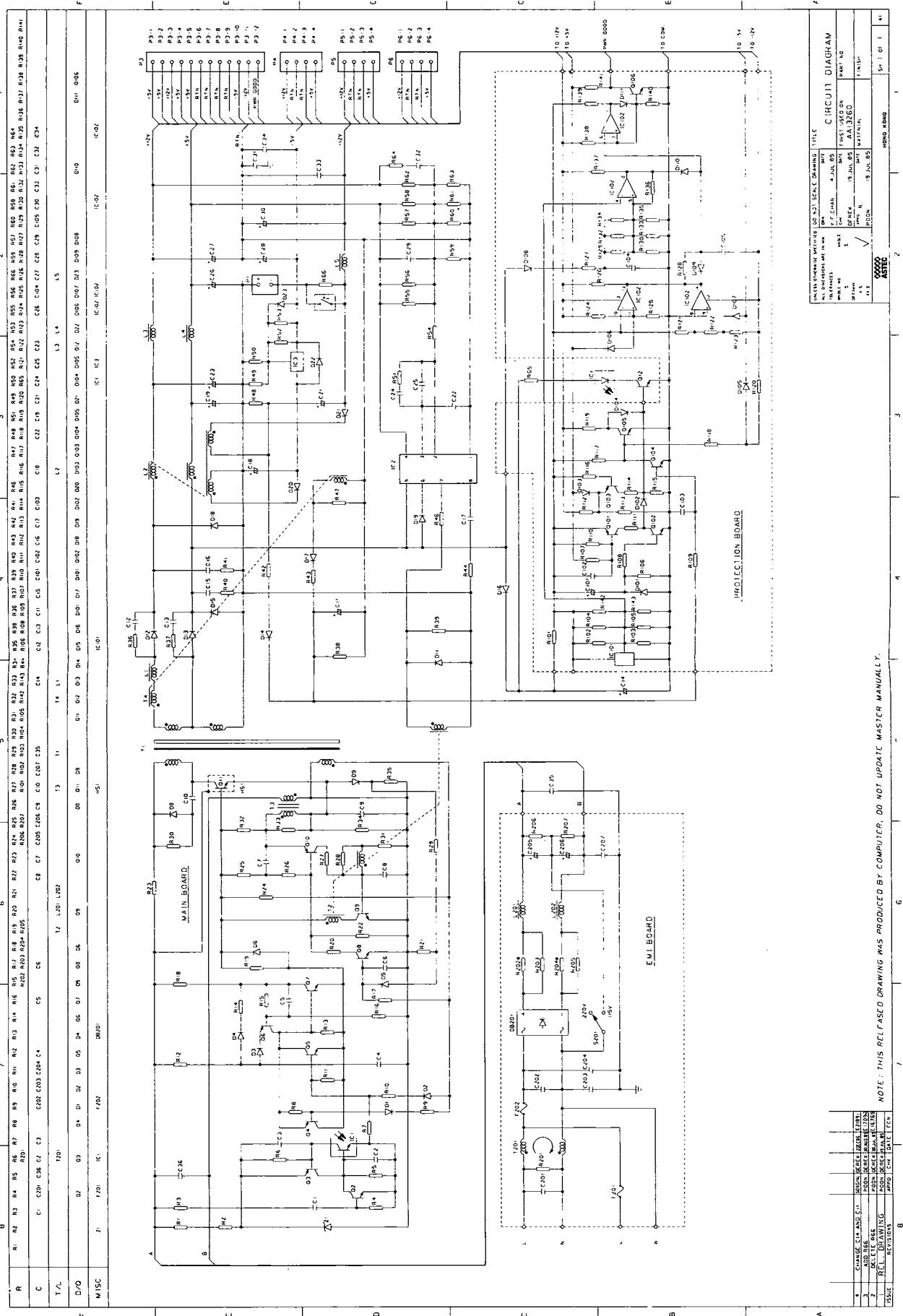






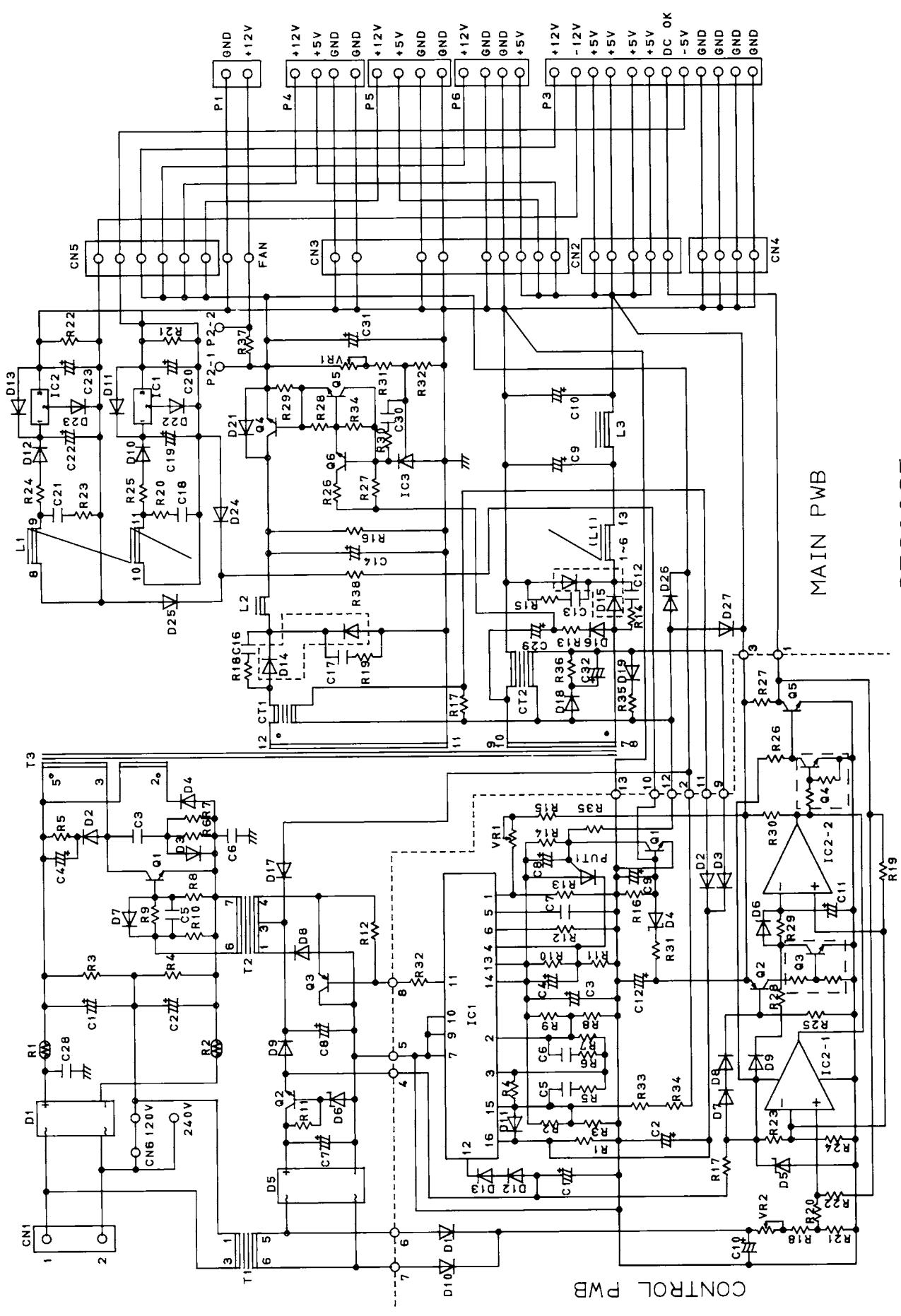








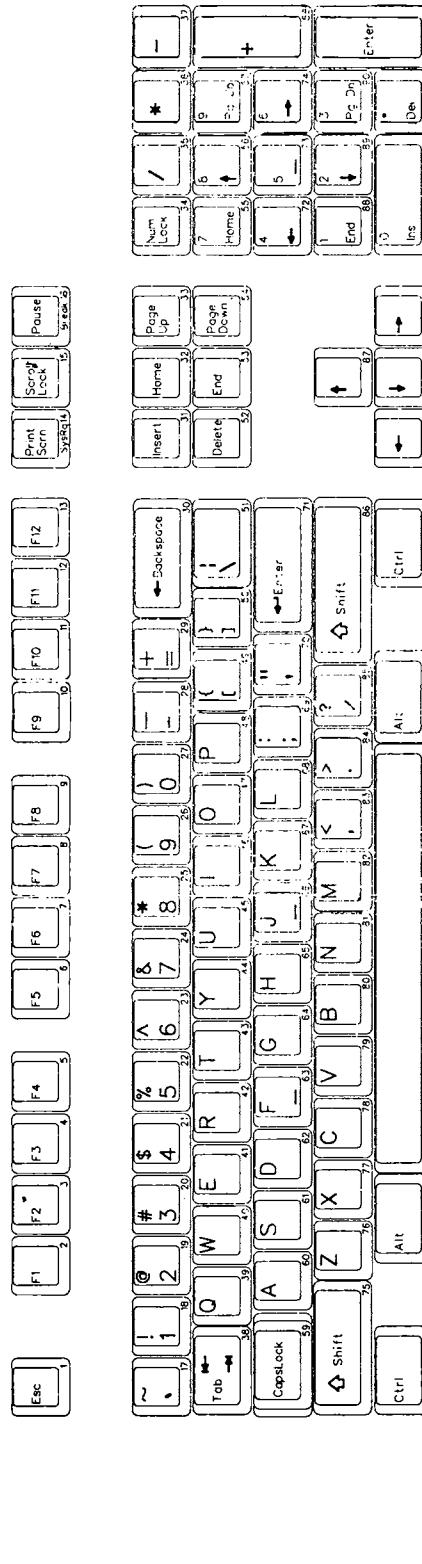
model no. 8790095





NOTICE OF INFORMATION: Information contained in this document is proprietary to Keytop Corporation. It is furnished under the terms of a license agreement or a nondisclosure agreement. It is to be used only in accordance with the terms of the agreement. Any unauthorized disclosure or use is illegal.

		REV	ECO	B	E	DESCRIPTION	ADDITIONAL
		-	-	P15	B-01TC	IN 1000	
A	-	CLW	-	V.G. RELEASE	F5R 6-7-86		
B	30856	MC	-	LEGEND C-HG	7-7		



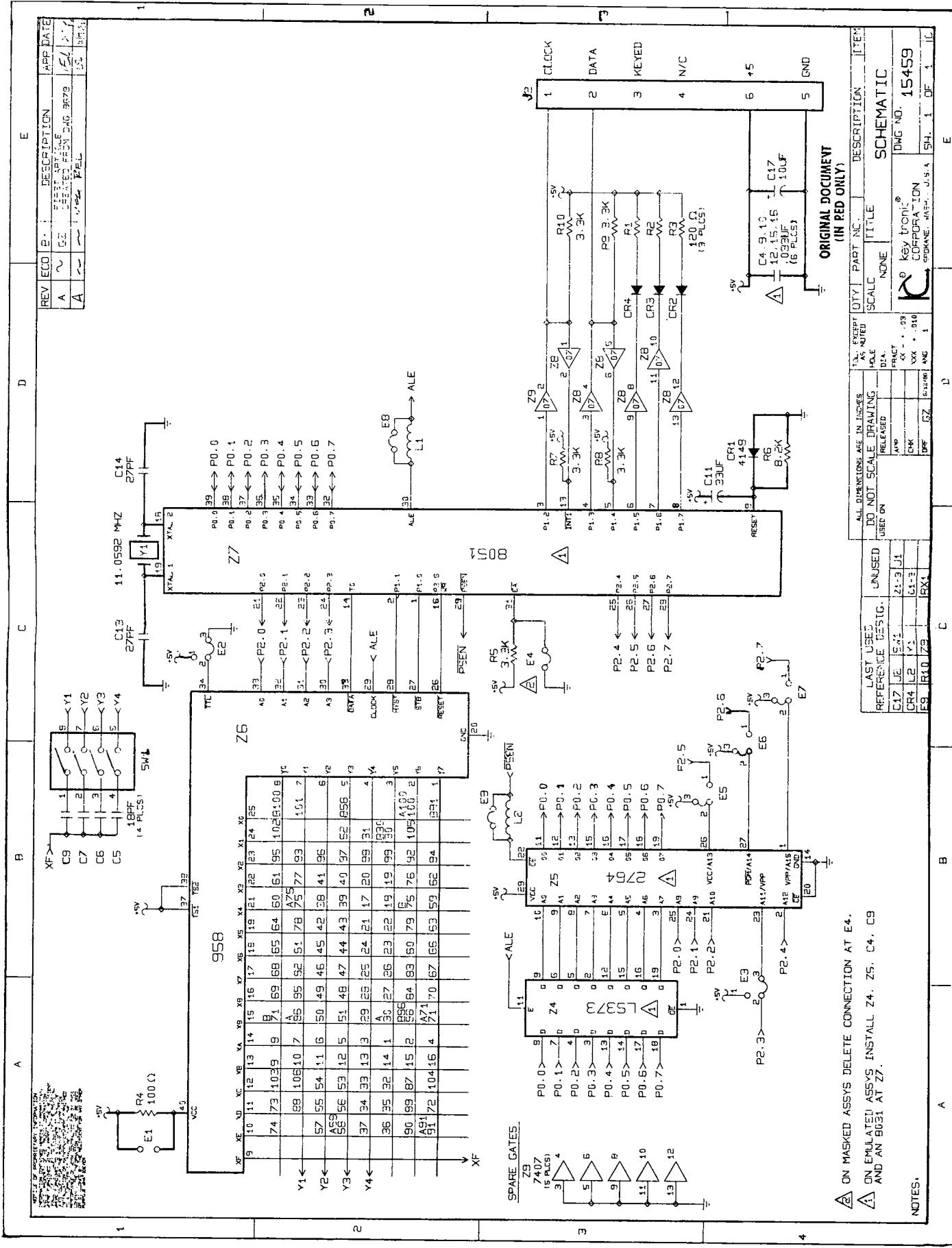
2. THIS DRAWING IS INTENDED TO AID IN THE PROPER ARRANGEMENT OF KEYTOPS ON THE KEYBOARD. THE KEYTOP LEGENDS ARE SHOWN FOR REFERENCE ONLY. ANY DISCREPANCIES BETWEEN LEGEND SHOWN AND EXACT LEGEND AS TOOLED SHOULD BE RESOLVED BY REFERRING TO THE KEYTOP BILL OF MATERIAL AND KEYTOP CODE BOOK.

NOTES: UNLESS OTHERWISE SPECIFIED

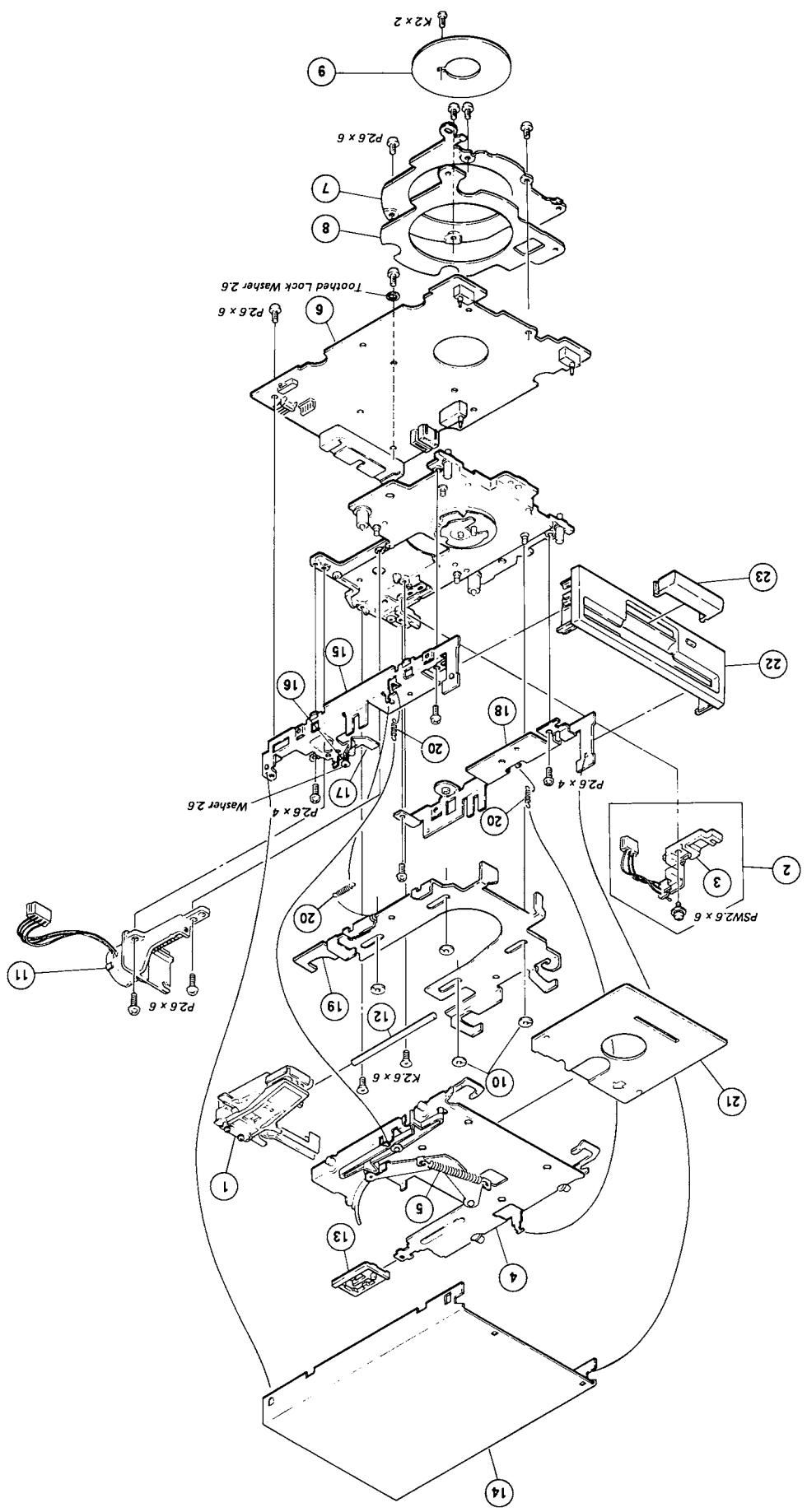
UNLESS OTHERWISE NOTED ALL DIMENSIONS ARE IN INCHES DO NOT SCALE DRAWINGS USED ON	TOL. EXCEPT AS NOTED AS NOTED RELEASED APR	DRAWING TITLE KEYTOP	DESCRIPTION		REV SET, KEYTOP
			TO	FROM	
	xx .03 xxx .03 .005 .005	key tronic @ nos no. CORPORATION, SPOKANE, WASH. STREET 1 OF 1	15597		C

1 A D S: V

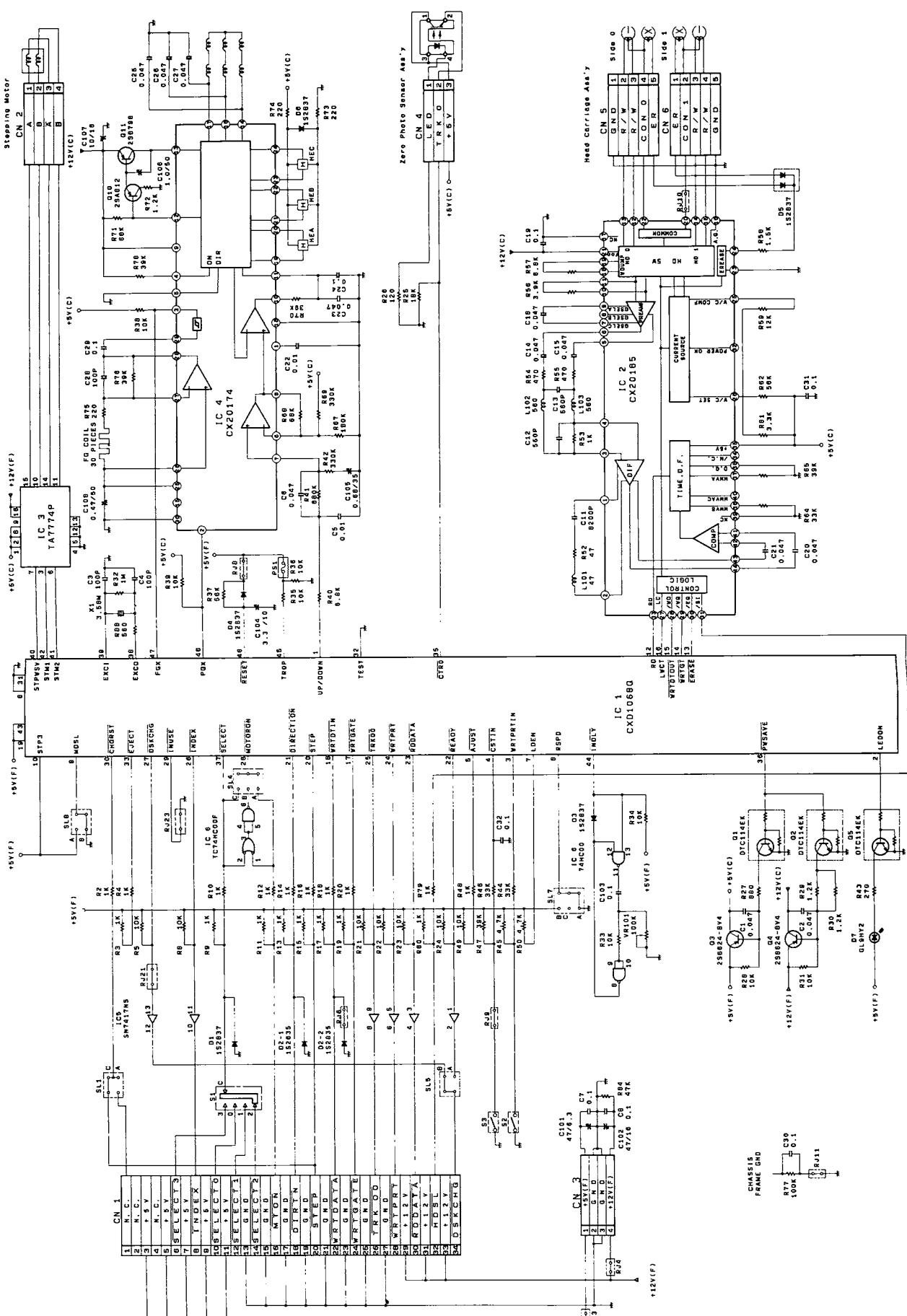




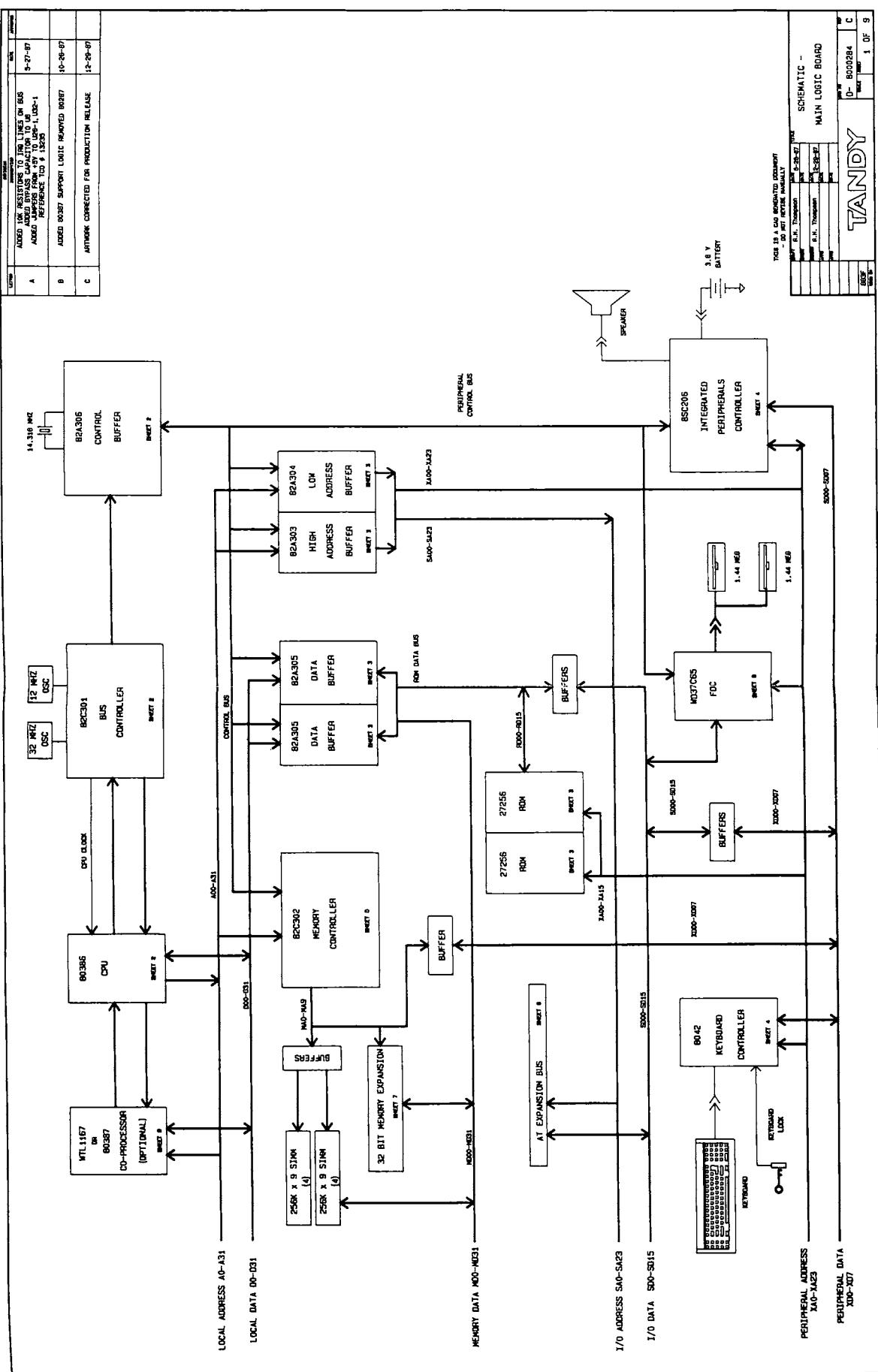




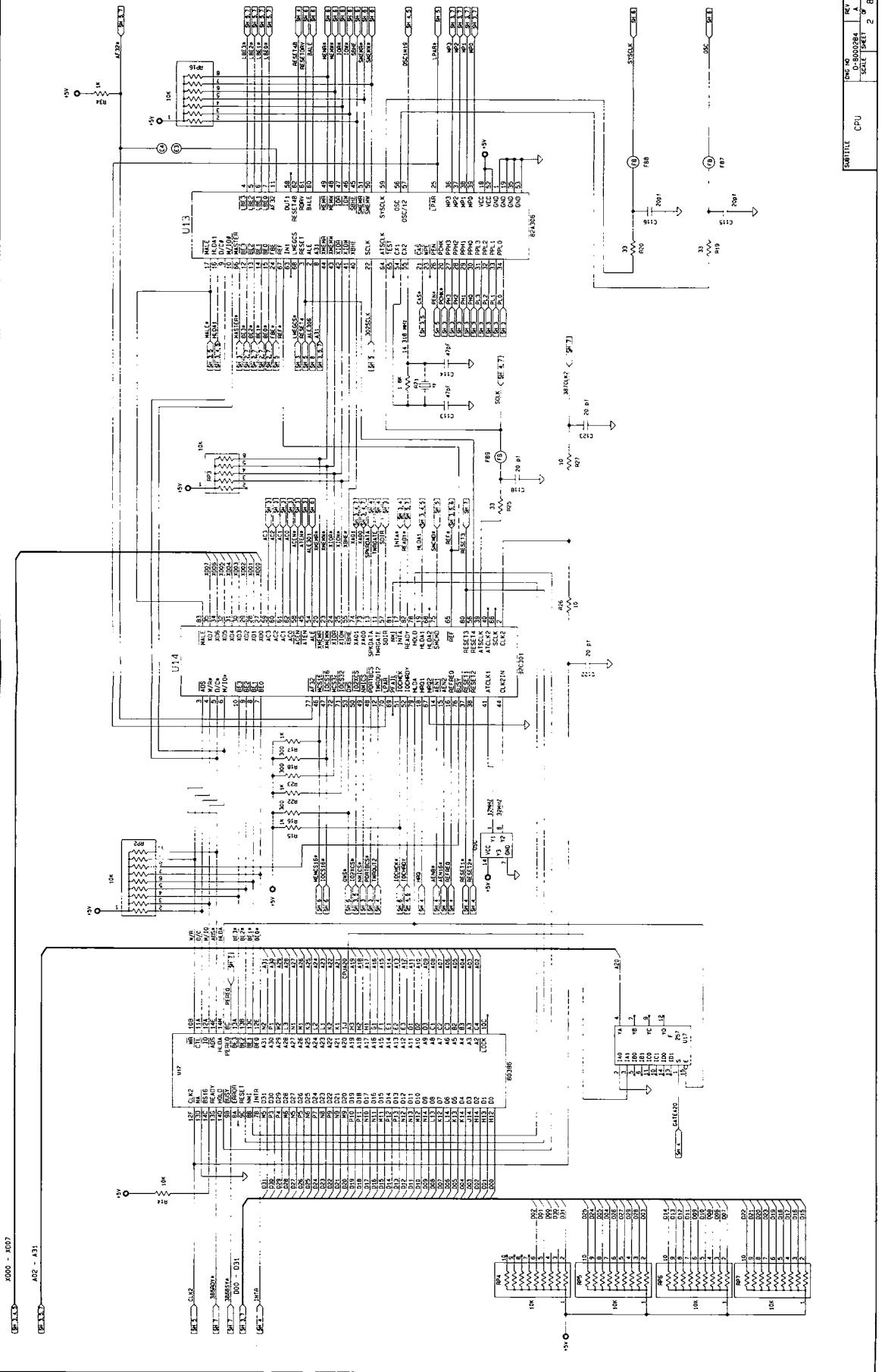




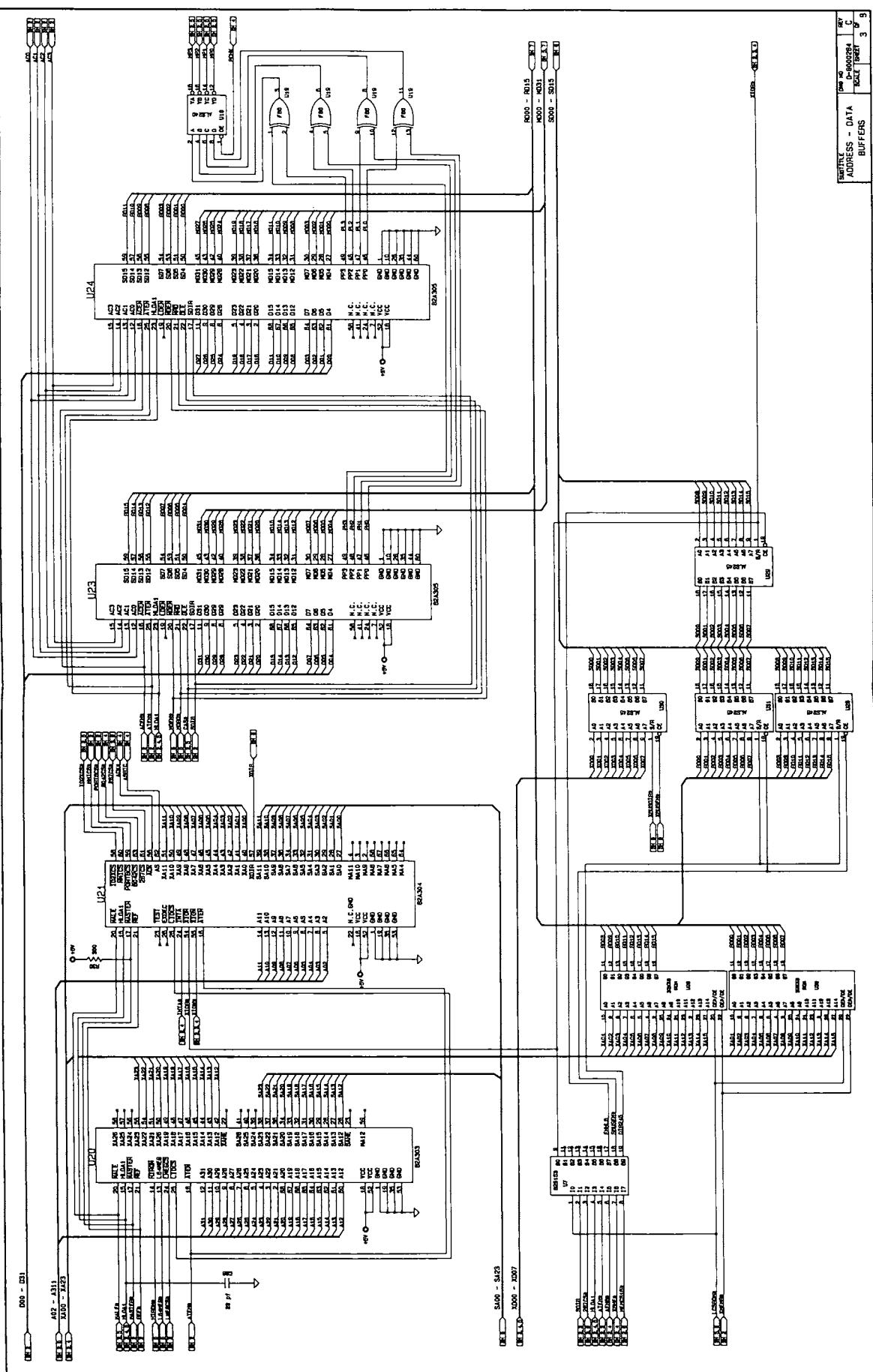




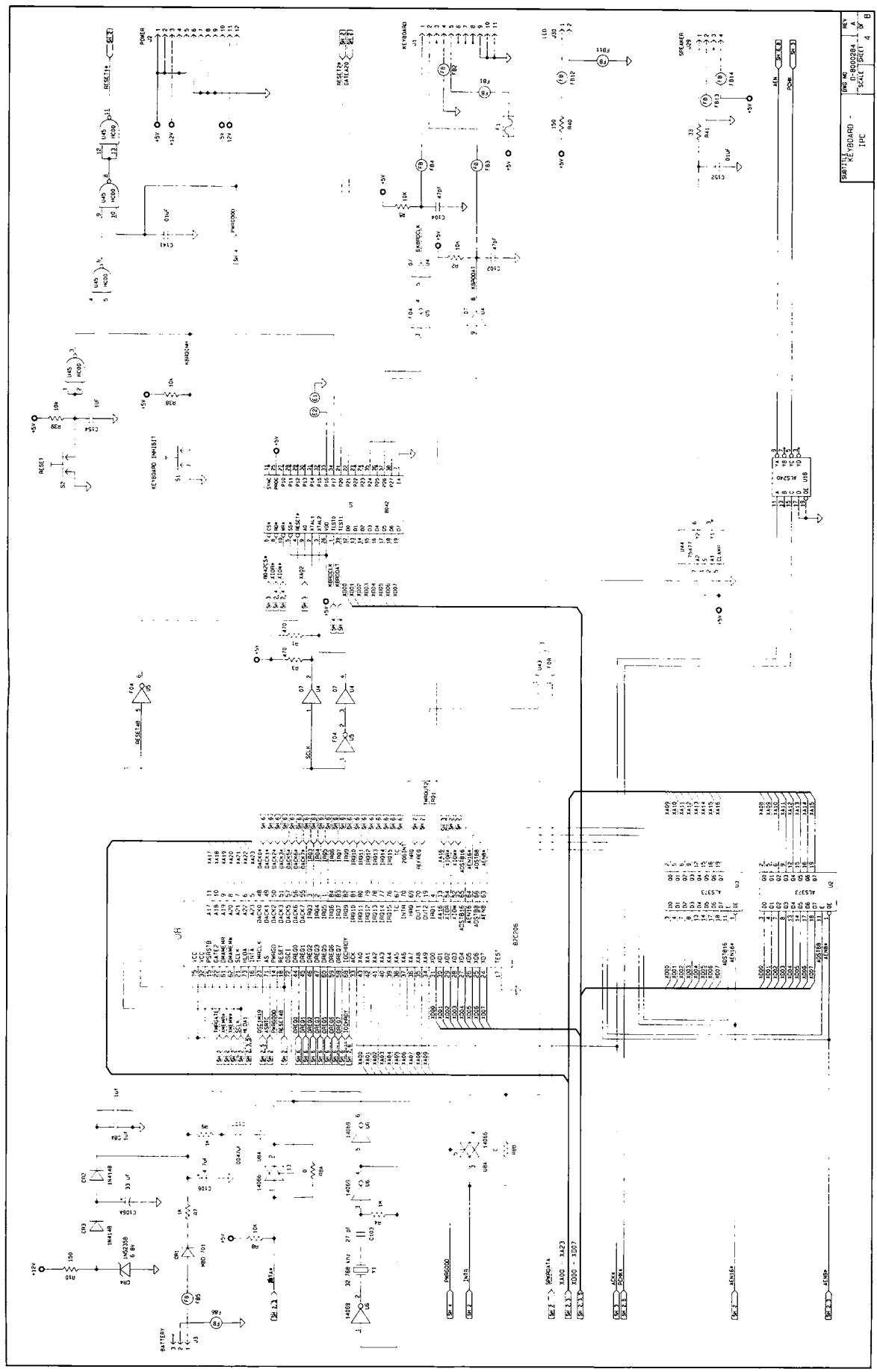




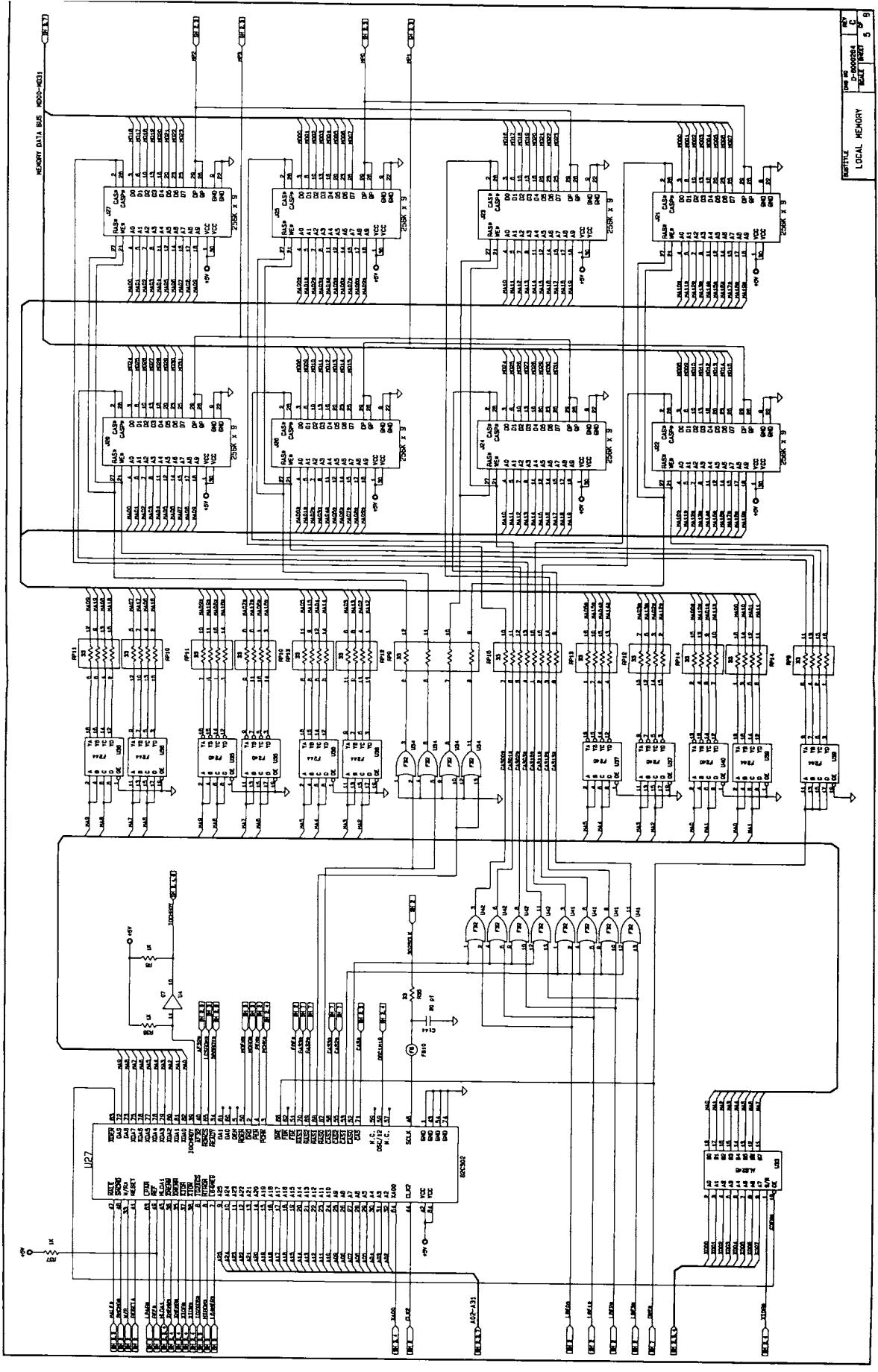




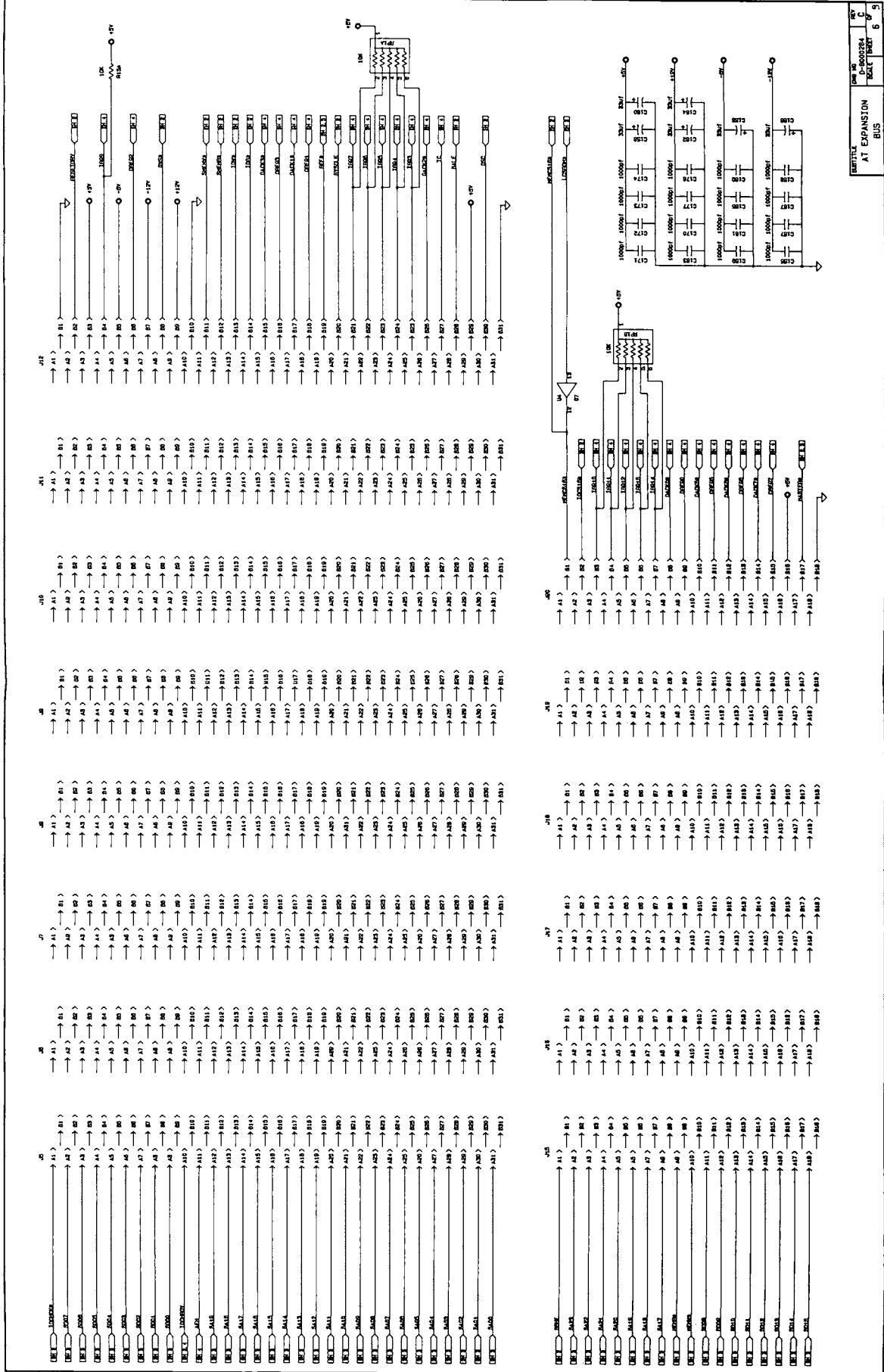




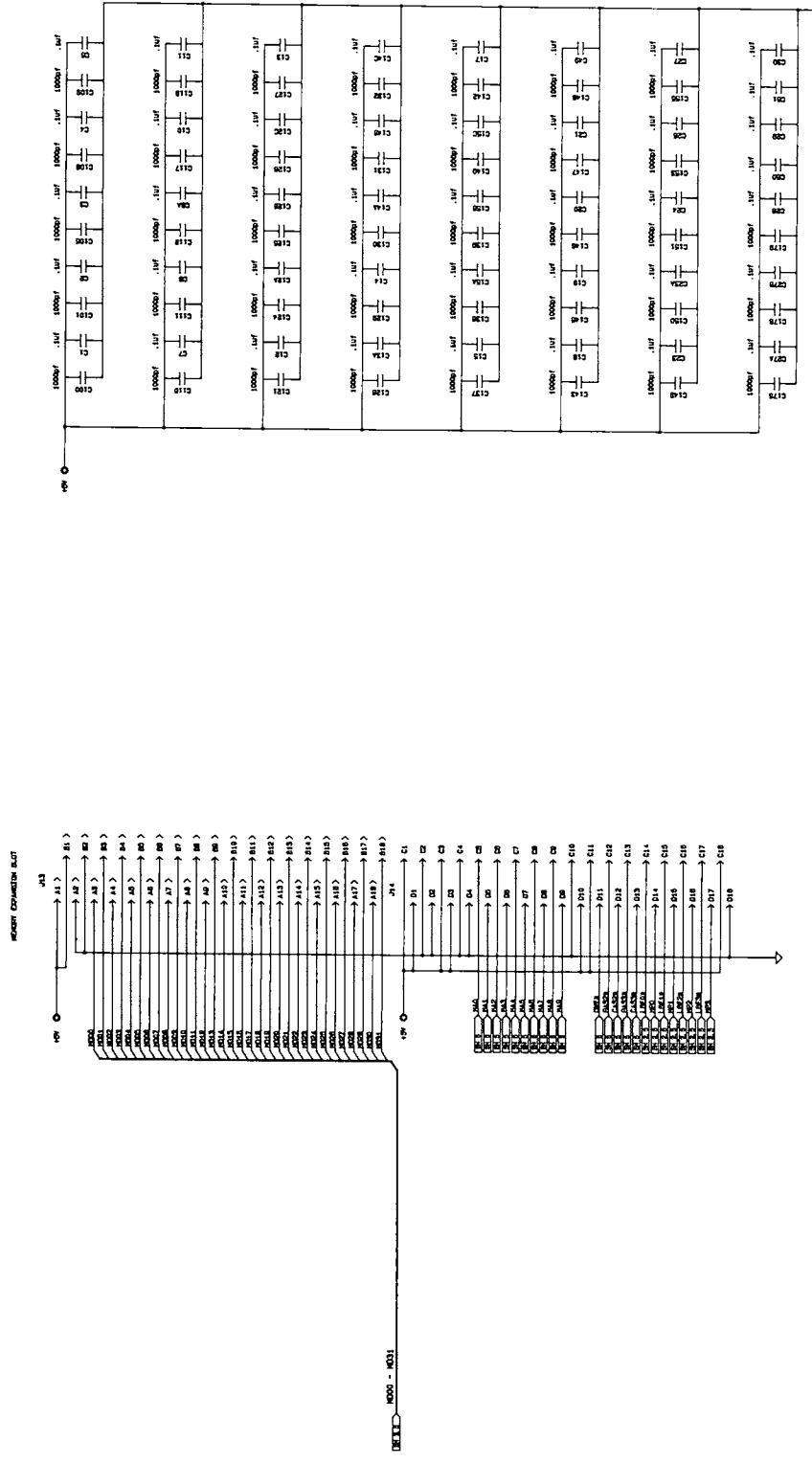






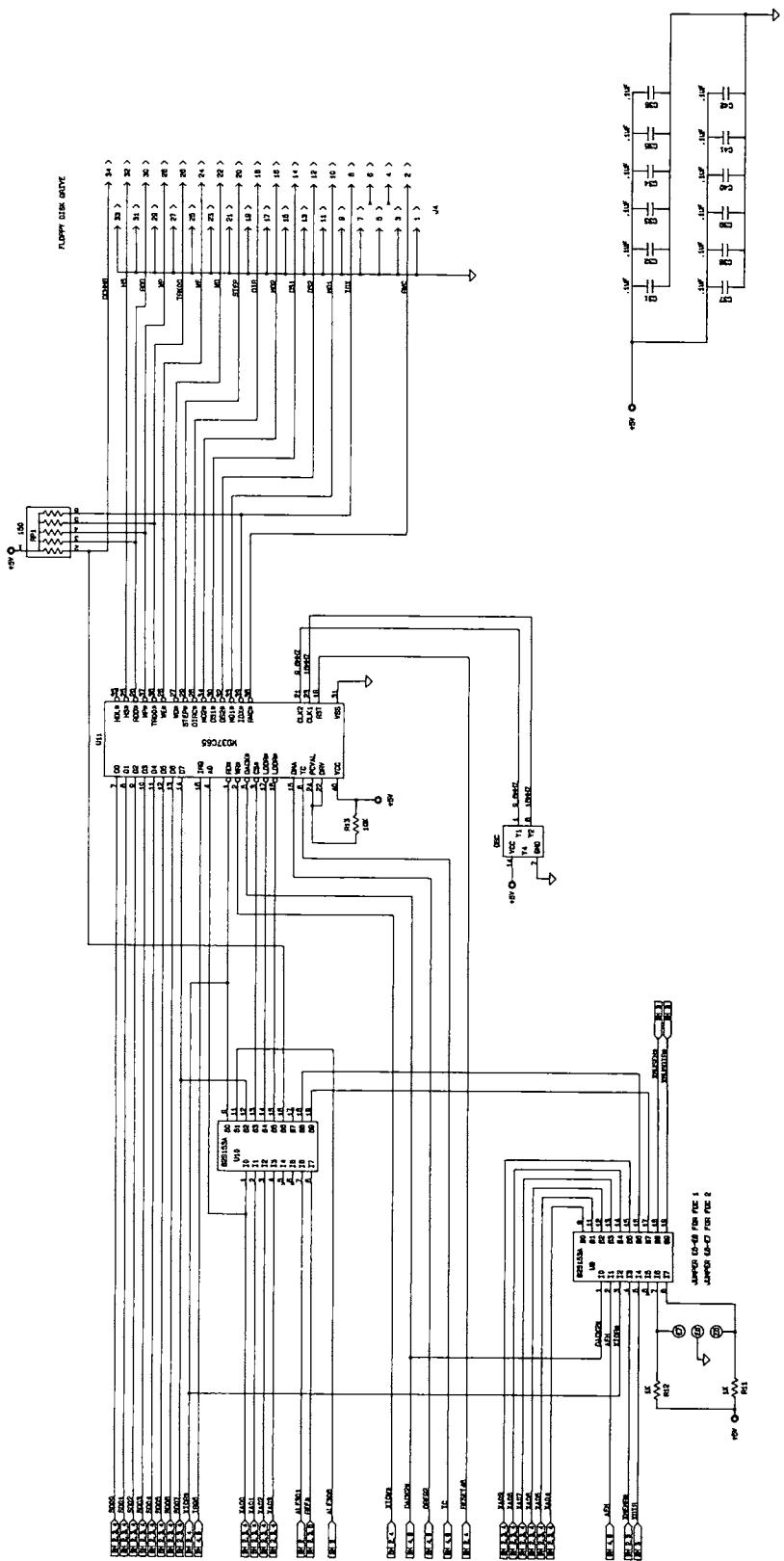






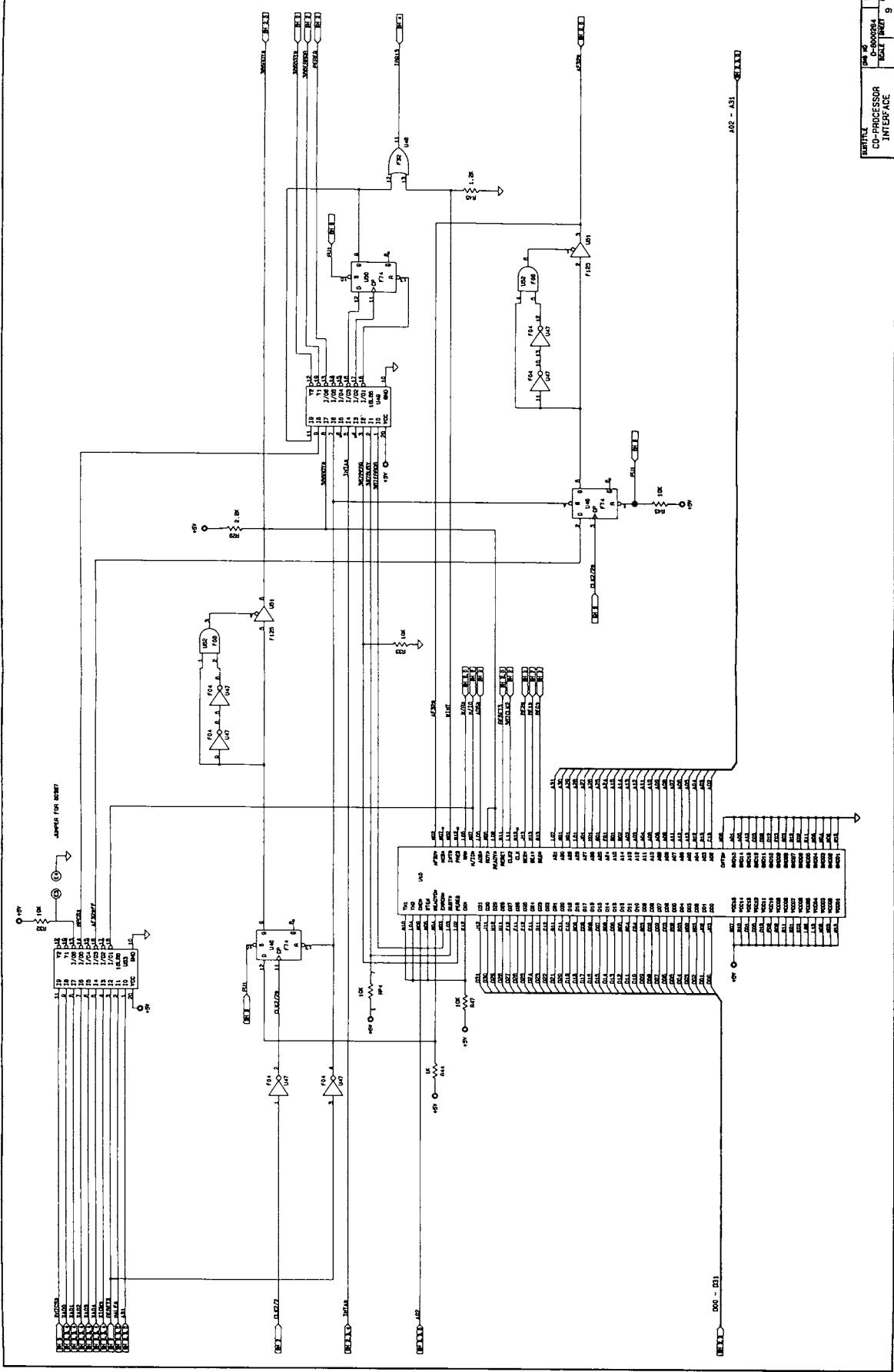
SEARCHED
INDEXED
SERIALIZED
FILED





प्राप्ति नं १८
फॉर्म नं १८
प्राप्ति नं १८
प्राप्ति नं १८







The Tandy 4000 Technical Reference Manual describes the computer hardware subunits and their interrelationships and the BIOS (Basic Input/Output Services).

This reference information is for hardware and software designers, engineers and programmers, and those with a need to understand the design and operation of the computer.

System Timing and Logic Diagrams and a Theory of Operation are provided. This manual also provides specifications, switch settings and jumper pin configurations, and other information (as appropriate) for the following hardware sections:

- Main Logic Board
- Devices
- Power Supplies
- Keyboards
- Disk Drives

The Software section contains:

- A Quick Reference List of Software Interrupts
(for all device I/O and system status services)
- Keyboard ASCII and Scan Codes
- The MS-DOS Memory Map

This information supplements the information found in:

- The prerequisite computer *Installation and Operation Guide*
(packaged with the computer)
- Tandy 4000 Service Manual*
- MS-DOS Reference Manual*
- Intel iAPX 386 Hardware Reference Manual*
- Intel iAPX 386 Programmer's Reference Manual*

Custom Manufactured in U.S.A. for Radio Shack, A Division of Tandy Corporation

