Personal System/2™

Model 80 Technical Reference

The following statement applies to the IBM products covered in this manual, unless otherwise specified herein. The statement for other IBM products will appear in their accompanying materials.

Federal Communications Commission (FCC) Statement

Warning: This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer when this computer is operated in a residential environment. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

CAUTION

This product is equipped with a 3-wire power cord and plug for the user's safety. Use this power cord in conjunction with a properly grounded electrical outlet to avoid electrical shock.

First Edition (April 1987)

Preface

This technical reference provides system-specific hardware and software interface information for the IBM Personal System/2™ Model 80 (8580-041 and -071). It is intended for developers who provide hardware and software products to operate with this IBM system.

You should understand the concepts of computer architecture and programming before using this publication.

This manual is divided into the following sections.

Section 1, "System Description" provides an overview of the Model 80. System board features, an I/O address map, and system unit specifications are provided.

Section 2, "Micro Channel™ Architecture" describes the system bus used in the Model 80. This section provides detailed descriptions of the Micro Channel signals, timing, and electrical characteristics. Other aspects of the Micro Channel architecture, such as Programmable Option Select, level-sensitive interrupts, and multidevice arbitration are discussed.

Adapter design information for the Model 80 is also in this section. Adapter dimensions, power requirements and design guidelines are provided.

Section 3, "System Board" discusses the operation of components on the Micro Channel. The system microprocessor, math coprocessor, interrupts, system timers, password, and the audio subsystem are discussed.

Section 4, "System Board I/O Controllers" describes the input and output interfaces of the system board. This includes the operation of the keyboard/auxiliary device controller, video subsystem, diskette drive controller, serial port controller, and

---

Personal System/2 and Micro Channel are trademarks of the International Business Machines Corporation.

parallel port controller. The different types of memory used by the system are also discussed.

Section 5, "Power Supply" provides electrical input/output specifications as well as theory of operation for the power supply.

Section 6, "Keyboard" contains layouts of the 101- and 102-key keyboards, as well as layouts of all of the keyboards that are used to support different languages. Keyboard scan code sets and keyboard specifications are provided.

Section 7, "Instruction Sets" provides a quick reference for the 80386 and 80387 assembly instruction sets.

Section 8, "Characters and Keystrokes" supplies the decimal and hexadecimal values for ASCII characters. All of the character codes are repeated in a set of two figures for quick reference.

Section 9, "Compatibility" discusses guidelines to help developers design programs, adapters, and attachments that are compatible with other IBM Personal System/2 products and IBM Personal Computer products.

A Glossary, Bibliography and Index are also provided.

Suggested Reading:

- IBM Personal System/2 Model 80 *Quick Reference*
- *BASIC for the IBM Personal Computer*
- IBM *Disk Operating System (DOS)*
- IBM Personal System/2 *Hardware Maintenance Service*
- IBM Personal System/2 *Hardware Maintenance Reference*
- *Macro Assembler for the IBM Personal Computer*
- *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Related publications are listed in the Bibliography.

Contents

Notes:

Figures

Section 1. System Description

Notes:

Description

The IBM Personal System/2 Model 80 (8580-041 and -071) is a self-contained, floor-standing computer system with a keyboard. The system can support two 3.5-inch diskette drives and two 5.25-inch fixed disk drives. Built into the system board is a keyboard port controller, an auxiliary device port controller, a serial port controller, a parallel port controller, and a video subsystem. The system features the 80386 microprocessor and Personal System/2 Micro Channel 16-/32-bit architecture. This new channel architecture supports adapters that are physically and electrically different from IBM Personal Computer adapters. A fixed disk adapter occupies one channel connector. Seven unoccupied channel connectors are provided for optional feature expansion. The keyboard is attached to the system unit by a coiled cable. The following shows the Model 80 system unit and keyboard.

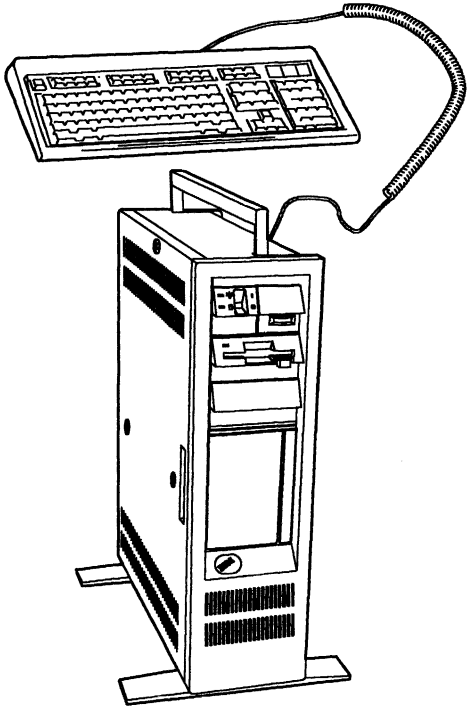


Figure 1-1. Model 80 System Unit and Keyboard

System Board Features

The Model 80 system board incorporates very large scale integration modules and surface-mount technology.

The following is a list of the system board features:

- Intel 80386 system microprocessor operating at 16 MHz
- Microprocessor support
  - Eight-channel direct memory access (DMA) controller
  - 16-level interrupt system
  - System clock
  - Three programmable timers
- ROM subsystem, 128K (K = 1024)
- RAM subsystem, 1M to 2M (M = 1,048,576)
- 16- and 32-bit channel
- CMOS RAM subsystem with battery backup
  - 64-byte CMOS RAM with real-time clock/calendar
  - 2K CMOS RAM extension
- Integrated video graphics subsystem with an auxiliary connector
- EIA RS-232-C serial communications controller and port
- Parallel port
- Audio subsystem with speaker
- Keyboard/Auxiliary device controller
- Keyboard connector
- Auxiliary device connector
- Password security
- Diskette drive controller
- Distributed arbitration mechanism with support for up to 15 devices
- Socket for the 80387 Math Coprocessor
- Eight Micro Channel connectors
  - Three 8-bit, 16-bit, or 32-bit channel connectors
  - Five 8-bit or 16-bit channel connectors
    - One with an auxiliary video connector
    - One occupied by the fixed disk adapter

The following shows the layout of the Model 80 system board.

1. Power supply connector
2. Diskette drive connector
3. System board memory connectors
4. Battery and speaker assembly connector
5. 80387 math coprocessor socket
6. 16-bit channel connector
7. 32-bit channel connector
8. Channel connectors
9. 16-bit channel connector with video extension
10. Display connector
11. Serial port connector
12. Parallel port connector
13. Auxiliary device connector
14. Keyboard connector

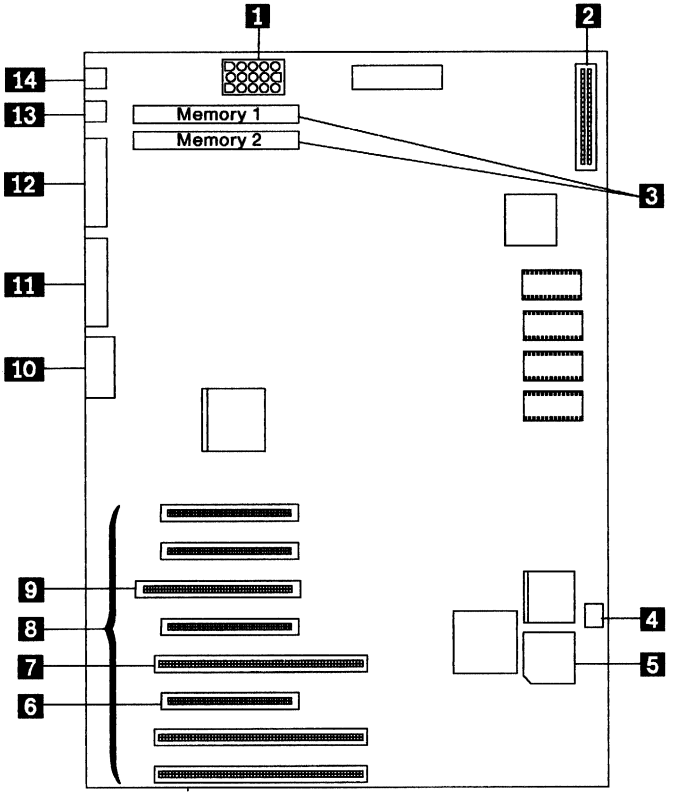


Figure 1-2. System Board Layout

The following is a block diagram of the system board.

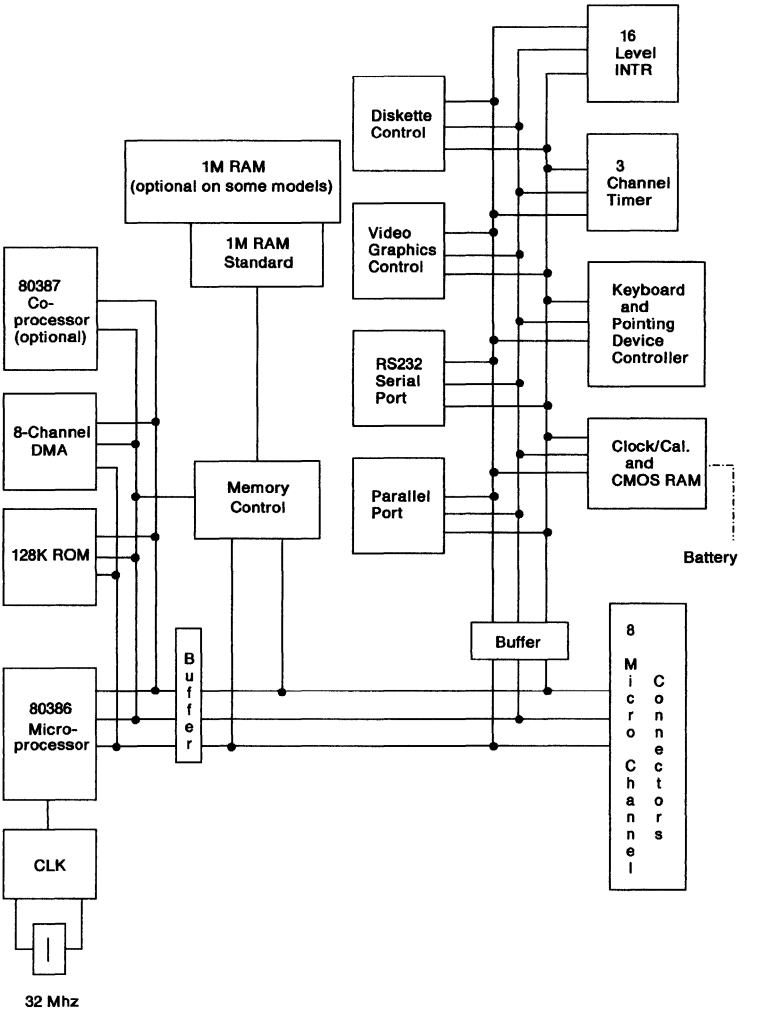


Figure 1-3. System Board Block Diagram

The Model 80 can accommodate up to three IBM 80386 Memory Expansion Options in the channel. These 32-bit memory adapters can provide up to 6M each for a total of 18M. The system board can provide up to 2M. This yields a total capacity of 20M. However, the DMA controller addressing is limited to 16M, therefore the total amount of system memory installed should not exceed the 16M limit.

System I/O Address Map

The following is the address map for the various system board I/O functions.

Hex Addresses	Device
0000 - 001F	DMA Controller
0020, 0021	Interrupt Controller 1, 8259A
0040, 0042, 0043, 0044, 0047	System Timers
0060	Keyboard, Auxiliary Device
0061	System Control Port B
0064	Keyboard, Auxiliary Device
0070, 0071	RT/CMOS and NMI Mask
0074, 0075, 0076	Reserved
0081, 0082, 0083, 0087	DMA Page Registers (0 - 3)
0089, 008A, 008B, 008F	DMA Page Registers (4 - 7)
0090	Central Arbitration Control Port
0091	Card Selected Feedback
0092	System Control Port A
0093	Reserved
0094	System Board Setup
0096, 0097	POS, Channel Connector Select
00A0 - 00A1	Interrupt Controller 2, 8259A
00C0 - 00DF	DMA Controller
00E0	Split Address Register
00E1	Memory Encoding Register
00F0 - 00FF	Math Coprocessor
0100 - 0107	Programmable Option Select
01F0 - 01F8	Fixed Disk Drive Adapter
0278 - 027B	Parallel Port 3
02F8 - 02FF	Serial Port 2 (RS-232-C)
0378 - 037B	Parallel Port 2
03BC - 03BF	Parallel Port 1
03B4, 03B5, 03BA, 03C0 - 03C5	Video Subsystem
03CE, 03CF, 03D4, 03D5, 03DA	Video Subsystem
03C6 - 03C9	Video DAC
03F0 - 03F7	Diskette Drive Controller
03F8 - 03FF	Serial Port 1 (RS-232-C)

Figure 1-4. System I/O Address Map

Specifications

The following are specifications for the Model 80 system unit.

Size

- Width: 165 millimeters (6.5 inches)
- Width (Feet extended): 318 millimeters (12.5 inches)
- Depth: 483 millimeters (19.0 inches)
- Height: 597 millimeters (23.5 inches)

Weight

- 20.6 kilograms (45.3 pounds) with one fixed disk drive

Cables

- Power Cable: 1.8 meters (6 feet)
- Keyboard Cable: 3.05 meters (10 feet)

Air Temperature

- System On: 15.6 to 32.2 degrees C (60 to 90 degrees F)
- System Off: 10.0 to 43.0 degrees C (50 to 110 degrees F)

Humidity

- System On: 8% to 80%
- System Off: 20% to 80%

Altitude

- Maximum Altitude: 2133.6 meters (7000 feet)

Heat Output

- 1245 BTU/hour

Acoustical

Readings from 1 meter (3.28 feet)

- 46 dB average, operating
- 40 dB average, idle

Electrical

Automatic Ranging

- Low Range
  - Minimum - 90 Vac
  - Maximum - 137 Vac
- High Range
  - Minimum - 180 Vac
  - Maximum - 265 Vac

Electro-Magnetic Compatibility

- FCC Class B

Notes:

Section 2. Micro Channel Architecture

Notes:

Description

The Micro Channel is assembled from an address bus, a data bus, a transfer control bus, an arbitration bus, and multiple support signals. The channel architecture uses asynchronous protocols for control and data transfer between memory, I/O devices, and the system microprocessor.

The following are characteristics of the Micro Channel.

- An I/O address width of 16 bits allows 8-, 16-, or 32-bit I/O transfers within a 64K range. A memory address width of 24 and 32 bits allow 8-, 16-, 24-, or 32-bit memory transfers within 16M- and 4G-byte ranges respectively. (K = 1024; M = 1,048,576; G = 1,073,741,824)
- Support of a central arbitration control point that allows up to 15 devices to arbitrate for control of the Micro Channel.
- A serial DMA protocol that supports eight DMA channels for 8- or 16-bit DMA transfers.
- Level sensitive interrupts with interrupt sharing on all levels.
- Programmable Option Select (POS) registers that replace hardware jumpers and switches. These registers allow high flexibility during system configuration at system power-on.
- Channel extension connectors that support the growth of additional channel features.
- Improved electro-magnetic compatibility
- Error reporting and recovery
- The Micro Channel Architecture does not support IBM Personal Computer adapters, and only supports adapters specifically designed for Micro Channel compatible systems.

Channel Definition

The channel provides all signal, power, and ground signals to the internal adapters.

The system board provides three types of channel connectors:

- 16-bit
- 16-bit with auxiliary video extension
- 32-bit.

The 16-bit channel has 77 signal lines, 29 power and ground lines, a separate audio ground line, 5 reserved lines, and 4 keyed positions in a dual 58-pin, 50-mil card edge connector. The 32-bit channel is an extended 16-bit channel designed to accommodate 32-bit addressing, 32-bit data transfers, and Matched Memory cycles. The 32-bit extension provides an additional 31 signal lines, 15 power and ground lines, and 16 reserved lines. The Matched Memory extension provides three signal lines, two ground lines, and three reserved lines. The 32-bit channel connector is a dual 93-pin, 50-mil card edge design. Every fourth pin on either side of each connector is at ac ground potential with side B offset from side A by 2 pins. This places each signal within 2.54 millimeters (0.1 inch) of a ground and minimizes current loop electromagnetic interference (EMI). The 50-mil channel connector reduces insertion force and matches surface mount technology line spacing. When designing adapters for the system, special design criteria must be considered. See "Adapter Design" on page 2-114.

The following is a diagram of the channel connectors.

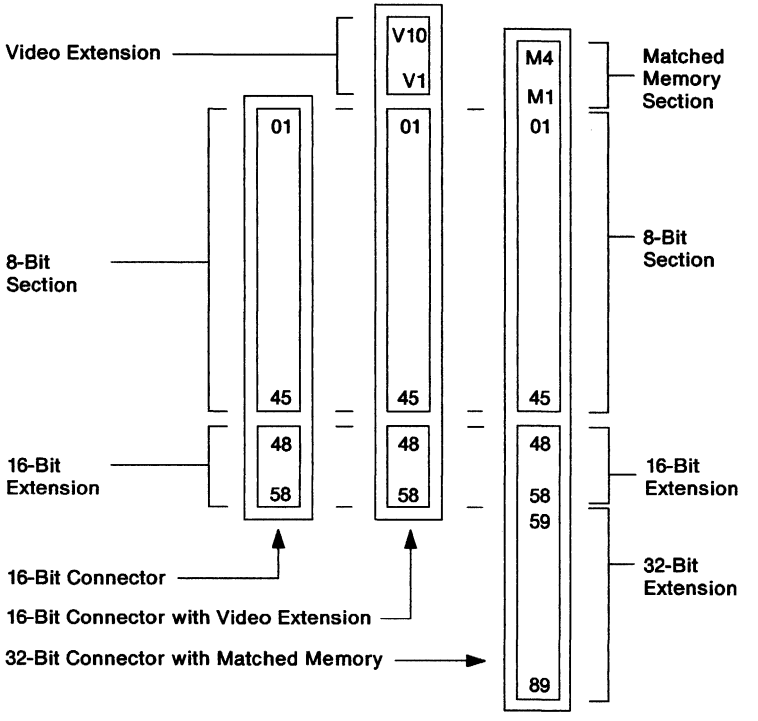


Figure 2-1. Micro Channel Connectors

Warning: Any signals shown or described as "Reserved" are not to be driven or received. These signals are reserved to allow compatibility with future implementations of the channel interface. Serious compatibility problems, loss of data, or permanent damage can result to features or the system if these signals are misused.

Signal Descriptions (16-Bit)

All of the logic signal lines are TTL-compatible. The following are the signals available on the channel.

A0 - A23: Address Bits 0 through 23: These lines are used to address memory and I/O slaves attached to the channel. A0 is the least-significant bit (LSB) while A23 is the most-significant bit (MSB). These 24 address lines allow access of up to 16 megabytes of memory. Only the lower 16 address lines (A0 - A15) are used for I/O operations and all 16 lines must be decoded by the I/O slave. A0 through A23 are generated by the system microprocessor. Valid addresses generated by the system microprocessor are unlatched on the channel and, if required, may be latched by the slaves using either the trailing edge of the '-address decode latch' (-ADL) signal or the leading edge of the 'command' (-CMD) signal. A0 through A23 must be driven with tri-state drivers.

D0 - D15: Data Bits 0 through 15: These lines provide data bus bits 0 to 7 (low byte) and 8 to 15 (high byte) for the system microprocessor and slaves. D0 is the LSB and D15 is the MSB. All 8-bit slaves on the channel must use D0 through D7 for communication with the system microprocessor. During read cycles, data is valid on these lines after the leading edge but before the trailing edge of -CMD, and must remain valid until after the trailing edge of -CMD. However, during write cycles, data is valid throughout the period when -CMD is active. D0 through D15 must be driven with tri-state drivers.

-ADL: -Address Decode Latch: This line, driven by the system microprocessor, is provided as a convenient mechanism for the slave to latch valid addresses and status bits. It is recommended that slaves use transparent latches to latch information, if required, with the trailing edge of -ADL or the leading edge of -CMD. -ADL is not active during Matched Memory cycles. -ADL is driven with a tri-state driver.

-CD DS 16 (n): -Card Data Size 16: This line is driven by 16-bit and 32-bit memory, I/O, or DMA slaves to provide an indication on the channel of a 16-bit or 32-bit data port at the location addressed. The signal is unlatched and derived as a valid address decode. All system logic receives this signal to support communication with 16-bit slaves. -CD DS 16 is not driven by an 8-bit memory, I/O, or DMA slave. The (n) indicates this signal line is unique to each channel

connector (one independent signal line per connector). -CD DS 16 is driven with a totem-pole driver.

-DS 16 RTN: -Data Size 16 Return: This output signal is a negative OR of the -CD DS 16 signals from each channel connector. If any device drives its -CD DS 16 active then this output is active. This signal is provided to allow channel resident bus masters (a device that arbitrates for, and controls the channel) to monitor the data size information. -DS 16 RTN must be driven with a bus driver.

-SBHE: -System Byte High Enable: This line indicates and enables transfer of data on the high byte of the data bus (D8 - D15), and is used with A0 to distinguish between high byte (D8 - D15) and low byte (D0 - D7) transfers. All 16-bit slaves decode this line but 8-bit slaves do not. -SBHE is driven with a tri-state driver.

MADE 24: Memory Address Enable 24: This line indicates when an extended address is used on the bus. If a memory cycle is in progress, MADE 24 inactive indicates an extended address space greater than 16M is being presented; MADE 24 active indicates an unextended address space less than or equal to 16M is being presented. This line is driven by the system microprocessor and decoded by all memory slaves, regardless of their address space size.

M/-IO: Memory/-Input Output: This signal distinguishes a memory cycle from an I/O cycle. When this signal is high, it indicates a memory cycle is in progress. When M/-IO is low, it indicates that an I/O cycle is in progress. M/-IO is driven with a tri-state driver.

-S0, -S1: -Status Bits 0 and 1: These lines indicate the start of a channel cycle and also define the type of channel cycle. When used with M/-IO, memory read/write operations are distinguished from I/O read/write operations. When M/-IO, -S0, and -S1 are all high, they do not signify a status cycle. These signals are latched by the slave, as required, using the leading edge of -CMD or the trailing edge of -ADL. -S0 and -S1 are driven with a tri-state driver.

Decoded commands are generated with latched status lines (-S0 and -S1), M/-IO, and -CMD.

Slaves must support a full decode of -S0 and -S1. The following figure shows the proper states of M/-IO, -S0, and -S1 in decoding I/O and memory read/write commands.

M/-IO	-S0	-S1	Function
0	0	0	Reserved A
0	0	1	IO Write
0	1	0	IO Read
0	1	1	Reserved B
1	0	0	Reserved C
1	0	1	Memory Write
1	1	0	Memory Read
1	1	1	Reserved D

Figure 2-2. I/O and Memory Transfer Controls

A decoded I/O Write command instructs an I/O slave to store the data on the data bus. The data must be valid on the bus from the leading edge of -CMD and must be held on the bus until after -CMD goes inactive. Addresses on the bus must be valid prior to -S0 going active.

A decoded I/O Read command instructs an I/O slave to drive its data onto the data bus. The data must be placed on the bus following the leading edge of -CMD, must be valid before the trailing edge of -CMD and must be held on the bus until -CMD goes inactive. Addresses on the bus must be valid prior to -S1 going active.

A decoded memory Write command instructs the memory to read the data on the data bus. The data must be valid on the bus from the leading edge of -CMD and must be held on the bus until after -CMD goes inactive. Addresses on the bus must be valid prior to -S0 going active.

A decoded memory Read command instructs the memory to drive its data onto the data bus. The data must be placed on the bus following the leading edge of -CMD. The data must be valid before the trailing edge of -CMD, and must be held on the bus until -CMD goes inactive. Addresses on the bus must be valid prior to -S1 going active.

-CMD: -Command: This signal is used to define when data is valid on the data bus. The trailing edge of this signal indicates the end of the bus cycle. This signal provides to the slave the indication of the duration during which the data is valid on the bus. During write operations, the data is valid on the bus throughout the period that -CMD is active. During read operations, the data is valid on the bus following the leading edge but before the trailing edge of -CMD and must be held on the bus until after the -CMD goes inactive. This signal can be used by the slaves to latch the address on the bus. Latched status lines gated by -CMD provide the timing control of valid data. It is recommended that slaves use transparent latches to latch address and status information with the leading edge of -CMD. -CMD is not active during Matched Memory cycles. -CMD must be driven with a tri-state driver.

-CD SFDBK (n): -Card Selected Feedback: When a memory slave or an I/O slave is addressed by the system microprocessor, this signal is driven active by the addressed slave, as a positive acknowledgment of its presence at the address specified. This signal can be used during diagnostics and installation to detect address space conflicts and/or defective field replaceable units or customer replaceable units. This signal is unlatched and driven by all slaves with a valid select decode. This signal is driven by a slave selected by any select mechanism except -CD SETUP. The slave does not drive -CD SFDBK during the configuration cycle.

Note: Memory supporting diagnostic software must not drive -CD SFDBK during the diagnostic operation.

The (n) indicates this signal line is unique to each channel connector (one independent signal line per connector). -CD SFDBK is driven with a totem-pole driver.

CD CHRDY (n): Channel Ready: This line, normally active (ready), is pulled inactive (not ready) by a memory or I/O slave to allow additional time to complete a channel operation. When using this line during a read operation, a slave promises that data will be valid on the data bus within the time specified after releasing the line to a ready state. The slave also holds the data for a sufficient amount of time for the system microprocessor to sample. A slave may also use this line during a write operation if more time is needed to store the data from the bus. The maximum time that CD CHRDY may be held inactive by a slave will not exceed 3.5 microseconds. The (n)

indicates this signal line is unique to each channel connector (one independent signal line per connector). This signal is derived with a valid address decode ANDed with Status. CD CHRDY is driven with a totem-pole driver.

CHRDYRTN: Channel Ready Return: This output signal is a positive AND of the CD CHRDY signals from each channel connector. If all devices drive CD CHRDY active then this output is active. It is provided to allow channel resident masters to monitor the ready information. CHRDYRTN must be driven with a bus driver.

ARB0 - ARB3: Arbitration Bus Priority Levels: These lines comprise the arbitration bus and are used to present arbitrating bus participant priority levels. ARB0 through ARB3, the least-significant and most-significant bits respectively, support up to 16 priority levels for arbitration by the bus participants.

The highest hexadecimal value of the arbitration bus (hex F) has the lowest priority, and the lowest value (hex 0) has the highest priority. The arbitrating bus participant is allowed to change the state of the arbitration bus only immediately after the rising edge of ARB/-GNT. All arbitrating bus participants monitor the arbitration bus and the lower priority participants withdraw their priority levels by not activating less-significant arbitration bits.

The hexadecimal code of the highest priority requester is valid on the arbitration bus after a settling time. After the channel is granted, the highest priority participant continues to drive its priority lines. These bidirectional lines are active high and must be driven with open collector drivers.

ARB/-GNT: Arbitrate/-Grant: When high, this signal indicates that an arbitration cycle is in process. When low, this signal is the acknowledgement from the central arbitration control point (CACP) to the winning arbiter that channel control has been awarded. This signal is driven high by the CACP within a specified time after -S0, -S1, -BURST, and -CMD become inactive. The negative to positive transition of ARB/-GNT initiates an arbitration cycle and the positive to negative transition of ARB/-GNT terminates this arbitration cycle. Only the CACP activates and deactivates this line. This signal must be used by all arbitrating devices to gate their address data and transfer control bus drivers off during arbitration cycles. ARB/-GNT is driven with a bus driver.

-PREEMPT: -Preempt: This signal is used by arbitrating bus participants (DMA slaves and intelligent bus controllers) to request usage of the channel through arbitration. Any arbitration bus participant with a channel request activates -PREEMPT and causes an arbitration cycle to occur. A requesting arbitration bus participant removes its preempt upon being granted the channel. This bidirectional line must be driven with an open collector driver.

-BURST: -Burst: This signal is used by arbitrating bus participants to indicate to the CACP the extended use of the channel when transferring a block of data. This type of data transfer is referred to as a *burst cycle*. This line is shared by all arbitrating bus participants. This signal is driven active by the arbitrating bus participant after being granted the channel. The participant must deactivate -BURST during or by the end of the last transfer cycle. -BURST must be driven with an open collector driver.

-TC: -Terminal Count: This line provides a pulse on the channel during a Read or Write command to indicate that the terminal count of the current DMA channel has been reached. This indicates to the DMA slave the last cycle to be performed of a pre-programmed DMA block transfer. -TC is available on the channel only during DMA operations. -TC is driven with a tri-state driver by the DMA Controller.

-IRQ 3 – 7, -IRQ 9 – 12, and -IRQ 14 – 15: -Interrupt Request: These lines are used to signal the system microprocessor that an I/O slave requires attention. They are prioritized with -IRQ 9 having the highest priority and -IRQ 7 having the lowest priority. The effective interrupt priority sequence is -IRQ (9-12, 14, 15, 3-7). An interrupt request is generated by the requesting interrupting slave driving one of the 'interrupt request' signals low. The polarity of 'interrupt request' signals makes it amenable for multiple slaves to share the same interrupt level. This is referred to as *interrupt sharing*. These lines must be driven with an open collector driver.

-CD SETUP (n): -Card Setup: This signal is driven by system board logic to individually select channel connectors during system configuration and error recovery procedures. When this signal is activated, a specific channel connector is selected and the configuration data space of the adapter is accessed. The card ID and the configuration data of the adapter can be obtained by an I/O read operation with -CD SETUP active. The configuration data is stored by the adapter as an I/O write operation with -CD SETUP active. Each

slot has a unique -CD SETUP. The (n) indicates this signal line is unique to each channel connector (one independent signal line per connector). This line is driven with a totem-pole driver.

-CHCK: -Channel Check: This line is used to indicate a serious error (such as a parity error) which threatens continued operation of the system. -CHCK is driven active to indicate the error condition and must remain low until the -CHCK interrupt handler resets it. -CHCK is driven with an open collector driver to allow sharing.

AUDIO: Audio Sum Node: This line on the channel is an audio voltage sum node. This line is used to drive audio signals from an adapter to the system audio output or to transfer audio signals between adapters. The frequency response of the audio line is 50 Hz to 10 kHz $\pm$ 3 dB. The maximum signal amplitude is 2.5 volts peak to peak, at a dc offset of 0 $\pm$ 50 millivolts. The noise level is limited to a maximum of 50 millivolts peak to peak.

AUDIO GND: Audio Ground: This is a separate ground return for the audio system.

OSC: Oscillator: This line is a high speed clock with a frequency of 14.31818 MHz $\pm$ 0.01%. The high level (more than 2.3 volts) pulse width and the low level (less than 0.8 volt) pulse width must not be less than 20 nanoseconds each.

CHRESET: Channel Reset: This signal is generated by the system board logic to reset or initialize all adapters upon power on or during a low line voltage condition. During a power-on sequence, CHRESET is active for a specified minimum time. In addition, the system can activate this signal under program control. CHRESET is driven with a bus driver.

-REFRESH: -Refresh: This line is driven by the system board logic and is used to indicate that a memory refresh operation is in progress. While this line is active, a memory read operation occurs. The address lines contain the memory locations being refreshed. Nine lines, A0 - A8, are activated. -REFRESH timing may be inconsistent and must not be used as a timing mechanism.

Signal Descriptions (32-Bit)

A24 – A31: Address Bits 24 through 31: These lines are used with A0 through A23 to address memory attached to the channel. A0 is the LSB and A31 is the MSB. These 32 address lines allow access of up to 4G of memory. Only the lower 16 address lines (A0 through A15) are used for I/O operations. A24 through A31 are generated by the system microprocessor. Valid addresses generated by the system microprocessor are unlatched on the bus and, if required, must be latched by the slaves using either the trailing edge of -ADL or the leading edge of -CMD. A0 through A31 are driven with tri-state drivers.

-BE0 – -BE3: -Byte Enable 0 through 3: These lines are used during data transfers involving 32-bit slaves to indicate which data bytes will be placed on the bus. Data transfers of 8-, 16-, 24-, or 32-contiguous bits are controlled by -BE0 through -BE3 during transfers involving 32-bit slaves only. These lines are driven by the system microprocessor or other master when TR 32 is inactive, and by the Central Translator logic (for those operations involving a 16-bit master with a 32-bit slave) when TR 32 is active. These lines are unlatched on the bus and, if required, must be latched by 32-bit slaves. -BE0 through -BE3 are driven with tri-state drivers.

D16 – D31: Data Bits 16 through 31: These lines are used with D0 through D15 to provide data bus bits to the system microprocessor, intelligent bus controller, and slaves. D0 is the LSB and D31 is the MSB. All 32-bit transfers from the bus master to 8-bit slaves are converted to four 8-bit transfers, all transmitted on lines D0 through D7. During read cycles, data is valid on these lines after the leading edge of -CMD but before the trailing edge of -CMD. However, during write cycles, data is valid throughout the period when -CMD is active. D0 through D31 must be driven with tri-state drivers.

-CD DS 32 (n): -Card Data Size 32: This line is driven by 32-bit memory, I/O, or DMA to provide an indication on the bus of a 32-bit data port at the location addressed. -CD DS 32(n) is unlatched and derived from a valid address decode. All 32-bit slaves must drive this signal. -CD DS 32(n) is inactive for an 8- or 16-bit data port of 32-bit slaves. The (n) indicates this signal line is specific to a channel connector position (one independent signal per connector). -CD DS 32(n) must be driven with a totem-pole driver.

Note: The DMA controller used in this system does not support 32-bit DMA slaves.

-DS 32 RTN: -Data Size 32 Return: This output signal is a negative OR of the -CD DS 32 signals from each channel connector. If any device drives its -CD DS 32 active then this output is active. This signal is provided to allow channel resident masters to monitor the data size information. -DS 32 RTN is be driven with a bus driver.

TR 32: Translate 32: This line is driven inactive by 32-bit masters and received by the Central Translator Logic. TR 32 can also be received by any 32-bit slave. When TR 32 is inactive, the 32-bit master drives BE0 through BE3. When TR 32 is active, the Central Translator Logic drives BE0 through BE3. TR 32 must be driven by a tri-state driver.

Signal Descriptions (Matched Memory)

The following are descriptions of the Matched Memory signals.

-MMC: -Matched Memory Cycle: This signal is driven by the system board logic to indicate to the bus channel slaves that the system microprocessor is controlling the bus and is able to run a Matched Memory bus cycle. -MMC is driven by a tri-state driver.

-MMCR: -Matched Memory Cycle Request: This is a bus cycle control input signal. -MMCR is driven by a 16- or 32-bit channel slave to request the faster cycle available on the system bus. -MMCR is driven active from an address decode and M/-IO. This signal is wired separately for three channel connectors on the system board and ORed by the system logic. The 80386 microprocessor is the only controlling device that can run Matched Memory cycles. If -MMCR is asserted active by an 8-bit channel slave, or by a 16- or 32-bit channel slave during a DMA bus cycle or other nonmicroprocessor bus cycle, a Basic Transfer cycle is run and -MMCR is not honored. CD CHRDY is used to extend the Matched Memory cycle if required. -ADL and -CMD remain inactive for the entire Matched Memory bus cycle.

-MMC CMD: -Matched Memory Cycle Command: This signal is an output to the bus and is generated for the 80386 microprocessor bus cycles only. -MMC CMD is used to define when data is valid on the bus during a Matched Memory bus cycle. The trailing edge of the signal indicates the end of the Matched Memory bus cycle. As with -CMD, this signal provides the slave with an indication of the duration during which the data is valid on the bus, and when to either latch the data from the bus or stop driving the data onto the bus.

Matched Memory Cycle

This section describes the Model 80 80386 *Matched Memory cycle* bus operation supported by the Micro Channel. The Model 80 implementation is such that a channel slave can be either a memory or I/O slave with a data bus width of 16 or 32 bits. No 8-bit devices are allowed to run Matched Memory cycles.

The Matched Memory cycle is a system-unique function supported by the Micro Channel, and is provided for memory subsystem expansion. This function allows the most efficient data transfer capability between the 80386 and channel memory. The system board ROM and RAM (held on the 80386 local bus) and the 32-bit memory expansion adapter (held on the channel) adhere to the Matched Memory cycle protocol described in this section.

The Model 80 supports two types of 80386 bus cycles on the channel; Matched Memory bus cycles and Basic Transfer bus cycles. Matched Memory bus cycles are a minimum of 3 (16 MHz) clocks or 187-nanosecond cycle time. Basic Transfer bus cycles are a minimum of 4 (16 MHz) clocks or 250-nanoseconds cycle time. The target channel slave must request the Matched Memory cycle on a bus-cycle by bus-cycle basis, through the -MMCR (Matched Memory Cycle Request) signal or it will get the default Basic Transfer cycle.

An 80386 bus cycle begins by the 'address', 'byte enables', and bus definition signals becoming valid in the first clock cycle of the bus cycle. Either of the status bits driven active, signals the start of the channel bus cycle. The status state (Ts) is the first clock cycle. After Ts, the command state Tc is entered. Channel slaves respond to the bus operation during Tc, either transferring read data to or accepting write data from the system bus. Tc states may be repeated as often as necessary to assure sufficient time for the channel slave to

respond. CD CHRDY is used to determine whether Tc is repeated. A repeated Tc state is called a 62.5-nanosecond wait state.

If -MMCR is asserted active by the channel slave, then -MMC CMD is driven by the system microprocessor.

If -MMCR is not returned from the channel slave, a Basic Transfer bus cycle is run. Basic Transfer and Matched Memory bus cycles can be mixed in any manner because the mechanism is completely dynamic on a bus cycle basis. Bus cycles can be extended until CD CHRDY is sampled as active.

CD CHRDY is asserted inactive by a memory or channel slave to allow additional time to complete a Basic Transfer or Matched Memory bus cycle when the default cycle length is not sufficient to service that device. See "Matched Memory Bus Timings" on page 2-110 for the timing diagrams related to the 80386 Matched Memory cycle.

Signal Descriptions (Auxiliary Video Extension)

The following are signal descriptions for the Auxiliary Video extension of the channel connector.

VSYNC: This signal is the vertical sync signal used to drive the display. See also the ESYNC signal description.

HSYNC: This signal is the horizontal sync signal used to drive the display. See also the ESYNC signal description.

BLANK: This signal is connected to the BLANK input of the video Digital to Analog Converter (DAC). When active (0 volts), this signal tells the DAC to drive its analog color outputs to 0 volts. See also the ESYNC signal description.

P7 - P0: These eight signals contain digital video information and comprise the picture element (PEL) address inputs to the video DAC. See also the EVIDEO signal description.

DCLK: This signal is the video PEL clock that is used by the video DAC to latch the digital video signals, P7 through P0. P7 through P0 are latched on the rising edge of DCLK inside the DAC.

This signal is also connected to the EXTCLK input of the Video Graphics Array (VGA), and may be driven by the auxiliary video connector and used as the input clock to the VGA.

Note: When this configuration is used, the VGA may not be the source of the digital video signals presented to the DAC (P7 - P0); rather, P7 through P0 must be driven from the auxiliary video connector. See also the EDCLK signal description.

ESYNC: This signal is the output enable signal for the buffer that drives the BLANK, VSYNC, and HSYNC signals. ESYNC is tied to 5V through a pull-up resistor so that an open circuit on the ESYNC pin produces 5V.

When ESYNC equals 5V, BLANK, VSYNC, and HSYNC are sourced from the VGA BLANK, VSYNC, and HSYNC outputs respectively. When ESYNC equals 0V, BLANK, VSYNC, and HSYNC are driven from the auxiliary video connector.

EVIDEO: This signal is the output enable signal for the buffer that drives P7 through P0. EVIDEO is tied to 5V through a pull-up resistor so that an open circuit on the EVIDEO pin produces 5V.

When the EVIDEO signal equals 5V, P7 through P0 are sourced from the VGA outputs, P7 through P0. When the EVIDEO signal equals 0V, P7 through P0 are driven from the auxiliary video connector.

EDCLK: This signal is the output enable signal for the buffer that drives DCLK. EDCLK is tied to 5V through a pull-up resistor so that an open circuit on the EDCLK pin produces 5V.

When EDCLK equals 5V, DCLK is sourced from the VGA DCLK output, and is received by the auxiliary video connector and DAC. The VGA Miscellaneous Output register should be configured so that it is not selecting clock source 2 when EDCLK equals 5V.

When EDCLK equals 0V, DCLK is driven from the auxiliary video connector to the EXTCLK input of the VGA chip and to the DAC.

Note: When this configuration is used, the VGA may not be the source of the digital video signals presented to the DAC (P7 - P0); rather, P7 through P0 must be driven from the auxiliary video connector. The Miscellaneous Output register (see "Miscellaneous Output Register" on page 4-59) must be programmed to select clock source 2.

Micro Channel Connector (16-Bit)

The 16-bit Micro Channel connector is organized into:

- An 8-bit section
- A 16-bit extension.

A key is provided between the 8-bit section and 16-bit extension for mechanical alignment.

Note: The auxiliary video extension, 32-bit extension, and Matched Memory extension are shown separately beginning on page 2-23.

The following figures show the signals and voltages assigned to the 16-bit channel connector.

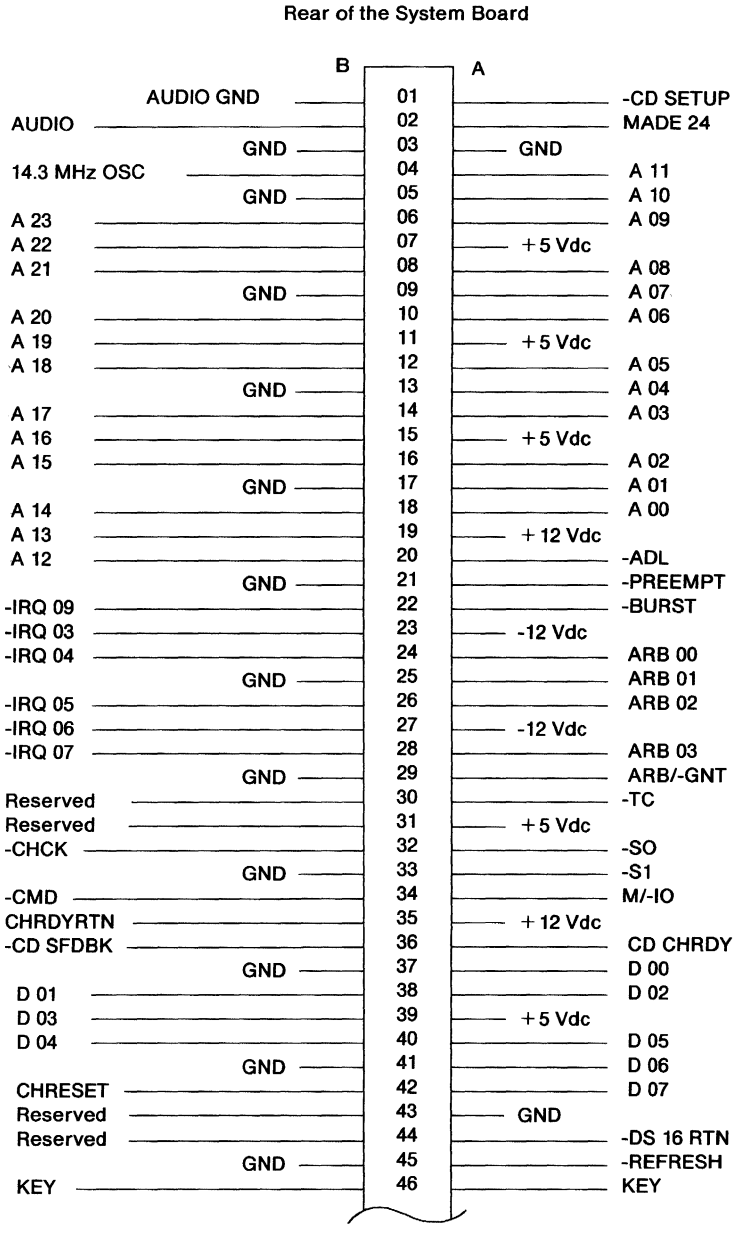


Figure 2-3. Channel Connector Voltage and Signal Assignments (8-Bit Section)

The following are the signals and voltages assigned to the channel connector 16-bit extension.

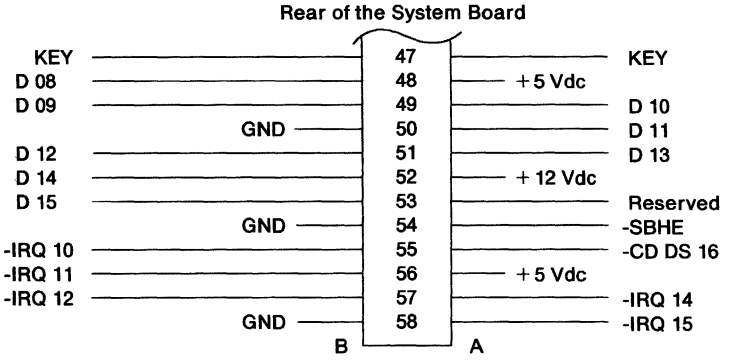


Figure 2-4. Micro Channel Connector 16-Bit Extension Voltage and Signal Assignments

Micro Channel Connector (Auxiliary Video Extension)

This connector extends the 16-bit Micro Channel connector to accommodate video adapters that interface with the system board video subsystem.

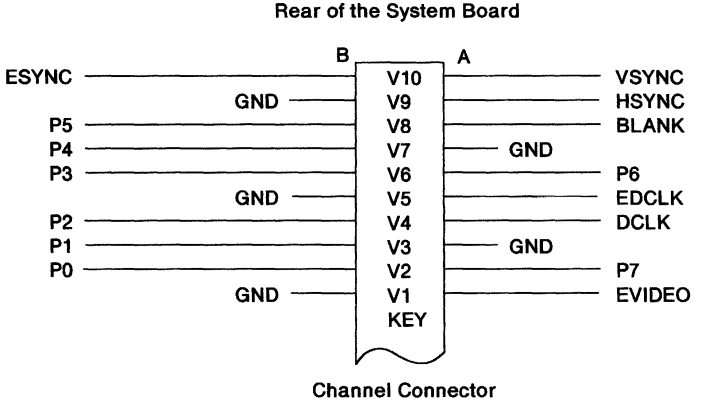


Figure 2-5. Auxiliary Video Connector

Micro Channel Connector (32-Bit Extension)

This connector extends the 16-bit Micro Channel connector to accommodate 32-bit addressing and 32-bit data transfers.

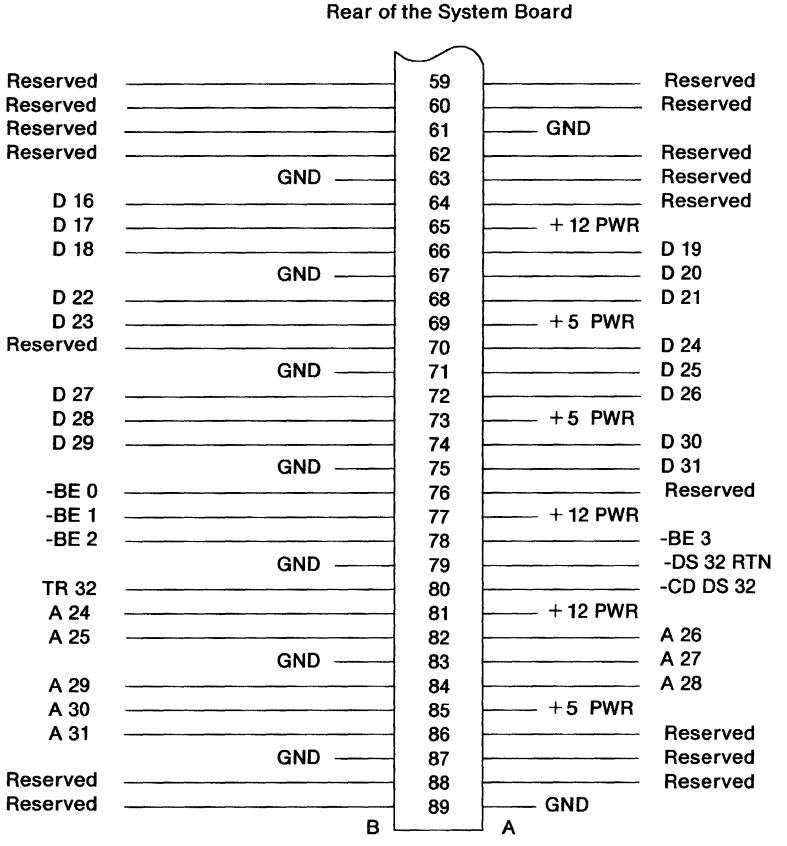


Figure 2-6. Micro Channel Connector 32-Bit Extension Voltage and Signal Assignments

Micro Channel Connector (Matched Memory Extension)

This connector extends the 32-bit Micro Channel connector to accommodate Matched Memory cycles.

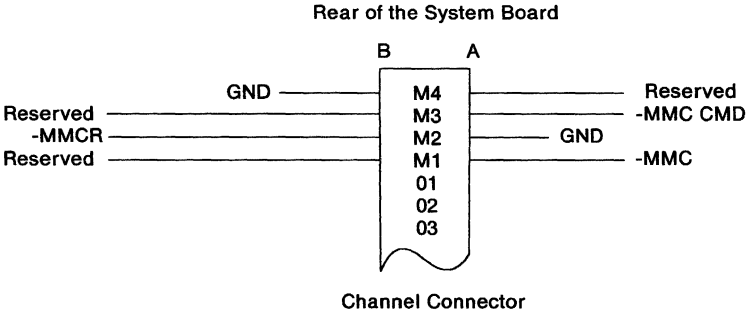


Figure 2-7. Micro Channel Connector Matched Memory Extension Voltage and Signal Assignments

Channel Signal Groups (16-Bit, 32-Bit, and Matched Memory)

The following figure lists 16-bit, 32-bit, and Matched Memory Micro Channel signals and shows what type of driver or receiver is required to be compatible with the system board.

Signal Name	Sys Logic	DMA Cntlr	Intelligent Bus Cntlr	DMA Slave	MEM Slave	I/O Slave	Driver Type
	D/R	D/R	D/R	D/R	D/R	D/R	(Signal Group)
A(0-23)	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
D(0-15)	D/R	D/R	D/R	D/R	D/R	D/R	TS (2)
-ADL	D/-	D/-	D/-	-/O	-/O	-/O	TS (1)
-CD DS 16 (n)	-/R	-/R	-/R	#/-	#/-	#/-	TP (3)
-DS 16 RTN	D/-	-/R	-/R	-/-	-/-	-/-	BD (4)
-SBHE	D/-	D/-	D/-	-/#	-/#	-/#	TS (1)
MADE 24	D/-	D/-	D/-	-/-	-/R	-/-	TS (1)
M/-IO	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
-S0,-S1	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
-CMD	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
-CD SFDBK (n)	-/R	-/-	-/-	D/-	D/-	D/-	TP (3)
CD CHRDY (n)	O/R	-/-	-/-	O/-	O/-	O/-	TP (3)
CHRDYRTN	D/-	-/R	-/R	-/-	-/-	-/-	BD (4)
ARB(0-3)	O/R	-/R	D/R	D/R	-/-	-/-	OC (5)
-BURST	O/R	D/R	D/-	#/-	-/-	-/-	OC (5)
-PREEMPT	O/R	-/-	D/R	D/#	-/-	-/-	OC (5)
ARB/-GNT	D/-	-/R	-/R	-/R	-/-	-/-	BD (4)
-TC	-/-	D/-	-/-	-/O	-/-	-/-	TS (1)
-IRQ (*)	-/R	O/-	O/-	O/-	-/-	O/-	OC (6)
-CD SETUP (n)	D/-	-/#	-/R	-/R	-/R	-/R	TP (7)
-CHCK	-/R	#/-	D/O	D/-	D/-	D/-	OC (6)
-REFRESH	D/R	-/O	-/O	-/O	-/R	-/O	TS (1)
OSC	D/-	-/O	-/O	-/O	-/O	-/O	CD (7)
CH RESET	D/-	-/O	-/R	-/R	-/R	-/R	BD (4)
A(24-31)	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
-BE(0-3)	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
D(16-31)	D/R	D/R	D/R	D/R	D/R	D/R	TS (2)
-CD DS 32 (n)	-/R	-/R	-/R	#/-	#/-	#/-	TP (3)
-DS 32 RTN	D/-	-/-	-/O	-/-	-/-	-/-	BD (4)
TR 32 RTN	-/-	-/-	-/O	-/-	O/-	O/-	TS (1)
-MMC	D/-	-/-	-/-	-/-	-/O	-/O	TS (1)
-MMCR	-/R	-/-	-/-	-/-	O/-	O/-	TP(3)
-MMC CMD	D/-	-/-	-/-	-/-	-/O	-/O	TS (1)

KEY

D = Drive Enabled
O = Optional
R = Receive Enabled
- = Not Implemented
= Some are Required

OC = Open Collector
TS = Tri-State
TP = Totem-Pole
BD = Bus Driver
CD = Clock Driver

* IRQ (9-12, 14, 15, 3-7)

Figure 2-8. Driver/Receiver Requirements and Options

The following notes apply to the driver and receiver options listed on page 2-26.

Notes:

1. During the Reset state, an active CHRESET must degate all bus drivers.
2. During the Reset state, the state of all signals is unknown.
3. A CD SETUP (n) line is driven to only one channel connector at a time.
4. All pull-up resistors are provided by the system logic and pulled up to +5 volts.
5. Loading Current: 1.6 milliamps maximum load per channel connector except signal group 5. The loading current of group 5 is 1.0 milliamp maximum per channel connector.
6. Loading Capacitance:
   - 15-pf maximum capacitance loading permitted for adapter for OSC and Group 5 signals (see Figure 2-9 on page 2-28 for Group 5 definition) and 20 pf maximum capacitance loading permitted for adapter for all other signals. (The value refers to the capacitance from the adapter side of the connector to the adapter driver/receiver.)
   - Total capacitance seen by the driver is 200 pf maximum for Group 5 signals and OSC and 240 pf maximum for all other signals. The symbol (#) refers to average capacitance across 0.1-volt to 2.3-volt interval.
7. An Open Collector can be either an Open Collector device or a tri-state device wired with the input grounded and using the 'enable' line to control the output.
8. The electromagnetic interference (EMI) potential of a bus driver increases as its output voltage transition time decreases. Therefore, the drivers with output transitions greater than 1 volt per nanosecond should be used only when essential to meet channel timing requirements. The figure lists the role of driver or receiver during a given operation being performed. The names of the physical packaging of the logic should not be confused with the performed functions.

Signal Group	Driver Type
1, 2	Tri-state with 24 mA sinking capacity.
3	Totem-pole (TP) with current sinking capacity of 6 mA.
4	Bus Driver (BD) with current sinking capacity of 24 mA.
5, 6	Open Collector (OC) with 24 mA sinking capacity.
7	Unique drivers: Totem-pole (TP) or Tri-state (TS) with current sinking capacity of 6 mA. Clock Driver (CD) with 24 mA sinking capacity.

Figure 2-9. Signal Driver Types

Channel Signal Groups (Auxiliary Video Extension)

An adapter using the auxiliary video connector must not exceed the loading limits shown below for any auxiliary video connector signal pin it is receiving:

- C max = 15.0 pF
- I_{IL} min = - 1.6 mA
- I_{IH} max = 50.0 μA.

An adapter using the auxiliary video connector must meet the following minimum requirements for any auxiliary video connector signal pin it is driving:

- C min = 150.0 pF
- I_{OL} min = 10.0 mA - VSYNC and HSYNC
  = 2.0 mA - All other signals
- I_{OH} max = - 4.0 mA - VSYNC and HSYNC
  = - 0.25 mA - All other signals.

Programmable Option Select

The Programmable Option Select (POS) eliminates switches from the system board and adapters by replacing their function with programmable registers.

Through the use of adapter description files (ADFs) for each adapter, the System Configuration utilities (described on page 2-51) can automatically create configuration data for the system board and each adapter. This is achieved by reading a unique adapter ID number from each adapter, matching it with an ADF file, and configuring the system accordingly. The resulting data is stored in battery-backed CMOS RAM along with the adapter ID numbers.

This permits the Power-On Self-Test (POST) to automatically reconfigure the system whenever the system is powered on. The POST will first verify that the configuration has not changed by reading the adapter ID numbers and comparing them with those values stored in the battery-backed CMOS. If the configuration has changed, it is necessary to rerun the System Configuration utilities.

The adapters and the system board setup functions all share I/O addresses hex 0100 through 0107. The system board POS port (I/O address hex 0094) and the adapter POS port (I/O address hex 0096) control the device unique setup signals for all devices.

Warning: IBM recommends that programmable options be set only through the System Configuration utilities. Direct setting of the POS registers and/or CMOS RAM POS parameters can result in multiple assignment of the same system resource, improper operation of the feature, loss of data, or possible damage to the system or options. Application programs should not use adapter identification (ID) information. Use of this information may cause compatibility problems between systems or options.

System Board Video Subsystem Setup: The video subsystem setup functions respond to I/O addresses hex 0100 through 0107 only when the unique video subsystem 'setup' signal is active.

Port hex 0094 bit 5 must be 0 for video subsystem setup with I/O addresses hex 0100 through 0107.

Port hex 0096 bit 3 must be 0 to avoid driving a 'setup' signal to an adapter.

Port hex 0094 bit 7 must be 1 to avoid driving a 'setup' signal to other system board functions.

Other System Board Setup: The other system board setup functions respond to I/O addresses hex 0100 through 0107 only when the unique device 'setup' signal is active.

Port hex 0094 bit 7 must be 0 for other system board setup with I/O addresses hex 0100 through 0107.

Port hex 0096 bit 3 must be 0 to avoid driving a 'setup' signal to an adapter.

Port hex 0094 bit 5 must be 1 to avoid driving a 'setup' signal to the VGA.

Adapter Setup: The adapter setup functions respond to I/O addresses hex 0100 through 0107 only when their unique 'setup' signal is active.

Port hex 0096 bit 3 must be 1 for adapter setup with I/O addresses hex 0100 through 0107.

Port hex 0094 bit 5 must be 1 to avoid driving a 'setup' signal to the VGA.

Port hex 0094 bit 7 must be 1 to avoid driving a 'setup' signal to a system board function.

Warning: If both an adapter and a portion of the system board are selected for setup at the same time, bus contention will result; no useful programming can take place and damage to the hardware can occur. Port hex 0096 should be set to hex 00 and port hex 0094 should be set to hex FF when setup operations are completed.

The following shows the organization of the I/O address space used by the POS.

Address (hex)	Function
0094	System Board Enable/Setup Register
0095	Reserved
0096	Adapter Enable/Setup Register
0097	Reserved
0100	POS Register 0 - Adapter Identification Byte (Least-significant byte)
0101	POS Register 1 - Adapter Identification Byte (Most-significant byte)
0102	POS Register 2 - Option Select Data Byte 1 Bit 0 of this byte is designated as Card Enable (CDEN).
0103	POS Register 3 - Option Select Data Byte 2
0104	POS Register 4 - Option Select Data Byte 3
0105	POS Register 5 - Option Select Data Byte 4 Bit 7 of this byte is designated as -CHCK. Bit 6 of this byte is reserved.
0106	POS Register 6 - Subaddress Extension (Least-significant byte)
0107	POS Register 7 - Subaddress Extension (Most-significant byte)

Figure 2-10. POS I/O Address Space

Bits 6 and 7 of address hex 0105 and bit 0 of address hex 0102 are fixed. All other fields within the address range of hex 0102 and 0105 are free form.

Card Selected Feedback Register

When an access to the address space of the adapter occurs, the adapter responds by setting the '-card selected feedback' (-CD SFDBK) signal to 0. -CD SFDBK is derived by the adapter from the address decode, and is driven by an open collector driver to the common '-card selected feedback' line. -CD SFDBK is latched by the system board and made available on subsequent cycles. -CD SFDBK may be used by automatic configuration or diagnostics to verify operation of an adapter at a given address or DMA port.

In order to determine if the VGA, the system board, or an adapter is addressed and functioning, the Card Select Feedback byte is provided. This byte may be read at address hex 0091. Bit 0 of this byte is set to 1 whenever the 'card selected feedback' signal was active on a previous cycle or when the system board I/O functions (diskette drive, serial, or parallel interfaces) are accessed by an I/O cycle. Bit 0 is reset by the read operation. Bits 1 through 7 are reserved. This byte is read-only.

System Board Setup

The integrated I/O functions of the system board use POS information during the setup procedure. The diskette drive controller, serial port, and parallel port are treated as a single device. Although the VGA is an integrated part of the system board, POS treats it as a separate device. The Setup Enable register is used to place the system board or the Video Subsystem into setup. The Setup Enable register is a read/write register at I/O address hex 0094. The bit definitions are provided in the following figure.

Bit	Function
7	System Board Functions +Enable/−Setup
6	Reserved
5	VGA +Enable/−Setup
4	Reserved
3	Reserved
2	Reserved
1	Reserved
0	Reserved

Figure 2-11. System Board Setup Enable Register (Hex 0094)

The default state of all bits is 1 (enable state).

Bit 7 When set to 1, this bit enables various functions of the system board and system board resident I/O functions. When set to 0, these functions are placed in setup.

Bit 5 When set to 1, this bit enables the VGA. When set to 0, the VGA is placed in the setup mode. See "Video Subsystem Programmable Option Select" on page 4-29 for more information.

When bit 7 of the Setup Enable Register is set to 0, the diskette drive controller, serial, and parallel interfaces are controlled by a bit pattern written to port hex 0102, the System Board I/O byte. A read operation to port hex 0102 returns the current state of the respective system board function. All bits are read/write. The bit definitions of port hex 0102 are provided in the following figure.

Bit	Function
7	Enable/Disable Parallel Port Extended Mode
6	Parallel Port Select (High Bit)
5	Parallel Port Select (Low Bit)
4	Enable/Disable Parallel Port
3	Serial Port Select
2	Enable/Disable Serial Port
1	Enable/Disable Diskette Drive Interface
0	Enable/Disable System Board

Figure 2-12. System Board I/O Byte (Hex 0102)

Bit 7 When set to 0, this bit allows the parallel port to be configured as an 8-bit parallel bidirectional interface. When set to 1, the bidirectional mode is disabled. The power-on reset state of this bit is 0 and the POST sets it to 1.

Bits 6, 5 The state of these bits selects the configuration of the system board parallel port.

Bit 6	Bit 5	Assignment	Address (hex)
0	0	Parallel 1	03BC - 03BE
0	1	Parallel 2	0378 - 037A
1	0	Parallel 3	0278 - 027A
1	1	Reserved	---

Bit 4 When set to 1, this bit enables the system board parallel port, if bit 0 is set to 1. When set to 0, the system board parallel port is disabled.

Bit 3 When set to 1, this bit sets the system board serial port as Serial 1. When set to 0, the system board serial port is set to Serial 2.

Bit 2 When set to 1, this bit enables the system board serial port, if bit 0 is set to 1. When set to 0, the system board serial port is disabled.

Bit 1 When set to 1, this bit enables the diskette drive interface providing bit 0 is set to 1. When set to 0, the diskette drive interface is disabled.

Bit 0 When set to 1, this bit allows bits 4, 2, and 1 to enable and disable their respective devices. When set to 0, the diskette drive interface, system board serial port, and system board parallel port are disabled regardless of the state of bits 4, 2, and 1.

Note: The diskette drive controller is always enabled by the POST.

System Board Memory Setup

The Programmable Option Select bytes give the system board the flexibility of accepting either one or two 1M memory cards without setting switches. They also permit control of how the first 1M of memory is addressed (split) and allow individual 1M blocks of system board memory to be activated and deactivated.

If a memory error is detected in the first 512K of memory by the POST, the first physical 1M block of memory is deactivated. The addresses assigned to the deactivated block are reassigned to the second physical 1M block of system board memory, if installed. This memory reassignment allows the system to recover from an otherwise catastrophic error. In the event an error is detected in the first 512K of each installed system board memory card, the first 512K of the memory address space cannot be reassigned to reside on an adapter installed in the channel, therefore the system cannot recover.

The POST does not deactivate memory for an error detected above the first 512K of memory. If an error is detected above the first 512K of memory, the 1M block containing the error can be deactivated and its addresses reassigned through a Setup program supplied with the system. Once a block of defective memory is deactivated, it is ignored by the POST during subsequent power-ons.

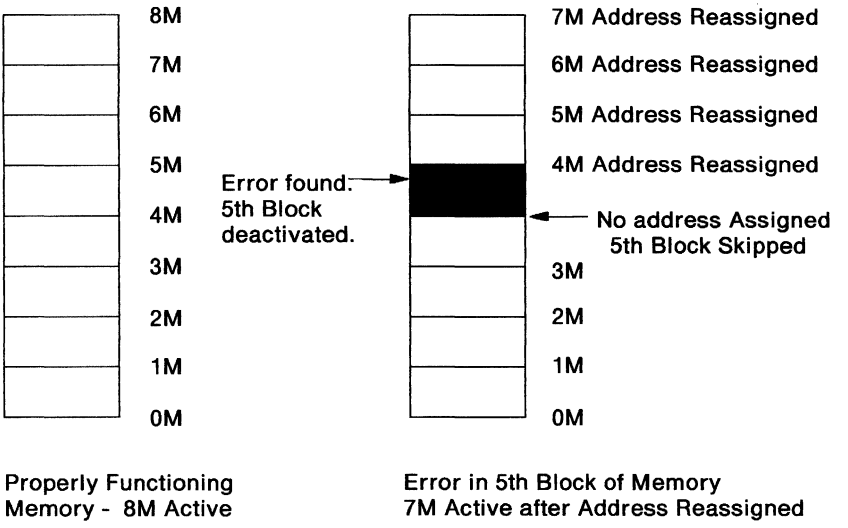


Figure 2-13. Memory Error Address Reassignment

Four registers define the memory setup feature:

- Memory Control Register
- Memory Card Definition Register
- Memory Encoding Register
- Split Address Register.

Memory Control Register: This register defines the refresh rate.

To access this register, I/O address hex 0096, bit 3 must be set to 0 to avoid sending a 'setup' signal to one of the channel connectors. Also, I/O address hex 0094, bit 7 must be set to 0, causing the system board to accept setup cycles. After accessing the Memory Control register, I/O address hex 0094, bit 7 must be set to 1. This register is accessed using I/O address hex 0103 and is write-only.

Bit	Function
7 - 2	Reserved = 0
1	-Fast Refresh
0	Reserved = 0

Figure 2-14. Memory Control Register

Bits 7 – 2 Reserved. These bits must be set to 0.

Bit 1 When this bit is set to 0, a fast refresh of approximately 0.8 microseconds is selected. When set to 1, a normal refresh of approximately 15.12 microseconds is selected.

Bit 0 Reserved. This bit must be set to 0.

Memory Card Definition Register: The information in this register indicates the presence of memory in each of the two system board memory connectors. Connector 1 is located closest to the power supply; connector 2 is located furthest from the power supply.

To access this register, I/O address hex 0096, Bit 3 must be set to 0 to avoid sending a 'setup' signal to one of the channel connectors. Also, I/O address hex 0094, bit 7 must be set to 0, causing the system board to accept setup cycles. After accessing the Memory Card Definition register, I/O address hex 0094, bit 7 must be set to 1. This register is accessed using address hex 0103 and is read-only.

Bit	Function
7 - 4	Reserved
3	T2 (T Signal, Connector 2)
2	R2 (R Signal, Connector 2)
1	T1 (T Signal, Connector 1)
0	R1 (R Signal, Connector 1)

Figure 2-15. Memory Card Definition Register

The following figures define how the T and R values determine the presence of memory.

Bit 3 T2	Bit 2 R2	Available RAM
0	0	80386 System Board Memory Expansion Kit in Connector 2
0	1	Reserved
1	0	Reserved
1	1	No Kit Present in Connector 2

Figure 2-16. T2 and R2 Bit Values

Bit 3 T2	Bit 2 R2	Available RAM
0	0	80386 System Board Memory Expansion Kit in Connector 1
0	1	Reserved
1	0	Reserved
1	1	No Kit Present in Connector 1

Figure 2-17. T1 and R1 Bit Values

Memory Encoding Register (I/O Address Hex 00E1)

This register determines the amount of enabled system board memory and how it is used. This register is read/write.

Bit	Function
7	EN4
6	EN3
5	EN2
4	EN1
3	ENSPLIT
2	640
1	ROMEN
0	ENPLRPCH

Figure 2-18. Memory Encoding Register

Bits 7, 6 EN4-EN3 — These bits define the amount of memory to be used in system board memory connector 2.

Bit 7 EN4	Bit 6 EN3	Amount of RAM	Function
1	0	1M	80386 System Board Memory Expansion Kit enabled in Connector 2
1	1	0M	Kit Disabled or No Kit Present in Connector 2
0	0	N/A	Reserved
0	1	N/A	Reserved

Figure 2-19. Memory Encoding Register, Bits 7 and 6

Bits 5, 4 EN2-EN1 — These bits define the amount of memory to be used in system board memory connector 1.

Bit 5 EN2	Bit 4 EN1	Amount of RAM	Function
1	0	1M	80386 System Board Memory Expansion Kit enabled in Connector 1
1	1	0M	Kit Disabled or No Kit Present in Connector 1
0	0	N/A	Reserved
0	1	N/A	Reserved

Figure 2-20. Memory Encoding Register, Bits 5 and 4

Bit 3 ENSPLIT — This bit determines whether the amount of memory beyond the split (384K or 512K) is to be addressed to another location in memory or disabled.

When set to 0, this bit enables the high portion of the first physical 1M of active memory (portion beyond the split) at any 1M boundary up to 16M (determined by the Split Address register). When set to 1, the portion beyond the split is not activated. Thus the high portion (384K or 512K) is unused unless it is used in the ROM address space (Bit ROMEN=0).

Bit 2 640 — This bit determines where the first active 1M block of RAM is split.

When this bit is set to 0, 640K of the first 1M of memory is mapped into the first 640K of address space (hex 00000000 to 0009FFFF). When set to 1, 512K of the first 1M of memory is mapped into the first 512K of address space (hex 00000000 to 0007FFFF).

Bit 1 ROMEN — This bit can deactivate the system board ROM and use its address space for system board RAM.

When set to 0, this bit places the ROM address space (hex 000E0000 to 000FFFFF) into DRAM. ENSPLIT must be set to 1 (inactive). Memory at 640K to 896K or 512K to 896K (depending on the 640 bit) is unused and the ROM modules are deactivated. When this bit is set to 1, the ROM modules are left active.

The following figure shows how bits 3, 2, and 1 are used together to define the various configurations.

Bit 3 ENSPLIT	Bit 2 640	Bit 1 ROMEN	Function
0	0	0	Invalid
0	0	1	ROM Enabled, Split at 640K, High 384K at Split Address
0	1	0	Invalid
0	1	1	ROM Enabled, Split at 512K, High 512K at Split Address
1	0	0	ROM Disabled, Split at 640K, Part of High 384K at ROM Address
1	0	1	ROM Enabled, Split at 640K, High 384K Not used
1	1	0	ROM Disabled, Split at 512K, Part of High 512K at ROM Address
1	1	1	ROM Enabled, Split at 512K, High 512K Not Used

Figure 2-21. Memory Encoding Register, Bits 3, 2, and 1

Bit 0 ENPLRPCH — This bit enables and disables parity checking of system board RAM.

When this bit is set to 0, parity checking of system board RAM is enabled. When set to 1, parity checking of RAM is disabled. To clear a parity error, this bit must be set to 1 (parity checking disabled) and then set to 0 (parity checking enabled).

Split Address Register (I/O Hex Address 00E0)

Bit	Name
7 - 4	Reserved = 0
3	SPA23
2	SPA22
1	SPA21
0	SPA20

Figure 2-22. Split Address Register

Bits 7 – 4 Reserved. These bits must be set to 0.

Bits 3 – 0 SPA23 through SPA20 — This register defines the starting location of the high portion of the first 1M of active memory (portion beyond the split).

This register may not be set to a value of 0 unless ENSPLIT is set to 1 (inactive). When ENSPLIT is set to 0 (active), this address determines the starting location of

the high portion of the split (384K or 512K depending on the value of the 640 bit in the Memory Encoding register). The starting location can be at any 1M boundary from 1M to 15M. This register is read/write.

Note: If the total system memory is equal to or greater than 16M, the memory split capability cannot be used

System Memory Maps

Memory is mapped by the Memory Encoding register and Split Address register. The mapping results in either 512K or 640K of system board RAM starting at address hex 00000000. A 256-byte portion of this RAM is reserved as a BIOS data area. A 1K portion of this RAM is reserved as an extended BIOS data area. See the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for details.

In the following figures, the variable *X* represents the number of 1M blocks of system board memory starting at or above the hex 00100000 boundary (*X* = 0 or 1). The variable *Y* represents the number of 1M blocks of memory installed in the channel starting at or above the hex 00100000 boundary (*Y* = 0 to 15).

The following figure shows the memory mapping when:

Enable Split bit = 1
640 bit = 1
ROM enable bit = 1
Split address byte = N/A.

Hex Range	Function
00000000 to 0007FFFF	512K System Board RAM
00080000 to 0009FFFF	Not Used
000A0000 to 000BFFFF	128K System Board Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128K System Board ROM
00100000 to (00100000 + *XM*)	*XM* System Board RAM
(00100000 + *XM*) to (00100000 + *XM* + *YM*)	*YM* Channel RAM
(00100000 + *XM* + *YM*) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128K System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 2-23. System Memory Map 1

The following figure shows the memory mapping when:

Enable Split bit = 1
640 bit = 0
ROM enable bit = 1
Split address byte = N/A.

Hex Range	Function
00000000 to 0009FFFF	640K System Board RAM
000A0000 to 000BFFFF	128K System Board Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128K System Board ROM
00100000 to (00100000 + *XM*)	*XM* System Board RAM
(00100000 + *XM*) to (00100000 + *XM* + *YM*)	*YM* Channel RAM
(00100000 + *XM* + *YM*) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128K System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 2-24. System Memory Map 2

The following figure shows the memory mapping when:

Enable Split bit = 0
640 bit = 1
ROM enable bit = 1
Split address byte = 1 + *X* + *Y* (Total Range = 1 to 15).

Hex Range	Function
00000000 to 0007FFFF	512K System Board RAM
00080000 to 0009FFFF	Not Used
000A0000 to 000BFFFF	128K System Board Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128K System Board ROM
00100000 to (00100000 + *XM*)	*XM* System Board RAM
(00100000 + *XM*) to (00100000 + *XM* + *YM*)	*YM* Channel RAM
(00100000 + *XM* + *YM*) to (00100000 + *XM* + *YM* + 512K - 1)	512K System Board RAM
(00100000 + *XM* + *YM* + 512K) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128K System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 2-25. System Memory Map 3

The following figure shows the memory mapping when:

Enable Split bit = 0
640 bit = 0
ROM enable bit = 1
Split address byte = 1 + *X* + *Y* (Total Range = 1 to 15).

Hex Range	Function
00000000 to 0009FFFF	640K System Board RAM
000A0000 to 000BFFFF	128K System Board Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128K System Board ROM
00100000 to (00100000 + *XM*)	*XM* System Board RAM
(00100000 + *XM*) to (00100000 + *XM* + *YM*)	*YM* Channel RAM
(00100000 + *XM* + *YM*) to (00100000 + *XM* + *YM* + 384K - 1)	384K System Board RAM
(00100000 + *XM* + *YM* + 384K) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128K System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 2-26. System Memory Map 4

The following figure shows the memory mapping when:

Enable Split bit = 1
640 bit = 1
ROM enable bit = 0
Split address byte = N/A.

Hex Range	Function
00000000 to 0007FFFF	512K System Board RAM
00008000 to 0009FFFF	Not Used
000A0000 to 000BFFFF	128K System Board Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128K System Board RAM
00100000 to (00100000 + *XM*)	*XM* System Board RAM
(00100000 + *XM*) to (00100000 + *XM* + *YM*)	*YM* Channel RAM
(00100000 + *XM* + *YM*) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128K System Board ROM

Figure 2-27. System Memory Map 5

The following figure shows the memory mapping when:

Enable Split bit = 1
640 bit = 0
ROM enable bit = 0
Split address byte = N/A.

Hex Range	Function
00000000 to 0009FFFF	640K System Board RAM
000A0000 to 000BFFFF	128K System Board Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128K System Board RAM
00100000 to (00100000 + XM)	XM System Board RAM
(00100000 + XM) to (00100000 + XM + YM)	YM Channel RAM
(00100000 + XM + YM) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128K System Board ROM

Figure 2-28. System Memory Map 6

Adapter Setup

-CD SETUP (n) is unique for each channel position. When -CD SETUP (n) is active, adapters recognize Setup read and write operations. The adapter decodes all three low-order address bits (A0 through A2) to determine the POS register that is to be read from or written to.

The setup (Automatic Configuration) routine obtains adapter information from Adapter Description files and uses I/O addresses hex 0100 through 0107 to address the POS bytes of the adapter. The following figure shows the organization of the address space used by POS during adapter setup operations.

Address (hex)	-CD SETUP	Address Bit A2	A1	A0	Function
0100 (POS Register 0)	0	0	0	0	Adapter Identification Byte (Least-significant byte)
0101 (POS Register 1)	0	0	0	1	Adapter Identification Byte (Most-significant byte)
0102 (POS Register 2)	0	0	1	0	Option Select Data (Byte 1)*
0103 (POS Register 3)	0	0	1	1	Option Select Data (Byte 2)
0104 (POS Register 4)	0	1	0	0	Option Select Data (Byte 3)
0105 (POS Register 5)	0	1	0	1	Option Select Data (Byte 4)*
0106 (POS Register 6)	0	1	1	0	Subaddress Extension (Least-significant byte)
0107 (POS Register 7)	0	1	1	1	Subaddress Extension (Most-significant byte)

* These bytes contain one or more bits with specific assignments.

Figure 2-29. POS I/O Address Decode

Bytes hex 0100 and 0101 are 8-bit read-only. Bytes hex 0102 through 0107 are 8-bit read and write if implemented.

All bits in bytes hex 0102 through 0105 are free form and adapter dependent except for the following:

- **Hex 0102, Bit 0:** Card Enable (CDEN): When this bit is set to 0, the adapter is disabled, responding only to setup read and write operations. It does not respond to any I/O or memory read or write operations nor does it make any interrupt requests. When set to 1, the adapter is fully enabled.
- **Hex 0105, Bit 7:** Channel Check Active Indicator: System memory and I/O functions that report a channel check must set a channel check active indicator to identify the source of the error. This indicator is bit 7 of address hex 0105, of each adapter POS address space. This bit is interrogated by the NMI handler responding to -CHCK for each adapter position until all reporting adapters have been identified.

The following shows a typical implementation of the channel check active indicator.

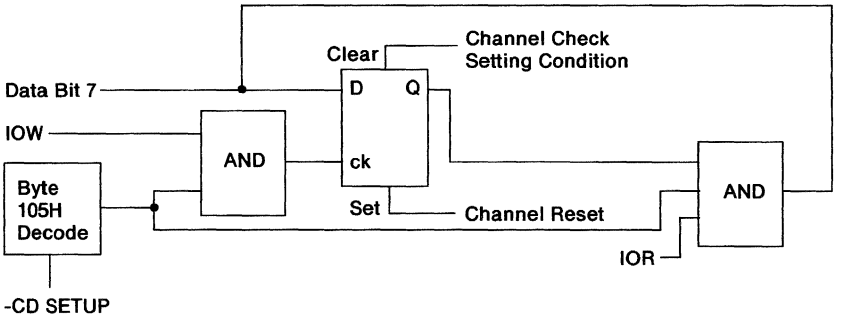


Figure 2-30. Channel Check Active Indicator

The indicator is set to 0 on a channel check condition, or when bit 7 of hex 0105 is 0. The indicator is set to 1 on a channel reset, or when bit 7 of hex 0105 is 1. This bit may be reset by any action that occurs during the channel check service routine. If the channel check active indicator is used by an attachment, hex 0105 bit 6 may be used to indicate that additional status is available through bytes hex 0106 and 0107.

- **Hex 0105, Bit 6:** Channel Check Status Indicator (STAT): When set to 0, this bit indicates that channel check exception status is available using POS bytes hex 0106 and 0107. When set to 1, it indicates no status is available. The field at hex 0106, 0107 may be the status, a pointer to status at another address, or a command port to present the address elsewhere (for example, in a subaddressed area).

  This bit is required by all devices supporting the CHCK bit. If a device does not report status, this bit is equal to 0. If a device does not support the CHCK bit, this bit can be programmed to contain device-unique information.

Adapter Identification

Each adapter must have a unique 2-byte adapter ID used for identification by diagnostic programs, configuration utilities, and POST routines that initialize the adapter when the system is powered on.

If an adapter is not ready or not initialized when interrogated, it must output an ID value of hex 0000 instead of the true ID until it is ready. If the value remains hex 0000 in excess of 1 second, the system assumes the adapter is malfunctioning. This function is not supported by the IBM Personal System/2 Model 50 and Model 60.

To minimize hardware, only those bits driven to 0 require drivers. Pullup resistors on the system board provide a 1 for remaining bits.

Required Fields

Several fields are not assigned specific bit locations within the *free form* POS bytes. However, the following fields are required if the adapter supports the function:

- **Fairness Enable Field:** All bursting devices using the arbitration mechanism must support the *fairness* feature through the programmable fairness-enable bit. The default state of this bit is a 1, establishing all devices to honor the fairness feature. When the fairness feature is honored, and an arbitrating device is preempted, the device enters the inactive state, and must wait for an inactive -PREEMPT before exiting from the inactive state and competing for the channel again. This allows the system to service all arbitrating devices in order of priority before the same device can gain control of the channel again. When this bit is set to 0, the fairness feature is disabled.
- **Arbitration Level Field:** All devices (bursting and nonbursting) using the arbitration mechanism must support a programmable arbitration level through a 4-bit allocation. This field allows devices that are prioritized incorrectly to be reassigned by diagnostic or system programs to reduce impacts on performance. Only one device may be assigned to each arbitration level.
- **Device ROM Segment Address Field:** All I/O devices containing memory mapped I/O ROM must support a programmable Device

ROM Segment Address Field. This field can be up to 4 bits and provides the ROM of a device a starting address at any one of sixteen 8K segments.

- **I/O Device Address Field:** All I/O devices that can simultaneously reside in a system with a device of the same type must support a programmable I/O device address. This field eliminates addressing conflicts.

Channel Position Select Register

Each channel position has a unique 'setup' line (-CD SETUP) associated with it. Prior to a setup cycle, an I/O operation to port hex 0096 sets up the bit pattern to select the channel position to which the subsequent setup operation will occur. The following figure shows the valid bit-patterns written.

Note: The status of port hex 0096 can be read by software. However, when read, bits 6, 5, and 4 are set to 1.

Channel Position Selected for Setup	7	6*	5*	4*	3	2	1	0
None	0	0	0	0	0	X	X	X
1	0	0	0	0	1	0	0	0
2	0	0	0	0	1	0	0	1
3	0	0	0	0	1	0	1	0
4	0	0	0	0	1	0	1	1
5	0	0	0	0	1	1	0	0
6	0	0	0	0	1	1	0	1
7	0	0	0	0	1	1	1	0
8	0	0	0	0	1	1	1	1
Channel Reset	1	0	0	0	X	X	X	X

* These bits are written to 0, but read as 1.

Figure 2-31. Channel Position Select Register (Hex 0096)

Setup information is then read from or written to the selected adapter by reading or writing to I/O addresses hex 0100 through 0107.

Note: -CD SETUP only goes active when an operation in the I/O address range hex 0100 through 0107 is performed.

The 'channel reset' line is driven active by setting bit 7 of port hex 0096 to 1.

Adapter POS Implementation

The following figure is a simplified presentation showing how POS is typically implemented by an adapter. All designs must latch the least-significant bit of the device-dependent option select byte. Bit 7 of POS register 5 is set to 1 unless -CHCK is active from the adapter. The remaining bits can be implemented as required.

Notes:

1. Any adapter not completely initialized by POS should implement a second enable (for such functions) that is activated by adapter ROM routines or loadable software. The card disable function (POS register 2, bit 0) must override a secondary enable.
2. Following a channel reset, the default POS-defined resources should not conflict with system RAM, ROM, arbitration level (diskette drive controller = 2), or I/O assignments. When bit 0 of POS register 2 is set active, adapter ROM (or RAM operating as ROM) must reside in the space hex 000C0000 to 000DFFFF.

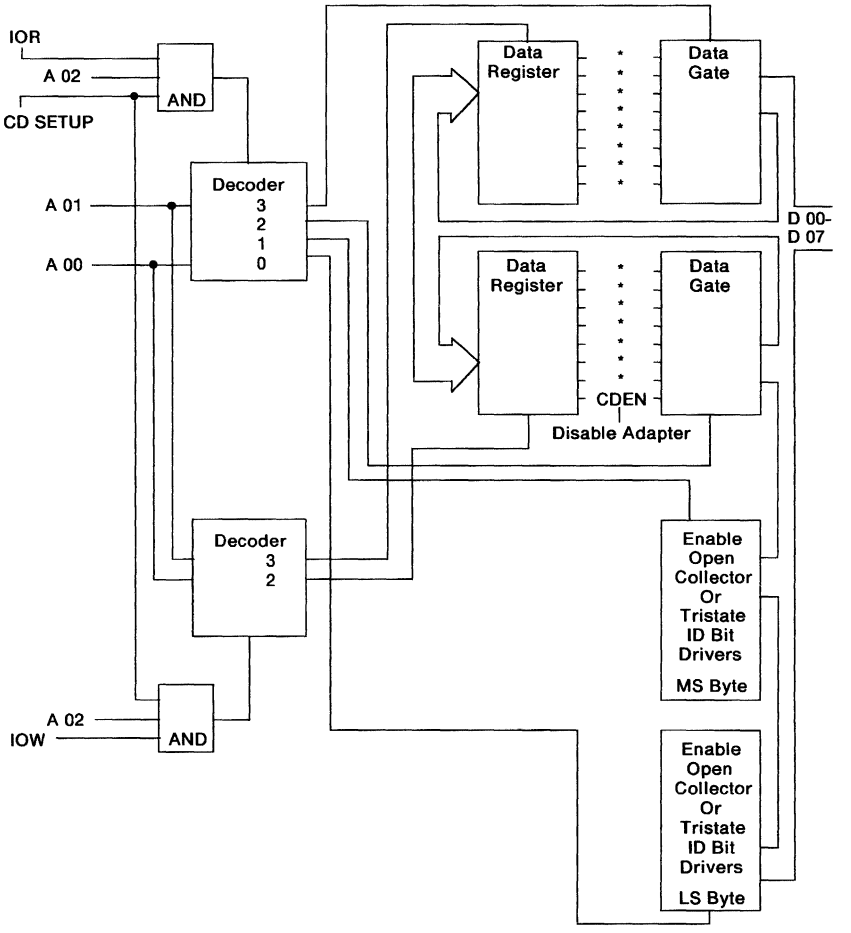


Figure 2-32. Typical Adapter Implementation of POS

The POS subaddressing extension allows the subaddressing of a block of initial program load (IPL) or setup information. Subaddressing bits (SAD00 through SAD15) may be used to address RAM, a register stack, or other 8-bit or 16-bit devices.

The following is a simplified presentation showing the subaddressing extension for memory. The counter registers are incremented after each least-significant byte of option select information is written.

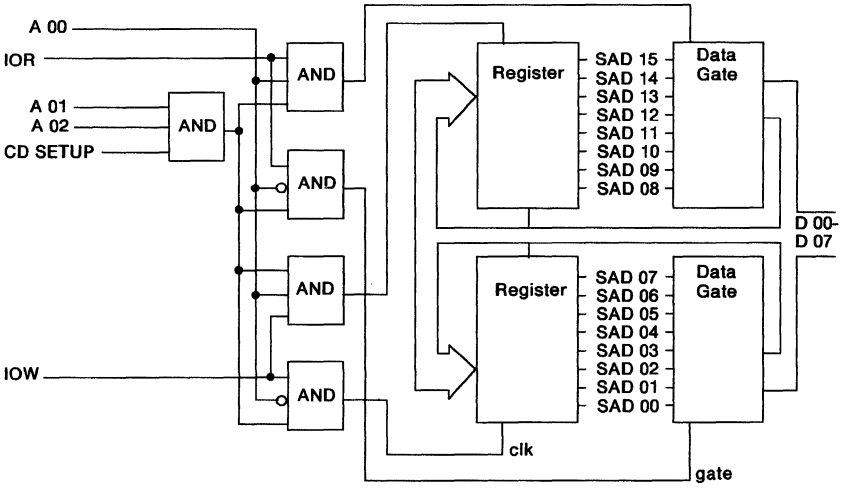


Figure 2-33. Subaddressing POS Extension Example

POS Implementation Procedure

While the design of the POS circuitry is the designer's choice, an example of a typical POS implementation is as follows:

1. Disable interrupts.
2. Select the adapter to be placed in setup (-CD SETUP = 0) by writing the appropriate value to port hex 0096.
3. Read the adapter ID by performing an I/O read at hex 0100 and hex 0101.
4. Perform an I/O write to hex 0102 with bit 0 off to disable the adapter and place it in setup.
5. Write POS data to hex 0103, 0104, and 0105 in any order.
6. With bit 0 on, do a write operation to hex 0102 with POS data.
7. Disable setup by writing hex 00 to the Setup Enable register at hex 0096.
8. Enable interrupts.

The system microprocessor is able to communicate with the adapter provided the adapter is enabled (bit 0 at hex 0102 set to 1). After setup, a subsequent I/O write does not affect these latches or permit

the ID circuitry of the adapter to operate, unless the adapter is returned to setup.

System Configuration Utilities

Each system is packaged with a Reference diskette containing the System Configuration utilities. These utilities are used to identify the hardware installed and interpret system resources (such as I/O ports and arbitration levels) required for each I/O device.

The Automatic Configuration function stores POS data for the system board and adapters in CMOS RAM. POS data, for the system board and a limited number of adapters, is contained on the Reference diskette. POS information for additional adapters is merged onto diskette in the form of *adapter description files*. Diagnostic code modules and power-on self-test (POST) error message files can be merged at the same time.

Automatic Configuration compares the POS data in CMOS RAM with the POS data in each adapter description file to determine what system resources are being used and how the adapter is configured. If an adapter does not have an adapter description file, the Set Configuration program disables the adapter during Automatic Configuration.

To keep track of what system resources are in use or are available during the configuration process, a temporary record of the resources required by each adapter is maintained. Resource status is kept for I/O address blocks, arbitration levels, interrupt levels, and memory address blocks.

The Set Configuration program has the following resource limits for each adapter:

- Up to two different memory blocks for each adapter
- Up to 16 different I/O blocks, arbitration levels, and interrupt levels for each adapter.

Change Configuration Utility

The Change Configuration utility is provided to selectively change the configuration from the Automatic Configuration default settings. This utility is generally used to resolve unusual conflicts or set items for personal preference.

The user interface is structured in terms of scrolling and paging through screens. Changes are made by rotating field value names through a set of choices with the F5 (Previous) and F6 (Next) keys. Changes are made to CMOS when the F10 (Save) key is used. Help text is provided for each item field.

Conflicts are indicated by an asterisk (*) next to the conflicting items and also by the "* Conflicts" flag string in the upper right corner of the Change Configuration window.

The View Configuration utility is provided to view the configuration without accidentally changing the CMOS RAM. This is the Change Configuration utility with the change capabilities disabled.

Adapter Identification and POS Data

The file name for each adapter description file contains a unique 16-bit card ID that corresponds to the card ID generated by the adapter. The CMOS location where the card ID and POS information are stored depends on the system type. Automatic Configuration determines the system type by reading the machine ID. Once the system type is known, the number of slots and CMOS storage differences between the Personal System/2 products can be handled. If the system is identified as a Model 50, Automatic Configuration stores the card ID and POS data in separate locations in the 64-byte CMOS RAM. Each channel position is allocated 2 bytes for each card ID and 4 bytes for POS data. If the system is identified as a Model 60 or Model 80, Automatic Configuration stores the card ID and POS data together in the 2K CMOS RAM extension.

Automatic Configuration compares stored card IDs with the card IDs generated by the adapters to determine if a channel connector is empty or an adapter has been added, repositioned, removed, or remains unchanged. Automatic Configuration takes place only for those channel positions where a change is detected.

A 2-byte CRC check character is maintained for the 64-byte CMOS RAM and a separate 2-byte CRC check character is maintained for the 2K CMOS RAM extension. These check characters are calculated and set by the configuration utilities whenever new information is recorded in CMOS. These check characters are used by the POST to check the validity of CMOS data.

POST Error Processor Files

If the POST finds any difference between the card IDs stored in CMOS RAM and card IDs generated by the adapters, a POST configuration error is generated. When the system is powered on with the Reference Diskette inserted, the POST Error Processor files provide a detailed description of the POST error. Depending on the type of POST Configuration Error, the mismatch may be resolved in one of two ways: Automatic Configuration or Selective Configuration.

If the source of the error is a battery failure or a CRC error, Automatic Configuration takes place immediately after all POST error screens are displayed. The entire system is configured to the first nonconflicting values for each *NamedItem* defined in the adapter description file. Adapters are configured in the order of the channel position in which they are installed. Position 1 is configured first, position 2 is second, and so on. Automatic Configuration does not backtrack to previously configured adapters to resolve conflicts that arise in adapters installed in higher positions. If conflicts can be resolved, they must be done selectively by toggling the resource options in the Change Configuration utility. Any adapter that has a resource conflict that cannot be resolved by the Set Configuration program is automatically disabled (bit 0 of POS byte hex 0102 is set to 0).

If the error is a mismatch between installed devices and what the valid CMOS RAM data indicates is in the system, the user is given a choice as to whether or not to run Automatic Configuration.

The type of configuration to be done (automatic or selective) and error codes are passed to the Set Configuration program on the command line. The command line is built by the COMMAND.COM program on the Reference Diskette. The Set Configuration program (SC.EXE) is then loaded and executed. If the character "A" is the first character on the command line, an Automatic Configuration is done. The characters after "A" tell the Set Configuration program the type

of configuration errors that occurred. The following are valid error codes to pass to the Set Configuration program when the "A" option is being used:

61 Battery error
62 Configuration changed but no adapters changed or CRC error
64 Memory size during the POST does not match CMOS RAM
65 Card IDs in CMOS do not match the system.

Backup and Restore Configuration Utility

The Backup Configuration utility is provided for backing up configuration data to the Reference Diskette. Thus, in the event of a battery failure/change, the user does not have to repeat items changed using the Change Configuration utility, but merely runs the Restore Configuration utility.

Note: A copy of the Reference Diskette that is not write-protected is needed for this backup/restore process.

Copy an Option Diskette Utility

Adapter description files are merged onto the Reference Diskette with the Copy an Option Diskette utility. Adapter description files must be named in the format of @CARDID.adf, where CARDID is an ASCII representation of the card ID with the high byte (I/O address hex 0101) first. Diagnostic (*.dgs) code modules and POST error messages (*.pep) are merged at the same time as the adapter description (*.adf) files. The Option Diskette must be a DOS-formatted diskette.

Note: A copy of the Reference Diskette that is not write-protected is needed for this process.

Adapter Description Files

Adapter description files provide POS information and system resource usage information for Automatic Configuration. The adapter description files also provide text for System Configuration utilities, help screens, and prompts. This section provides the guidelines needed to develop the adapter description files.

Adapter Description File Format Information

- File names: @CARDID.adf (high byte of the Card ID first)
- Type of file: ASCII text
- Not case sensitive: Key words can be in either lowercase, uppercase, or mixed. The case is preserved within the user interface text strings.
- Blanks, tabs, new lines: These are considered as white space and ignored, except when in text strings for the Change Configuration user interface.
- Comments: Lines beginning with semicolons are comments and are ignored.

The following figure shows the meaning of special symbols used in the Adapter Description File syntax.

Symbol	Meaning
{ }	an optional item
{ }*	0,1,2,... items allowed
{ }+	1,2,3,... items allowed
x | y	either x or y allowed
x{n}	n x's required
[0-9]+	one or more decimal digits

Figure 2-34. Syntax Symbol Key

Syntax

```
adf_file => card_id card_name nbytes {fixed_resources} {named_item}*
```

This defines the contents of an ADF file. The following definitions describe each portion of an ADF file in detail.

```
card_id => AdapterId number
```

Each ADF file must contain a card_id. The character string 'AdapterId' is a keyword and must be present in the ADF file. The Configuration program looks for this ID, which must match the ID used in the filename.
Example: `AdapterId 0DEFFh`

```
card_name => AdapterName string
```

Each ADF file must contain a card_name. The character string 'AdapterName' is a keyword and must be present in the ADF file. The string following the 'AdapterName' keyword is displayed as the adapter name in the Change Configuration and View Configuration screens. The length of the AdapterName string is limited to (74 - (length of 'SlotX - ')) characters.
(US English length = 66 characters)
Example: `AdapterName "IBM Multi-Protocol Communications Adapter"`

```
nbytes => NumBytes number
```

Each ADF file must contain a nbytes. The character string 'NumBytes' is a keyword and must be present in the ADF file. This is used to define the number of POS bytes used by the adapter.
Example: `NumBytes 4`

```
fixed_resources => FixedResources pos_setting resource_setting
```

A fixed_resources is not a requirement for ADF files. It is used to define resources required by an adapter and corresponding POS data. The character string 'FixedResources' is a keyword and must be present in the ADF file only if the adapter needs to define resources that it requires.
Example: `FixedResources  POS[1]=XXXX01XXb int 3`

Figure 2-35. Adapter Description File Syntax (Part 1 of 5)

```
named_item => NamedItem prompt {named_choice}+  help
```

A named_item is not a requirement for ADF files. The named_item is used to define a field providing a choice of one or more resources. Each choice sets specified POS bits to a unique setting used to identify resources assigned to the adapter. The character string 'NamedItem' is a keyword and must be present in the ADF file only if the adapter can be configured to use different resources. The adapter determines the resources it is configured to by how the POS bytes are set. When a 'NamedItem' is defined in an ADF file it must be accompanied by a prompt (defined later), at least one named_choice (defined later), and help (defined later).
Example:

```
NamedItem
Prompt "Communications Port"
choice "SDLC_1"   pos[0]=XXX1000Xb  io 0380h-038ch  int 3 4
choice "SDLC_2"   pos[0]=XXX1001Xb  io 03a0h-03ach  int 3 4
choice "BISYNC_1" pos[0]=XXX1100Xb  io 0380h-0389h  int 3 4
choice "BISYNC_2" pos[0]=XXX1101Xb  io 03a0h-03a9h  int 3 4
choice "SERIAL_1" pos[0]=XXX0000Xb  io 03f8h-03ffh  int 4
choice "SERIAL_2" pos[0]=XXX0001Xb  io 02f8h-02ffh  int 3
choice "SERIAL_3" pos[0]=XXX0010Xb  io 3220h-3227h  int 3
choice "SERIAL_4" pos[0]=XXX0011Xb  io 3228h-322fh  int 3
choice "SERIAL_5" pos[0]=XXX0100Xb  io 4220h-4227h  int 3
choice "SERIAL_6" pos[0]=XXX0101Xb  io 4228h-422fh  int 3
choice "SERIAL_7" pos[0]=XXX0110Xb  io 5220h-5227h  int 3
choice "SERIAL_8" pos[0]=XXX0111Xb  io 5228h-522fh  int 3
Help
"This port can be assigned as a:  primary (SDLC1) or
secondary (SDLC2) sdlc, primary (BISYNC1) or secondary
(BISYNC2) bisync, or as a serial port (Serial 1 through Serial
8).  Use the F5=Previous and the F6=Next keys to change this
assignment in the 'Change configuration' window.  Conflicting
assignments are marked with an asterisk and must be changed
to use the adapter."
```

```
prompt => Prompt string
```

The prompt is required when a NamedItem is defined. The prompt is used to define a title for a NamedItem field. The character string 'Prompt' is a keyword and must be present in the ADF file when there is a NamedItem present. The string following the 'Prompt' keyword appears after the adapter name in the Change Configuration and View Configuration screens. Following the prompt string is a field that can be toggled in the Change Configuration screen if two or more named_items are defined. The length of the prompt string plus the length of the named_choice string cannot exceed 66 characters.
Example: (See the example for named_item).

Figure 2-36. Adapter Description File Syntax (Part 2 of 5)

```
named_choice => Choice choice_name pos_setting resource_setting
```

At least one named_choice is required when a NamedItem is defined. The character string 'Choice' is a keyword and must be present in the ADF file when there is a NamedItem present. One or more named_choices must follow a prompt. Each named_choice must contain a string that describes the current choice in the prompt field. Each named_choice must define a pos_setting, for at least one POS byte, which will uniquely identify the resource_setting defined in the named_choice. The length of the prompt string plus the length of the named_choice string cannot exceed 66 characters.
Example: (See the example for named_item).

```
help => Help string
```

The help is a string of text used to give the user assistance at a prompt. This text is displayed in the Change Configuration and View Configuration screen when the cursor is at the associated prompt and the F1 key is pressed. The character string 'Help' is a keyword and must be present in the ADF file when there is a NamedItem present. The string following the keyword 'Help' is the text that describes the prompt defined in the same NamedItem as the help. The length of the help string is limited to 1000 characters.
Example: (See the example for named_item).

```
pos_setting => {pos_byte_setting}+
```

The pos_setting must contain at least one pos_byte_setting. See the definition of pos_byte_setting for more information.

```
pos_byte_setting => pos[number]=pos_bit{8}b
```

This is the definition of the pos_byte_setting in the ADF file.
The character string 'pos' is a keyword and must be present in a pos_byte_setting, followed by a number in brackets. The number in brackets refers to the following POS bytes:

- number = 0, POS byte at port 102h
- number = 1, POS byte at port 103h
- number = 2, POS byte at port 104h
- number = 3, POS byte at port 105h
- number > 3, subaddressing data

The end bracket must be followed by an equal sign and then a bit definition of the POS byte (See pos_bit for information on the bit definition). The bit definition must define all 8 bits of the byte.
Bit 0 of pos[0] should always be defined as X.
Example: `pos[0]=XXX1001Xb`

Figure 2-37. Adapter Description File Syntax (Part 3 of 5)

```
pos_bit => x | X | 0 | 1
```

A pos_bit can be defined as a mask bit (x or X), a clear bit (0), or a set bit (1).
Example: (See the example for pos_byte_setting).

```
resource_setting =>
          {ioblock_list} {interrupt_list} {arb_list} {memaddr_list}
```

The resource_setting defines a list of system resources. They may be fixed resources that are required by the adapter or they may be resources that the adapter uses when configured to a specific choice in a named_item. The resources can consist of the following:

- Range of I/O addresses (limited to 16).
- List of interrupt levels (limited to 16).
- List of arbitration levels (limited to 16).
- Range of memory addresses (limited to 2).

Example: (See the following resource definitions).

```
ioblock_list => IO {range}+
```

The ioblock_list must be a list of one or more ranges of I/O addresses. The character string 'IO' is a keyword and must be present in the ioblock_list.
Example: `io 4220h-4227h`

```
Interrupt_list => INT {number}+
```

The interrupt_list must be a list of one or more interrupt levels. The character string 'INT' is a keyword and must be present in the interrupt_list
Example: `INT 3 4`

```
arb_list => ARB {number}+
```

The arb_list must be a list of one or more arbitration levels. The character string 'ARB' is a keyword and must be present in the arb_list.
Example: `ARB 1`

```
memaddr_list => MEM {range}+
```

The memaddr_list must be a list of one or more ranges of RAM or ROM addresses. The character string 'MEM' is a keyword and must be present in the memaddr_list.
Example: `MEM 0CC000h - 0CDFFFh`

```
range => number - number
```

Figure 2-38. Adapter Description File Syntax (Part 4 of 5)

```
number => [0-9]+ {d} |  [0-9a-f]+ h |  [0-9A-F]+ H

string => " [ascii except for "]+ "
```

A string is a set of ASCII characters that begins with a double quote (") and ends with a double quote.
Example:

```
"This port can be assigned as a:  primary (SDLC1) or
secondary (SDLC2) sdlc, primary (BISYNC1) or secondary
(BISYNC2) bisync, or as a serial port (Serial 1 through Serial
8).  Use the F5=Previous and the F6=Next keys to change this
assignment in the 'Change configuration' window.  Conflicting
assignments are marked with an asterisk and must be changed to
use the adapter."
```

Figure 2-39. Adapter Description File Syntax (Part 5 of 5)

Adapter Description File Example

This is an example of an Adapter Description File for the IBM Personal System/2 Multiprotocol Adapter/A. The name of the file for this adapter is @DEFF.adf. An explanation of each numbered item begins on page 2-62.

```
AdapterId 0DEFFh 1

AdapterName "IBM Multi-Protocol Communications Adapter" 2

NumBytes 2 3

NamedItem 4

Prompt "Communications Port"

choice "SDLC_1"   pos[0]=XXX1000Xb  io 0380h-038ch  int 3 4
choice "SDLC_2"   pos[0]=XXX1001Xb  io 03a0h-03ach  int 3 4
choice "BISYNC_1" pos[0]=XXX1100Xb  io 0380h-0389h  int 3 4
choice "BISYNC_2" pos[0]=XXX1101Xb  io 03a0h-03a9h  int 3 4
choice "SERIAL_1" pos[0]=XXX0000Xb  io 03f8h-03ffh  int 4
choice "SERIAL_2" pos[0]=XXX0001Xb  io 02f8h-02ffh  int 3
choice "SERIAL_3" pos[0]=XXX0010Xb  io 3220h-3227h  int 3
choice "SERIAL_4" pos[0]=XXX0011Xb  io 3228h-322fh  int 3
choice "SERIAL_5" pos[0]=XXX0100Xb  io 4220h-4227h  int 3
choice "SERIAL_6" pos[0]=XXX0101Xb  io 4228h-422fh  int 3
choice "SERIAL_7" pos[0]=XXX0110Xb  io 5220h-5227h  int 3
choice "SERIAL_8" pos[0]=XXX0111Xb  io 5228h-522fh  int 3

Help

"This port can be assigned as a: primary (SDLC1) or
secondary (SDLC2) sdlc, primary (BISYNC1) or secondary
(BISYNC2) bisync, or as a serial port (Serial 1 through Serial
8).  Use the F5=Previous and the F6=Next keys to change this
assignment.  Conflicting assignments are marked with an
asterisk and must be changed."
```

```
NamedItem 5

Prompt "Arbitration Level for SDLC"

choice "Level_1"   pos[1]=XXXX0001b  arb  1
choice "Level_0"   pos[1]=XXXX0000b  arb  0
choice "Level_2"   pos[1]=XXXX0010b  arb  2
choice "Level_3"   pos[1]=XXXX0011b  arb  3
choice "Level_4"   pos[1]=XXXX0100b  arb  4
choice "Level_5"   pos[1]=XXXX0101b  arb  5
choice "Level_6"   pos[1]=XXXX0110b  arb  6
choice "Level_7"   pos[1]=XXXX0111b  arb  7
choice "Level_8"   pos[1]=XXXX1000b  arb  8
choice "Level_9"   pos[1]=XXXX1001b  arb  9
choice "Level_10"  pos[1]=XXXX1010b  arb 10
choice "Level_11"  pos[1]=XXXX1011b  arb 11
choice "Level_12"  pos[1]=XXXX1100b  arb 12
choice "Level_13"  pos[1]=XXXX1101b  arb 13
choice "Level_14"  pos[1]=XXXX1110b  arb 14

Help

"This assignment need only be changed if it is in conflict
with another assignment.  Conflicting assignments are marked
with an asterisk.  Use the F5=Previous and the F6=Next keys
to change arbitration level assignments.  Using arbitration
levels, this adapter accesses memory directly without
burdening the computer's main microprocessor.  An
arbitration level of 0 has the highest priority, and
increasing levels have corresponding decreased priority"
```

1 The card_id for this adapter is hex 0DEFF. This is an ASCII representation of the ID generated by the adapter. The high byte is followed by the low byte. The card_id is required for all ADF files.

2 The card_name is "IBM Multi-Protocol Communications Adapter." The card_name is required for all ADF files.

3 The nbytes (NumBytes 2) in this file indicates the adapter uses two POS bytes located at hex 0102 and 0103.

4 This is the first named_item for the adapter. The title of the field is "Communications Port." The user can toggle between the 12 different named_choices. Each named_choice has a unique pos_setting assigned to it in bit locations 1 through 4 of POS byte hex 0102 (pos [0]). Also shown is a resource_setting that corresponds to the pos_setting of the named_choice. The resources allocated in this named_item are I/O addresses and interrupt levels. A help string for this named_item is provided below the last named_choice.

5 This is the second named_item for the adapter. The title of the field is "Arbitration Level for SDLC." The user can toggle between the 14 different named_choices. Each named_choice has a unique pos_setting assigned to it in bit locations 0 through 3 of POS byte hex 0103 (pos [1]). Also shown is a resource_setting that corresponds to the pos_setting of the named_choice. The resources allocated in this named_item are arbitration levels. A help string for this named_item is provided below the last named_choice.

Level-Sensitive Interrupt Sharing

The main objective of level-sensitive interrupt sharing is to:

- Simplify the logic-sharing design of adapters
- Reduce transient sensitivity of the interrupt controller
- Provide compatibility with existing software
- Allow for a mixture of sharing and nonsharing hardware on the same interrupt level.

All adapters designed for the Micro Channel employ a level-sensitive, active-low, interface mechanism. An open-collector driver (or tri-state driver gated active low) is required by each adapter to drive the interrupt request line for levels assigned for the adapter function.

Note: Designers may want to limit the number of devices sharing an interrupt level for performance and latency considerations.

An adapter must hold the level-sensitive interrupt active until it is reset as a direct result of servicing the interrupt (reset). Service routines must not attempt to end the interrupt sequence (EOI) until it has reset the interrupt line of the device being serviced. All adapters must also provide an interrupt pending latch that is readable at an I/O address bit position and reset by normal servicing of the device.

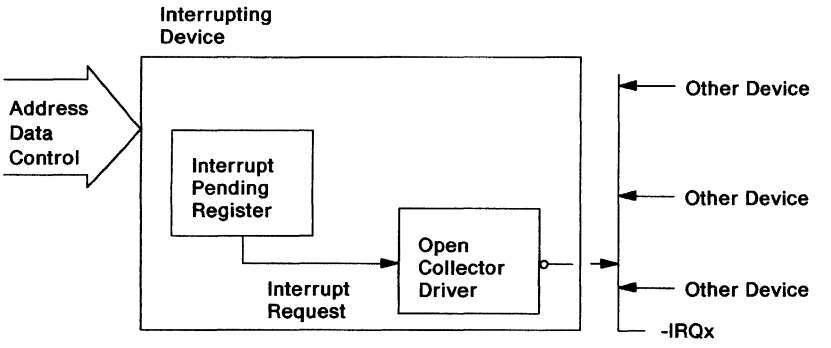


Figure 2-40. Typical Adapter Interrupt Sharing Implementation

Compatibility

To maintain software compatibility, the polling mechanism used by IBM Personal Computer products is retained. Software that interfaces to the reset port for the IBM Personal Computer positive-edge type of interrupt sharing[1] does not create interference (see "Hardware Interrupts" on page 9-6 for restrictions). Level-sensitive interrupt hardware allows several devices to set a common interrupt line active (low) simultaneously without interference.

Existing application code that deals directly with the interrupt controller may try to reset the controller to the positive edge-sensitive mode when exiting control. The interrupt control circuitry of the system board prevents setting the controller to the edge-sensitive mode by blocking positive edge-sensitive commands to the interrupt controllers.

Sequence of Operation

Level-sensitive interrupts are interlocked between the hardware and software that supports the interrupt service. Lost or spurious interrupts are more easily isolated. The following figure shows the sequence of interrupt sharing and the interaction of hardware and software.

---

[1] Hex address 02FX or 06FX, where X is the interrupt level.

Hardware Operation	Software Operation
1. An interrupt condition sets hardware interrupt line X to an active (low) level with an open collector driver, and sets an interrupt pending latch readable by code.	
2. An interrupt controller presents the interrupt to the system microprocessor.	
	3. The system microprocessor begins executing code at the beginning of the appropriate chain of interrupt handlers.
	4. The interrupt handler code reads the interrupt pending latch of the first card in the chain. If the latch is not pending, the next card in the chain is tested. When a reporting card is detected, the handler executes the appropriate service routine.
	5. The interrupt service routine operates the device hardware.
6. The adapter hardware resets the interrupt pending latch and the hardware interrupt line as a consequence of the interrupt service routine actions.	
	7. The interrupt service routine finishes executing code resetting the interrupt controller as its final action (End of Interrupt).
8. The interrupt controller resets.	
9. If an interrupt is pending (IRQ active by another device), the interrupt controller sets immediately and the sequence starts again.	

Figure 2-41. Interrupt Sharing Sequence

Central Arbitration Control Point

The Central Arbitration Control Point gives intelligent subsystems on the channel the ability to share and control the system. It allows burst data transfers and prioritization of control between devices. This arbiter supports up to 16 arbitrating devices, such as a DMA slave, a bus master and the system microprocessor.

The central arbiter uses seven signals to coordinate arbitration for all devices from a single arbitration point on the system board. These signals are -PREEMPT, ARB/-GNT, -BURST, and ARB0 through ARB3.

Local arbiters requesting use of the system channel, drive -PREEMPT active. The central arbiter initiates an arbitration cycle as soon as the present device releases the channel. The central arbiter indicates an arbitration cycle by driving ARB/-GNT to the arbitrate state. The requesting local arbiters then drive their assigned 4-bit arbitration level onto the arbitration bus. Any arbiter seeing a more significant bit low on the arbitration bus than those driven low by that arbiter, stops driving its lower order bits onto the arbitration bus. The device driving the lowest arbitration level thereby wins control of the system channel when ARB/-GNT goes to the grant state.

Devices desiring to perform multiple transfers must signal the central arbiter by driving -BURST active until all transfers have been completed or until another device drives the -PREEMPT signal active, in which case further transfers are postponed until the device wins the system channel again. Because -PREEMPT and ARB0 through ARB3 may be driven by multiple devices, they must be driven through an open collector driver. ARB/-GNT may only be driven by the central arbiter.

The central arbiter recognizes an End of Transfer when both status signals (-S0 and -S1) are inactive and -BURST or -CMD goes inactive, whichever occurs last. Control of the channel is then transferred to the next higher priority device or the system microprocessor by default.

A programmable fairness feature allows each device a share of channel time. If fairness is active and an arbitrating device that owns the channel is preempted, the device enters the Inactive State and

must wait for an inactive -PREEMPT before becoming active and competing for the channel again. This allows the system to service all arbitrating devices in order of priority before the same device can gain control of the channel again.

The following is a block diagram of central arbitration:

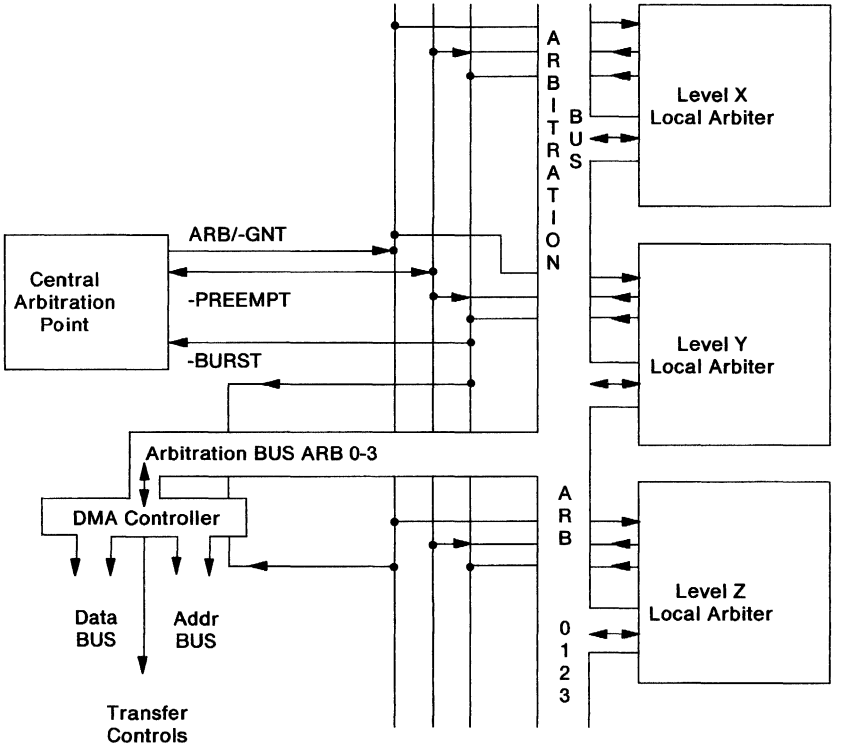


Figure 2-42. Central Arbitration

Note: The Central Arbiter is located on the system board. Local arbiters are located on other devices.

Local Arbiters

Devices requesting the use of the channel must implement logic to drive the arbitration bus in a manner that allows all competing devices to recognize the winner. This logic is known as a *Local Arbiter*. An arbitrating device should compete for control of the channel only if it has driven -PREEMPT active and subsequently ARB/-GNT has gone to the arbitrate state. A competing local arbiter drives its arbitration level onto the arbitration bus and compares its arbitration level with the value appearing on the arbitration bus on a bit-by-bit basis beginning with the most significant bit, ARB3. If the competing local arbiter detects a mismatch on one of the bits, it should cease to drive all lower-order bits immediately. If the local arbiter subsequently recognizes a match on that bit, it may continue driving lower-order bits until another mismatch is detected. Because the arbitration bus is driven by open collector drivers, multiple arbiters may safely drive the bus. The following is an example of bus arbitration:

1. Two devices with arbitration levels 1010 and 0101 (hex A and 5) compete for the channel. Both devices drive their arbitration levels on the bus that now appears as 0000.
2. The first device (1010) detects a mismatch on ARB3 and ceases to drive all other arbitration bits.
3. The second device (0101) detects a mismatch on ARB2 and ceases to drive arbitration bits. The arbitration bus now shows 0111.
4. The second device now sees a match on ARB2 and resumes driving bit 1 of the arbitration bus.
5. The arbitration bus now shows a value of 0101 and the second device wins control of the channel.

The following is a simplified example of a local arbiter.

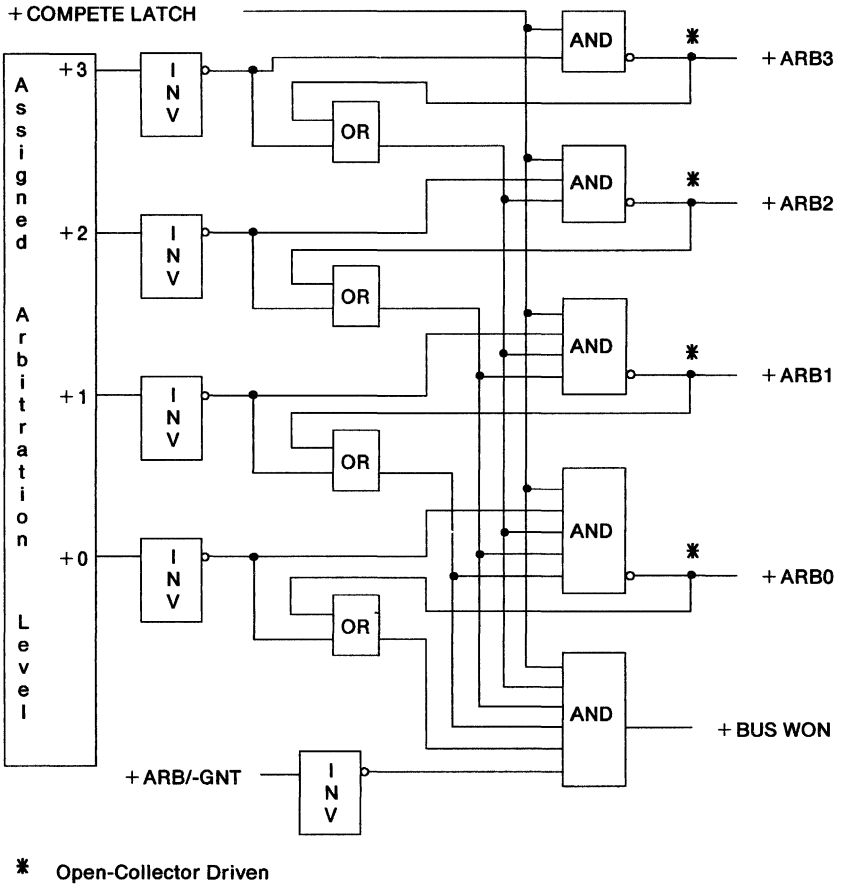


Figure 2-43. Local Arbiter Example

Burst Mode

Some devices, such as a fixed disk, transfer data in bursts, which are often separated by long inactive periods. The burst mode makes these devices more efficient.

In order to use burst mode, the local arbiter activates -BURST and holds it active. The local arbiter releases -BURST after the leading edge of the last -CMD pulse in the burst sequence. The following diagram shows a burst operation without interference:

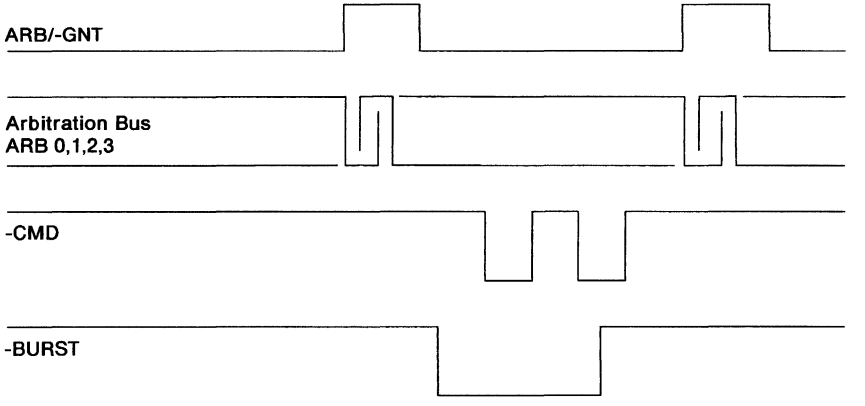


Figure 2-44. Burst Mode Timing

Preemption

Whenever an arbitrating device needs service, it activates -PREEMPT. The following timing diagram shows -PREEMPT occurring during a burst operation:

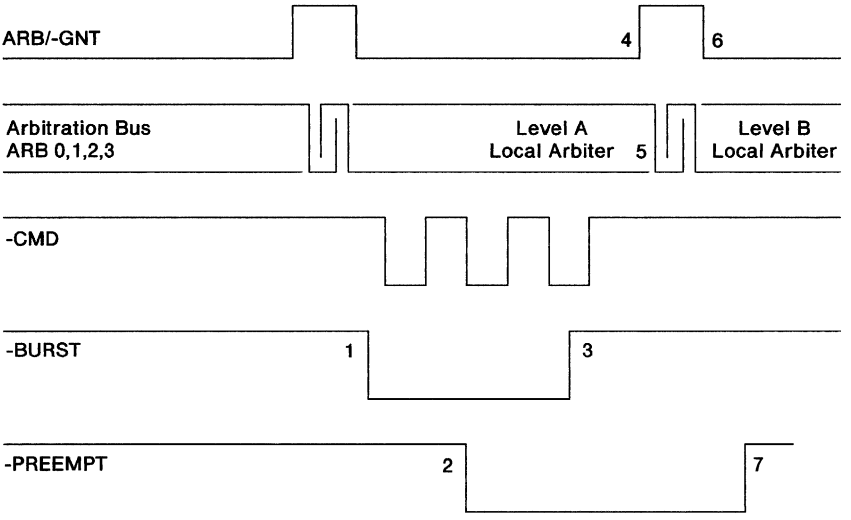


Figure 2-45. Preempt Timing

The sequence is as follows:

1. Burst mode arbitrating device A gains control of the channel.
2. Device B, nearing an overrun condition, requests preemption.
3. Device A, while still in control of the channel, completes any partial transfers and removes -BURST. Device A does not participate in the arbitration cycle if fairness is active.
4. When the central arbitration point recognizes the End of Transfer it removes the grant.
5. Arbitration for channel control begins.
6. When ARB/-GNT is in the grant state, the new arbiter gains control of the channel.
7. Device B, the preempting device, removes -PREEMPT in response to the grant.

If an attachment holds -BURST active more than 7.8 microseconds after an active -PREEMPT, an error condition may exist and a channel time-out may occur. The ARB/-GNT signal is driven high immediately,

forcibly taking channel control away from the channel owner. An NMI will be driven active, and bits 5 and 6 of port hex 0090 will be set active. The channel will remain in the arbitration state with the system microprocessor in control until bit 6 of port hex 0090 is reset.

Arbitration Bus Priority Assignments

The following figure shows the assignment of arbitration levels. The functions with the lowest number for its arbitration level has the highest priority.

ARB Level	Primary Assignment
-2	Memory Refresh
-1	NMI
0	DMA Channel 0 *
1	DMA Channel 1
2	DMA Channel 2
3	DMA Channel 3
4	DMA Channel 4 *
5	DMA Channel 5
6	DMA Channel 6
7	DMA Channel 7
8	Reserved
9	Reserved
A	Reserved
B	Reserved
C	Reserved
D	Reserved
E	Reserved
F	System Microprocessor

* These DMA Channels are programmable to any arbitration level.

Figure 2-46. Arbitration Bus Priority Assignments

Note: Devices designed for arbitration level 0 or 1 should have limited bandwidth or short bursts such that diskette overruns can be prevented or recovered by retry operations. The diskette drive controller on arbitration level 2 may be held inactive by devices on arbitration levels 0 and 1, refresh, and the previous bus owner (system microprocessor or other device). The diskette drive controller should not be held inactive for longer than 12 microseconds to prevent overrun.

NMI service is executed at a level higher than 0 called -1. Memory Refresh is prioritized at -2, 2 levels higher than 0. Levels -1 and -2 are reached on the system board only, while the 'arbitrate/-grant' signal is in the arbitrate state.

When the central arbitration point receives a level -1 request (NMI, a system board internal signal), it activates -PREEMPT, waits for End of Transfer, and then places ARB/-GNT in the arbitrate state which prevents channel activity from arbitrating devices. The central arbitration control point gives the grant to the level -1 request, and holds ARB/-GNT in the arbitrate state until the operation is complete and the NMI is reset.

Central Arbitration Programmable Options

The Central Arbitration Control point provides access to programmable options through the Arbitration register, which is accessed at I/O address hex 0090. The bits are defined differently for read and write operations. The following defines the Arbitration register.

Bit	Definition
7	Enable System Microprocessor Cycle
6	Arbitration Mask
5	Enable Extended Arbitration
4 - 0	Reserved = 0

Figure 2-47. Arbitration Register, Write Operations

Bit	Definition
7	Enable System Microprocessor Cycle
6	Arbitration Mask by NMI
5	Bus Timeout
4	Reserved = 0
3 - 0	Value of Arbitration Bus During Previous Grant State

Figure 2-48. Arbitration Register, Read Operations

Bit 7 A write, setting this bit to 1, enables system microprocessor cycles during arbitration cycles. This bit can be cleared, if an arbitrating device requires total control of the channel bandwidth. This bit is cleared on a system reset.

A read, with this bit set to 1, indicates system microprocessor cycles are enabled during arbitration.

Bit 6 A write, setting this bit, causes the central arbitration control point to enter an arbitration state. The system microprocessor controls the channel until this bit is reset. This bit can be reset either by a system reset or writing this bit equal to 0.

Warning: This bit should only be set to 1 by diagnostics and system error recovery.

A read, with this bit set, indicates that an NMI has occurred and has masked arbitration.

Bit 5 A write, setting this bit to a 1, enables extended arbitration. With this bit set, the minimum arbitration cycle is 750 nanoseconds, as opposed to 375 nanoseconds, when this function is disabled. The default is disabled.

A read, with this bit set to 1, indicates that a bus time-out has occurred. This bit is reset for a read by clearing bit 6, of port hex 90, to 0. This bit is set to 0 during a system reset.

Bit 4 Reserved

Bits 3 - 0 These bits are undefined for a write.

A read of these bits returns the arbitration level of the arbiter controlling the channel during the most recent grant state. This information allows the system microprocessor to determine the arbitration level of the arbitrating device that caused a bus time-out.

Channel Support

Unique logic is required for the 32-bit bus to permit 16-bit data masters on the channel to communicate with 32-bit data slaves.

Address Bus Translator

A 32-bit slave uses the 'byte enable' bus lines (-BE0 through -BE3) as part of its address rather than A0 and -SBHE. A 16-bit master does not provide these four 'byte enable' signals; the system generates these signals when a 16-bit master has control of the bus.

Data Bus Steering

32-bit slaves write and read data to and from data bits 16 through 23 and 24 through 31. A 16-bit controlling device does not use these data lines; the system must cross data over from the low 16 data lines (D0 through D15) to the high data lines (D16 through D31) and back at the appropriate times.

A0	-SBHE	Description
0	1	Byte Zero Only (D0 - D7)
1	0	Byte One Only (D8 - D15)
0	0	Byte Zero and Byte One (D0 - D15)
1	1	Invalid

Figure 2-49. Steering Control

TR 32: This signal is driven inactive by 32-bit controlling devices only. When TR 32 is active, it is used by:

- The Central Translator logic to drive -BE0 through -BE3
- The Central Steering logic to know to perform bus steering.

When TR 32 is active, it can be used by 32-bit slaves to recognize the master is not 32-bit and compensation may be required for additional delay attributable to the Central Steering logic.

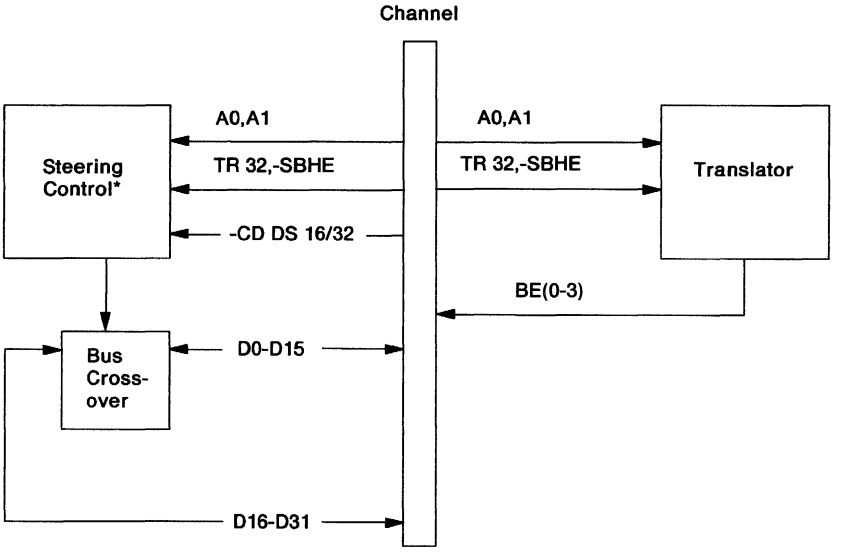
Central Steering Logic: Central Steering logic uses A0, A1, -SBHE, -CD DS 16, and -CD DS 32 to steer data in support of 16-bit masters communicating with 32-bit slaves.

Central Translator Logic: Central Translator logic translates A0, A1, and -SBHE to -BE0 through -BE3 respectively, when TR 32 is active.

-BE(0-3): Signals -BE0 through -BE3 are:

- Driven by 32-bit masters when they have control of the bus
- Created by the Central Translator logic when a 16-bit or 8-bit master has the bus
- Used only by 32-bit slaves.

The following block diagram shows the implementation of Data Bus Steering.



* For 16-bit devices to 32-bit devices

Figure 2-50. Data Bus Steering Implementation

Micro Channel Critical Timing Parameters

This section provides timing diagrams for the Micro Channel. All timings are related to a nominal cycle. Systems and adapters may alter this nominal cycle through various mechanisms. Developers should use care that hardware and software designs operate over the ranges specified and do not depend on a given performance level. Portability may be adversely affected by time-dependent hardware and software.

Basic Transfer Cycle

This section provides the specification for critical timing parameters for the Basic Transfer cycle.

Simplified Basic Transfer Cycle

Most microprocessor and DMA operations transfer data with the same control sequence. For transfers other than Matched Memory transfers, the signals appear on the channel in the following sequence:

1. Address bus, MADE 24, M/-IO, and -REFRESH (if applicable) become valid, beginning the cycle.
2. The 'status' signals become valid.
3. The 'address decode latch' (-ADL) signal becomes valid.
4. In response to an unlatched address decode, MADE 24, and M/-IO, the adapter returns:
   - -CD SFDBK
   - -CD DS 16 (if the attachment is capable of 16-bit operation).
   - -CD DS 16 and -CD DS 32 (if the attachment is capable of 32-bit operation).
5. In response to an unlatched address decode, MADE 24, M/-IO, and Status, the adapter drives CH RDY inactive if the cycle is to be extended.
6. Write data appears on the bus (for the Write cycle).
7. -CMD becomes active and -ADL become inactive.

8. The 'status' signals become inactive.
9. The 'address' signals become invalid in preparation for the next cycle.
10. In response to an address change:
    - -CD SFDBK is set inactive by the attachment.
    - -CD DS 16 is set inactive by the attachment.
    - -CD DS 32 is set inactive by the attachment.
11. If CD CHRDY has been set inactive, the system holds in this state indefinitely until CD CHRDY is set active. This line should not be held inactive longer than 3.0 microseconds.
12. The attachment places Read data on the bus in preparation for the trailing edge of -CMD (for the Read cycle).
13. The address, 'status' signals, and M/-IO for the next cycle may become valid.
14. -CMD goes inactive, ending the cycle.

Note: The address and Status can be overlapped with the preceding cycle to minimize the memory access time impact on performance.

The sequence is as follows:

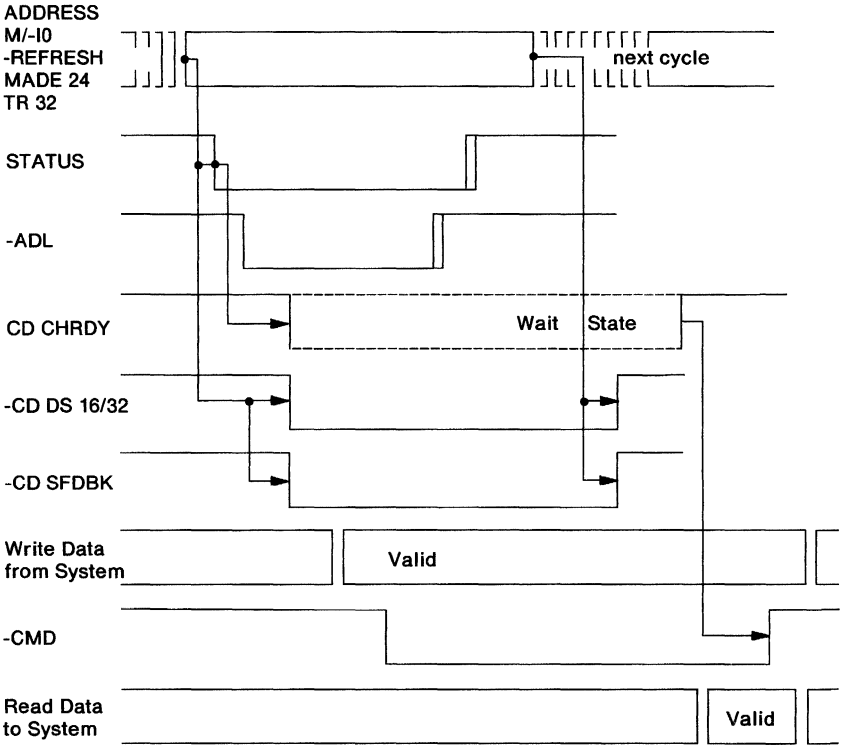


Figure 2-51. Overview of the Basic Transfer Cycle

Notes:

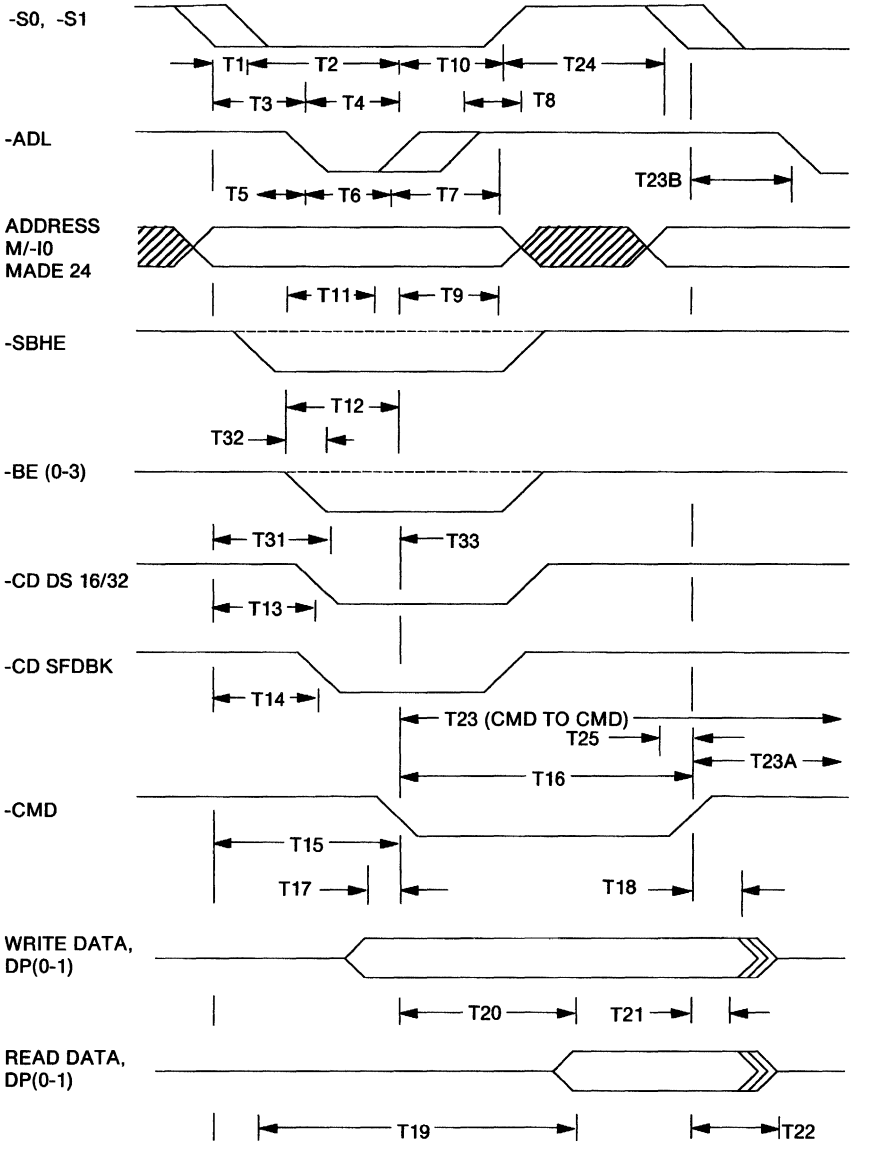
I/O and Memory Cycle

The basic I/O and memory cycle timing diagrams are shown on the following pages. Matched Memory cycle timing diagrams begin on page 2-110. The basic I/O and memory cycle timing diagrams appear in the order listed below.

- Default cycle (200 nanoseconds minimum)
- Synchronous Extended cycle (300 nanoseconds minimum) - Special case
- Asynchronous Extended cycle ($\geq$300 nanoseconds minimum) - General case.

A Synchronous or an Asynchronous Extended cycle is caused by a slave depending upon its use of CD CHRDY. In general, a slave releases CD CHRDY asynchronously causing an Asynchronous Extended cycle.

Default Cycle



	Timing Parameter	Min/Max	Note
T1	Status active (low) from ADDRESS,M/-IO,-REFRESH valid	10 / - ns	
T2	-CMD active (low) from Status active (low)	55 / - ns	2
T3	-ADL active (low) from ADDRESS,M/IO,-REFRESH valid	45 / - ns	
T4	-ADL active (low) to -CMD active (low)	40 / - ns	
T5	-ADL active (low) from Status active (low)	12 / - ns	
T6	-ADL pulse width	40 / - ns	
T7	Status hold from -ADL inactive (high)	25 / - ns	2
T8	ADDRESS,M/-IO,-REFRESH,-SBHE hold from -ADL inactive	25 / - ns	2
T9	ADDRESS,M/-IO,-REFRESH,-SBHE hold from -CMD active (low)	30 / - ns	3
T10	Status hold from -CMD active	30 / - ns	2
T11	-SBHE setup to -ADL inactive	40 / - ns	2
T12	-SBHE setup to -CMD active	40 / - ns	2
T13	-CD DS 16/32 active (n) (low) from ADDRESS,M/-IO,-REFRESH valid	- / 55 ns	3
T14	-CD SFDBK active (low) from ADDRESS,M/-IO,-REFRESH valid	- / 60 ns	1
T15	-CMD active (low) from Address valid	85 / - ns	2
T16	-CMD pulse width	90 / - ns	
T17	Write data setup to -CMD active (low)	0 / - ns	
T18	Write data hold from -CMD inactive (high)	30 / - ns	
T19	Status to Read Data valid (Access Time)	- / 125 ns	
T20	Read data valid from -CMD active (low)	- / 60 ns	
T21	Read data hold from -CMD inactive (high)	0 / - ns	
T22	Read data bus tri-state from -CMD inactive (high)	- / 40 ns	
T23	-CMD active to next -CMD active	190 / - ns	4
T23A	-CMD inactive to next -CMD active	80 / - ns	
T23B	-CMD inactive to next -ADL active	40 / - ns	
T24	Next Status active (low) from Status inactive	30 / - ns	
T25	Next Status active (low) to -CMD inactive	- / 20 ns	
T31	-BE(0-3) active from Address valid (32-bit masters only)	- / 40 ns	
T32	-BE(0-3) active from -SBHE, A0, A1 active	- / 30 ns	
T33	-BE(0-3) active to -CMD active	10 / - ns	

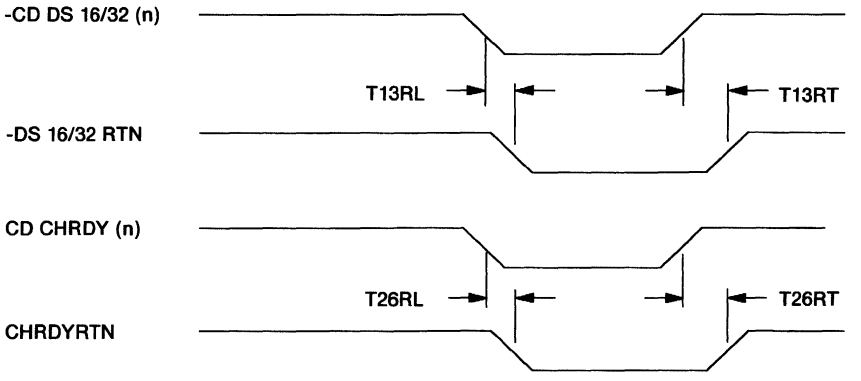
Figure 2-52. I/O and Memory Default Cycle (200 nanoseconds minimum)

Notes:

1. All slaves must drive -CD SFDBK whenever selected either by the system microprocessor or the DMA Controller. The slaves do not drive -CD SFDBK when they are selected by the 'setup' signal.
2. It is recommended that slaves use transparent latches to latch information with the leading or trailing edge of -ADL or with the leading edge of -CMD.
3. -CD DS 16/32 and -CD SFDBK must be driven by *unlatched address decodes* because the next address may come early into the current cycle.

4. Any master in any system, including the system microprocessor or DMA controller, can operate at a performance less than the level specified. Designers should not design to a given performance level as this level can be reduced by CD CHRDY, a lower microprocessor rate, a lower DMA controller rate, or system contention.

Default Cycle Return Signals



	Timing Parameter	Min/Max	Note
T13RL	-CD DS 16/32 (n) low to -DS 16/32 RTN low	- / 20 ns	1
T13RT	-CD DS 16/32 (n) high to -DS 16/32 RTN high	- / 20 ns	1
T26RL	CD CHRDY (n) low to CHRDYRTN low	- / 20 ns	2
T26RT	CD CHRDY (n) high to CHRDYRTN high	- / 20 ns	3

Figure 2-53. Default Cycle Return Signals (200 nanoseconds minimum)

Notes:

1. This signal is developed from a negative OR of signals received from each channel connector.
2. This signal is developed from a positive AND of signals received from each channel connector. This is a change from other systems using the Micro Channel architecture.
3. This signal is developed from a positive AND of signals received from each channel connector.

Synchronous Special Case of Extended Cycle

A Synchronous Extended cycle occurs when a slave releases CD CHRDY synchronously within the specified time after the leading edge of -CMD. The slave provides the Read data within a specified time from -CMD. The timing sequence is illustrated by the following figure.

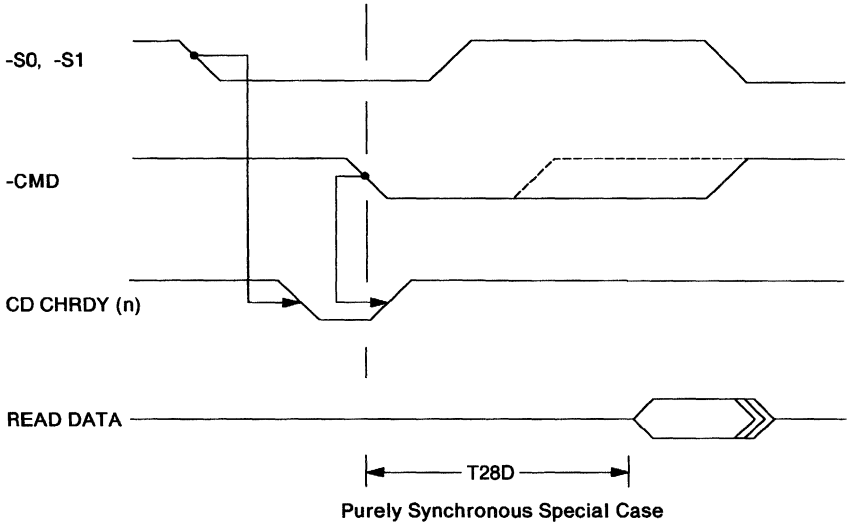
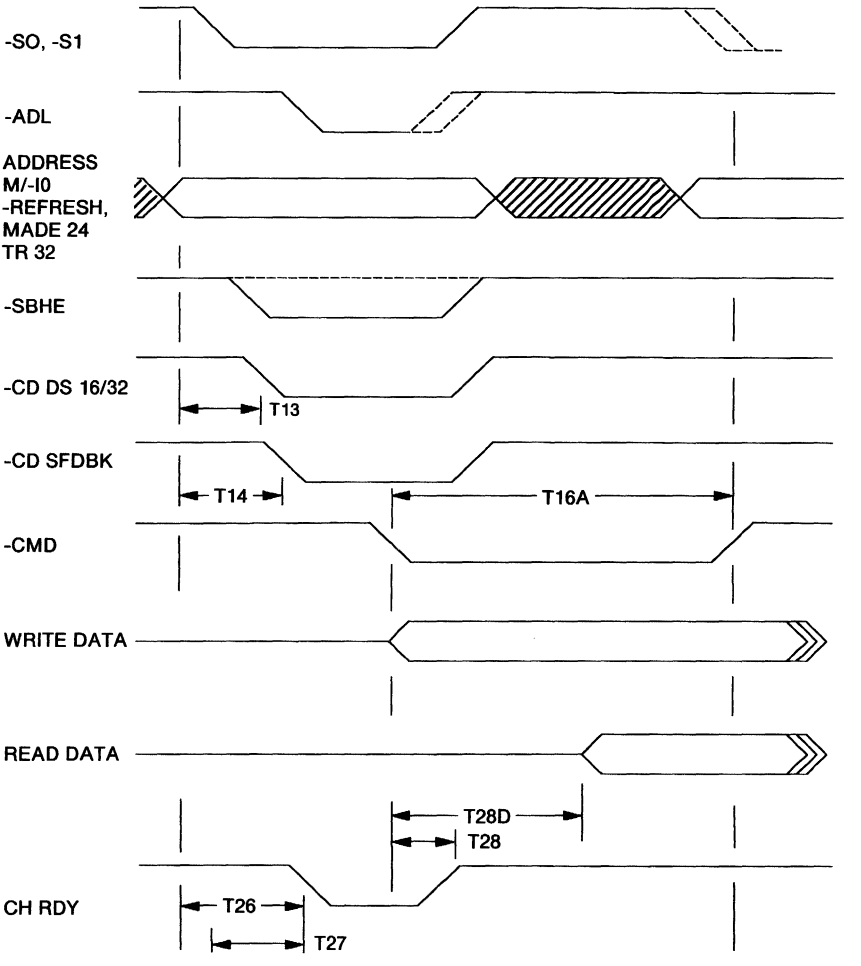


Figure 2-54. Timing Sequence for the Synchronous Special Case of Extended Cycle

Synchronous Extended Cycle (300 nanoseconds minimum - Special Case)



	Timing Parameter	Min/Max	Note
T13	-CD DS 16/32 (n) active (low) from ADDRESS,M/-IO,-REFRESH valid	- / 55 ns	2
T14	-CD SFDBK (n) active (low) ADDRESS,M/-IO,-REFRESH valid	- / 60 ns	2
T16A	-CMD pulse width	190 / - ns	
T26	CD CHRDY (n) inactive (low) from ADDRESS valid	- / 60 ns	3, See T27
T27	CD CHRDY (n) inactive (low) from Status active	0 / 30 ns	3
T28	CD CHRDY (n) release (high) from -CMD active (low)	0 / 30 ns	1
T28D	Read Data valid from -CMD active (when used along with T28)	0 / 160 ns	1

This figure shows only the parameters additional to the default cycle. All other parameters are the same as the default cycle.

Figure 2-55. Synchronous Extended Cycle (300 nanoseconds minimum - Special Case)

Notes:

1. CD CHRDY is released by a slave performing a 300 nanoseconds extended cycle synchronous with the leading edge of -CMD. Since CD CHRDY is generally an asynchronous signal, this is referred to as a purely synchronous special case.
2. This is the same as default cycle timing (listed here for emphasis).
3. T27 is valid only when Status becomes active 30 nanoseconds or more after the address is valid.
4. If Status overlaps with previous -CMD, then CD CHRDY state is not valid during the overlapped period.
5. Slaves must not hold CD CHRDY inactive (low) in excess of 3.0 microseconds.

Notes:

Asynchronous Extended Cycle (General Case)

An Asynchronous Extended cycle occurs when a slave releases CD CHRDY asynchronously. However, the slave provides the Read data within the specified time from CD CHRDY release. The timing sequence is illustrated by the following figure.

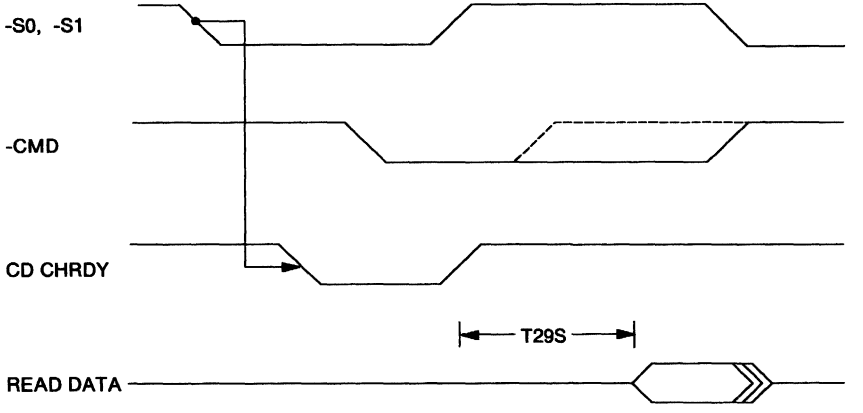
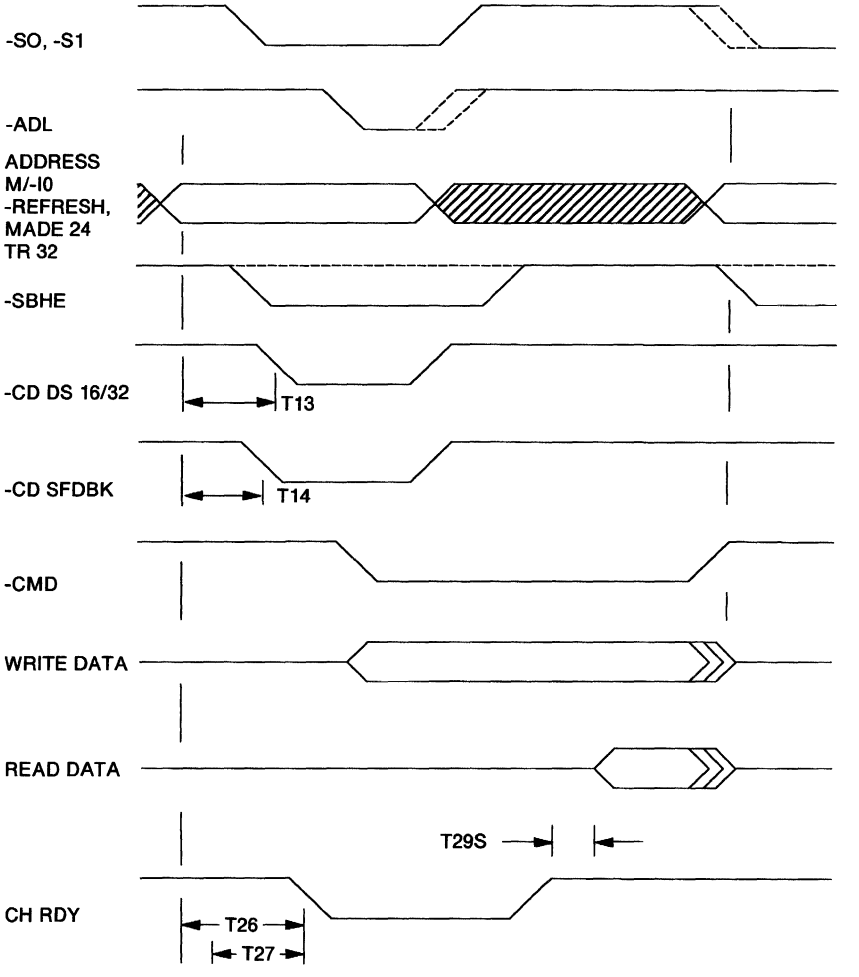


Figure 2-56. Timing Sequence for the Asynchronous Extended Cycle (General Case)

Asynchronous Extended Cycle (≥300 nanoseconds minimum - General Case)



	Timing Parameter	Min/Max	Note
T13	-CD DS 16/32 (n) active (low) from ADDRESS,M/-IO,-REFRESH valid	- / 55 ns	2
T14	-CD SFDBK (n) active (low) ADDRESS,M/-IO,-REFRESH valid	- / 60 ns	2
T26	CD CHRDY (n) inactive (low) from ADDRESS valid	- / 60 ns	See T27
T27	CD CHRDY (n) inactive (low) from Status active	0 / 30 ns	
T29S	Read data from slave valid from CD CHRDY (n) active (high)	- / 60 ns	1

This figure shows only the parameters additional to the default cycle. All other parameters are the same as the default cycle.

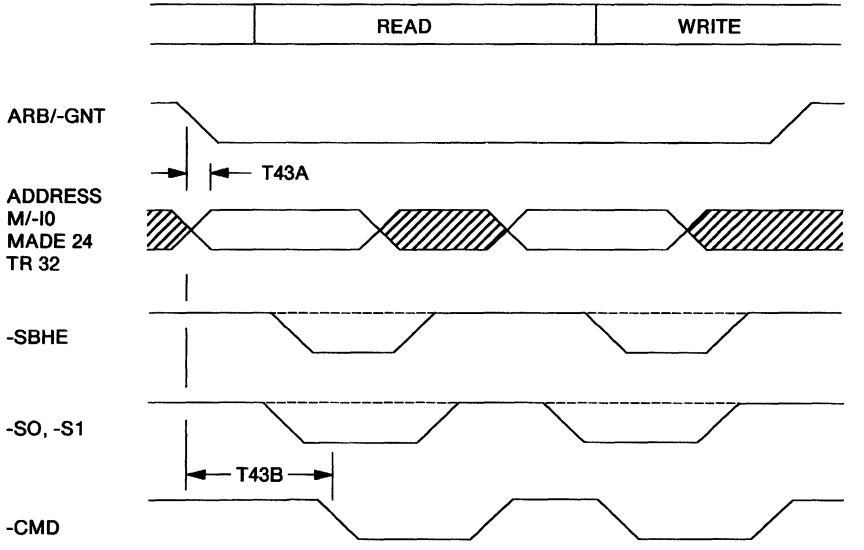
Figure 2-57. Asynchronous Extended Cycle (≥300 nanoseconds minimum - General Case)

Notes:

1. CD CHRDY is released asynchronously by a slave performing 300 nanoseconds minimum cycle. The slave must present the Read data within the time specified after the release of CD CHRDY.
2. This is the same as default cycle timing.
3. T27 is valid only when Status becomes active 30 nanoseconds or more after the address is valid.
4. If Status overlaps with the previous -CMD, then the CD CHRDY state is not valid during the overlapped period.
5. Slaves must not hold CD CHRDY inactive (low) in excess of 3.0 microseconds.

DMA Timing

First Cycle After Grant

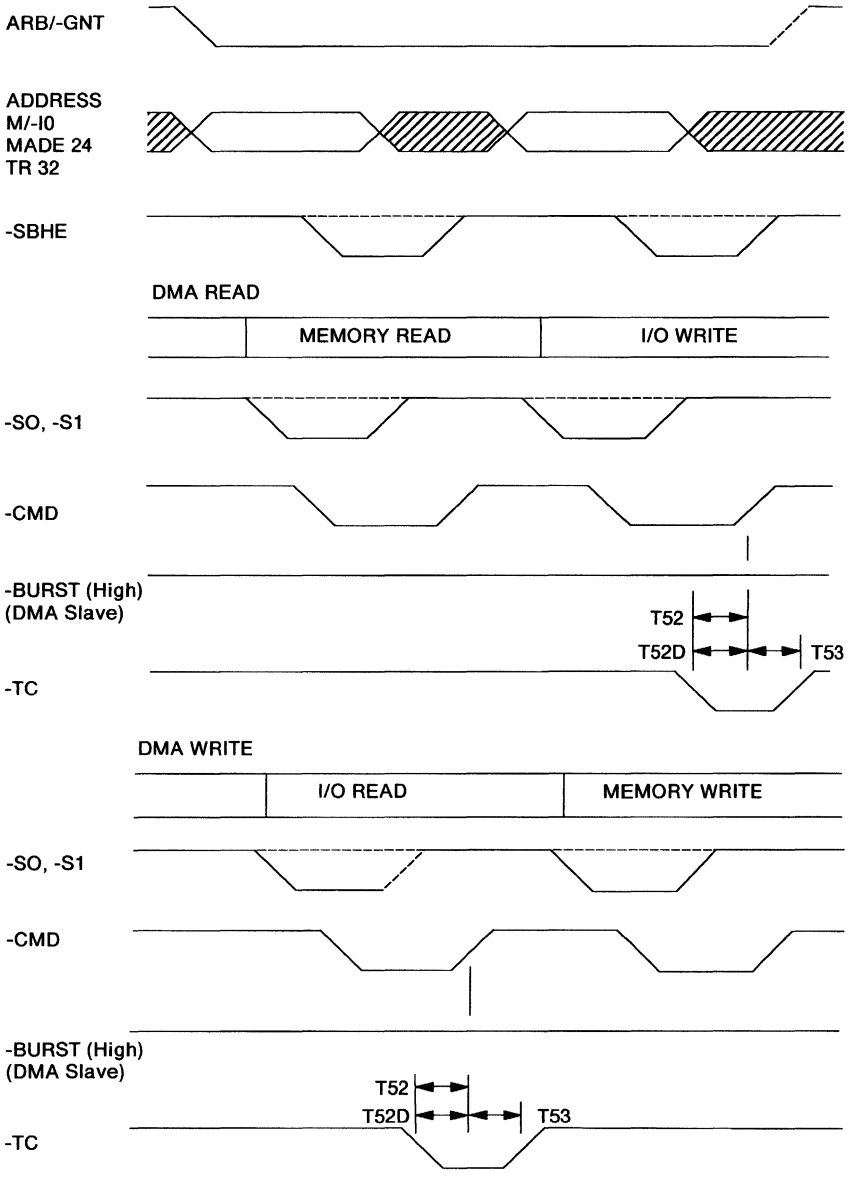


	Timing Parameter	Min/Max	Note
T43A	ADDR valid from ARB/-GNT low	0 / - ns	
T43B	-CMD active from ARB/-GNT low	115 / - ns	

Figure 2-58. First Cycle After Grant

Note: A controller must allow 30 nanoseconds after the grant, for a slave to generate an internal acknowledgment that it has been selected. During the first cycle, the controller must additionally allow 30 nanoseconds before sampling -CD DS 16/32, -CD SFDBK, and CD CHRDY, if it places an address on the bus within 30 nanoseconds after grant. However, if the controller places the address on the bus 30 nanoseconds after grant, this 30 nanoseconds allowance is not needed.

Single DMA Transfer (DMA Controller Controlled)



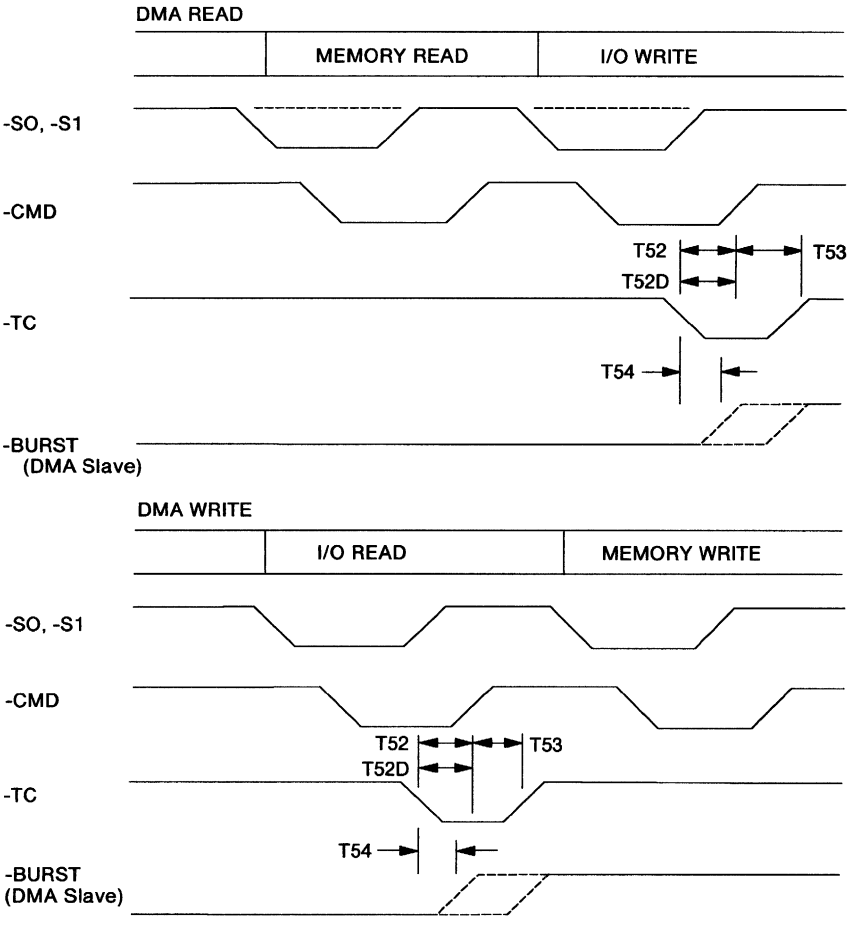
	Timing Parameter	Min/Max	Note
T52	-TC setup to -CMD inactive	30 / - ns	
T52D	-TC setup to -CMD inactive	15 / - ns	2
T53	-TC hold to -CMD inactive	10 / - ns	

Figure 2-59. Single DMA Transfer (DMA Controller Controlled)

Notes:

1. Only those timing parameters additional to those specified for the Basic Transfer cycle are included here.
2. Only for devices using a 200-nanosecond minimum default cycle.

Burst DMA Transfer (DMA Controller Terminated)



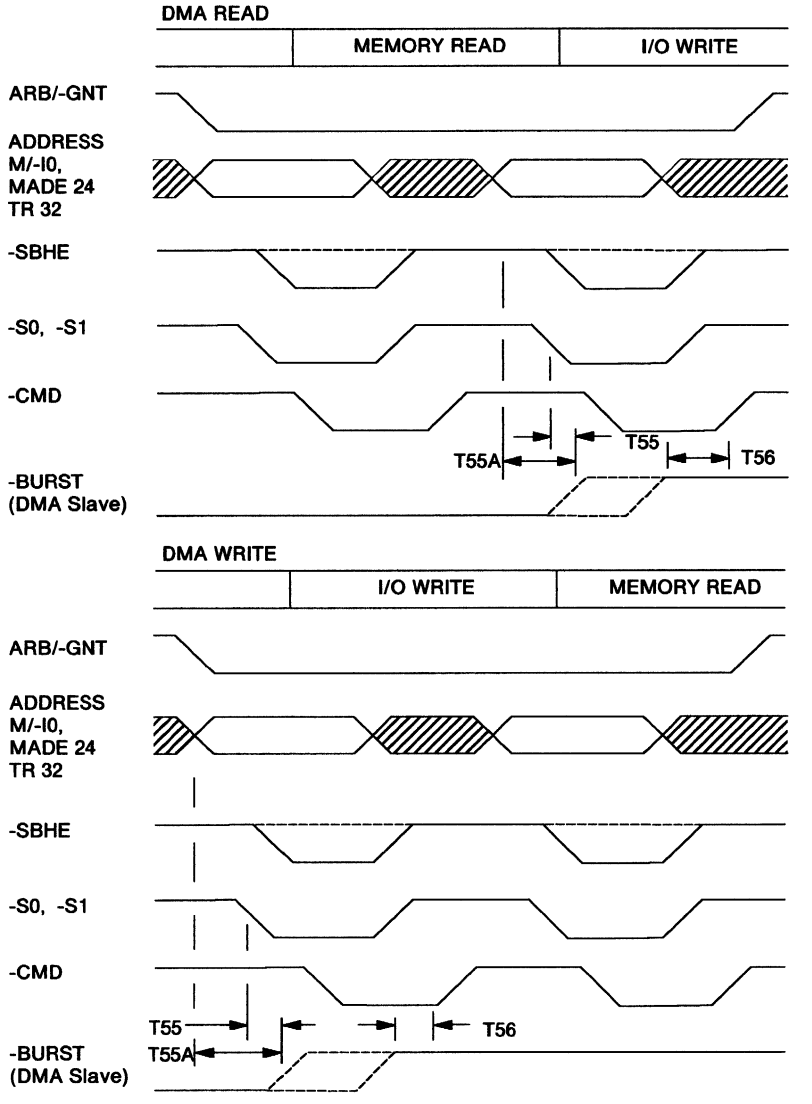
	Timing Parameter	Min/Max	Note
T52	-TC setup to -CMD inactive	30 / - ns	
T52D	-TC setup to -CMD inactive	15 / - ns	2
T53	-TC hold from -CMD inactive	10 / - ns	
T54	-BURST released by the DMA slave from -TC active	- / 30 ns	

Figure 2-60. Burst DMA Transfer (DMA Controller Terminated)

Notes:

1. Only those timing parameters additional to those specified for the Basic Transfer cycle are included here.
2. Only for devices using a 200-nanosecond minimum default cycle.

Burst DMA Transfer (DMA Slave Terminated - Default Cycle 200 nanoseconds)



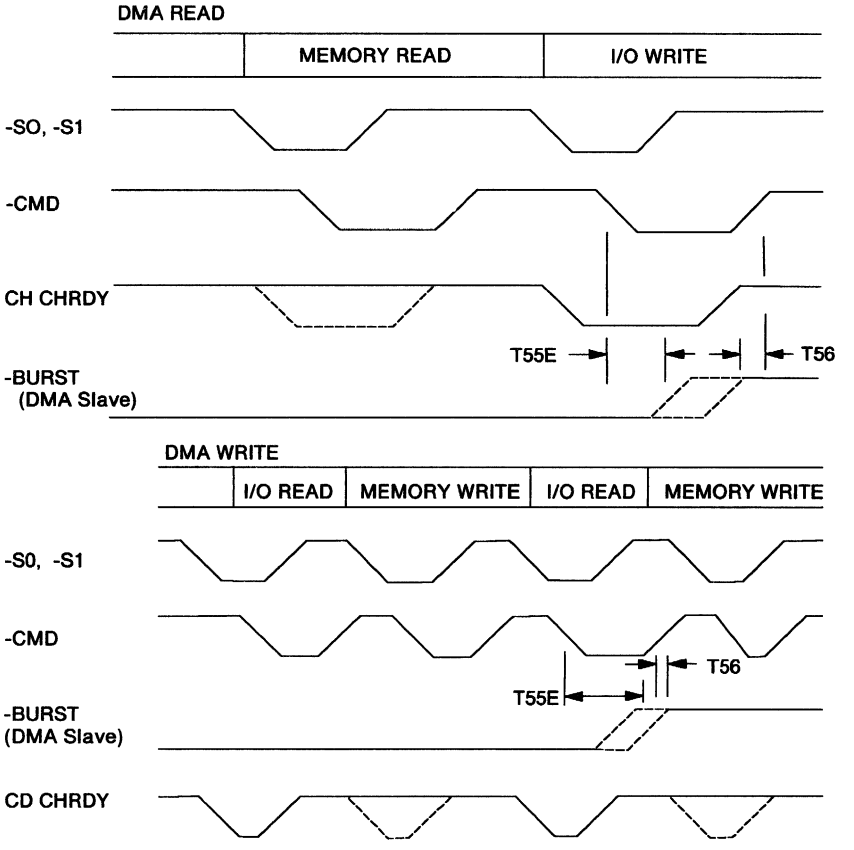
	Timing Parameter	Min/Max	Note
T55	-BURST released by the DMA slave from the last I/O Status active (Default cycle only)	- / 40 ns	2
T55A	-BURST released by the DMA slave from the last I/O ADDRESS valid (Default cycle only)	- / 70 ns	2
T56	-BURST inactive (high) setup to -CMD inactive	35 / - ns	3

Figure 2-61. Burst DMA Transfer (DMA Slave Terminated - Default Cycle 200 nanoseconds)

Notes:

1. Only those timing parameters additional to those specified for the Basic Transfer cycle are included here.
2. After releasing -BURST and on receiving -SBHE, if the DMA slave has another cycle to perform, it must redrive -BURST.
3. -BURST (high) setup time (T56) to the end of -CMD must be guaranteed during the last I/O Write cycle to prevent the DMA Controller from starting the next cycle. This setup time (T56) is guaranteed by the sum of -BURST release (T55/T55A) by the DMA slave and the -BURST resistor capacitor (RC) restoration time. The RC restoration time must not exceed 70 nanoseconds. Note that T56 is the same for Default and Extended cycles.

Burst DMA Transfer (DMA Slave Terminated - Synchronous Extended Cycle 300 nanoseconds)



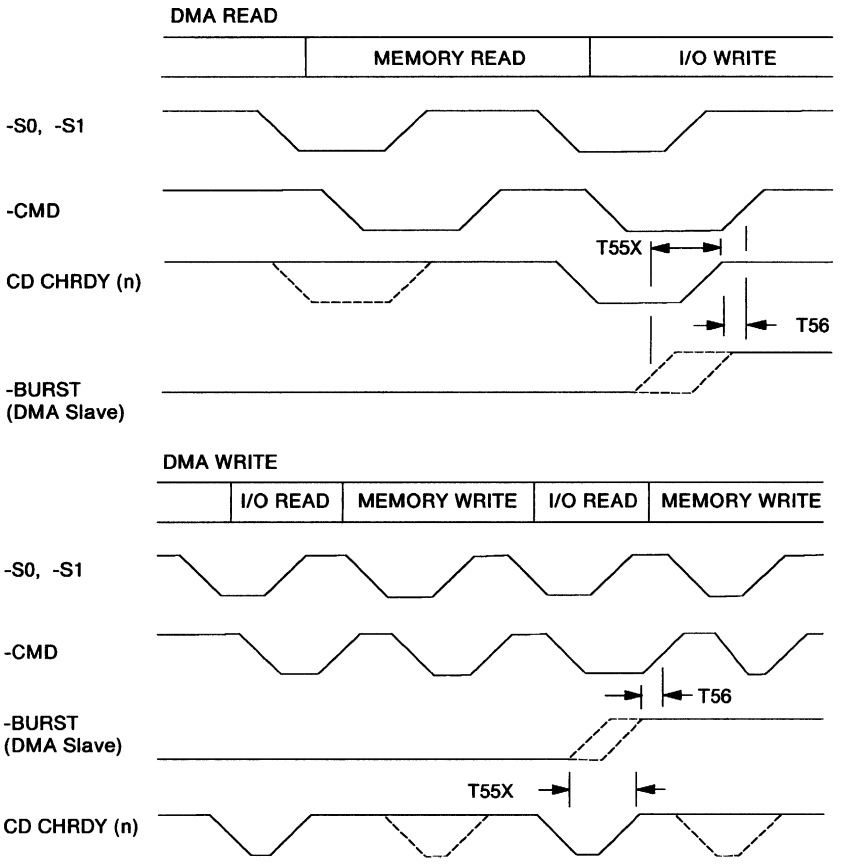
	Timing Parameter	Min/Max	Note
T55E	-BURST released by the DMA slave from the last -CMD active (Extended cycles only)	- / 80 ns	
T56	-BURST inactive (high) setup to -CMD inactive	35 / - ns	2

Figure 2-62. Burst DMA Transfer (DMA Slave Terminated - Synchronous Extended Cycle 300 nanoseconds)

Notes:

1. Only those timing parameters additional to those specified for the Basic Transfer cycle are included here.
2. -BURST (high) setup time (T56) to the end of -CMD must be guaranteed during the last I/O Write cycle to prevent the DMA Controller from starting the next cycle. This setup time (T56) is guaranteed by the sum of -BURST release (T55E) by the DMA slave and the -BURST RC restoration time. The RC restoration time must not exceed 70 nanoseconds. Note that T56 is the same for Default and Extended cycles.

Burst DMA Transfer (DMA Slave Terminated - Asynchronous Extended Cycle ≥300 nanoseconds)



	Timing Parameter	Min/Max	Note
T55X	-BURST released by the DMA slave before CD CHRDY (n) active (high) (Async. Extended cycles only)	50 / - ns	
T56	-BURST inactive (high) setup to -CMD inactive	35 / - ns	2

Figure 2-63. Burst DMA Transfer (DMA Slave Terminated - Asynchronous Extended Cycle ≥300 nanoseconds)

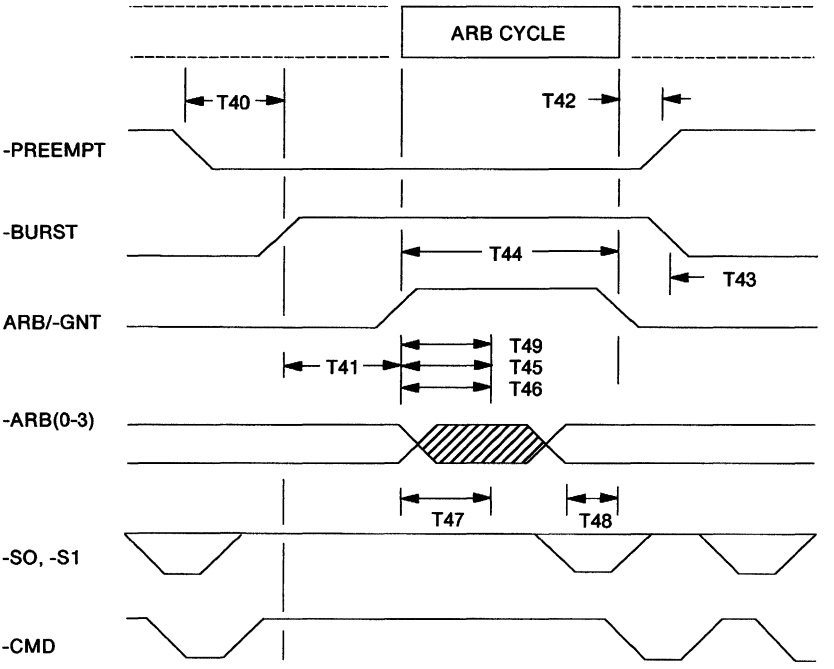
Notes:

1. Only those timing parameters additional to those specified for the Basic Transfer cycle are included here.
2. -BURST (high) setup time (T56) to the end of -CMD must be guaranteed during the last I/O Write cycle to prevent the DMA Controller from starting the next cycle. This setup time (T56) is guaranteed by the sum of -BURST release (T55X) by the DMA slave and the -BURST RC restoration time. The RC restoration time must not exceed 70 nanoseconds. Note that T56 is the same for Default and Extended cycles.

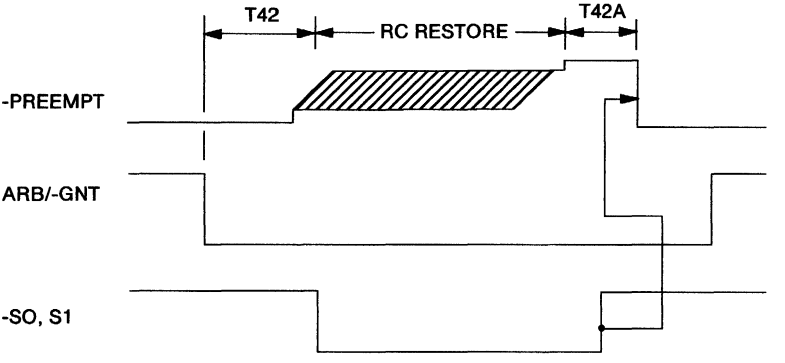
Arbitration Timing

This section provides the specification for critical timing parameters for arbitration protocol.

Arbitration Cycle



Exiting Inactive State



	Timing Parameter	Min/Max	Note
T40	-PREEMPT active (low) to End of Transfer	0 / 7.8 μs	1
T41	ARB/-GNT high from End of Transfer	30 / - ns	6
T42	-PREEMPT inactive (high) from ARB/-GNT low	0 / 50 ns	
T42A	-PREEMPT inactive (high) to Status inactive (Exiting Inactive State)	20 / - ns	5
T43	-BURST active (low) from ARB/-GNT low (By Bursting DMA slave)	- / 50 ns	4
T44	ARB/-GNT high	300 / - ns	2
T45	Driver turn-on delay from ARB/-GNT high	0 / 50 ns	3
T45A	Driver turn-on delay from lower priority line	0 / 50 ns	3
T46	Driver turn-off delay from ARB/-GNT high	0 / 50 ns	3
T47	Driver turn-off delay from higher priority line	0 / 50 ns	3
T48	Arbitration bus stable before ARB/-GNT low	10 / - ns	
T49	Tri-state drivers from ARB/-GNT high	- / 50 ns	

Figure 2-64. Arbitration Cycle

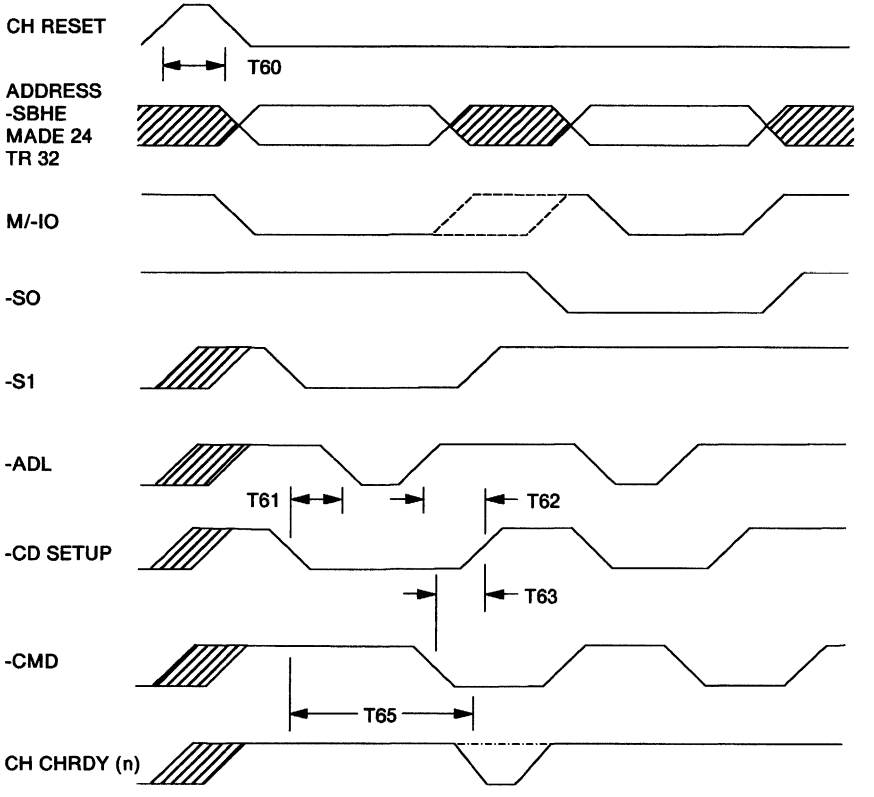
Notes:

1. The intent of this parameter is to limit the maximum non-preemptive ownership of the bus.
2. The value shown applies only to the special case implementation involving the central arbitration control point and is provided for pulse width and portability considerations only. Arbitration can be extended by refresh or error recovery procedures. An arbiter should decode a win of the grant by a combined decode of the arbitration bus and the ARB/-GNT. The minimum arbitration time can be 100 nanoseconds when a level 0 arbiter and the central arbitration control point coordinate. In this special case, a central arbitration control point can terminate arbitration prematurely at 100 nanoseconds.
3. T45, T45A, T46, and T47 must be satisfied by ARB (0-3) drivers of all arbitrating bus participants.
4. This parameter is applicable to all bus winners.
5. This represents the timing requirement after the RC line delay. This window is available for devices to detect inactive -PREEMPT and exit Inactive State.
6. Because no maximum is specified, a channel attached Bus Master must degate bus drivers at the End of Transfer condition. The End of Transfer condition must be held stable until arbitration begins.

Configuration Timing

This section provides the specification for critical timing parameters for the system configuration protocol.

Setup Cycle



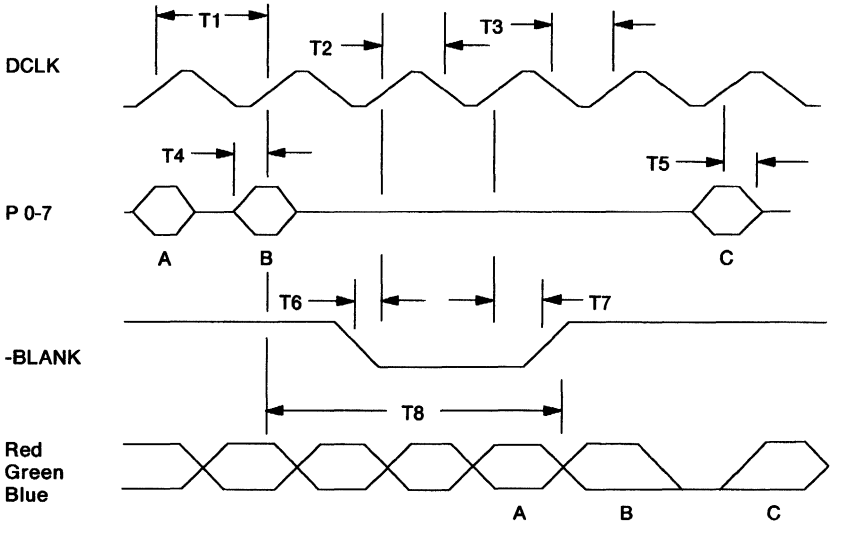
	Timing Parameter	Min/Max	Note
T60	CHRESET active (high) pulse width	100 / - ms	
T61	-CD SETUP (n) active (low) to -ADL active (low)	15 / - ns	
T62	-CD SETUP (n) hold from -ADL inactive (high)	25 / - ns	
T63	-CD SETUP (n) hold from -CMD active (low)	30 / - ns	
T65	CD CHRDY (n) inactive (low) from -CD SETUP (n) active	- / 100 ns	3

Figure 2-65. Setup Cycle

Notes:

1. Only those timing parameters that are different or additional to those specified for the Basic Transfer cycle are included here.
2. The Setup cycle is 300 nanoseconds minimum (default). A valid non-adapter selecting address must be present on the bus during system configuration.
3. A slave is allowed to extend the Setup cycle beyond 300 nanoseconds using CD CHRDY. The slave qualifies the leading edge of CD CHRDY with active Status.
4. Setup cycles are restricted to 8-bit transfers.

Auxiliary Video Connector Timing



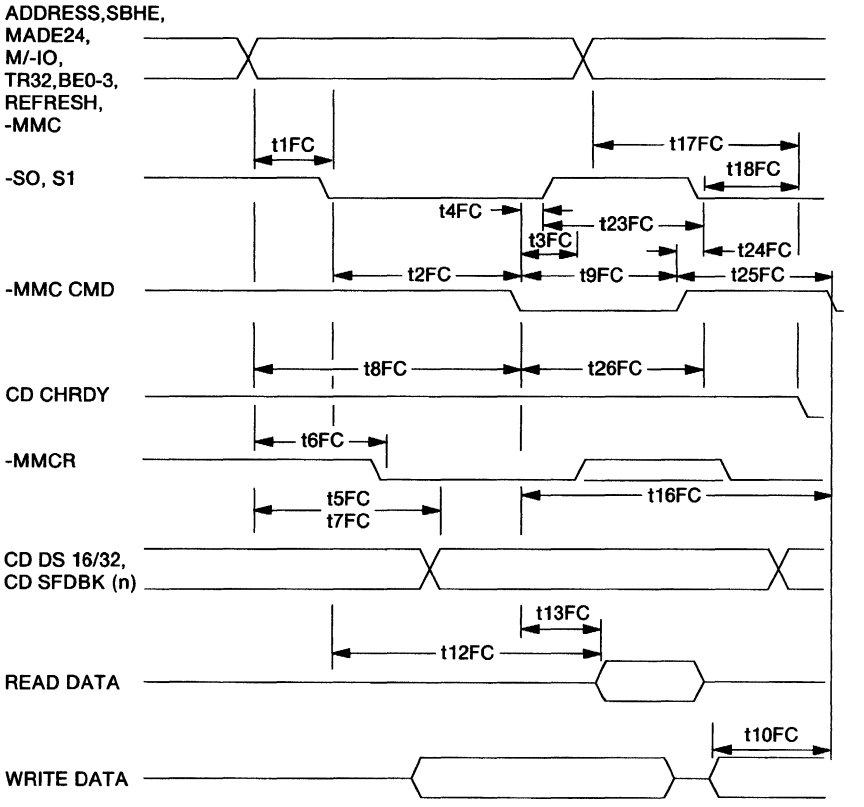
Symbol	Description	Min.(ns)	Max.(ns)
T1	PEL Clock Period (tclk)	28	10,000
T2	Clock Pulse Width High (tch)	7	10,000
T3	Clock Pulse Width Low (tcl)	9	10,000
T4	PEL Set-up Time (tps)	4	-
T5	PEL Hold Time (tph)	4	-
T6	Blank Set-up Time (tbs)	4	-
T7	Blank Hold Time (tbh)	4	-
T8	Analog Output Delay (taod)	3(T1) + 5	3(T1) + 30

Figure 2-66. Auxiliary Video Connector Timing (DAC Signals)

Note: See "15-Pin Display Connector Timing (Sync Signals)" on page 4-121 for additional video timing information.

Matched Memory Bus Timings

Matched Memory Cycle Timing (No Waits)



Symbol	Description	Min/Max
t1 FC	ADDR valid to Status active	10 / - ns
t2 FC	Status valid to -MMC CMD active	82 / - ns
t3 FC	ADDR hold from -MMC CMD active	20 / - ns
t4 FC	Status hold from -MMC CMD active	25 / - ns
t5 FC	CD DS 16/32 active from ADDR valid	- / 55 ns
t6 FC	-MMCR active from ADDR valid	- / 55 ns
t7 FC	-CD SFDBK active from ADDR valid	- / 60 ns
t8 FC	ADDR valid to -MMC CMD active	100 / - ns
t9 FC	-MMC CMD pulse width	85 / - ns
t10 FC	Write data valid to -MMC CMD active	0 / - ns
t11 FC	Write data hold from -MMC CMD inactive	30 / - ns
t12 FC	Read data valid from Status active	- /145 ns
t13 FC	Read data valid from -MMC CMD active	- / 60 ns
t14 FC	Read data hold from -MMC CMD inactive	0 / - ns
t15 FC	Read data off delay from -MMC CMD inactive	- / 40 ns
t16 FC	-MMC CMD active to next -MMC CMD active	180 / - ns
t17 FC	CD CHRDY valid from ADDR valid	- / 70 ns
t18 FC	CD CHRDY valid from Status active	- / 30 ns
t23 FC	Status inactive pulse width	30 / - ns
t24 FC	-MMC CMD inactive to Status active	- / 5 ns
t25 FC	-MMC CMD inactive pulse width	85 / - ns
t26 FC	-MMC CMD active to Status active	90 / - ns

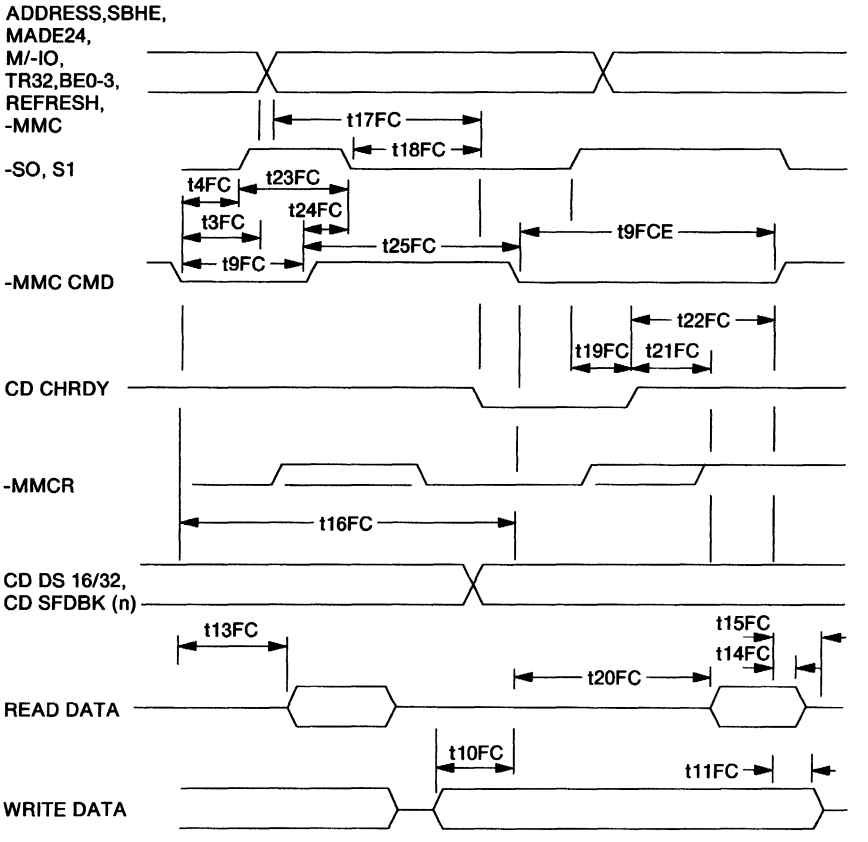
Figure 2-67. Matched Memory Cycle (No Waits)

Matched Memory cycles support 32-bit and 16-bit channel devices only. Eight-bit channel devices cannot run Matched Memory cycles.

Note: Adapters using this protocol can only be installed in the three 32-bit channel connectors and therefore sacrifice some *portability*. These adapters may not fit or operate in other systems.

When MMC is active, Matched Memory cycle 32-bit and 16-bit devices should not use -MADE 24, A0, A1, or -SBHE in logic that generates -MMCR, -CD DS 16/32, -SEL FBK, and -CD CHRDY.

Matched Memory Cycle Timing (One Wait)



Symbol	Description	Min/Max
t3 FC	ADDR hold from -MMC CMD active	20 / - ns
t4 FC	Status hold from -MMC CMD active	25 / - ns
t9 FC	-MMC CMD pulse width	85 / - ns
t9 FCE	-MMC CMD pulse width (Extended Cycle)	145 / - ns
t10 FC	Write data valid to -MMC CMD active	0 / - ns
t11 FC	Write data hold from -MMC CMD inactive	30 / - ns
t13 FC	Read data valid from -MMC CMD active	- / 60 ns
t14 FC	Read data hold from -MMC CMD inactive	0 / - ns
t15 FC	Read data off delay from -MMC CMD inactive	- / 40 ns
t16 FC	-MMC CMD active to next -MMC CMD active	180 / - ns
t17 FC	CD CHRDY valid from ADDR valid	- / 70 ns
t18 FC	CD CHRDY valid from Status active	- / 30 ns
t19 FC	CD CHRDY active from Status inactive	- / 9 ns
t20 FC	Read data valid from -MMC CMD active	- /125 ns
t21 FC	Read data valid from CD CHRDY active	- / 45 ns
t22 FC	-MMC CMD inactive from CD CHRDY active	65 / - ns
t23 FC	Status inactive pulse width	30 / -
t24 FC	-MMC CMD inactive to Status active	- / 5 ns
t25 FC	-MMC CMD inactive pulse width	85 / - ns

Figure 2-68. Matched Memory Cycle with One Wait

Matched Memory cycles support 32-bit and 16-bit channel devices only. Eight-bit channel devices cannot run Matched Memory cycles.

Note: Adapters using this protocol can only be installed in the three 32-bit channel connectors and therefore sacrifice some *portability*. These adapters may not fit or operate in other systems.

When MMC is active, Matched Memory cycle 32-bit and 16-bit devices should not use -MADE 24, A0, A1, or -SBHE in logic that generates -MMCR, -CD DS 16/32, -SEL FBK, and -CD CHRDY.

Adapter Design

This section provides some basic guidelines to design adapters for the Micro Channel Architecture 16- and 32-bit products. Topics include physical specifications, power requirements and limitations, and configuration program support.

The system board provides three types of connectors in the channel:

- 8- or 16-bit connector
- 8- or 16-bit connector with video extension
- 32-bit connector with matched memory extension (Model 80 only).

Plated connector contacts are not required for signals not used by an adapter. Three channel positions (1, 2, and 4) are provided on the Model 80 system board for 32-bit devices. Channel position 6 has the video extension connector. The video extension signal connector is used for display-only applications. See "Channel Definition" on page 2-6 for more information on the channel connectors and signals.

Physical Dimensions

The following figures show the dimensions of each type of adapter and the associated mounting hardware. The tolerances shown include all individual process tolerances and are noncumulative. The maximum height for components mounted on the adapter is 15 millimeters (0.6 inch) on the component (A) side. The maximum height for pins and components on the B side of the adapter is 2 millimeters (0.078 inches). Adapters using CMOS technology should be made with all plated connector contacts the same length to reduce the exposure of incorrect bias to modules.

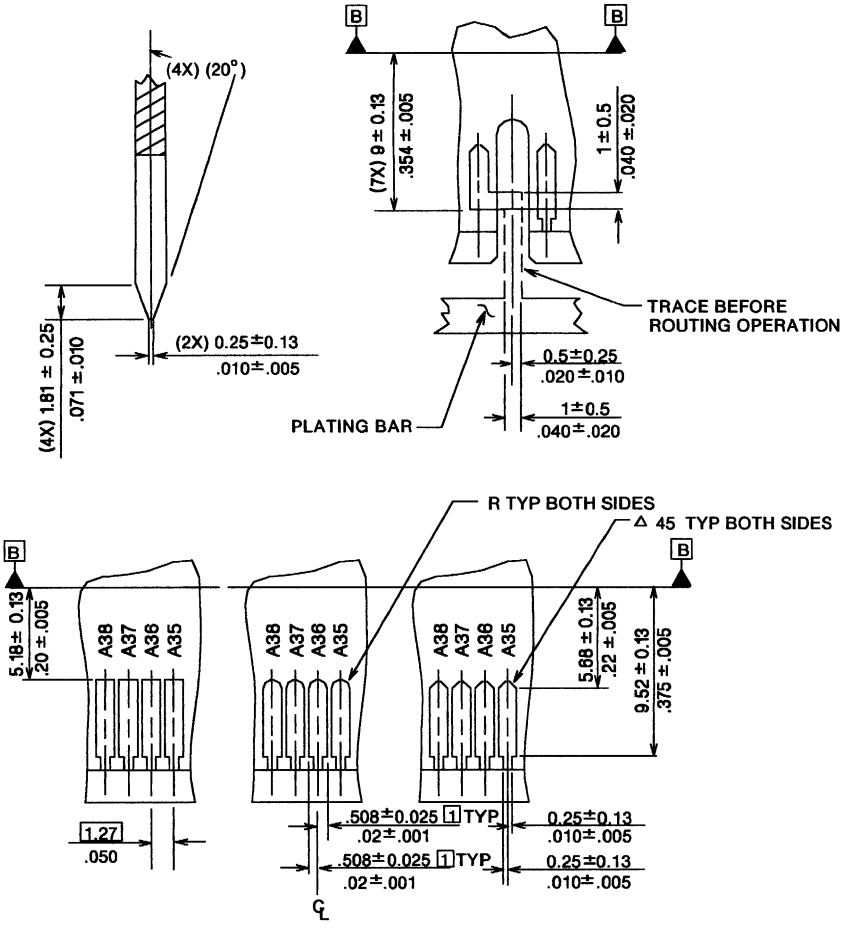


Figure 2-69. Connector (Common Detail)

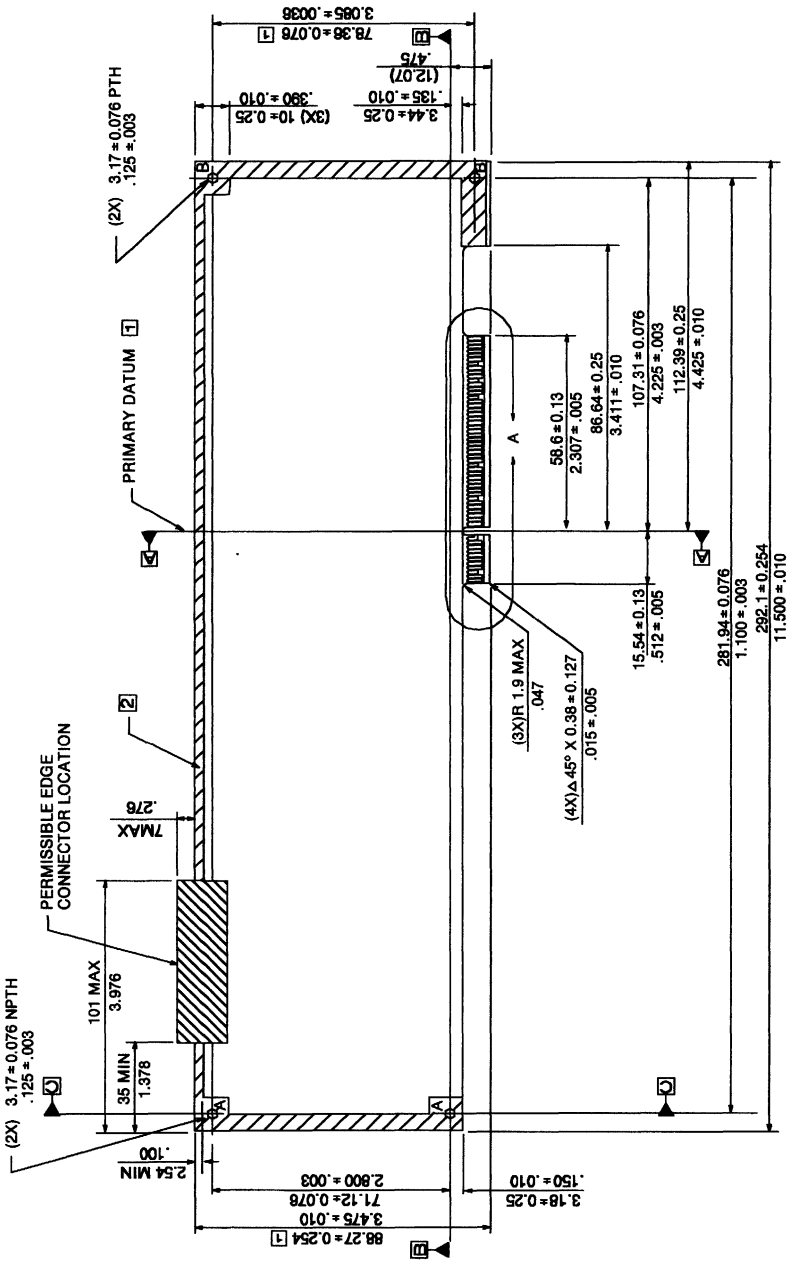


Figure 2-70. Adapter Dimensions (8- or 16-Bit)

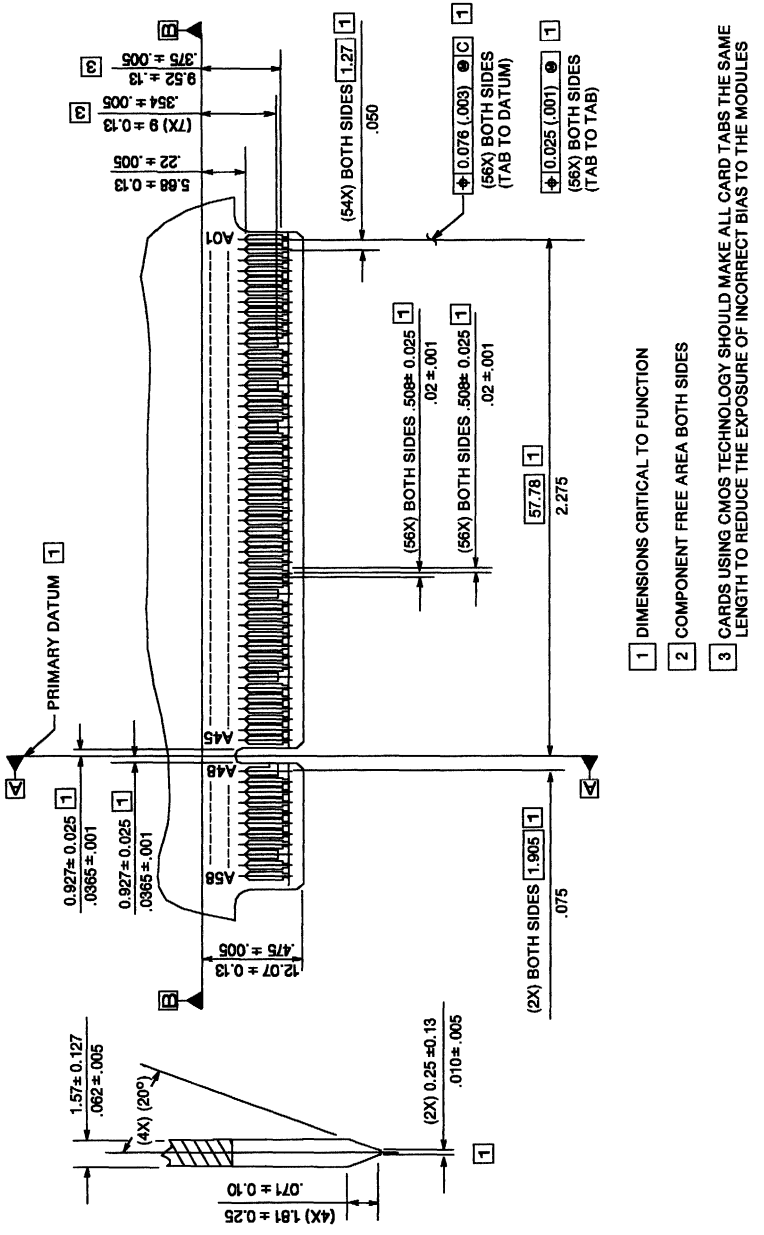


Figure 2-71. Connector Dimensions (8- or 16-Bit)

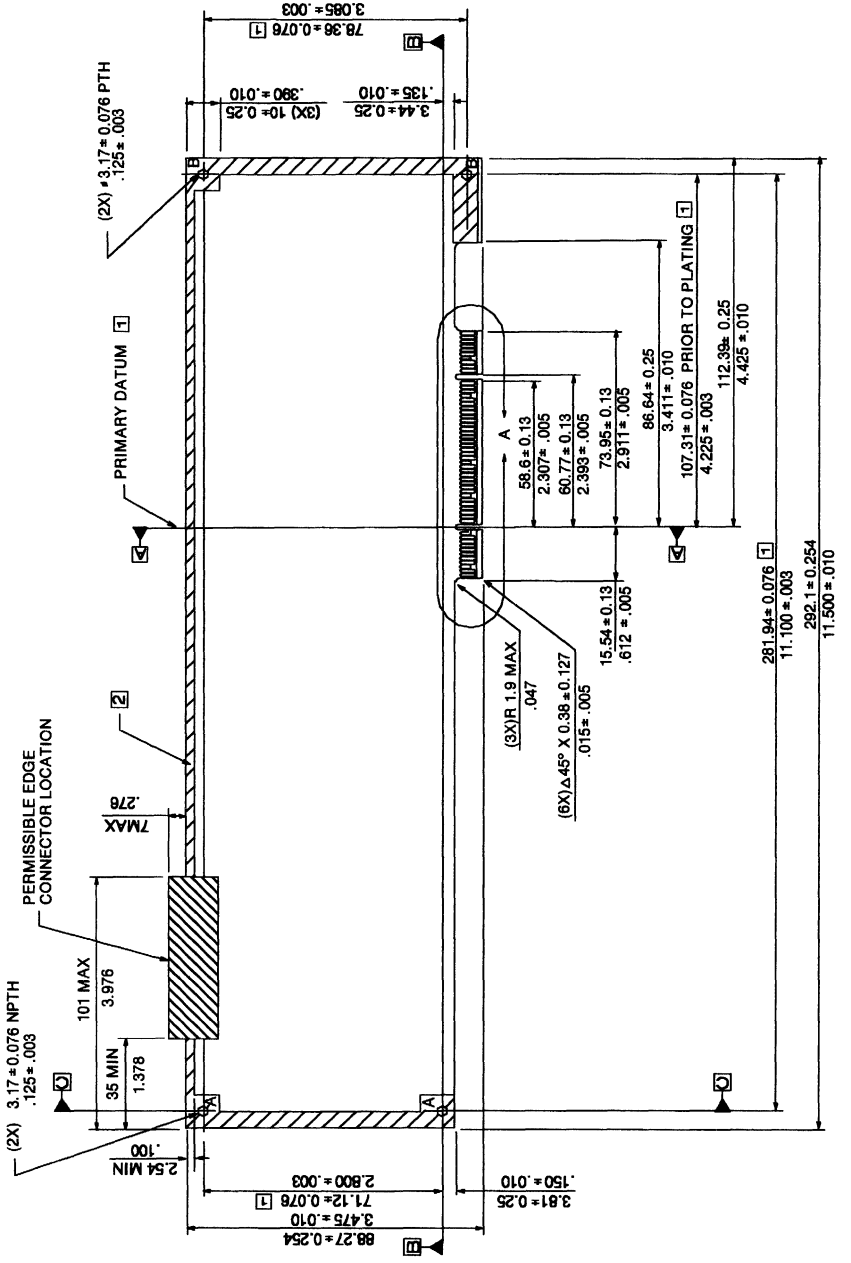


Figure 2-72. Adapter Dimensions (8- or 16-Bit with Video Extension)

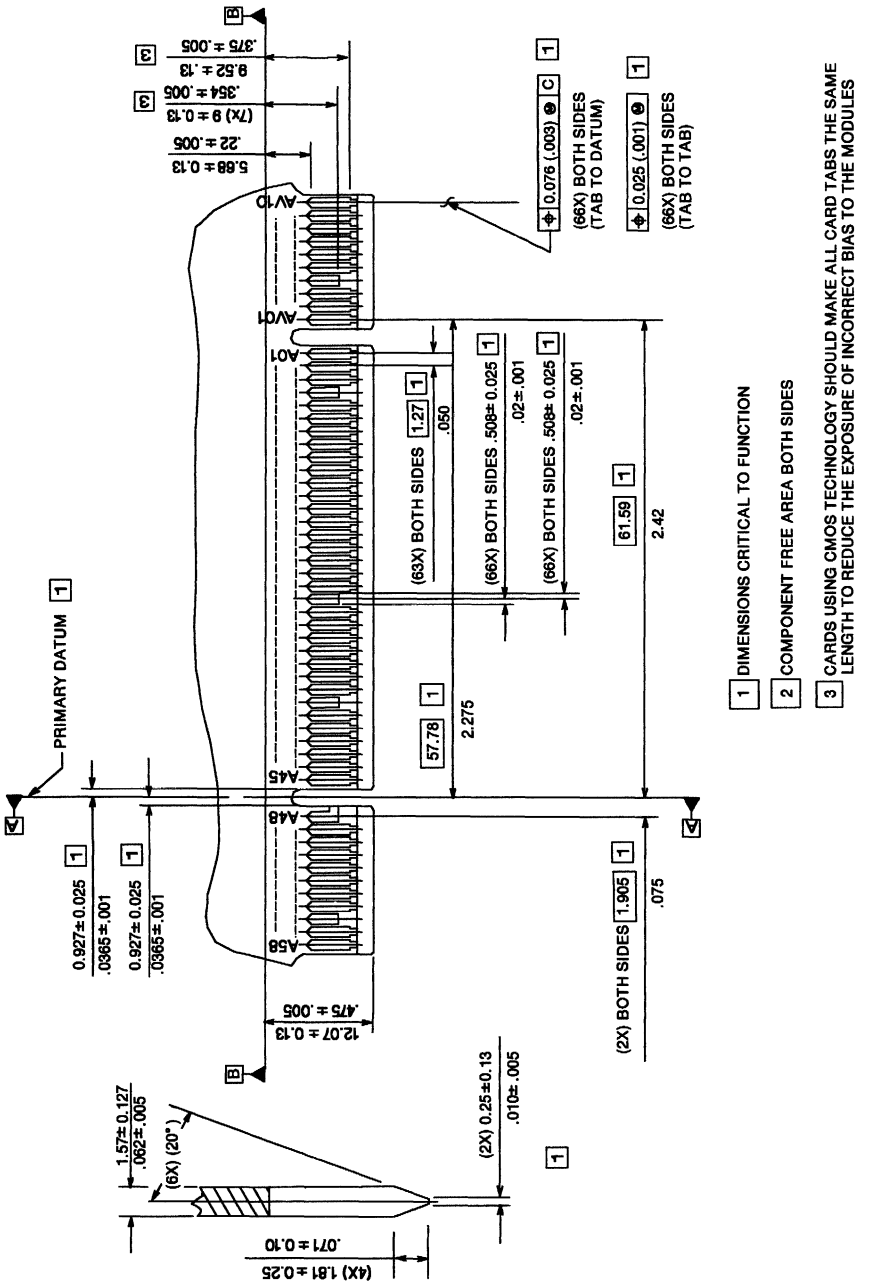


Figure 2-73. Connector Dimensions (8- or 16-Bit with Video Extension)

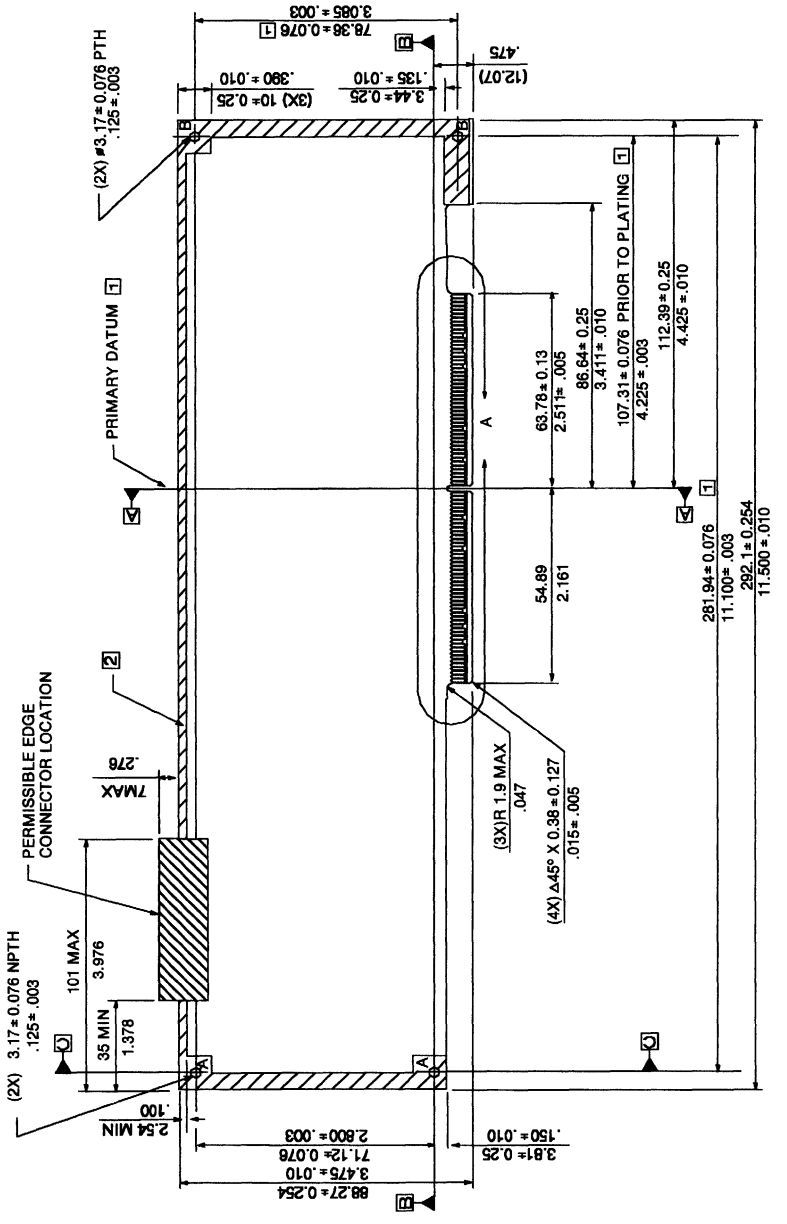


Figure 2-74. Adapter Dimensions (32-Bit with Matched Memory)

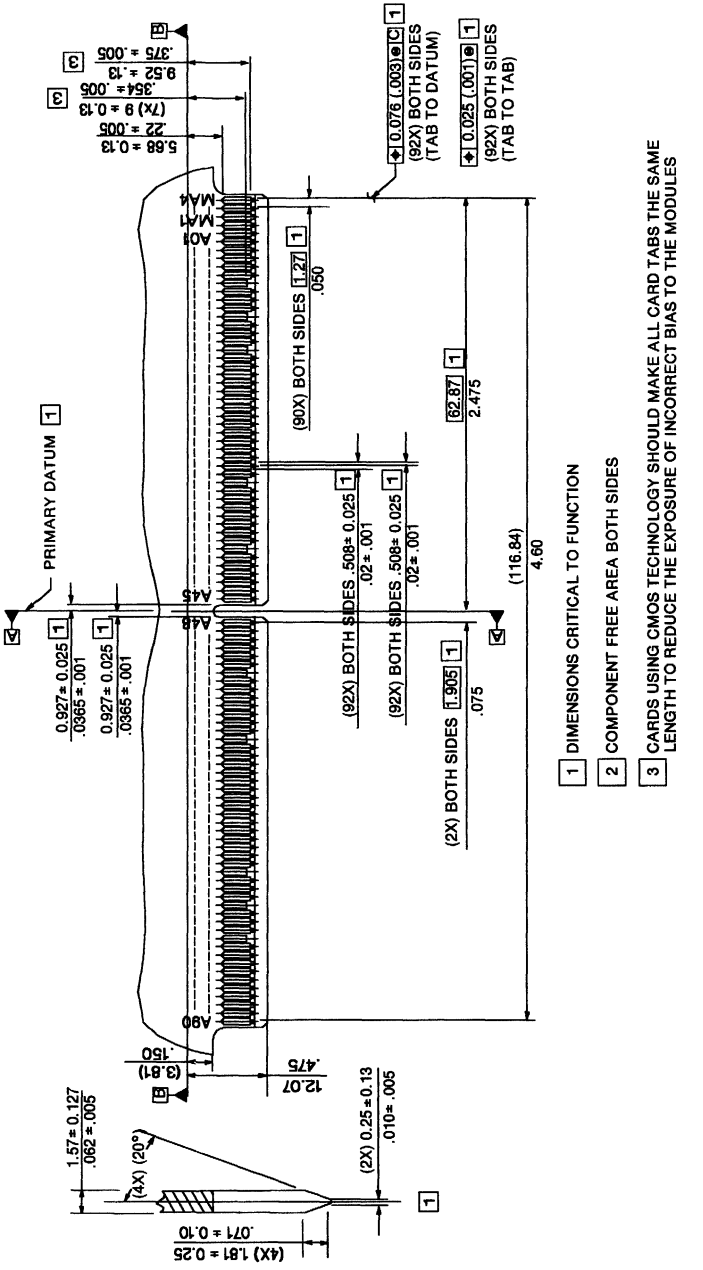
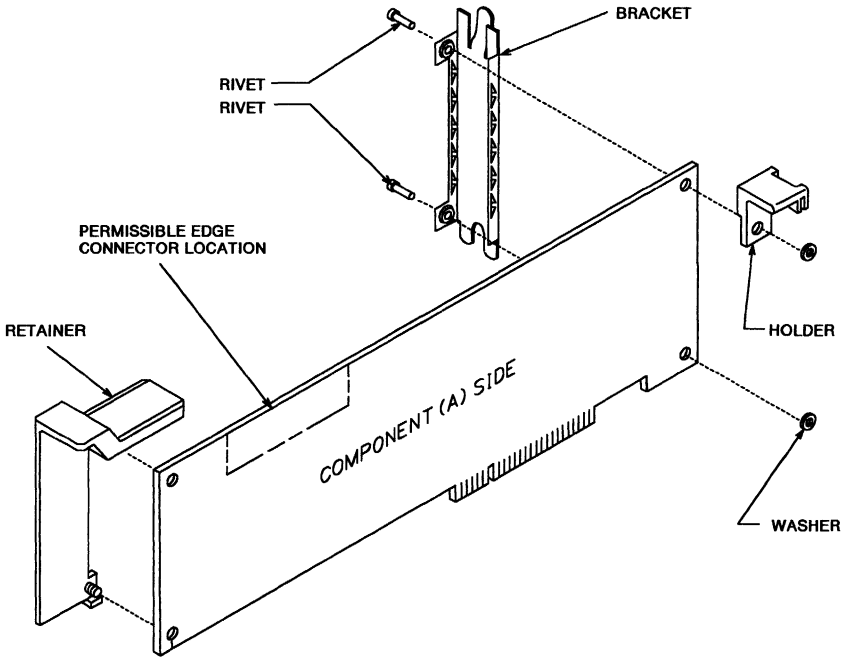


Figure 2-75. Connector Dimensions (32-Bit with Matched Memory)



MATERIALS:
HOLDER AND RETAINER - POLYCARBONATE UL 94 V-0
BRACKET - AISI TYPE 302 1/4 HARD STAINLESS STEEL

Figure 2-76. Typical Adapter Assembly

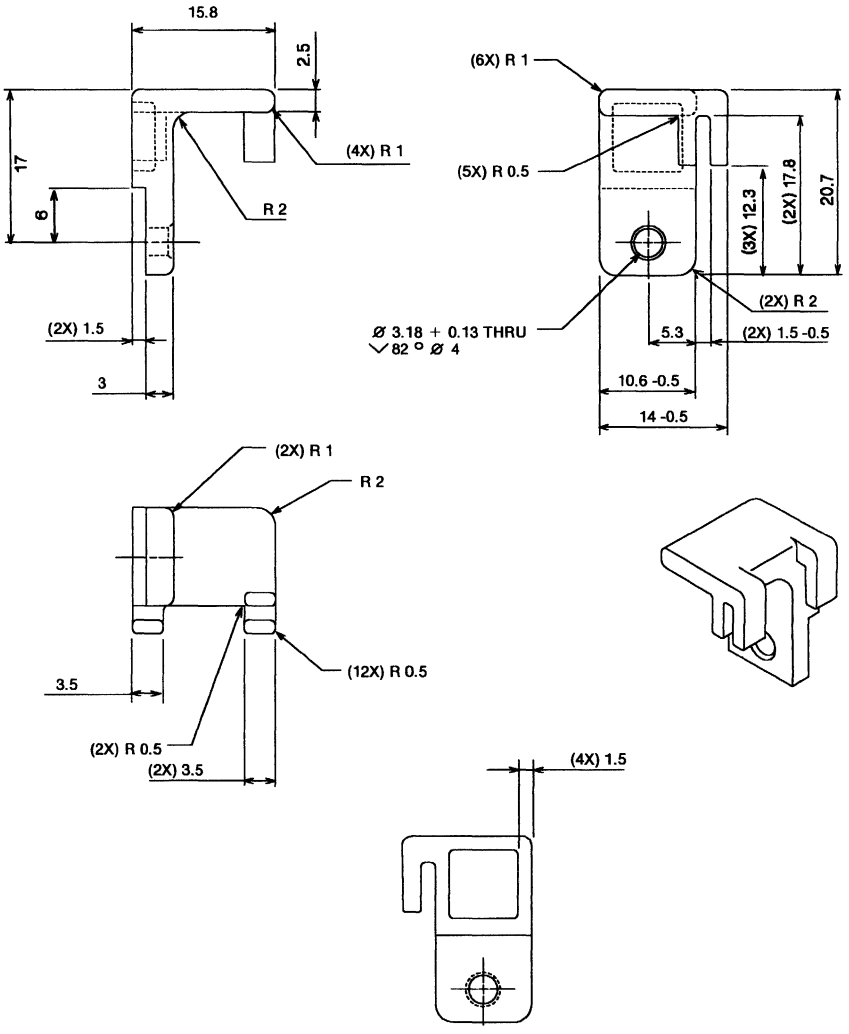


Figure 2-77. Adapter Holder

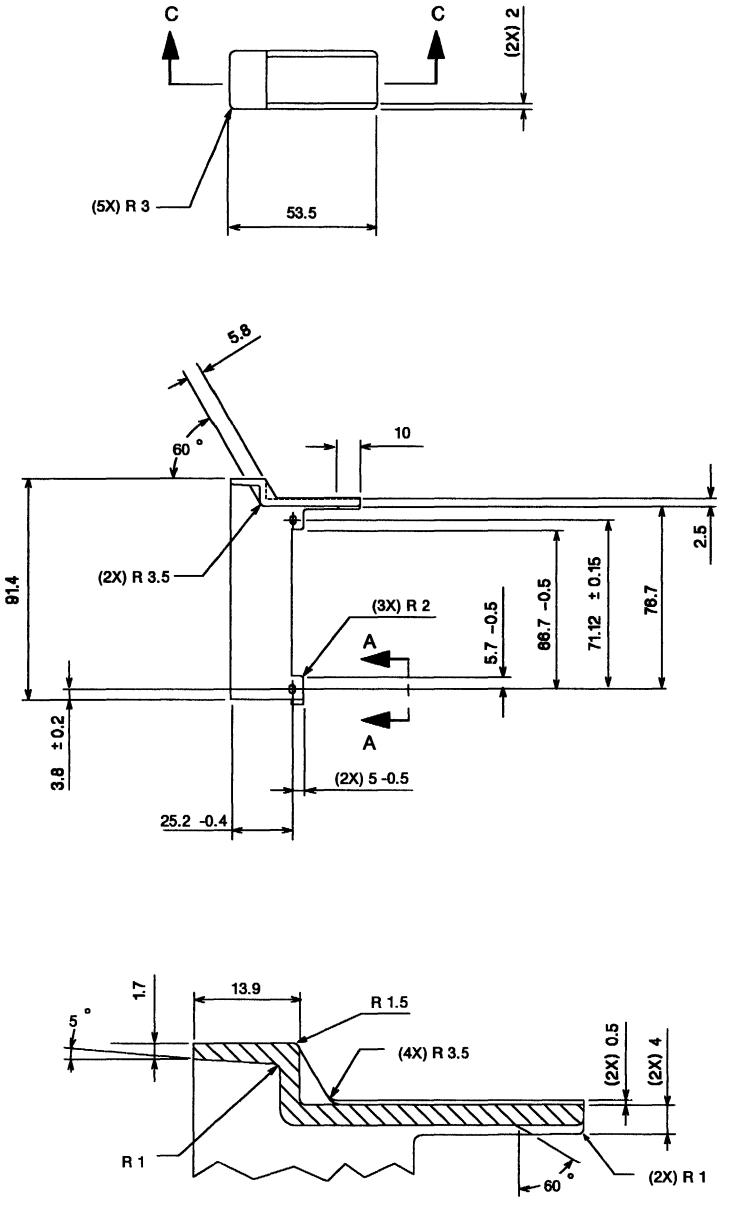
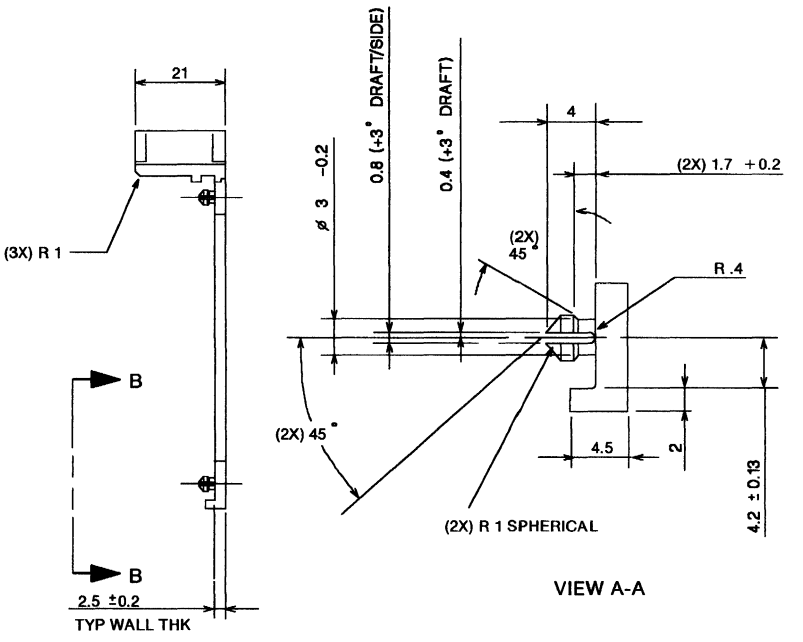


Figure 2-78. Adapter Retainer (Part 1 of 2)



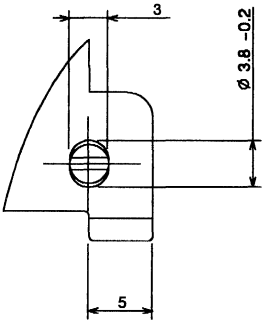
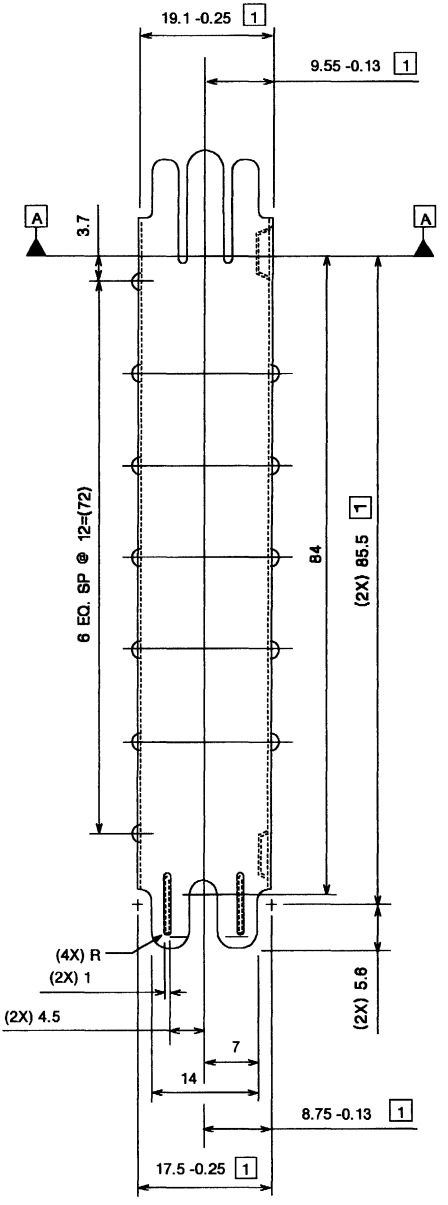


Figure 2-79. Adapter Retainer (Part 2 of 2)



1 DIMENSION IS CRITICAL TO FUNCTION.

Figure 2-80. Adapter Bracket (Part 1 of 4)

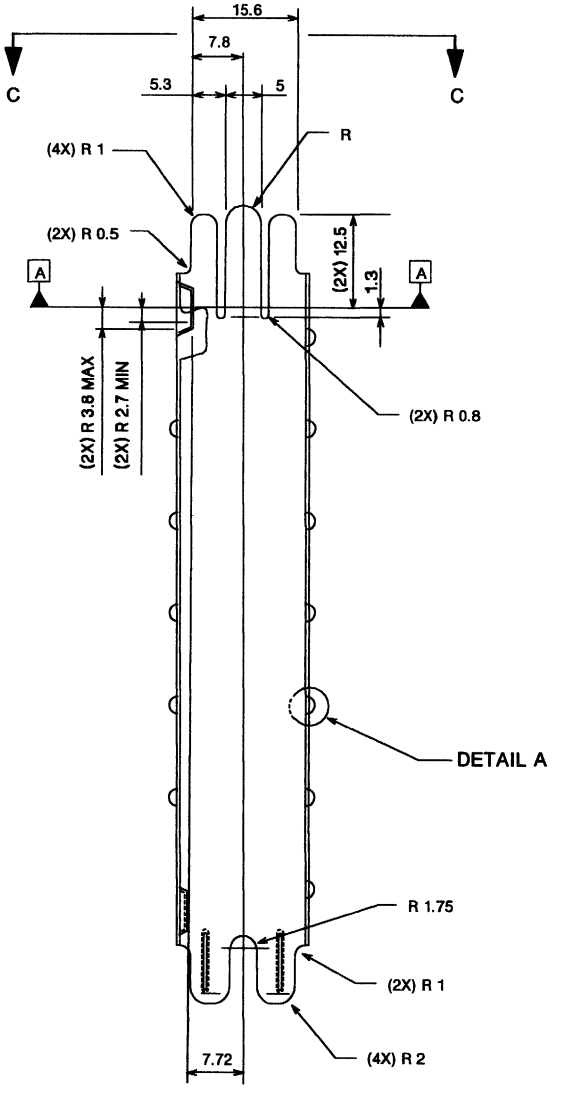


Figure 2-81. Adapter Bracket (Part 2 of 4)

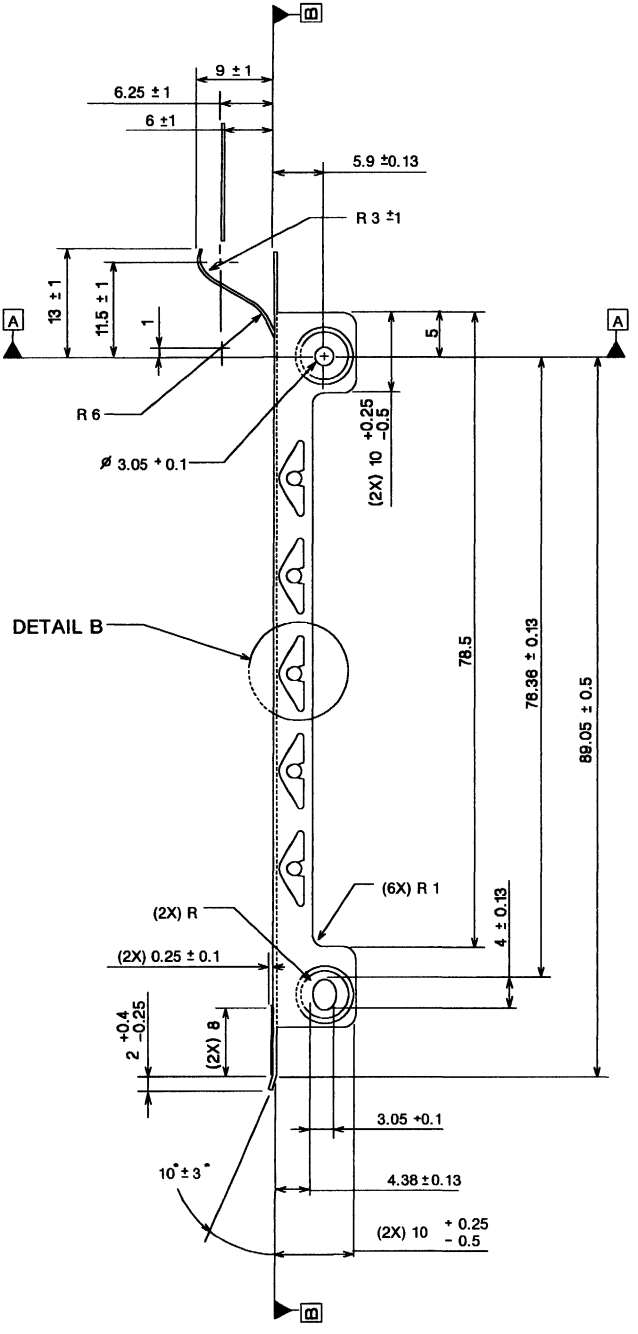


Figure 2-82. Adapter Bracket (Part 3 of 4)

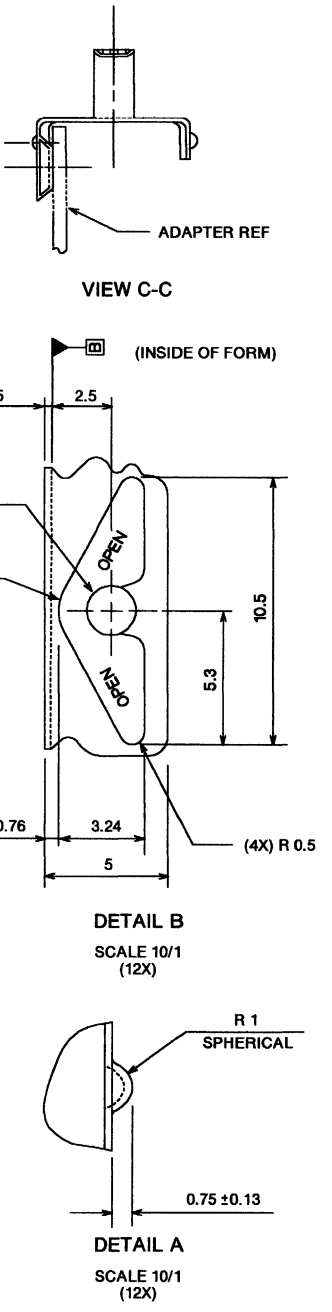


Figure 2-83. Adapter Bracket (Part 4 of 4)

Power

The allowable load current for each of the voltages present on each channel connector is as follows:

Supply Voltage	Maximum Current* Per Connector: 8/16-Bit Connector	8/16/32-Bit Connector
+5.0 Vdc	1.6 A	2.0 A
+12.0 Vdc	0.175 A	0.175 A
-12.0 Vdc	0.040 A	0.040 A

* This maximum current may not apply to other systems.

Figure 2-84. Channel Load Current

Voltage Regulation

The voltage regulation at the channel connector is shown in the following table:

Voltage	Pins	Tolerance
Ground	A3, B3, B5, B9, B13, B17, B21, B25, B29,	N/A
	B33, B37, B41, A43, B45, B50, B54, B58	N/A
	A61, B63, B67, B71, B75, B79, B83, B87,	N/A
	A89, BM4*, AM2*	
+5.0 Vdc	A7, A11, A15, A31, A39, A48, A56, A69,	+5% -4.5%
	A73, A85	
+12.0 Vdc	A19, A35, A52, A65, A77, A81	+5% -4.5%
-12.0 Vdc	A23, A27	+10% -9.5%
Ground (Auxiliary Video)	BV1, AV3, BV5, AV7, BV9	N/A

* These connectors are in the Matched Memory portion of the connector.

Figure 2-85. Channel Voltage Regulation

The tolerance includes all power distribution losses in both power and ground planes up to the pins of the channel connector. It does not include the drop due to the connector (30 milliohm maximum per contact), nor the drop due to distribution within the adapter.

General Design Considerations

Each designer is responsible for taking the necessary precautions to protect the safety of the end user, provide reliable operation of the device, and ensure the device does not interfere with the operation of the system or any other installed devices. This section is not intended to be an all-inclusive list of design considerations, but rather to point out some areas of design that may otherwise be overlooked.

Safety

Avoid exposed high voltage or current points, sharp edges, and exposed components that operate at high temperatures. Devices must not channel dc power outside of the system unit in any manner that violates Underwriter's Laboratory and Canadian Standards Association guidelines.

Note: Canadian Standards Association C22.2, paragraph 4.11.3, number 154 requires protection of conductors of external interconnecting cords and cables connected to secondary circuits.

IBM does not support installing or removing adapters or components when the system power is on.

Thermal

The system unit is cooled internally by low-volume forced air. Adapter designs must allow for adequate airspace between the adapters. Internal cables should be avoided as a mechanism for signal communication inside the system unit. These cables can interfere with the air flow. If internal cables are required, they must be positioned to minimize the impact on airflow. The maximum height for components mounted on the adapter should not exceed the dimensions specified under "Physical Dimensions" on page 2-114. The adapter design should avoid clustering of high temperature components. No component should exceed its maximum thermal rating.

Electro-Magnetic Compatibility

Adhere to the following guidelines to reduce electro-magnetic compatibility (EMC) problems.

- The adapter end brackets make a continuous 360° connection to the outside "skin" of the system unit cabinet. A similar 360° connection to the inside skin should also be provided. The adapter bracket must not be used as a dc voltage return path, a logic-ground connection, or an audio ground connection.
- The adapter end bracket at the rear of the adapter is isolated from dc ground on the adapter. The bracket must be grounded through a screw connection to the system unit and designed as shown in Figure 2-80 on page 2-126.
- All connector ground pins must be connected to the interplane ground at the channel connector. Likewise, the +5 Vdc power must be immediately connected to the +5 Vdc power plane.
- All adapters must provide nonsegmented internal power and ground planes.
- Each surface-mount technology (SMT) module position should provide a decoupling capacitor pad with minimal connection inductance. Pin-in-hole (PIH) modules should be decoupled if they drive or contain edge-triggered logic. Capacitors can range between 0.01 and 0.10 μF and should be low inductance ceramic or layered design.
- Internal cables should be avoided as a mechanism for signal communication inside the system unit. The channel should not be extended outside of the system unit, except by an adapter.
- Clocks should be properly imbedded and terminated. When clocks, strobes, and handshakes are generated or received, care should be taken to control the rise-and-fall times to minimize radiation.
- External cables should connect through 360° shielded D-shell or equivalent connectors. Avoid the use of "pigtail" shield connections. Shield terminations should be connected to the external shield of the cable connector. Do not bring the shield through the connector and connect it to either logic ground or the inside skin of the cabinet.

- High-current power within the system unit should provide adjacent or sandwiched return paths to allow the maximum cancelation of radiated magnetic fields by the mutual coupling between the supply and return lines.

Diagnostics

All writable registers typically are readable at the same address. External interfaces typically include 100% diagnostic wrap capability by electronic switching or an external wrap tool.

Design Guidelines

Adapters designed to be used in the Micro Channel architecture must comply with the following design guidelines:

- Each I/O adapter design must decode all 16 bits of the I/O address.
- Each memory adapter design must decode all 24 bits of the memory address and MADE 24.
- Each 32-bit memory adapter must decode all 32 bits of the memory address if MADE 24 is inactive.
- Each adapter design must replace the function of switches and jumpers with registers and incorporate POS logic.
- Each adapter must output a *card ID* to the data bus when interrogated.

  To minimize the number of drivers required, only the logical 0 bits in the card ID need to be driven. This provides 39,202 combinations with 8 drivers or less.

  The following figure shows the recommended ID values for vendors.

ID	Definition
0000	Device Not Ready
0001 to 0FFF	Bus Master
5000 to 5FFF	Direct Memory Access Devices
6000 to 6FFF	Direct Program Control (Including Memory-Mapped I/O)
7000 to 7FFF	Storage*
8000 to 8FFF	Video
FFFF	Device Not Attached

* Multiple-function adapters containing storage typically respond as storage.

Figure 2-86. ID Assignments

- Each enabled adapter must return a 'card selected feedback' signal (CD SFDBK) to the system microprocessor when an access is made to the address space of the adapter, or when the adapter is selected by arbitration level. CD SFDBK must not be generated when the '-card setup' (-CD SETUP) line is active.
- Each adapter design must be capable of de-gating all outputs to the system board (including CD SFDBK, -CD DS 16/32, interrupts, and so on) if the least-significant bit of the option select word at address hex 0102 is set to 0.
- Each adapter design must implement an open collector driver (or tri-state driver gated negative active) to drive the interrupt request line. The design must also implement a status register (readable at an I/O address bit position) that remains active when the interrupt is set and stays active until reset by the service routine. The adapter must hold the level-sensitive interrupt active until it is reset as a direct result of servicing the interrupt. The service routine must not reset the interrupt controller until after the interrupt bus signal has been reset by the adapter.
- Following a reset, each adapter must set bit 0 (Card Enable) of its POS register 2 to 0.
- If applicable, the adapter can have the ability to reside at an alternate address (corresponding to one selected by switches on a Personal Computer-type adapter).
- All adapters must provide system configuration (.adf) files on a 3.5-inch diskette.

Section 3. System Board

Description

This section describes the various system board components and how they interact. Hardware descriptions and programming information is provided to acquaint hardware designers and programmers with the basic operation of the system.

System Microprocessor

The 80386 microprocessor sub-system has the following:

- 16-MHz clock operation
- 32-bit address
- 32-bit data interface
- Extensive instruction set, including string I/O
- Hardware fixed-point multiply and divide
- Three operational modes
  - Real Address Mode
  - Protected Virtual Address Mode
  - Virtual 8086 Mode
- 4G (G = 1,073,741,824 or 2^{30}) of physical address space
- 8 General Purpose 32-bit registers
- 64T (T = 1,099,511,627,776 or 2^{40}) of total virtual address space per task.

Real Address Mode

In the Real Address Mode, the system microprocessor address space is a contiguous array of up to 1M. The system microprocessor addresses memory by generating 20-bit physical addresses.

The segment portion of the pointer is interpreted as the upper 16 bits of a 20-bit segment address. The lower 4 bits of the 20-bit segment address are always zero. Therefore, segment addresses begin on multiples of 16 bytes.

All segments in the Real Address Mode are 64K in size and can be read, written, or executed. An exception or interrupt can occur if data operands or instructions attempt to wrap around the end of a segment

(for example, a word with its low-order byte at offset hex FFFF and its high-order byte at hex 0000). If, in the Real Address Mode, the information contained in the segment does not use the full 64K, the unused end of the segment can be overlaid by another segment to reduce physical memory requirements.

Protected Virtual Address Mode

The Protected Virtual Address Mode (Protected Mode) offers extended physical and virtual memory address space, memory protection mechanisms, and new operations to support operating systems and virtual memory.

The Protected Mode provides up to 64T of virtual address space per task mapped into a 4G physical address space.

From a programmer's point of view, the main difference between the Real Mode and Protected Mode is the increased address space and the method of calculating the base address. The Protected Mode uses 32- or 48-bit pointers, consisting of 16-bit selector and 16- or 32-bit offset components. The selector specifies an index into one of two memory-resident tables, the global descriptor table (GDT) or the local descriptor table (LDT). These tables contain the 32-bit base address of a given segment. The 32-bit effective offset is added to the segment base address to form the physical address. The tables are automatically referenced by the microprocessor whenever a segment register is loaded with a selector. All instructions that load a segment register refer to the memory-resident tables without additional program support. The memory-resident tables contain 8-byte values called descriptors. For additional information about descriptors, see Section 7, "Instruction Sets."

The paging option provides an additional memory management mechanism that operates in the Protected Mode only. 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. When paging is not enabled, the physical address is the same as the linear address. The following figure shows the 80386 addressing mechanism.

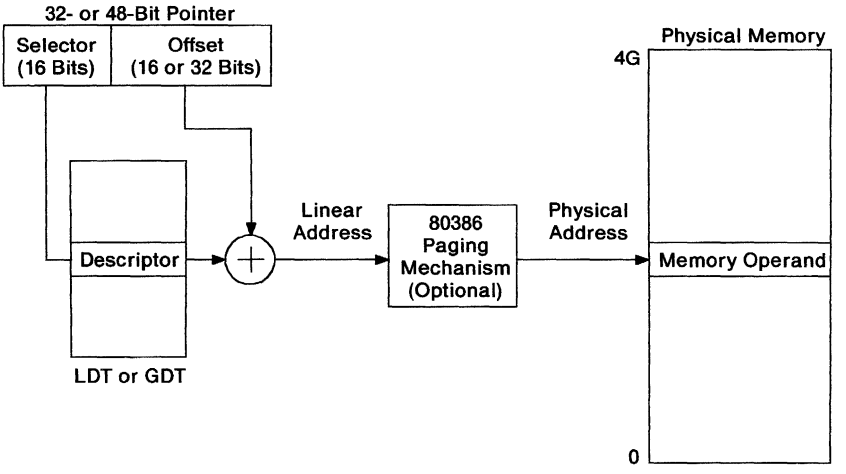


Figure 3-1. 80386 Addressing

Virtual 8086 Mode

The Virtual 8086 Mode ensures compatibility with programs written for 8086- and 8088-based systems by establishing a protected 8086 environment within the 80386 multitasking framework.

Since the address space of an 8086 is limited to 1M, the logical addresses generated by a Virtual 8086 Mode lie within the first M byte of the 80386 linear address space. In order to support multiple Virtual 8086 tasks, paging can be used to give each Virtual 8086 task a 1M address space anywhere in the 80386 physical address space.

On a task-by-task basis, the value of the Virtual 8086 flag (VM86 flag in the Flags register) determines whether the 80386 behaves as an 80386 or emulates an 8086. Some instructions such as the Clear Interrupt Flag instruction can disrupt all operations in a multitasking environment. The 80386 raises an exception when a Virtual 8086 Mode task attempts to execute an I/O instruction, interrupt-related instruction, or other sensitive instruction. Anytime an exception or interrupt occurs, the 80386 leaves the Virtual 8086 Mode making the full resources of the 80386 available to an interrupt handler or exception handler. These handlers can determine if the source of the exception was a Virtual 8086 Mode task by inspecting the VM86 flag in the Flags image on the stack. If the source is a Virtual 8086 Mode

task, the handler calls on a routine[1] in the operating system to simulate an 8086 instruction and return to the Virtual 8086 Mode.

80386 Paging Mechanism

The 80386 uses two levels of tables to transfer the linear address from the segmentation unit into a physical address. There are three components to the paging mechanism:

- Page directory
- Page tables
- Page frame (the page itself).

The following figure shows how the two-level paging mechanism works.

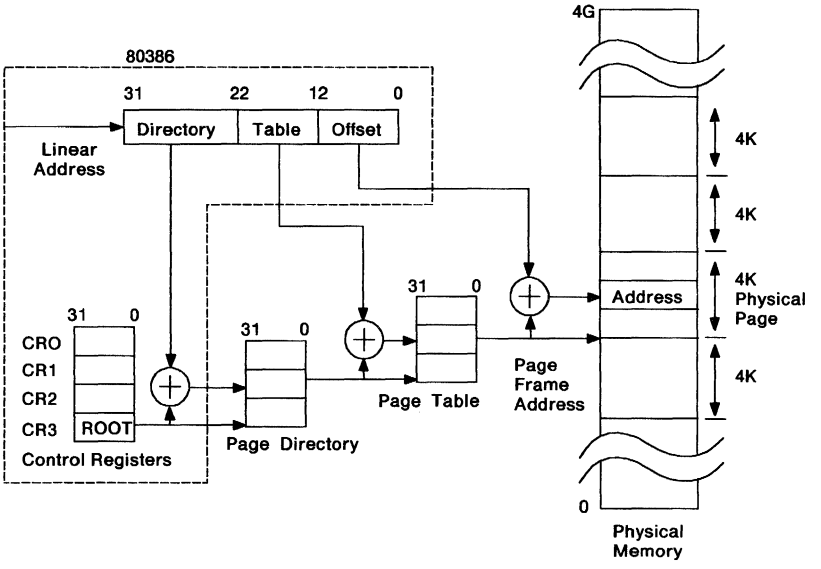


Figure 3-2. Paging Mechanism

CR2 is the Page Fault Linear Address register. It holds the 32-bit linear address that caused the last page fault detected.

---

[1] The routine in the operating system, called a *virtual machine monitor*, simulates a limited number of 8086 instructions.

CR3 is the Page Directory Physical Base Address register. It contains the physical starting address of the Page Directory.

The Page Directory is 4K and allows up to 1024 page-directory entries. Each page-directory entry contains the address of the next level of tables, the Page Tables, and information about the Page Tables. The upper 10 bits of the linear address (A22 through A31) are used as an index to select the correct page-directory entry.

Each Page Table is 4K 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. Address bits A12 through A21 are used as an index to select one of the 1024 page-table entries. The upper 20 bits of the page-frame address (from the page table entry) are linked with the lower 12 bits of the linear address to form the physical address. The page-frame address bits become the most-significant bits and the linear-address bits become the least-significant bits.

Performance

The 80386 microprocessor operates at 16 MHz for a clock cycle time of 62.5 nanoseconds.

The system inserts one wait state whenever it accesses system board RAM or ROM, which results in a 187.5 nanosecond memory cycle time. The system inserts six wait states whenever it does a system board I/O operation, which results in an I/O cycle time of 500 nanoseconds.

The refresh controller operates at 8 MHz, which results in a clock cycle time of 125 nanoseconds. Each refresh cycle takes 625 nanoseconds. Refresh overhead is less than 5%.

80387 Math Coprocessor

The optional 80387 Math Coprocessor provides arithmetic instructions for a variety of numeric data types. It also executes numerous built-in transcendental functions such as tangent, sine, cosine, and log functions. The 80387 effectively extends the 80386 register and instruction set for existing data types and adds several new data

types as well. The following figure shows the four data type classifications and the instructions associated with each.

Classification	Size	Instructions
Integer	16, 32, 64 bits	Load, Store, Compare, Add, Subtract, Multiply, Divide
Packed BCD*	80 Bits	Load, Store
Real	32, 64 bits	Load, Store, Compare, Add, Subtract, Multiply, Divide
Temporary Real	80 bits	Add, Subtract, Multiply, Divide, Square Root, Scale, Remainder, Integer Part, Change, Sign, Absolute Value, Extract Exponent and Significand, Compare, Examine, Test, Exchange Tangent, Arctangent, 2^X-1, $Y*Log_2 (X+1)$, $Y*Log_2 (X)$, Load Constant (0.0, π, etc.), Sine, Cosine, Unordered Compare

* BCD = Binary-coded decimal notation

Figure 3-3. Data Type Classifications and Instructions

The 80386/80387 fully conforms to the ANSI/IEEE[2] floating-point standard and is upward object-code compatible from the 80286/80287- and 8086/8087-based systems.

Programming Interface

The 80387 is not sensitive to the processing mode of the 80386. The 80387 functions 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.

All communication between the 80386 and 80387 is transparent to application software. The 80386 automatically controls the 80387 whenever a numeric instruction is executed. All physical and virtual memory is available for storage of instructions and operands of programs that use the 80387. All memory address modes, including use of displacement, base register, index register, and scaling are

---

[2] American National Standards Institute/Institute of Electrical and Electronics Engineers

available for addressing numeric operands. See Section 7, "Instruction Sets," for additional address mode information.

The coprocessor has eight 80-bit registers. The total capacity of these eight registers is equivalent to twenty 32-bit registers. This register space allows constants and temporary results to be held in registers during calculations, thus reducing memory access and improving speed as well as bus availability. The register set can be used as a stack or as a fixed register set. When used as a stack, only the top two stack elements are operated on.

The following figure shows the seven data types supported by the 80387 Math Coprocessor.

Data Type	Range	Precision
Word Integer	10^4	16 Bits
Short Integer	10^9	32 Bits
Long Integer	10^{19}	64 Bits
Packed BCD	10^{18}	18 Digits (2 Digits per byte)
Single Precision (Short Real)	$10^{\pm 38}$	24 Bits
Double Precision (Long Real)	$10^{\pm 308}$	53 Bits
Extended Precision (Temporary Real)	$10^{\pm 4932}$	64 Bits

Figure 3-4. Data Types

Hardware Interface

The 80387 Math Coprocessor uses the same clock generator as the 80386 system microprocessor and operates at 16 MHz in the synchronous mode. The coprocessor is wired so that it functions as an I/O device through I/O port addresses hex 00F8, 00FA, and 00FC. The system microprocessor sends opcodes and operands through these I/O ports. The coprocessor 'busy' signal informs the system microprocessor that it is executing; the system microprocessor Wait instruction forces the system microprocessor to wait until the coprocessor is finished executing.

The coprocessor detects six different exception conditions that can occur during instruction execution:

- Invalid operation
- Denormal operand
- Zero-divide
- Overflow
- Underflow
- Precision.

If the appropriate exception mask-bit within the coprocessor is not set, the coprocessor activates the 'error' signal. The 'error' signal generates a hardware interrupt 13 (IRQ 13) causing the 'busy' signal to be held in the busy state. The 'busy' signal may be cleared by an 8-bit I/O Write command to address hex 00F0 with D7 through D0 equal to 0. This action also clears IRQ 13.

The power-on self-test code in the system ROM enables IRQ 13 and sets up its vector to point to a routine in ROM. The ROM routine clears the 'busy' signal latch and then transfers control to the address pointed to by the nonmaskable interrupt (NMI) vector. This maintains code compatibility across the IBM Personal Computer and Personal System/2 product lines. The NMI handler reads the status of the coprocessor to determine if the NMI was caused by the coprocessor. If the interrupt was not generated by the coprocessor, control is passed to the original NMI handler.

Detailed information about the internal functions of the Intel 80387 Math Coprocessor can be found in the publications listed in the Bibliography.

Math Coprocessor Compatibility

IBM systems use three math coprocessors: the 8087, 80287, and 80387. In the Real Address Mode and Virtual 8086 Mode, the 80386/80387 is upward object-code compatible with software for the 8086/8087 and 80286/80287 Real-Address Mode systems.

In the Protected Mode, the 80386/80387 is upward object-code compatible with software for the 80286/80287 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 the Virtual 8086 Mode), are in the format of operands for the administration instructions FLDENV, FSTEN, FRSTOR, and FSAVE. These instructions are normally used only by exception handlers and operating systems, not by application programs.

Operating Modes	Software Written for:		
	8087 Real	80287 Real	80287 Protected
8087 Real Mode	Yes	Yes*	No
80287 Real Mode	Yes*	Yes	No
80387 Real Mode	Yes***	Yes**	No
80387 8086 Virtual Mode	Yes***	Yes**	No
80287 Protected Mode	No	Yes**	Yes
80387 Protected Mode	No	No	Yes**

* See "8087 to 80287 Compatibility."
** See "80287 to 80387 Compatibility."
***See "8087 to 80287 Compatibility" and "80287 to 80387 Compatibility."

Figure 3-5. Math Coprocessor Software Compatibility

Many changes have been designed into the 80387 to directly support the IEEE standards in hardware. These changes result in increased performance by eliminating the need for software that supports the IEEE standard.

80287 to 80387 Compatibility

The following summarizes the differences between the 80387 and 80287 Math Coprocessors, and provides details showing how 80287 software can be ported to the 80387 Math Coprocessor.

Note: Any migration from 8087 directly to the 80387 must also take into account the differences between the 8087 and the 80287. This information is provided on page 3-13.

- The 80387 supports only affine closure for infinity arithmetic, not projective closure.
- Operands for FSCALE and FPATAN are no longer restricted in range (except $\pm\infty$); F2XM1 and FPTAN accept a wider range of operands.
- Rounding control is in effect for FLD *constant*.

- Software cannot change entries of the tag word to values (other than empty) that differ from actual register contents.
- In conformance with the IEEE standard, the 80387 does not support special data formats pseudozero, pseudo-NaN, pseudoinfinity, and unnormal.

Exceptions

When the overflow or underflow exception is masked, the only difference from the 80287 is in rounding when overflow or underflow occurs. The 80387 produces results that are consistent with the rounding mode.

For exceptions that are not masked, a number of differences exist due to the IEEE standard and to functional improvements to the architecture of the 80387:

- There are 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.
- The FSQRT, FBSTP, and FPREM instructions may cause underflow, because they support denormal operands.
- The denormal exception can occur during the transcendental instructions and the FXTRACT instruction.
- The denormal exception no longer takes precedence over all other exceptions.
- When the operand is zero, the FXTRACT instruction reports a zero-divide exception and leaves $-\infty$ in ST(1).
- The status word has a new bit (SF) that signals when invalid-operation exceptions are due to stack underflow or overflow.
- FLD *extended precision* no longer reports denormal exceptions, because the instruction is not numeric.
- 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.

- 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).
- Most 80387 numeric instructions are automatically synchronized by the 80386. No explicit Wait instructions are required for these instructions. To maintain compatibility with systems using the 8087, an explicit Wait is required before each numeric instruction.
- The FLDENV and FRSTOR instructions should be followed by an explicit Wait when used in the 80387 environment. An explicit Wait is not required after these instructions in the 80287 environment.
- The 80287 FSETPM (set Protected Mode) instruction performs no useful purpose in the 80387 environment; if encountered, it is ignored.
- The format of the FSAVE and FSTENV instructions is determined by the current mode of the 80386; the Real Address Mode format is used when the 80386 is in the Real Address Mode, and the Protected Mode format is used when the 80386 is in the Protected Mode.
- The following applies only to the B1 stepping level 80386: An interrupt 9 does not occur for an operand outside a segment size; an interrupt 13 occurs.

8087 to 80287 Compatibility

The 80287 operating in the Real Address Mode can execute 8087 software without major modifications. However, because of differences in the handling of numeric exceptions by the 80287 and the 8087, exception-handling routines *may* need to be changed.

The following summarizes the differences between the 80287 and 8087 Math Coprocessors, and provides details showing how 8087 software can be ported to the 80287 Math Coprocessor.

- The 8087 instructions FENI/FNENI and FDISI/FNDISI perform no useful function in the 80287 environment. If the 80287 encounters one of these opcodes in its instruction stream, the instruction is effectively ignored; none of the 80287 internal states are updated.

While 8086 code containing these instructions may be executed on an 80287, it is unlikely that the exception-handling routines containing these instructions will be completely portable to the 80287.

- 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.
- In the Protected Mode, the format of the 80287 saved instruction and address pointers is different than the format of the 8087. The instruction opcode is not saved in the Protected Mode; exception handlers have to retrieve the opcode from memory if needed.
- Interrupt 7 occurs 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 also causes interrupt 7. An exception handler should be included in 80286 code to handle these exceptions.
- Interrupt 9 occurs if the second or subsequent words of a floating-point operand fall outside a segment size. Interrupt 13 occurs if the starting address of a numeric operand falls outside a segment size. An exception handler should be included in the 80286 code to report these programming errors.
- Most 80287 numeric instructions are automatically synchronized by the 80286. The 80286 automatically tests the 'busy' signal from the 80287 to ensure that the 80287 has completed its previous instruction before executing the next ESC instruction. Explicit Wait instructions are not required to ensure this synchronization. An 8087 used with 8086 and 8088 system microprocessors requires explicit Waits before each numeric instruction to ensure synchronization. Although 8086 software having explicit Wait instructions executes perfectly on the 80286 without reassembly, these Wait instructions are unnecessary.

The processor control instructions for the 80287 may be coded using either a WAIT or No-WAIT form of the mnemonic. The WAIT forms of these instructions cause the assembler to precede the ESC instruction with a microprocessor Wait instruction.

DMA Controller

The Direct Memory Access (DMA) controller allows I/O devices to transfer data directly to and from memory. This allows a higher system microprocessor throughput by freeing the system microprocessor of I/O tasks.

The DMA controller is software programmable. The system microprocessor can address the DMA controller and read or modify the internal registers to define the various DMA modes, transfer addresses, transfer counts, channel masks, and page registers.

The functions of the DMA controller can be grouped into two categories: program condition and DMA transfer. During program condition, the DMA registers can be programmed or read.

Program condition commences when the system microprocessor refers to the DMA controller within a specific address range. These addresses are identified in Figure 3-7 on page 3-19. The DMA controller needs a minimum of 375 nanoseconds to complete any program command generated by the system microprocessor. While in the program control mode, the DMA registers can be programmed or read.

The second function of the controller is the actual DMA transfer. This period is initiated when a DMA slave has won the arbitration bus and the DMA controller has been programmed to service the winning request in process. DMA transfers can be in the form of a single transfer, multiple transfer (burst), or read verification. During read verification, one byte is read from memory and no transfer to I/O occurs. Burst transfers require a minimum of [375 + (500)N] nanoseconds where N is the number of byte or word transfers during the burst. Burst transfers to or from system board memory require a minimum of [375 + (625)N] nanoseconds. Single transfers require a minimum of 875 nanoseconds to transfer either a byte or a word (single transfers to or from system board memory require a minimum of 1000 nanoseconds).

Deactivation of CD CHRDY by a device can extend these accesses for slower I/O or memory devices.

The DMA controller supports the following:

- Register/Program compatibility with the IBM Personal Computer AT® DMA channels (8237 Compatible Mode)
- Control status for the bus control logic
- 16M-byte (24-bit) address capability for memory and 64K-byte (16-bit) address capability for I/O
- Eight independent DMA channels capable of transferring data between memory and I/O devices
- Serial DMA operation with a separate read and write cycle for each transfer operation
- Each channel programmable to byte or word transfer
- Sharing of the system bus interface and control logic
- Extended Operations
  - Extended program control
  - Extended Mode register.

DMA Controller Operations

The DMA Controller does three types of operations. They are:

- Data transfers between memory and I/O devices
- Read verification operations
- Memory refresh cycles.

Data Transfers Between Memory and I/O Devices

The DMA controller performs serial transfers with a minimum of two 125-nanosecond clock cycles for all read and write operations. Read and write operations to the system board memory require three clock cycles. These transfers can be between memory and I/O on any channel. Data is read from a device and latched in the DMA controller before it is written back to a second device. The memory address (target address) needs to be specified only for the DMA data transfer. If the programmable 16-bit I/O address is not selected, the I/O address is forced to hex 0000 during the I/O transfer. A programmable 16-bit I/O address can be provided during the I/O portion of the transfer as a programmable option.

Personal Computer AT is a registered trademark of the International Business Machines Corporation.

Read Verification Operations

The DMA controller can do a memory read operation without a transfer. The address and the count are updated and terminal count provided.

DMA Controller Bus Cycle States

The DMA controller has three basic bus states:

- Idle (Ti)
- Status (Ts)
- Command (Tc).

Idle State (Ti)

The Idle state indicates the DMA controller has no data transfer in progress. It is only during this state that the DMA controller can be programmed. The DMA controller enters this state upon detecting a reset (hardware or software), or upon completing the last DMA transfer for the current active device.

Status State (Ts)

The beginning of the active state is signaled by -S0 or -S1 being driven low by the DMA controller upon activating a channel. The DMA controller sends the status to the bus controller for generation of a command. All bus operations are coded in the status signals as shown in the following figure.

M/-IO	-S0	-S1	DMA Operation
0	0	0	Reserved A
0	0	1	IO Write
0	1	0	IO Read
0	1	1	Reserved B
1	0	0	Reserved C
1	0	1	Memory Write
1	1	0	Memory Read
1	1	1	Reserved D

Figure 3-6. -S0, -S1 Bus States

Command State (Tc)

The second active state is Tc. Memory or I/O devices respond during this period to either a Read or a Write command. Tc states are repeated as long as the 'channel ready' signal (CD CHRDY) is driven inactive to allow time for slow devices to respond.

Byte Pointer

A byte pointer allows 8-bit ports to access consecutive bytes of registers greater than 8 bits. These registers are the Memory Address registers (3 bytes), the Transfer Count registers (2 bytes), and the I/O Address registers (2 bytes).

DMA I/O Address Map

The following figure shows the I/O addresses decoded in the DMA controller.

Address (hex)	Description	Bit Description	Byte Pointer
0000	Channel 0, Memory Address Register	00-15	Yes
0001	Channel 0, Transfer Count Register	00-15	Yes
0002	Channel 1, Memory Address Register	00-15	Yes
0003	Channel 1, Transfer Count Register	00-15	Yes
0004	Channel 2, Memory Address Register	00-15	Yes
0005	Channel 2, Transfer Count Register	00-15	Yes
0006	Channel 3, Memory Address Register	00-15	Yes
0007	Channel 3, Transfer Count Register	00-15	Yes
0008	Channel 0-3, Status Register	00-07	
000A	Channel 0-3, Mask Register (Set/Reset)	00-02	
000B	Channel 0-3, Mode Register (Write)	00-07	
000C	Clear Byte Pointer (Write)	NA	
000D	Master Clear (Write)	NA	
000E	Channel 0-3, Clear Mask Register (Write)	NA	Yes
000F	Channel 0-3, Write Mask Register	00-03	Yes
0018	Extended Function Register (Write)	00-07	
001A	Extended Function Execute	00-07	Yes *
0081	Channel 2, Page Table Address Register **	00-07	
0082	Channel 3, Page Table Address Register **	00-07	
0083	Channel 1, Page Table Address Register **	00-07	
0087	Channel 0, Page Table Address Register **	00-07	
0089	Channel 6, Page Table Address Register **	00-07	
008A	Channel 7, Page Table Address Register **	00-07	
008B	Channel 5, Page Table Address Register **	00-07	
008F	Channel 4, Page Table Address Register **	00-07	
00C0	Channel 4, Memory Address Register	00-15	Yes
00C2	Channel 4, Transfer Count Register	00-15	Yes
00C4	Channel 5, Memory Address Register	00-15	Yes
00C6	Channel 5, Transfer Count Register	00-15	Yes
00C8	Channel 6, Memory Address Register	00-15	Yes
00CA	Channel 6, Transfer Count Register	00-15	Yes
00CC	Channel 7, Memory Address Register	00-15	Yes
00CE	Channel 7, Transfer Count Register	00-15	Yes
00D0	Channel 4-7, Status Register	00-07	
00D4	Channel 4-7, Mask Register (Set/Reset)	00-02	
00D6	Channel 4-7, Mode Register (Write)	00-07	
00D8	Clear Byte Pointer (Write)	NA	
00DA	Master Clear (Write)	NA	
00DC	Channel 4-7, Clear Mask Register (Write)	NA	
00DE	Channel 4-7, Write Mask Register	00-03	

* Dependent upon the function used.
** Upper byte of Memory Address register.

Figure 3-7. DMA I/O Address Map

DMA Registers

All system microprocessor access to the DMA must be 8-bit I/O instructions. The following figure lists the name and size of the DMA registers:

Register	Size (bits)	Quantity of Registers	Allocation
Memory Address	24	8	1 per Channel
I/O Address	16	8	1 per Channel
Transfer Count	16	8	1 per Channel
Temporary Holding	16	1	All Channels
Mask	4	2	1 for Channels 7 - 4 1 for Channels 3 - 0
Arbus	4	2	1 for Channel 4 1 for Channel 0
Mode	8	8	1 per Channel
Status	8	2	1 for Channel 7 - 4 1 for Channel 3 - 0
Function	8	1	All Channels
Refresh	9	1	Independent of DMA

Figure 3-8. DMA Registers

Memory Address Register

Each channel has a 24-bit Memory Address register. The register is loaded by the system microprocessor. The address is incremented or decremented as specified by the Mode register. This register can be read by the system microprocessor in successive I/O byte operations. To read this register, the microprocessor can use the 8237 commands or the extended DMA commands.

I/O Address Register

Each channel has a 16-bit I/O Address register. The register is loaded by the system microprocessor. The bits in this register do not change during DMA transfers. This register can be read by the system microprocessor in successive I/O byte operations. To read this register, the system microprocessor can use only the extended DMA commands.

Transfer Count Register

Each channel has a 16-bit Transfer Count register. The register is loaded by the system microprocessor. The transfer count determines the number of transfers to be executed by the DMA channel before reaching terminal count (TC). The number of transfers is always 1 more than the count specifies. If the count is 0, the DMA does one transfer. The contents of this register does not change during DMA transfers. When the value in the register goes from hex 0000 to FFFF a terminal count pulse is generated by the DMA. This register can be read by the system microprocessor in successive I/O bytes using the 8237 commands or the extended DMA commands.

Temporary Holding Register

This 16-bit register holds the intermediate value for the serial DMA transfer taking place. A serial DMA requires the data to be held in the register before it is written back. This register is not accessible by the system microprocessor.

Mask Register

Bit	Function
7 - 3	Reserved = 0
2	0 Clear Mask Bit 1 Set Mask Bit
1, 0	00 Select Channel 0 or 4 01 Select Channel 1 or 5 10 Select Channel 2 or 6 11 Select Channel 3 or 7

Figure 3-9. Set/Clear Single Mask Bit Using 8237 Compatible Mode

Bit	Function
7 - 4	Reserved = 0
3	0 Unmask Channel 3 or 7 Mask Bit 1 Set Channel 3 or 7 Mask Bit
2	0 Unmask Channel 2 or 6 Mask Bit 1 Set Channel 2 or 6 Mask Bit
1	0 Unmask Channel 1 or 5 Mask Bit 1 Set Channel 1 or 5 Mask Bit
0	0 Unmask Channel 0 or 4 Mask Bit 1 Set Channel 0 or 4 Mask Bit

Figure 3-10. DMA Mask Register Write Using 8237 Compatible Mode

Each channel has a corresponding mask bit that, when set, disables the DMA from servicing the requesting device. Each mask bit can be set or cleared by the system microprocessor. A system reset or DMA master clear sets all mask bits to 1. A Clear Mask Register command sets all mask bits to 0.

When a device requesting DMA cycles wins the arbitration cycle, and the mask bit is set to 1 on the corresponding channel, the DMA controller does not execute any cycles in its behalf and allows external devices to provide the transfer. If no device responds, the bus times out and causes a nonmaskable interrupt (NMI).

Note: There is an exception to this rule. When channel 2 (diskette controller channel) gets a diskette request and the mask bit is set to 1 on this channel, the DMA controller does not arbitrate on behalf of the diskette controller but ignores the diskette request.

This register can be programmed using the 8237 compatible mode commands (used by the IBM Personal Computer AT) or the extended DMA commands.

Arbus Register

This register is used for virtual DMA operations.

Bit	Function
7 - 4	Reserved
3 - 0	Arbitration Level

Figure 3-11. Arbus Register

Mode Register

The Mode register for each channel identifies the type of operation that takes place when that channel is activated.

Bit	Function
7, 6	Reserved = 0
5	Reserved (Must be set to 0)
4	Reserved = 0
3, 2	00 Verify Operation 01 Write Operation 10 Read Operation 11 Reserved
1, 0	00 Select Channel 0 or 4 01 Select Channel 1 or 5 10 Select Channel 2 or 6 11 Select Channel 3 or 7

Figure 3-12. 8237 Compatible Mode Register

The Mode register is programmed by the system microprocessor and its contents reformatted and stored internally in the DMA controller. In the 8237 compatible mode, this register can only be written.

In addition to the 8237 compatible mode, all channels support an extended 8-bit Mode register. This extended mode can be programmed and read by the system microprocessor.

Status Register

The Status register, which can be read by the system microprocessor, contains information about the status of the devices. This information tells which channels have reached terminal count and which channels have requested the bus since the last time the register was read.

Bit	Function
7	Channel 3 or 7 Request
6	Channel 2 or 6 Request
5	Channel 1 or 5 Request
4	Channel 0 or 4 Request
3	TC on Channel 3 or 7
2	TC on Channel 2 or 6
1	TC on Channel 1 or 5
0	TC on Channel 0 or 4

Figure 3-13. Status Register

Bits 3 through 0 in each Status register are set every time a terminal count is reached by a corresponding channel. Bits 7 through 4 are set every time a corresponding channel has controlled the bus. All bits are cleared by reset or following a system microprocessor Status Read command. This register can be read using the 8237 commands or extended DMA commands.

Function Register (F0)

This 8-bit register minimizes I/O address requirements and provides the extended program functions. The system microprocessor loads this register using the data bus.

DMA Extended Operations

The function register supports an extended set of commands for the DMA channels. The system microprocessor uses the following addresses to gain control of the internal DMA registers.

I/O Address (hex)	Command
0018	Function Register
0019	Reserved
001A	Execute Function Register
001B	Reserved

Figure 3-14. DMA Extended Address Decode

The system microprocessor uses the following steps to write to or read from any of the DMA internal registers:

1. Write to the Function register by executing an Out command to address hex 0018, with the proper data to indicate the function and the channel number. The internal Byte Pointer register is always reset to 0 when the Out to address hex 0018 is detected.

Bit	Function
7 - 4	Program Command
3	Reserved = 0
2 - 0	Channel Number

Figure 3-15. DMA Function Register

2. Execute the function by doing an In or Out to address hex 001A. The byte pointer automatically increments by 1 and points to the next byte every time the port address hex 001A is used. This step is not required for Direct commands since they are executed upon detection of the Out to address hex 0018.

Extended Mode Register

DMA supports an Extended Mode register for each channel that can be programmed and read by the system microprocessor. This register is activated whenever a DMA channel requests a DMA data transfer.

The DMA channel must be programmed to match the transfer size of the DMA slave on the channel. Bit 6 of this register is used to program the size of the DMA transfer.

Bit	Function
7	Reserved = 0
6	0 = 8-Bit transfer 1 = 16-Bit transfer
5	Reserved = 0
4	Reserved (Must be set to 0)
3	0 = Read Memory Transfer 1 = Write Memory Transfer
2	0 = Verify 1 = Transfer Data
1	Reserved = 0
0	0 = I/O Address equals 0000H 1 = Use programmed I/O Address

Figure 3-16. Extended Mode Register

Extended Commands

The following figure shows the available commands contained in the Function register.

Registers/Bits Accessed	Bits	Program Command (hex)	Byte Pointer
I/O Address Register	00-15	0	Yes
Reserved		1	
Memory Address Register Write	00-23	2	Yes
Memory Address Register Read	00-23	3	Yes
Transfer Count Register Write	00-15	4	Yes
Transfer Count Register Read	00-15	5	Yes
Status Register Read	00-07	6	
Mode Register	00-07	7	
Arbus Register	00-07	8	
Mask Register Set Single Bit *		9	
Mask Register Reset Single Bit *		A	
Reserved		B	
Reserved		C	
Master Clear *		D	
Reserved		E	
Reserved		F	

* Direct commands to the Function register

Figure 3-17. DMA Extended Commands

The following is an example showing the programming of channel 2 using the 8237 and the extended mode.

Program Step	8237 Compatible Mode	Extended Mode
	OUT to ADRS Data	OUT to ADRS Data
Set Channel Mask Bit	(000AH) x6H	(0018H) 92H
Clear Byte Pointer	(000CH) xxH	(0018H) 22H
Write Memory Address	(0004H) xxH	(001AH) xxH
Write Memory Address	(0004H) xxH	(001AH) xxH
Write Page Table Address	(0081H) xxH	(001AH) xxH
Clear Byte Pointer	(000CH) xxH	(0018H) 42H
Write Register Count	(0005H) xxH	(001AH) xxH
Write Register Count	(0005H) xxH	(001AH) xxH
Write Mode Register	(000BH) xxH	(0018H) 72H (001AH) xxH
Clear Channel 2 Mask Bit	(000AH) x2H	(0018H) A2H
Values inside of ( ) are addresses, x's represent data.		

Figure 3-18. DMA Channel 2 Programming Example, Extended Commands

Virtual DMA Channel Operation

This feature permits programming of the arbitration level assignment for channel 0 and channel 4 using the two 4-bit Arbus registers. These registers enable the system microprocessor to dynamically reassign the arbitration ID value by which the DMA controller responds to bus arbitration for DMA requests. This allows channels 0 and 4 to service devices at any arbitration level. The value of hex F is reserved. The extended command hex 8 programs the Arbus registers. The upper 4 bits of the Function register should be set to a value of 8, and the lower 4 bits should be set to the channel number (0 or 4).

Interrupts

The system provides 16 levels of system interrupts. Any or all of the interrupts may be masked, including the non-maskable interrupt (except for the Watchdog timer). The system board uses two Intel 8259A interrupt controllers. The interrupt controllers are initialized to level sensitive mode. Hardware external to the interrupt controllers prevents initializing edge-triggered mode.

Nonmaskable Interrupt

The nonmaskable interrupt (NMI) signals the system microprocessor that a parity error, a channel check, a system channel time-out, or a system Watchdog timer time-out has occurred. The NMI stops all arbitration on the bus until bit 6 of the Arbitration register (I/O address hex 0090) is set to 0. This can result in lost data or an overrun error on some I/O devices. The NMI masks all other interrupts and the IRET restores the interrupt flag to the state it was in prior to the interrupt. A system reset causes a reset of the NMI.

Nonmaskable interrupt requests from system board parity and channel check are subject to mask control from the NMI Mask bit at I/O address hex 0070. The Watchdog timer and system channel time-out are not masked by this bit. This address is shared with the address for the RT/CMOS RAM (See "RT/CMOS RAM I/O Operations" on page 4-184). The power-on default of the NMI mask is 1 (NMI disabled). Prior to enabling the NMI after a power-on reset (write to address hex 0070 with bit 7 equal to 0) the parity check and channel check state are initialized by the POST.

Warning: When writing port hex 0070 to enable or disable an NMI, a read to port hex 0071 must be accessed immediately. Failure to do this may cause intermittent malfunctions and unreliable operation of the MC146818A Real-Time Clock/Complementary Metal Oxide Semiconductor RAM.

Interrupt Assignments

The following figure shows the interrupt assignments, interrupt levels, and their functions. The interrupt levels are listed in the figure by order of priority. The highest priority is the NMI, and the lowest is IRQ 7.

Level	Function		
NMI	Parity, Watchdog Timer, Arbitration time-out, Channel Check		
IRQ 0	Timer		
IRQ 1	Keyboard		
IRQ 2	Cascade Interrupt Control —	IRQ 8	Real Time Clock
		IRQ 9	Redirect Cascade
		IRQ 10	Reserved
		IRQ 11	Reserved
		IRQ 12	Mouse
		IRQ 13	Math Coprocessor Exception
		IRQ 14	Fixed Disk
		IRQ 15	Reserved
IRQ 3	Serial Alternate		
IRQ 4	Serial Primary		
IRQ 5	Reserved		
IRQ 6	Diskette		
IRQ 7	Parallel Port		

IRQ 8 through 15 are cascaded through IRQ 2

Figure 3-19. Interrupt Level Assignments by Priority

Hardware interrupt IRQ9 is defined as the replacement interrupt level for the cascade level IRQ2. Program interrupt sharing should be implemented on IRQ2, interrupt hex 0A. The following processing occurs to maintain compatibility with the IRQ2 used by IBM Personal Computer products:

1. A device drives the interrupt request active on IRQ2 of the channel.
2. This interrupt request is mapped in hardware to IRQ9 input on the second interrupt controller.
3. When the interrupt occurs, the system microprocessor passes control to the IRQ9 (interrupt hex 71) interrupt handler.
4. This interrupt handler performs an end of interrupt (EOI) to the second interrupt controller and passes control to IRQ2 (interrupt hex 0A) interrupt handler.

5. This IRQ2 interrupt handler, when handling the interrupt, causes the device to reset the interrupt request prior to performing an EOI to the master interrupt controller that finishes servicing the IRQ2 request.

Note: Prior to programming the interrupt controllers, interrupts should be disabled by executing a CLI instruction. This includes the Mask register, EOIs, initialization control words, and operational control words.

System Timers

The system has three programmable timer/counters. They are Channel 0, Channel 2, and Channel 3. Channel 0 and Channel 2 are similar to Channel 0 and Channel 2 of the IBM Personal Computer, IBM Personal Computer XT™, and the IBM Personal Computer AT. Channel 3 does not have a counterpart in earlier IBM Personal Computer systems. The following is a block diagram of the counters.

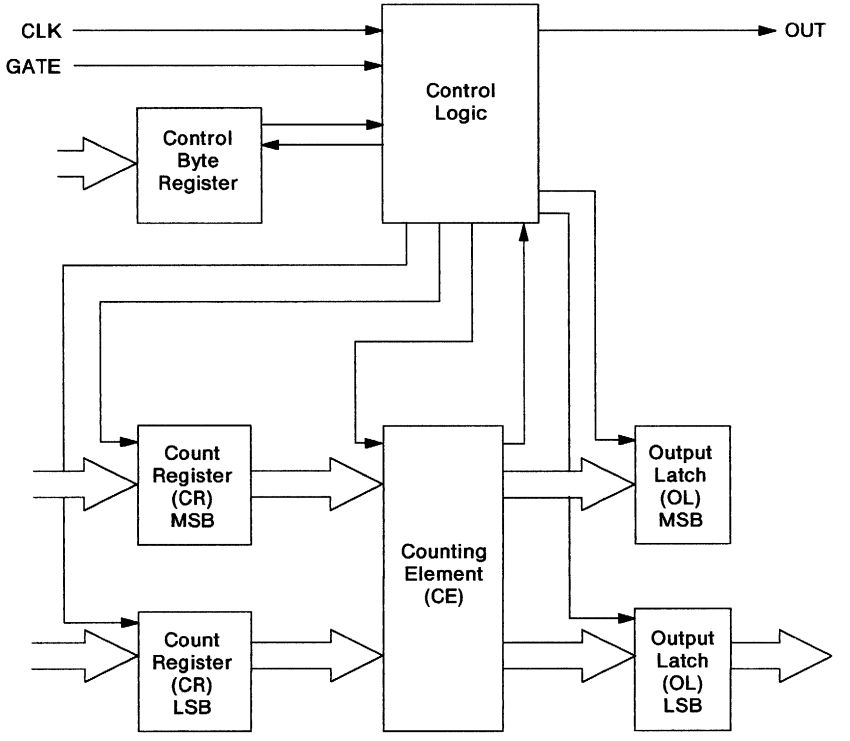


Figure 3-20. Counter Block Diagram

Personal Computer XT is a trademark of the International Business Machines Corporation.

Channel 0 - System Timer

- GATE 0 is always enabled.
- CLK IN 0 is driven by 1.190 MHz.
- CLK OUT 0 indirectly drives 8259A, IRQ 0.

  The 'interrupt request 0' signal (IRQ 0) is driven by a latch. This latch is set by a rising edge of the 'clock out 0' (CLK OUT 0) signal. The latch may be cleared by a system reset, an interrupt acknowledge cycle with a vector of hex 08, or an I/O write to port hex 0061 with bit 7 equal to 1.

  Signals derived from CLK OUT 0 are used to gate and clock Channel 3.

Channel 2 - Tone Generation for Speaker

- GATE 2 is controlled by bit 0 of port hex 0061.
- CLK IN 2 is driven by 1.190 MHz.
- CLK OUT 2 is connected to two places. One connection is to the input port hex 0061, bit 5. CLK OUT 2 is also logically ANDed with port hex 0061, bit 1. The output of the AND gate drives the 'audio sum node' signal.

Channel 3 - Watchdog Timer

This channel operates only in Mode 0, 8-bit binary.

- GATE 3 is tied to IRQ 0.
- CLK IN 3 is tied to CLK OUT 0 inverted.
- CLK OUT 3 when high drives the NMI active.

The Watchdog timer detects when IRQ 0 is active for more than one period of CLK OUT 0. If IRQ 0 is active when a rising edge of CLK OUT 0 occurs, the count is decremented. When the count is decremented to 0 an NMI is generated. Thus, the Watchdog timer may be used to detect when IRQ 0 is not being serviced. This is useful to detect error conditions.

BIOS interfaces are provided to enable and disable the Watchdog timer. When the Watchdog timer times out, it causes an NMI and sets port hex 0092, bit 4. This bit may be cleared by using the BIOS interface to disable the Watchdog timer.

Note: The NMI stops all arbitration on the bus until bit 6 of the Arbitration register (I/O address hex 0090) is set to 0. This can result in lost data or an overrun error on some I/O devices.

If the Watchdog timer is used to detect "tight looping" software tasks that inhibit interrupts, some I/O devices may be overrun (not serviced in time). The operating system may be required to restart these devices.

When the Watchdog timer is enabled, the 'inhibit' signal (INHIBIT) is active only when IRQ 0 is pending for longer than one period of CLK OUT 0. When INHIBIT is active any data written to Channel 0 or Channel 3 is ignored. INHIBIT is never active if the Watchdog timer is disabled.

The Watchdog timer operation is only defined when Channel 0 is programmed in Mode 2 or Mode 3. The operation of the Watchdog timer is undefined when Channel 0 is programmed in any other mode.

Counters 0, 2, and 3

Each counter is independent. Counters 0 and 2 are 16-bit down counters that can be preset. They can count in binary or binary coded decimal (BCD). Counter 3 is an 8-bit down counter that can be preset. It counts in binary only.

Programming the System Timers

The system treats the programmable interval timer as an arrangement of five external I/O ports. Three ports are treated as count registers and two are control registers for mode programming. Counters are programmed by writing a control word and then an initial count. All control words are written into the Control Word registers, which are located at I/O Address hex 0043 for Counters 0 and 2, and I/O address hex 0047 for Counter 3. Initial counts are written into the Count registers, not the Control Word registers. The format of the initial count is determined by the control word used.

The count is written to the Count register. It is then transferred to the counting element, according to the mode definition. When the count is read, the data is presented by the output latch.

Counter Write Operations

- The control word must be written before the initial count is written.
- The count must follow the count format specified in the control word.

A new initial count may be written to the counters at any time without affecting the counter's programmed mode. Counting is affected as described in the mode definitions. The new count must follow the programmed count format.

Counter Read Operations

The counters can be read using the Counter Latch command. For more information on the Counter Latch command, see "Counter Latch Command" on page 3-38.

If the counter is programmed for 2-byte counts, 2 bytes must be read. The 2 bytes need not be read consecutively; read, or write, or programming operations of other counters may be inserted between them.

Note: If the counters are programmed to read or write 2-byte counts, the program must not transfer control between writing the first and second byte to another routine that also reads or writes into the same counter. This will cause an incorrect count.

Registers

I/O Address (hex)	Register
0040	Read/Write Counter 0
0042	Read/Write Counter 2
0043	Write Control Byte for counter 0 or 2
0044	Read/Write Counter 3
0047	Write Control Byte for counter 3

Figure 3-21. System Timer/Counter Registers

Port Hex 0040 - Count Register - Channel 0

The control byte is written to port hex 0043 indicating the format of the count (least-significant byte only, most-significant byte only, or least-significant byte followed by most-significant byte). This must be done before writing the count to I/O port hex 0040.

Port Hex 0042 - Count Register - Channel 2

The control byte is written to port hex 0043 indicating the format of the count (least-significant byte only, most-significant byte only, or least-significant byte followed by most-significant byte). This must be done before writing the count to I/O port hex 0042.

Port Hex 0043 - Control Byte - Channel 0 or 2

This is a write-only register. The following gives the format for the control byte (I/O port hex 0043) for counters 0 and 2.

Bit 7 SC1

Bit 6 SC0

SC1	SC0	
0	0	Select Counter 0
0	1	Reserved
1	0	Select Counter 2
1	1	Reserved

Figure 3-22. SC - Select Counter, Port Hex 0043

Bit 5 RW1

Bit 4 RW0

RW1	RW0	
0	0	Counter Latch Command
0	1	Read/Write Counter Bits 0 - 7 only
1	0	Read/Write Counter Bits 8 - 15 only
1	1	Read/Write Counter Bits 0 - 7 first, then bits 8 - 15

Figure 3-23. RW - Read/Write Counter, Port Hex 0043

Bit 3 M2

Bit 2 M1

Bit 1 M0

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

Don't care bits (X) should be set to 0.

Figure 3-24. M - Counter Mode

Bit 0 BCD

0	Binary Counter 16 bits
1	Binary Coded Decimal Counter (4 Decades)

Figure 3-25. Binary Coded Decimal (BCD)

Port Hex 0044 - Count Register - Channel 3

The control byte is written to port hex 0047 indicating the format of the count (least-significant byte only). This must be done before writing the count to I/O port hex 0044.

Port Hex 0047 - Control Byte - Channel 3

This is a write-only register. The following gives the format for the control byte (I/O port hex 0047) for counter 3.

Bit 7 SC1

Bit 6 SC0

SC1	SC0	
0	0	Select Counter 3
0	1	Reserved
1	0	Reserved
1	1	Reserved

Figure 3-26. SC - Select Counter, Port Hex 0047

Bit 5 RW1

Bit 4 RW0

RW1	RW0	
0	0	Counter Latch Command Select Counter 0
0	1	R/W Counter Bits 0 - 7 only
1	0	Reserved
1	1	Reserved

Figure 3-27. RW - Read/Write Counter, Port Hex 0047

Bits 3 - 0 0

Counter Latch Command

The Counter Latch command is written to the Control Byte register. The SC0 and SC1 bits select the counter and bits D5 and D4 distinguish this command from a control byte. The following figure shows the format of the Counter Latch command.

Bit	Function
7	SC1 - Specifies the Counter to be latched
6	SC0 - Specifies the Counter to be latched
5	0 - Specifies the Counter Latch command
4	0 - Specifies the Counter Latch command
3	0 - Reserved = 0
2	0 - Reserved = 0
1	0 - Reserved = 0
0	0 - Reserved = 0

Figure 3-28. Counter Latch Command

The count is latched into the selected counter's output latch at the time the Counter Latch command is received. This count is held in the latch until it is read by the system microprocessor (or until the counter is reprogrammed). After the count is read by the system microprocessor it is automatically unlatched and the output latch (OL) returns to following the counting element (CE). Counter Latch commands do not affect the programmed mode of the counter in any way. All subsequent latch commands to a given counter, issued before the count is read, are ignored. A read cycle to the counter latch returns the value latched by the first Counter Latch command.

System Timer Modes

The following definitions are used when describing the timer modes.

CLK pulse A rising edge, then a falling edge on the counter CLK input.

Trigger A rising edge on a counter's input GATE.

Counter Load The transfer of a count from the Counter register to the counting element.

Mode 0 - Interrupt on Terminal Count

Event counting can be done using Mode 0. Counting is enabled when GATE is equal to 1, and disabled when GATE is equal to 0. If GATE is equal to 1 when the control byte and initial count are written to the counter, the sequence is as follows:

1. The control byte is written to the counter, and OUT goes low.
2. The initial count is written.
3. Initial count is loaded on the next CLK pulse. The count is not decremented for this CLK pulse.

   The count is decremented until the counter reaches 0. For an initial count of N, the counter reaches 0 after N+1 CLK pulses.
4. OUT goes high.

OUT remains high until a new count or new Mode 0 control byte is written into the counter.

If GATE equals 0 when an initial count is written to the counter, it is loaded on the next CLK pulse even though counting is not enabled. After GATE enables counting, OUT will go high N CLK pulses later.

If a new count is written to a counter while counting, it is loaded on the next CLK pulse. Counting then continues from the new count. If a 2-byte count is written to the counter the following occurs:

1. The first byte written to the counter disables the counting. OUT goes low immediately, and there is no delay for the CLK pulse.
2. When the second byte is written to the counter, the new count is loaded on the next CLK pulse. Again, OUT goes high when the counter reaches 0.

Mode 1 - Hardware Retriggerable One-Shot

The sequence for Mode 1 is as follows:

1. OUT is high.
2. On the CLK pulse following a trigger, OUT goes low, and begins the one-shot pulse.
3. When the counter reaches zero, OUT goes high.

OUT remains high until the CLK pulse after the next trigger.

The counter is armed by writing the control word and initial count to the counter. When a trigger occurs, the counter is loaded. OUT goes low on the next CLK pulse, starting the one-shot pulse. For an initial count of N, a one-shot pulse is N CLK pulses long. The one-shot pulse repeats the same count of N for the next triggers. OUT remains low N CLK pulses following any trigger. GATE does not affect OUT. The current one-shot pulse is not affected by a new count written to the counter, unless the counter is retriggered. If the counter is retriggered, the new count is loaded and the one-shot pulse continues.

Note: Mode 1 is only valid on Counter 2.

Mode 2 - Rate Generator

This mode causes the counter to perform a divide-by-N function. Counting is enabled when GATE is equal to 1, and disabled when GATE is equal to 0.

The sequence for Mode 2 is as follows:

1. OUT is high.
2. The initial count decrements to 1.
3. OUT goes low for one CLK pulse.
4. OUT goes high.
5. The counter reloads the initial count.
6. The process is repeated.

If GATE goes low during the OUT pulse, OUT goes high. On the next CLK pulse a trigger reloads the counter with the initial count. OUT goes low N CLK pulses after the trigger. This allows the GATE input to be used to synchronize the counter.

The counter is loaded on the next CLK pulse after a control byte and initial count are written to the counter. OUT goes low N CLK pulses after writing the initial count. This allows software synchronization of the counter.

The current counting sequence is not effected by a new count being written to the counter. If the counter receives a trigger after a new count is written and before the end of the current count, the new count is loaded on the next CLK pulse and counting continues from the new count. If the trigger is not received by the counter, the new count is loaded following the current counting cycle.

Mode 3 - Square Wave

Mode 3 is similar to Mode 2 except for the duty cycle of OUT. Counting is enabled when GATE is equal to 1, and disabled when GATE is equal to 0. An initial count of N results in a square wave on the 'out' signal. The period of the square wave is N CLK pulses. If OUT is low and GATE goes low, OUT goes high. On the next CLK pulse, a trigger reloads the counter with the initial count.

After writing a control byte and initial count, the counter is loaded on the next CLK pulse. This allows software synchronization of the counter.

The current counting sequence is not effected by a new count being written to the counter. If the counter receives a trigger after a new count is written, and before the end of the current count half-cycle of the square wave, the new count is loaded on the next CLK pulse, and counting continues from the new count. If the trigger is not received by the counter, the new count is loaded following the current half-cycle.

The implementation of Mode 3 differs, depending on whether the count written is an odd or even number. If the count is even, OUT begins high and the following applies:

1. The initial count is loaded on the first CLK pulse.

2. The count is decremented by 2 on succeeding CLK pulses.
3. The count decrements to 0.
4. OUT changes state.
5. The counter is reloaded with the initial count.
6. The process repeats indefinitely.

If the count is odd, the following applies:

1. OUT is high.
2. The initial count minus 1 is loaded on the first CLK pulse.
3. The count is decremented by 2 on succeeding CLK pulses.
4. The count decrements to 0.
5. One CLK pulse after the count reaches 0, OUT goes low.
6. The counter is reloaded with the initial count minus 1.
7. Succeeding CLK pulses decrement the count by 2.
8. The count decrements to 0.
9. OUT goes high.
10. The counter is reloaded with the initial count minus 1.
11. The process repeats indefinitely.

Mode 3, using an odd count, causes OUT to go high for a count of (N+1)/2 and low for a count of (N-1)/2.

Mode 3 may operate such that OUT is initially set low, when the control byte is written. For this condition, Mode 3 operates as follows:

1. OUT is low.
2. The count decrements to half of the initial count.
3. OUT goes high.
4. The count decrements to 0.
5. OUT goes low.
6. The process repeats indefinitely.

This process results in a square wave with a period of N CLK pulses.

Note: If it is required that OUT be high after the control byte is written, the control byte must be written twice. This applies only to Mode 3.

Mode 4 - Software Retriggerable Strobe

Counting is enabled when GATE is equal to 1, and disabled when GATE is equal to 0. Counting is triggered when an initial count is written.

The sequence for Mode 4 is as follows:

1. OUT is high.
2. The control byte and initial count are written to the counter.
3. The initial count is loaded on the next CLK pulse. The count is not decremented for this clock pulse.
4. The count is decremented to 0. For an initial count of N, the counter reaches 0 after N + 1 CLK pulses.
5. OUT goes low for one CLK pulse.
6. OUT goes high.

GATE should not go low one half CLK pulse before or after OUT goes low. If this occurs, OUT remains low until GATE transitions high.

If a new count is written to a counter while counting, it is loaded on the next CLK pulse. Counting then continues from the new count. If a 2-byte count is written, the following occurs:

1. Writing the first byte does not affect counting.
2. The new count is loaded on the next CLK pulse after writing the second byte.

The Mode 4 sequence can be retriggered by software. The period from when the new count of N is written to when OUT strobes low is (N + 1) pulses.

Mode 5 - Hardware Retriggerable Strobe

The sequence for Mode 5 is as follows:

1. OUT is high.
2. The control byte and initial count are written to the counter.
3. Counting is triggered by a rising edge of GATE.
4. The counter is loaded on the next CLK pulse after the trigger. This CLK pulse does not decrement the count.
5. The count is decremented to 0.
6. OUT goes low for one CLK pulse. This occurs (N + 1) CLK pulses after the trigger.
7. OUT goes high.

The counting sequence can be retriggered. OUT strobes low (N + 1) pulses after the trigger. GATE does not effect OUT.

The current counting sequence is not effected by a new count being written to the counter. If the counter receives a trigger after a new count is written and before the end of the current count, the new count is loaded on the next CLK pulse and counting continues from the new count.

Note: Mode 5 is valid only on Counter 2.

Operations Common to All Modes

Control bytes written to a counter cause all control logic to reset. OUT goes to a known state. This does not take a CLK pulse.

The falling edge of the CLK pulse is when new counts are loaded and counters are decremented.

Counters do not stop when they reach zero. In Modes 0, 1, 4, and 5 the counter wraps around to the highest count, and continues counting. Modes 2 and 3 are periodic; the counter reloads itself with the initial count and continues from there.

The GATE is sampled on the rising edge of the CLK pulse.

The following shows the minimum and maximum initial counts for the counters.

Mode	Min Count	Max Count
0	1	0 = 2^{16} (Binary Counting) or 10^4 (BCD Counting)
1	1	0 = 2^{16} (Binary Counting) or 10^4 (BCD Counting)
2	2	0 = 2^{16} (Binary Counting) or 10^4 (BCD Counting)
3	2	0 = 2^{16} (Binary Counting) or 10^4 (BCD Counting)
4	1	0 = 2^{16} (Binary Counting) or 10^4 (BCD Counting)
5	1	0 = 2^{16} (Binary Counting) or 10^4 (BCD Counting)

Figure 3-29. Minimum and Maximum Initial Counts, Counters 0, 2

Counter 3 can use only Mode 0 - Interrupt on Terminal Count. The minimum initial count is 1 and the maximum is hex FF.

Power-On Password

RT/CMOS RAM has 8 bytes reserved for the password and its check character. The 8 bytes are initialized to hex 00. The microprocessor can only access these bytes during power-on self-test (POST). After POST is completed, if a password is installed, the password bytes are locked and cannot be accessed by a program. A password can be from 1 to 7 characters.

During password installation, the password (1 to 7 keyboard scan codes), is stored in the security space.

Password installation is a function of a program contained on the Reference diskette. Once the password utility has been installed, the password can be changed only during the POST. When the new password is installed, changed, or removed, the password is not visible on the display.

The system unit cover can be physically locked to prevent unauthorized access to the battery. This helps prevent unauthorized battery removal and loss of password and configuration information.

Audio Subsystem

The audio subsystem is a 66 mm (2.6 in.) speaker driven by a linear amplifier. The linear amplifier input node can be driven from the following sources:

- System-timer Channel 2 can be enabled to drive the speaker using bit 1 of I/O port hex 0061 set to 1. For information about system timer Channel 2 see "System Timers" on page 3-31.
- The channel using the 'audio sum node' signal.

The following block diagram shows the audio subsystem.

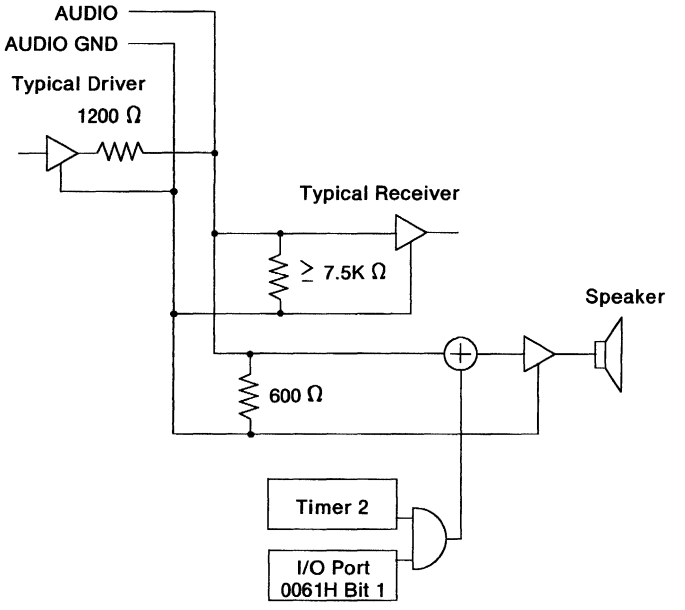


Figure 3-30. Audio Subsystem Block Diagram

Each audio driver must have a 1200 ohm source impedance, and a 7.5 kilohm or greater impedance is required for each audio receiver. Volume control is provided by the driver. Output level is a function of the number of drivers and receivers that share the AUDIO line.

Logic ground is connected to AUDIO GND at the amplifier.

Section 4. System Board I/O Controllers

Notes:

Keyboard/Auxiliary Device Controller

The keyboard/auxiliary device controller is a function of the Intel 8042 chip. The keyboard is connected to one of the two controller connectors in the rear of the system unit. This connector is dedicated to the keyboard. An auxiliary device connects to the other controller connector. The auxiliary device may be any type of serial input device compatible with the 8042 interface. Some of these are:

- Mouse
- Touchpad
- Trackball
- Keyboard.

The 8042 receives the serial data, checks the parity, translates keyboard scan codes (see bit 6 of the 8042 Command byte on page 4-9), and presents the data to the system as a byte of data at data port I/O address hex 0060. The interface can interrupt the system when data is available or can wait for polling from the microprocessor.

I/O address hex 0064 is the command/status port. When the system reads port hex 0064 it receives status information from the 8042. When the system writes to the port, the 8042 interprets the byte as a command.

Secondary circuit protection is provided on the system board for the +5 Vdc line to the keyboard and auxiliary device.

Keyboard Password Security

The 8042 provides for a password security mechanism. Three commands are available regarding password operation:

A4	Test Password Installed
A5	Load Security
A6	Enable Security.

The system microprocessor may issue a Test Password Installed command to determine if a password is currently installed. This feature allows the controlling program to decide whether or not to write over the existing password.

The system microprocessor may issue a Load Security command and set a password in the 8042 at any time. Any existing password is lost, and the new password becomes the active password. The password must be installed in scan code format.

The system microprocessor must issue the Enable Security command to set the 8042 into secure mode. At this point the 8042 does not pass any information along to the system microprocessor. The 8042 intercepts the keyboard data stream, continuously comparing it to the installed password pattern. Until a match is encountered, all keyboard and auxiliary device data are not passed to the system microprocessor. When a match occurs, the state of the 8042 is restored and data is allowed to pass to the system microprocessor.

The password may be changed as often as the system microprocessor chooses. No command to verify the installed password is provided. No commands are accepted by the 8042 when keyboard security is active.

8042 Command and Status Bytes

The following shows the 8042 Command and Status bytes:

Bit	Function
7	Reserved = 0
6	IBM Keyboard Translate Mode
5	Disable Auxiliary Device
4	Disable Keyboard
3	Reserved = 0
2	System Flag
1	Enable Auxiliary Interrupt
0	Enable Keyboard Interrupt

Figure 4-1. 8042 Command Byte, Port Hex 0064 Write

Bit 7 Reserved

Bit 6 When this bit is set to 1, the 8042 translates the incoming scan codes to scan code set 1 (used on the IBM Personal Computer and IBM Personal Computer XT). When this bit is set to 0, the 8042 passes the keyboard scan codes without translation.

Bit 5 Writing a 1 to this bit disables the auxiliary device interface by driving the 'clock' line low. Data is not sent or received.

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

Bit 3 Reserved

Bit 2 The value written to this bit is placed in the system flag bit of the 8042 Status register.

Bit 1 Writing a 1 to this bit causes the 8042 to generate an interrupt when it places auxiliary device data into its output buffer.

Bit 0 Writing a 1 to this bit causes the 8042 to generate an interrupt when it places keyboard data into its output buffer.

Bit	Function
7	Parity Error
6	General Time Out
5	Auxiliary Output Buffer Full
4	Inhibit Switch
3	Command/Data
2	System Flag
1	Input Buffer Full
0	Output Buffer Full

Figure 4-2. 8042 Status Byte, Port Hex 0064 Read

Input and Output Buffers

The output buffer is an 8-bit read-only register at I/O address hex 0060. When the output buffer is read, the 8042 uses it to send information to the system microprocessor. The information can be scan codes received from the keyboard, data from an auxiliary device, or data bytes that result from a command from the system microprocessor.

The input buffer is an 8-bit write-only register at I/O address hex 0060. When the input buffer is written, a flag is set that indicates a data write. Data written to I/O address hex 0060 is sent to the keyboard unless the 8042 is expecting a data byte following a 8042 command.

Data should be written to the 8042 input buffer only if the Input Buffer Full bit (bit 1) in the Status register (I/O address hex 0064) is equal to 0.

8042 Commands

A command is a data byte written to the 8042 through I/O address hex 0064. The following are the recognized commands, shown in hex values.

20-3F Read the 8042 RAM — Bits D5-D0 specify the address.

20 Read the 8042 Command Byte — The 8042 puts the command byte in its output buffer.

60-7F Write the 8042 RAM — Bits D5-D0 specify the address.

60 Write the 8042 Command Byte — The next byte of data written to I/O address hex 0060 is placed in the 8042 command byte. For more information on the 8042 Command byte see Figure 4-1 on page 4-9.

A4 Test Password Installed — This command checks if there is currently a password installed in the 8042. The test result is placed in the output buffer (I/O address hex 0060 and IRQ01). Hex FA means that the password is installed and hex F1 means that the password is not installed.

A5 Load Security — This command initiates the Password Load procedure. Following this command the 8042 will input from the data port until a null (0) is detected. The null terminates password entry.

A6 Enable Security — This command enables the 8042 security feature. This command is valid only when a password pattern is currently loaded into the 8042.

A7 Disable Auxiliary Device Interface — This command sets bit 5 of the 8042 command byte. This disables the auxiliary device interface by driving the 'clock' line low. Data is not sent or received.

A8 Enable Auxiliary Device Interface — This command clears bit 5 of the 8042 command byte, which releases the auxiliary device interface.

A9 Interface Test — This command causes the 8042 to test the auxiliary device clock and data lines. The test result is placed in the output buffer (I/O address hex 0060 and IRQ01) as shown in the following figure.

Test Result (hex)	Meaning
00	No Error Detected
01	The Auxiliary device 'Clock' line is stuck low.
02	The Auxiliary device 'Clock' line is stuck high.
03	The Auxiliary device 'Data' line is stuck low.
04	The Auxiliary device 'Data' line is stuck high.

Figure 4-3. Command A9 Test Results

AA Self Test — This command causes the 8042 to perform internal diagnostic tests. A hex 55 is placed in the output buffer if no errors are detected.

AB Interface Test — This command causes the 8042 to test the keyboard 'clock' and 'data' lines. The test result is placed in the output buffer (I/O address hex 0060 and IRQ01) as shown in the following figure.

Test Result (hex)	Meaning
00	No Error Detected
01	The keyboard 'Clock' line is stuck low.
02	The keyboard 'Clock' line is stuck high.
03	The keyboard 'Data' line is stuck low.
04	The keyboard 'Data' line is stuck high.

Figure 4-4. Command AB Test Results

AC Reserved

AD Disable Keyboard Interface — This command sets bit 4 of the 8042 Command byte. This disables the keyboard interface by driving the 'clock' line low. Data will not be sent or received.

AE Enable Keyboard Interface — This command clears bit 4 of the 8042 command byte, which releases the keyboard interface.

C0 Read Input Port — This command causes the 8042 to read its input port and place the data in its output buffer. This command should be used only if the output buffer is empty.

C1 Poll Input Port Low — Port 1 bits 0-3, in Status bits 4-7.

C2 Poll Input Port High — Port 1 bits 4-7, in Status bits 4-7.

D0 Read Output Port — This command causes the 8042 to read its output port and place the data in its output buffer. This command should be used only if the output buffer is empty.

D1 Write Output Port — The next byte of data written to I/O address hex 0060 is placed in the 8042 output port.

Note: Bit 0 of the 8042 output port is connected to System Reset. This bit should not be written low.

D2 Write Keyboard Output Buffer — The next byte written to I/O address hex 0060 input buffer is written to I/O address hex 0060 output buffer as if initiated by a device. An interrupt occurs if the interrupt is enabled in the Command byte.

D3 Write Auxiliary Device Output Buffer — The next byte written to I/O address hex 0060 input buffer is written to I/O address hex 0060 output buffer as if initiated by a device. An interrupt occurs if the interrupt is enabled in the Command byte.

D4 Write to Auxiliary Device — The next byte written to I/O address hex 0060 input buffer is transmitted to the auxiliary device.

E0 Read Test Inputs — This command causes the 8042 to read its T0 and T1 inputs. This data is placed in the output buffer. Data bit 0 represents T0 and data bit 1 represents T1.

F0-FF Pulse Output Port — Bits 0 through 3 of the 8042 output port may be pulsed low for approximately 6 microseconds. Bits 0 through 3 of this command indicate which bits are to be pulsed. A 0 indicates that the bit should be pulsed, a 1 indicates the bit should not be modified.

Note: Bit 0 of the 8042 output port is connected to System Reset. Pulsing of this bit resets the system microprocessor.

Keyboard/Auxiliary Device Programming Considerations

The following are some programming considerations for the Keyboard/Auxiliary device controller.

I/O address hex 0064 (Status register) can be read at any time.

I/O address hex 0060 should be read only when the Output Buffer Full bit in the Status register is a 1.

The Auxiliary Output Buffer Full bit in the Status register indicates that the data in I/O address hex 0060 came from the Auxiliary Device. This bit is valid only when the Output Buffer Full bit is a 1.

I/O address hex 0060 and I/O address hex 0064 should be written only when the Status register Input Buffer Full bit and the Output Buffer Full bit is 0.

The devices connected to the 8042 should be disabled before initiating a command that generates output. If output is generated, any value in the output buffer is overwritten.

An external latch is used to hold the level sensitive IRQ until an I/O read from address hex 0060 is executed by the system

Auxiliary Device/System Timings

Data transmissions to and from the auxiliary device connector consist of an 11-bit data stream sent serially over the 'data' line. The following figure shows the function of each bit.

Bit	Function
11	Stop bit (always 1)
10	Parity bit (odd parity)
9	Data bit 7 (most-significant)
8	Data bit 6
7	Data bit 5
6	Data bit 4
5	Data bit 3
4	Data bit 2
3	Data bit 1
2	Data bit 0 (least-significant)
1	Start bit (always 0)

Figure 4-5. Auxiliary Device Data Stream Bit Definitions

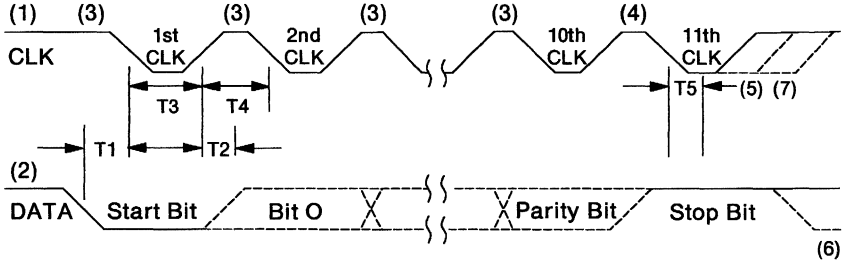
The parity bit is either 1 or 0, and the 8 data bits, plus the parity bit, always have an odd number of 1's.

System Receiving Data

The following describes the typical sequence of events that takes place when the system is receiving data from the auxiliary device. A graphic representation showing the timing relationships is presented in Figure 4-6 on page 4-15.

1. The auxiliary device checks the 'clock' line. If the line is inactive, output from the device is not allowed.
2. The auxiliary device checks the 'data' line. If the line is inactive, the device receives data from the system.

3. The auxiliary device checks the 'clock' line periodically during the transmission at intervals not exceeding 100 microseconds. If the device finds the system holding the 'clock' line inactive, the transmission is terminated. The system can terminate transmission anytime during the first ten clock cycles.
4. A final check for terminated transmission is performed at least 5 microseconds after the tenth clock.
5. The system can hold the 'clock' signal inactive to inhibit the next transmission.
6. The system can set the 'data' line inactive if it has a byte to transmit to the device. When the 'data' line is inactive, the system has data to transmit. The 'data' line is set inactive when the Start bit (always 0) is placed on the 'data' line.
7. The system raises the 'clock' line to allow the next transmission.



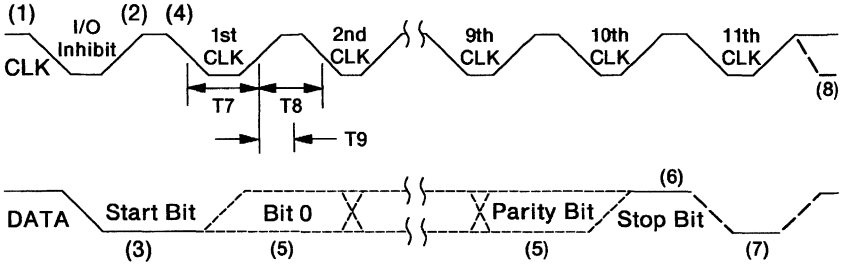
	Timing Parameter	Min/Max
T1	Time from DATA transition to falling edge of CLK	5 / 25 μsec
T2	Time from rising edge of CLK to DATA transition	5 / T4 - 5 μsec
T3	Duration of CLK inactive	30 / 50 μsec
T4	Duration of CLK active	30 / 50 μsec
T5	Time to auxiliary device inhibit after clock 11 to ensure the auxiliary device does not start another transmission	>0 / 50 μsec

Figure 4-6. Receiving Data Timings

System Sending Data

The following describes the typical sequence of events that takes place when the system is sending data to the auxiliary device. A graphic representation showing the timing relationships is presented in Figure 4-7 on page 4-17.

1. The system checks for an auxiliary device transmission in process. If a transmission is in process and beyond the tenth clock, the system must receive the data.
2. The auxiliary device checks the 'clock' line. If the line is inactive, an I/O operation is not allowed.
3. The auxiliary device checks the 'data' line. If the line is inactive, the system has data to transmit. The 'data' line is set inactive when the Start bit (always 0) is placed on the 'data' line.
4. The auxiliary device sets the 'clock' line inactive. The system then places the first bit on the 'data' line. Each time the auxiliary device sets the 'clock' line inactive, the system places the next bit on the 'data' line until all bits are transmitted.
5. The auxiliary device samples the 'data' line for each bit while the 'clock' line is active. Data must be stable within 1 microsecond after the rising edge of the 'clock' line.
6. The auxiliary device checks for a positive level stop bit after the tenth clock. If the 'data' line is inactive, the auxiliary device continues to clock until the 'data' line becomes active, clocks the line-control bit, and at the next opportunity sends a Resend command to the system.
7. The auxiliary device pulls the 'data' line inactive, producing the line-control bit.
8. The system can pull the 'clock' line inactive, inhibiting the auxiliary device.



	Timing Parameter	Min/Max
T7	Duration of CLK inactive	30 / 50 μsec
T8	Duration of CLK active	30 / 50 μsec
T9	Time from inactive to active CLK transition, used to time when the auxiliary device samples DATA	5 / 25 μsec

Figure 4-7. Sending Data Timings

Signals

The keyboard and auxiliary device signals are driven by open-collector drivers pulled to 5 Vdc through 10 kilohm resistors. The following lists the characteristics of the signals.

Sink Current	20 mA	Maximum
High-level Output Voltage	5.0 Vdc minus pullup	Minimum
Low-level Output Voltage	0.5 Vdc	Maximum
High-level Input Voltage	2.0 Vdc	Minimum
Low-level Input Voltage	0.8 Vdc	Maximum

Figure 4-8. Keyboard/Auxiliary Device Signals

Connector

The keyboard and the auxiliary device connectors use 6-pin miniature DIN connectors. The signals and voltages are the same for both connectors. The following are the voltages and signals assigned to the keyboard and auxiliary device connectors.

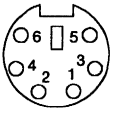


Figure 4-9. Keyboard and Auxiliary Device Connectors

Pin	I/O	Signal Name
1	I/O	Data
2	NA	Reserved
3	NA	Ground
4	NA	+5 Vdc
5	I/O	Clock
6	NA	Reserved

Figure 4-10. Keyboard and Auxiliary Device Connectors Signal and Voltage Assignments

Video Subsystem

The system video is generated by the IBM Video Graphics Array (VGA) and its associated circuitry. The associated circuitry consists of the video memory and a video digital to analog converter (DAC). The 256K bytes of video memory consists of four 64K by 8 memory maps. The red, green, and blue (RGB) outputs from the video DAC drive 31.5 kHz direct drive analog displays.

All video modes available on the IBM Monochrome Display Adapter, IBM Color/Graphics Monitor Adapter, and IBM Enhanced Graphics Adapter are supported, regardless of which analog display is connected. All video modes supported by the video subsystem are available on all of the supported analog displays. Colors will be displayed as shades of gray when the monochrome analog display is connected.

The new modes available are:

- 640 x 480 graphics in both 2 and 16 colors
- 720 x 400 alphanumeric in both 16-color and monochrome
- 360 x 400 16-color alphanumeric
- 320 x 200 graphics with 256 colors.

In addition, all 200 line modes are double scanned by the video subsystem and displayed as 400 lines on the display. This means that each 1-PEL-high horizontal scan line is displayed twice on the display.

The VGA does the interfacing between the system microprocessor and video memory. All data passes through the VGA when the system microprocessor writes to or reads from video memory. The VGA controls the arbitration for video memory between the system microprocessor and the cathode-ray tube (CRT) controller function contained within the VGA. Programs do not need to wait for horizontal retrace to update the display buffer. The system microprocessor will receive better performance when accessing the display buffer during non-active display times, because there is less interference from the CRT controller.

Video memory addressing is controlled by the VGA. The starting address of the video memory is programmable to three different starting addresses for compatibility with previous video adapters. BIOS will program the VGA appropriately during a video mode set.

In alphanumeric modes, the system microprocessor writes ASCII character code and attribute data to the video memory maps 0 and 1 respectively. The character generator is stored in video map 2 and is loaded by BIOS during an alphanumeric video mode set. BIOS downloads the character generator data (character generator = font = character set) from system ROM. Three fonts are contained in ROM. Two fonts contain dot patterns identical to those provided by the IBM Monochrome Display Adapter, the IBM Color/Graphics Monitor Adapter, and the IBM Enhanced Graphics Adapter. The third font is a new 8 x 16 character font. Up to eight 256-character fonts can be loaded into video memory map 2 at one time (the IBM Enhanced Graphics Adapter allows up to four fonts). A BIOS interface exists to load user-defined fonts. As on the IBM Enhanced Graphics Adapter, a register selects which font is actually used to form characters. Also, as on the IBM Enhanced Graphics Adapter, the intensity bit in the attribute byte may be redefined as a switch between two 256-character fonts. This allows 512 characters to be displayed on the screen at one time. See "Character Map Select Register," on page 4-68, and "RAM Loadable Character Generator" on page 4-112 for more information.

The VGA formats the information stored in video memory into an 8-bit digital value that is sent to the video DAC. This 8-bit value allows access to a maximum of 256 registers inside the video DAC. For example, in the 2-color graphics modes, only two different 8-bit values would be presented to the video DAC. In the 256-color graphics mode, 256 different 8-bit values would be presented to the video DAC. Each register inside the video DAC contains a color value that is selected from a choice of 256K colors.

The video DAC outputs three analog color signals (red, green, and blue) that are sent to the 15-pin display connector. The monochrome analog display uses only the green analog output. This output is used to determine the shade of gray that will be displayed.

The video subsystem supports attachment of only 31.5 kHz direct drive analog displays. Other IBM displays are *not* supported because

they have digital interfaces, or have a different horizontal sweep frequency.

A BIOS call exists to enable or disable the VGA. Disable means that the VGA will not respond to video memory or I/O reads or writes. The contents of registers and video memory are preserved with the values present when the disable is invoked. Because of this, the VGA continues to generate valid video output if it is doing so before it is disabled.

Compatibility with other hardware is best achieved by using the BIOS interface whenever possible. If an application is forced to write directly to the VGA, the following rules should be followed.

- When programming address registers, all currently reserved bits should be set to 0 to maximize compatibility with other hardware.
- When programming data registers, all currently reserved bits should be read out and written back unmodified to maximize compatibility with other hardware.

Previous video adapters required that the video mode used correspond to the display attached. For example, the IBM Enhanced Graphics Adapter required that the Enhanced Color Display be attached to run mode hex (3*), and required that the monochrome display be attached to run mode hex 7. All the modes supported by the VGA are supported by the IBM 31.5 kHz direct drive analog displays. Colors are displayed as shades of gray when the monochrome analog display is connected. Circuitry exists on the system board to detect which type of analog display is connected (color or monochrome). BIOS maps (sums) the colors into shades of gray.

The Auxiliary Video connector is a 20-pin connector located in line with one of the channel connectors on the system board. This connector allows passing of video data from the VGA to an adapter plugged into a channel connector. Also, the system board video buffers can be turned off and video from the adapter can drive the video DAC and the 15-pin connector that drives the analog display. The full channel is available for use by the adapter. For more information on the Auxiliary Video Connector see "Auxiliary Video Connector" on page 4-119.

The following is a block diagram of the video subsystem, which is part of the system board.

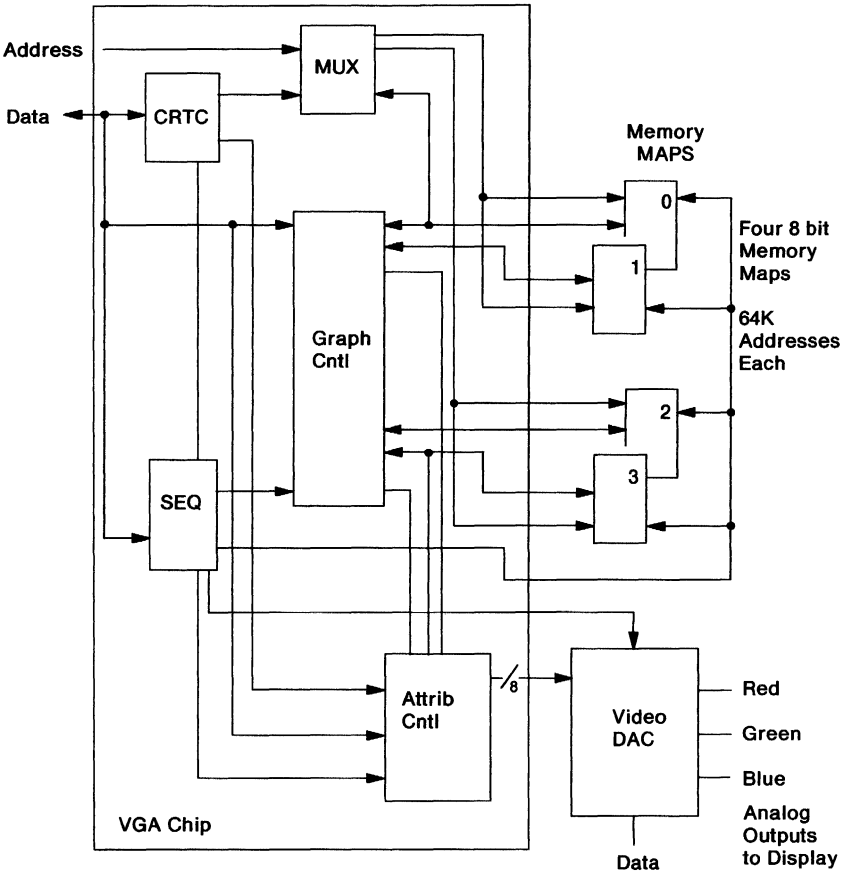


Figure 4-11. Video Subsystem Block Diagram

Major Components

Most of the logic for the VGA is contained in one module. This module contains all the circuits necessary to generate the timing for the video memory, and generates the video information that goes to the video digital-to-analog converter (DAC).

BIOS ROM

Software support is provided by video BIOS. Video BIOS is part of the system BIOS. BIOS is contained in a read-only memory (ROM) on the system board. This ROM BIOS contains the character generators and the control code to run the video subsystem.

Support Logic

Two clock sources (25.175 MHz and 28.322 MHz) provide the dot rate. The clock source is selected by setting a bit in a sequencer register. This is done by BIOS when a mode set is done.

The digital video output is sent to the digital-to-analog converter (DAC), which contains a color look-up table. Three analog signals (red, green, blue) are output from the DAC and are then sent to the display. Also see the attribute controller block diagram on page 4-26 and the auxiliary video connector block diagram on page 4-120. The 'sync' signals to the monitor are TTL levels. The analog video signals are 0 to 0.7 volts.

The maximum number of colors displayed is 16 out of 256K[1], except mode hex 13, which can display 256 out of 256K. The maximum number of shades of gray is 16 out of 64, except mode hex 13, which can display 64 out of 64 shades of gray.

Video Graphics Array Components

The Video Graphics Array has four major components. They are: the CRT Controller, the Sequencer, the Graphics Controller, and the Attribute Controller.

---

[1] K = 1024

CRT Controller

The Cathode Ray Tube Controller (CRTC) generates horizontal and vertical synchronous timings, addressing for the regenerative buffer, cursor and underline timings, and refresh addressing for the dynamic RAMs.

Sequencer

The Sequencer generates basic memory timings for the dynamic RAMs and the character clock for controlling regenerative memory fetches. It allows the system microprocessor to access memory during active display intervals by inserting dedicated system microprocessor memory cycles periodically between the display memory cycles. Map mask registers are available to protect entire memory maps from being changed.

Graphics Controller

The graphics controller is the interface between video memory and both the attribute controller during active display times and the system microprocessor during video memory reads or writes. During display times, memory data is latched and sent to the attribute controller. In All Points Addressable (APA) modes, the parallel memory data is converted to serial bit-plane data before being sent; in alphanumeric (A/N) modes, the parallel attribute data is sent directly. During a system microprocessor write or read to video memory, the graphics controller can perform logical operations on the memory data before it reaches video memory or the system microprocessor data bus, respectively. These logical operations are composed of four logical write modes and two logical read modes. These features allow enhanced operations such as a color compare read mode, individual bit masking during write modes, 32-bit writes in a single memory cycle, and writing to the display buffer on non-byte boundaries.

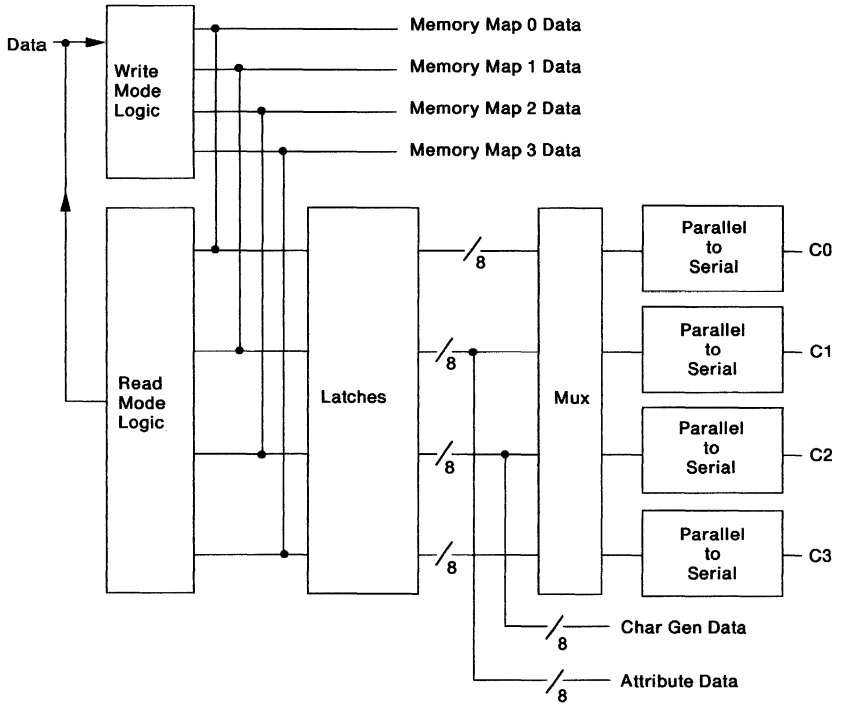


Figure 4-12. Graphics Controller Block Diagram

Attribute Controller

The attribute controller takes in data from video memory via the graphics controller and formats it for display on the display screen. Incoming attribute data in A/N mode, and serialized bit-plane data in APA mode, is converted to an 8-bit output digital color value. Each output color value is selected from an internal color palette of 64 possible colors (except in 256-color mode). The output color value is sent to the integrated DAC, where it is used as an address into an 18-bit color register whose value is in turn converted to three analog color signals that drive the display. Blinking, underlining, cursor insertion, and PEL panning are also controlled in the attribute controller.

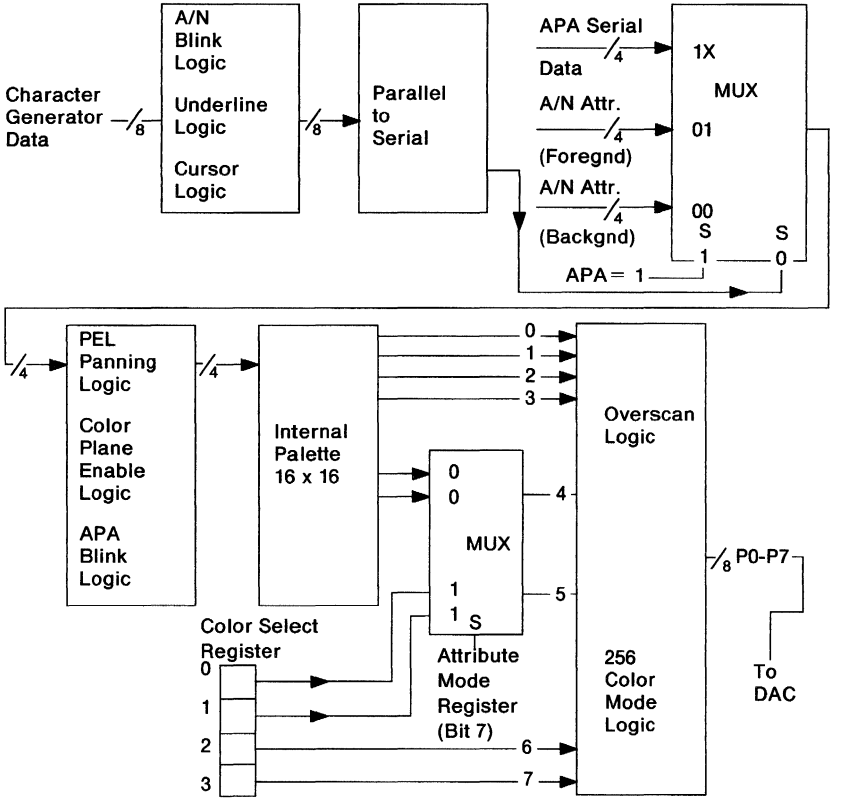


Figure 4-13. Attribute Controller Block Diagram

Modes of Operation

The following figure describes the modes supported by BIOS on IBM 31.5 kHz direct drive analog color and monochrome displays.

Mode (hex)	Type	Colors	Alpha Format	Buffer Start	Box Size	Max. Pgs.	Vert. Freq.	PELS
0, 1	A/N	16/256K	40 x 25	B8000	8 x 8	8	70 Hz	320 x 200
2, 3	A/N	16/256K	80 x 25	B8000	8 x 8	8	70 Hz	640 x 200
0*, 1*	A/N	16/256K	40 x 25	B8000	8 x 14	8	70 Hz	320 x 350
2*, 3*	A/N	16/256K	80 x 25	B8000	8 x 14	8	70 Hz	640 x 350
0+, 1+	A/N	16/256K	40 x 25	B8000	9 x 16	8	70 Hz	360 x 400
2+, 3+	A/N	16/256K	80 x 25	B8000	9 x 16	8	70 Hz	720 x 400
4, 5	APA	4/256K	40 x 25	B8000	8 x 8	1	70 Hz	320 x 200
6	APA	2/256K	80 x 25	B8000	8 x 8	1	70 Hz	640 x 200
7	A/N	-	80 x 25	B0000	9 x 14	8	70 Hz	720 x 350
7+	A/N	-	80 x 25	B0000	9 x 16	8	70 Hz	720 x 400
D	APA	16/256K	40 x 25	A0000	8 x 8	8	70 Hz	320 x 200
E	APA	16/256K	80 x 25	A0000	8 x 8	4	70 Hz	640 x 200
F	APA	-	80 x 25	A0000	8 x 14	2	70 Hz	640 x 350
10	APA	16/256K	80 x 25	A0000	8 x 14	2	70 Hz	640 x 350
11	APA	2/256K	80 x 30	A0000	8 x 16	1	60 Hz	640 x 480
12	APA	16/256K	80 x 30	A0000	8 x 16	1	60 Hz	640 x 480
13	APA	256/256K	40 x 25	A0000	8 x 8	1	70 Hz	320 x 200

* Enhanced modes from the IBM Enhanced Graphics Adapter.
\+ Enhanced modes

Figure 4-14. BIOS Video Modes

When the color display is used, each color is selected from a palette of 256K colors.

When the monochrome display is used, the colors are displayed as shades of gray. Each shade of gray is selected from a palette of 64.

Modes hex 0 through 6 emulate the support provided by the IBM Color/Graphics Monitor Adapter.

Modes hex 0, 2 and 4 are identical to modes hex 1,3 and 5 respectively. On the Color/Graphics Monitor Adapter there was a difference in these modes. On modes hex 0, 2, and 4, the color burst was turned off. Color burst is not provided by the video subsystem. Mode hex 3+ is the default mode with an analog color display attached to the system. Mode hex 7+ is the default mode with an analog monochrome display attached to the system.

Mode hex 7 emulates the support provided by the IBM Monochrome Display Adapter.

Modes hex 0*, 1*, 2*, 3*, D, E, F, and 10 emulate the support provided by the IBM Enhanced Graphics Adapter.

Double-scan means that each horizontal scan line is displayed twice, This is used for 200-line modes; they are displayed as 400 lines on the display. Border support is dependent on the BIOS mode selected. The following shows which BIOS modes support Double Scan and Border.

Mode (hex)	Double Scan	Border Support
0, 1	Yes	No
2, 3	Yes	Yes
0*, 1*	No	No
2*, 3*	No	Yes
0+, 1+	No	No
2+, 3+	No	Yes
4, 5	Yes	No
6	Yes	Yes
7	No	Yes
7+	No	Yes
D	Yes	No
E	Yes	Yes
F	No	Yes
10	No	Yes
11	No	Yes
12	No	Yes
13	Yes	Yes

* Enhanced modes from the IBM Enhanced Graphics Adapter.
\+ Enhanced modes

Figure 4-15. BIOS Double Scan and Border Support

Display Support

The video subsystem supports attachment of 31.5 kHz horizontal sweep frequency direct drive analog displays. These displays have a vertical sweep frequency capability of 50 to 70 cycles per second, providing extended color and sharpness and reduced flicker in most modes. Other IBM displays are *not* supported because they have digital interfaces, or have a different horizontal and vertical sweep frequency. The following figure summarizes the analog display characteristics.

Parameter	Color	Monochrome
Horizontal Scan Rate	31.5 kHz	31.5 kHz
Vertical Scan Rate	50 to 70 Hz	50 to 70 Hz
Video Bandwidth	28 MHz	28 MHz
Displayable Colors*	256/256K Maximum	64/64 Shades Gray
Maximum Horizontal Resolution	720 PELs	720 PELs
Maximum Vertical Resolution	480 PELs	480 PELs

* Controlled by Video Circuit

Figure 4-16. IBM 31.5 kHz Direct Drive Analog Displays

Since both color and monochrome displays run at the same sweep rate, all modes work on either type. The vertical gain of the display is controlled by the polarity of the vertical and horizontal sync pulses. This is done so that 350, 400, or 480 lines can be displayed without adjusting the display. See "15-Pin Display Connector Timing (Sync Signals)" on page 4-121 for more information.

Video Subsystem Programmable Option Select

The video subsystem supports Programmable Option Select (POS). When the POS sleep bit is set, the video subsystem does not respond to any memory or I/O reads or writes. Video is still generated if the video subsystem is programmed to do so. Programmable Option Select has to be enabled for video subsystem operation.

The implementation of Programmable Option Select (POS) for the video subsystem is described below:

- When in setup mode (I/O address hex 0094, bit 5 equals 0) the VGA responds to a single option select byte at I/O address hex 0102 and treats the LSB (bit 0) of that byte as the VGA sleep bit. When the LSB equals 0, the VGA does not respond to commands, addresses, or data, on the data bus. When the LSB equals 1, the VGA responds. If the VGA was set up and is generating video output when the LSB is set to 0, the output is still generated.
- The VGA responds only to address hex 0102 when in the setup mode. No other addresses are valid at that time. Conversely, the VGA ignores address hex 0102 when in the enabled mode (I/O address hex 0094, bit 5 equals 1), and decodes normal I/O and memory addresses.

Note: When VGA is disabled, accesses to the video DAC registers are disabled.

When the system is powered on, the power-on self-test (POST) initializes and enables the video subsystem.

For information on BIOS calls to enable or disable the VGA, see the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference*.

Alphanumeric Modes

This section describes the alphanumeric modes supported by the video subsystem and BIOS. Note that the colors in this section are generated when the BIOS is used to set the mode. BIOS initializes the video subsystem and the video DAC palette to generate these colors. If the video DAC palette is changed, different colors are generated.

The alphanumeric modes are modes hex 0 - 3 and 7. The mode chart lists the variations of these modes. The data format for alphanumeric modes is the same as the data format on the IBM Color/Graphics Monitor Adapter, the IBM Monochrome Display Adapter, and the IBM Enhanced Graphics Adapter. As in the EGA, bit 3 of the attribute byte may be redefined by the Character Map Select register to act as a switch between character sets. This gives the programmer access to 512 characters at one time.

When an alphanumeric mode is selected, the BIOS transfers character patterns from the ROM to map 2. The system microprocessor stores the character data in map 0, and the attribute data in map 1. The programmer can view map 0 and 1 as a single buffer in alphanumeric modes. The CRTC generates sequential addresses, and fetches one character code byte and one attribute byte at a time. The character code and row scan count are combined to address map 2, which contains the character generators. The appropriate dot patterns are then sent to the palette in the attribute section, where color is assigned according to the attribute data.

Every display-character position in the alphanumeric mode is defined by two bytes in the display buffer. Both the color/graphics and the monochrome emulation modes use the following 2-byte character/attribute format.

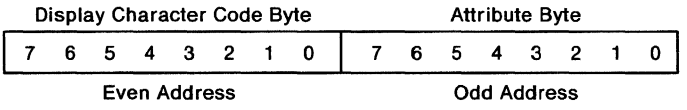


Figure 4-17. Character/Attribute Format

See Section 8, "Characters and Keystrokes" on page 8-1, for characters loaded during BIOS mode sets.

The functions of the attribute byte are defined in the following table:

Attribute Function	**Attribute Byte**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	B/I	R	G	B	I/CS	R	G	B
	Background				Foreground			
Normal (White on Black)	B/I	0	0	0	I/CS	1	1	1
Reverse (Black on White)	B/I	1	1	1	I/CS	0	0	0
Nondisplay (Black)	B/I	0	0	0	I/CS	0	0	0
Nondisplay (White)	B/I	1	1	1	I/CS	1	1	1
Mono = Underline / Color = Blue	B/I	0	0	0	I/CS	0	0	1

I = Highlighted
B = Blinking Foreground (Character)
CS = Character select

The BIOS defaults on a mode set are blink for bit 7 (B/I), and intensity for bit 3 (I/CS).

Figure 4-18. Attribute Byte Functions

The attribute byte definitions are:

Bit	Color	Function
7	B/I	Blinking/Background Intensity
6	R	Background Color
5	G	
4	B	
3	I/CS	Intensity/Character Select
2	R	Foreground Color
1	G	
0	B	

This figure shows the same information as the previous figure. It is included here for clarity.

Figure 4-19. Attribute Byte Definitions

For more information about the attribute byte, see "Character Map Select Register" on page 4-68 and "Attribute Mode Control Register" on page 4-102.

Any other code combination produces white-on-white in the monochrome emulation mode and the following colors in the color emulation mode:

I	R	G	B	Color
0	0	0	0	Black
0	0	0	1	Blue
0	0	1	0	Green
0	0	1	1	Cyan
0	1	0	0	Red
0	1	0	1	Magenta
0	1	1	0	Brown
0	1	1	1	White
1	0	0	0	Gray
1	0	0	1	Light Blue
1	0	1	0	Light Green
1	0	1	1	Light Cyan
1	1	0	0	Light Red
1	1	0	1	Light Magenta
1	1	1	0	Yellow
1	1	1	1	White (High Intensity)

Figure 4-20. Attribute Byte Colors

Both 40 column and 80 column alphanumeric modes are supported. The 40-column alphanumeric modes (all variations of modes hex 0 and 1) have the following features:

- Display up to 25 rows of 40 characters each
- Require 2,000 bytes of read/write memory per page
- One character and attribute for each character.

The 80-column alphanumeric modes (all variations of modes hex 2, 3, and 7) have the following features:

- Display up to 25 rows of 80 characters each
- Require 4,000 bytes of read/write memory per page
- One character and attribute for each character.

Graphics Modes

This section describes the graphics modes supported by the video subsystem and BIOS. Note that the colors in this section are generated when the BIOS is used to set the mode. BIOS initializes the video subsystem and the video DAC palette to generate these colors. If the video DAC palette is changed, different colors are generated.

320 x 200 Four-Color Graphics (Modes Hex 4 and 5)

Addressing, mapping and data format are the same as the 320 x 200 PEL mode of the IBM Color/Graphics Monitor Adapter. The display buffer is configured at hex B8000. Bit image data is stored in memory maps 0 and 1. The two bit planes (C0 and C1) are each formed from bits from both memory maps.

This mode has the following features:

- Contains a maximum of 200 rows of 320 PELs
- Selects one of four colors for each PEL
- Requires 16,000 bytes of read/write memory
- Uses memory-mapped graphics
- Double-scanned on display to 400 rows
- Formats four PELs-per-byte in the following manner:

Bit	Function
7	C1 - First Display PEL
6	C0 - First Display PEL
5	C1 - Second Display PEL
4	C0 - Second Display PEL
3	C1 - Third Display PEL
2	C0 - Third Display PEL
1	C1 - Fourth Display PEL
0	C0 - Fourth Display PEL

Figure 4-21. PEL Format, Modes hex 4 and 5

- Organizes graphics memory in two banks of 8,000 bytes, using the following format:

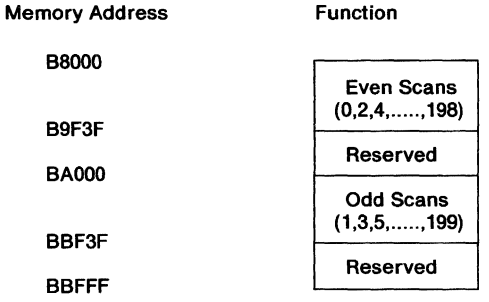


Figure 4-22. Video Memory Format

Address hex B8000 contains the PEL information for the upper-left corner of the display area.

- Color selection is determined by the following:

C1	C0	Color Selected	
0	0	Black	
0	1	Light Cyan	* Green
1	0	Light Magenta	* Red
1	1	Intensified White	* Brown

* Selectable by a video BIOS call

Figure 4-23. Color Selections, Modes Hex 4 and 5

640 x 200 Two Color Graphics (Mode Hex 6)

Addressing, mapping and data format are the same as the 640 x 200 PEL black and white mode of the IBM Color/Graphics Monitor Adapter. The display buffer is configured at hex B8000. Bit image data is stored in memory map 0 and comprises a single bit plane (C0).

This mode has the following features:

- Contains a maximum of 200 rows of 640 PELs.
- Supports two colors only.
- Requires 16,000 bytes of read/write memory.
- Addressing and mapping procedures are the same as 320 x 200 two- and four-color graphics, but the data format is different. In this mode, each bit in memory is mapped to a PEL on the screen.
- Double scanned on display to 400 rows.
- Formats eight PELs per byte in the following manner:

Bit	Function
7	First Display PEL
6	Second Display PEL
5	Third Display PEL
4	Fourth Display PEL
3	Fifth Display PEL
2	Sixth Display PEL
1	Seventh Display PEL
0	Eighth Display PEL

Figure 4-24. PEL Format, Mode hex 6

The bit definition for each PEL is (0 = Black) and (1 = Intensified White).

640 x 480 Two Color Graphics (Mode Hex 11)

This mode provides two color graphics with the same data format as mode 6. Addressing and mapping is shown under "Video Memory Organization" on page 4-40.

The bit image data is stored in map 0 and comprises a single bit plane (C0). A sequential buffer starting at address hex A0000 is provided. Location hex A0000 contains the byte with information for

the first eight PELs; location hex A0001 contains information for the second eight PELs, and so on. The bit definition for each PEL is (0 = Black) and (1 = Intensified White).

640 x 350 Graphics (Mode Hex F)

This mode emulates the EGA graphics on the IBM Monochrome Display with the following attributes: black, video, blinking video, and intensified video. Resolution of 640 x 350 requires 56K bytes to support four attributes. Maps 0 and 2 are used in this mode. Map 0 is the video bit plane, and map 2 is the intensity bit plane. Both planes reside at address hex A0000.

Two bits, one from each bit plane, define one picture element (PEL) on the screen. The bit definitions for the PELs are given in the following table. The video bit plane is denoted by C0 and the Intensity Bit Plane is denoted by C2.

C2	C0	PEL Color
0	0	Black
0	1	White
1	0	Blinking White
1	1	Intensified White

Figure 4-25. PEL Bit Definitions

The byte organization in memory is linear. The first eight PELs on the screen are defined by the contents of memory in location hex A0000, the second eight PELs by location hex A0001, and so on. The first PEL within any one byte is defined by bit 7 in the byte. The last PEL within the byte is defined by bit 0 in the byte.

Since both bit planes reside at address hex A0000, the user must select which plane or planes to update. This is accomplished by the Map Mask register of the sequencer. For more information, see "Video Memory Organization" on page 4-40.

16 Color Graphics Modes (Mode Hex 10, D, E, and 12)

These modes support graphics in 16 colors. The bit image data is stored in all four memory maps in these modes. Each memory map contains the data for one bit plane. Each bit plane represents a color as shown below. The bit planes are denoted as C0, C1, C2 and C3 respectively.

C0 = Blue PELs
C1 = Green PELs
C2 = Red PELs
C3 = Intensified PELs

Four bits (one from each plane) define one PEL on the screen. The color combinations are illustrated in the following figure:

C3	C2	C1	C0	Color
0	0	0	0	Black
0	0	0	1	Blue
0	0	1	0	Green
0	0	1	1	Cyan
0	1	0	0	Red
0	1	0	1	Magenta
0	1	1	0	Brown
0	1	1	1	White
1	0	0	0	Dark Gray
1	0	0	1	Light Blue
1	0	1	0	Light Green
1	0	1	1	Light Cyan
1	1	0	0	Light Red
1	1	0	1	Light Magenta
1	1	1	0	Yellow
1	1	1	1	Intensified White

Figure 4-26. Palette Colors

The display buffer resides at address hex A0000. The Map Mask register is used to select any or all of the maps to be updated when a memory write to the display buffer is executed by the system microprocessor.

256 Color Graphics Mode (Mode Hex 13)

This mode provides graphics with the capability to display 256 colors on the screen at one time.

The display buffer is linear, starting at address hex A0000, and 64,000 bytes long. The first location contains the 8-bit color for the upper left hand PEL. The second byte contains the second PEL and so on for 64,000 PELs (320 x 200). The bit image data is stored in all four memory maps and comprises four bit planes. The four bit planes are sampled twice to produce eight bit planes that address the video DAC.

The internal palette of the video subsystem is not used to select colors. It is set by BIOS and should not be changed. The external palette in the video DAC is programmed by the BIOS such that the first 16 locations contain colors that are compatible with colors of other modes. See Figure 4-27 on page 4-40. The second 16 registers contain 16 evenly spaced gray shades. The remaining 216 locations are loaded based on a hue-saturation-intensity model tuned to provide a usable, generic color set that covers a wide range of color values.

The following figure shows the colors that are compatible with colors of other modes.

Attribute Byte								Analog Output Color
C7	C6	C5	C4	C3	C2	C1	C0	
0	0	0	0	0	0	0	0	Black
0	0	0	0	0	0	0	1	Blue
0	0	0	0	0	0	1	0	Green
0	0	0	0	0	0	1	1	Cyan
0	0	0	0	0	1	0	0	Red
0	0	0	0	0	1	0	1	Magenta
0	0	0	0	0	1	1	0	Brown
0	0	0	0	0	1	1	1	White
0	0	0	0	1	0	0	0	Dark Gray
0	0	0	0	1	0	0	1	Light Blue
0	0	0	0	1	0	1	0	Light Green
0	0	0	0	1	0	1	1	Light Cyan
0	0	0	0	1	1	0	0	Light Red
0	0	0	0	1	1	0	1	Light Magenta
0	0	0	0	1	1	1	0	Yellow
0	0	0	0	1	1	1	1	Intensified White

Figure 4-27. Attribute Byte

The video DAC palette can be programmed with 256K different colors.

This mode has the following features:

- Contains a maximum of 200 rows of 320 PELs
- Double scanned on display to 400 rows
- Selects one of 256K colors for each PEL
- Requires 64,000 bytes of read/write memory
- Uses memory-mapped graphics
- Uses one byte of memory for each PEL.

Video Memory Organization

The video display buffer on the system board consists of 256K bytes of dynamic read/write memory configured as four 64K-byte video maps.

The address of the display buffer can be changed to remain compatible with other video adapters and application software. Four locations are provided. The buffer can be configured at segment address hex A0000 for a length of 128K bytes, at hex A0000 for a length of 64K bytes, at hex B0000 for a length of 32K bytes, or at hex B8000 for a length of 32K bytes.

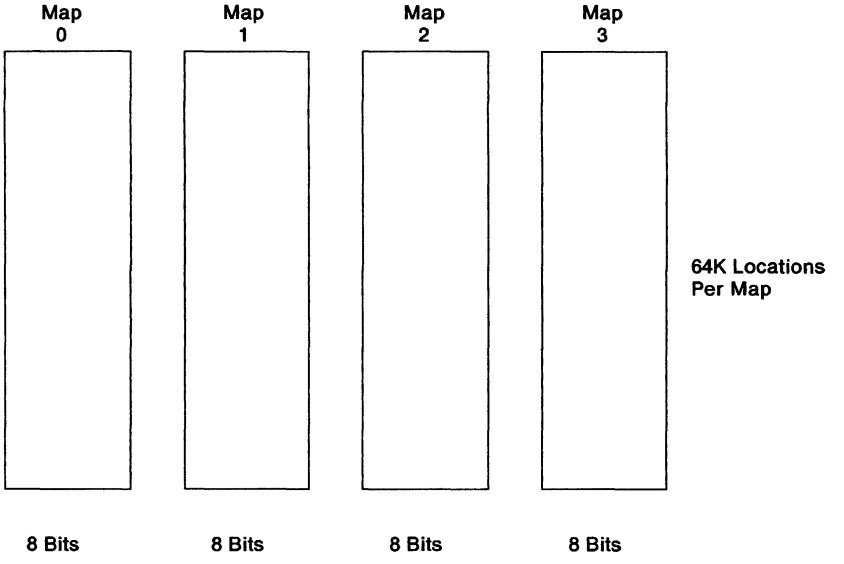


Figure 4-28. 256K Video Memory Map

Maps 0 through 3 usually form bit planes 0 through 3.

Map 0 = Bit Plane 0

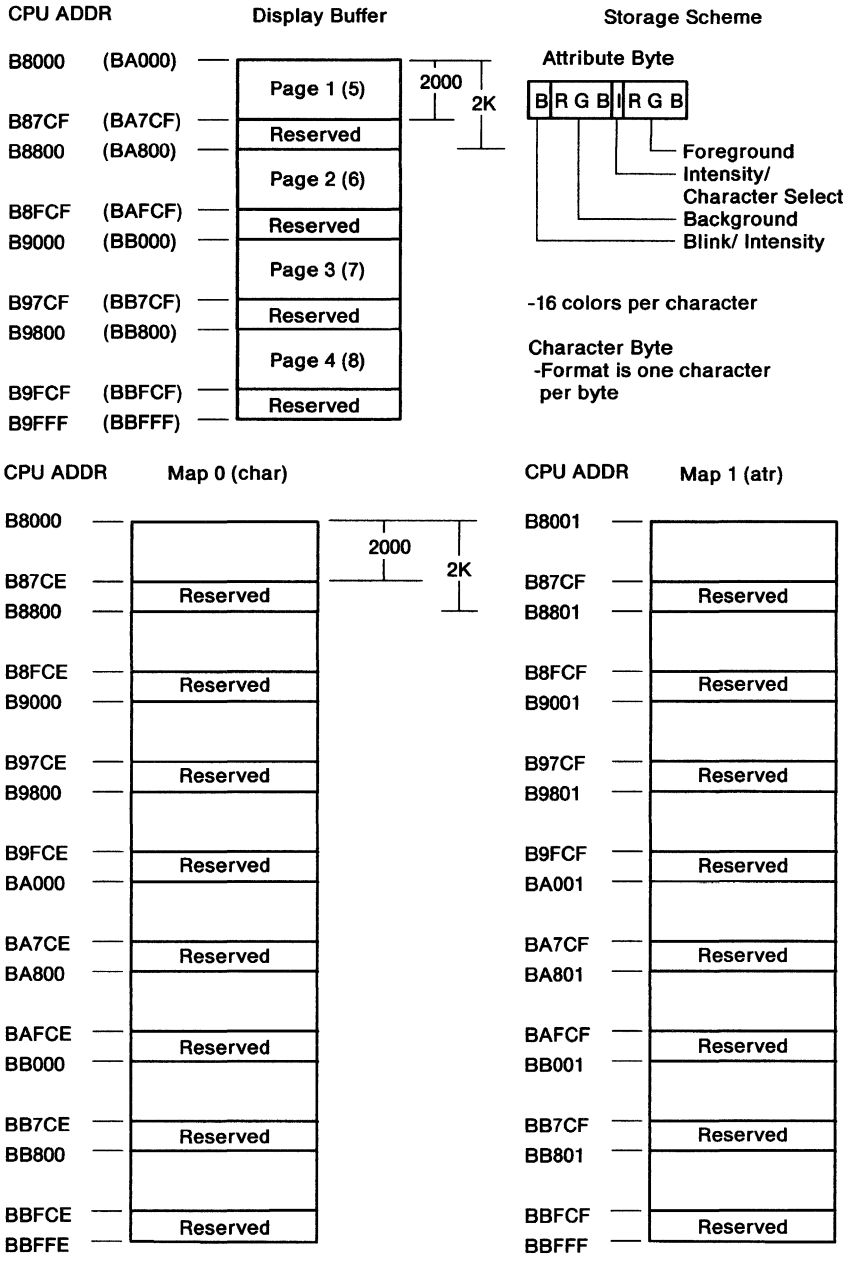
Map 1 = Bit Plane 1

Map 2 = Bit Plane 2

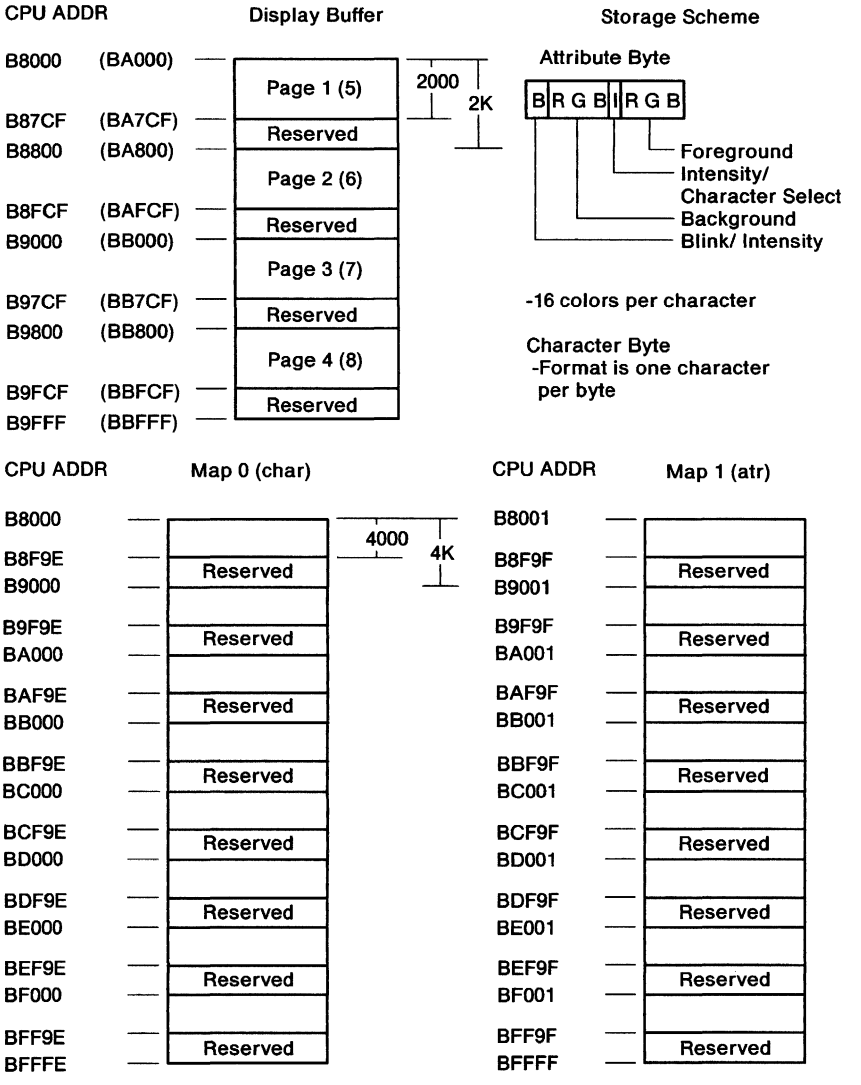
Map 3 = Bit Plane 3

In mode hex 13, each of the four bit planes is formed with data from all four maps. The four bits are sampled twice internally to produce the eight bit values needed to select 256 colors.

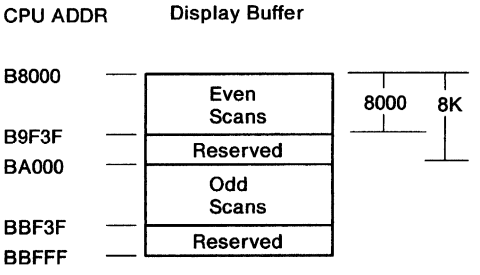
Modes hex 0, 1 (All Variations of Modes Hex 0 and 1)

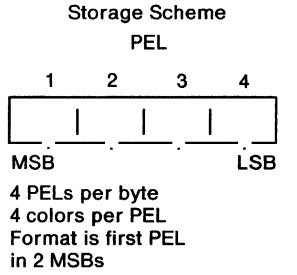


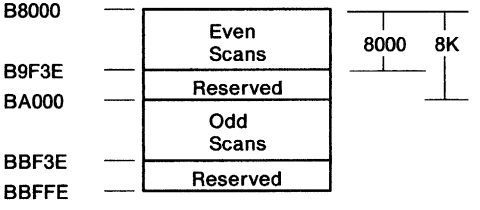
Modes hex 2, 3 (All Variations of Modes Hex 2 and 3)

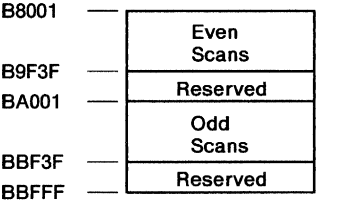


Modes hex 4, 5

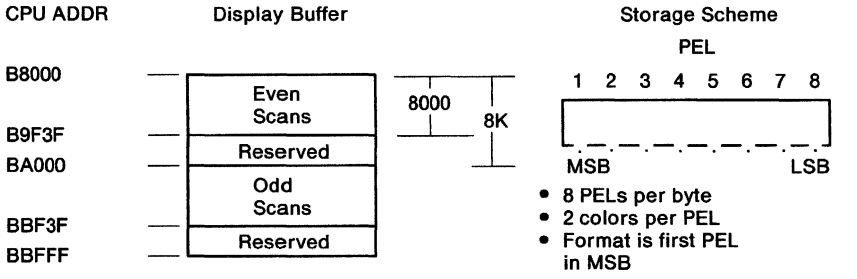

CPU ADDR
Display Buffer
B8000
B9F3F
BA000
BBF3F
BBFFF
Even Scans
Reserved
Odd Scans
Reserved
8000
8K

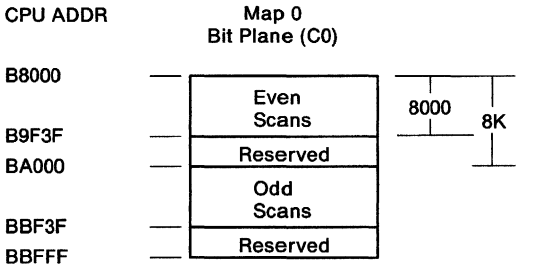

Storage Scheme
PEL
1
2
3
4
MSB
LSB
• 4 PELs per byte
• 4 colors per PEL
• Format is first PEL in 2 MSBs


CPU ADDR
Map 0
B8000
B9F3E
BA000
BBF3E
BBFFE
Even Scans
Reserved
Odd Scans
Reserved
8000
8K

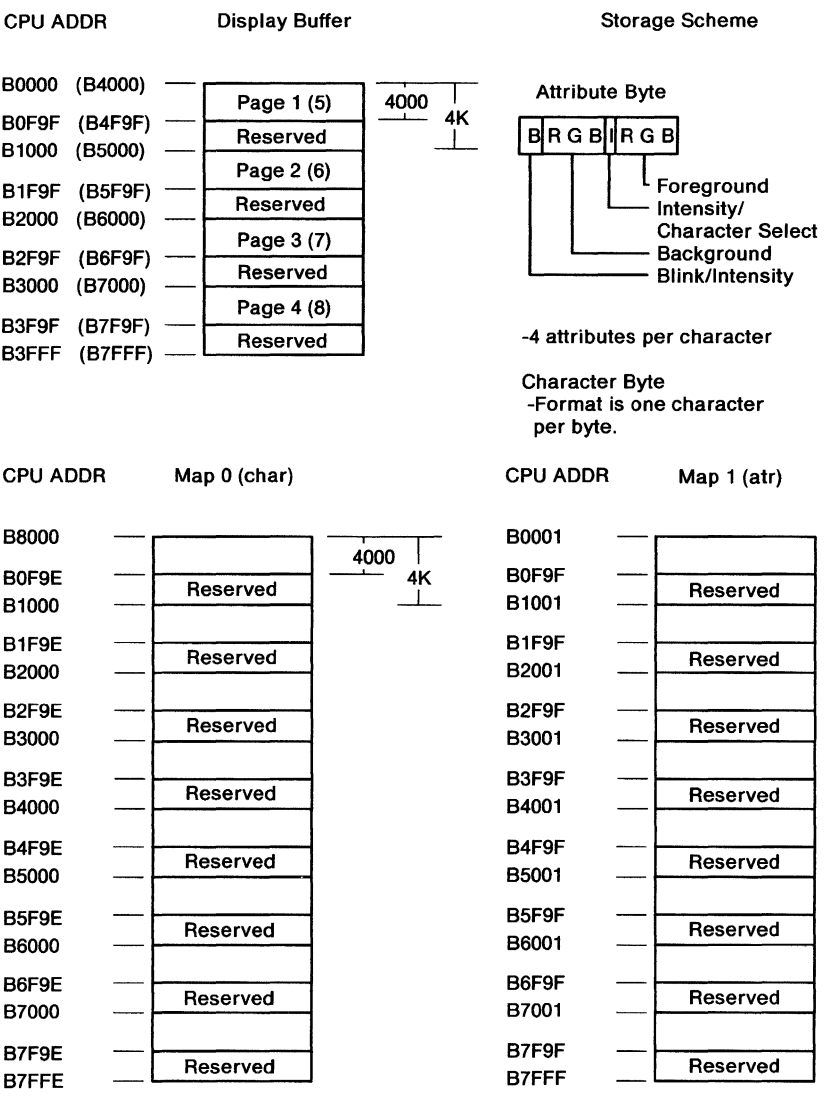

CPU ADDR
Map 1
B8001
B9F3F
BA001
BBF3F
BBFFF
Even Scans
Reserved
Odd Scans
Reserved

Mode Hex 6

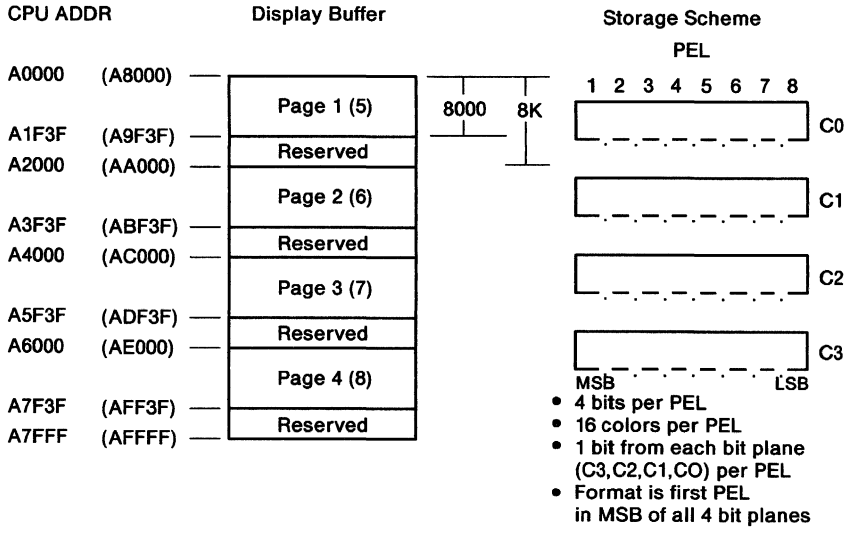

CPU ADDR
Display Buffer
Storage Scheme
PEL
1 2 3 4 5 6 7 8
B8000
B9F3F
BA000
BBF3F
BBFFF
Even Scans
Reserved
Odd Scans
Reserved
8000
8K
MSB
LSB
• 8 PELs per byte
• 2 colors per PEL
• Format is first PEL in MSB

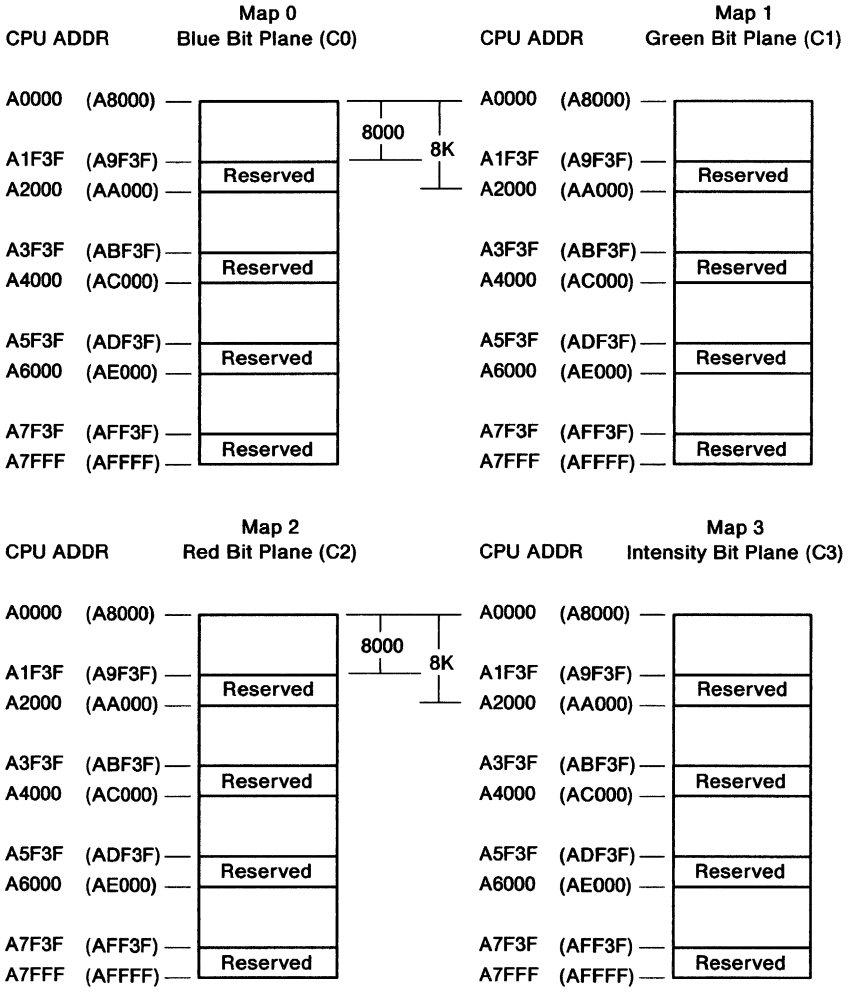

CPU ADDR
Map 0
Bit Plane (C0)
B8000
B9F3F
BA000
BBF3F
BBFFF
Even Scans
Reserved
Odd Scans
Reserved
8000
8K

Mode Hex 7 (All Variations of Mode Hex 7)

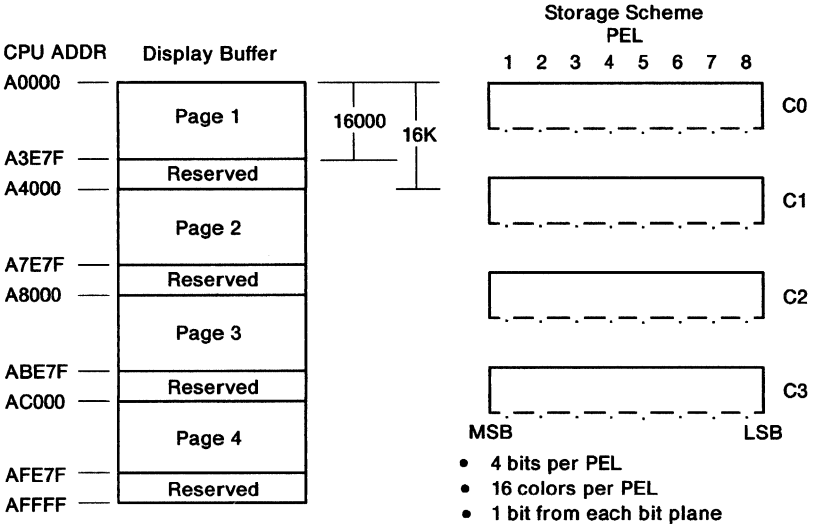


Mode Hex D

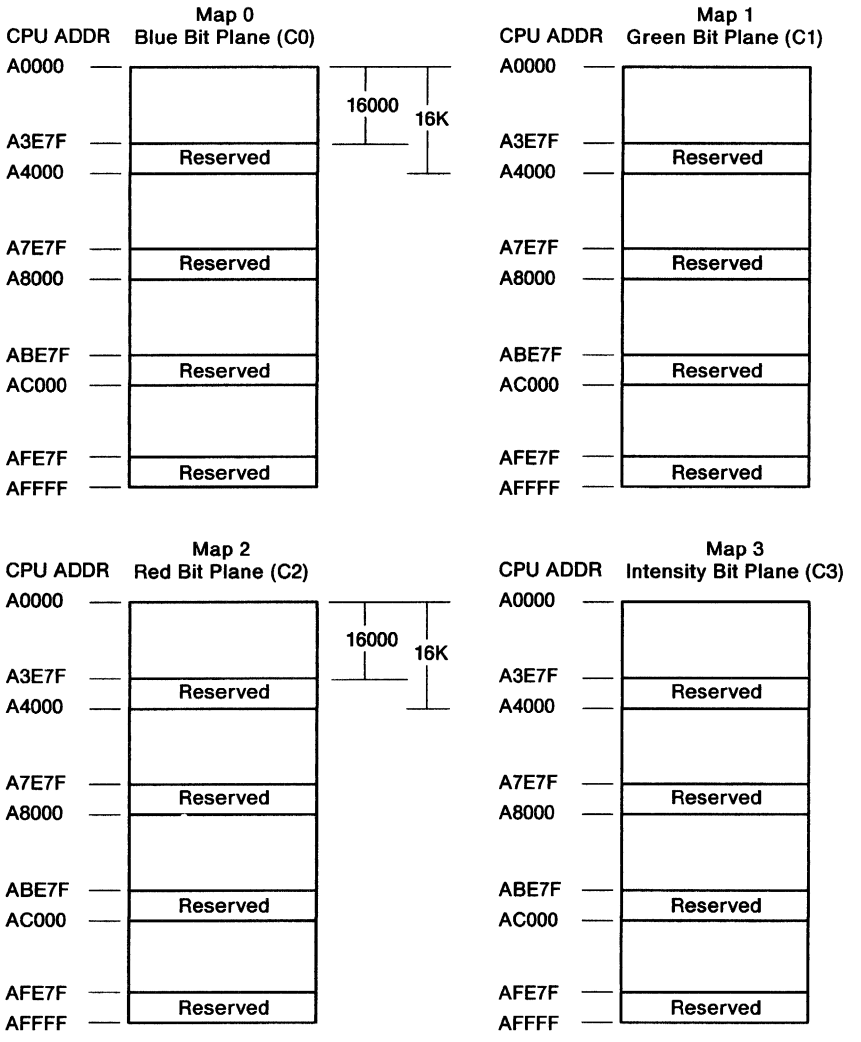

CPU ADDR
Display Buffer
Storage Scheme
PEL
A0000 (A8000)
A1F3F (A9F3F)
A2000 (AA000)
A3F3F (ABF3F)
A4000 (AC000)
A5F3F (ADF3F)
A6000 (AE000)
A7F3F (AFF3F)
A7FFF (AFFFF)
Page 1 (5)
Reserved
Page 2 (6)
Reserved
Page 3 (7)
Reserved
Page 4 (8)
Reserved
8000
8K
1 2 3 4 5 6 7 8
C0
C1
C2
C3
MSB
LSB
• 4 bits per PEL
• 16 colors per PEL
• 1 bit from each bit plane (C3,C2,C1,CO) per PEL
• Format is first PEL in MSB of all 4 bit planes


Map 0
CPU ADDR
Blue Bit Plane (C0)
Map 1
CPU ADDR
Green Bit Plane (C1)
Map 2
CPU ADDR
Red Bit Plane (C2)
Map 3
CPU ADDR
Intensity Bit Plane (C3)
A0000 (A8000)
A1F3F (A9F3F)
A2000 (AA000)
A3F3F (ABF3F)
A4000 (AC000)
A5F3F (ADF3F)
A6000 (AE000)
A7F3F (AFF3F)
A7FFF (AFFFF)
Reserved
8000
8K

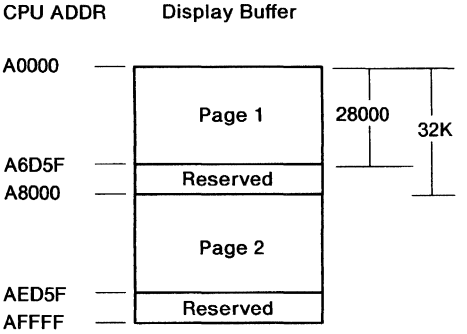
Mode Hex E

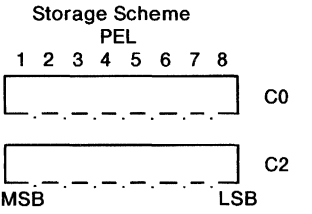


- 4 bits per PEL
- 16 colors per PEL
- 1 bit from each bit plane (C3,C2,C1,C0) per PEL
- Format is first PEL in MSB of all 4 bit planes

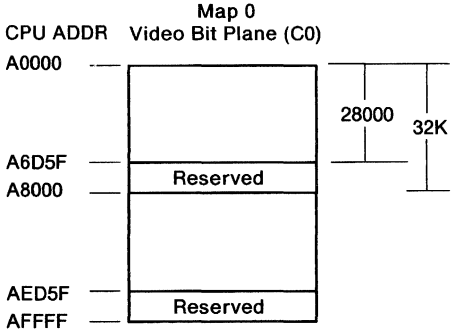

Map 0
CPU ADDR
Blue Bit Plane (C0)
A0000
A3E7F
A4000
A7E7F
A8000
ABE7F
AC000
AFE7F
AFFFF
Reserved
16000
16K
Map 1
CPU ADDR
Green Bit Plane (C1)
Map 2
CPU ADDR
Red Bit Plane (C2)
Map 3
CPU ADDR
Intensity Bit Plane (C3)

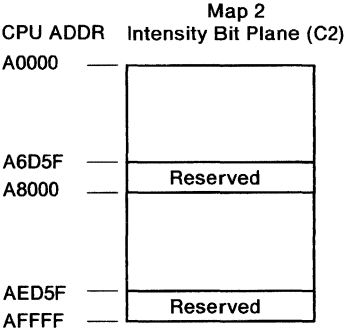
Mode Hex F



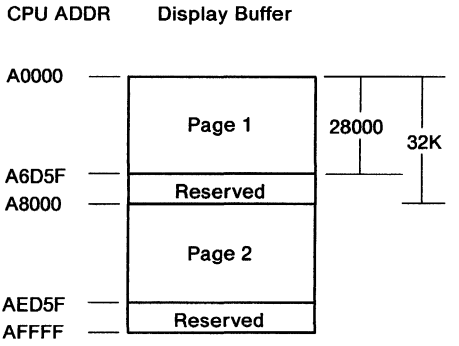


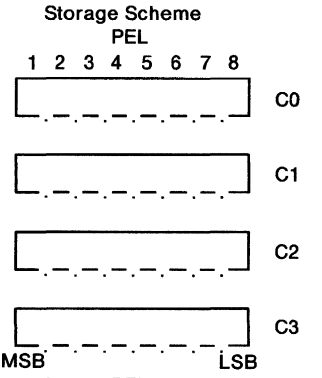
- 2 bits per PEL
- 4 attributes per PEL
- 1 bit from each bit plane (C2,C0)
- Format is first PEL is MSB of video and intensity bit planes



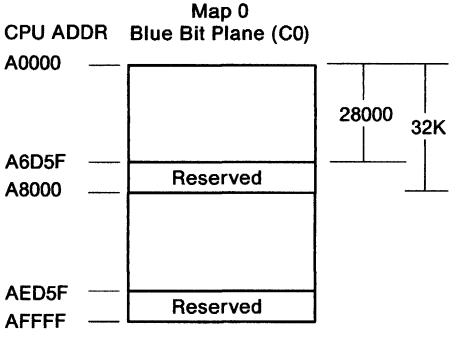


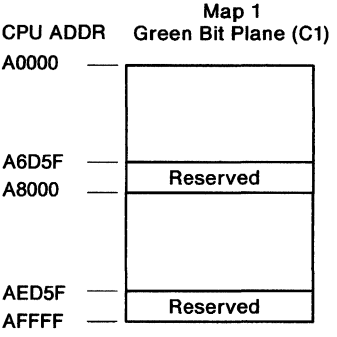
Mode Hex 10

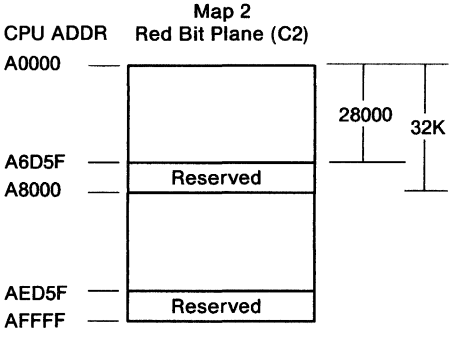


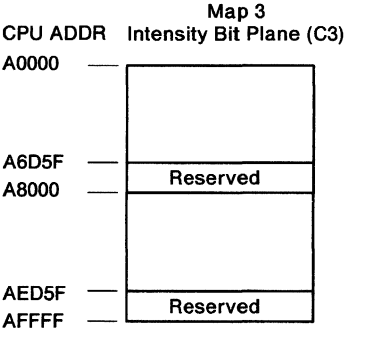


- 4 bits per PEL
- 16 colors per PEL
- 1 bit from each bit plane (C3,C2,C1,C0) per PEL
- Format is first PEL in MSB of all 4 bit planes

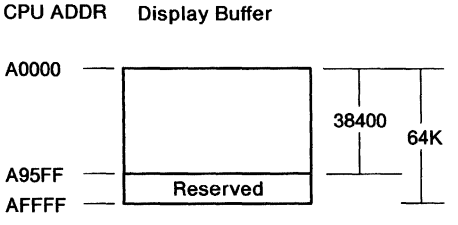


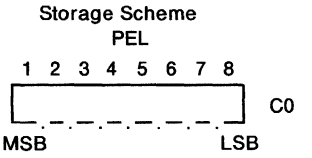




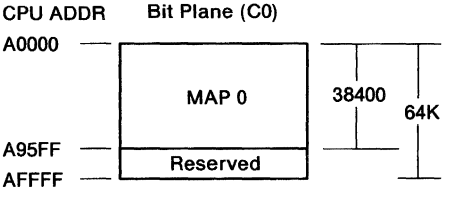


Mode Hex 11

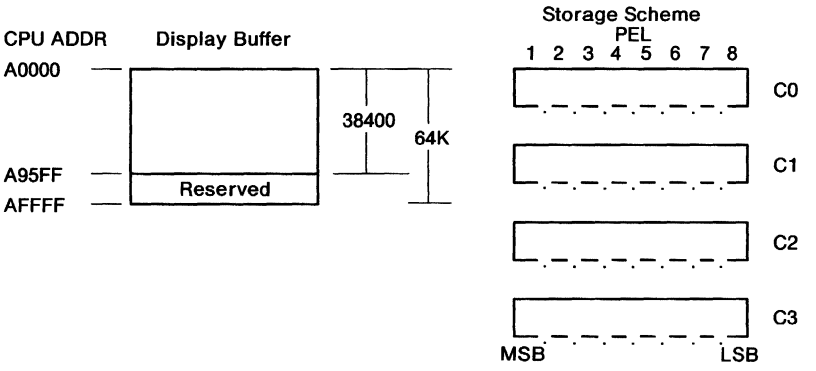




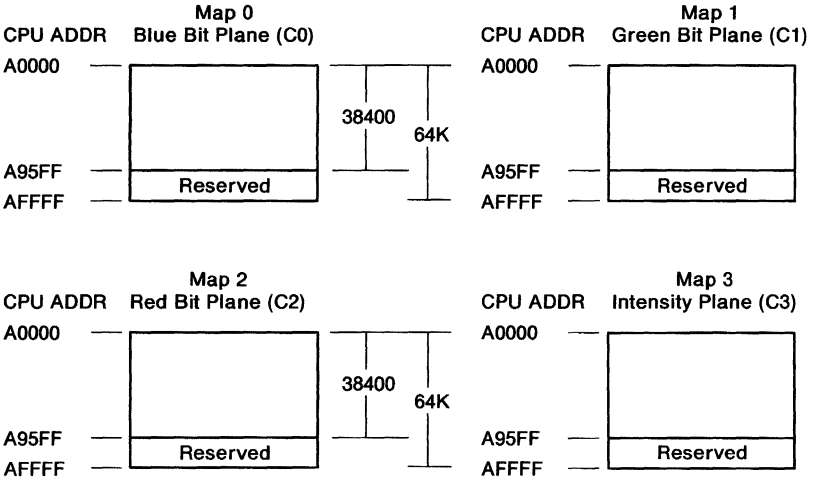
- 1 bit per PEL
- 2 attributes per PEL
- Format is first PEL in MSB position



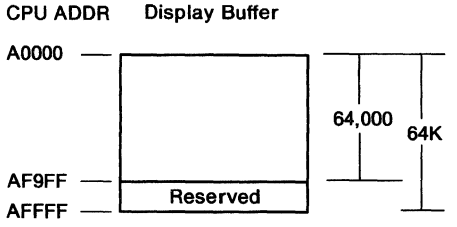
Mode Hex 12



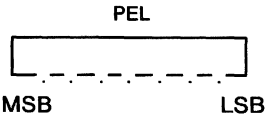
- 4 bits per PEL
- 16 colors per PEL
- 1 bit from each bit plane (C3,C2,C1,C0) per PEL
- Format is first PEL in MSB of all 4 bit planes



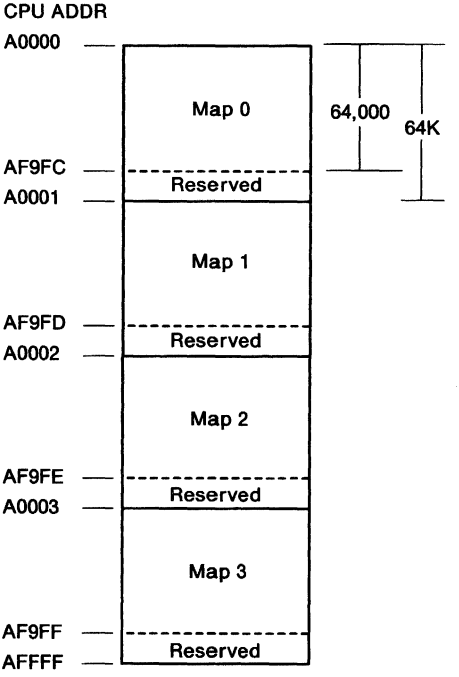
Mode Hex 13



Storage Scheme



- 8 bits per PEL
- 256 colors per PEL
- 1 PEL per byte
- Format is PEL 1 at address A0000



Video Memory Read/Write Operations

Video Memory Write Operations

During system microprocessor writes to the video memory, the maps are enabled by the logical decode of the memory address and, depending on the video mode, the Map Mask register. The data flow for a system microprocessor write operation is illustrated in the following figure.

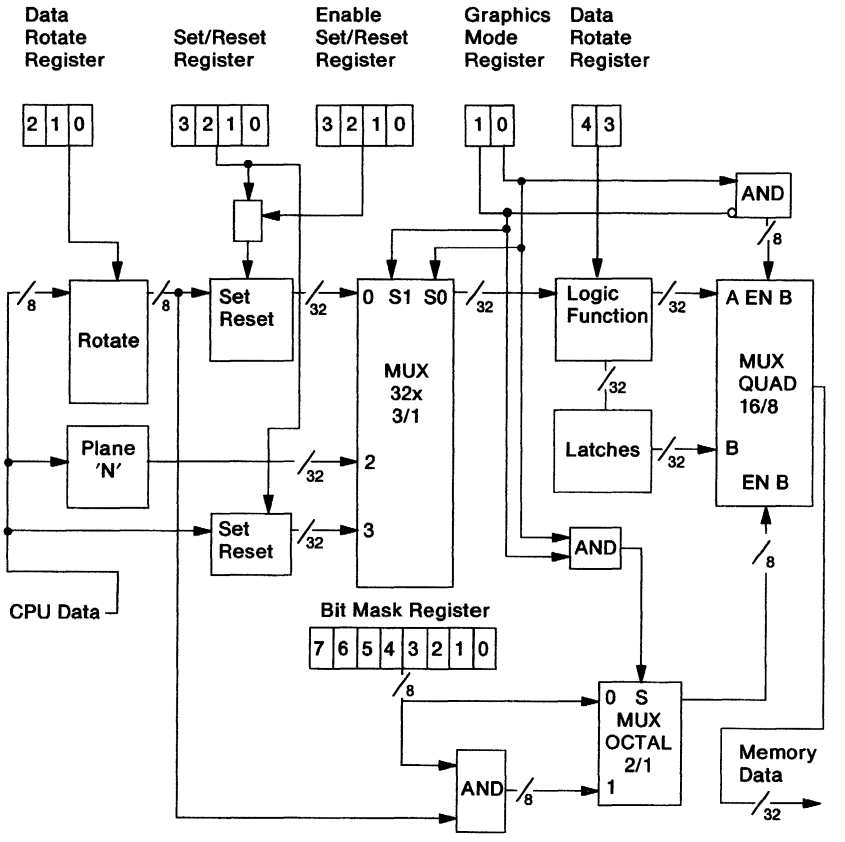


Figure 4-29. Data Flow for VGA Memory Write Operations

Note: Which maps are actually written with data is under control of the system microprocessor memory address, which depends on the mode selected and the Map Mask register.

Video Memory Read Operations

There are two ways to do video memory Reads. When Read type 0 is selected using the Graphics Mode register, the system microprocessor Reads to the video memory return the 8-bit value that is determined by the logical decode of the memory address, and the Read Map Select register if applicable. When Read type 1 is selected using the Graphics Mode register, the 8-bit value returned is the result of the color compare operation controlled by the Color Compare and Color Don't Care registers. The data flow for the color compare operations is illustrated in the following figure.

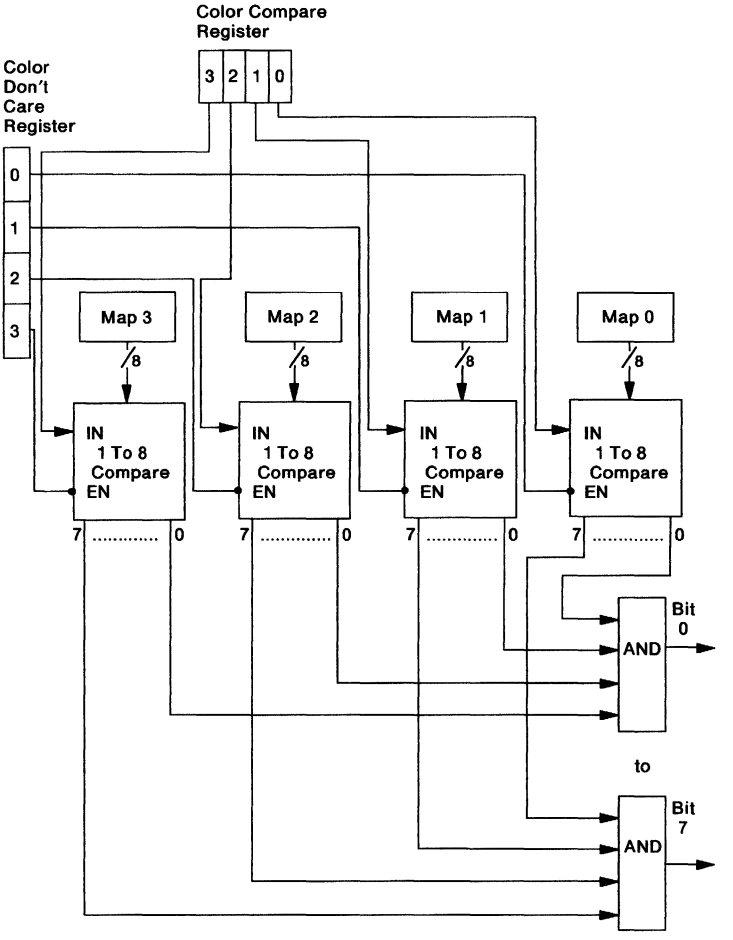


Figure 4-30. VGA Color Compare Operations

Registers

There are six sets of registers in the video subsystem. All but the system microprocessor data latches and the attribute address flip-flop are readable. The following figure lists the registers and the I/O address where they are located. The figure also lists whether or not they are read/write, read-only, or write-only.

	R/W	Monochrome Emulation	Color Emulation	Either
General Registers	**Addresses used 03BA or 03DA; 03C2; 03CA; and 03CC**			
Miscellaneous Output Reg	W			03C2
	R			03CC
Input Status Register 0	RO			03C2
Input Status Register 1	RO	03BA	03DA	
Feature Control Register	W	03BA	03DA	
	R			03CA
Video Subsystem Enable	RW			03C3
Attribute Registers	**Addresses used 03C0 - 03C1**			
Address Register	RW			03C0
Other Attribute Registers	W			03C0
(03C0 write, 03C1 read)	R			03C1
CRT Controller Registers	**Addresses used 03D4 to 03D5 or 03B4 to 03B5**			
Index Register	RW	03B4	03D4	
Other CRT Controller Regs.	RW	03B5	03D5	
Sequencer Registers	**Addresses used 03C4 to 03C5**			
Address Register	RW			03C4
Other Sequencer Registers	RW			03C5
Graphics Registers	**Addresses used 03CE to 03CF**			
Address Register	RW			03CE
Other Graphics Registers	RW			03CF

RO = Read Only, RW = Read / Write
Register addresses are in hex.

Figure 4-31. VGA Register Overview

The VGA also provides the system microprocessor interface for the video DAC. The DAC has one address register. It is written to address hex 03C7 or 03C8, and read from address hex 03C8.

PEL Address (Write Mode)	RW	03C8
PEL Address (Read Mode)	WO	03C7
DAC State Register	RO	03C7
PEL Data Register	RW	03C9
PEL Mask *	RW	03C6

RO = Read-Only, RW = Read/Write, WO = Write-Only.
Register addresses are in hex.

* This register must not be written to by application code or destruction of the color look-up table may occur. See also Video Digital-to-Analog Converter on page 4-115 for programming considerations.

Figure 4-32. DAC Register Usage

General Registers

This section contains descriptions of the following registers.

Name	Read Port	Write Port	Index
Miscellaneous Output Register	03CC	03C2	-
Input Status Register 0	03C2	-	-
Input Status Register 1	03?A	-	-
Feature Control Register	03CA	03?A	-
VGA Enable Register	03C3	03C3	-
DAC State Register	03C7	-	-

The (?) is controlled by Bit 0 of the Miscellaneous Output register.
? = B in Monochrome Emulation Modes and
? = D in Color Emulation Modes
Register addresses are in hex.

Figure 4-33. General Register Overview

Miscellaneous Output Register

This is a read/write register. A hardware reset causes all bits to reset to 0.

Read address = hex 03CC; Write address = hex 03C2.

Bit	Function
7	Vertical Sync Polarity
6	Horizontal Sync Polarity
5	Page Bit For Odd/Even (diagnostic use)
4	Reserved = 0
3	Clock Select 1
2	Clock Select 0
1	Enable RAM
0	I/O Address Select

Figure 4-34. Miscellaneous Output Register

Bit 7 Vertical Sync Polarity - A logical 0 selects positive vertical retrace; a logical 1 selects negative vertical retrace.

Bit 6 Horizontal Sync Polarity - A logical 0 selects positive horizontal retrace; a logical 1 selects negative horizontal retrace.

Note: Bits 7 and 6 are selected based on the vertical size as shown in the following figure.

Bit 7	Bit 6	Vertical Size
0	0	Reserved = 0
1	0	400 lines
0	1	350 lines
1	1	480 lines

Figure 4-35. Display, Vertical Size

Bit 5 Page Bit For Odd/Even - Selects between two 64K pages of memory when in the Odd/Even modes (0-5). A logical 0 selects the low page of memory; a logical 1 selects the high page of memory. This bit is provided for diagnostic use.

Bit 4 Reserved

Bits 3, 2 Clock Select - These two bits select the clock source according to the following table:

Bit 3	Bit 2	Function
0	0	Selects 25.175 MHz clock for 640 Horizontal PELs
0	1	Selects 28.322 MHz clock for 720 Horizontal PELs
1	0	Selects external clock input from the auxiliary video connector for the input clock. This input clock should be kept between 14.3 MHz and 28.4 MHz.
1	1	Reserved

Figure 4-36. Clock Select 3 and 2 Bit Definitions

Bit 1 Enable RAM - A logical 0 disables Video RAM address decode from the system microprocessor; a logical 1 enables Video RAM to the system microprocessor.

Bit 0 I/O Address Select - CRTC I/O address hex 03BX/03DX . This bit maps the CRTC I/O addresses for IBM Monochrome or Color/Graphics Monitor Adapter emulation. A logical 0 sets CRTC addresses to hex 03BX and Input Status register 1's address to hex 03BA for Monochrome emulation. A logical 1 sets CRTC addresses to hex 03DX and Input Status register 1's address to hex 03DA for IBM Color/Graphics Monitor Adapter emulation.

Input Status Register 0

This is a read-only register.

Read address = hex 03C2

Bit	Function
7	CRT Interrupt
6, 5	Reserved = 0
4	Switch Sense Bit
3 - 0	Reserved = 0

Figure 4-37. Input Status Register 0

Bit 7 CRT Interrupt - A logical 1 indicates a vertical retrace interrupt is pending; a logical 0 indicates the vertical retrace interrupt is cleared.

Bits 6, 5 Reserved

Bit 4 Switch Sense Bit - This bit allows the system microprocessor to read the switch sense line. This bit allows the power-on self-test to determine if a monochrome or color display is connected to the system.

Bits 3 - 0 Reserved

Input Status Register 1

This is a read-only register.

Read address = hex 03?A

Bit	Function
7, 6	Reserved = 0
5	Diagnostic 0
4	Diagnostic 1
3	Vertical Retrace
2, 1	Reserved = 0
0	Display Enable

Figure 4-38. Input Status Register 1

Bits 7, 6 Reserved

Bits 5, 4 Diagnostic Usage - These bits are selectively connected to two of the eight color outputs of the Attribute Controller. The Color Plane Enable register controls the multiplexer for the video wiring. The following figure illustrates the combinations available and the color output wiring.

Color Plane Register		Input Status Register 1	
Bit 5	Bit 4	Bit 5	Bit 4
0	0	P2	P0
0	1	P5	P4
1	0	P3	P1
1	1	P7	P6

Figure 4-39. Diagnostic Bits

Bit 3 Vertical Retrace - A logical 0 indicates that video information is being displayed; a logical 1 indicates a vertical retrace interval. This bit can be programmed to interrupt the system microprocessor on interrupt level 2 at the start of the vertical retrace. This is done through bits 5 and 4 of the Vertical Retrace End register.

Bits 2, 1 Reserved

Bit 0 Display Enable - Logical 1 indicates a horizontal or vertical retrace interval. This bit is the real-time status of the inverted display enable signal. To avoid glitches on the display, some programs use this status bit to restrict screen updates to inactive display intervals. The video subsystem has been designed to eliminate this software requirement, so display screen updates may be made at any time.

Feature Control Register

Write address = 03?A; Read address = 03CA.

All bits in this register are reserved, bit 3 must equal 0.

Video Subsystem Enable Register

Bit	Function
7 - 1	Reserved = 0
0	Video Subsystem Enable

Figure 4-40. Video Subsystem Enable Register, Address hex 03C3)

Bits 7 - 1 Reserved

Bit 0 When this bit is set to a 1, the video I/O and memory address decoding is enabled. A 0 disables the video I/O and memory address decoding.

Note: Accesses to the Video Subsystem Enable register are not affected by the VGA sleep bit (I/O port 102, bit 0), described in "Video Subsystem Programmable Option Select" on page 4-29.

Sequencer Registers

Name	Port (hex)	Index (hex)
Sequencer Address	03C4	-
Reset	03C5	00
Clocking Mode	03C5	01
Map Mask	03C5	02
Character Map Select	03C5	03
Memory Mode	03C5	04

Figure 4-41. Sequencer Register Overview

Sequencer Address Register

The Sequencer Address register is a pointer register located at address hex 03C4. This register is loaded with a binary value that points to the Sequencer Data register where data is to be written. This value is referred to as "Index" in the figure above.

Bit	Function
7 - 3	Reserved = 0
2 - 0	Sequencer Address

Figure 4-42. Sequencer Address Register

Bits 7 - 3 Reserved

Bits 2 - 0 Sequencer Address Bits - A binary value pointing to the register where data is to be written.

Reset Register

This is a read/write register pointed to when the value in the Address register is hex 00. The port address for this register is hex 03C5.

Bit	Function
7 - 2	Reserved = 0
1	Synchronous Reset
0	Asynchronous Reset

Figure 4-43. Reset Register, Index Hex 00

Bits 7 - 2 Reserved

Bit 1 Synchronous Reset - A logical 0 commands the sequencer to synchronously clear and halt. Bits 1 and 0 must both be 1 to allow the sequencer to operate. This bit must be set to 0 before changing either bit 3 or bit 0 of the Clocking Mode register, or bit 3 or bit 2 of the Miscellaneous Output register.

Bit 0 Asynchronous Reset - A logical 0 commands the sequencer to asynchronously clear and halt. A logical 1 commands the sequencer to run unless bit 1 is set to 0. Resetting the sequencer with this bit can cause data loss in the dynamic RAMs.

Clocking Mode Register

This is a read/write register pointed to when the value in the Address register is hex 01. The port address for this register is hex 03C5.

Bit	Function
7, 6	Reserved = 0
5	Screen Off
4	Shift 4
3	Dot Clock
2	Shift Load
1	Reserved = 0
0	8/9 Dot Clocks

Figure 4-44. Clocking Mode Register, Index Hex 01

Bits 7, 6 Reserved

Bit 5 Screen Off - A logical 1 turns off the video screen and assigns maximum memory bandwidth to the system microprocessor. A logical 0 selects normal screen operation. The screen is blanked when this bit is set, and the sync pulses are maintained. Use this bit for rapid full-screen updates.

Bit 4 Shift 4 - When set to 0, the video serializers are loaded every character clock; when set to 1, the serializers are loaded every fourth character clock. This mode is useful when 32 bits are fetched per cycle and chained together in the shift registers.

Bit 3 Dot Clock - A logical 0 selects normal dot clocks derived from the sequencer master clock input. When this bit is set to 1, the master clock will be divided by 2 to generate the dot clock. All the other timings are affected because they are derived from the dot clock. Dot clock divided by 2 is used for 320 and 360 horizontal PEL modes.

Bit 2 Shift Load - When set to 0, and Bit 4 is set to 0, the video serializers are reloaded every character clock; when set to 1, the video serializers are loaded every other character clock. This mode is useful when 16 bits are fetched per cycle and chained together in the shift registers.

Bit 1 Reserved

Bit 0 8/9 Dot Clocks - A logical 0 directs the sequencer to generate character clocks 9 dots wide; a logical 1 directs the sequencer to generate character clocks 8 dots wide. Alphanumeric modes hex 0+, 1+, 2+, 3+, 7 and 7+ are the only modes that use character clocks 9 dots wide. All other modes must use 8 dots per character clock. The 9 dot mode is for Alphanumeric modes only. The 9th dot equals the 8th dot for ASCII codes C0 through DF hex. See the attribute line graphics character bit in the Attribute Mode Control register (index hex 10) on page 4-102.

Map Mask Register

This is a read/write register pointed to when the value in the address register is hex 02. The port address for this register is hex 03C5.

Bit	Function
7 - 4	Reserved = 0
3	Map 3 Enable
2	Map 2 Enable
1	Map 1 Enable
0	Map 0 Enable

Figure 4-45. Map Mask Register, Index Hex 02

Bits 7 - 4 Reserved

Bits 3 - 0 Map Mask - A logical 1 enables the system microprocessor to write to the corresponding map. If this register is programmed with a value of 0FH, the system microprocessor can perform a 32-bit write operation with only one memory cycle. This substantially reduces the overhead on the system microprocessor during display update cycles in graphics modes. Data scrolling operations are also enhanced by setting this register to a value of 0FH and writing the display buffer address with the data stored in the system microprocessor data latches. This is a read-modify-write operation. When odd/even modes are selected, maps 0 and 1 and maps 2 and 3 should have the same map mask value. When chain 4 mode is selected, all maps should be enabled.

Character Map Select Register

This is a read/write register pointed to when the value in the Address register is hex 03. The port address for this register is hex 03C5.

Bit	Function
7, 6	Reserved = 0
5	Character Map Select High Bit A
4	Character Map Select High Bit B
3, 2	Character Map Select A
1, 0	Character Map Select B

Figure 4-46. Character Map Select Register, Index Hex 03

Bits 7, 6 Reserved

Bit 5 Character Map Select High Bit A

Bit 4 Character Map Select High Bit B

Bits 3, 2 Character Map Select A - Selects the portion of map 2 used to generate Alpha characters when attribute bit 3 is a 1, according to the following figure:

Bit 5 Value	Bit 3 Value	Bit 2 Value	Map Selected	Table Location
0	0	0	0	1st 8K of Map 2
0	0	1	1	3rd 8K of Map 2
0	1	0	2	5th 8K of Map 2
0	1	1	3	7th 8K of Map 2
1	0	0	4	2nd 8K of Map 2
1	0	1	5	4th 8K of Map 2
1	1	0	6	6th 8K of Map 2
1	1	1	7	8th 8K of Map 2

Figure 4-47. Character Map Select A

In alphanumeric modes, bit 3 of the attribute byte normally has the function of turning the foreground intensity on or off. This bit, however, may be redefined as a switch between character sets. For this feature to be enabled, the following must be true:

- Memory Mode register bit 1, must be equal to 1.
- The value of Character Map Select A does not equal the value of Character Map Select B.

If either condition is not met, the first 16K of Map 2 is used.

Bits 1, 0 Character Map Select B - Selects the portion of map 2 used to generate Alpha characters when attribute bit 3 is a 0, according to the following table:

Bit 4 Value	Bit 1 Value	Bit 0 Value	Map Selected	Table Location
0	0	0	0	1st 8K of Map 2
0	0	1	1	3rd 8K of Map 2
0	1	0	2	5th 8K of Map 2
0	1	1	3	7th 8K of Map 2
1	0	0	4	2nd 8K of Map 2
1	0	1	5	4th 8K of Map 2
1	1	0	6	6th 8K of Map 2
1	1	1	7	8th 8K of Map 2

Figure 4-48. Character Map Select B

Memory Mode Register

This is a read/write register pointed to when the value in the Address register is hex 04. The system microprocessor output port address for this register is 03C5.

Bit	Function
7 - 4	Reserved = 0
3	Chain 4
2	Odd/Even
1	Extended Memory
0	Reserved = 0

Figure 4-49. Memory Mode Register, Index Hex 04

Bits 7 - 4 Reserved

Bit 3 Chain 4 - A logical 0 enables system microprocessor addresses to sequentially access data within a bit map by use of the Map Mask register. A logical 1 causes the two low-order bits to select the map that will be accessed as shown in following figure.

Note: This bit controls the map selected in the graphics subsection during system microprocessor reads.

A1	A0	Map Selected
0	0	0
0	1	1
1	0	2
1	1	3

Figure 4-50. Memory Mode, Chain 4

Bit 2 Odd/Even - A logical 0 directs even system microprocessor addresses to access maps 0 and 2, while odd system microprocessor addresses access maps 1 and 3. A logical 1 causes system microprocessor addresses to sequentially access data within a bit map. The maps are accessed according to the value in the Map Mask register.

Bit 1 Extended Memory - A logical 1 indicates that greater than 64K of video memory is present. This bit must be set to allow the VGA to use the 256K of video memory on the system board, and to enable the character map selection on the previous page.

Bit 0 Reserved

CRT Controller Registers

Name	Port (hex)	Index (hex)
CRT Controller Address	03?4	-
Horizontal Total	03?5	00
Horizontal Display Enable End	03?5	01
Start Horizontal Blanking	03?5	02
End Horizontal Blanking	03?5	03
Start Horizontal Retrace Pulse	03?5	04
End Horizontal Retrace	03?5	05
Vertical Total	03?5	06
Overflow	03?5	07
Preset Row Scan	03?5	08
Maximum Scan Line	03?5	09
Cursor Start	03?5	0A
Cursor End	03?5	0B
Start Address High	03?5	0C
Start Address Low	03?5	0D
Cursor Location High	03?5	0E
Cursor Location Low	03?5	0F
Vertical Retrace Start	03?5	10
Vertical Retrace End	03?5	11
Vertical Display Enable End	03?5	12
Offset	03?5	13
Underline Location	03?5	14
Start Vertical Blank	03?5	15
End Vertical Blank	03?5	16
CRTC Mode Control	03?5	17
Line Compare	03?5	18

? = B in Monochrome Emulation Modes and
? = D in Color Emulation Modes
This is controlled by Bit 0 of the Miscellaneous Output register

Figure 4-51. CRT Controller Register Overview

CRT Controller Address Register

This register is a pointer register located at hex 03B4 or hex 03D4. Which address is used depends on Bit 0 of the Miscellaneous Output register at address hex 03C2. The CRT Address register is loaded with a binary value that points to the CRT Controller Data register where data is to be written. This value is referred to as "Index" in the preceding table. All CRT controller registers are read/write registers.

Bit	Function
7	Reserved = 0
6	Reserved = 0
5	For test, must = 0
4	CRTC Address
3	CRTC Address
2	CRTC Address
1	CRTC Address
0	CRTC Address

Figure 4-52. CRT Controller Address Register

Bits 7, 6 Reserved

Bit 5 This is a test bit for chip testing and must remain 0.

Bits 4 - 0 CRT Controller Address Bits - A binary value pointing to the CRT Controller register where data is to be written.

Horizontal Total Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 00. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7 - 0	Horizontal Total (-5)

Figure 4-53. Horizontal Total Register, Index Hex 00

This register defines the total number of characters in the horizontal scan interval including the retrace time. The value directly controls the period of the horizontal retrace output signal. An internal horizontal character counter counts character clock inputs to the CRT Controller, and all horizontal and vertical timings are based upon the Horizontal register. Comparators are used to compare register values with horizontal character values to provide horizontal timings.

Bits 7 - 0 Horizontal Total - The total number of characters less 5.

Horizontal Display Enable End Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 01. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7 - 0	Horizontal Display Enable End (-1)

Figure 4-54. Horizontal Display Enable End Register, Index Hex 01

This register defines the length of the horizontal display enable signal. It determines the number of displayed character positions per horizontal line.

Bits 7 - 0 Horizontal Display Enable End - A value one less than the total number of displayed characters.

Start Horizontal Blanking Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 02. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7 - 0	Start Horizontal Blanking

Figure 4-55. Start Horizontal Blanking Register, Index Hex 02

This register determines when the horizontal blanking output signal becomes active.

Bits 7 - 0 Start Horizontal Blanking - The horizontal blanking signal becomes active when the horizontal character counter reaches this value.

End Horizontal Blanking Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 03. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7	Test, must be 1
6	Display Enable Skew Control
5	Display Enable Skew Control
4 - 0	End Blanking

Figure 4-56. End Horizontal Blanking Register, Index Hex 03

This register determines when the horizontal blanking output signal becomes inactive.

Bit 7 Test Bit - This bit is used for chip testing and must be set to logical 1.

Bits 6, 5 Display Enable Skew Control - These two bits determine the amount of display enable skew. Display enable skew control is required to provide sufficient time for the CRT Controller to access the display buffer to obtain a character and attribute code, access the character generator font, and then go through the Horizontal PEL Panning register in the Attribute Controller. Each access

requires the display enable signal to be skewed one character clock unit so that the video output is in synchronization with the horizontal and vertical retrace signals. The bit values and amount of skew are shown in the following table:

Bit 6	Bit 5	Skew
0	0	Zero character clock skew
0	1	One character clock skew
1	0	Two character clock skew
1	1	Three character clock skew

Figure 4-57. Bit Values and Amount of Skew

Bits 4 - 0 End Horizontal Blanking - A value equal to the six least-significant bits of the horizontal character counter value at which time the horizontal blanking signal becomes inactive (logical 0). To obtain a blanking signal of width W, the following algorithm is used: Value of Start Blanking register + width of blanking signal in character clock units = 6-bit result to be programmed into the End Horizontal Blanking register. Bit number 5 is located in the End Horizontal Retrace register (index hex 05).

Start Horizontal Retrace Pulse Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 04. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7 - 0	Start Horizontal Retrace Pulse

Figure 4-58. Start Horizontal Retrace Pulse Register, Index Hex 04

This register is used to center the screen horizontally, and to specify the character position at which the Horizontal Retrace Pulse becomes active.

Bits 7 - 0 Start Horizontal Retrace Pulse - The value programmed is a binary count of the character position number at which the signal becomes active.

End Horizontal Retrace Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 05. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7	End Horizontal Blanking, Bit 5
6, 5	Horizontal Retrace Delay
4 - 0	End Horizontal Retrace

Figure 4-59. End Horizontal Retrace Register, Index Hex 05

This register specifies the character position at which the Horizontal Retrace Pulse becomes inactive (logical 0).

Bit 7 End Horizontal Blanking, Bit number 5. The first 4 bits are located in the End Horizontal Blanking register (index hex 03).

Bits 6, 5 Horizontal Retrace Delay - These bits control the skew of the horizontal retrace signal. Binary 00 equals no horizontal retrace delay. For some modes, it is necessary to provide a horizontal retrace signal that takes up the entire blanking interval. Some internal timings are generated by the falling edge of the horizontal retrace signal. To guarantee the signals are latched properly, the retrace signal is started before the end of the display enable signal, and then skewed several character clock times to provide the proper screen centering.

Bits 4 - 0 End Horizontal Retrace - A value equal to the five least-significant bits of the horizontal character counter value at which time the horizontal retrace signal becomes inactive (logical 0). To obtain a retrace signal of width W, the following algorithm is used: Value of Start Retrace register + width of horizontal retrace signal in character clock units = 5-bit result to be programmed into the End Horizontal Retrace register.

Vertical Total Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 06. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7 - 0	Vertical Total (-2)

Figure 4-60. Vertical Total Register, Index Hex 06

Bits 7 - 0 Vertical Total - This is the low-order eight bits of a 10-bit register. The binary value represents the number of horizontal raster scans on the CRT screen, minus 2, including vertical retrace. The value in this register determines the period of the vertical retrace signal.

Bit 8 of this register is contained in the CRT Controller Overflow register hex 07 bit 0.

Bit 9 of this register is contained in the CRT Controller Overflow register hex 07 bit 5.

CRT Controller Overflow Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 07. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7	Vertical Retrace Start Bit 9
6	Vertical Display Enable End Bit 9
5	Vertical Total Bit 9
4	Line Compare Bit 8
3	Start Vertical Blank Bit 8
2	Vertical Retrace Start Bit 8
1	Vertical Display Enable End Bit 8
0	Vertical Total Bit 8

Figure 4-61. CRTC Overflow Register, Index Hex 07

Bit 7 Vertical Retrace Start - Bit 9 of the Vertical Retrace Start register (index hex 10).

Bit 6 Vertical Display Enable End - Bit 9 of the Vertical Display Enable End register (index hex 12).

Bit 5 Vertical Total - Bit 9 of the Vertical Total register (index hex 06).

Bit 4 Line Compare - Bit 8 of the Line Compare register (index hex 18).

Bit 3 Start Vertical Blank - Bit 8 of the Start Vertical Blank register (index hex 15).

Bit 2 Vertical Retrace Start - Bit 8 of the Vertical Retrace Start register (index hex 10).

Bit 1 Vertical Display Enable End - Bit 8 of the Vertical Display Enable End register (index hex 12).

Bit 0 Vertical Total - Bit 8 of the Vertical Total register (index hex 06).

Preset Row Scan Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 08. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7	Reserved = 0
6	Byte Panning Control
5	Byte Panning Control
4 - 0	Starting Row scan count after a Vertical Retrace

Figure 4-62. Preset Row Scan Register, Index Hex 08

This register is used for PEL scrolling.

Bit 7 Reserved

Bits 6, 5 Byte Panning Control - These two bits control byte panning in modes programmed as multiple shift modes. (Currently, no modes are programmed for multiple shift operation.) This is required for PEL-panning operations. The PEL Panning register in the attribute section provides panning of up to 7 or 8 individual PELs. In single byte shift modes, to pan to the next higher PEL (8 or 9), the CRT Controller start address is incremented and attribute panning is reset to 0. In multiple shift modes, the byte pan bits are used as extensions to the attribute PEL Panning register. This allows panning across the width of the video output shift. For example, in the 32-bit shift mode, the byte pan and PEL-panning bits provide up to 31 bits of panning capability. To pan from position 31 to 32, the CRT controller start address is incremented and PEL- and

byte-panning is reset to 0. These bits should normally be set to 0.

Bits 4 - 0 Preset Row Scan (PEL Scrolling) - This register specifies the starting row scan count after a vertical retrace. The row scan counter increments each horizontal retrace time until a maximum row scan occurs. At maximum row scan compare time, the row scan is cleared (not preset).

Maximum Scan Line Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 09. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7	200 --> 400 Line Conversion
6	Line Compare Bit 9
5	Start Vertical Blank Bit 9
4 - 0	Maximum Scan Line

Figure 4-63. Maximum Scan Line Register, Index Hex 09

Bit 7 200 to 400 Line Conversion is done when this bit is set to 1. The clock in the row scan counter is divided by 2. This allows the older 200-line modes to be displayed as 400 lines on the display (this is line doubling; each line is displayed twice). When this bit is a 0, the clock to the row scan counter is equal to the horizontal scan rate. Line doubling is not enabled.

Bit 6 Line Compare - Bit 9 of the Line Compare register (index hex 18).

Bit 5 Start Vertical Blank - Bit 9 of the Start Vertical Blank register (index hex 15).

Bits 4 - 0 Maximum Scan Line - This register specifies the number of scan lines per character row. The number to be programmed is the maximum row scan number minus 1.

Cursor Start Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 0A. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7, 6	Reserved = 0
5	Cursor Off
4 - 0	Row Scan Cursor Begins

Figure 4-64. Cursor Start Register, Index Hex 0A

Bits 7, 6 Reserved

Bit 5 Cursor Off - A logical 1 turns off the cursor, a logical 0 turns on the cursor.

Bits 4 - 0 Cursor Start - This register specifies the row scan of a character line where the cursor is to begin. The number programmed is one less than the starting cursor row scan.

When the Cursor Start register is programmed with a value greater than the Cursor End register, no cursor is generated.

Cursor End Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 0B. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7	Reserved = 0
6, 5	Cursor Skew Control
4 - 0	Row Scan Cursor Ends

Figure 4-65. Cursor End Register, Index Hex 0B

Bit 7 Reserved

Bits 6, 5 Cursor Skew - These bits control the skew of the cursor signal. Cursor skew delays the cursor by the selected number of clocks. For example, a skew of 1 moves the cursor right one position on the screen.

Bit 6	Bit 5	Function
0	0	Zero-character clock skew
0	1	One-character clock skew
1	0	Two-character clock skew
1	1	Three-character clock skew

Figure 4-66. Cursor Skew

Bits 4 - 0 Cursor End - These bits specify the row scan of a character line where the cursor is to end.

Start Address High Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 0C. The system microprocessor input/output port address for this register is hex 03?5.

Bit	Function
7 - 0	High Order Start Address

Figure 4-67. Start Address High Register, Index Hex 0C

Bits 7 - 0 Start Address High - These are the high-order 8 bits of the start address. The 16-bit value, from the high-order and low-order Start Address registers, is the first address after the vertical retrace on each screen refresh.

Start Address Low Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 0D. The system microprocessor input/output port address for this register is hex 03?5.

Bit	Function
7 - 0	Low Order Start Address

Figure 4-68. Start Address Low Register, Index Hex 0D

Bits 7 - 0 Start Address Low - These are the low-order 8 bits of the start address.

Cursor Location High Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 0E. The system microprocessor input/output port address for this register is hex 03?5.

Bit	Function
7 - 0	High Order Cursor Location

Figure 4-69. Cursor Location High Register, Index Hex 0E

Bits 7 - 0 Cursor Location High - These are the high-order 8 bits of the cursor location.

Cursor Location Low Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 0F. The system microprocessor input/output port address for this register is hex 03?5.

Bit	Function
7 - 0	Low Order Cursor Location

Figure 4-70. Cursor Location Low Register, Index Hex 0F

Bits 7 - 0 Cursor Location Low - These are the low-order 8 bits of the cursor location.

Vertical Retrace Start Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 10. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7 - 0	Low order Vertical Retrace Pulse

Figure 4-71. Vertical Retrace Start Register, Index Hex 10

Bits 7 - 0 Vertical Retrace Start - This is the low-order 8 bits of the vertical retrace pulse start position, programmed in horizontal scan lines. Bit 8 is in the CRTC Overflow register (index hex 07).

Vertical Retrace End Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 11. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7	Protect R0-7
6	Select 5 Refresh Cycles
5	0=Enable Vertical Interrupt
4	0=Clear Vertical Interrupt
3 - 0	Vertical Retrace End

Figure 4-72. Vertical Retrace End Register, Index Hex 11

Bit 7 Protect R0-7 - A logical 1 disables writing to CRTC registers 0-7. A logical 0 enables writing to R0-7. The line compare bit 4 in register hex 07 is not protected.

Bit 6 Select 5 Refresh Cycles - A logical 1 generates five DRAM refresh cycles per horizontal line. A logical 0 selects three refresh cycles. Selecting five refresh cycles allows use of the VGA chip with slow sweep rate displays (15.75 kHz). This bit is set to 0 by BIOS during a mode set, a reset, or power on.

Bit 5 Enable Vertical Interrupt - A logical 0 enables a vertical retrace interrupt. The vertical retrace interrupt is on IRQ2. This interrupt level may be shared so the Input Status register 0, bit 7, should be checked to find out if the VGA

generated the interrupt. As with bit 4, do not change the value of the other bits in this register.

Bit 4 Clear Vertical Interrupt - A logical 0 clears a vertical retrace interrupt. At the end of the active vertical display time, a flip-flop is set in the VGA for vertical interrupt. The output of this flip-flop goes to the system board interrupt controller. An interrupt handler has to reset this flip-flop by writing a 0 to this bit, then setting the bit to 1 so that the flip-flop does not hold interrupts inactive. Do not change the other bits in this register. The register is readable, so a read can be done to determine what the other bits are before the flip-flop is reset.

Bits 3 - 0 Vertical Retrace End - These bits determine the horizontal scan count value when the vertical retrace output signal becomes inactive. The register is programmed in units of horizontal scan lines. To obtain a vertical retrace signal of width W, the following algorithm is used: Value of Start Vertical Retrace register + width of vertical retrace signal in horizontal scan units = 4-bit result to be programmed into the End Horizontal Retrace register.

Vertical Display Enable End Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 12. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7 - 0	Low Order Vertical Display Enable End (-1)

Figure 4-73. Vertical Display Enable End Register, Index Hex 12

Bits 7 - 0 Vertical Display Enable End - These are the low-order 8 bits of a 10-bit register that defines the vertical display enable end position.

Bit 8 of this register is contained in the CRT Controller Overflow register hex 07, bit 1.

Bit 9 of this register is contained in the CRT Controller Overflow register hex 07, bit 6.

This register specifies which scan line ends the active video area of the screen. It is programmed with the total number of lines minus one.

Offset Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 13. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7 - 0	Logical Line Width of the Screen

Figure 4-74. Offset Register, Index Hex 13

Bits 7 - 0 Offset - This register specifies the logical line width of the screen. The starting memory address for the next character row is larger than the current character row by two or four times this amount. The Offset register is programmed with a word address. Depending on the method of clocking the CRT Controller, this word address is either a word or doubleword address.

Underline Location Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 14. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7	Reserved = 0
6	Doubleword Mode
5	Count By 4
4 - 0	Horizontal row scan where underline will occur

Figure 4-75. Underline Location Register, Index Hex 14

Bit 7 Reserved

Bit 6 Doubleword Mode - When this bit is a logical 1, memory addresses are doubleword addresses. See the description of the CRTC Mode Control register (index hex 17), bit 6, on page 4-87.

Bit 5 Count By 4 - When this bit is a logical 1, the memory address counter is clocked with the character clock divided by 4. This bit is used when doubleword addresses are used.

Bits 4 - 0 Underline Location - This register specifies the horizontal row scan of a character row on which an underline occurs. The value programmed is one less than the scan line number desired.

Start Vertical Blanking Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 15. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7 - 0	Start Vertical Blanking (-1)

Figure 4-76. Start Vertical Blanking Register, Index Hex 15

Bits 7 - 0 Start Vertical Blank - These are the low 8 bits of a 10-bit register.

Bit 8 is in the CRTC Overflow register (index hex 07).

Bit 9 is in the Maximum Scan Line register (index hex 09).

The value of these 10 bits is one less than the horizontal scan line count at which the vertical blanking signal becomes active.

End Vertical Blanking Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 16. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7 - 0	End Vertical Blanking

Figure 4-77. End Vertical Blanking Register, Index Hex 16

Bits 7 - 0 End Vertical Blank - This register specifies the horizontal scan count value when the vertical blank output signal becomes inactive. The register is programmed in units of horizontal scan lines.

To obtain a vertical blank signal of width W, the following algorithm is used: (Value of Start Vertical Blank register minus 1) + width of vertical blank signal in horizontal scan units = 8-bit result to be programmed into the End Vertical Blank register.

CRTC Mode Control Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 17. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7	Hardware Reset
6	Word/Byte Mode
5	Address Wrap
4	Reserved = 0
3	Count By Two
2	Horizontal Retrace Select
1	Select Row Scan Counter
0	CMS 0

Figure 4-78. CRTC Mode Control Register, Index Hex 17

Bit 7 Hardware Reset - A logical 0 forces horizontal and vertical retrace to clear. A logical 1 forces horizontal and vertical retrace to be enabled. This bit does not reset any other registers or outputs.

Bit 6 Word Mode or Byte Mode - When this bit is a logical 0, the word mode shifts all memory address counter bits down one bit, and the most-significant bit of the counter appears on the least-significant bit of the memory address outputs. See Figure 4-79 on page 4-88 for address output details. A logical 1 selects the byte address mode.

Bit 6 of the End Vertical Blanking register in the CRT Controller also controls the addressing. When it is a 0,bit 6 above has control. When it is a 1, the addressing is forced to be shifted by two bits. See doubleword addressing in Figure 4-79 on page 4-88.

Memory Address	Byte Address Mode	Word Address Mode	Doubleword Address Mode
MA 0/RFA 0	MA 0	MA 15 or MA 13	MA 12
MA 1/RFA 1	MA 1	MA 0	MA 13
MA 2/RFA 2	MA 2	MA 1	MA 0
MA 3/RFA 3	MA 3	MA 2	MA 1
MA 4/RFA 4	MA 4	MA 3	MA 2
MA 5/RFA 5	MA 5	MA 4	MA 3
MA 6/RFA 6	MA 6	MA 5	MA 4
MA 7/RFA 7	MA 7	MA 6	MA 5
MA 8/RFA 8	MA 8	MA 7	MA 6
MA 9	MA 9	MA 8	MA 7
MA 10	MA 10	MA 9	MA 8
MA 11	MA 11	MA 10	MA 9
MA 12	MA 12	MA 11	MA 10
MA 13	MA 13	MA 12	MA 11
MA 14	MA 14	MA 13	MA 12
MA 15	MA 15	MA 14	MA 13

Figure 4-79. Internal Memory Address Counter Wiring to the Output Multiplexer

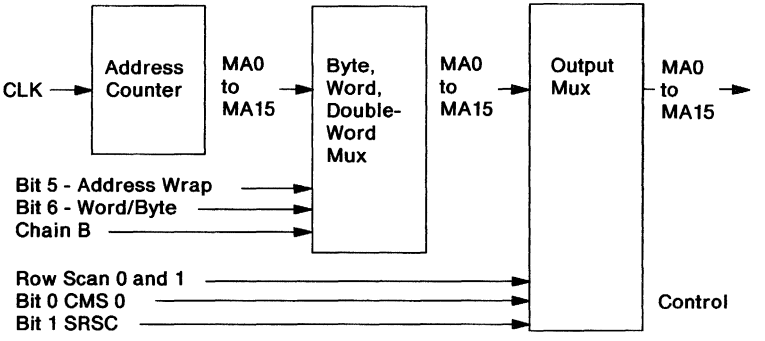


Figure 4-80. CRTC Memory Address Mapping

Bit 5 Address Wrap - This bit selects memory address counter bit MA 13 or bit MA 15, and it appears on the MA 0 output pin in the word address mode. If the VGA is not in the word address mode, MA 0 counter output appears on the MA 0 output pin. A logical 1 selects MA 15. In odd/even mode, bit MA 15 should be selected since 256K of video memory is installed on the system board. (Bit MA 13 is selected in applications where only 64K memory is

present. This function implements IBM Color/Graphics Monitor Adapter compatibility.)

Bit 4 Reserved

Bit 3 Count By Two - When this bit is set to 0, the memory address counter is clocked with the character clock input. A logical 1 clocks the memory address counter with the character clock input divided by 2. This bit is used to create either a byte or word refresh address for the display buffer.

Bit 2 Horizontal Retrace Select - This bit selects horizontal retrace or horizontal retrace divided by 2 as the clock that controls the vertical timing counter. This bit can be used to effectively double the vertical resolution capability of the CRT Controller. The vertical counter has a maximum resolution of 1024 scan lines due to the 10-bit wide Vertical Total register. If the vertical counter is clocked with the horizontal retrace divided by 2, then the vertical resolution is doubled to 2048 horizontal scan lines. A logical 0 selects HRTC and a logical 1 selects HRTC divided by 2.

Bit 1 Select Row Scan Counter - A logical 0 selects row scan counter bit 1 on the MA 14 output pin. A logical 1 selects the MA 14 counter bit on the MA 14 output pin.

Bit 0 Compatibility Mode Support - When this bit is a logical 0, row scan address bit 0 is substituted for memory address bit 13 during active display time. A logical 1 enables memory address bit 13 to appear on the memory address output bit 13 signal of the CRT Controller. The CRT Controller used on the IBM Color/Graphics Monitor Adapter is the 6845. The 6845 has 128 horizontal scan line address capability. To obtain 640-by-200 graphics resolution, the CRTC is programmed for 100 horizontal scan lines with 2 row scan addresses per character row. Row scan address bit 0 becomes the most-significant address bit to the display buffer. Successive scan lines of the display image are displaced in memory by 8K bytes. This bit allows compatibility with the 6845 and Color Graphics APA modes of operation.

Line Compare Register

This is a read/write register pointed to when the value in the CRT Controller Address register is hex 18. The system microprocessor output port address for this register is hex 03?5.

Bit	Function
7 - 0	Line Compare Target

Figure 4-81. Line Compare Register, Index Hex 18

Bits 7 - 0 Line Compare - This register is the low-order 8 bits of the compare target. When the vertical counter reaches this value, the internal start of the line counter is cleared. Because of this, an area of the screen is not affected by scrolling. Bit 8 of this register is in the Overflow register hex 07. Bit 9 is in the maximum scan line register hex 09.

Graphics Controller Registers

Name	Port (hex)	Index (hex)
Graphics Address	03CE	-
Set/Reset	03CF	00
Enable Set/Reset	03CF	01
Color Compare	03CF	02
Data Rotate	03CF	03
Read Map Select	03CF	04
Graphics Mode	03CF	05
Miscellaneous	03CF	06
Color Don't Care	03CF	07
Bit Mask	03CF	08

Figure 4-82. Graphics Controller Register Overview

Graphics Address Register

This is a read/write register and the system microprocessor output port address for this register is hex 03CE.

Bit	Function
7 - 4	Reserved = 0
3 - 0	Graphics Address

Figure 4-83. Graphics Address Register

Bits 7 - 4 Reserved

Bits 3 - 0 Graphics Address Bits - These bits point to the other registers in the graphics section.

Set/Reset Register

This is a read/write register pointed to by the value in the Graphics Address register. This value must be hex 00 before writing can take place. The system microprocessor output port address for this register is hex 03CF.

Bit	Function
7 - 4	Reserved = 0
3	Set/Reset Map 3
2	Set/Reset Map 2
1	Set/Reset Map 1
0	Set/Reset Map 0

Figure 4-84. Set/Reset Register, Index Hex 00

Bits 7 - 4 Reserved

Bits 3 - 0 Set/Reset - These bits represent the value written to all 8 bits of the respective memory map when the system microprocessor does a memory write with write mode 0 selected and Set/Reset mode is enabled. Set/Reset can be enabled on a map-by-map basis with separate Out commands to the Enable Set/Reset register.

Enable Set/Reset Register

This is a read/write register and is pointed to by the value in the Graphics Address register. This value must be hex 01 before writing can take place. The system microprocessor output port for this register is hex 03CF.

Bit	Function
7 - 4	Reserved = 0
3	Enable Set/Reset Map 3
2	Enable Set/Reset Map 2
1	Enable Set/Reset Map 1
0	Enable Set/Reset Map 0

Figure 4-85. Enable Set/Reset Register, Index Hex 01

Bits 7 - 4 Reserved

Bits 3 - 0 Enable Set/Reset - These bits enable the set/reset function. When enabled (bit = 1) the respective memory map is written with the value of the Set/Reset register provided the write mode is 0. When write mode is 0 and Set/Reset is not enabled (bit equals 0) on a map, that map is written with the value of the system microprocessor data.

Color Compare Register

This is a read/write register pointed to by the value in the Graphics Address register. This value must be hex 02 before writing can take place. The system microprocessor output port address for this register is hex 03CF.

Bit	Function
7 - 4	Reserved = 0
3	Color Compare Map 3
2	Color Compare Map 2
1	Color Compare Map 1
0	Color Compare Map 0

Figure 4-86. Color Compare Register, Index Hex 02

Bits 7 - 4 Reserved

Bits 3 - 0 Color Compare - These bits represent a 4-bit color value to be compared. If the system microprocessor sets read mode 1 in the graphics section and does a memory read, the data returned from the memory cycle will be a 1 in each bit position where the 4 maps equal the Color Compare register.

The color compare bit is the value that all bits of the corresponding map's byte are compared with. Each of the 8 bit positions of the selected byte are then compared across the four maps and a 1 is returned in each bit position where the bits of all four maps equal their respective color compare values.

Data Rotate Register

This is a read/write register pointed to by the value in the Graphics Address register. This value must be hex 03 before writing can take place. The system microprocessor output port address for this register is hex 03CF.

Bit	Function
7 - 5	Reserved = 0
4	Function Select
3	Function Select
2	Rotate Count 2
1	Rotate Count 1
0	Rotate Count 0

Figure 4-87. Data Rotate Register, Index Hex 03

Bits 7 - 5 Reserved

Bits 4, 3 Function Select - Data written to memory can operate logically with data already in the system microprocessor latches.

Data can be any of the choices selected by the Write Mode register except system microprocessor latches, which may not be modified. If rotated data is selected, the rotate applies before the logical function. The bit functions are defined in the following.

Bit 4	Bit 3	Function
0	0	Data unmodified
0	1	Data ANDed with latched data.
1	0	Data ORed with latched data.
1	1	Data XORed with latched data.

Figure 4-88. Function Select Bit Definitions

Bits 2 - 0 Rotate Count - These bits represent a binary encoded value of the number of positions to right-rotate the system microprocessor data bus during system microprocessor memory writes. This operation is done when the write mode is 0. To write non-rotated data the system microprocessor must select a count of 0.

Read Map Select Register

This is a read/write register pointed to by the value in the Graphics Address register. This value must be hex 04 before writing can take place. The system microprocessor output port address for this register is hex 03CF.

Bit	Function
7 - 2	Reserved = 0
1	Map Select 1
0	Map Select 0

Figure 4-89. Read Map Select Register, Index Hex 04

Bits 7 - 2 Reserved

Bits 1, 0 Map Select - These bits represent a binary encoded value of the memory map number from which the system microprocessor reads data. This register has no effect on the color compare read mode. In odd/even modes the value may be 00 or 01 (10 or 11) for chained maps 0, 1 (2, 3).

Graphics Mode Register

This is a read/write register pointed to by the value in the Graphics Address register. This value must be hex 05 before writing can take place. The system microprocessor output port address for this register is 03CF.

Bit	Function
7	Reserved = 0
6	256 color mode
5	Shift Register Mode
4	Odd/Even
3	Read Type
2	Reserved = 0
1, 0	Write Mode

Figure 4-90. Graphics Mode Register, Index Hex 05

Bit 7 Reserved

Bit 6 256 Color Mode - A logical 0 allows bit 5 to control the loading of the Shift registers. A logical 1 causes the Shift registers to be loaded in a manner that supports the 256-color mode.

Bit 5 Shift register - A logical 1 directs the Shift registers in the graphics section to format the serial data stream with even-numbered bits from both maps on the even-numbered maps and odd-numbered bits from both maps on the odd maps. This bit is used for modes 4 and 5.

Bit 4 Odd/Even - A logical 1 selects the odd/even addressing mode, which is useful for emulation of the IBM Color/Graphics Monitor Adapter compatible modes. Normally the value here follows the value of the Memory Mode register bit 2 of the Sequencer.

Bit 3 Read Type - When this bit is a logical 0, the system microprocessor reads data from the memory map selected by the Read Map Select register, unless bit 3 of the Sequencer Memory Mode register equals 1. In this case the Read Map Select register has no effect. When this bit is a logical 1, the system microprocessor reads the results of the comparison of the four memory maps and the Color Compare register.

Bit 2 Reserved

Bits 1, 0 Write Mode

The bit functions are defined as follows:

Bit 1	Bit 0	Function
0	0	Each memory map is written with the system microprocessor data rotated by the number of counts in the Rotate register, unless Set/Reset is enabled for the map. Maps for which Set/Reset is enabled are written with 8 bits of the value contained in the Set/Reset register for that map.
0	1	Each memory map is written with the contents of the system microprocessor latches. These latches are loaded by a system microprocessor Read operation.
1	0	Memory map *n* (0 through 3) is filled with 8 bits of the value of data bit *n*.
1	1	Each map is written with 8 bits of the value contained in the Set/Reset register for that map (the Enable Set/Reset register has no effect). Rotated system microprocessor data is ANDed with the Bit Mask register data to form an 8-bit value that performs the same function as the Bit Mask register does in write modes 0 and 2 (see also Bit Mask register on page 4-99).

Figure 4-91. Write Mode Bit Definitions

The logic function specified by the Function Select register is applied to data being written to memory following modes 0, 2, and 3 above.

Miscellaneous Register

This is a read/write register pointed to by the value in the Graphics Address register. This value must be hex 06 before writing can take place. The system microprocessor output port for this register is hex 03CF.

Bit	Function
7 - 4	Reserved = 0
3	Memory Map 1
2	Memory Map 0
1	Odd/Even
0	Graphics Mode

Figure 4-92. Miscellaneous Register, Index Hex 06

Bits 7 - 4 Reserved

Bits 3, 2 Memory Map - These bits control the mapping of the regenerative buffer into the system microprocessor address space. The bit functions are defined below:

Bit 3	Bit 2	Function
0	0	Hex A0000 for 128K bytes
0	1	Hex A0000 for 64K bytes
1	0	Hex B0000 for 32K bytes
1	1	Hex B8000 for 32K bytes

Figure 4-93. Miscellaneous Register, Bits 3 and 2

Bit 1 Odd/Even - When set to 1, this bit directs the system microprocessor address bit 0 to be replaced by a higher-order bit and odd/even maps to be selected with odd/even values of the system microprocessor A0 bit, respectively.

Bit 0 Graphics Mode - This bit controls alphanumeric mode addressing. A logical 1 selects graphics mode. When set to graphics mode, the character generator address latches are disabled.

Color Don't Care Register

This is a read/write register and is pointed to by the value in the Graphics Address register. This value must be hex 07 before writing can take place. The system microprocessor output port for this register is hex 03CF.

Bit	Function
7 - 4	Reserved = 0
3	Map 3 = Don't Care
2	Map 2 = Don't Care
1	Map 1 = Don't Care
0	Map 0 = Don't Care

Figure 4-94. Color Don't Care Register, Index Hex 07

Bits 7 - 4 Reserved

Bit 3 Map 3:

1 - Participate in the color compare cycle.
0 - Don't participate in the color compare cycle.

Bit 2 Map 2:

1 - Participate in the color compare cycle.
0 - Don't participate in the color compare cycle.

Bit 1 Map 1:

1 - Participate in the color compare cycle.
0 - Don't participate in the color compare cycle.

Bit 0 Map 0:

1 - Participate in the color compare cycle.
0 - Don't participate in the color compare cycle.

Bit Mask Register

This is a read/write register and is pointed to by the value in the Graphics Address register. This value must be hex 08 before writing can take place. The system microprocessor output port for this register is hex 03CF.

Bit	Function
7 - 0	0-Immune to Change, 1-Unimpeded Writes

Figure 4-95. Bit Mask Register, Index Hex 08

Bits 7 - 0 Bit Mask - Any bit programmed to 0 causes the corresponding bit **n** in each map to be immune to change, provided that the location being written was the last location read by the system microprocessor. Bits programmed to a 1 allow unimpeded writes to the corresponding bits in the maps.

The bit mask applies to write modes 0 and 2. To preserve bits using the bit mask, data must be latched internally by reading the location. When data is written to preserve the bits, the most current data in the latches is written in those positions. The bit mask applies to all maps simultaneously.

Attribute Controller Registers

Name	Port (hex)	Index (hex)
Attribute Address	03C0	-
Palette	03C0	00-0F
Attribute Mode Control	03C0	10
Overscan Color	03C0	11
Color Plane Enable	03C0	12
Horizontal PEL Panning	03C0	13
Color Select	03C0	14

Figure 4-96. Attribute Controller Register Overview

Each attribute data register is written at 03C0 as described below. Data is read from them at address 03C1.

Attribute Address Register

This is a read/write register. The system microprocessor output port is hex 03C0.

Bit	Function
7, 6	Reserved = 0
5	Palette Address Source
4 - 0	Attribute Address

Figure 4-97. Attribute Address Register

Bits 7, 6 Reserved

Bit 5 Palette Address Source - When loading the Color Palette registers, bit 5 must be cleared to 0.

For normal operation of the attribute controller, bit 5 must be set to 1. This enables the video memory data to access the palette registers.

Bits 4 - 0 Attribute Address Bits - The Attribute Address register is a pointer register located at hex 03C0. This register is loaded with a binary value that points to the Attribute Data register where data is to be written.

The Attribute Controller register does not have an address bit input to control selection of the Address and Data registers. An internal address flip-flop controls selection

of either the Address or Data registers. To initialize the flip-flop, an IOR instruction is issued to the Attribute Controller at address 03BA or 03DA. This clears the flip-flop, and selects the Address register. After the Address register has been loaded with an out to 03C0, the next Out instruction to 03C0 loads the Data register. The flip-flop toggles each time an Out instruction is issued to the Attribute Controller. It does not toggle on In instructions for Read to 03C1. Also see "Video Graphics Array Programming Considerations" on page 4-107.

Palette Registers Hex 00 through Hex 0F

This is a read/write register. Write at address hex 03C0; Read at address hex 03C1.

Bit	Function
7, 6	Reserved = 0
5	P5
4	P4
3	P3
2	P2
1	P1
0	P0

Figure 4-98. Palette Registers Hex 00 through Hex 0F, Index Hex 00-0F

Bits 7, 6 Reserved

Bits 5 - 0 Palette - These 6-bit registers allow a dynamic mapping between the text attribute or graphic color input value and the display color on the CRT screen. A logical 1 selects the appropriate color. The Palette registers should be modified only during the vertical retrace interval to avoid problems with the displayed image. These internal Palette register values are sent off the chip to the video DAC, where they in turn serve as addresses into the DAC internal registers. Also see the Attribute Controller block diagram on page 4-26.

Attribute Mode Control Register

This is a read/write register pointed to by the value in the Attribute Address register. This value must be hex 10 before writing can take place. The system microprocessor output port address for this register is hex 03C0. The system microprocessor input port address for this register is hex 03C1.

Bit	Function
7	P5, P4 Select
6	PEL Width
5	PEL Panning Compatibility
4	Reserved = 0
3	Select Background Intensity or Enable Blink
2	Enable Line Graphics Character Code
1	Mono Emulation
0	Graphics/Alphanumeric Mode

Figure 4-99. Attribute Mode Control Register, Index Hex 10

Bit 7 P5, P4 Select - This bit selects the source for the P5 and P4 digital video bits that go off the chip. When this bit equals 0, P5, P4 are the outputs of the Palette registers. When this bit equals 1, P5, P4 are bits 1, 0 of the Color Select register. Also see the Attribute Controller block diagram on page 4-26, and "Video DAC Programming Considerations" on page 4-117.

Bit 6 PEL Width - When this bit equals 1, the video pipeline is sampled so that 8 bits are available to select a color in the 256-color mode (hex 13). This bit should equal 0 in all other modes.

Bit 5 PEL Panning Compatibility - When this bit equals 0, line compare has no effect on the output of the PEL Panning register. When this bit equals 1, a successful line compare in the CRT controller forces the output of the PEL Panning register to 0 until +VSYNC occurs, at which time the output returns to its programmed value. This bit allows a selected portion of a screen to be panned.

Bit 4 Reserved

Bit 3 Enable Blink/Select Background Intensity - A logical 0 selects the background intensity of the attribute input. This mode was available on the Monochrome and Color/Graphics Monitor adapters. A logical 1 enables the

blink attribute in alphanumeric modes. This bit must also be set to 1 for blinking graphics modes.

Bit 2 Enable Line Graphics Character Codes - When this bit is set to 0, the ninth dot will be the same as the background. A logical 1 enables the special line graphics character codes for the IBM Monochrome emulation mode. This bit when enabled forces the ninth dot of a line graphic character to be identical to the eighth dot of the character. The line graphics character codes for the Monochrome emulation mode are hex C0 through hex DF.

For character fonts that do not utilize the line graphics character codes in the range of hex C0 through hex DF, bit 2 should be a logical 0. Otherwise unwanted video information will be displayed on the CRT screen.

BIOS will set this bit, the correct dot clock, and other registers when a 9-dot alphanumeric mode is set.

Bit 1 Mono Emulation - A logical 1 indicates a monochrome emulation mode is set. A logical 0 indicates a color emulation mode is set.

Bit 0 Graphics/Alphanumeric Mode - A logical 0 selects alphanumeric mode. A logical 1 selects graphics mode.

Overscan Color Register

This is a read/write register pointed to by the value in the Attribute Address register. This value must be hex 11 before writing can take place. The system microprocessor output port address for this register is hex 03C0. The system microprocessor input port address for this register is hex 03C1.

Bit	Function
7	P7
6	P6
5	P5
4	P4
3	P3
2	P2
1	P1
0	P0

Figure 4-100. Overscan Color Register, Index Hex 11

Bits 7 - 0 Overscan Color - This 8-bit register determines the overscan (border) color displayed on the CRT screen.

The border is a band of color around the perimeter of the display area. Its width is the same as one 80-column character. This border is not supported in the 40-column alphanumeric modes or the 320-PEL graphics modes, except for mode hex 13.

Color Plane Enable Register

This is a read/write register pointed to by the value in the Attribute Address register. This value must be hex 12 before writing can take place. The system microprocessor output port address for this register is 03C0. The system microprocessor input port address for this register is 03C1.

Bit	Function
7, 6	Reserved = 0
5, 4	Video Status MUX
3 - 0	Enable Color Plane

Figure 4-101. Color Plane Enable Register, Index Hex 12

Bits 7, 6 Reserved

Bits 5, 4 Video Status MUX - Selects two of the eight color outputs to be available on the status port. The following figure illustrates the combinations available and the color output wiring.

Color Plane Register		Input Status Register 1	
Bit 5	**Bit 4**	**Bit 5**	**Bit 4**
0	0	P2	P0
0	1	P5	P4
1	0	P3	P1
1	1	P7	P6

Figure 4-102. Color Output Wiring

Bits 3 - 0 Enable Color Plane - Writing a logical 1 in any of bits 0 through 3 enables the respective display memory color plane.

Horizontal PEL Panning Register

This is a read/write register pointed to by the value in the Attribute Address register. This value must be hex 13 before writing can take place. The system microprocessor output port address for this register is hex 03C0. The system microprocessor input port address for this register is hex 03C1.

Bit	Function
7 - 4	Reserved = 0
3 - 0	Horizontal PEL Panning

Figure 4-103. Horizontal PEL Panning Register, Index Hex 13

Bits 7 - 4 Reserved

Bits 3 - 0 Horizontal PEL Panning - This 4-bit register selects the number of picture elements (PELs) to shift the video data horizontally to the left. PEL panning is available in both A/N and APA modes. In monochrome emulation A/N modes, and modes 0+, 1+, 2+, 3+, the image can be shifted a maximum of 8 PELs. In 256-color APA mode, the image can be shifted a maximum of three PELs. Further panning may be accomplished by changing the start address in the CRT controller. In all other A/N and APA modes, the image can be shifted a maximum of seven PELs. The sequence for shifting the image is given below:

PEL Panning Register Value	Number of PELs Shifted to the Left		
	0+, 1+, 2+, 3+, 7, 7+	All Other Modes	Mode 13
0	1	0	0
1	2	1	-
2	3	2	1
3	4	3	-
4	5	4	2
5	6	5	-
6	7	6	3
7	8	7	-
8	0	-	-

Figure 4-104. Image Shifting

Color Select Register

This is a read/write register pointed to by the value in the Attribute Address register. This value must be hex 14 before writing can take place. The system microprocessor port address for this register is hex 03C0 (write), and 03C1 (read).

Bit	Function
7 - 4	Reserved = 0
3	S_color 7
2	S_color 6
1	S_color 5
0	S_color 4

Figure 4-105. Color Select Register, Index Hex 14

Bits 3, 2 S_color 7-6 - In all modes but the 256 color graphics, these bits are the two high-order bits of the 8-bit digital color value sent off-chip. In 256-color graphics, the 8-bit attribute stored in video memory becomes the 8-bit digital color value sent off-chip to the video DAC. These bits are also used to rapidly switch between sets of colors in the video DAC. Also see "Video Graphics Array Programming Considerations."

Bits 1, 0 S_color 5-4 - These bits can be used in place of the P4 and P5 bits from the Attribute Palette registers to form the 8-bit digital color value sent off-chip. (See Attribute Mode Control register bit 7 on page 4-102.) This feature is used to rapidly switch between sets of colors in the video DAC.

Video Graphics Array Programming Considerations

The following are some programming considerations for the Video Graphics Array.

- Certain internal timings must be guaranteed by the user, in order to have the CRTC perform properly. This is due to the physical design of the chip. These timings can be guaranteed by ensuring that the rules listed below are followed when programming the CRTC.

  1. The Horizontal Total register (R0) must be greater than or equal to a value of 25 decimal.

2. The minimum positive pulse width of the HSYNC output must be four character clock units.

3. Register R5 (Horizontal Sync End) must be programmed such that the HSYNC output goes to a logic 0 a minimum of one character clock time before the 'horizontal display enable' signal goes to a logical 1.

4. Register R16 (Vsync Start) must be a minimum of one horizontal scan line greater than register R18. Register R18 defines where the 'vertical display enable' signal ends.

All the above rules are satisfied when the video mode is set by the BIOS.

- When bit 5 of the Attribute Mode Control register equals 1, a successful line compare (see Line Compare register on page 4-90) in the CRT Controller forces the output of the PEL Panning register to 0's until Vsync occurs. When Vsync occurs, the output returns to the programmed value. This allows the portion of the screen indicated by the Line Compare register to be operated on by the PEL Panning register.

- A write to the Character Map Select register becomes valid on the next whole character line. No deformed characters are displayed by changing character generators in the middle of a character scan line.

- For 256-color 320 x 200 graphics mode hex 13, the attribute controller is configured so that the 8-bit attribute stored in video memory for each PEL becomes the 8-bit address (P0 - P7) into the integrated DAC. The user should not modify the contents of the internal Palette registers when using this mode.

- The following sequence should be followed when accessing any of the Attribute Data registers pointed to by the Attribute Index register:

  1. Disable interrupts

  2. Reset read/write flip/flop

  3. Write to Index register

  4. Read from or write to a data register

  5. Enable interrupts.

- The Color Select register in the Attribute Controller section may be used to rapidly switch between sets of colors in the video DAC. When bit 7 of the Attribute Mode Control register equals 0, the 8-bit color value presented to the video DAC is composed of 6 bits from the internal Palette registers and bits 2 and 3 from the Color Select register. When bit 7 of the Attribute Mode Control register equals 1, the 8-bit color value presented to the video DAC is composed of the lower four bits from the internal Palette registers and the four bits in the Color Select register. By changing the value in the Color Select register, software rapidly switches between sets of colors in the video DAC. Note that BIOS does not support multiple sets of colors in the video DAC. The user must load these colors if this function is to be used. Also see the Attribute Controller block diagram on page 4-26. Note that the above discussion applies to all modes except 256 Color Graphics mode. In this mode the Color Select register is not used to switch between sets of colors.

- An application that saves the "Video State" must store the 4 bytes of information contained in the system microprocessor latches in the graphics controller subsection. These latches are loaded with 32 bits from video memory (8 bits per map) each time the system microprocessor does a read from video memory. The application needs to:

  1. Use write mode 1 to write the values in the latches to a location in video memory that is not part of the display buffer. The last location in the address range is a good choice.

  2. Save the values of the latches by reading them back from video memory.

  Note: If in a chain 4 or odd/even mode, it will be necessary to reconfigure the memory organization as four sequential maps prior to performing the sequence above. BIOS provides support for completely saving and restoring video state. See the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for more information.

- The description of the Horizontal PEL Panning register includes a figure showing the number of PELs shifted left for each valid value of the PEL Panning register and each valid video mode. Further panning beyond that shown in the figure may be accomplished by changing the start address in the CRT Controller

registers, Start Address High and Start Address Low. The sequence involved in further panning would be as follows:

1. Use the PEL Panning register to shift the maximum number of bits to the left. See Figure 4-103 on page 4-106 for the appropriate values.
2. Increment the start address.
3. If you are not using Modes 0+, 1+, 2+, 3+,7, or 7+, set the PEL Panning register to 0. If you are using these modes, set the PEL Panning register to 8. The screen will now be shifted one PEL left of the position it was in at the end of step 1. Step 1 through Step 3 may be repeated as desired.

- The Line Compare register (CRTC register hex 18) should be programmed with even values in 200 line modes when used in split screen applications that scroll a second screen on top of a first screen. This is a requirement imposed by the scan doubling logic in the CRTC.
- If the Cursor Start register (CRTC register hex 0A) is programmed with a value greater than that in the Cursor End register (CRTC register hex 0B), then no cursor is displayed. A split cursor is not possible.
- In 8-dot character modes, the underline attribute produces a solid line across adjacent characters, as in the IBM Color/Graphics Monitor Adapter, Monochrome Display Adapter and the Enhanced Graphics Adapter. In 9-dot modes, the underline across adjacent characters is dashed, as in the IBM 327X display terminals. In 9-dot modes, the line graphics characters (C0 - DF character codes) have solid underlines.

Programming the Registers

Each of the video subsections has an Index or Address register and a number of data registers. The Index or Address register serves as a pointer to the other registers on the device. These registers are loaded by the system microprocessor by executing an Out instruction to its I/O address with the index of the selected data register.

The data registers on each subsection are accessed through a common I/O address. They are distinguished by the pointer (index) in the Address register. To write to a data register, the Address register is loaded with the index of the appropriate data register, then the

selected data register is loaded by executing an Out instruction to the common I/O address.

The general registers are not accessed through an address register; they are written to directly.

See "Video DAC/System Microprocessor Interface" on page 4-116, for details on accessing the video DAC.

For compatibility with the IBM Enhanced Graphics Adapter (EGA), the internal VGA palette is programmed the same as the EGA. The video DAC is programmed by BIOS so that the compatible values in the internal VGA palette produce a color compatible with what was produced by EGA. Mode hex 13 (256 colors) is programmed so that the first 16 locations in the DAC produce compatible colors.

Summing - When BIOS is used to load the video DAC palette for a color mode and a monochrome display is connected to the system unit, the color palette is changed. The colors are summed to produce shades of gray that allow color applications to produce a readable screen.

There are 4 bits that should not be modified unless the sequencer is reset by setting bit 1 of the Reset register to 0. These bits are:

- Bit 3, or bit 0 of the Clocking Mode register.
- Bit 3, or bit 2 of the Miscellaneous Output register.

RAM Loadable Character Generator

The character generator is RAM loadable and can support characters up to 32 scan lines high. Three character generators are stored within the BIOS and one is automatically loaded into the RAM by the BIOS when an alphanumeric mode is selected. The Character Map Select register can be programmed to define the function of bit 3 of the attribute byte to be a character generator switch. This allows the user to select between any two character sets residing in map 2. This effectively gives the user access to 512 characters instead of 256. Character tables may be loaded offline. Up to eight tables can be loaded.

The structure of the character tables is described in the following figure. The character generator is in map 2 and must be protected using the map mask function.

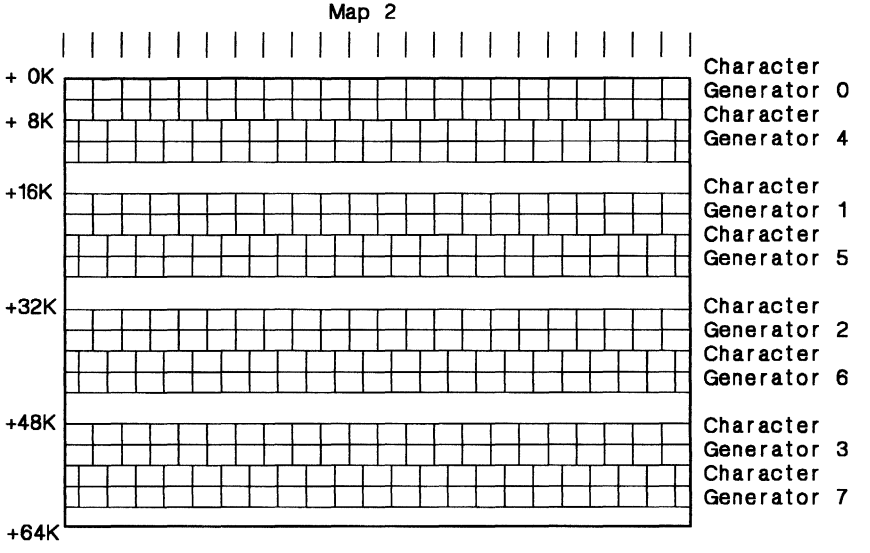


Figure 4-106. Character Table Structure

The following figure illustrates the structure of each character pattern. If the CRT controller is programmed to generate *n* row scans, then *n* bytes must be filled in for each character in the character generator. The example assumes eight row scans per character.

Address	Byte Image								Data
CC * 32 + 0				X	X				18H
1			X	X	X	X			3EH
2		X	X			X	X		66H
3		X	X			X	X		66H
4		X	X	X	X	X	X		7EH
5		X	X			X	X		66H
6		X	X			X	X		66H
7		X	X			X	X		66H

Figure 4-107. Character Pattern Example

CC equals the value of the character code. For example, hex 41 equals an ASCII "A."

Creating a Split Screen

The Video Graphics Array hardware supports a dual screen display. The top portion of the screen is designated as screen A, and the bottom portion of the screen is designated as screen B as in the following figure.

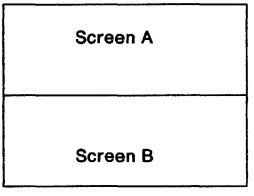


Figure 4-108. Dual Screen Definition

The following figure shows the screen mapping for a system containing a 32K byte alphanumeric storage buffer. Note that the Video Graphics Array has a 32K byte storage buffer in alphanumeric mode. Information displayed on screen A is defined by the Start Address High and Low registers (0CH and 0DH) of the CRTC. Information displayed on screen B always begins at address hex 0000.

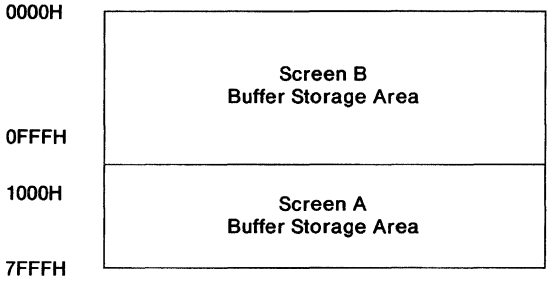


Figure 4-109. Screen Mapping within the Display Buffer Address Space

The Line Compare register (18H) of the CRT Controller performs the split screen function. The CRTC has an internal horizontal scan line counter. The CRTC also has logic that compares the horizontal scan line counter value to the Line Compare register value and clears the memory address generator when a compare occurs. The linear address generator then sequentially addresses the display buffer starting at location 0, and each subsequent row address is determined by the 16-bit addition of the start of line latch and the Offset register.

Screen B can be smoothly scrolled onto the CRT screen by updating the Line Compare in synchronization with the vertical retrace signal. The information on screen B is not affected by scrolling operations that utilize the Start Address High and Low registers to scroll through the Screen A address map.

When bit 5 of the Attribute Mode Control register equals 1, a successful line compare forces the output of the PEL Panning register to 0 until vertical synchronization occurs. When Vsync occurs, the output returns to its programmed value. This feature allows the information on screen B to remain unaffected by PEL-panning operations on screen A.

Video Digital-to-Analog Converter (Video DAC)

The video digital-to-analog converter (DAC) integrates the function of a color look-up table with three internal DACs for driving an analog display.

The size of the color look-up table is 256 by 18 bits to allow the display of 256 colors from a palette of 256K possible colors. Each RGB analog output is driven by a 6-bit DAC. Each register in the color look-up table contains 6 bits each for the red, green, and blue DACs.

Video Digital to Analog Converter (DAC) Addresses in Hex

PEL Address (Write mode)	RW	03C8
PEL Address (Read mode)	WO	03C7
DAC State Register	RO	03C7
PEL Data Register	RW	03C9
PEL Mask *	RW	03C6

RO = Read Only, RW = Read / Write, WO = Write Only.

* This register must not be written to by application code or destruction of the color look-up table may occur.
See also "Video DAC Programming Considerations" on page 4-117.

Figure 4-110. Video DAC I/O Address Usage

Device Operation

The PEL address inputs (P0 - P7) and the blanking input are sampled on the rising edge of the PEL clock. After three further rising edges of the PEL clock, the analog outputs reflect the state of these inputs.

During normal operation the PEL address inputs (P0 - P7) are used as a pointer to one of the 256 internal registers (color look-up table). The value in each register is then, in turn, converted to an analog signal for each of the three analog outputs (red, green, blue). The blanking input can also be used to force the analog outputs to 0 volts. The blanking operation is independent of the state of the PEL address inputs.

During system microprocessor accesses, the 8-bit PEL Address register acts as a pointer to the 256 internal registers. Each internal register is 18 bits wide; 6 bits each for red, green, and blue. The internal registers are accessible through the system microprocessor interface as described below.

The system microprocessor interface is asynchronous with the video path. The timing of this interface is controlled by the 'write enable' and 'read enable' signals.

Video DAC/System Microprocessor Interface

The PEL Address register holds an 8-bit value that is used to address a location within the color look-up table. The PEL Address register may be written to at two different addresses to establish a read or write mode respectively. Once the PEL Address register has been written to and an access has been made to a location in the color-look up table, the PEL Address register automatically increments and further accesses may occur to successive locations.

Each time the PEL Address register is written to at address hex 03C8 it identifies that a write sequence will occur. A write sequence consists of three successive byte writes to the Data register at address hex 03C9. The least-significant 6 bits of each byte are concatenated to form the value placed in the 18-bit Data register. The order is red byte first, then green, and finally blue. Once the third byte has been written, the value in the Data register is written to the location pointed to by the PEL Address register. The order of events for a write cycle is:

1. Write to the PEL Address register at hex 03C8.
2. Three bytes are written to the Data register at hex 03C9.
3. The contents of the Data register are transferred to the location in the color look-up table pointed to by the PEL Address register.
4. The PEL Address register auto-increments by 1.
5. Go to step 2.

Each time the PEL Address register is written to at address hex 03C7 it identifies that a read sequence will occur. A read sequence consists of three successive byte reads from the Data register at address hex 03C9. The least significant 6 bits of each byte taken from

the Data register contain the corresponding color value. The order is red byte first, then green, and finally blue. The order of events for a read cycle is:

1. Write to the PEL Address register at hex 03C7.
2. The contents of the location in the color look-up table pointed to by the PEL Address register are transferred to the Data register.
3. The PEL Address register auto-increments by one.
4. Three bytes are read back from the Data register at hex 03C9.
5. Go to step 2.

If the PEL Address register is written to during either a read or write cycle, a mode is initialized and the unfinished cycle is aborted. The effects of writing to the Data register during a read cycle or reading from the Data register during a write cycle are undefined and may change the look-up table contents.

A read from address hex 03C7 returns 0's in bit positions 0 and 1, if the DAC is currently in a read mode. A read from address hex 03C7 returns ones in bit positions 0 and 1, if the DAC is currently in a write mode.

Reads from the PEL Address register at hex 03C8 or the State register at hex 03C7 do not interfere with read or write cycles and may take place at any time.

Video DAC Programming Considerations

1. As explained in "Video DAC/System Microprocessor Interface" on page 4-116, the effects of writing to the Data register during a read cycle or reading from the Data register during a write cycle are undefined and may change the look-up table contents. Therefore, the following sequence must be followed to ensure the color look-up table integrity during accesses to it:
   a. Write out address to the PEL Address register
   b. Disable Interrupts
   c. Write or read three bytes of data
   d. Go to step C. Repeat this step for the desired number of locations.

e. Enable interrupts.

Note: The above sequence assumes that any interrupting process will return the DAC in the correct mode (write or read). If this is not the case, the sequence shown below should be followed:

a. Disable interrupts

b. Write out address to PEL Address register

c. Write or read three bytes of data

d. Go to step b. Repeat this step for the desired number of locations.

e. Enable interrupts.

2. There is a timing requirement on the minimum amount of time that must separate the trailing edge of one Read or Write command to the DAC and the leading edge of the next Read or Write command. The minimum separation is 240 nanoseconds. Software must ensure that the 240-nanosecond separation exists between two successive accesses to the DAC. Assembly language programs can meet this requirement by placing a JMP instruction between successive accesses to the DAC.

3. To prevent "snow" on the screen, an application reading data from or writing data to the DAC's Data register should ensure that the BLANK input to the DAC is asserted. This can be accomplished either by restricting data transfers to retrace intervals (use Input Status register 1 to determine when retrace is occurring) or by using the Screen Off bit located in the Clocking Mode register of the Sequencer subsection.

   Note: BIOS provides read and write interfaces to the Video DAC.

4. The Mask register (address hex 03C6) must not be written to by application code, or destruction of the color look-up table may result. This register is correctly initialized to hex FF by BIOS during a video mode set.

Auxiliary Video Connector

The Auxiliary Video connector is a 20-pin connector located in line with one of the channel connectors on the system board. This connector allows video data to be passed to an adapter. The system board video buffers can be turned off and video from the adapter can drive the video DAC and the 15-pin video output connector. The full channel is available for use by the adapter. Video cannot be passed in both directions at the same time. The DCLK signal cannot be used to drive both the EXTCLK VGA input and the PCLK DAC input.

For auxiliary video connector signal descriptions, electrical specifications, and connector pin descriptions see Section 2, "Micro Channel Architecture" on page 2-1. Timing diagrams for the signals that interface with the video DAC are contained on page 2-108.

A functional block diagram of the auxiliary video connector and timing diagrams for the display control signals appear on the following pages.

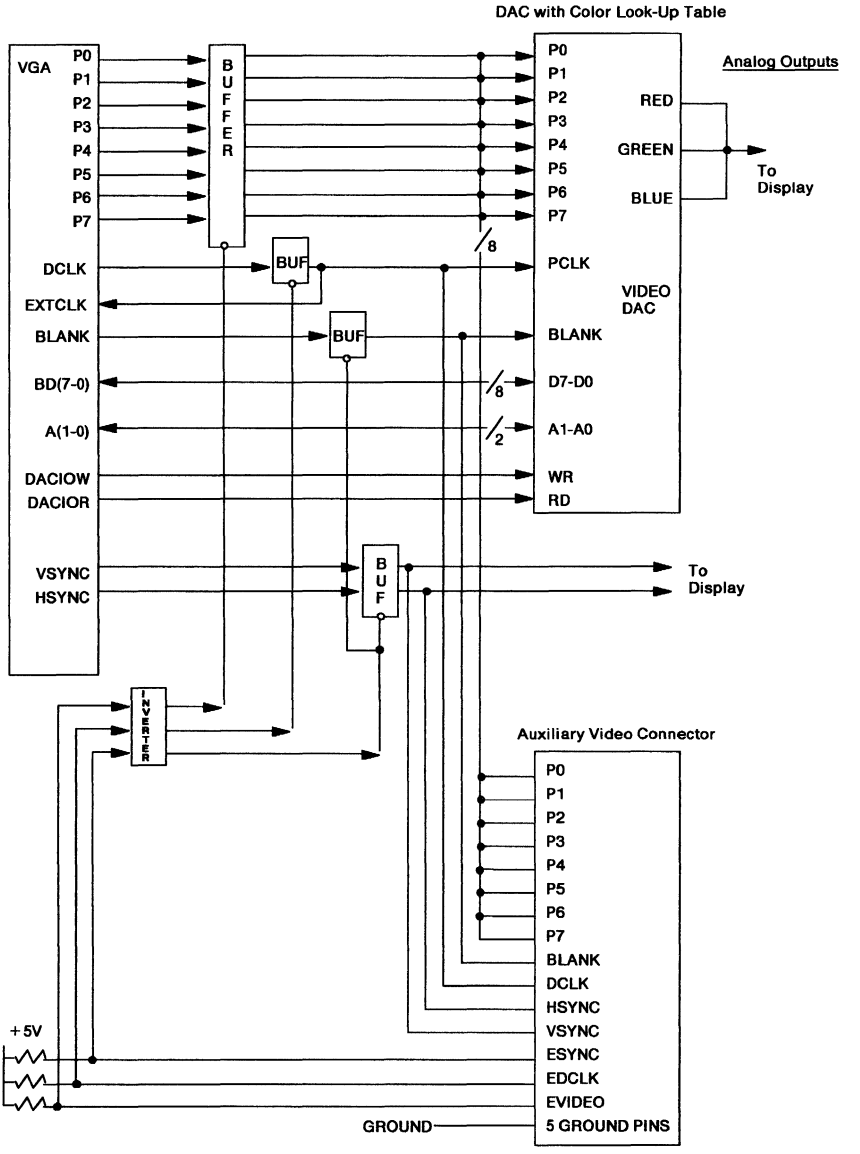


Figure 4-111. Auxiliary Video Connector Block Diagram

15-Pin Display Connector Timing (Sync Signals)

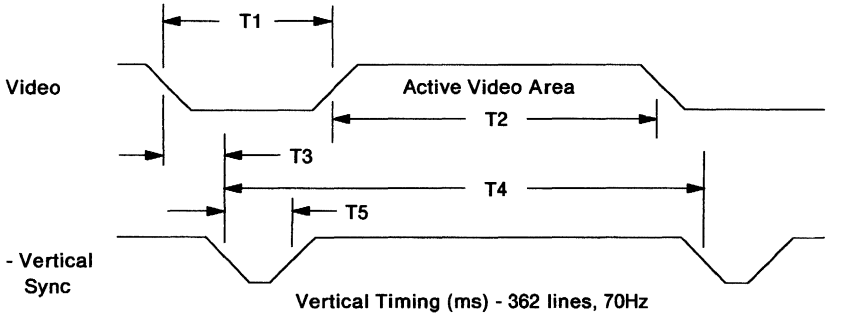
BIOS sets the VGA registers to generate the video modes. The video modes are shown in Figure 4-14 on page 4-27. All of these modes are 70 Hz vertical retrace except for modes 11 and 12. These two modes are 60 Hz vertical retrace. The VGA generates timings that are within the specifications for the supported displays using these modes.

The analog displays operate from 50 to 70 Hz vertical retrace frequency. The following timing diagrams represent only the vertical frequencies set by BIOS.

Note: The vertical size of the display is encoded using the polarity of the sync signals as shown in the following figure. See "Miscellaneous Output Register" on page 4-59 for more information.

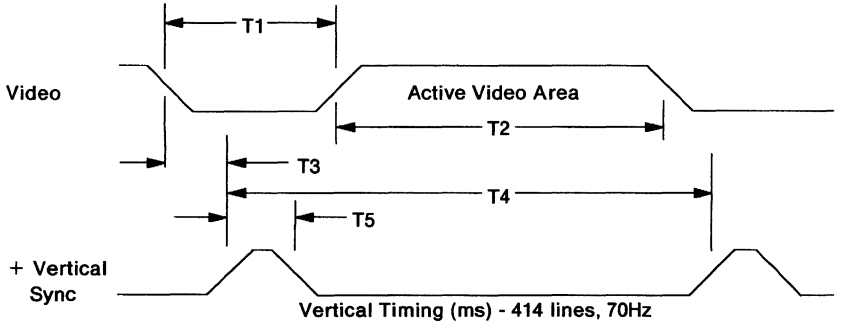
VSYNC Polarity	HSYNC Polarity	Vertical Size
+	+	Reserved
-	+	400 lines
+	-	350 lines
-	-	480 lines

Figure 4-112. Display Vertical Size



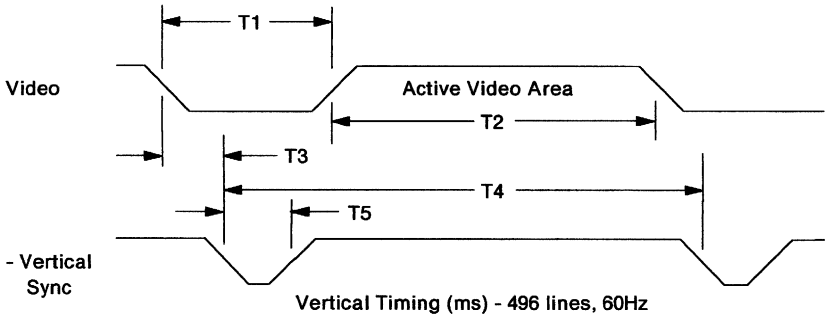
	Signal Time
T1	2.765 milliseconds
T2	11.504 milliseconds
T3	0.985 milliseconds
T4	14.268 milliseconds
T5	0.064 milliseconds

Figure 4-113. Display Vertical Sync, 350 Lines



	Signal Time
T1	1.112 milliseconds
T2	13.156 milliseconds
T3	0.159 milliseconds
T4	14.268 milliseconds
T5	0.064 milliseconds

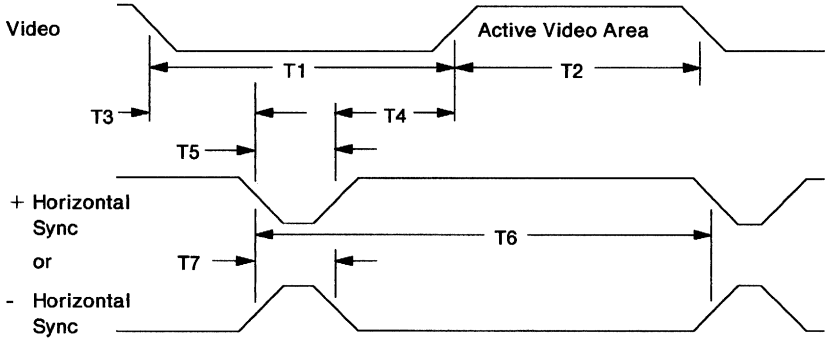
Figure 4-114. Display Vertical Sync, 400 Lines



Vertical Timing (ms) - 496 lines, 60Hz

	Signal Time
T1	0.922 milliseconds
T2	15.762 milliseconds
T3	0.064 milliseconds
T4	16.683 milliseconds
T5	0.064 milliseconds

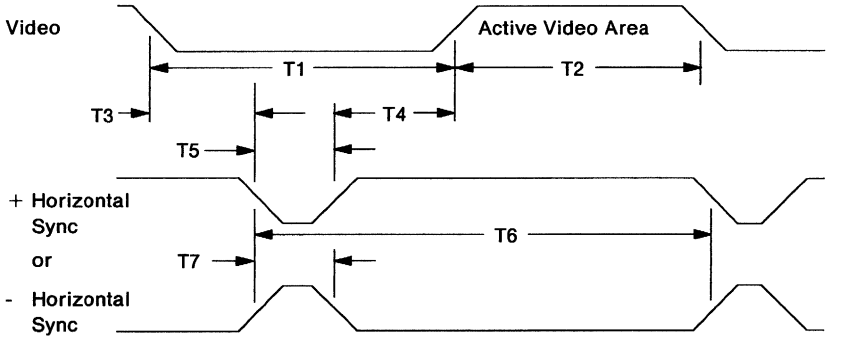
Figure 4-115. Display Vertical Sync, 480 Lines



Horizontal Timing (μs) – 80 column with border

	Signal Time
T1	5.720 microseconds
T2	26.058 microseconds
T3	0.318 microseconds
T4	1.589 microseconds
T5	3.813 microseconds
T6	31.778 microseconds
T7	3.813 microseconds

Figure 4-116. Display Horizontal Timing, 80 Column with Border



Horizontal Timing (μs) – 40/80 column, no border

	Signal Time
T1	6.356 microseconds
T2	25.422 microseconds
T3	0.636 microseconds
T4	1.907 microseconds
T5	3.813 microseconds
T6	31.778 microseconds
T7	3.813 microseconds

Figure 4-117. Display Horizontal Timing, 40/80 Column, no Border

Display Connector

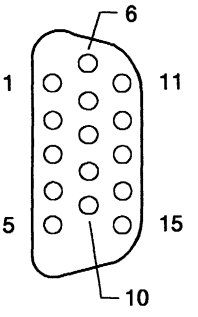


Figure 4-118. 15-Pin D-Shell Display Connector

			Display Type	
Pin	**I/O**	**Output**	**Monochrome**	**Color**
1	O	Red	No Pin	Red
2	O	Green	Mono	Green
3	O	Blue	No Pin	Blue
4	NA	Reserved	No Pin	No Pin
5	NA	Digital G	Self Test	Self Test
6	NA	Red Rtn	Dummy Pin	Red Rtn
7	NA	Green Rtn	Mono Rtn	Green Rtn
8	NA	Blue Rtn	No Pin	Blue Rtn
9	NA	Plug	No Pin	No Pin
10	NA	Digital G	Digital G	Digital G
11	NA	Reserved	No Pin	Digital G
12	NA	Reserved	Digital G	No Pin
13	O	Hsync	Hsync	Hsync
14	O	Vsync	Vsync	Vsync
15	NA	Reserved	No Pin	No Pin

Red Rtn, Green Rtn, Blue Rtn = Analog Grounds
Digital G = digital ground for sync returns and self test.

Figure 4-119. 15-Pin D-Shell Display Connector Signals

Diskette Drive Controller

The diskette drive controller and connector reside on the system board. A cable is used to connect the diskette drives to the system board.

Note: This section is included as a guide to compatibility for existing software. New software should access the diskette drive controller through BIOS.

The diskette drive controller supports:

- Two drives
- 1M unformatted media, 720K formatted
- 2M unformatted media, 1.44M formatted
- 250,000 bits-per-second mode
- 500,000 bits-per-second mode.

Precompensation of 125 nanoseconds is provided for all cylinders.

The diskette drive controller reads and writes both high- and low-density media. The 720K and 1.44M formatted diskette densities are supported for single and multithread operations.

Warning: 32-bit operations to the video subsystem can cause a diskette overrun in the 1.44M mode because data width conversions may require more than 12 microseconds. If an overrun occurs, BIOS returns an error code and the operation should be retried.

When switching from one density to another, the following changes occur:

- The clock rate changes.
  - Eight MHz for high density
  - Four MHz for low density
- The step rate changes.

Warning: The diskette controller does not check if the media supports the density selected. 1M media may not be reliably formatted to the 2M density and loss of data may result.

Registers

The diskette drive controller has five registers; three registers show the status of signals used in diskette operations, and two registers control certain interface signals. Two additional registers are accessed by the system microprocessor: the Diskette Drive Controller Status register and the Data register.

Status Register A: Status Register A, hex address 03F0, is a read-only register that shows the status of the corresponding signals.

Bit	Function
7	Interrupt Pending
6	-2nd Drive Installed
5	Step
4	-Track 0
3	Head 1 Select
2	-Index
1	-Write Protect
0	Direction

Figure 4-120. Status Register A (Hex 03F0)

Status Register B: Status Register B, address hex 03F1, is a read-only register that shows the status of signals between the diskette drive and the controller.

Bit	Function
7, 6	Reserved
5	Drive Select
4	Write Data (A positive transition on WR DATA causes this bit to toggle.)
3	Read Data (A positive transition on -RD DATA causes this bit to toggle.)
2	Write Enable
1	Motor Enable 1 *
0	Motor Enable 0 *

* In the Model 80, if either bit 0 or bit 1 is active, both 'motor enable' signals are active. This eliminates motor start time when switching from drive to drive.

Figure 4-121. Status Register B (Hex 03F1)

Digital Output Register: The Digital Output register, address hex 03F2, is write-only and used to control drive motors, drive selection, and feature enable. All bits are cleared by a Reset.

Bit	Function
7, 6	Reserved
5	Motor Enable 1 *
4	Motor Enable 0 *
3	Reserved
2	-8272A Reset
1	Reserved
0	Drive Select (0 = Drive 0, 1 = Drive 1) *

* The Motor Enable bit is qualified by the Drive Select bit.

Figure 4-122. Digital Output Register (Hex 03F2)

Digital Input Register: The Digital Input register, address hex 03F7, is read-only and used to sense the state of the '-diskette change' signal and '-high density select' signal.

Bit	Function
7	Diskette Change
6 - 1	Reserved
0	-High Density Select

Figure 4-123. Digital Input Register (Hex 03F7)

Configuration Control Register: The Configuration Control register, address hex 03F7, is write-only and used to set the transfer rate.

Bit	Function
7 - 2	Reserved
1, 0	DRC1, DRC0 00 = 500,000-bit mode 01 = Reserved 10 = 250,000-bit mode 11 = Reserved

Figure 4-124. Configuration Register (Hex 03F7)

Diskette Drive Controller Status Register: This is a read-only register and is used to facilitate the transfer of data between the system microprocessor and the controller.

Bit	Function
7	Request for Master
6	Data Input/Output
5	Non-DMA Mode
4	Diskette Controller Busy
3	Reserved
2	Reserved
1	Drive 1 Busy
0	Drive 0 Busy

Figure 4-125. Diskette Drive Controller Status Register (Hex 03F4)

Bit 7 When this bit is set to 1, the Data register is ready for transfer with the system microprocessor.

Bit 6 This bit indicates the direction of data transfer between the diskette drive controller and the system microprocessor. If this bit is set to 1, the transfer is from the controller to the system microprocessor; if it is set to 0, the transfer is from the microprocessor.

Bit 5 When this bit is set to 1, the controller is in the non-DMA mode.

Bit 4 When this bit is set to 1, a Read or Write command is being executed.

Bits 3, 2 Reserved

Bit 1 Drive 1 Busy - When set to 1, diskette drive 1 is in the seek mode.

Bit 0 Drive 0 Busy - When set to 1, diskette drive 0 is in the seek mode.

Data Registers, Hex 03F5: Address hex 03F5, consists of several registers in a stack with only one register presented to the data bus at a time. It stores data, commands, and parameters, and provides diskette-drive status information. Data bytes are passed through the Data register to program or obtain results after a command.

Diskette Drive Controller Programming Considerations

Each command is initiated by a multibyte transfer from the system microprocessor, and the result can also be a multibyte transfer back to the system microprocessor. Because of this multibyte interchange of information between the controller and the microprocessor, each command is considered to consist of three phases:

Command Phase: The system microprocessor issues a series of Writes to the controller that direct it to perform a specific operation.

Execution Phase: The controller performs the specified operation.

Result Phase: After completion of the operation, status and other housekeeping information is made available to the system microprocessor through a sequence of Read commands to the system microprocessor.

The following is a summary of the commands that are issued to the diskette drive controller.

- Read Data command
- Read Deleted Data command
- Read a Track command
- Read ID command
- Write Data command
- Write Deleted Data command
- Format a Track command
- Scan Equal command
- Scan Low or Equal command
- Scan High or Equal command
- Recalibrate command
- Sense Interrupt Status command
- Specify command
- Sense Drive Status command
- Seek command.

The following figure shows the symbols used in the format of the individual commands. The individual command formats start on page 4-133.

Symbol	Name	Description
C	Cylinder Number	C contains the current or selected cylinder number in binary notation.
D	Data	D contains the data pattern to be written to a sector.
D7-D0	Data Bus	An 8-bit data bus in which D7 is the most-significant bit and D0 is the least significant.
DTL	Data Length	When N is 00, DTL is the data length to be read from or written to a sector.
EOT	End of Track	The final sector number on a cylinder.
GPL	Gap Length	The length of gap 3 (spacing between sectors excluding the voltage-controlled oscillator synchronous field).
H	Head Address	The head number, either 0 or 1, as specified in the ID field.
HD	Head	The selected head number, 0 or 1. (H=HD in all command words.)
HLT	Head Load Time	The head load time in the selected drive (2 to 254 milliseconds in 2-millisecond increments).
HUT	Head Unload Time	The head unload time after a Read or Write operation (16 to 240 milliseconds in 16-millisecond increments).
MF	FM or MFM Mode	A 0 selects FM mode and a 1 selects MFM.
MT	Multitrack	A 1 selects multitrack operation. (Both HD0 and HD1 will be read or written.)

Figure 4-126 (Part 1 of 2). Command Symbols, Diskette Drive Controller

Symbol	Name	Description
N	Number	The number of data bytes written in a sector.
NCN	New Cylinder Number	The new cylinder number for a seek operation.
ND	Non-DMA Mode	This indicates an operation in the non-DMA mode.
PCN	Present Cylinder Number	The cylinder number at the completion of a Sense Interrupt Status command (present position of the head).
R	Record	The sector number to be read or written.
SC	Sector	The number of sectors per cylinder.
SK	Skip	This stands for skip deleted-data address mark.
SRT	Step Rate	The stepping rate for the diskette drive (1 to 16 milliseconds in 1-millisecond increments).
ST 0 - ST 3	Status 0-Status 3	Four registers that store status information after a command is executed.
STP	Scan Test	If STP is 1, the data in contiguous sectors is compared with the data sent by the system microprocessor during a scan operation. If STP is 2, then alternate sectors are read and compared.
US0 - US1	Unit Select	The selected drive number encoded the same as bits 0 and 1 of the Digital Output register.

Figure 4-126 (Part 2 of 2). Command Symbols, Diskette Drive Controller

The following are commands that are issued to the controller. An X is used to indicate a don't care condition.

Read Data Command Format

Command Phase

MT = Multitrack
MF = MFM mode
SK = Skip deleted-data address mark
HD = Head number
USx = Unit select

	7	6	5	4	3	2	1	0
Byte 0	MT	MF	SK	0	0	1	1	0
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number (C) Bits 7 - 0							
Byte 3	Head Address (H) Bits 7 - 0							
Byte 4	Sector Number (R) Bits 7 - 0							
Byte 5	Number of Data Bytes in Sector (N) Bits 7 - 0							
Byte 6	End of Track (EOT) Bits 7 - 0							
Byte 7	Gap Length (GPL) Bits 7 - 0							
Byte 8	Data Length (DTL) Bits 7 - 0							

Figure 4-127. Read Data Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0 (ST 0) Bits 7 - 0							
Byte 1	Status Register 1 (ST 1) Bits 7 - 0							
Byte 2	Status Register 2 (ST 2) Bits 7 - 0							
Byte 3	Cylinder Number (C) Bits 7 - 0							
Byte 4	Head Address (H) Bits 7 - 0							
Byte 5	Sector Number (R) Bits 7 - 0							
Byte 6	Number of Data Bytes in Sector (N) Bits 7 - 0							

Figure 4-128. Read Data Result

Read Deleted Data Command Format

Command Phase

MT = Multitrack
MF = MFM mode
SK = Skip deleted-data address mark
HD = Head number
USx = Unit select

	7	6	5	4	3	2	1	0
Byte 0	MT	MF	SK	0	1	1	0	0
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number (C) Bits 7 - 0							
Byte 3	Head Address (H) Bits 7 - 0							
Byte 4	Sector Number (R) Bits 7 - 0							
Byte 5	Number of Data Bytes in Sector (N) Bits 7 - 0							
Byte 6	End of Track (EOT) Bits 7 - 0							
Byte 7	Gap Length (GPL) Bits 7 - 0							
Byte 8	Data Length (DTL) Bits 7 - 0							

Figure 4-129. Read Deleted Data Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0 (ST 0) Bits 7 - 0							
Byte 1	Status Register 1 (ST 1) Bits 7 - 0							
Byte 2	Status Register 2 (ST 2) Bits 7 - 0							
Byte 3	Cylinder Number (C) Bits 7 - 0							
Byte 4	Head Address (H) Bits 7 - 0							
Byte 5	Sector Number (R) Bits 7 - 0							
Byte 6	Number of Data Bytes in Sector (N) Bits 7 - 0							

Figure 4-130. Read Deleted Data Result

Read a Track Command Format

Command Phase

MF = MFM mode
SK = Skip deleted-data address mark
HD = Head number
USx = Unit select

	7	6	5	4	3	2	1	0
Byte 0	0	MF	SK	0	0	0	1	0
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number (C) Bits 7 - 0							
Byte 3	Head Address (H) Bits 7 - 0							
Byte 4	Sector Number (R) Bits 7 - 0							
Byte 5	Number of Data Bytes in Sector (N) Bits 7 - 0							
Byte 6	End of Track (EOT) Bits 7 - 0							
Byte 7	Gap Length (GPL) Bits 7 - 0							
Byte 8	Data Length (DTL) Bits 7 - 0							

Figure 4-131. Read a Track Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0 (ST 0) Bits 7 - 0							
Byte 1	Status Register 1 (ST 1) Bits 7 - 0							
Byte 2	Status Register 2 (ST 2) Bits 7 - 0							
Byte 3	Cylinder Number (C) Bits 7 - 0							
Byte 4	Head Address (H) Bits 7 - 0							
Byte 5	Sector Number (R) Bits 7 - 0							
Byte 6	Number of Data Bytes in Sector (N) Bits 7 - 0							

Figure 4-132. Read a Track Result

Read ID Command Format

Command Phase

MF = MFM mode
HD = Head number
USx = Unit select

	7	6	5	4	3	2	1	0
Byte 0	0	MF	0	0	1	0	1	0
Byte 1	X	X	X	X	X	HD	US1	US0

Figure 4-133. Read ID Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0 (ST 0) Bits 7 - 0							
Byte 1	Status Register 1 (ST 1) Bits 7 - 0							
Byte 2	Status Register 2 (ST 2) Bits 7 - 0							
Byte 3	Cylinder Number (C) Bits 7 - 0							
Byte 4	Head Address (H) Bits 7 - 0							
Byte 5	Sector Number (R) Bits 7 - 0							
Byte 6	Number of Data Bytes in Sector (N) Bits 7 - 0							

Figure 4-134. Read ID Result

Write Data Command Format

Command Phase

MT = Multitrack
MF = MFM mode
HD = Head number
USx = Unit select

	7	6	5	4	3	2	1	0
Byte 0	MT	MF	0	0	0	1	0	1
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number (C) Bits 7 - 0							
Byte 3	Head Address (H) Bits 7 - 0							
Byte 4	Sector Number (R) Bits 7 - 0							
Byte 5	Number of Data Bytes in Sector (N) Bits 7 - 0							
Byte 6	End of Track (EOT) Bits 7 - 0							
Byte 7	Gap Length (GPL) Bits 7 - 0							
Byte 8	Data Length (DTL) Bits 7 - 0							

Figure 4-135. Write Data Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0 (ST 0) Bits 7 - 0							
Byte 1	Status Register 1 (ST 1) Bits 7 - 0							
Byte 2	Status Register 2 (ST 2) Bits 7 - 0							
Byte 3	Cylinder Number (C) Bits 7 - 0							
Byte 4	Head Address (H) Bits 7 - 0							
Byte 5	Sector Number (R) Bits 7 - 0							
Byte 6	Number of Data Bytes in Sector (N) Bits 7 - 0							

Figure 4-136. Write Data Result

Write Deleted Data Command Format

Command Phase

MT = Multitrack
MF = MFM mode
HD = Head number
USx = Unit select

	7	6	5	4	3	2	1	0
Byte 0	MT	MF	0	0	1	0	0	1
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number (C) Bits 7 - 0							
Byte 3	Head Address (H) Bits 7 - 0							
Byte 4	Sector Number (R) Bits 7 - 0							
Byte 5	Number of Data Bytes in Sector (N) Bits 7 - 0							
Byte 6	End of Track (EOT) Bits 7 - 0							
Byte 7	Gap Length (GPL) Bits 7 - 0							
Byte 8	Data Length (DTL) Bits 7 - 0							

Figure 4-137. Write Deleted Data Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0 (ST 0) Bits 7 - 0							
Byte 1	Status Register 1 (ST 1) Bits 7 - 0							
Byte 2	Status Register 2 (ST 2) Bits 7 - 0							
Byte 3	Cylinder Number (C) Bits 7 - 0							
Byte 4	Head Address (H) Bits 7 - 0							
Byte 5	Sector Number (R) Bits 7 - 0							
Byte 6	Number of Data Bytes in Sector (N) Bits 7 - 0							

Figure 4-138. Write Deleted Data Result

Format a Track Command Format

Command Phase

MF = MFM mode
HD = Head number
USx = Unit select

	7	6	5	4	3	2	1	0
Byte 0	0	MF	0	0	1	1	0	0
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Number of Data Bytes in Sector (N) Bits 7 - 0							
Byte 3	Sectors per Cylinder (SC) Bits 7 - 0							
Byte 4	Gap Length (GPL) Bits 7 - 0							
Byte 5	Data (D) Bits 7 - 0							

Figure 4-139. Format a Track Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0 (ST 0) Bits 7 - 0							
Byte 1	Status Register 1 (ST 1) Bits 7 - 0							
Byte 2	Status Register 2 (ST 2) Bits 7 - 0							
Byte 3	Cylinder Number (C) Bits 7 - 0							
Byte 4	Head Address (H) Bits 7 - 0							
Byte 5	Sector Number (R) Bits 7 - 0							
Byte 6	Number of data bytes in sector (N) Bits 7 - 0							

Figure 4-140. Format a Track Result

Scan Equal Command Format

Command Phase

MT = Multitrack
MF = MFM mode
SK = Skip deleted-data address mark
HD = Head number
USx = Unit select

	7	6	5	4	3	2	1	0
Byte 0	MT	MF	SK	1	0	0	0	1
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number (C) Bits 7 - 0							
Byte 3	Head Address (H) Bits 7 - 0							
Byte 4	Sector Number (R) Bits 7 - 0							
Byte 5	Number of Data Bytes in Sector (N) Bits 7 - 0							
Byte 6	End of Track (EOT) Bits 7 - 0							
Byte 7	Gap Length (GPL) Bits 7 - 0							
Byte 8	Scan Test (STP) Bits 7 - 0							

Figure 4-141. Scan Equal Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0 (ST 0) Bits 7 - 0							
Byte 1	Status Register 1 (ST 1) Bits 7 - 0							
Byte 2	Status Register 2 (ST 2) Bits 7 - 0							
Byte 3	Cylinder Number (C) Bits 7 - 0							
Byte 4	Head Address (H) Bits 7 - 0							
Byte 5	Sector Number (R) Bits 7 - 0							
Byte 6	Number of Data Bytes in Sector (N) Bits 7 - 0							

Figure 4-142. Scan Equal Result

Scan Low or Equal Command Format

Command Phase

MT = Multitrack
MF = MFM mode
SK = Skip deleted-data address mark
HD = Head number
USx = Unit select

	7	6	5	4	3	2	1	0
Byte 0	MT	MF	SK	1	1	0	0	1
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number (C) Bits 7 - 0							
Byte 3	Head Address (H) Bits 7 - 0							
Byte 4	Sector Number (R) Bits 7 - 0							
Byte 5	Number of Data Bytes in Sector (N) Bits 7 - 0							
Byte 6	End of Track (EOT) Bits 7 - 0							
Byte 7	Gap Length (GPL) Bits 7 - 0							
Byte 8	Scan Test (STP) Bits 7 - 0							

Figure 4-143. Scan Low or Equal Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0 (ST 0) Bits 7 - 0							
Byte 1	Status Register 1 (ST 1) Bits 7 - 0							
Byte 2	Status Register 2 (ST 2) Bits 7 - 0							
Byte 3	Cylinder Number (C) Bits 7 - 0							
Byte 4	Head Address (H) Bits 7 - 0							
Byte 5	Sector Number (R) Bits 7 - 0							
Byte 6	Number of Data Bytes in Sector (N) Bits 7 - 0							

Figure 4-144. Scan Low or Equal Result

Scan High or Equal Command Format

Command Phase

MT = Multitrack
MF = MFM mode
SK = Skip deleted-data address mark
HD = Head number
USx = Unit select

	7	6	5	4	3	2	1	0
Byte 0	MT	MF	SK	1	1	1	0	1
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number (C) Bits 7 - 0							
Byte 3	Head Address (H) Bits 7 - 0							
Byte 4	Sector Number (R) Bits 7 - 0							
Byte 5	Number of Data Bytes in Sector (N) Bits 7 - 0							
Byte 6	End of Track (EOT) Bits 7 - 0							
Byte 7	Gap Length (GPL) Bits 7 - 0							
Byte 8	Scan Test (STP) Bits 7 - 0							

Figure 4-145. Scan High or Equal Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0 (ST 0) Bits 7 - 0							
Byte 1	Status Register 1 (ST 1) Bits 7 - 0							
Byte 2	Status Register 2 (ST 2) Bits 7 - 0							
Byte 3	Cylinder Number (C) Bits 7 - 0							
Byte 4	Head Address (H) Bits 7 - 0							
Byte 5	Sector Number (R) Bits 7 - 0							
Byte 6	Number of Data Bytes in Sector (N) Bits 7 - 0							

Figure 4-146. Scan High or Equal Result

Recalibrate Command Format

Command Phase

USx = Unit select

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	0	1	1	1
Byte 1	X	X	X	X	X	0	US1	US0

Figure 4-147. Recalibrate Command

Result Phase: This command has no result phase.

Sense Interrupt Status Command Format

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	1	0	0	0

Figure 4-148. Sense Interrupt Status Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0 (ST 0)							
Byte 1	Present Cylinder Number (PCN)							

Figure 4-149. Sense Interrupt Status Result

Specify Command Format

Command Phase

SRT = Diskette stepping rate
HUT = Head unload time
HLT = Head load time
ND = Non-data mode

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	0	0	1	1
Byte 1	SRT	SRT	SRT	SRT	HUT	HUT	HUT	HUT
Byte 2	HLT	HLT	HLT	HLT	HLT	HLT	HLT	ND

Figure 4-150. Specify Command

Result Phase: This command has no result phase.

Sense Drive Status Command Format

Command Phase

HD = Head number
USx = Unit select

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	0	1	0	0
Byte 1	X	X	X	X	X	HD	US1	US0

Figure 4-151. Sense Drive Status Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 3 (ST 3) Bits 7 - 0							

Figure 4-152. Sense Drive Status Result

Seek Command Format

Command Phase

USx = Unit select
HD = Head select

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	1	1	1	1
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	New Cylinder Number for Seek (NCN) Bits 7 - 0							

Figure 4-153. Seek Command

Result Phase: This command has no result phase.

Invalid Command Format

Result Phase: The following status byte is returned to the system microprocessor when an invalid command has been received.

	7	6	5	4	3	2	1	0
Byte 0	Status 0 Register (ST 0) Bits 7 - 0							

Figure 4-154. Invalid Command Result

Command Status Registers

The following are definitions of status registers ST 0 through ST 3.

Status Register 0: The following are bit definitions of status register ST 0.

Bit	Function
7, 6	Interrupt Code (IC) 00 = Normal Termination of Command (NT) - The command was completed and properly executed. 01 = Abrupt Termination of Command (AT) - The execution of the command was started but not successfully completed. 10 = Invalid Command Issue (IC) - The issued command was never started. 11 = Reserved
5	Seek End (SE) - Set to 1 when the diskette drive completes the Seek command.
4	Equipment Check (EC) - Set to 1 if the '-track 0' signal fails to occur after 77 step pulses (Recalibrate command).
3	Not Ready (NR) - This flag is set to 1 when the diskette drive is in the not-ready state and a Read or Write command is issued. It is also set if a Read or Write command is issued to side 1 of a single-sided diskette drive.
2	Head Address (HD) - Indicates the state of the head at interrupt.
1	Unit select 1 (US 1) - Indicates drive 1 is at interrupt.
0	Unit select 0 (US 0) - Indicates drive 0 is at interrupt.

Figure 4-155. Status Register 0 (ST 0)

Status Register 1: The following are bit definitions of status register ST 1.

Bit	Function
7	End of Cylinder (EN) - Set to 1 when the controller tries to gain access to a sector beyond the final sector of a cylinder.
6	Reserved - This bit is always set to 0.
5	Data Error (DE) - Set to 1 when the controller detects a CRC error in either the ID field or the data field.
4	Overrun (OR) - Set to 1 if the controller is not serviced by the main system within a certain time limit during data transfers.
3	Reserved - This bit is always set to 0.
2	No Data (ND) - Set to 1 if the controller cannot find the sector specified in the ID register during the execution of a Read Data, Write Deleted Data, or Scan command. This flag is also set to 1 if the controller cannot read the ID field without an error during the execution of a Read ID command or if the starting sector cannot be found during the execution of a Read Cylinder command.
1	Not Writable (NW) - Set to 1 if the controller detects a '-write-protect' signal from the diskette drive during execution of a Write Data, Write Deleted Data, or Format Cylinder command.
0	Missing Address Mark (MA) - Set to 1 if the controller cannot detect the ID address mark. At the same time, the MD of Status register 2 is set to 1.

Figure 4-156. Status Register 1 (ST 1)

Status Register 2: The following are bit definitions of status register ST 2.

Bit	Function
7	Reserved - This bit is always set to 0.
6	Control Mark (CM) - This flag is set to 1 if the controller encounters a sector that has a deleted data-address mark during execution of a Read Data or Scan command.
5	Data Error in Data Field (DD) - Set to 1 if the controller detects an error in the data.
4	Wrong Cylinder (WC) - This flag is related to ND (no data) bit. When the contents of C on the medium are different from that stored in the ID register, this flag is set.
3	Scan Equal Hit (SH) - Set to 1 if the contiguous sector data equals the processor data during the execution of a Scan command.
2	Scan Not Satisfied (SN) - Set to 1 if the controller cannot find a sector on the cylinder that meets the condition during a Scan command.
1	Bad Cylinder (BC) - Related to ND; when the contents of C on the medium are different from that stored in the ID register, and the contents of C is FF, this flag is set to 1.
0	Missing Address Mark in Data Field (MD) - Set to 1 if the controller cannot find a data address mark or a deleted data address mark when data is read from the medium.

Figure 4-157. Status Register 2 (ST 2)

Status Register 3: The following are bit definitions of status register ST 3.

Bit	Function
7	Reserved
6	Write Protect (WP) - Status of the '-write-protect' signal from the diskette drive.
5	Reserved
4	Track 0 (T0) - Status of the '-track 0' signal from the diskette drive.
3	Reserved
2	Head Address (HD) - Status of the '-head 1 select' signal to the diskette drive.
1, 0	Reserved

Figure 4-158. Status Register 3 (ST 3)

Signal Descriptions

The diskette drive controller interface signal sequences and timings are all compatible with the industry standard 5.25-inch diskette interface. All interface signals are TTL compatible at the driver/receivers both in the rise and fall times as well as the interface levels.

The following describes the interface to the diskette drive.

Output Signals

All output signals to the diskette drive operate between 5 Vdc and ground with the following definitions:

- The inactive level is 2.0 Vdc minimum.
- The active level is 0.8 Vdc maximum.

All inputs from the drive can sink 4.0 mA at the active (low) level.

- The inactive level is 3.7 Vdc minimum.
- The active level is 0.4 Vdc maximum.

The following are descriptions of the diskette drive controller output signals.

-HIGH DENSITY SELECT: When active, the 2M mode is selected. Diskettes are formatted with 18 sectors per track and a capacity of 1.44M. When inactive, the 1M mode is selected. Diskettes are formatted with 9 sectors per track and a capacity of 720K. This signal is not used by the 720K diskette drive.

-DRIVE SELECT 0 - 1: The drive select signals enable or disable all drive interface signals except -MOTOR ENABLE. When a drive select signal is active, the drive is enabled. When it is inactive, all controlled inputs are ignored, and all drive outputs are disabled.

-MOTOR ENABLE 0 - 1: When this signal is made active, along with the proper drive select signals, the spindle starts to turn. There must be a 500-millisecond delay after -MOTOR ENABLE 0 or -MOTOR ENABLE 1 becomes active before a read, write, or seek operation. When inactive, this signal causes the spindle motor to decelerate and stop.

-DIRECTION: When this signal is active, -STEP moves the heads toward the drive spindle. When this signal is inactive, -STEP moves the heads away from the drive spindle. This signal is stable for 1 microsecond before and 1 microsecond after the trailing edge of the '-step' pulse.

Note: After a direction change, a 15-millisecond delay is required before the next '-step' pulse.

-STEP: A 1-microsecond active pulse of this signal causes the read/write heads to move one track. The state of -DIRECTION at the trailing edge of -STEP determines the direction of motion.

Note: Before a read or write operation, a 15-millisecond seek settle time must be allowed.

-WRITE DATA: A 250-nanosecond pulse of this signal causes a bit to be written if -WRITE ENABLE is active. Data written on the diskette will have 125-nanosecond write-precompensation.

-WRITE ENABLE: When active, this signal enables the write current circuits, and -WRITE DATA controls the writing of information. Motor-start and head-settle times must be observed before this line becomes active.

-HEAD 1 SELECT: Making this signal active selects the upper head; otherwise the lower head is selected.

Input Signals

All inputs from the drive can sink 4.0 mA at the active (low) level.

- The inactive level is 3.7 Vdc minimum.
- The active level is 0.4 Vdc maximum.

-INDEX: When the drive senses the index, it generates an active pulse of at least 1 millisecond on this line. This signal is gated to the interface only when a diskette is in the drive.

-TRACK 0: This signal is active when the read/write head is on track 0. Track 0 is determined by a sensor, not a track counter.

The drive is able to seek to track 0 under control of the system even if there is no diskette inserted at the time. This is required so the system software can determine how many drives are attached to the system. Software selects each drive and attempts to recalibrate that drive to track 0. The track 0 indication is used to determine whether or not each drive is installed in the system.

-WRITE PROTECT: When active, this signal indicates that a diskette with an open write-protect window (write-protected diskette) is in the drive, and the drive will not write.

-READ DATA: Each bit detected provides a 250-nanosecond active pulse on this line for the 250,000 bit rate or a 125-nanosecond pulse for the 500,000 bit rate.

-DISKETTE CHANGE: This signal is active at power-on and latched inactive when a diskette is present, the drive is selected, and a step pulse occurs. This signal goes active when the diskette is removed from the drive. The presence of a diskette is determined by a media sensor.

Power Sequencing

The WRITE GATE signal is turned off and is kept off before power is switched on or off. The read/write heads return to track 0 when the system power is switched on.

Connector

The system board has a single 2-by-20 pin connector for the attachment of one or two diskette drives. Signals and data are transmitted to and from the drives through a 40-wire L-shaped cable.

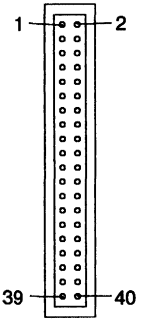


Figure 4-159. Diskette Drive Controller Connector

The following are the voltages and signals assigned to the diskette drive controller connector.

Pin No.	I/O	Signal Name	Pin No.	I/O	Signal Name
1	I	2nd Drive Installed	2	O	-High Density Select
3	NA	Ground	4	NA	Ground
5	NA	Ground	6	NA	Reserved
7	NA	Signal Ground	8	I	-Index
9	NA	Signal Ground	10	O	-Motor Enable 0
11	NA	Signal Ground	12	O	-Drive Select 1
13	NA	Ground	14	O	-Drive Select 0
15	NA	Signal Ground	16	O	-Motor Enable 1
17	NA	Signal Ground	18	O	-Direction
19	NA	Signal Ground	20	O	-Step
21	NA	Signal Ground	22	O	-Write Data
23	NA	Signal Ground	24	O	-Write Enable
25	NA	Signal Ground	26	I	-Track 0
27	NA	Signal Ground	28	I	-Write Protect
29	NA	Signal Ground	30	I	-Read Data
31	NA	Signal Ground	32	O	-Head 1 Select
33	NA	Signal Ground	34	I	-Diskette Change
35	NA	Ground	36	NA	Ground
37	NA	Ground	38	NA	+5 Vdc
39	NA	Ground	40	NA	+12 Vdc

Figure 4-160. Diskette Drive Controller Connector Voltage and Signal Assignments

Serial Port Controller

The serial port is controlled by a NS16550 serial communications controller. It is programmable and supports asynchronous communications. The controller automatically adds and removes start, stop, and parity bits. A programmable baud-rate generator allows operation from 50 baud to 19,200 baud. The port supports 5-, 6-, 7- and 8-bit characters with 1, 1.5, or 2 stop bits. A prioritized interrupt system controls transmit, receive, error, and line status as well as data-set interrupts.

The NS16550 controller is functionally compatible to the NS16450 controller. To programs, the NS16550 appears to be identical to the serial portion of the IBM Personal Computer AT Serial/Parallel adapter. Support for the controller is restricted to the functions which are identical to the NS16450. Using the controller in the FIFO mode may result in non-detectable data errors.

The serial port controller provides the following functions:

- Full double buffering in the character mode, eliminating the need for precise synchronization
- False-start bit detection
- Line-break generation and detection
- Modem control functions:
  - Clear to send (CTS)
  - Request to send (RTS)
  - Data set ready (DSR)
  - Data terminal ready (DTR)
  - Ring indicator (RI)
  - Data carrier detect (DCD).

The following figure is a block diagram of the serial port controller.

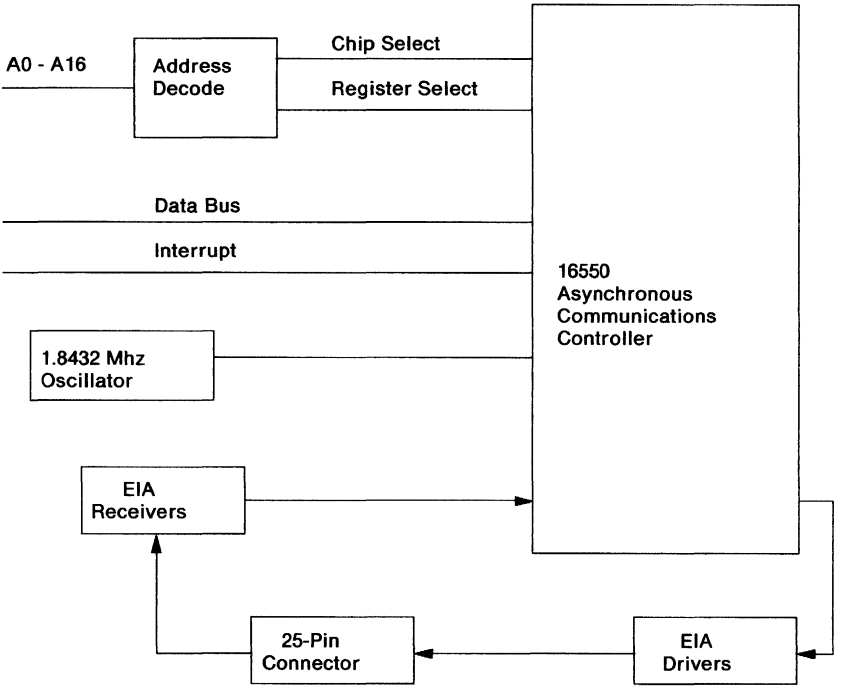


Figure 4-161. Serial Port Block Diagram

Communications Application

The serial output port can be addressed as either serial output port 1 (Serial 1) or serial output port 2 (Serial 2). In this section port addresses contain an **n**. The **n** can be either a 3 for Serial 1 or a 2 for Serial 2. The selection of either port is discussed in "System Board Setup" on page 2-32.

Two interrupt lines are provided to the system. Interrupt level 4 (IRQ4) is for Serial 1 and interrupt level 3 (IRQ3) is for Serial 2. To allow the controller to send interrupts to the interrupt controller, bit 3 of the Modem Control register must be set to 1. At this point, any interrupts allowed by the Interrupt Enable register will cause an interrupt.

The data format is as follows:

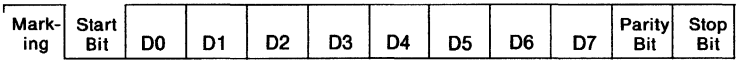


Figure 4-162. Serial Port, Data Format

Data bit 0 is the first bit to be sent or received. The controller automatically inserts the start bit, the correct parity bit (if programmed to do so), and the stop bits (1, 1.5, or 2 depending on the command in the Line Control register).

Programmable Baud-Rate Generator

The controller has a programmable baud-rate generator that can divide the clock input (1.8432 MHz) by any divisor from 1 to 65,535. The output frequency of the baud-rate generator is the baud rate multiplied by 16. Two 8-bit latches store the divisor in a 16-bit binary format. These divisor latches must be loaded during setup to ensure desired operation of the baud-rate generator. When either of the divisor latches is loaded, a 16-bit baud counter is immediately loaded. This prevents long counts on the first load.

Registers

The controller has a number of accessible registers. The system programmer may gain access to or control any of the controller registers through the system microprocessor. These registers are used to control the controller operations and to transmit and receive data.

Specific registers are selected according to the following figure:

DLAB State *	Port Address (hex)	R/W	Register
0	0nF8 **	W	Transmitter Holding Register **n**
0	0nF8 **	R	Receiver Buffer Register **n**
1	0nF8 **	R/W	Divisor Latch, Low Byte
1	0nF9 **	R/W	Divisor Latch, High Byte
0	0nF9 **	R/W	Interrupt Enable Register
X	0nFA **	R	Interrupt Identification Register
X	0nFA **	W	FIFO Control Register
X	0nFB **	R/W	Line Control Register
X	0nFC **	R/W	Modem Control Register
X	0nFD **	R	Line Status Register
X	0nFE **	R	Modem Status Register
X	0nFF **	R/W	Scratch Register

* The DLAB state is controlled by bit 7 of the Line Control register.
** The **n** determines the port selected; 3 is for Serial 1, and 2 is for Serial 2.

Figure 4-163. Serial Port Register Addresses

Transmitter Holding Register (hex nF8)

The Transmitter Holding register contains the character to be sent. Bit 0 is the least-significant bit and the first bit sent serially as shown in the following figure:

Bit	Function
7	Data Bit 7
6	Data Bit 6
5	Data Bit 5
4	Data Bit 4
3	Data Bit 3
2	Data Bit 2
1	Data Bit 1
0	Data Bit 0

Figure 4-164. Transmitter Holding Register (THR)

Receiver Buffer Register (hex nF8)

The Receiver Buffer register (RBR) contains the received character. Bit 0 is the least-significant bit and the first bit received serially as shown in the following figure:

Bit	Function
7	Data Bit 7
6	Data Bit 6
5	Data Bit 5
4	Data Bit 4
3	Data Bit 3
2	Data Bit 2
1	Data Bit 1
0	Data Bit 0

Figure 4-165. Receiver Buffer Register (RBR)

Divisor Latch Register LSB (hex nF8)

Bit	Function
7	Bit 7
6	Bit 6
5	Bit 5
4	Bit 4
3	Bit 3
2	Bit 2
1	Bit 1
0	Bit 0

Figure 4-166. Divisor Latch Register (LSB)

Divisor Latch Register MSB (hex nF9)

Bit	Function
7	Bit 7
6	Bit 6
5	Bit 5
4	Bit 4
3	Bit 3
2	Bit 2
1	Bit 1
0	Bit 0

Figure 4-167. Divisor Latch Register (MSB)

The Divisor Latch registers are used to program the baud rate generator. The values in these two registers form the divisor of the clock input (1.8432 MHz), which establishes the desired baud rate.

The following figure illustrates the use of the baud-rate generator with a frequency of 1.8432 MHz. For baud rates of 19,200 and below, the error obtained is minimal.

Note: In no case should the data speed be greater than 19,200 baud.

Desired Baud Rate	Divisor Used to Generate 16x Clock (Decimal)	(Hex)	Percent of Error Difference Between Desired and Actual
50	2304	0900	--
75	1536	0600	--
110	1047	0417	0.026
134.5	857	0359	0.058
150	768	0300	--
300	384	0180	--
600	192	00C0	--
1200	96	0060	--
1800	64	0040	--
2000	58	003A	0.69
2400	48	0030	--
3600	32	0020	--
4800	24	0018	--
7200	16	0010	--
9600	12	000C	--
19200	6	0006	--

Figure 4-168. Baud Rates at 1.8432 MHz

Interrupt Enable Register (hex nF9)

This 8-bit register allows the four types of controller interrupts to separately activate the 'chip-interrupt' output signal. The interrupt system can be totally disabled by clearing bits 0 through 3 of the Interrupt Enable register. Similarly, by setting the appropriate bits of this register to 1, selected interrupts can be enabled. Disabling the interrupts inhibits the 'chip-interrupt' output signal from the controller. All other system functions operate normally, including the setting of the Line Status and Modem Status registers.

Bit	Function
7 - 4	Reserved = 0
3	Modem-Status Interrupt
2	Receiver-Line-Status Interrupt
1	Transmitter-Holding-Register-Empty Interrupt
0	Received Data Available Interrupt

Figure 4-169. Interrupt Enable Register

Bits 7 - 4 Reserved. These bits are always cleared to 0.

Bit 3 When set to 1, this bit enables the modem-status interrupt.

Bit 2 When set to 1, this bit enables the receiver-line-status interrupt.

Bit 1 When set to 1, this bit enables the transmitter-holding-register-empty interrupt.

Bit 0 When set to 1, this bit enables the received-data-available interrupt.

Interrupt Identification Register (hex nFA)

In order to minimize programming overhead during data character transfers, the controller prioritizes interrupts into four levels:

- Priority 1 - Receiver-line-status
- Priority 2 - Received-data-available
- Priority 3 - Transmitter-holding-register-empty
- Priority 4 - Modem status.

Information about a pending interrupt is stored in the Interrupt Identification register. When the Interrupt Identification register is addressed, the pending interrupt with the highest priority is held and no other interrupts are acknowledged until the system microprocessor services that interrupt.

Bit	Function
7 - 3	Reserved = 0
2	Interrupt ID, Bit 1
1	Interrupt ID, Bit 0
0	Interrupt Pending = 0

Figure 4-170. Interrupt Identification Register

Bits 7 - 3 Reserved. These bits are always cleared to 0.

Bits 2, 1 These two bits identify the pending interrupt with the highest priority, as shown in Figure 4-171 on page 4-161.

Bit 0 When this bit is set to 1, no interrupt is pending, and polling (if used) continues. When this bit is cleared to 0, an interrupt is pending, and the contents of this register can be used as a pointer to the appropriate interrupt service routine.

This bit can be used in either hard-wired, prioritized, or polled conditions to indicate if an interrupt is pending.

Bits 2 - 0 select Interrupt Control Functions as shown in the following figure:

Bits 2 1 0	Priority	Type	Cause	Interrupt Reset Control
0 0 1	-	None	None	-
1 1 0	Highest	Receiver Line Status	Overrun, Parity, or Framing Error or Break Interrupt	Read the Line Status Register
1 0 0	Second	Received Data Available	Data in Receiver Buffer	Read the Receiver Buffer Register
0 1 0	Third	Transmitter Holding Register Empty	Transmitter Holding Register is Empty	Read Interrupt Identification register or Write to Transmitter Holding Register
0 0 0	Fourth	Modem Status	Change in signal status from modem	Read the Modem Status Register

Figure 4-171. Interrupt Control Functions

Line Control Register (hex nFB)

The format of asynchronous communications is programmed through the Line Control register.

Bit	Function
7	Divisor Latch Access Bit
6	Set Break
5	Stick Parity
4	Even Parity Select
3	Parity Enable
2	Number of Stop Bits
1	Word Length Select, Bit 1
0	Word Length Select, Bit 0

Figure 4-172. Line Control Register (LCR)

Bit 7 This bit must be set to 1 during a read or write operation to gain access to the divisor latches of the baud-rate generator. It must be cleared to 0 to gain access to the Receiver Buffer, Transmitter Holding, or Interrupt Enable registers.

Bit 6 When this bit is set to 1 set-break is enabled, serial output is forced to the spacing state and remains there regardless of other transmitter activity. When this bit is cleared to 0, set-break is disabled.

Bit 5 When bits 5, 4, and 3 are set to 1, the parity bit is sent and checked as a logical 0. When bits 5 and 3 are set to 1, and bit 4 is cleared to 0, the parity bit is sent and checked as a logical 1.

Bit 4 When this bit and bit 3 are set to 1, an even number of logical ones are transmitted and checked in the data word bits and parity bit. When this bit is cleared to 0, and bit 3 is set to 1, an odd number of logical ones are transmitted and checked in the data word bits and parity bit.

Bit 3 When set to 1, a parity bit is generated (transmit data) or checked (receive data) between the last data word bit and stop bit. (The parity bit is used to produce an even or odd number of ones when the data-word bits and the parity bit are summed.)

Bit 2 This bit, along with bits 0 and 1, specifies the number of stop bits in each serial character that is sent or received as shown in the following figure:

Bit 2	Word Length *	Number of Stop Bits
0	N/A	1
1	5-Bits	1-1/2
1	6-Bits	2
1	7-Bits	2
1	8-Bits	2

* Word Length is specified by bits 1 and 0 in this register.

Figure 4-173. Stop Bits

Bits 1, 0 These two bits specify the number of bits in each serial character that is sent or received. Word length is selected as shown in the following figure:

Bit 1 0	Word Length
0 0	5-Bits
0 1	6-Bits
1 0	7-Bits
1 1	8-Bits

Figure 4-174. Word Length

Modem Control Register (Hex nFC)

This 8-bit register controls the data exchange with the modem, data set, or peripheral device emulating a modem.

Bit	Function
7 - 5	Reserved = 0
4	Loop
3	Out 2
2	Out 1
1	Request-to-Send
0	Data-Terminal-Ready

Figure 4-175. Modem Control Register (MCR)

Bits 7 - 5 Reserved. These bits are always cleared to 0.

Bit 4 This bit provides a loopback feature for diagnostic testing of the serial port. When bit 4 is set to 1:

- Transmitter-serial-output is set to the marking state.
- Receiver-serial-input is disconnected.
- Output of the Transmitter Shift register is "looped back" to the Receiver Shift register input.

  Note: The Transmitter and Receiver Shift registers are not accessible NS16550 registers.

- The modem control inputs (CTS, DSR, DCD, and RI) are disconnected.
- The modem control outputs (DTR, RTS, OUT 1, and OUT 2) are internally connected to the four modem control inputs.
- The modem control output pins are forced inactive.

When the serial port is in the diagnostic mode, transmitted data is immediately received. This feature allows the

system microprocessor to verify the transmit-data and receive-data paths of the serial port.

When the serial port is in the diagnostic mode, the receiver and transmitter interrupts are fully operational. The modem control interrupts are also operational, but their sources are the lower four bits of the Modem Control register instead of the four modem control input signals. The interrupts are still controlled by the Interrupt Enable register.

Bit 3 This bit controls the 'output 2' (OUT 2) signal which is an auxiliary user-designated interrupt enable signal. OUT 2 is used to control the interrupt signal to the channel. Setting this bit to 1 enables the interrupt. Clearing this bit to 0 disables the interrupt.

Bit 2 This bit controls the 'output 1' (OUT 1) signal which is an auxiliary user-designated output signal. When set to 1, OUT 1 is forced active. When cleared to 0, OUT 1 is forced inactive.

Bit 1 This bit controls the 'request-to-send' (RTS) modem control output signal. When set to 1, RTS is forced active. When cleared to 0, RTS is forced inactive.

Bit 0 This bit controls the 'data-terminal-ready' (DTR) modem control output signal. When set to 1, DTR is forced active. When cleared to 0, DTR is forced inactive.

Line Status Register (hex nFD)

This 8-bit register provides the system microprocessor with status information about the data transfer.

Bit	Function
7	Reserved = 0
6	Transmitter Shift Register Empty (TEMT)
5	Transmitter Holding Register Empty (THRE)
4	Break Interrupt (BI)
3	Framing Error (FE)
2	Parity Error (PE)
1	Overrun Error (OR)
0	Data Ready (DR)

Figure 4-176. Line Status Register (LSR)

Bit 7 Reserved. This bit is always cleared to 0.

Bit 6 This bit is set to 1 when the Transmitter Holding register and the Transmitter Shift register are both empty. It is cleared to 0 when either the Transmitter Holding register or the Transmitter Shift register contains a data character.

Bit 5 This bit indicates that the serial port controller is ready to accept a new character for transmission. This bit is set to 1 when a character is transferred from the Transmitter Holding register into the Transmitter Shift register. This bit is cleared to 0 when the system microprocessor loads the Transmitter Holding register.

This bit also causes the controller to issue an interrupt to the system microprocessor when bit 1 in the Interrupt Enable register is set to 1.

Bit 4 This bit is set to 1 when the received data input is held in the spacing state for longer than a fullword transmission time (that is, the total time of start bit + data bits + parity + stop bits).

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 3 This bit is set to 1 when the stop bit, following the last data bit or parity bit, is at a spacing level. This indicates that the received character did not have a valid stop bit.

Bit 2 This bit is set to 1 when a parity error is detected (the received character does not have the correct even or odd parity, as selected by the even-parity-select bit). This bit is cleared to 0 when the system microprocessor reads the contents of the Line Status register.

Bit 1 When set to 1, this bit indicates that data in the Receiver Buffer register was not read by the system microprocessor before the next character was transferred into the Receiver Buffer register, destroying the previous character. This bit is cleared to 0 when the system microprocessor reads the contents of the Line Status register.

Bit 0 This bit is set to 1 when a complete incoming character has been received and transferred into the Receiver Buffer register. This bit is cleared to 0 by reading the Receiver Buffer register.

Modem Status Register (hex nFE)

This 8-bit register provides the current state of the control lines from the modem (or external device) to the system microprocessor. In addition, bits 3 through 0 of this register provide change information. These four bits are set to logical 1 whenever a control input from the modem changes state. They are reset to logical 0 whenever the system microprocessor reads this register.

Bit	Function
7	Data-Carrier-Detect
6	Ring Indicator
5	Data-Set-Ready
4	Clear-to-Send
3	Delta-Data-Carrier-Detect
2	Trailing Edge Ring Indicator
1	Delta-Data-Set-Ready
0	Delta-Clear-to-Send

Figure 4-177. Modem Status Register (MSR)

Bit 7 This bit is the inverted 'data-carrier-detect' modem control input signal. If bit 4 of the Modem Control register is set to 1, this bit is equivalent to bit 3 in the Modem Control register.

Bit 6 This bit is the inverted 'ring-indicator' modem control input signal. If bit 4 of the Modem Control register is set to 1, this bit is equivalent to bit 2 in the Modem Control register.

Bit 5 This bit is the inverted 'data-set-ready' modem control input signal. If bit 4 of the Modem Control register is set to 1, this bit is equivalent to bit 0 in the Modem Control register.

Bit 4 This bit is the inverted 'clear-to-send' modem control input signal. If bit 4 of the Modem Control register is set to 1, this bit is equivalent to bit 1 in the Modem Control register.

Bit 3 When set to 1, this bit indicates that the 'data-carrier-detect' modem control input signal has changed state since the last time it was read by the system microprocessor.

Note: Whenever bit 0, 1, 2, or 3 is set to 1, a modem status interrupt is generated.

Bit 2 When set to 1, this bit indicates that the 'ring-indicator' modem control input signal has changed from an active condition to an inactive condition.

Bit 1 When set to 1, this bit indicates that the 'data-set-ready' modem control input signal has changed state since the last time it was read by the system microprocessor.

Bit 0 When set to 1, this bit indicates that the 'clear-to-send' modem control input signal has changed state since the last time it was read by the system microprocessor.

Scratch Register

This register does not control the serial port in any way. It can be used by the system microprocessor to temporarily hold data.

Serial Port Controller Programming Considerations

The serial port uses the NS16550 serial communications controller. The NS16550 is functionally compatible with the NS16450 and appears identical to software as the IBM Personal Computer AT serial port on the serial/parallel adapter. See "Hardware Interrupts" on page 9-6 for additional programming considerations.

The serial port can be configured to either Serial 1 or Serial 2 using the system configuration utilities.

Signal Descriptions

Modem Control Input Signals

The following are input signals from the modem or external device to the controller. Bits 7 through 4 in the Modem Status register indicate the condition of these signals. Bits 3 through 0 in the Modem Status register monitor these signals to indicate when the modem changes state.

Clear to Send (CTS): When active, this signal indicates that the modem is ready for the serial port to transmit data.

Data Set Ready (DSR): When active, this signal indicates the modem or data set is ready to establish the communications link and transfer data with the controller.

Ring Indicator (RI): When active, this signal indicates the modem or data set detected a telephone ringing signal.

Data Carrier Detect (DCD): When active, this signal indicates that the modem or data set detected a data carrier.

Modem Control Output Signals

The following are controller output signals. They are all set inactive upon a master reset operation. These signals are controlled by bits 3 through 0 in the Modem Control register.

Data Terminal Ready (DTR): When active, this signal informs the modem or data set that the controller is ready to communicate.

Request to Send (RTS): When active, this signal informs the modem or data set that the controller is ready to send data.

Output 1 (OUT 1): This signal is pulled high.

Output 2 (OUT 2): User-designated output. This signal controls interrupts to the system.

Voltage Interchange Information

The signal is considered in the *marking* condition when the voltage on the interchange circuit, measured at the interface point, is more negative than -3 Vdc with respect to signal ground. The signal is considered in the *spacing* condition when the voltage is more positive than +3 Vdc with respect to signal ground. The region between +3 Vdc and -3 Vdc is defined as the transition region, and considered an invalid level. The voltage that is more negative than -15 Vdc or more positive than +15 Vdc is also considered an invalid level.

Interchange Voltage	Binary State	Signal Condition	Interface Control Function
Positive Voltage	Binary 0	Spacing	On
Negative Voltage	Binary 1	Marking	Off

Figure 4-178. Voltage Levels

Connector

The interface uses the standard D-shell connector and pin assignments defined for RS-232-C. The voltage levels are EIA only. Current loop interface is not supported.

The following figure shows the pin assignments for the serial port in a communications environment.

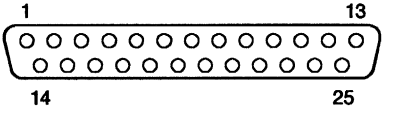


Figure 4-179. Serial Port Connector

Pin No.	I/O	Signal Name	Pin No.	I/O	Signal Name
1	NA	Not Connected	14	NA	Not Connected
2	O	Transmit Data	15	NA	Not Connected
3	I	Receive Data	16	NA	Not Connected
4	O	Request to Send	17	NA	Not Connected
5	I	Clear to Send	18	NA	Not Connected
6	I	Data Set Ready	19	NA	Not Connected
7	NA	Signal Ground	20	O	Data Terminal Ready
8	I	Data Carrier Detect	21	NA	Not Connected
9	NA	Not Connected	22	I	Ring Indicate
10	NA	Not Connected	23	NA	Not Connected
11	NA	Not Connected	24	NA	Not Connected
12	NA	Not Connected	25	NA	Not Connected
13	NA	Not Connected			

Figure 4-180. Serial Port Connector Signal and Voltage Assignments

Parallel Port Controller

The parallel port allows the attachment of various devices that transfer 8 bits of parallel data at standard TTL levels. It has a 25-pin, D-shell connector. This port may be addressed as parallel port 1, 2, or 3.

The parallel port is designed to be compatible with previous IBM Personal Computer parallel port implementations. The primary function of the parallel port is to attach a printer with a parallel interface to the system. The parallel port has an extended mode that allows support of bidirectional input and output. The port also supports level sensitive interrupts and a readable interrupt pending status.

The following figure is a block diagram of the parallel port controller.

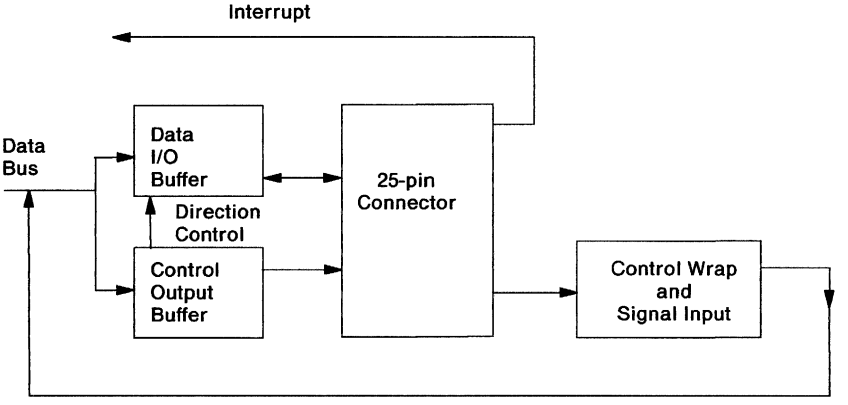


Figure 4-181. Parallel Port Controller Block Diagram

Parallel Port Programmable Option Select

The parallel port can be configured to three different address spaces previously used in IBM Personal Computer products. These addresses are selected by placing the system board in Setup (see "Programmable Option Select" on page 2-29) and performing an I/O write to port hex 0102. Bits 6 and 5 in port hex 0102 are used to select the address spaces shown below.

Bit 6	Bit 5	Function
0	0	Parallel 1
0	1	Parallel 2
1	0	Parallel 3

Figure 4-182. Parallel Port Configuration

The address assignments for each configuration are shown in the following figure.

	Data Address (hex)	Status Address (hex)	Parallel Control Address (hex)
Parallel 1	03BC	03BD	03BE
Parallel 2	0378	0379	037A
Parallel 3	0278	0279	027A

Figure 4-183. Parallel Port Address Assignments

Parallel Port Extended Mode

The extended mode option of the parallel port is selected through Programmable Option Select. With the system board in Setup, the extended mode is selected by writing a 0 to bit 7 of I/O address hex 0102. The extended mode makes the parallel port an 8-bit parallel and bidirectional interface. The following figure shows the possible configurations for the parallel port in the extended mode.

Port Mode	Port Direction	POS Mode Bit	Parallel Control Direction Bit	System Reset
Extended	Write	0	0	1
Extended	Write	0	0	0
Extended	Read	0	1	0
Compatible	Write	1	NA	0

Figure 4-184. Parallel Port Extended Mode Configurations

Parallel Port Controller Programming Considerations

The following are some considerations for programming the parallel port controller.

The interface responds to five I/O instructions: two output and three input. In the compatible mode, the output instructions transfer data into two latches whose outputs are presented on the pins of the D-shell connector. In the extended mode, the 8-bit data latch output to the D-shell connector is controlled by bit 5 in the Parallel Control port.

In the compatible mode, two of the three input instructions allow the processor to read back the contents of the two latches. In the extended mode, the read-back of the 8-bit data in the Data Address is controlled by bit 5 in the Parallel Control port. The third input instruction allows the system microprocessor to read the real-time status of a group of pins on the connector.

The extended mode can be used by externally attached equipment.

During POST the parallel port is configured as an output port. POST status information is written to this port during the power-on initialization as well as initialization after a reset from the keyboard (Ctrl, Alt, Del).

The following is a detailed description of each interface port instruction. For specific signal timing parameters, refer to the specifications for the equipment connected to the parallel port connector.

Data Address Port

The Data Address port is the 8-bit data port for both the compatible and extended modes. For the compatible mode, a write operation to this port immediately presents data to the connector pins. A read operation from this port in the compatible mode produces the data that was last written to it.

In the extended mode, a write operation to this port latches the data but it is only presented to the connector pins if the direction bit was set to Write in the Parallel Control port. A Read operation in the extended mode produces either:

- The data previously written if the direction bit in the Parallel Control port is set to Write.
- The data on the connector pins from another device if the direction bit is set to Read.

Port Bit	Port Data
Bit 7 - 0	Data

Figure 4-185. Data Address Port

Bits 7 - 0 These bits represent the 'data' (D7 - D0) signal lines.

Status Port

The status port is a read-only port in either mode. A read operation to this port presents the system microprocessor with the interrupt pending status of the interface as well as the real-time status of the connector pins as shown in the following figure. An interrupt is pending when the interrupt status bit is set to 0.

Port Bit	Port Data
Bit 7	-BUSY
Bit 6	-ACK
Bit 5	PE
Bit 4	Select
Bit 3	-ERROR
Bit 2	-IRQ Status
Bit 1, 0	Reserved

Figure 4-186. Status Port

Bit 7 This bit represents the state of the '-busy' (-BUSY) signal. When this signal is active, the printer is busy and cannot accept data.

Bit 6 This bit represents the current state of the printer '-acknowledge' (-ACK) signal. When this bit is set to 0, the printer has received a character and is ready to accept another.

Bit 5 This bit represents the current state of the printer 'paper end' (PE) signal. When this bit is set to 1, the printer has detected the end of the paper.

Bit 4 This bit represents the current state of the 'select' (SLCT) signal. When this bit is set to 1, the printer has been selected.

Bit 3 This bit represents the current state of the printer '-error' (-ERROR) signal. When this bit is set to 0, the printer has encountered an error condition.

Bit 2 When this bit is set to 0, the printer has acknowledged the previous transfer using the '-acknowledge' signal.

Bits 1, 0 Reserved

Parallel Control Port

The Parallel Control port is a read or write port. A write operation this port latches the six least-significant data bits of the bus. The sixth bit corresponds to the direction control bit and is only applicable in the extended mode. The remaining five bits are compatible with previous implementations as shown in the following figure. A read operation to the Parallel Control port presents the system microprocessor the data that was last written to it, with the exception of the write-only direction bit.

Port Bit	Port Data
Bit 7, 6	Reserved
Bit 5	Direction
Bit 4	IRQ EN
Bit 3	Pin 17 (SLCT IN)
Bit 2	Pin 16 (-INIT)
Bit 1	Pin 14 (AUTO FD XT)
Bit 0	Pin 1 (STROBE)

Figure 4-187. Parallel Control Port

Bits 7, 6 Reserved

Bit 5 This bit controls the direction of the data port. For more information on the use of this bit see Figure 4-184 on page 4-172. This is a write-only bit.

Bit 4 This bit enables the parallel port interrupt. When this bit is set to 1, an interrupt occurs when the '-acknowledge' signal changes from active to inactive.

Bit 3 This bit controls the 'select in' (SLCT IN) signal. When this bit is set to 1, the printer is selected.

Bit 2 This bit controls the 'initialize printer' (-INIT) signal. When this bit is set to 0, the printer starts.

Bit 1 This bit controls the 'automatic feed XT' (AUTO FD XT) signal. When this bit is set to 1, the printer will automatically line feed after each line is printed.

Bit 0 This bit controls the 'strobe' signal to the printer. When this bit is set to 1, data is pulse-clocked into the printer.

Parallel Port Timing

The timing for the parallel port depends on the timing of the devices connected to the port. The following diagram shows the sequence for typical parallel port signal timing.

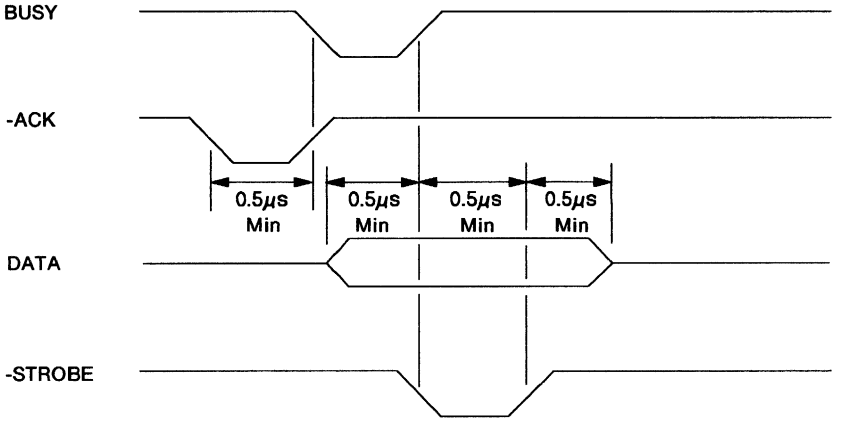


Figure 4-188. Parallel Port Timing Sequence.

Signal Descriptions

The following figures list characteristics of the signals.

Sink Current	24 mA	Maximum
Source Current	15 mA	Maximum
High-level Output Voltage	2.4 Vdc	Minimum
Low-level Output Voltage	0.5 Vdc	Maximum

Figure 4-189. Data and Interrupt Signals

Pins 1, 14, 16, and 17 are driven by open collector drivers pulled to 5 Vdc through 4.7 kilohm resistors.

Sink Current	20 mA	Maximum
Source Current	0.55 mA	Maximum
High-level Output Voltage	5.0 Vdc minus pullup	Minimum
Low-level Output Voltage	0.5 Vdc	Maximum

Figure 4-190. Control Signals

Connector

The parallel port connector is a standard 25-pin D-shell connector. The D0 - D7 lines on the connector are driven by drivers capable of sourcing 15 milliamps and sinking 24 milliamps.

The following are the voltages and signals assigned to the parallel port controller connector.

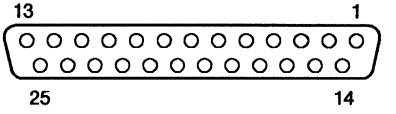


Figure 4-191. Parallel Port Connector

Pin No.	I/O	Signal Name	Pin No.	I/O	Signal Name
1	I/O	-STROBE	14	O	-AUTO FEED XT
2	I/O	Data Bit 0	15	I	-ERROR
3	I/O	Data Bit 1	16	O	-INIT
4	I/O	Data Bit 2	17	O	-SLCT IN
5	I/O	Data Bit 3	18	NA	Ground
6	I/O	Data Bit 4	19	NA	Ground
7	I/O	Data Bit 5	20	NA	Ground
8	I/O	Data Bit 6	21	NA	Ground
9	I/O	Data Bit 7	22	NA	Ground
10	I	-ACK	23	NA	Ground
11	I	BUSY	24	NA	Ground
12	I	PE	25	NA	Ground
13	I	SLCT			

Figure 4-192. Parallel Port Connector Signal and Voltage Assignments

Memory

The system has the following types of memory:

- Read Only Memory (ROM)
- Random Access Memory (RAM)
- Real-Time Clock/Complementary Metal Oxide Semiconductor RAM (RT/CMOS RAM).

Read Only Memory (ROM) Subsystem

The system board ROM subsystem consists of four 32K by 8-bit modules in a 32K by 32-bit arrangement. ROM is assigned at the top of the first and last 1M address space (000E0000 and FFFE0000). ROM is not parity-checked. It operates with one 62.5-nanosecond wait state.

Random Access Memory (RAM) Subsystem

The system board RAM subsystem starts at address hex 00000000 of the address space. Memory modules are not mounted directly on the system board; they are mounted on a printed-circuit board (card) approximately 114.3 mm (4.5 in.) high by 101.6 mm (4.0 in.) long. Each memory card[2] contains 1M of memory and attaches to the system board through a 3- by 32-pin connector. The system board can accommodate two memory cards for a total capacity of 2M. At least one memory card must be installed before additional memory can be used from the Micro Channel.

The RAM subsystem is 36 bits; 32 data bits and 4 parity bits. One parity bit is generated for each byte of data written. During a read operation, one parity bit is checked for each byte of data read by the device controlling the bus.

The memory cycle time depends on the device in control of the system bus. The following figure shows the various memory cycle times.

---

2 IBM 80386 System Board Memory Expansion Kit

Controlling Device	Wait	Total Cycle Time (Including Wait)
80386 (16 MHz)	62.5 ns	187.5 ns
DMA (8 MHz)	125 ns	375 ns
Other	N/A	300 ns minimum

Figure 4-193. Memory Cycle Times

Each time the system is powered on, a routine in the POST performs the following functions:

- Waits for memory refreshes.

  Note: The memory must be refreshed 64K times before it can be used. This requires approximately 1.1 seconds with a normal refresh or 60 milliseconds with a fast refresh. The refresh rate is defined by the System Board Memory Control register. See page 2-35 for more information about this register.

- Performs a memory write operation to each memory location.

Memory is activated and deactivated in 1M blocks. Each 1M block must start on a 1M boundary. Because 128K of system-board ROM, 128K of I/O ROM, and 128K of video RAM are mapped within the first 1M address space, an overflow is created from the first 1M memory card installed on the system board. The first 1M of physical memory can be *split* at either 512K or 640K. The overflow is typically remapped to the beginning of the first available 1M boundary following the last full 1M block of memory activated (either system board memory or channel memory). Alternately, the overflow can be either disabled or remapped to the last 128K of the first 1M replacing the system board ROM. The memory split, remapping, and disabling are controlled by the Memory Encoding register and the Split Address register. These registers are described on pages 2-37 and 2-39 respectively.

System Memory Signals

The following is a list of signals used by the system board memory cards.

-MW: -Memory Write

-MA(0 – 8): -Memory Address (0 through 8)

-RAS(0 – 3): -Row Address Strobe (0 through 3)

-CAS(0 – 3): -Column Address Strobe (0 through 3)

-CASP: -Column Address Strobe Parity

MDP(0 – 3): Memory Data Parity (0 through 3)

MD(0 – 31): Memory Data (0 through 31)

-BE(0 – 3): -Byte Enable (0 through 3)

R: This signal is used with the 'T' signal (pin A32) by the Memory Card Definition register to indicate the presence of memory in each of the two system board memory connectors. See "Memory Card Definition Register" on page 2-36 for additional information.

T: This signal is used with the 'R' signal (pin A17) by the Memory Card Definition register to indicate the presence of memory in each of the two system board memory connectors. See "Memory Card Definition Register" on page 2-36 for additional information

System Board Memory Connectors

The following figure shows the pin locations of the 3- by 32-pin system board memory connectors. The pin locations are the same for connectors 1 and 2. Connector 1 is located closest to the power supply; connector 2 is located furthest from the power supply. Pin number 32 is closest to the rear of the system board.

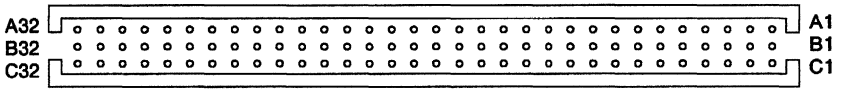


Figure 4-194. System Board Memory Connector Pin Locations

The following figure shows the pin assignments for the system board memory connector. Inputs (I) to and Outputs (O) from the memory cards are also shown.

Row A			Row B			Row C		
Pin	**I/O**	**Signal**	**Pin**	**I/O**	**Signal**	**Pin**	**I/O**	**Signal**
A1	NA	Reserved	B1	NA	Ground	C1	I/O	MD0
A2	I	-MW	B2	NA	+5 Vdc	C2	I/O	MD1
A3	I	MA0	B3	NA	Ground	C3	I/O	MD2
A4	I	MA1	B4	NA	+5 Vdc	C4	I/O	MD3
A5	I	MA2	B5	NA	Ground	C5	I/O	MD4
A6	I	MA3	B6	NA	+5 Vdc	C6	I/O	MD5
A7	I	MA4	B7	NA	Ground	C7	I/O	MD6
A8	I	MA5	B8	NA	+5 Vdc	C8	I/O	MD7
A9	I	MA6	B9	NA	Ground	C9	I/O	MD8
A10	I	MA7	B10	NA	+5 Vdc	C10	I/O	MD9
A11	I	MA8	B11	NA	Ground	C11	I/O	MD10
A12	I	-RAS0	B12	NA	+5 Vdc	C12	I/O	MD11
A13	I	-RAS1	B13	NA	Ground	C13	I/O	MD12
A14	I	-RAS2	B14	NA	+5 Vdc	C14	I/O	MD13
A15	I	-RAS3	B15	NA	Ground	C15	I/O	MD14
A16	NA	Reserved	B16	NA	+5 Vdc	C16	I/O	MD15
A17	O	R	B17	NA	Ground	C17	I/O	MD16
A18	NA	Reserved	B18	NA	+5 Vdc	C18	I/O	MD17
A19	I	-CAS0	B19	NA	Ground	C19	I/O	MD18
A20	I	-CAS1	B20	NA	+5 Vdc	C20	I/O	MD19
A21	I	-CAS2	B21	NA	Ground	C21	I/O	MD20
A22	I	-CAS3	B22	NA	+5 Vdc	C22	I/O	MD21
A23	I/O	MDP0	B23	NA	Ground	C23	I/O	MD22
A24	I/O	MDP1	B24	NA	+5 Vdc	C24	I/O	MD23
A25	I/O	MDP2	B25	NA	Ground	C25	I/O	MD24
A26	I/O	MDP3	B26	NA	+5 Vdc	C26	I/O	MD25
A27	I	-BE0	B27	NA	Ground	C27	I/O	MD26
A28	I	-BE1	B28	NA	+5 Vdc	C28	I/O	MD27
A29	I	-BE2	B29	NA	Ground	C29	I/O	MD28
A30	I	-BE3	B30	NA	+5 Vdc	C30	I/O	MD29
A31	I	-CASP	B31	NA	Ground	C31	I/O	MD30
A32	O	T	B32	NA	+5 Vdc	C32	I/O	MD31

Figure 4-195. Memory Pin Assignments

Real-Time Clock/Complementary Metal Oxide Semiconductor RAM

The real-time clock/complementary metal oxide semiconductor (RT/CMOS) RAM chip (Motorola MC146818A) contains the real-time clock and 64 bytes of nonvolatile RAM. The internal clock circuitry uses 14 bytes of this memory, and the rest is allocated to configuration and system status information. In addition to the 64-byte CMOS RAM, a 2K nonvolatile RAM extension is provided for configuration and system information.

A 6 Vdc lithium battery maintains voltage to the RT/CMOS and 2K extension when the power supply is not in operation.

The system unit cover can be physically locked to prevent unauthorized access to the battery. This helps prevent unauthorized battery removal and loss of password and configuration information.

The following figure shows the RT/CMOS RAM bytes and their addresses.

Address (hex)	Function
000 - 00D	Real-Time Clock Information
00E	Diagnostic Status Byte
00F	Shut Down Status Byte
010	Diskette Drive Byte
011	First Fixed Disk Type Byte
012	Second Fixed Disk Type Byte
013	Reserved
014	Equipment Byte
015 - 016	Low and High Base Memory Bytes
017 - 018	Low and High Memory Expansion Bytes
019 - 031	Reserved
032 - 033	Configuration CRC Bytes
034 - 036	Reserved
037	Date Century Byte
038 - 03F	Reserved

Figure 4-196. RT/CMOS RAM Address Map

RT/CMOS RAM I/O Operations

When performing I/O operations to the RT/CMOS RAM addresses (described in Figure 4-196 and the following pages), interrupts should be inhibited to avoid having interrupt routines change the CMOS address register before data is read or written. Port hex 0070 should be left to point to Status register D of the RT/CMOS.

The following steps are required to perform I/O operations to the RT/CMOS RAM addresses:

1. OUT to port hex 0070 with the RT/CMOS address that will be written to.

   Note: The NMI mask bit resides in port hex 0070 (see "Nonmaskable Interrupt" on page 3-28).

2. OUT to port hex 0071 with the data to be written.

Reading RT/CMOS RAM requires the following steps:

1. OUT to port hex 0070 with the CMOS address that is to be read from.
2. IN from port hex 0071, and the data read is returned in the AL register.

Warning: When writing port hex 0070, a read to port hex 0071 must be accessed immediately. Failure to do this may cause intermittent malfunctions and unreliable operation of the MC146818A Real-Time Clock/Complementary Metal Oxide Semiconductor RAM.

Real-Time Clock

The following shows bit definitions for the real-time clock bytes (addresses hex 000 - 00D).

Address (hex)	Function	Byte Number
000	Seconds	0
001	Second Alarm	1
002	Minutes	2
003	Minute Alarm	3
004	Hours	4
005	Hour Alarm	5
006	Day Of Week	6
007	Date Of Month	7
008	Month	8
009	Year	9
00A	Status Register A	10
00B	Status Register B	11
00C	Status Register C	12
00D	Status Register D	13

Figure 4-197. Real-Time Clock (Addresses Hex 000 - 00D)

Note: The Setup program initializes registers A, B, C, and D when the time and date are set. Also, interrupt hex 1A is the BIOS interface to read and set the time and date. It initializes the status bytes the same as the Setup program.

Status Register A

Bit 7 Update in Progress (UIP)—When set to 1, this bit indicates the time update cycle is in progress. When set to 0, this bit indicates the current date and time is available to read.

Bits 6 - 4 22-Stage Divider (DV2 through DV0)—These three divider-selection bits identify which time-base frequency is being used. The system initializes the stage divider to 010, which selects a 32.768 kHz time base. This is the only stage divider supported by the system unit for proper time keeping.

Bits 3 - 0 Rate Selection Bits (RS3 through RS0)—These bits allow the selection of a divider output frequency. The system initializes the rate selection bits to 0110, which selects a 1.024 kHz square-wave output frequency and a 976.562 microsecond periodic interrupt rate.

Status Register B

Bit 7 Set—When set to 0, this bit updates the cycle normally by advancing the counts at one per second. When set to 1, this bit aborts any update cycle in progress and the program can initialize the 14 time-bytes without any further updates occurring until a 0 is written to this bit.

Bit 6 Periodic Interrupt Enable (PIE)—This bit is a read/write bit that allows an interrupt to occur at a rate specified by the rate and divider bits in register A. When set to 1, this bit enables an interrupt. When set to 0, the interrupt is disabled. The system initializes this bit to 0.

Bit 5 Alarm Interrupt Enable (AIE)—When set to 1, this bit enables the alarm interrupt. When set to 0, the alarm interrupt is disabled. The system initializes this bit to 0.

Bit 4 Update-Ended Interrupt Enabled (UIE)—When set to 1, this bit enables the update-ended interrupt. When set to 0, the update-ended interrupt is disabled. The system initializes this bit to 0.

Bit 3 Square Wave Enabled (SQWE)—When set to 1, this bit enables the square-wave frequency as set by the rate selection bits in register A. When set to 0, the square wave is disabled. The system initializes this bit to 0.

Bit 2 Date Mode (DM)—This bit indicates whether the time and date calendar updates are to use binary or binary-coded-decimal (BCD) formats. When set to 1, this bit indicates a binary format. When set to 0, a BCD format is indicated. The system initializes this bit to 0.

Bit 1 24/12—This bit establishes whether the hours byte is in the 24-hour or 12-hour mode. When set to 1, this bit indicates the 24-hour mode. When set to 0, the 12-hour mode is indicated. The system initializes this bit to 1.

Bit 0 Daylight Savings Enabled (DSE)—When set to 1, this bit enables daylight savings time. When set to 0, this bit disables daylight savings time (reverts to standard time). The system initializes this bit to 0.

Status Register C

Bits 7 - 4 IRQF, PF, AF, UF—These flag bits are read-only and are affected when the AIE, PIE, and UIE interrupts are enabled with register B.

Bits 3 - 0 Reserved

Status Register D

Bit 7 Valid RAM Bit (VRB)—This bit is read-only and indicates the condition of the contents of the CMOS RAM through the power sense pin. A low state of the power sense pin indicates the real-time clock has lost its power (battery dead). When set to 1, this bit indicates power on the real-time clock. When set to 0, this bit indicates that the real-time clock has lost power.

Bits 6 - 0 Reserved

CMOS RAM Configuration

The following shows the bit definitions for the CMOS configuration bytes (addresses hex 00E - 03F).

Diagnostic Status Byte (Hex 00E)

Bit 7 Indicates the real-time clock chip has lost power. When set to 0, this bit indicates the chip has not lost power. When set to 1, this bit indicates the chip has lost power.

Bit 6 Configuration record and checksum status—When set to 0, this bit indicates the checksum is correct. When set to 1, the checksum is incorrect.

Bit 5 Incorrect configuration—This is a check, at power-on time, of the equipment byte of the configuration record. When set to 0, this bit indicates that the configuration information is correct. When set to 1, the configuration information is incorrect. Power-on checks require at least one diskette drive installed (bit 0 of the equipment byte set to 1).

Bit 4 Memory Size Miscompare—When set to 0, this bit indicates the power-on check determined the same memory size as in the configuration record. When set to 1, the memory size is different.

Bit 3 Fixed Disk Controller/Drive C Initialization Status—When set to 0, this bit indicates the controller and fixed disk drive C are functioning properly and the system can attempt a power-on reset. When set to 1, the controller and/or drive C failed initialization, which prevents the system from attempting a power-on-reset.

Bit 2 Time Status Indicator (POST validity check)—When set to 0, this bit indicates the time is valid. When set to 1, the time is invalid.

Bit 1 This bit indicates whether or not the installed adapters match the configuration information. When set to 0, this bit indicates the adapters match the configuration. When set to 1, the adapters do not match the configuration.

Bit 0 When set to 1, this bit indicates a time-out occurred while trying to read an adapter ID. When set to 0, no timeout occurred.

Shut Down Status Byte (Hex 00F): The bits in this byte are defined by the power-on diagnostics.

Diskette Drive Type Byte (Hex 010)

Bits 7 - 4 Type of first diskette drive:

0000 No drive present

0001 Double-sided diskette drive (48 tracks per inch, 360K)

0010 Reserved

0011 High-capacity diskette drive (720K)

0100 High-density diskette drive (1.44M)

Note: 0101 through 1111 are reserved.

Bits 3 - 0 Type of second diskette drive installed:

0000 No drive present

0001 Double-sided diskette drive (48 tracks per inch, 360K)

0010 Reserved

0011 High-capacity diskette drive (720K)

0100 High-density diskette drive (1.44M)

Note: 0101 through 1111 are reserved.

First Fixed Disk Type Byte (Hex 011): Bits 7 through 0 define the type of first fixed disk drive (drive C) installed. Hex 00 indicates a fixed disk drive is *not* present. See Figure 4-198 on page 4-190 for specific fixed disk drive types.

Second Fixed Disk Type Byte (Hex 012): Bits 7 through 0 define the type of second fixed disk drive (drive D) installed. Hex 00 indicates a fixed disk drive is *not* present. See Figure 4-198 on page 4-190 for specific fixed disk drive types.

The following figure shows the BIOS fixed disk parameters.

Type	Number of Cylinders	Number of Heads	Number Write Precompensation	Landing Zone	Defect Map
0 (0H)		—No fixed disk installed—			No
1 (1H)	306	4	128	305	No
2 (2H)	615	4	300	615	No
3 (3H)	615	6	300	615	No
4 (4H)	940	8	512	940	No
5 (5H)	940	6	512	940	No
6 (6H)	615	4	0FFFFH (none)	615	No
7 (7H)	462	8	256	511	No
8 (8H)	733	5	0FFFFH (none)	733	No
9 (9H)	900	15	0FFFFH (none)	901	No
10 (AH)	820	3	0FFFFH (none)	820	No
11 (BH)	855	5	0FFFFH (none)	855	No
12 (CH)	855	7	0FFFFH (none)	855	No
13 (DH)	306	8	128	319	No
14 (EH)	733	7	0FFFFH (none)	733	No
15 (FH)		—Reserved—			
16 (10H)	612	4	0 (all cyl.)	663	No
17 (11H)	977	5	300	977	No
18 (12H)	977	7	0FFFFH (none)	977	No
19 (13H)	1024	7	512	1023	No
20 (14H)	733	5	300	732	No
21 (15H)	733	7	300	732	No
22 (16H)	733	5	300	733	No
23 (17H)	306	4	0 (all cyl.)	336	No
24 (18H)	612	4	305	663	No
25 (19H)	306	4	0FFFFH (none)	340	No
26 (1AH)	612	4	0FFFFH (none)	670	No
27 (1BH)	698	7	300	732	Yes
28 (1CH)	976	5	488	977	Yes
29 (1DH)	306	4	0 (all cyl.)	340	No
30 (1EH)	611	4	306	663	Yes
31 (1FH)	732	7	300	732	Yes
32 (20H)	1023	5	FFFFH(none)	1023	Yes

Types 33 (21H) through 255 (FF) are reserved.

Figure 4-198. Fixed Disk BIOS Parameters

Reserved (Hex 013)

Equipment Byte (Hex 014)

Bits 7, 6 Indicates the number of diskette drives installed:

00 One drive

01 Two drives

10 Reserved

11 Reserved

Bits 5, 4 Primary display

00 Reserved

01 Primary display is attached to the Video Graphics Array in the 40-column mode.

10 Primary display is attached to the Video Graphics Array in the 80-column mode.

11 Primary display is attached to the Video Graphics Array in monochrome mode.

Bits 3, 2 Reserved.

Bit 1 Math coprocessor bit:

0 Math coprocessor not installed.

1 Math coprocessor installed.

Bit 0 Diskette drive bit:

0 No diskette drives installed.

1 Diskette drives installed.

Note: The equipment byte defines basic equipment in the system for power-on diagnostic tests.

Low and High Base Memory Bytes (Hex 015 and 016): These bytes define the amount of memory installed below the 640K address space.

The hex value from these bytes represent the number of 1K-byte blocks of base memory. For example, hex 0280 is equal to 640K. Hex 015 is the low-byte of the base memory size. Hex 016 is the high-byte of the base memory size.

Low and High Memory Expansion Bytes (Hex 017 and 018): These bytes define the amount of memory installed above the 1M address space.

The hex value from these bytes represent the number of 1K byte blocks of expansion memory. For example, hex 0800 is equal to 2048K. Hex 017 is the low-byte of the expansion memory size. Hex 018 is the high-byte of the expansion memory size.

Reserved (Hex 019 through 031)

Configuration CRC Bytes (Hex 032 and 033): These bytes contain the cyclic-redundancy-check data for bytes hex 010 through hex 031 of the 64-byte CMOS. Hex 032 is the high-byte of the configuration CRC. Hex 033 is the low-byte of the configuration CRC.

Reserved (Hex 034 through 036)

Date Century Byte (Hex 037): Bits 7 through 0 of this byte contain the BCD value for the century (BIOS interface to read and set).

Reserved (Hex 038 through 03F)

2K CMOS RAM Extension

The system board has a 2K CMOS RAM extension for configuration and diagnostic information. These bytes are reserved.

Miscellaneous System Ports

Ports hex 0061, 0070, and 0092 contain information that is used for system control.

System Control Port B (Hex 0061)

Port B is accessed by I/O read or write operations to I/O address hex 0061. The following shows the bit definitions.

Bit	Function
7	Reset timer 0 output latch (IRQ 0)
6 - 4	Reserved
3	Enable Channel Check
2	Enable Parity Check
1	Speaker Data Enable
0	Timer 2 Gate to Speaker

Figure 4-199. System Control Port B, Write Operations

Bit	Function
7	Parity Check
6	Channel Check
5	Timer 2 output
4	Toggles with each refresh request
3	Enable Channel Check
2	Enable Parity Check
1	Speaker Data Enable
0	Timer 2 Gate to Speaker

Figure 4-200. System Control Port B, Read Operations

Bit 7 A write operation with this bit set to 1, resets IRQ 0.

For a read operation, this bit indicates the state of the Parity Check latch. If the bit is equal to 1, a parity check has occurred.

Bit 6 For a read operation, this bit returns the state of the Channel Check latch. If the bit is equal to 1, a channel check has occurred.

Bit 5 For a read operation this bit returns the condition of timer/counter 2 'output' signal.

Bit 4 For a read operation, this bit toggles for each refresh request.

Bit 3 A write operation with this bit set to 0 enables Channel Check. This bit is set to 1 during a power-on reset. A read operation returns the result of the last write operation to this bit.

Bit 2 A write operation with this bit set to 0 enables parity check. This bit is set to 1 during a power-on reset. A read operation returns the result of the last write operation to this bit.

Bit 1 A write operation with this bit set to 1 enables speaker data. This bit is set to 0 during a power-on reset. A read operation returns the result of the last write operation to this bit.

Bit 0 A write operation with this bit set to 1 enables the 'timer 2' gate. Setting this bit to 0 turns the gate off. A read operation returns the result of the last write operation to this bit.

RT/CMOS and NMI Mask (Hex 0070)

Bit	Function
7	Non-maskable Interrupt (NMI)
6	Reserved
5 - 0	RT/CMOS RAM

Figure 4-201. RT/CMOS and NMI Mask

Bit 7 When set to 0, this bit enables the NMI. When set to 1, the NMI is masked off. This bit is set to 1 by a power-on reset. This bit is write-only. See "Interrupts" on page 3-28 for more information about the NMI.

Bit 6 Reserved

Bits 5 - 0 See "RT/CMOS RAM I/O Operations" on page 4-184.

Note: Port hex 0071 is used with port hex 0070 to read and write to the RT/CMOS RAM and the NMI Mask register.

System Control Port A (Hex 0092)

Port hex 0092 supports the fixed disk drive light, alternate system microprocessor reset, PASS A20, Watchdog timer status, and CMOS security lock. The following shows the bit definitions for port hex 0092.

Bit	Function
7	Fixed Disk activity light bit A
6	Fixed Disk activity light bit B
5	Reserved = 0
4	Watchdog Timer Status
3	Security Lock Latch
2	Reserved = 0
1	Alternate Gate A20
0	Alternate Hot Reset

Figure 4-202. System Control Port A

Bits 7, 6 These read/write bits control the fixed disk activity light. Setting either bit to 1 turns the fixed disk activity light on. Setting both bits to 0 turns the light off. The power-on reset condition of each bit is 0.

Bit 5 Reserved

Bit 4 This read-only bit indicates the Watchdog timer status. When this bit is equal to 1 a Watchdog timeout has occurred. A 0 means that no Watchdog timeout has occurred.

Bit 3 This read/write bit provides the security lock for the secured area of the RT/CMOS. Setting this bit to 1 electrically locks the 8-byte password. Once this bit is set by POST it can only be cleared by turning the system power off and then turning the power on.

Bit 2 Reserved

Bit 1 This read/write bit can be used to control address bit A20 when the microprocessor is in the Real Address Mode. When this bit is set to 1, the 'A20' signal is active. When this bit is set to 0, A20 is inactive. This bit is set to 0 during a system reset.

Bit 0 This read/write bit provides an alternate system microprocessor reset function. This function provides an alternate means to reset the system microprocessor to effect a mode switch from the Protected Virtual Address

Mode to the Real Address Mode. This mechanism supports operating systems requiring faster operation than is provided by the original implementation on the IBM Personal Computer AT using the Intel 8042. It is provided in addition to the 8042 implementation so that compatibility with existing systems and software is not affected. The alternate system microprocessor reset takes 13.4 microseconds.

This bit must be set to 0 either by a system reset or a write operation. When a write operation changes this bit from 0 to 1, the alternate reset pin is pulsed high for 100 to 125 nanoseconds. The reset occurs after a minimum delay of 6.72 microseconds. After writing this bit from 0 to 1, the latch remains set so POST can read this bit. If the bit is a 0, POST assumes the system was just powered on. If the bit is a 1, POST assumes a switch from the Protected Mode to the Real Mode has taken place.

Section 5. Power Supply

Notes:

Description

The power supply automatically selects between two ranges of ac input power and converts the input power into three separate dc outputs. The power supply provides power for the following:

- System board
- Channel adapters
- Diskette drives
- Fixed disk drives
- Auxiliary device
- Keyboard.

The power switch and two light-emitting diodes (LED) are located on the front. The green LED indicates the power supply is operating. The yellow LED indicates fixed disk activity. Bits 6 and 7 of System Control Port A (hex 0092) control the 'system status' line used to illuminate the fixed disk activity LED.

Inputs

The power supply operates with two ranges of input power and automatically selects the appropriate range. The following figure shows the input requirements for both ranges:

Input Voltage		Maximum Current Draw	
Nominal	**Range**	**(Nominal)**	**Frequency**
100 - 125 Vac	90 - 137 Vac	5 Amps	50 - 60 ± 3 Hz
200 - 240 Vac	180 - 265 Vac	3 Amps	50 - 60 ± 3 Hz

Figure 5-1. Input Requirements

Input Protection

The input power line is protected against an overcurrent condition by an internal fuse.

Ground Leakage Current

The system unit ground leakage current does not exceed 500 microamps at any nominal input voltage.

Outputs

The power supply provides three voltages; +5, +12, and -12 Vdc.

Output Protection

A short circuit placed on any dc output (between outputs or between an output and dc return) latches all dc outputs into a shutdown state with no damage to the power supply.

If an overvoltage fault occurs (internal to the power supply), the supply latches all dc outputs into a shutdown state before any output exceeds 130% of its nominal value.

If either of these shutdown states is actuated, the power supply returns to normal operation only after the fault has been removed and the power switch has been turned off for at least one second.

Voltage Sequencing

At power on, the output voltages track within 50 milliseconds of each other when measured at the 50% points.

No Load Operation

The power supply is capable of operation with "no load" on the outputs. The output regulation is ± 25% and the 'power good' signal can be in either state.

Auto Restart

If the power supply outputs drop out of regulation due to an ac line outage, the power supply automatically restarts (generates output voltages) when ac power returns.

Power Good Signal

A 'power good' signal indicates proper operation of the power supply and is active high during normal operation. The 'power good' signal can source 100 microamps and sink 1 milliamp.

The power supply provides the 'power good' signal to indicate proper operation of the power supply and to reset system logic. At power on, the 'power good' signal has a turn-on delay of at least 100 milliseconds but not greater than 500 milliseconds after the +5 output has reached its minimum sense level. At power off, the 'power good' signal remains active for at least 10 milliseconds after ac power is removed and goes inactive before any voltage falls below the output regulation limits. If the 'power good' signal goes inactive due to an ac line outage, it is regenerated as described above when ac power returns. The following figure shows the minimum sense levels for the output voltages.

Level (Vdc)	Minimum (Vdc)
+5	+4.5
+12	+10.8
-12	-10.2

Figure 5-2. Sense Level

Power Supply Connectors

The power supply uses a 15-pin connector for the system board and two 4-pin connectors for the two fixed-disk drive locations. The following figures show the signals and voltages assigned to the connectors on the end of the power supply cables.

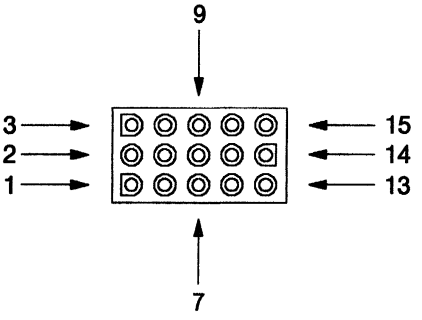


Figure 5-3. System Board Power Supply Cable Connector

Pin	I/O	Signal	Pin	I/O	Signal
1	NA	+5 Vdc	2	NA	Signal Ground
3	NA	+12 Vdc	4	NA	+5 Vdc
5	NA	Signal Ground	6	NA	Signal Ground
7	NA	+5 Vdc	8	NA	Signal Ground
9	NA	-12 Vdc	10	NA	+5 Vdc
11	NA	Signal Ground	12	O	Power Good
13	NA	+5 Vdc	14	NA	Signal Ground
15	I	System Status			

Figure 5-4. Power Supply Connector Voltage and Signal Assignments, System Board

Figure 5-5. Fixed Disk Power Supply Cable Connector

Pin	I/O	Signal	Pin	I/O	Signal
1	NA	+12 Vdc	2	NA	Signal Ground
3	NA	Signal Ground	4	NA	+5 Vdc

Figure 5-6. Power Supply Connectors Voltage and Signal Assignments, Fixed Disk

Notes:

Section 6. Keyboard

Description

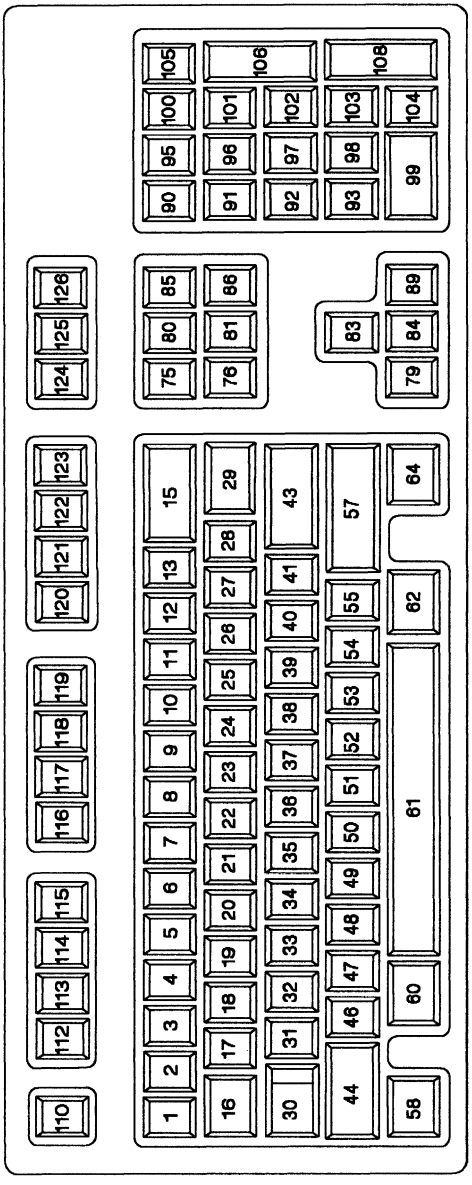
The keyboard has 101 keys (102 in countries outside the U. S.). At system power-on, the keyboard monitors the signals on the 'clock' and 'data' lines and establishes its line protocol. A bidirectional serial interface in the keyboard converts the 'clock' and 'data' signals and sends this information to and from the keyboard through the keyboard cable.

Keyboard Layouts

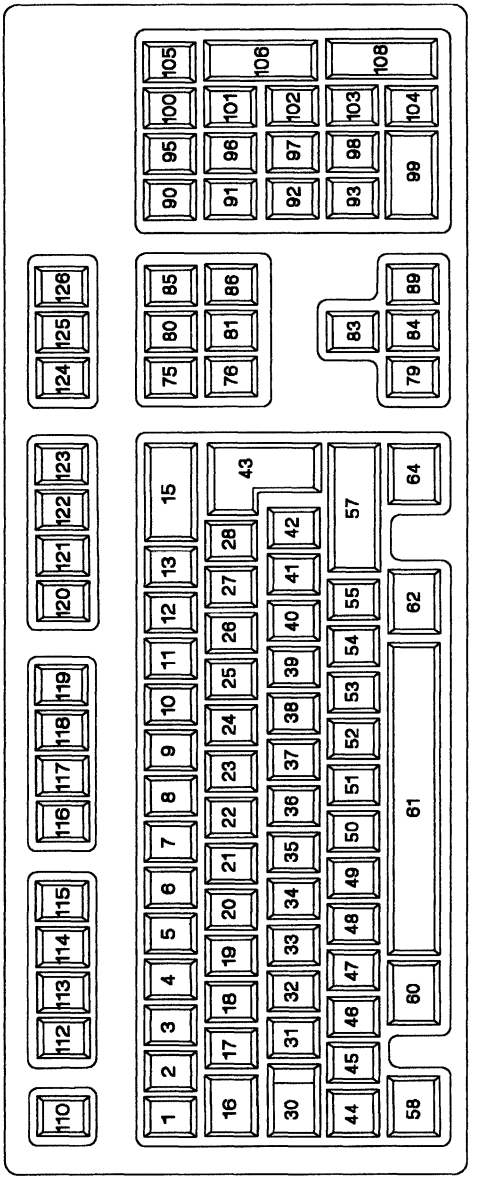
- Belgian
- Canadian French
- Danish
- Dutch
- French
- German
- Italian
- Latin American
- Norwegian
- Portuguese
- Spanish
- Swedish
- Swiss
- U.K. English
- U.S. English.

The various layouts are shown in alphabetic order on the following pages. Nomenclature is on both the top and front face of the keybuttons.

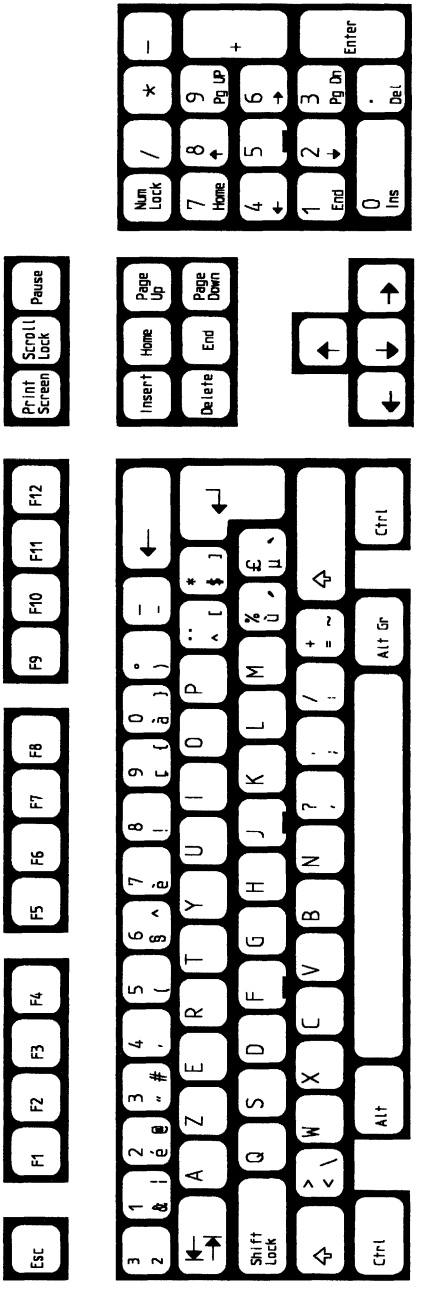
The following figure shows the keyboard layout for the 101-key keyboard.



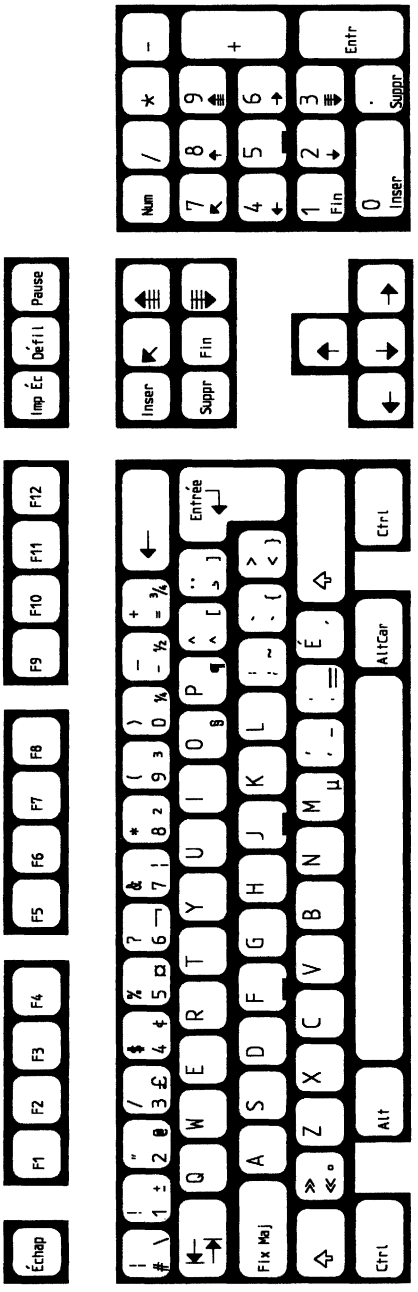
The following figure shows the keyboard layout for the 102-key keyboard.



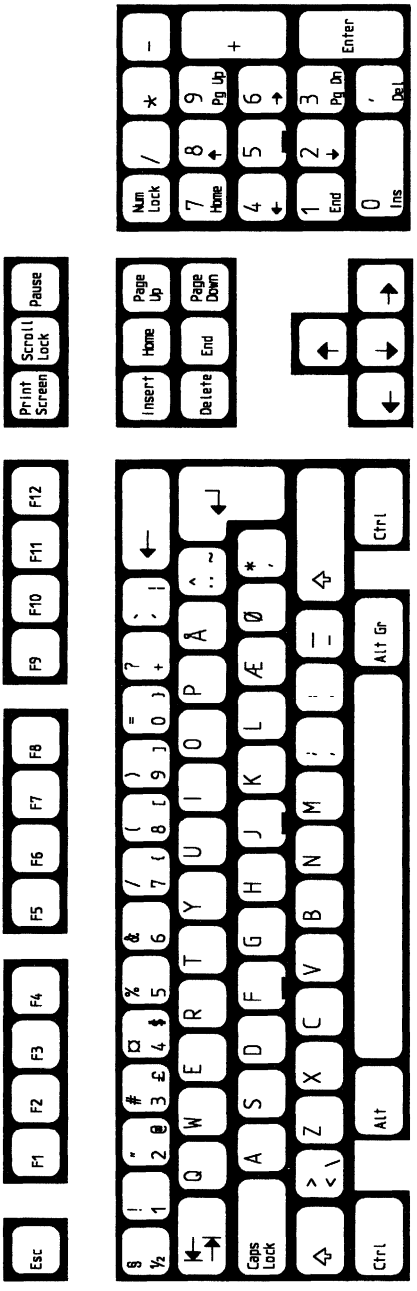
Belgian Keyboard



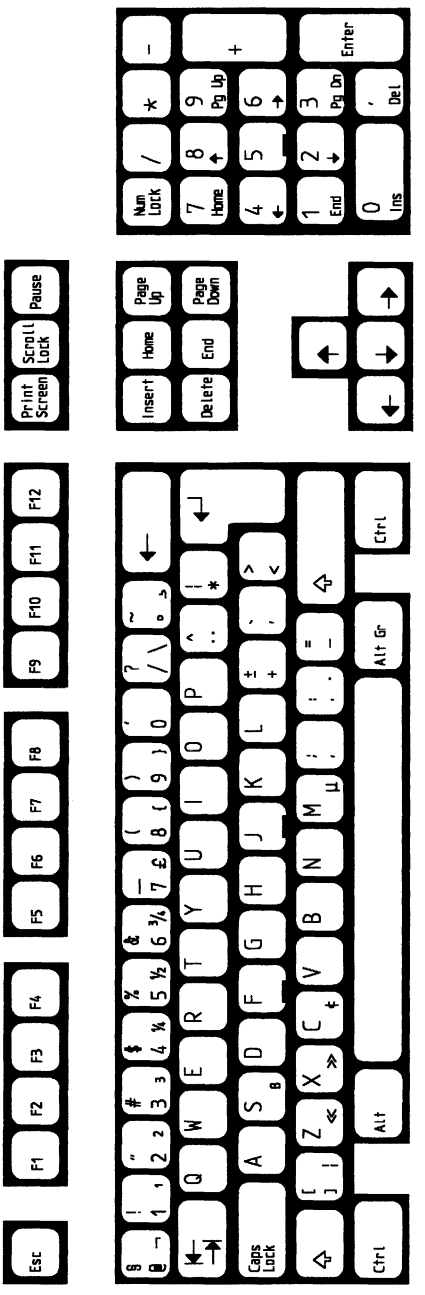
Canadian French Keyboard



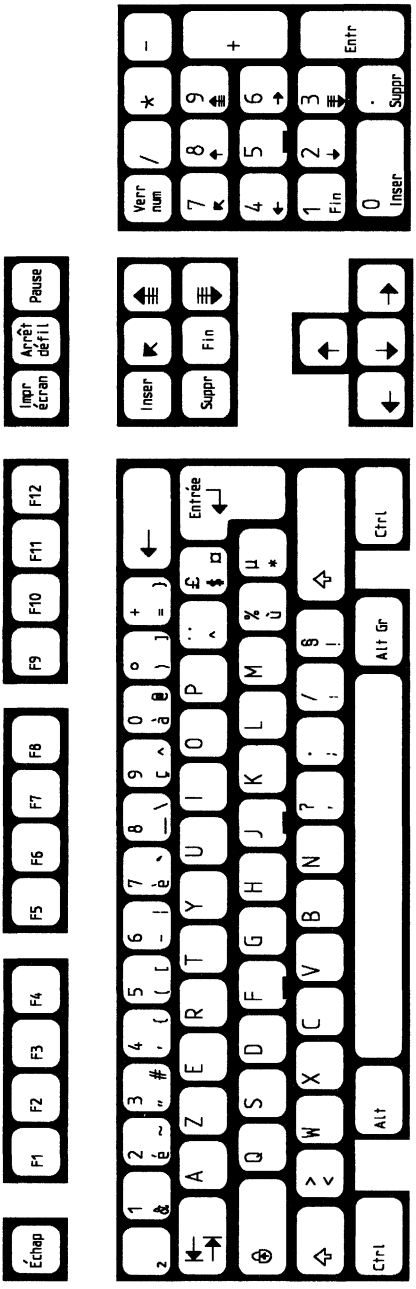
Danish Keyboard



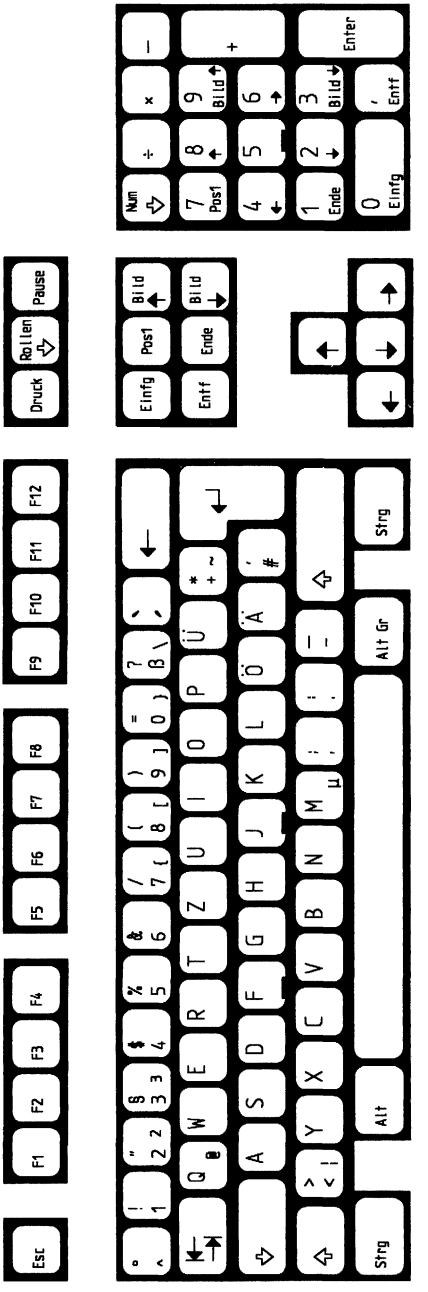
Dutch Keyboard



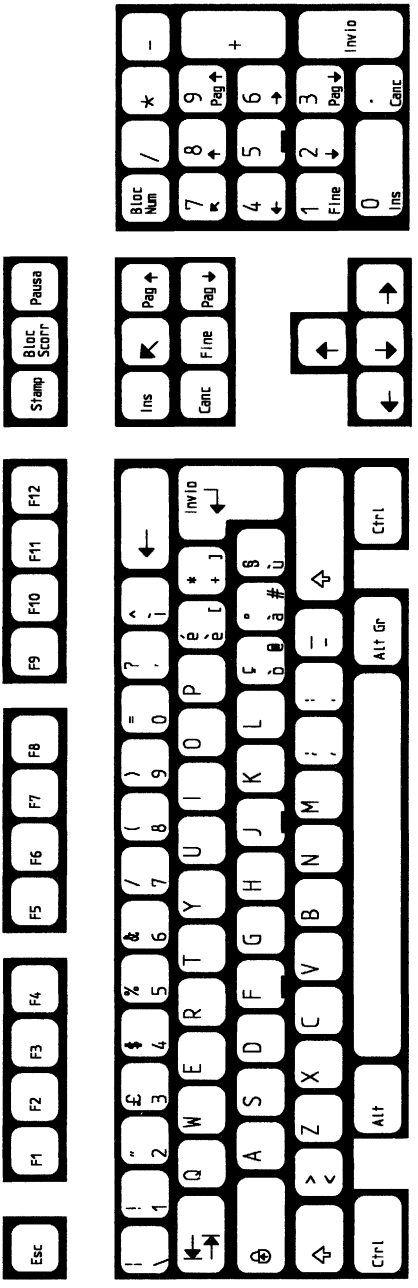
French Keyboard



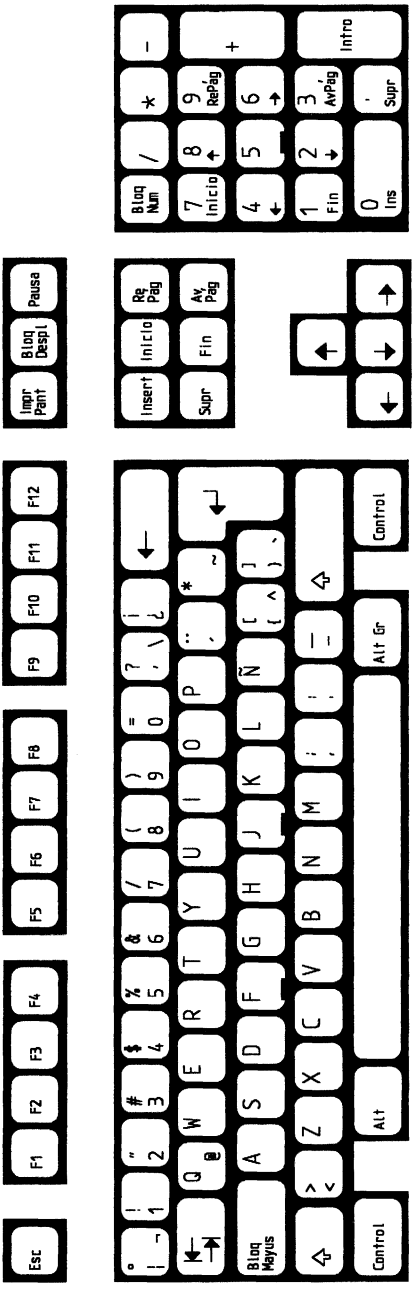
German Keyboard



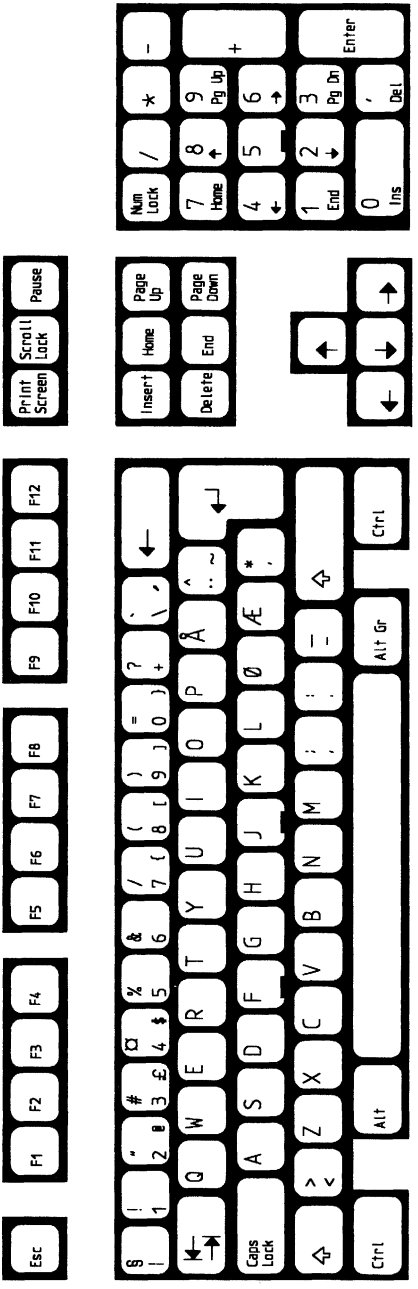
Italian Keyboard



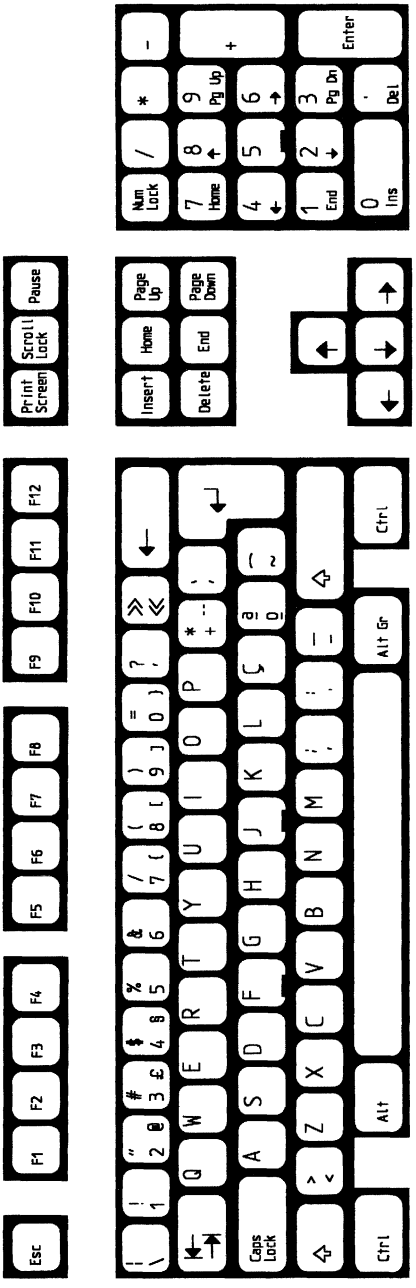
Latin American Keyboard



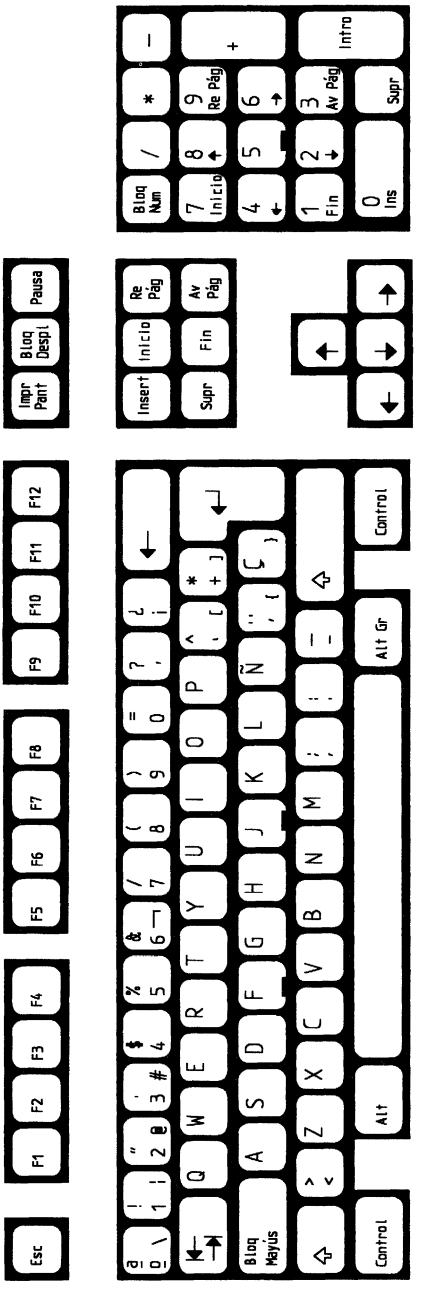
Norwegian Keyboard



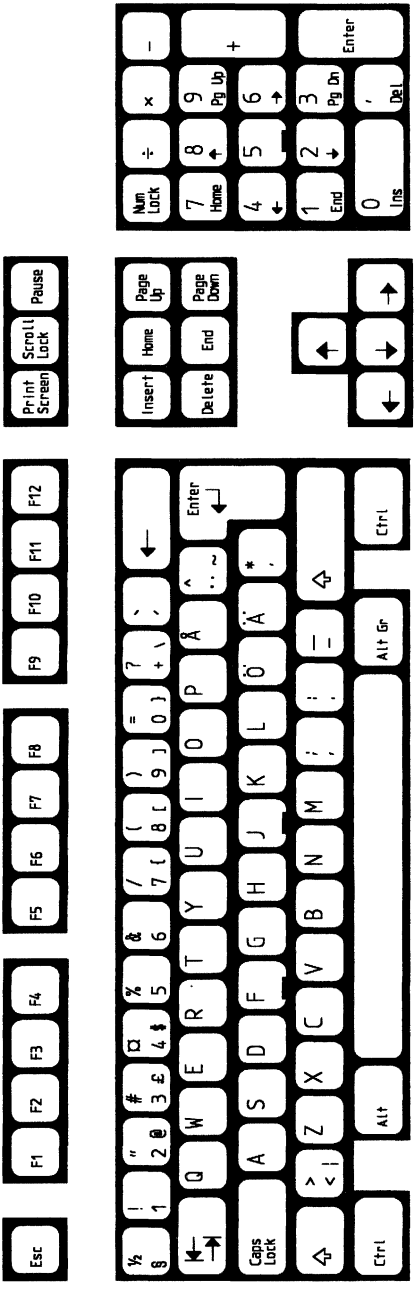
Portuguese Keyboard



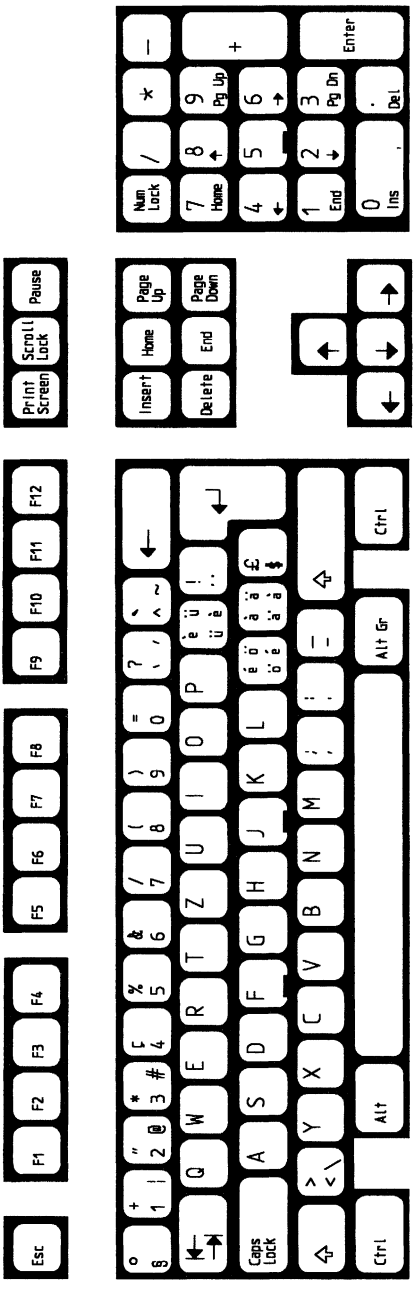
Spanish Keyboard



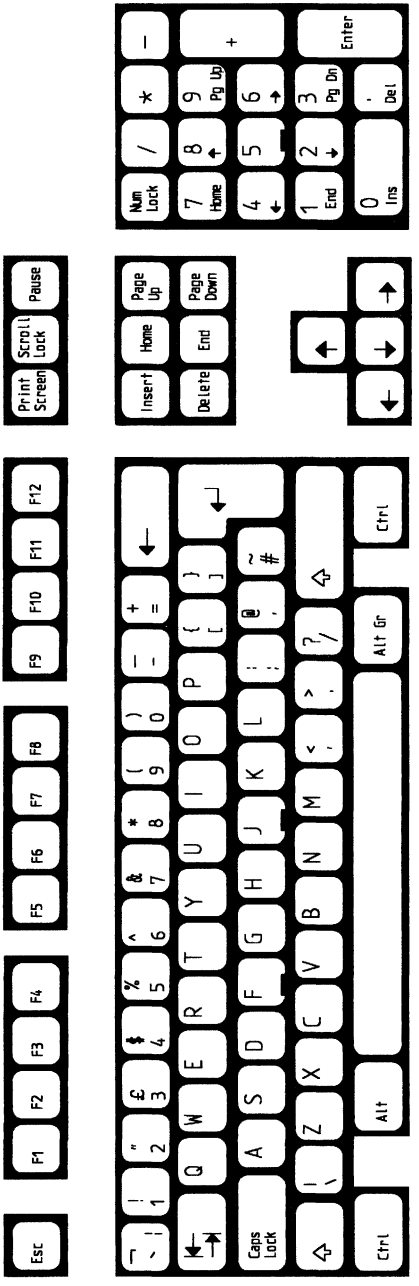
Swedish Keyboard



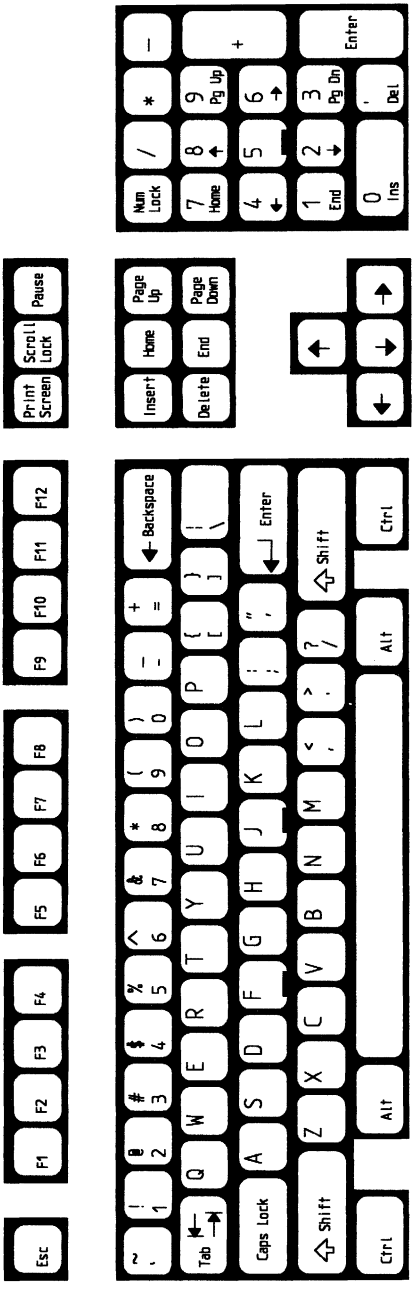
Swiss Keyboard



U.K. English Keyboard



U.S. English Keyboard



Sequential Key-Code Scanning

The keyboard detects all keys pressed, and sends each scan code in the correct sequence. When not serviced by the system, the keyboard stores the scan codes in its buffer.

Buffer

A 16-byte first-in-first-out (FIFO) buffer in the keyboard stores the scan codes until the system is ready to receive them. A buffer-overrun condition occurs when more than 16 bytes are placed in the keyboard buffer. An overrun code replaces the 17th byte. If more keys are pressed before the system allows keyboard output, the additional data is lost.

When the keyboard is allowed to send data, the bytes in the buffer are sent as in normal operation, and new data entered is detected and sent. Response codes do not occupy a buffer position.

If keystrokes generate a multiple-byte sequence, the entire sequence must fit into the available buffer space or the keystroke is discarded and a buffer-overrun condition occurs.

Keys

With the exception of the Pause key, all keys are *make/break*. The make scan code of a key is sent to the keyboard controller when the key is pressed. When the key is released, its break scan code is sent.

Additionally, except for the Pause key, all keys are *typematic*. When a key is pressed and held down, the keyboard sends the make code for that key, delays 500 milliseconds $\pm$20%, and begins sending a make code for that key at a rate of 10.9 characters per second $\pm$20%. The typematic rate and delay can be modified (see "Set Typematic Rate/Delay (hex F3)" on page 6-27).

If two or more keys are held down, only the last key pressed repeats at the typematic rate. Typematic operation stops when the last key pressed is released, even if other keys are still held down. If a key is pressed and held down while keyboard transmission is inhibited, only

the first make code is stored in the buffer. This prevents buffer overflow as a result of typematic action.

Note: Scan code set 3 allows key types to be changed by the system. See "Set 3 Scan Code Tables" on page 6-37 for the default settings.

Power-On Routine

The following activities take place when power is first applied to the keyboard.

Reset (POR)

The keyboard logic generates a 'power-on reset' signal (POR) when power is first applied to the keyboard. POR occurs a minimum of 150 milliseconds and a maximum of 2.0 seconds from the time power is first applied to the keyboard.

Basic Assurance Test

The basic assurance test (BAT) consists of a keyboard processor test, a checksum of the read-only memory (ROM), and a random-access memory (RAM) test. During the BAT, activity on the 'clock' and 'data' lines is ignored. The LEDs are turned on at the beginning and off at the end of the BAT. The BAT takes a minimum of 300 milliseconds and a maximum of 500 milliseconds. This is in addition to the time required by the POR.

Upon satisfactory completion of the BAT, a completion code (hex AA) is sent to the system, and keyboard scanning begins. If a BAT failure occurs, the keyboard sends an error code to the system. The keyboard is then disabled pending command input. Completion codes are sent between 450 milliseconds and 2.5 seconds after POR, and between 300 and 500 milliseconds after a Reset command is acknowledged.

Immediately following POR, the keyboard monitors the signals on the keyboard 'clock' and 'data' lines and sets the line protocol.

Commands from the System

The following table shows the commands that the system may send and their hexadecimal values.

Command	Hex Value
Set/Reset Status Indicators	ED
Echo	EE
Invalid Command	EF
Select Alternate Scan Codes	F0
Invalid Command	F1
Read ID	F2
Set Typematic Rate/Delay	F3
Enable	F4
Default Disable	F5
Set Default	F6
Set All Keys - Typematic	F7
- Make/Break	F8
- Make	F9
- Typematic/Make/Break	FA
Set Key Type - Typematic	FB
- Make/Break	FC
- Make	FD
Resend	FE
Reset	FF

Figure 6-1. Keyboard Commands from the System

These commands can be sent to the keyboard at any time. The keyboard responds within 20 milliseconds, except when performing the basic assurance test (BAT), or executing a Reset command.

Note: Mode 1 accepts only the 'reset' command.

The following commands are in alphabetical order. They have different meanings when issued by the keyboard (see "Commands to the System" on page 6-29).

Default Disable (hex F5): The Default Disable command resets all conditions to the power-on default state. The keyboard responds with ACK, clears its output buffer, sets the default key types (scan code set 3 operation only) and typematic rate/delay, and clears the last typematic key. The keyboard stops scanning, and awaits further instructions.

Echo (hex EE): Echo is a diagnostic aid. When the keyboard receives this command, it issues a hex EE response and, if the keyboard was previously enabled, continues scanning.

Enable (hex F4): Upon receipt of this command, the keyboard responds with ACK, clears its output buffer, clears the last typematic key, and starts scanning.

Invalid Command (hex EF and F1): Hex EF and hex F1 are invalid commands and are not supported. If one of these is sent, the keyboard does not acknowledge the command, but returns a Resend command and continues in its prior scanning state. No other activities occur.

Read ID (hex F2): This command requests identification information from the keyboard. The keyboard responds with ACK, discontinues scanning, and sends the two keyboard ID bytes. The second byte must follow completion of the first by no more than 500 microseconds. After the output of the second ID byte, the keyboard resumes scanning.

Resend (hex FE): The system sends this command when it detects an error in any transmission from the keyboard. It is sent only after a keyboard transmission and before the system allows the next keyboard output. When a Resend is received, the keyboard sends the previous output again (unless the previous output was Resend, in which case the keyboard sends the last byte before the Resend command).

Reset (hex FF): The system issues a Reset command to start a program reset and a keyboard internal self test. The keyboard acknowledges the command with an ACK and ensures the system accepts ACK before executing the command. The system signals acceptance of ACK by raising the 'clock' and 'data' lines for a minimum of 500 microseconds. The keyboard is disabled from the time it receives the Reset command until ACK is accepted, or until another command is sent that overrides the previous command.

Following acceptance of ACK, the keyboard is reinitialized and performs the BAT. After returning the completion code, the keyboard defaults to scan code set 2.

Select Alternate Scan Codes (hex F0): This command instructs the keyboard to select one of three sets of scan codes. The keyboard acknowledges receipt of this command with ACK and clears both the output buffer and the typematic key (if one is active). The system then sends the option byte and the keyboard responds with another ACK. An option byte value of hex 01 selects scan code set 1, hex 02 selects set 2, and hex 03 selects set 3.

An option byte value of hex 00 causes the keyboard to acknowledge with ACK and send a byte telling the system which scan code set is currently in use.

After establishing the new scan code set, the keyboard returns to the scanning state it was in before receiving the Select Alternate Scan Codes command.

Set All Keys (hex F7, F8, F9, FA)

These commands instruct the keyboard to set all keys to the type listed below:

Hex Value	Command
F7	Set All Keys - Typematic
F8	Set All Keys - Make/Break
F9	Set All Keys - Make
FA	Set All Keys - Typematic/Make/Break

Figure 6-2. Set All Keys Commands

The keyboard responds with ACK, clears its output buffer, sets all keys to the type indicated by the command, and continues scanning (if it was previously enabled). Although these commands can be sent using any scan code set, they affect only the operation of scan code set 3.

Set Default (hex F6): The Set Default command resets all conditions to the power-on default state. The keyboard responds with ACK, clears its output buffer, sets the default key types (scan code set 3 operation only) and typematic rate/delay, clears the last typematic key, and continues scanning.

Set Key Type (hex FB, FC, FD): These commands instruct the keyboard to set individual keys to the type listed below:

Hex Value	Command
FB	Set Key Type - Typematic
FC	Set Key Type - Make/Break
FD	Set Key Type - Make

Figure 6-3. Set Key Type Commands

The keyboard responds with ACK, clears its output buffer, and prepares to receive key identification. Key identification is accomplished by the system identifying each key by its scan code value as defined in scan code set 3. Only scan code set 3 values are valid for key identification. The type of each identified key is set to the value indicated by the command.

These commands can be sent using any scan code set, but affect only the operation of scan code set 3.

Set/Reset Status Indicators (hex ED): Three status indicators on the keyboard— Num Lock, Caps Lock, and Scroll Lock— are accessible by the system. The keyboard activates or deactivates these indicators when it receives a valid command-code sequence from the system. The command sequence begins with the command byte (hex ED). The keyboard responds to the command byte with ACK, discontinues scanning, and waits for the option byte from the system. The bit assignments for this option byte are as follows:

Bit	Function
7 - 3	Reserved (must be 0's)
2	Caps Lock Indicator
1	Num Lock Indicator
0	Scroll Lock Indicator

Figure 6-4. Set/Reset Status Indicators

If a bit for an indicator is set to 1, the indicator is turned on. If a bit is set to 0, the indicator is turned off.

The keyboard responds to the option byte with ACK, sets the indicators and, if the keyboard was previously enabled, continues scanning. The state of the indicators will reflect the bits in the option byte and can be activated or deactivated in any combination. If another command is received in place of the option byte, execution of the Set/Reset Mode Indicators command is stopped, with no change to the indicator states, and the new command is processed.

Immediately after power-on, the lights default to the Off state. If the Set Default and Default Disable commands are received, the lamps remain in the state they were in before the command was received.

Set Typematic Rate/Delay (hex F3): The system issues the Set Typematic Rate/Delay command to change the typematic rate and delay. The keyboard responds to the command with ACK, stops scanning, and waits for the system to issue the rate/delay value byte. The keyboard responds to the rate/delay value byte with another ACK, sets the rate and delay to the values indicated, and continues scanning (if it was previously enabled). Bits 6 and 5 indicate the delay, and bits 4, 3, 2, 1, and 0 (the least-significant bit) the rate. Bit 7, the most-significant bit, is always 0. The delay is equal to 1 plus the binary value of bits 6 and 5, multiplied by 250 milliseconds $\pm$20%.

The period (interval from one typematic output to the next) is determined by the following equation:

$$\text{Period} = (8 + A) \times (2^B) \times 0.00417 \text{ seconds.}$$

where:

A = binary value of bits 2, 1, and 0.

B = binary value of bits 4 and 3.

The typematic rate (make codes per second) is 1 for each period.

Bit	Typematic Rate ± 20%	Bit	Typematic Rate ± 20%
00000	30.0	10000	7.5
00001	26.7	10001	6.7
00010	24.0	10010	6.0
00011	21.8	10011	5.5
00100	20.0	10100	5.0
00101	18.5	10101	4.6
00110	17.1	10110	4.3
00111	16.0	10111	4.0
01000	15.0	11000	3.7
01001	13.3	11001	3.3
01010	12.0	11010	3.0
01011	10.9	11011	2.7
01100	10.0	11100	2.5
01101	9.2	11101	2.3
01110	8.6	11110	2.1
01111	8.0	11111	2.0

Figure 6-5. Typematic Rate

The default values for the system keyboard are as follows:

Typematic rate = 10.9 characters per second ± 20%.

Delay = 500 milliseconds ± 20%.

The execution of this command stops without change to the existing rate if another command is received instead of the rate/delay value byte.

Commands to the System

The following table shows the commands that the keyboard may send to the system, and their hexadecimal values.

Command	Hex Value
Key Detection Error/Overrun	00 (Code Sets 2 and 3)
Keyboard ID	83AB
BAT Completion Code	AA
BAT Failure Code	FC
Echo	EE
Acknowledge (ACK)	FA
Resend	FE
Key Detection Error/Overrun	FF (Code Set 1)

Figure 6-6. Keyboard Commands from the System

The commands the keyboard sends to the system are described below, in alphabetic order. They have different meanings when issued by the system.

Acknowledge (hex FA): The keyboard issues Acknowledge (ACK) to any valid input other than an Echo or Resend command. If the keyboard is interrupted while sending ACK, it discards ACK and accepts and responds to the new command.

BAT Completion Code (hex AA): Following satisfactory completion of the BAT, the keyboard sends hex AA. Any other code indicates a failure of the keyboard.

BAT Failure Code (hex FC): If a BAT failure occurs, the keyboard sends this code, discontinues scanning, and waits for a system response or reset.

Echo (hex EE): The keyboard sends this code in response to an Echo command.

Keyboard ID (hex 83AB): The Keyboard ID consists of 2 bytes, hex 83AB. The keyboard responds to the Read ID with ACK, discontinues scanning, and sends the 2 ID bytes. The low byte is sent first followed by the high byte. Following the output of the Keyboard ID, the keyboard begins scanning.

Key Detection Error (hex 00 or FF): The keyboard sends a key detection error character if conditions in the keyboard make it impossible to identify a switch closure. If the keyboard is using scan code set 1, the code is hex FF. For sets 2 and 3, the code is hex 00.

Overrun (hex 00 or FF): An overrun character is placed in the keyboard buffer and replaces the last code when the buffer capacity has been exceeded. The code is sent to the system when it reaches the top of the buffer queue. If the keyboard is using scan code set 1, the code is hex FF. For sets 2 and 3, the code is hex 00.

Resend (hex FE): The keyboard issues a Resend command following receipt of an invalid input or any input with incorrect parity. If the system sends nothing to the keyboard, no response is required.

Scan Codes

The following tables list the key numbers of the three scan code sets and their hexadecimal values. The system defaults to scan set 2, but can be switched to set 1 or set 3 (see "Select Alternate Scan Codes (hex F0)" on page 6-25).

Set 1 Scan Code Tables

In scan code set 1, each key is assigned a base scan code and, in some cases, extra codes to generate artificial shift states in the system. The typematic scan codes are identical to the base scan code for each key.

The following keys send the codes as shown, regardless of any shift states in the keyboard or the system. Refer to "Keyboard Layouts" beginning on page 6-3 to determine the character associated with each key number.

Key Number	Make Code	Break Code	Key Number	Make Code	Break Code
1	29	A9	47	2D	AD
2	02	82	48	2E	AE
3	03	83	49	2F	AF
4	04	84	50	30	B0
5	05	85	51	31	B1
6	06	86	52	32	B2
7	07	87	53	33	B3
8	08	88	54	34	B4
9	09	89	55	35	B5
10	0A	8A	57	36	B6
11	0B	8B	58	1D	9D
12	0C	8C	60	38	B8
13	0D	8D	61	39	B9
15	0E	8E	62	E0 38	E0 B8
16	0F	8F	64	E0 1D	E0 9D
17	10	90	90	45	C5
18	11	91	91	47	C7
19	12	92	92	4B	CB
20	13	93	93	4F	CF
21	14	94	96	48	C8
22	15	95	97	4C	CC
23	16	96	98	50	D0
24	17	97	99	52	D2
25	18	98	100	37	B7
26	19	99	101	49	C9
27	1A	9A	102	4D	CD
28	1B	9B	103	51	D1
29 *	2B	AB	104	53	D3
30	3A	BA	105	4A	CA
31	1E	9E	106	4E	CE
32	1F	9F	108	E0 1C	E0 9C
33	20	A0	110	01	81
34	21	A1	112	3B	BB
35	22	A2	113	3C	BC
36	23	A3	114	3D	BD
37	24	A4	115	3E	BE
38	25	A5	116	3F	BF
39	26	A6	117	40	C0
40	27	A7	118	41	C1
41	28	A8	119	42	C2
42 **	2B	AB	120	43	C3
43	1C	9C	121	44	C4
44	2A	AA	122	57	D7
45 **	56	D6	123	58	D8
46	2C	AC	125	46	C6

* 101-key keyboard only.
** 102-key keyboard only.

Figure 6-7. Keyboard Scan Codes, Set 1 (Part 1 of 5)

The remaining keys send a series of codes dependent on the state of the various shift keys (Ctrl, Alt, and Shift), and the state of Num Lock (On or Off). Because the base scan code is identical to that of another key, an extra code (hex E0) has been added to the base code to make it unique.

Key No.	Base Case, or Shift + Num Lock Make/Break	Shift Case Make/Break *	Num Lock on Make/Break
75	E0 52	E0 AA E0 52	E0 2A E0 52
	/E0 D2	/E0 D2 E0 2A	/E0 D2 E0 AA
76	E0 53	E0 AA E0 53	E0 2A E0 53
	/E0 D3	/E0 D3 E0 2A	/E0 D3 E0 AA
79	E0 4B	E0 AA E0 4B	E0 2A E0 4B
	/E0 CB	/E0 CB E0 2A	/E0 CB E0 AA
80	E0 47	E0 AA E0 47	E0 2A E0 47
	/E0 C7	/E0 C7 E0 2A	/E0 C7 E0 AA
81	E0 4F	E0 AA E0 4F	E0 2A E0 4F
	/E0 CF	/E0 CF E0 2A	/E0 CF E0 AA
83	E0 48	E0 AA E0 48	E0 2A E0 48
	/E0 C8	/E0 C8 E0 2A	/E0 C8 E0 AA
84	E0 50	E0 AA E0 50	E0 2A E0 50
	/E0 D0	/E0 D0 E0 2A	/E0 D0 E0 AA
85	E0 49	E0 AA E0 49	E0 2A E0 49
	/E0 C9	/E0 C9 E0 2A	/E0 C9 E0 AA
86	E0 51	E0 AA E0 51	E0 2A E0 51
	/E0 D1	/E0 D1 E0 2A	/E0 D1 E0 AA
89	E0 4D	E0 AA E0 4D	E0 2A E0 4D
	/E0 CD	/E0 CD E0 2A	/E0 CD E0 AA

* If the left Shift key is held down, the AA/2A shift make and break is 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.

Figure 6-8. Keyboard Scan Codes, Set 1 (Part 2 of 5)

Key No.	Scan Code Make/Break	Shift Case Make/Break *
95	E0 35/E0 B5	E0 AA E0 35/E0 B5 E0 2A

* If the left Shift key is held down, the AA/2A shift make and break is 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.

Figure 6-9. Keyboard Scan Codes, Set 1 (Part 3 of 5)

Key No.	Scan Code Make/Break	Ctrl Case, Shift Case Make/Break	Alt Case Make/Break
124	E0 2A E0 37 /E0 B7 E0 AA	E0 37/E0 B7	54/D4

Figure 6-10. Keyboard Scan Codes, Set 1 (Part 4 of 5)

Key No.	Make Code	Ctrl Key Pressed
126 *	E1 1D 45 E1 9D C5	E0 46 E0 C6

* This key is not typematic. All associated scan codes occur on the make of the key.

Figure 6-11. Keyboard Scan Codes, Set 1 (Part 5 of 5)

Set 2 Scan Code Tables

In scan code set 2, each key is assigned a unique 8-bit make scan code, which is sent when the key is pressed. Each key also sends a break code when the key is released. The break code consists of 2 bytes, the first of which is the break code prefix, hex F0; the second byte is the same as the make scan code for that key. The typematic scan code for a key is the same as the key's make code.

The following keys send the codes shown, regardless of any shift states in the keyboard or system. Refer to "Keyboard Layouts" beginning on page 6-3 to determine the character associated with each key number.

Key Number	Make Code	Break Code	Key Number	Make Code	Break Code
1	0E	F0 0E	47	22	F0 22
2	16	F0 16	48	21	F0 21
3	1E	F0 1E	49	2A	F0 2A
4	26	F0 26	50	32	F0 32
5	25	F0 25	51	31	F0 31
6	2E	F0 2E	52	3A	F0 3A
7	36	F0 36	53	41	F0 41
8	3D	F0 3D	54	49	F0 49
9	3E	F0 3E	55	4A	F0 4A
10	46	F0 46	57	59	F0 59
11	45	F0 45	58	14	F0 14
12	4E	F0 4E	60	11	F0 11
13	55	F0 55	61	29	F0 29
15	66	F0 66	62	E0 11	E0 F0 11
16	0D	F0 0D	64	E0 14	E0 F0 14
17	15	F0 15	90	77	F0 77
18	1D	F0 1D	91	6C	F0 6C
19	24	F0 24	92	6B	F0 6B
20	2D	F0 2D	93	69	F0 69
21	2C	F0 2C	96	75	F0 75
22	35	F0 35	97	73	F0 73
23	3C	F0 3C	98	72	F0 72
24	43	F0 43	99	70	F0 70
25	44	F0 44	100	7C	F0 7C
26	4D	F0 4D	101	7D	F0 7D
27	54	F0 54	102	74	F0 74
28	5B	F0 5B	103	7A	F0 7A
29 *	5D	F0 5D	104	71	F0 71
30	58	F0 58	105	7B	F0 7B
31	1C	F0 1C	106	79	F0 79
32	1B	F0 1B	108	E0 5A	E0 F0 5A
33	23	F0 23	110	76	F0 76
34	2B	F0 2B	112	05	F0 05
35	34	F0 34	113	06	F0 06
36	33	F0 33	114	04	F0 04
37	3B	F0 3B	115	0C	F0 0C
38	42	F0 42	116	03	F0 03
39	4B	F0 4B	117	0B	F0 0B
40	4C	F0 4C	118	83	F0 83
41	52	F0 52	119	0A	F0 0A
42 **	5D	F0 5D	120	01	F0 01
43	5A	F0 5A	121	09	F0 09
44	12	F0 12	122	78	F0 78
45 **	61	F0 61	123	07	F0 07
46	1A	F0 1A	125	7E	F0 7E

* 101-key keyboard only.
** 102-key keyboard only.

Figure 6-12. Keyboard Scan Codes, Set 2 (Part 1 of 5)

The remaining keys send a series of codes dependent on the state of the various shift keys (Ctrl, Alt, and Shift), and the state of Num Lock (On or Off). Because the base scan code is identical to that of another key, an extra code (hex E0) has been added to the base code to make it unique.

Key No.	Base Case, or Shift + Num Lock Make/Break	Shift Case Make/Break *	Num Lock on Make/Break
75	E0 70	E0 F0 12 E0 70	E0 12 E0 70
	/E0 F0 70	/E0 F0 70 E0 12	/E0 F0 70 E0 F0 12
76	E0 71	E0 F0 12 E0 71	E0 12 E0 71
	/E0 F0 71	/E0 F0 71 E0 12	/E0 F0 71 E0 F0 12
79	E0 6B	E0 F0 12 E0 6B	E0 12 E0 6B
	/E0 F0 6B	/E0 F0 6B E0 12	/E0 F0 6B E0 F0 12
80	E0 6C	E0 F0 12 E0 6C	E0 12 E0 6C
	/E0 F0 6C	/E0 F0 6C E0 12	/E0 F0 6C E0 F0 12
81	E0 69	E0 F0 12 E0 69	E0 12 E0 69
	/E0 F0 69	/E0 F0 69 E0 12	/EO F0 69 E0 F0 12
83	E0 75	E0 F0 12 E0 75	E0 12 E0 75
	/E0 F0 75	/E0 F0 75 E0 12	/E0 F0 75 E0 F0 12
84	E0 72	E0 F0 12 E0 72	E0 12 E0 72
	/E0 F0 72	/E0 F0 72 E0 12	/E0 F0 72 E0 F0 12
85	E0 7D	E0 FO 12 E0 7D	E0 12 E0 7D
	/E0 F0 7D	/E0 F0 7D E0 12	/E0 F0 7D E0 F0 12
86	E0 7A	E0 F0 12 E0 7A	E0 12 E0 7A
	/E0 F0 7A	/E0 F0 7A E0 12	/E0 F0 7A E0 F0 12
89	E0 74	E0 F0 12 E0 74	E0 12 E0 74
	/E0 F0 74	/E0 F0 74 E0 12	/E0 F0 74 E0 F0 12

* If the left Shift key is held down, the F0 12/12 shift make and break is sent with the other scan codes. If the right Shift key is held down, F0/59/59 is sent. If both Shift keys are down, both sets of codes are sent with the other scan code.

Figure 6-13. Keyboard Scan Codes, Set 2 (Part 2 of 5)

Key No.	Scan Code Make/Break	Shift Case Make/Break *
95	E0 4A/E0 F0 4A	E0 F0 12 4A/E0 12 F0 4A

* The left Shift key is held down, the F0 12/12 shift make and break is sent with the other scan codes. If the right Shift key is held down, F0 59/59 is sent. If both Shift keys are down, both sets of code are sent with the other scan code.

Figure 6-14. Keyboard Scan Codes, Set 2 (Part 3 of 5)

Key No.	Scan Code Make/Break	Ctrl Case, Shift Case Make/Break	Alt Case Make/Break
124	E0 12 E0 7C /E0 F0 7C E0 F0 12	E0 7C/E0 F0 7C	84/F0 84

Figure 6-15. Keyboard Scan Codes, Set 2 (Part 4 of 5)

Key No.	Make Code	Ctrl Key Pressed
126 *	E1 14 77 E1 F0 14 F0 77	E0 7E E0 F0 7E

* This key is not typematic. All associated scan codes occur on the make of the key.

Figure 6-16. Keyboard Scan Codes, Set 2 (Part 5 of 5)

Set 3 Scan Code Tables

In scan code set 3, each key is assigned a unique 8-bit make scan code, which is sent when the key is pressed. Each key also sends a break code when the key is released. The break code consists of 2 bytes, the first of which is the break-code prefix, hex F0; the second byte is the same as the make scan code for that key. The typematic scan code for a key is the same as the key's make code. With this scan code set, each key sends only one scan code, and no keys are affected by the state of any other keys.

The following keys send the codes shown, regardless of any shift states in the keyboard or system. Refer to "Keyboard Layouts" beginning on page 6-3 to determine the character associated with each key number.

Key Number	Make Code	Break Code	Default Key State
1	0E	F0 0E	Typematic
2	16	F0 16	Typematic
3	1E	F0 1E	Typematic
4	26	F0 26	Typematic
5	25	F0 25	Typematic
6	2E	F0 2E	Typematic
7	36	F0 36	Typematic
8	3D	F0 3D	Typematic
9	3E	F0 3E	Typematic
10	46	F0 46	Typematic
11	45	F0 45	Typematic
12	4E	F0 4E	Typematic
13	55	F0 55	Typematic
15	66	F0 66	Typematic
16	0D	F0 0D	Typematic
17	15	F0 15	Typematic
18	1D	F0 1D	Typematic
19	24	F0 24	Typematic
20	2D	F0 2D	Typematic
21	2C	F0 2C	Typematic
22	35	F0 35	Typematic
23	3C	F0 3C	Typematic
24	43	F0 43	Typematic
25	44	F0 44	Typematic
26	4D	F0 4D	Typematic
27	54	F0 54	Typematic
28	5B	F0 5B	Typematic
29 *	5C	F0 5C	Typematic
30	14	F0 14	Make/Break
31	1C	F0 1C	Typematic
32	1B	F0 1B	Typematic

Figure 6-17 (Part 1 of 3). Keyboard Scan Codes, Set 3

Key Number	Make Code	Break Code	Default Key State
33	23	F0 23	Typematic
34	2B	F0 2B	Typematic
35	34	F0 34	Typematic
36	33	F0 33	Typematic
37	3B	F0 3B	Typematic
38	42	F0 42	Typematic
39	4B	F0 4B	Typematic
40	4C	F0 4C	Typematic
41	52	F0 52	Typematic
42 **	53	F0 53	Typematic
43	5A	F0 5A	Typematic
44	12	F0 12	Make/Break
45 **	13	F0 13	Typematic
46	1A	F0 1A	Typematic
47	22	F0 22	Typematic
48	21	F0 21	Typematic
49	2A	F0 2A	Typematic
50	32	F0 32	Typematic
51	31	F0 31	Typematic
52	3A	F0 3A	Typematic
53	41	F0 41	Typematic
54	49	F0 49	Typematic
55	4A	F0 4A	Typematic
57	59	F0 59	Make/Break
58	11	F0 11	Make/Break
60	19	F0 19	Make/Break
61	29	F0 29	Typematic
62	39	F0 39	Make only
64	58	F0 58	Make only
75	67	F0 67	Make only
76	64	F0 64	Typematic
79	61	F0 61	Typematic
80	6E	F0 6E	Make only
81	65	F0 65	Make only
83	63	F0 63	Typematic
84	60	F0 60	Typematic
85	6F	F0 6F	Make only
86	6D	F0 6D	Make only
89	6A	F0 6A	Typematic
90	76	F0 76	Make only
91	6C	F0 6C	Make only
92	6B	F0 6B	Make only
93	69	F0 69	Make only
95	77	F0 77	Make only
96	75	F0 75	Make only
97	73	F0 73	Make only
98	72	F0 72	Make only
99	70	F0 70	Make only
100	7E	F0 7E	Make only
101	7D	F0 7D	Make only
102	74	F0 74	Make only
103	7A	F0 7A	Make only
104	71	F0 71	Make only

Figure 6-17 (Part 2 of 3). Keyboard Scan Codes, Set 3

Key Number	Make Code	Break Code	Default Key State
105	84	F0 84	Make only
106	7C	F0 7C	Typematic
108	79	F0 79	Make only
110	08	F0 08	Make only
112	07	F0 07	Make only
113	0F	F0 0F	Make only
114	17	F0 17	Make only
115	1F	F0 1F	Make only
116	27	F0 27	Make only
117	2F	F0 2F	Make only
118	37	F0 37	Make only
119	3F	F0 3F	Make only
120	47	F0 47	Make only
121	4F	F0 4F	Make only
122	56	F0 56	Make only
123	5E	F0 5E	Make only
124	57	F0 57	Make only
125	5F	F0 5F	Make only
126	62	F0 62	Make only

* 101-key keyboard only.
** 102-key keyboard only.

Figure 6-17 (Part 3 of 3). Keyboard Scan Codes, Set 3

Clock and Data Signals

The keyboard and system communicate over the 'clock' and 'data' lines. The source of each of these lines is an open-collector device on the keyboard that allows either the keyboard or the system to force a line to an inactive (low) level. When no communication is occurring, the 'clock' line is at an active (high) level. The state of the 'data' line is held active (high) by the keyboard.

When the system sends data to the keyboard, it forces the 'data' line to an inactive level and allows the 'clock' line to go to an active level.

An inactive signal will have a value of at least 0, but not greater than +0.7 volts. A signal at the inactive level is a logical 0. An active signal will have a value of at least +2.4, but not greater than +5.5 volts. A signal at the active level is a logical 1. Voltages are measured between a signal source and the dc network ground.

When the keyboard sends data to, or receives data from the system, it generates the 'clock' signal to time the data. The system can prevent the keyboard from sending data by forcing the 'clock' line to an inactive level; the 'data' line may be active or inactive during this time.

During the BAT, the keyboard allows the 'clock' and 'data' lines to go to an active level.

Data Stream

Data transmissions to and from the keyboard consist of an 11-bit data stream (Mode 2) sent serially over the 'data' line. The following table shows the functions of the bits.

Bit	Function
11	Stop bit (always 1)
10	Parity bit (odd parity)
9	Data bit 7 (most-significant)
8	Data bit 6
7	Data bit 5
6	Data bit 4
5	Data bit 3
4	Data bit 2
3	Data bit 1
2	Data bit 0 (least-significant)
1	Start bit (always 0)

Figure 6-18. Keyboard Data Stream Bit Definitions

The parity bit is either 1 or 0, and the 8 data bits, plus the parity bit, always have an odd number of 1's.

Note: Mode 1 is a 9-bit data stream that does not have a parity bit or stop bit and the start bit is always 1.

Data Output

When the keyboard is ready to send data, it first checks for a keyboard-inhibit or system request-to-send status on the 'clock' and 'data' lines. If the 'clock' line is inactive (low), data is stored in the keyboard buffer. If the 'clock' line is active (high) and the 'data' line is inactive (request-to-send), data is stored in the keyboard buffer, and the keyboard receives system data.

If the 'clock' and 'data' lines are both active, the keyboard sends the 0 start bit, 8 data bits, the parity bit, and the stop bit. Data will be valid before the trailing edge and beyond the leading edge of the clock pulse. During transmission, the keyboard checks the 'clock' line for an active level at least every 60 milliseconds. If the system lowers the 'clock' line from an active level after the keyboard starts sending data, a condition known as *line contention* occurs, and the keyboard stops sending data. If line contention occurs before the leading edge of the 10th clock signal (parity bit), the keyboard buffer returns the 'clock' and 'data' lines to an active level. If contention does not occur by the 10th clock signal, the keyboard completes the transmission. Following line contention, the system may or may not request the keyboard to resend the data.

Following a transmission, the system can inhibit the keyboard until the system processes the input, or until it requests that a response be sent.

Data Input

When the system is ready to send data to the keyboard, it first checks to see if the keyboard is sending data. If the keyboard is sending, but has not reached the 10th 'clock' signal, the system can override the keyboard output by forcing the keyboard 'clock' line to an inactive (low) level. If the keyboard transmission is beyond the 10th 'clock' signal, the system must receive the transmission.

If the keyboard is not sending, or if the system elects to override the keyboard's output, the system forces the keyboard 'clock' line to an inactive level for more than 60 microseconds while preparing to send data. When the system is ready to send the start bit (the 'data' line will be inactive), it allows the 'clock' line to go to an active (high) level.

The keyboard checks the state of the 'clock' line at intervals of no more than 10 milliseconds. If a system request-to-send (RTS) is detected, the keyboard counts 11 bits. After the 10th bit, the keyboard checks for an active level on the 'data' line, and if the line is active, forces it inactive, and counts one more bit. This action signals the system that the keyboard has received its data. Upon receipt of this signal, the system returns to a ready state, in which it can accept keyboard output, or goes to the inhibited state until it is ready.

If the keyboard 'data' line is found at an inactive level following the 10th bit, a framing error has occurred, and the keyboard continues to count until the 'data' line becomes active. The keyboard then makes the 'data' line inactive and sends a Resend.

Each system command or data transmission to the keyboard requires a response from the keyboard before the system can send its next output. The keyboard will respond within 20 milliseconds unless the system prevents keyboard output. If the keyboard response is invalid or has a parity error, the system sends the command or data again. However, the two byte commands require special handling. If hex F3 (Set Typematic Rate/Delay), hex F0 (Select Alternate Scan Codes), or hex ED (Set/Reset Mode Indicators) have been sent and acknowledged, and the value byte has been sent but the response is

invalid or has a parity error, the system will resend both the command and the value byte.

Encode and Usage

The keyboard routine, provided by IBM in the ROM BIOS, is responsible for converting the keyboard scan codes into what is called *Extended ASCII*. The extended ASCII codes returned by the ROM routine are mapped to the U.S. English keyboard layout. Some operating systems may make provisions for alternate keyboard layouts by providing an interrupt replacement routine, which resides in the read/write memory. This section discusses only the ROM routine.

Extended ASCII encompasses 1-byte character codes, with possible values of 0 to 255, an extended code for certain extended keyboard functions, and functions handled within the keyboard routine or through interrupts.

The character codes are passed through the BIOS keyboard routine to the system or application program. A "-1" means the combination is suppressed in the keyboard routine. The codes are returned in the AL register. See Section 8, "Characters and Keystrokes" for the exact codes.

Key	Base Case	Uppercase	Ctrl	Alt
1	`	~	-1	(*)
2	1	!	-1	(*)
3	2	@	Null(000) (*)	(*)
4	3	#	-1	(*)
5	4	$	-1	(*)
6	5	%	-1	(*)
7	6	^	RS(030)	(*)
8	7	&	-1	(*)
9	8	*	-1	(*)
10	9	(	-1	(*)
11	0	)	-1	(*)
12	-	_	US(031)	(*)
13	=	+	-1	(*)
15	Backspace (008)	Backspace (008)	Del(127)	(*)
16	→| (009)	|← (*)	(*)	(*)
17	q	Q	DC1(017)	(*)
18	w	W	ETB(023)	(*)
19	e	E	ENQ(005)	(*)
20	r	R	DC2(018)	(*)
21	t	T	DC4(020)	(*)
22	y	Y	EM(025)	(*)
23	u	U	NAK(021)	(*)
24	i	I	HT(009)	(*)
25	o	O	SI(015)	(*)
26	p	P	DLE(016)	(*)
27	[	{	Esc(027)	(*)
28	]	}	GS(029)	(*)
29	\	|	FS(028)	(*)
30 Caps Lock	-1	-1	-1	-1
31	a	A	SOH(001)	(*)
32	s	S	DC3(019)	(*)
33	d	D	EOT(004)	(*)
34	f	F	ACK(006)	(*)
35	g	G	BEL(007)	(*)
36	h	H	BS(008)	(*)
37	j	J	LF(010)	(*)
38	k	K	VT(011)	(*)
39	l	L	FF(012)	(*)
40	;	:	-1	(*)
41	'	"	-1	(*)
43	CR(013)	CR(013)	LF(010)	(*)
44 Shift (Left)	-1	-1	-1	-1

Figure 6-19 (Part 1 of 2). Character Codes

Key	Base Case	Uppercase	Ctrl	Alt
46	z	Z	SUB(026)	(*)
47	x	X	CAN(024)	(*)
48	c	C	ETX(003)	(*)
49	v	V	SYN(022)	(*)
50	b	B	STX(002)	(*)
51	n	N	SO(014)	(*)
52	m	M	CR(013)	(*)
53	,	<	-1	(*)
54	.	>	-1	(*)
55	/	?	-1	(*)
57 Shift (Right)	-1	-1	-1	-1
58 Ctrl (Left)	-1	-1	-1	-1
60 Alt (Left)	-1	-1	-1	-1
61	Space	Space	Space	Space
62 Alt (Right)	-1	-1	-1	-1
64 Ctrl (Right)	-1	-1	-1	-1
90 Num Lock	-1	-1	-1	-1
95	/	/	(*)	(*)
100	*	*	(*)	(*)
105	-	-	(*)	(*)
106	+	+	(*)	(*)
108	Enter	Enter	LF(010)	(*)
110	Esc	Esc	Esc	(*)
112	Null (*)	Null (*)	Null (*)	Null(*)
113	Null (*)	Null (*)	Null (*)	Null(*)
114	Null (*)	Null (*)	Null (*)	Null(*)
115	Null (*)	Null (*)	Null (*)	Null(*)
116	Null (*)	Null (*)	Null (*)	Null(*)
117	Null (*)	Null (*)	Null (*)	Null(*)
118	Null (*)	Null (*)	Null (*)	Null(*)
119	Null (*)	Null (*)	Null (*)	Null(*)
120	Null (*)	Null (*)	Null (*)	Null(*)
121	Null (*)	Null (*)	Null (*)	Null(*)
122	Null (*)	Null (*)	Null (*)	Null(*)
123	Null (*)	Null (*)	Null (*)	Null(*)
125 Scroll Lock	-1	-1	-1	-1
126	Pause(**)	Pause(**)	Break(**)	Pause(**)

(*) Refer to "Extended Functions" on page 6-46.
(**) Refer to "Special Handling" on page 6-50.

Figure 6-19 (Part 2 of 2). Character Codes

The following figure is a list of keys that have meaning only in Num Lock, Shift, or Ctrl states.

The Shift key temporarily reverses the current Num Lock state.

Key	Num Lock	Base Case	Alt	Ctrl
91	7	Home (*)	-1	Clear Screen
92	4	← (*)	-1	Reverse Word(*)
93	1	End (*)	-1	Erase to EOL(*)
96	8	↑ (*)	-1	(*)
97	5	(*)	-1	(*)
98	2	↓ (*)	-1	(*)
99	0	Ins	-1	(*)
101	9	Page Up (*)	-1	Top of Text and Home
102	6	→ (*)	-1	Advance Word (*)
103	3	Page Down (*)	-1	Erase to EOS (*)
104	.	Delete (*,**)	(**)	(**)
105	-	Sys Request	-1	-1
106	+	+ (*)	-1	-1

(*) Refer to "Extended Functions."
(**) Refer to "Special Handling" on page 6-50.

Figure 6-20. Special Character Codes

Extended Functions

For certain functions that cannot be represented by a standard ASCII code, an extended code is used. A character code of 000 (null) is returned in AL. This indicates that the system or application program should examine a second code, which will indicate the actual function. Usually, but not always, this second code is the scan code of the primary key that was pressed. This code is returned in AH.

The following figure is a list of the extended codes and their functions.

Second Code	Function
1	Alt Esc
3	Null Character
14	Alt Backspace
15	⇤ (Back-tab)
16-25	Alt Q, W, E, R, T, Y, U, I, O, P
26-28	Alt [ ] ↵
30-38	Alt A, S, D, F, G, H, J, K, L
39-41	Alt ; ′ ′
43	Alt \
44-50	Alt Z, X, C, V, B, N, M
51-53	Alt , . /
55	Alt Keypad *
59-68	F1 to F10 Function Keys (Base Case)
71	Home
72	↑ (Cursor Up)
73	Page Up
74	Alt Keypad -
75	← (Cursor Left)
76	Center Cursor
77	→ (Cursor Right)
78	Alt Keypad +
79	End
80	↓ (Cursor Down)
81	Page Down
82	Ins (Insert)
83	Del (Delete)
84-93	Shift F1 to F10
94-103	Ctrl F1 to F10
104-113	Alt F1 to F10
114	Ctrl PrtSc (Start/Stop Echo to Printer)
115	Ctrl ← (Reverse Word)
116	Ctrl → (Advance Word)
117	Ctrl End (Erase to End of Line-EOL)
118	Ctrl PgDn (Erase to End of Screen-EOS)
119	Ctrl Home (Clear Screen and Home)
120-131	Alt 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, -, = keys 2-13
132	Ctrl PgUp (Top 25 Lines of Text and Cursor Home)
133-134	F11, F12
135-136	Shift F11, F12
137-138	Ctrl F11, F12
139-140	Alt F11, F12
141	Ctrl Up/8
142	Ctrl Keypad -
143	Ctrl Keypad 5
144	Ctrl Keypad +
145	Ctrl Down/2
146	Ctrl Ins/0
147	Ctrl Del/.
148	Ctrl Tab

Figure 6-21 (Part 1 of 2). Keyboard Extended Functions

Second Code	Function
149	Ctrl Keypad /
150	Ctrl Keypad *
151	Alt Home
152	Alt Up
153	Alt Page Up
155	Alt Left
157	Alt Right
159	Alt End
160	Alt Down
161	Alt Page Down
162	Alt Insert
163	Alt Delete
164	Alt Keypad /
165	Alt Tab
166	Alt Enter

Figure 6-21 (Part 2 of 2). Keyboard Extended Functions

Shift States

Most shift states are handled within the keyboard routine, and are not apparent to the system or application program. In any case, the current status of active shift states is available by calling an entry point in the BIOS keyboard routine. The following keys result in altered shift states:

Shift: This key temporarily shifts keys 1 through 13, 16 through 29, 31 through 41, and 46 through 55, to uppercase (base case if in Caps Lock state). Also, the Shift temporarily reverses the Num Lock or non-Num Lock state of keys 91 through 93, 96, 98, 99, and 101 through 104.

Ctrl: This key temporarily shifts keys 3, 7, 12, 15 through 29, 31 through 39, 43, 46 through 52, 75 through 89, 91 through 93, 95 through 108, 112 through 124, and 126 to the Ctrl state. The Ctrl key is also used with the Alt and Del keys to cause the system-reset function, with the Scroll Lock key to cause the break function, and with the Num Lock key to cause the pause function. The system-reset, break, and pause functions are described under "Special Handling" on page 6-50.

Alt: This key temporarily shifts keys 1 through 29, 31 through 43, 46 through 55, 75 through 89, 95, 100, and 105 through 124 to the Alt state. The Alt key is also used with the Ctrl and Del keys to cause a system reset.

The Alt key also allows the user to enter any character code from 1 to 255. The user holds down the Alt key and types the decimal value of the characters desired on the numeric keypad (keys 91 through 93, 96 through 99, and 101 through 103). The Alt key is then released. If the number is greater than 255, a modulo-256 value is used. This value is interpreted as a character code and is sent through the keyboard routine to the system or application program. Alt is handled internal to the keyboard routine.

Caps Lock: This key shifts keys 17 through 26, 31 through 39, and 46 through 52 to uppercase. When Caps Lock is pressed again, it reverses the action. Caps Lock is handled internal to the keyboard routine. When Caps Lock is pressed, it changes the Caps Lock Mode indicator. If the indicator was on, it will go off; if it was off, it will go on.

Scroll Lock: When interpreted by appropriate application programs, this key indicates that the cursor-control keys will cause windowing over the text rather than moving the cursor. When the Scroll Lock key is pressed again, it reverses the action. The keyboard routine simply records the current shift state of the Scroll Lock key. It is the responsibility of the application program to perform the function. When Scroll Lock is pressed, it changes the Scroll Lock Mode indicator. If the indicator was on, it will go off; if it was off, it will go on.

Num Lock: This key shifts keys 91 through 93, 96 through 99, and 101 through 104 to uppercase. When Num Lock is pressed again, it reverses the action. Num Lock is handled internal to the keyboard routine. When Num Lock is pressed, it changes the Num Lock Mode indicator. If the indicator was on, it will go off; if it was off, it will go on.

Shift Key Priorities and Combinations: If combinations of the Alt, Ctrl, and Shift keys are pressed and only one is valid, the priority is as follows: the Alt key is first, the Ctrl key is second, and the Shift key is third. The only valid combination is Alt and Ctrl, which is used in the system-reset function.

Special Handling

System Reset

The combination of any Alt, Ctrl, and Del keys results in the keyboard routine that starts a system reset or restart. System reset is handled by BIOS.

Break

The combination of the Ctrl and Pause/Break keys results in the keyboard routine signaling interrupt hex 1B. The extended characters AL = hex 00, and AH = hex 00 are also returned.

Pause

The Pause key causes the keyboard interrupt routine to loop, waiting for any character or function key to be pressed. This provides a method of temporarily suspending an operation, such as listing or printing, and then resuming the operation. The method is not apparent to either the system or the application program. The key stroke used to resume operation is discarded. Pause is handled internal to the keyboard routine.

Print Screen

The Print Screen key results in an interrupt invoking the print-screen routine. This routine works in the alphanumeric or graphics mode, with unrecognizable characters printing as blanks.

System Request

When the System Request (Alt and Print Screen) key is pressed, a hex 8500 is placed in AX, and an interrupt hex 15 is executed. When the SysRq key is released, a hex 8501 is placed in AX, and another interrupt hex 15 is executed. If an application is to use System Request, the following rules must be observed:

Save the previous address.

Overlay interrupt vector hex 15.

Check AH for a value of hex 85:

If yes, process may begin.
If no, go to previous address.

The application program must preserve the value in all registers, except AX, upon return. System Request is handled internal to the keyboard routine.

Other Characteristics

The keyboard routine does its own buffering, and the keyboard buffer is large enough to support entries by a fast typist. However, if a key is pressed when the buffer is full, the key will be ignored and the alarm will sound.

The keyboard routine also suppresses the typematic action of the following keys: Ctrl, Shift, Alt, Num Lock, Scroll Lock, Caps Lock, and Ins.

During each interrupt hex 09 from the keyboard, an interrupt hex 15, function (AH) = hex 4F is generated by the BIOS after the scan code is read from the keyboard adapter. The scan code is passed in the (AL) register with the carry flag set. This is to allow an operating system to intercept each scan code prior to its being handled by the interrupt hex 09 routine, and have a chance to change or act on the scan code. If the carry flag is changed to 0 on return from interrupt hex 15, the scan code will be ignored by the interrupt handler.

Cables and Connectors

The keyboard cable connects to the system with a 6-pin miniature DIN connector, and to the keyboard with a 6-position connector. The following table shows the pin configuration and signal assignments.

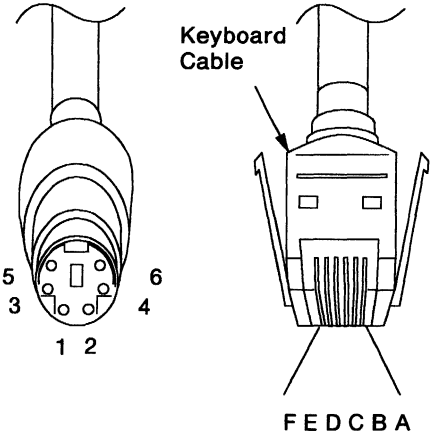


Figure 6-22. Keyboard Cable Connectors

DIN Connector Pins	Signal Name	Connector Pins
1	+KBD DATA	B
2	Reserved	F
3	Ground	C
4	+5.0 Vdc	E
5	+KBD CLK	D
6	Reserved	A
Shield	Frame Ground	Shield

Figure 6-23. Keyboard Connectors Signal and Voltage Assignments

Specifications

The specifications for the keyboard follow.

Power Requirements

- +5 Vdc ± 10%
- 275 mA

Size

- Length: 492 millimeters (19.4 inches)
- Depth: 210 millimeters (8.3 inches)
- Height: 58 millimeters (2.3 inches), legs extended.

Weight

- 2.25 kilograms (5.0 pounds)

Notes:

Section 7. Instruction Sets

Introduction to the 80386 Instruction Set

The 80386 instruction set is an extended version of the 8086 and 80286 instruction sets. The instruction sets have been extended in two ways:

- The instructions have extensions that allow operations on 32-bit wide operands, registers, and memory.
- A new 32-bit addressing mode allows flexible selection of registers for base and index as well as index scaling capabilities (x2, x4, x8) for computing a 32-bit effective address. The 32-bit effective address yields a 4G address range.

  Note: The effective address size must be less than 64K in the Real or Virtual Address Modes to avoid an exception.

Code and Data Segment Descriptors

Although the 80386 supports all 80286 code and data segment descriptors, there are some differences in the format. The 80286 segment descriptors contain a 24-bit base address and a 16-bit limit field, while the 80386 segment descriptors have a 32-bit base address, a 20-bit limit field, a default bit, and a granularity bit. The following figure shows the format for the 80386 descriptor.

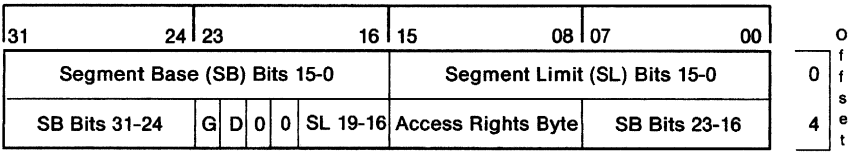


Figure 7-1. 80386 Code and Data Segment Descriptor

Note: Bits 31 through 16 shown at offset 4 are set to 0 for all 80286 segment descriptors.

The default (D) bit of the code segment register is used to determine whether the instruction is carried out as a 16-bit or 32-bit instruction. Code segment descriptors are not used in either the Real Address Mode or the Virtual 8086 Mode. When the system unit microprocessor is operating in either the Real Address Mode or

Virtual 8086 Mode, a D-bit value of 0 is assumed and operations default to a 16-bit length compatible with 8086 and 80286 programs.

The granularity (G) bit is used to determine the granularity of the segment length (1 = page granular, 0 = byte granular). If the value of the 20 segment-limit bits is defined as *N*, a G-bit value of 1 defines the segment size as follows:

$$\text{Segment size} = (N + 1) \times 4K$$

4K represents the size of a page.

Prefixes

Two prefixes have been added to the instruction set; the Operand Size Prefix overrides the default selection of the operand size and the Effective Address Size Prefix overrides the effective address size. The presence of either prefix toggles the default setting to its opposite condition. For example:

- If the operand size defaults to 32-bit data operations, the presence of the Operand Size Prefix sets it for 16-bit data operation.
- If the effective address size is 16-bits, the presence of the Effective Address Size Prefix toggles the instruction to use 32-bit effective address computations.

The prefixes are available in all 80386 modes, including the Real Address Mode and the Virtual 8086 Mode. Since the default of these modes is always 16 bits, the prefixes are used to specify 32-bit operation. If needed, either one or both of the prefixes may precede any opcode bytes and affect only the instruction they precede.

Instruction Format

The instructions are presented in the following format.

Opcode	Mode Specifier	Address Displacement	Immediate Data

Term	Description
Opcode	The opcode may be one or two bytes in length. Within each byte, smaller encoding fields may be defined.
Mode Specifier	Consists of the "mod r/m" byte and the "scale-index-base" (s-i-b) byte. The mod r/m byte specifies the address mode to be used. Format: mod T T T r/m The "s-i-b" byte is optional and can be used only in 32-bit address modes. It follows the mod r/m byte to fully specify the manner in which the effective address is computed. Format: ss index base
Address Displacement	Follows the "mod r/m" byte or "s-i-b" byte. It may be 8, 16, or 32 bits in length.
Immediate Data	If specified, follows any displacement bytes and becomes the last field of the instruction. It may be 8, 16, or 32 bits in length. The term "8-bit data" indicates a fixed data length of 8 bits. The term "8-, 16-, or 32-bit data" indicates a variable data length. The length is determined by the w field and the current operand size. If w = 0, the data is always 8 bits. If w = 1, the size is determined by the operand size of the instruction.

Figure 7-2. Instruction Format

The instructions use a variety of fields to indicate register selection, the addressing mode, and so on. The following figure is a summary of the most commonly used fields.

Field Name	Description	Bits
w	Specifies if data is byte or full size. (Full size is either 16 or 32 bits.)	1
d	Specifies the direction of data operation.	1
s	Specifies if an immediate data field must be sign-extended.	1
reg	General address 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 an index register.	3
base	General register to be used as base register.	3
sreg2	Segment register specifier for CS, SS, DS, and ES.	2
sreg3	Segment register specifier for CS, SS, DS, ES, FS, and GS.	3
tttn	For conditional instructions; specifies a condition asserted or a condition negated.	4

Figure 7-3. Instruction Set Encoding Field Summary

Encoding

This section defines the encoding of the fields used in the instruction sets.

Address Mode

The first addressing byte is the "mod r/m" byte. The effective address (EA) of the memory operand is computed according to the mod and r/m fields. The mod r/m byte can be interpreted as either a 16-bit or 32-bit addressing mode specifier. Interpretation of the byte depends on the address components used to calculate the EA. The following defines the encoding of 16-bit and 32-bit addressing modes with the mod r/m byte.

mod r/m	16-Bit Mode	32-Bit Mode (No s-i-b byte)
00 000	DS:[BX + SI]	DS:[EAX]
00 001	DS:[BX + DI]	DS:[ECX]
00 010	SS:[BP + SI]	DS:[EDX]
00 011	SS:[BP + DI]	DS:[EBX]
00 100	DS:[SI]	s-i-b present (see Figure 7-8 on page 7-10)
00 101	DS:[DI]	d32
00 110	d16	DS:[ESI]
00 111	DS:[BX]	DS:[EDI]
01 000	DS:[BX + SI + d8]	DS:[EAX + d8]
01 001	DS:[BX + DI + d8]	DS:[ECX + d8]
01 010	SS:[BP + SI + d8]	DS:[EDX + d8]
01 011	SS:[BP + DI + d8]	DS:[EBX + d8]
01 100	DS:[SI + d8]	s-i-b present (see Figure 7-8 on page 7-10)
01 101	DS:[DI + d8]	SS:[EBP + d8]
01 110	SS:[BP + d8]	DS:[ESI + d8]
01 111	DS:[BX + d8]	DS:[EDI + d8]
10 000	DS:[BX + SI + d16]	DS:[EAX + d32]
10 001	DS:[BX + DI + d16]	DS:[ECX + d32]
10 010	DS:[BP + SI + d16]	DS:[EDX + d32]
10 011	DS:[BP + DI + d16]	DS:[EBX + d32]
10 100	DS:[SI + d16]	s-i-b present (see Figure 7-8 on page 7-10)
10 101	DS:[DI + d16]	DS:[EBP + d32]
10 110	DS:[BP + d16]	DS:[ESI + d32]
10 111	DS:[BX + d16]	DS:[EDI + d32]

Figure 7-4. Effective Address (16-Bit and 32-Bit Address Modes)

The displacement follows the second byte of the instruction (before data if required).

The scale-index-base (s-i-b) byte can be specified as a second byte of addressing information. The s-i-b byte is specified when using a 32-bit addressing mode and the mod r/m byte has the following values:

- r/m = 100
- mod = 00, 01 , or 10.

When the s-i-b byte is present, the 32-bit effective address is a function of the mod, ss, index, and base fields. The following figures show the scale factor, index register selected, and base register selected when the s-i-b byte is present.

ss	Scale Factor
00	x 1
01	x 2
10	x 4
11	x 8

Figure 7-5. Scale Factor (s-i-b Byte Present)

index	Index Register	
000	EAX	
001	ECX	
010	EDX	
011	EBX	
100	No Index Register	The ss field must equal 00 when the index field is 100; if not, the effective address is undefined.
101	EBP	
110	ESI	
111	EDI	

Figure 7-6. Index Registers (s-i-b Byte Present)

base	Base Register	
000	EAX	
001	ECX	
010	EDX	
011	EBX	
100	ESP	
101	EBP	If mod = 00, then EBP is not used to form the EA; immediate 32-bit address displacement follows the mode specifier byte.
110	ESI	
111	EDI	

Figure 7-7. Base Registers (s-i-b Byte Present)

The "scaled index" information is determined by multiplying the contents of the index register by the scale factor. The following example shows the use of the 32-bit addressing mode with scaling where:

- EAX is the base of ARRAY_A
- ECX is the index of the desired element
- 2 is the scale factor.

```
; ARRAY_A is an array of words
MOV EAX, offset ARRAY_A
MOV ECX, element_number
MOV BX, [EAX][ECX*2]
```

The following defines the encoding of the 32-bit addressing mode when the s-i-b byte is present.

Note: The mod field is from the mod r/m byte. The base field and scaled-index information are from the s-i-b byte.

mod base	32-Bit Address Mode
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]

Figure 7-8. Effective Address (32-Bit Address Mode — s-i-b Byte Present)

Operand Length (w) Field

For any given instruction performing a data operation, the instruction is executed as either a 32-bit or 16-bit operation. Within the constraints of the operation size, the w field encodes the operand size as either one byte or full operation.

w	16-Bit Data Operation	32-Bit Data Operation
0	8 Bits	8 Bits
1	16 Bits	32 Bits

Figure 7-9. Operand Length Field Encoding

Segment Register (sreg) Field

The 2-bit segment register field (sreg2) allows one of the four 80286 segment registers to be specified. The 3-bit segment register (sreg3) allows the 80386 FS and GS segment registers to be specified. The sreg fields are defined in the following figure.

sreg2	sreg3	Segment Register
00	000	ES
01	001	CS
10	010	SS
11	011	DS
--	100	FS
--	101	GS
--	110	Reserved
--	111	Reserved

Figure 7-10. Segment Register Field Encoding

General Register (reg) Field

The general register is specified by the reg field, which may appear in the primary opcode bytes, as the reg field of the mod reg r/m byte, or as the r/m field of the mod reg r/m byte when mod = 11.

reg	16-Bit (without w)	16-Bit (w = 0)	16-Bit (w = 1)	32-Bit (without w)	32-Bit (w = 0)	32-Bit (w = 1)
000	AX	AL	AX	EAX	AL	EAX
001	CX	CL	CX	ECX	CL	ECX
010	DX	DL	DX	EDX	DL	EDX
011	BX	BL	BX	EBX	BL	EBX
100	SP	AH	SP	ESP	AH	ESP
101	BP	CH	BP	EBP	CH	EBP
110	SI	DH	SI	ESI	DH	ESI
111	DI	BH	DI	EDI	BH	EDI

Figure 7-11. General Register Field Encoding

The physical addresses of all operands addressed by the BP register are computed using the SS segment register. The physical addresses of the destination operands of the string primitive operations (those addressed by the DI register) are computed using the ES segment, which may not be overridden.

Operation Direction (d) Field

The operation direction (d) field is used in many two-operand instructions to indicate which operand is the source and which is the destination.

d	Direction of Operation
0	Register/Memory <-- Register The "reg" field indicates the source operand; "mod r/m" or "mod ss index base" indicates the destination operand.
1	Register <-- Register/Memory The "reg" field indicates the destination operand; "mod r/m" or "mod ss index base" indicates the source operand.

Figure 7-12. Operand Direction Field Encoding

Sign-Extend (s) Field

The sign-extend (s) field primarily appears in instructions having immediate data fields. The s field effects only 8-bit immediate data being placed in a 16-bit or 32-bit destination.

s	8-Bit Immediate Data	16/32-Bit Immediate Data
0	No effect on data	No effect on data
1	Sign-extend 8-Bit data to fill 16-bit or 32-bit destination	No effect on data

Figure 7-13. Sign-Extend Field Encoding

Conditional Test (tttn) Field

For conditional instructions (conditional jumps and set on condition), the conditional test (tttn) field is encoded with n indicating to use the condition (n = 0) or its negation (n = 1), and ttt defining the condition to test.

The following figure shows the conditional test field encoding.

tttn	Condition	Mnemonic
0000	Overflow	O
0001	No Overflow	NO
0010	Below/Not Above or Equal	B/NAE
0011	Not Below/Above or Equal	NB/AE
0100	Equal/Zero	E/Z
0101	Not Equal/Not Zero	NE/NZ
0110	Below or Equal/Not Above	BE/NA
0111	Not Below or Equal/Above	NBE/A
1000	Sign	S
1001	Not Sign	NS
1010	Parity/Parity Even	P/PE
1011	Not Parity/Parity Odd	NP/PO
1100	Less Than/Not Greater or Equal	L/NGE
1101	Not Less Than/Greater or Equal	NL/GE
1110	Less Than or Equal/Not Greater Than	LE/NG
1111	Not Less or Equal/Greater Than	NLE/G

Figure 7-14. Conditional Test Field Encoding

Control, Debug, or Test Register (eee) Field

The following shows the encoding for loading and storing the Control, Debug, and Test registers (eee).

eee Code	Interpreted as Control Register	Interpreted as Debug Register	Interpreted as Test Register
000	CR0	DR0	---
001	---	DR1	---
010	CR2	DR2	---
011	CR3	DR3	---
100	---	---	---
101	---	---	---
110	---	DR6	TR6
111	---	DR7	TR7

Figure 7-15. Control, Debug, and Test Register Field Encoding

Notes:

80386 Microprocessor Instruction Set

General Data Transfer

MOV = Move

Register to Register/Memory

1000100w	mod reg r/m

Register/Memory to Register

1000101w	mod reg r/m

Immediate to Register/Memory

1100011w	mod 000 r/m	8-, 16-, or 32-bit data

Immediate to Register (Short Form)

1011w reg	8-, 16-, or 32-bit data

Memory to Accumulator (Short Form)

1010000w	full 16- or 32-bit displacement

Accumulator to Memory (Short Form)

1010001w	full 16- or 32-bit displacement

Register/Memory to Segment Register

10001110	mod sreg3 r/m

Segment Register to Register/Memory

10001100	mod sreg3 r/m

MOVSX = Move with Sign Extension

Register from Register/Memory

00001111	1011111w	mod reg r/m

MOVZX = Move with Zero Extension

Register from Register/Memory

0 0 0 0 1 1 1 1	1 0 1 1 0 1 1 w	mod reg r/m

PUSH = Push

Register/Memory

1 1 1 1 1 1 1 1	mod 1 1 0 r/m

Register (Short Form)

0 1 0 1 0 reg

Segment Register (ES, CS, SS, or DS) Short Form

0 0 0 sreg2 1 1 0

Segment Register (FS or GS)

0 0 0 0 1 1 1 1	1 0 sreg3 0 0 0

Immediate

0 1 1 0 1 0 s 0	8-, 16-, or 32-bit data

PUSHA = Push All

0 1 1 0 0 0 0 0

POP = Pop

Register/Memory

1 0 0 0 1 1 1 1	mod 0 0 0 r/m

Register (Short Form)

0 1 0 1 1 reg

Segment Register (ES, SS, or DS) Short Form

0 0 0 sreg2 1 1 1

Segment Register (FS or GS)

00001111	10 sreg3 001

POPA = Pop All

01100001

XCHG = Exchange

Register/Memory with Register

1000011w	mod reg r/m

Register with Accumulator (Short Form)

10010 reg

IN = Input From:

Fixed Port

1110010w	port number

Variable Port

1110110w

OUT = Output To:

Fixed Port

1110011w	port number

Variable Port

1110111w

LEA = Load EA to Register

10001101	mod reg r/m

Segment Control

LDS = Load Pointer to DS

11000101	mod reg r/m

LES = Load Pointer to ES

11000100	mod reg r/m

LFS = Load Pointer to FS

00001111	10110100	mod reg r/m

LGS = Load Pointer to GS

00001111	10110101	mod reg r/m

LSS = Load Pointer to SS

00001111	10110010	mod reg r/m

Flag Control

CLC = Clear Carry Flag

11111000

CLD = Clear Direction Flag

11111100

CLI = Clear Interrupt Enable Flag

11111010

CLTS = Clear Task Switched Flag

00001111	00000110

CMC = Complement Carry Flag

11110101

LAHF = Load AH Into Flag

10011111

POPF = Pop Flags

10011101

PUSHF = Push Flags

10011100

SAHF = Store AH into Flags

10011110

STC = Set Carry Flag

11111001

STD = Set Direction Flag

11111101

STI = Set Interrupt Enable Flag

11111011

Arithmetic

ADD = Add

Register to Register

000000d w	mod reg r/m

Register to Memory

0000000 w	mod reg r/m

Memory to Register

0000001 w	mod reg r/m

Immediate to Register/Memory

100000s w	mod 000 r/m	8-, 16-, or 32-bit data

Immediate to Accumulator (Short Form)

0 0 0 0 0 1 0 w	8-, 16-, or 32-bit data

ADC = Add with Carry

Register to Register

0 0 0 1 0 0 d w	mod reg r/m

Register to Memory

0 0 0 1 0 0 0 w	mod reg r/m

Memory to Register

0 0 0 1 0 0 1 w	mod reg r/m

Immediate to Register/Memory

1 0 0 0 0 0 s w	mod 0 1 0 r/m	8-, 16-, or 32-bit data

Immediate to Accumulator (Short Form)

0 0 0 1 0 1 0 w	8-, 16-, or 32-bit data

INC = Increment

Register/Memory

1 1 1 1 1 1 1 w	mod 0 0 0 r/m

Register (Short Form)

0 1 0 0 0 reg

SUB = Subtract

Register from Register

0 0 1 0 1 0 d w	mod reg r/m

Register from Memory

0 0 1 0 1 0 0 w	mod reg r/m

Memory from Register

0 0 1 0 1 0 1 w	mod reg r/m

Immediate from Register/Memory

1 0 0 0 0 0 s w	mod 1 0 1 r/m	8-, 16-, or 32-bit data

Immediate from Accumulator (Short Form)

0 0 1 0 1 1 0 w	8-, 16-, or 32-bit data

SBB = Subtract with Borrow

Register from Register

0 0 0 1 1 0 d w	mod reg r/m

Register from Memory

0 0 0 1 1 0 0 w	mod reg r/m

Memory from Register

0 0 0 1 1 0 1 w	mod reg r/m

Immediate from Register/Memory

1 0 0 0 0 0 s w	mod 0 1 1 r/m	8-, 16-, or 32-bit data

Immediate from Accumulator (Short Form)

0 0 0 1 1 1 0 w	8-, 16-, or 32-bit data

DEC = Decrement

Register/Memory

1 1 1 1 1 1 1 w	mod 0 0 1 r/m

Register (Short Form)

0 1 0 0 1 reg

CMP = Compare

Register with Register

0 0 1 1 1 0 d w	mod reg r/m

Memory with Register

0 0 1 1 1 0 0 w	mod reg r/m

Register with Memory

0 0 1 1 1 0 1 w	mod reg r/m

Immediate with Register/Memory

1 0 0 0 0 0 s w	mod 1 1 1 r/m	8-, 16-, or 32-bit data

Immediate with Accumulator (Short Form)

0 0 1 1 1 1 0 w	8-, 16-, or 32-bit data

NEG = Change Sign

1 1 1 1 0 1 1 w	mod 0 1 1 r/m

AAA = ASCII Adjust for Add

0 0 1 1 0 1 1 1

AAS = ASCII Adjust for Subtract

0 0 1 1 1 1 1 1

DAA = Decimal Adjust for Add

0 0 1 0 0 1 1 1

DAS = Decimal Adjust for Subtract

0 0 1 0 1 1 1 1

MUL = Multiply (Unsigned)

1 1 1 1 0 1 1 w	mod 1 0 0 r/m

IMUL = Integer Multiply (Signed)

Accumulator with Register/Memory

1 1 1 1 0 1 1 w	mod 1 0 1 r/m

Register with Register/Memory

0 0 0 0 1 1 1 1	1 0 1 0 1 1 1 1	mod reg r/m

Register/Memory with Immediate to Register

0 1 1 0 1 0 s 1	mod reg r/m	8-, 16-, or 32-bit data

DIV = Divide (Unsigned)

Accumulator by Register/Memory

1 1 1 1 0 1 1 w	mod 1 1 0 r/m

IDIV = Integer Divide (Signed)

Accumulator by Register/Memory

1 1 1 1 0 1 1 w	mod 1 1 1 r/m

AAD = ASCII Adjust for Divide

1 1 0 1 0 1 0 1	0 0 0 0 1 0 1 0

AAM = ASCII Adjust for Multiply

1 1 0 1 0 1 0 0	0 0 0 0 1 0 1 0

CBW = Convert Byte to Word

1 0 0 1 1 0 0 0

CWD = Convert Word to Double Word

1 0 0 1 1 0 0 1

Logic

Shift/Rotate Instructions
Not through Carry (ROL, ROR, SAL, SAR, SHL, and SHR)

Register/Memory by 1

1 1 0 1 0 0 0 w	mod T T T r/m

Register/Memory by CL

1 1 0 1 0 0 1 w	mod T T T r/m

Register/Memory by Immediate Count

1 1 0 0 0 0 0 w	mod T T T r/m	8-bit data

Shift/Rotate Instructions Through Carry (RCL and RCR)

Register/Memory by 1

1 1 0 1 0 0 0 w	mod T T T r/m

Register/Memory by CL

1 1 0 1 0 0 1 w	mod T T T r/m

Register/Memory by Immediate Count

1 1 0 0 0 0 0 w	mod T T T r/m	8-bit data

T T T	Instruction
0 0 0	ROL
0 0 1	ROR
0 1 0	RCL
0 1 1	RCR
1 0 0	SHL/SAL
1 0 1	SHR
1 1 1	SAR

SHLD = Shift Left Double

Register/Memory by Immediate

0 0 0 0 1 1 1 1	1 0 1 0 0 1 0 0	mod reg r/m	8-bit data

Register/Memory by CL

0 0 0 0 1 1 1 1	1 0 1 0 0 1 0 1	mod reg r/m

SHRD = Shift Right Double

Register/Memory by Immediate

0 0 0 0 1 1 1 1	1 0 1 0 1 1 0 0	mod reg r/m	8-bit data

Register/Memory by CL

0 0 0 0 1 1 1 1	1 0 1 0 1 1 0 1	mod reg r/m

AND = And

Register to Register

0 0 1 0 0 0 d w	mod reg r/m

Register to Memory

0 0 1 0 0 0 0 w	mod reg r/m

Memory to Register

0 0 1 0 0 0 1 w	mod reg r/m

Immediate to Register/Memory

1 0 0 0 0 0 s w	mod 1 0 0 r/m	8-, 16-, or 32-bit data

Immediate to Accumulator (Short Form)

0 0 1 0 0 1 0 w	8-, 16-, or 32-bit data

TEST = AND Function to Flags; No Result

Register/Memory and Register

1 0 0 0 0 1 0 w	mod reg r/m

Immediate Data and Register/Memory

1 1 1 1 0 1 1 w	mod 0 0 0 r/m	8-, 16-, or 32-bit data

Immediate Data and Accumulator (Short Form)

1 0 1 0 1 0 0 w	8-, 16-, or 32-bit data

OR = Or

Register to Register

0 0 0 0 1 0 d w	mod reg r/m

Register to Memory

0 0 0 0 1 0 0 w	mod reg r/m

Memory to Register

0 0 0 0 1 0 1 w	mod reg r/m

Immediate to Register/Memory

1 0 0 0 0 0 s w	mod 0 0 1 r/m	8-, 16-, or 32-bit data

Immediate to Accumulator (Short Form)

0 0 0 0 1 1 0 w	8-, 16-, or 32-bit data

XOR = Exclusive OR

Register to Register

0 0 1 1 0 0 d w	mod reg r/m

Register to Memory

0 0 1 1 0 0 0 w	mod reg r/m

Memory to Register

0 0 1 1 0 0 1 w	mod reg r/m

Immediate to Register/Memory

1 0 0 0 0 0 s w	mod 1 1 0 r/m	8-, 16-, or 32-bit data

Immediate to Accumulator (Short Form)

0 0 1 1 0 1 0 w	8-, 16-, or 32-bit data

NOT = Invert Register/Memory

1 1 1 1 0 1 1 w	mod 0 1 0 r/m

String Manipulation

CMPS = Compare Byte Word

1 0 1 0 0 1 1 w

INS = Input Byte/Word from DX Port

0 1 1 0 1 1 0 w

LODS = Load Byte/Word to AL/AX/EAX

1010110w

MOVS = Move Byte Word

1010010w

OUTS = Output Byte/Word to DX Port

0110111w

SCAS = Scan Byte Word

1010111w

STOS = Store Byte/Word from AL/AX/EX

1010101w

XLAT = Translate String

11010111

Repeated String Manipulation

Repeated by Count in CX or ECX

REPE CMPS = Compare String (Find Non-Match)

11110011	1010011w

REPNE CMPS = Compare String (Find Match)

11110010	1010011w

REP INS = Input String

11110010	0110110w

REP LODS = Load String

11110010	1010110w

REP MOVS = Move String

11110010	1010010w

REP OUTS = Output String

11110010	0110111w

REPE SCAS = Scan String (Find Non-AL/AX/EAX)

11110011	1010111w

REPNE SCAS = Scan String (Find AL/AX/EAX)

11110010	1010111w

REP STOS = Store String

11110010	1010101w

Bit Manipulation

BSF = Scan Bit Forward

00001111	10111100	mod reg r/m

BSR = Scan Bit Reverse

00001111	10111101	mod reg r/m

BT = Test Bit

Register/Memory, Immediate

00001111	10111010	mod 100 r/m	8-bit data

Register/Memory, Register

00001111	10100011	mod reg r/m

BTC = Test Bit and Complement

Register/Memory, Immediate

00001111	10111010	mod 111 r/m	8-bit data

Register/Memory, Register

00001111	10111011	mod reg r/m

BTR = Test Bit and Reset

Register/Memory, Immediate

00001111	10111010	mod 110 r/m	8-bit data

Register/Memory, Register

00001111	10110011	mod reg r/m

BTS = Test Bit and Set

Register/Memory, Immediate

00001111	10111010	mod 101 r/m	8-bit data

Register/Memory, Register

00001111	10101011	mod reg r/m

Control Transfer

CALL = Call

Direct within Segment

11101000	full 16- or 32-bit displacement

Register/Memory Indirect within Segment

11111111	mod 010 r/m

Direct Intersegment

10011010	offset, selector

Indirect Intersegment

11111111	mod 011 r/m

JMP = Unconditional Jump

Short

1 1 1 0 1 0 1 1	8-bit disp.

Direct within Segment

1 1 1 0 1 0 0 1	full 16- or 32-bit displacement

Register/Memory Indirect within Segment

1 1 1 1 1 1 1 1	mod 1 0 0 r/m

Direct Intersegment

1 1 1 0 1 0 1 0	offset, selector

Indirect Intersegment

1 1 1 1 1 1 1 1	mod 1 0 1 r/m

RET = Return from Call

Within Segment

1 1 0 0 0 0 1 1

Within Segment Adding Immediate to SP

1 1 0 0 0 0 1 0	16-bit displacement

Intersegment

1 1 0 0 1 0 1 1

Intersegment Adding Immediate to SP

1 1 0 0 1 0 1 0	16-bit displacement

Conditional Jumps

JO = Jump on Overflow

8-Bit Displacement

01110000	8-bit disp.

Full Displacement

00001111	10000000	full 16- or 32-bit displacement

JNO = Jump on Not Overflow

8-Bit Displacement

01110001	8-bit disp.

Full Displacement

00001111	10000001	full 16- or 32-bit displacement

JB/JNAE = Jump on Below/Not Above or Equal

8-Bit Displacement

01110010	8-bit disp.

Full Displacement

00001111	10000010	full 16- or 32-bit displacement

JNB/JAE = Jump on Not Below/Above or Equal

8-Bit Displacement

01110011	8-bit disp.

Full Displacement

00001111	10000011	full 16- or 32-bit displacement

JE/JZ = Jump on Equal/Zero

8-Bit Displacement

01110100	8-bit disp.

Full Displacement

00001111	10000100	full 16- or 32-bit displacement

JNE/JNZ = Jump on Not Equal/Not Zero

8-Bit Displacement

01110101	8-bit disp.

Full Displacement

00001111	10000101	full 16- or 32-bit displacement

JBE/JNA = Jump on Below or Equal/Not Above

8-Bit Displacement

01110110	8-bit disp.

Full Displacement

00001111	10000110	full 16- or 32-bit displacement

JNBE/JA = Jump on Not Below or Equal/Above

8-Bit Displacement

01110111	8-bit disp.

Full Displacement

00001111	10000111	full 16- or 32-bit displacement

JS = Jump on Sign

8-Bit Displacement

01111000	8-bit disp.

Full Displacement

00001111	10001000	full 16- or 32-bit displacement

JNS = Jump on Not Sign

8-Bit Displacement

01111001	8-bit disp.

Full Displacement

00001111	10001001	full 16- or 32-bit displacement

JP/JPE = Jump on Parity/Parity Even

8-Bit Displacement

01111010	8-bit disp.

Full Displacement

00001111	10001010	full 16- or 32-bit displacement

JNP/JPO = Jump on Not Parity/Parity Odd

8-Bit Displacement

01111011	8-bit disp.

Full Displacement

00001111	10001011	full 16- or 32-bit displacement

JL/JNGE = Jump on Less/Not Greater or Equal

8-Bit Displacement

01111100	8-bit disp.

Full Displacement

00001111	10001100	full 16- or 32-bit displacement

JNL/JGE = Jump on Not Less/Greater or Equal

8-Bit Displacement

01111101	8-bit disp.

Full Displacement

00001111	10001101	full 16- or 32-bit displacement

JLE/JNG = Jump on Less or Equal/Not Greater

8-Bit Displacement

01111110	8-bit disp.

Full Displacement

00001111	10001110	full 16- or 32-bit displacement

JNLE/JG = Jump on Not Less or Equal/Greater

8-Bit Displacement

01111111	8-bit disp.

Full Displacement

00001111	10001111	full 16- or 32-bit displacement

JCXZ = Jump on CX Zero

11100011	8-bit disp.

JECXZ = Jump on ECX Zero

11100011	8-bit disp.

Note: The operand size prefix differentiates JCXZ from JECXZ.

LOOP = Loop CX Times

11100010	8-bit disp.

LOOPZ/LOOPE = Loop with Zero/Equal

11100001	8-bit disp.

LOOPNZ/LOOPNE = Loop while Not Zero

11100000	8-bit disp.

Conditional Byte Set

SETO = Set Byte on Overflow

To Register/Memory

00001111	10010000	mod 000 r/m

SETNO = Set Byte on Not Overflow

To Register/Memory

00001111	10010001	mod 000 r/m

SETB/SETNAE = Set Byte on Below/Not Above or Equal

To Register/Memory

00001111	10010010	mod 000 r/m

SETNB = Set Byte on Not Below/Above or Equal

To Register/Memory

00001111	10010011	mod 000 r/m

SETE/SETZ = Set Byte on Equal/Zero

To Register/Memory

00001111	10010100	mod 000 r/m

SETNE/SETNZ = Set Byte on Not Equal/Not Zero

To Register/Memory

00001111	10010101	mod 000 r/m

SETBE/SETNA = Set Byte on Below or Equal/Not Above

To Register/Memory

00001111	10010110	mod 000 r/m

SETNBE/SETA = Set Byte on Not Below or Equal/Above

To Register/Memory

00001111	10010111	mod 000 r/m

SETS = Set Byte on Sign

To Register/Memory

00001111	10011000	mod 000 r/m

SETNS = Set Byte on Not Sign

To Register/Memory

00001111	10011001	mod 000 r/m

SETP/SETPE = Set Byte on Parity/Parity Even

To Register/Memory

00001111	10011010	mod 000 r/m

SETNP/SETPO = Set Byte on Not Parity/Parity Odd

To Register/Memory

00001111	10011011	mod 000 r/m

SETL/SETNGE = Set Byte on Less/Not Greater or Equal

To Register/Memory

00001111	10011100	mod 000 r/m

SETNL/SETGE = Set Byte on Not Less/Greater or Equal

To Register/Memory

00001111	01111101	mod 000 r/m

SETLE/SETNG = Set Byte on Less or Equal/Not Greater

To Register/Memory

00001111	10011110	mod 000 r/m

SETNLE/SETG = Set Byte on Not Less or Equal/Greater

To Register/Memory

00001111	10011111	mod 000 r/m

ENTER = Enter Procedure

11001000	16-bit displacement	8-bit level

LEAVE = Leave Procedure

11001001

Interrupt Instructions

INT = Interrupt

Type Specified

11001101	type

Type 3

11001100

INTO = Interrupt 4 if Overflow Flag Set

11001110

BOUND = Interrupt 5 if Detect Value Out of Range

01100010	mod reg r/m

IRET = Interrupt Return

11001111

Processor Control

HLT = Halt

11110100

MOV = Move to and from Control/Debug/Test Registers

CR0/CR2/CR3 from Register

00001111	00100010	11 eee reg

Register from CR0-3

00001111	00100000	11 eee reg

DR0-3, DR6-7 from Register

00001111	00100011	11 eee reg

Register from DR0-3, DR6-7

00001111	00100001	11 eee reg

TR6-7 from Register

00001111	00100110	11 eee reg

Register from TR6-7

00001111	00100100	11 eee reg

NOP = No Operation

10010000

WAIT = Wait until BUSY Pin is Negated

10011011

Processor Extension Instructions

ESC = Processor Extension Escape

11011TTT	mod L L L r/m

Note: TTT and LLL bits are opcode information for the coprocessor.

Prefix Bytes

Address Size Prefix

01100111

Operand Size Prefix

01100110

LOCK = Bus Lock Prefix

11110000

Note: The use of LOCK is restricted to an exchange with memory, or bit test and reset type of instruction.

Segment Override Prefix

CS:

00101110

DS:

00111110

ES:

00100110

FS:

01100100

GS:

01100101

SS:

00110110

Protection Control

ARPL = Adjust Requested Privilege Level from Register/Memory

01100011	mod reg r/m

LAR = Load Access Rights from Register/Memory

00001111	00000010	mod reg r/m

LGDT = Load Global Descriptor Table Register

00001111	00000001	mod 010 r/m

LIDT = Load Interrupt Descriptor Table Register

00001111	00000001	mod 011 r/m

LLDT = Load Local Descriptor Table Register to Register/Memory

00001111	00000000	mod 010 r/m

LMSW = Load Machine Status Word from Register/Memory

00001111	00000001	mod 110 r/m

LSL = Load Segment Limit from Register/Memory

00001111	00000011	mod reg r/m

LTR = Load Task Register from Register/Memory

00001111	00000000	mod 001 r/m

SGDT = Store Global Descriptor Table Register

00001111	00000001	mod 000 r/m

SIDT = Store Interrupt Descriptor Table Register

00001111	00000001	mod 001 r/m

SLDT = Store Local Descriptor Table Register to Register/Memory

00001111	00000000	mod 000 r/m

SMSW = Store Machine Status Word

00001111	00000001	mod 100 r/m

STR = Store Task Register to Register/Memory

00001111	00000000	mod 001 r/m

VERR = Verify Read Access; Register/Memory

00001111	00000000	mod 100 r/m

VERW = Verify Write Access

00001111	00000000	mod 101 r/m

Notes:

Introduction to the 80387 Instruction Set

The 80387 instructions use many of the same fields defined earlier in this section for the 80386 instructions. Additional fields used by the 80387 instructions are defined in the following figure.

Field	Description	Bit Information
escape	80386 Extension Escape	Bit Pattern = 11011
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 the 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*	0 = Destination (op) source 1 = Source (op) destination

* When d = 1, reverse the sense of R.

Figure 7-16. 80387 Encoding Field Summary

Within the 80387 Instruction Set:

Temporary (Extended) Real is 80-bit Real.
Long Integer is a 64-bit integer.

80387 Usage of the Scale-Index-Base Byte

The "mod r/m" byte of an 80387 instruction can be followed by a scale-index-base (s-i-b) byte having the same address mode definition as in the 80386 instruction. The mod field in the 80387 instruction is never equal to 11.

Instruction and Data Pointers

The parallel operation of the 80386 and 80387 may allow errors detected by the 80387 to be reported after the 80386 has executed the ESC instruction that caused the error. The 80386/80387 provides two pointer registers to identify the failing numeric instruction. The pointer registers supply the address of the failing numeric instruction and the address of its numeric memory operand when applicable.

Although the pointer registers are located in the 80386, they appear to be located in the 80387 because they are accessed by the ESC instructions FLDENV, FSTENV, FSAVE, and FRSTOR. Whenever the 80386 decodes a new ESC instruction, it saves the address of the instruction along with any prefix bytes that may be present, the address of the operand (if present), and the opcode.

The instruction and data pointers appear in one of four available formats:

- 16-bit Real Mode/Virtual 8086 Mode
- 32-bit Real Mode
- 16-bit Protected Mode
- 32-bit Protected Mode

The Real Mode formats are used whenever the 80386 is in the Real Mode or Virtual 8086 Mode. The Protected Mode formats are used when the 80386 is in the Protected Mode. The Operand Size Prefix can also be used with the 80387 instructions. The operand size of the 80387 instruction determines whether the 16-bit or 32-bit format is used.

Note: FSAVE and FRSTOR have an additional eight fields (10 bytes per field) that contain the current contents of ST(0) through ST(7). These fields follow the instruction and data pointer image shown in the following figures.

The following figures show the instruction and data pointer image format used in the various address modes. The ESC instructions FLDENV, FSTENV, FSAVE, and FRSTOR are used to transfer these values between the 80386/80387 registers and memory.

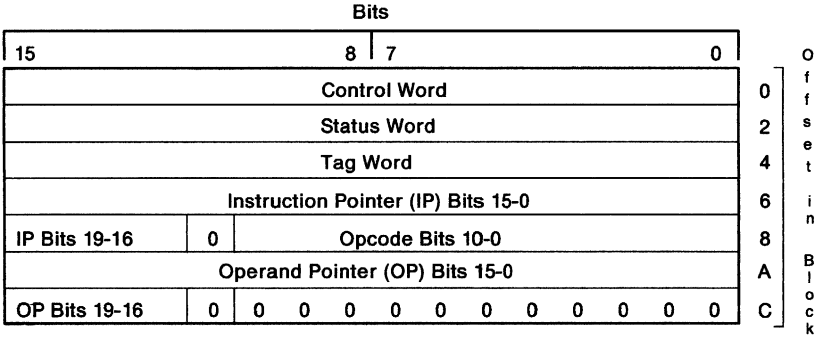


Figure 7-17. Instruction and Pointer Image (16-Bit Real Mode)

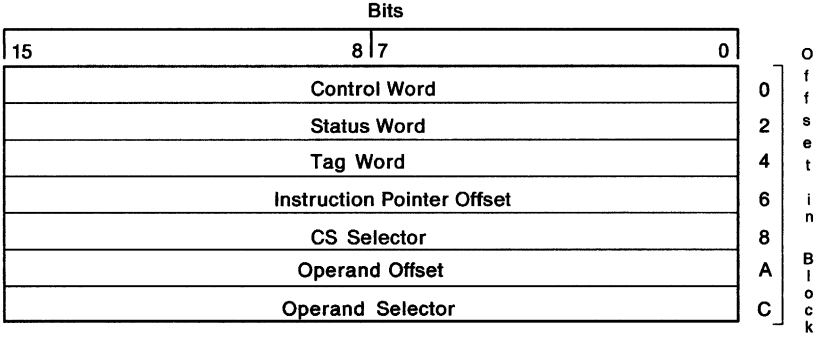


Figure 7-18. Instruction and Pointer Image (16-Bit Protected Mode)

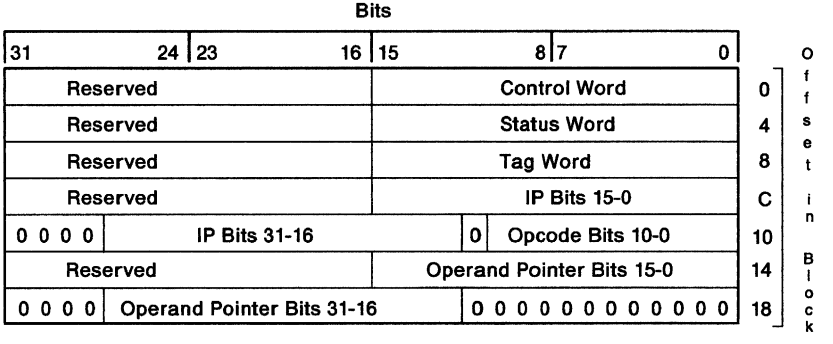


Figure 7-19. Instruction and Pointer Image (32-Bit Real Mode)

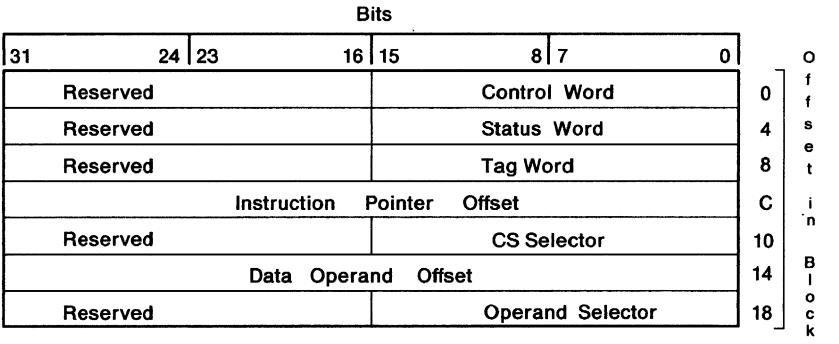


Figure 7-20. Instruction and Pointer Image (32-Bit Protected Mode)

New Instructions

Several new instructions are included in the 80387 instruction set that are not available to the 80287 or 8087 math coprocessors. The new instructions are:

FUCOM (Unordered Compare Real)
FUCOMP (Unordered Compare Real and Pop)
FUCOMPP (Unordered Compare Real and Pop Twice)
FPREM1 (IEEE Partial Remainder)
FSINE (Sine)
FCOS (Cosine)
FSINCOS (Sine and Cosine).

80387 Coprocessor Instruction Set

The following is an instruction set summary for the 80387 coprocessor. In the following, the bit pattern for escape is 11011.

Data Transfer

FLD = Load

Integer/Real Memory to ST(0)

escape MF 1	mod 0 0 0 r/m

Long Integer Memory to ST(0)

escape 1 1 1	mod 1 0 1 r/m

Temporary Real Memory to ST(0)

escape 0 1 1	mod 1 0 1 r/m

BCD Memory to ST(0)

escape 1 1 1	mod 1 0 0 r/m

ST(i) to ST(0)

escape 0 0 1	1 1 0 0 0 ST(i)

FST = Store

ST(0) to Integer/Real Memory

escape MF 1	mod 0 1 0 r/m

ST(0) to ST(i)

escape 1 0 1	1 1 0 1 0 ST(i)

FSTP = Store and Pop

ST(0) to Integer/Real Memory

escape MF 1	mod 0 1 1 r/m

ST(0) to Long Integer Memory

escape 1 1 1	mod 1 1 1 r/m

ST(0) to Temporary Real Memory

escape 0 1 1	mod 1 1 1 r/m

ST(0) to BCD Memory

escape 1 1 1	mod 1 1 0 r/m

ST(0) to ST(i)

escape 1 0 1	1 1 0 1 1 ST(i)

FXCH = Exchange ST(i) and ST(0)

escape 0 0 1	1 1 0 0 1 ST(i)

Comparison

FCOM = Compare

Integer/Real Memory to ST(0)

escape MF 0	mod 0 1 0 r/m

ST(i) to ST(0)

escape 0 0 0	1 1 0 1 0 ST(i)

FCOMP = Compare and Pop

Integer/Real Memory to ST(0)

escape MF 0	mod 0 1 1 r/m

ST(i) to ST(0)

escape 0 0 0	1 1 0 1 1 ST(i)

FCOMPP = Compare ST(1) to ST(0) and Pop Twice

escape 1 1 0	1 1 0 1 1 0 0 1

FUCOM = Unordered Compare Real

escape 1 0 1	1 1 1 0 0 ST(i)

FUCOMP = Unordered Compare Real and Pop

escape 1 0 1	1 1 1 0 1 ST(i)

FUCOMPP = Unordered Compare Real and Pop Twice

escape 0 1 0	1 1 1 0 1 0 0 1

FTST = Test ST(0)

escape 0 0 1	1 1 1 0 0 1 0 0

FXAM = Examine ST(0)

escape 0 0 1	1 1 1 0 0 1 0 1

Constants

FLDZ = Load +0.0 into ST(0)

escape 0 0 1	1 1 1 0 1 1 1 0

FLD1 = Load +1.0 into ST(0)

escape 0 0 1	1 1 1 0 1 0 0 0

FLDPI = Load π into ST(0)

escape 0 0 1	1 1 1 0 1 0 1 1

FLDL2T = Load $\log_2 10$ into ST(0)

escape 0 0 1	1 1 1 0 1 0 0 1

FLDL2E = Load $\log_2 e$ into ST(0)

escape 0 0 1	1 1 1 0 1 0 1 0

FLDLG2 = Load $\log_{10} 2$ into ST(0)

escape 0 0 1	1 1 1 0 1 1 0 0

FLDLN2 = Load $\log_e 2$ into ST(0)

escape 0 0 1	1 1 1 0 1 1 0 1

Arithmetic

FADD = Addition

Integer/Real Memory with ST(0)

escape MF 0	mod 0 0 0 r/m

ST(i) and ST(0)

escape d P 0	1 1 0 0 0 ST(i)

FSUB = Subtraction

Integer/Real Memory with ST(0)

escape MF 0	mod 1 0 R r/m

ST(i) and ST(0)

escape d P 0	1 1 1 0 R r/m

FMUL = Multiplication

Integer/Real Memory with ST(0)

escape MF 0	mod 0 0 1 r/m

ST(i) and ST(0)

escape d P 0	1 1 0 0 1 r/m

FDIV = Division

Integer/Real Memory with ST(0)

escape MF 0	mod 1 1 R r/m

ST(i) and ST(0)

escape d P 0	1 1 1 1 R r/m

FSQRT = Square Root of ST(0)

escape 0 0 1	1 1 1 1 1 0 1 0

FSCALE = Scale ST(0) by ST(1)

escape 0 0 1	1 1 1 1 1 1 0 1

FPREM = Partial Remainder of ST(0) ÷ ST(1)

escape 0 0 1	1 1 1 1 1 0 0 0

FPREM1 = IEEE Partial Remainder

escape 0 0 1	1 1 1 1 0 1 0 1

FRNDINT = Round ST(0) to Integer

escape 0 0 1	1 1 1 1 1 1 0 0

FXTRACT = Extract Components of ST(0)

escape 0 0 1	1 1 1 1 0 1 0 0

FABS = Absolute Value of ST(0)

escape 0 0 1	1 1 1 0 0 0 0 1

FCHS = Change Sign of ST(0)

escape 0 0 1	1 1 1 0 0 0 0 0

Transcendental

FPTAN = Partial Tangent of ST(0)

escape 0 0 1	1 1 1 1 0 0 1 0

FPATAN = Partial Arctangent of ST(1) ÷ ST(0)

escape 0 0 1	1 1 1 1 0 0 1 1

FSIN = Sine

escape 0 0 1	1 1 1 1 1 1 1 0

FCOS = Cosine

escape 0 0 1	1 1 1 1 1 1 1 1

FSINCOS = Sine and Cosine

escape 0 0 1	1 1 1 1 1 0 1 1

F2XM1 = $2^{ST(0)}$ -1

escape 0 0 1	1 1 1 1 0 0 0 0

FYL2X = ST(1) x Log_2 [ST(0)]

escape 0 0 1	1 1 1 1 0 0 0 1

FYL2XP1 = ST(1) x Log_2 [ST(0) + 1]

escape 0 0 1	1 1 1 1 1 0 0 1

Processor Control

FINIT = Initialize NPX

escape 0 1 1	1 1 1 0 0 0 1 1

FSTSW AX = Store Control Word

escape 1 1 1	1 1 1 0 0 0 0 0

FLDCW = Load Control Word

escape 0 0 1	mod 1 0 1 r/m

FSTCW = Store Control Word

escape 0 0 1	mod 1 1 1 r/m

FSTSW = Store Status Word

escape 1 0 1	mod 1 1 1 r/m

FCLEX = Clear Exceptions

escape 0 1 1	1 1 1 0 0 0 1 0

FSTENV = Store Environment

escape 0 0 1	mod 1 1 0 r/m

FLDENV = Load Environment

escape 0 0 1	mod 1 0 0 r/m

FSAVE = Save State

escape 1 0 1	mod 1 1 0 r/m

FRSTOR = Restore State

escape 1 0 1	mod 1 0 0 r/m

FINCSTP = Increment Stack Pointer

escape 0 0 1	1 1 1 1 0 1 1 1

FDECSTP = Decrement Stack Pointer

escape 0 0 1	1 1 1 1 0 1 1 0

FFREE = Free ST(i)

escape 1 0 1	1 1 0 0 0 ST(i)

FNOP = No Operation

escape 0 0 1	1 1 0 1 0 0 0 0

Section 8. Characters and Keystrokes

Notes:

Character Codes

This section provides charts showing the decimal values, hexadecimal values, and keystrokes for each character. The notes referred to in the charts are located on page 8-9.

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
00	0	Blank (Null)	Ctrl 2	
01	1	☺	Ctrl A	
02	2	☻	Ctrl B	
03	3	♥	Ctrl C	
04	4	♦	Ctrl D	
05	5	♣	Ctrl E	
06	6	♠	Ctrl F	
07	7	•	Ctrl G	
08	8	◘	Ctrl H, Backspace, Shift Backspace	
09	9	○	Ctrl I	
0A	10	◙	Ctrl J, Ctrl ↵	
0B	11	♂	Ctrl K	
0C	12	♀	Ctrl L	
0D	13	♪	Ctrl M, ↵ Shift ↵	
0E	14	♫	Ctrl N	
0F	15	☼	Ctrl O	
10	16	►	Ctrl P	
11	17	◄	Ctrl Q	
12	18	↕	Ctrl R	
13	19	‼	Ctrl S	
14	20	¶	Ctrl T	
15	21	§	Ctrl U	
16	22	▬	Ctrl V	
17	23	↨	Ctrl W	

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
18	24	↑	Ctrl X	
19	25	↓	Ctrl Y	
1A	26	→	Ctrl Z	
1B	27	←	Ctrl [Esc, Shift Esc, Crtl Esc	
1C	28	∟	Ctrl \	
1D	29	↔	Ctrl]	
1E	30	▲	Ctrl 6	
1F	31	▼	Ctrl -	
20	32	Blank Space	Space Bar, Shift, Space, Ctrl Space, Alt Space	
21	33	!	!	Shift
22	34	"	"	Shift
23	35	#	#	Shift
24	36	$	$	Shift
25	37	%	%	Shift
26	38	&	&	Shift
27	39	'	'	Shift
28	40	(	(	Shift
29	41	)	)	
2A	42	*	*	Note 1
2B	43	+	+	Shift
2C	44	,	,	
2D	45	-	-	
2E	46	.	.	Note 2

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
2F	47	/	/	
30	48	0	0	Note 3
31	49	1	1	Note 3
32	50	2	2	Note 3
33	51	3	3	Note 3
34	52	4	4	Note 3
35	53	5	5	Note 3
36	54	6	6	Note 3
37	55	7	7	Note 3
38	56	8	8	Note 3
39	57	9	9	Note 3
3A	58	:	:	Shift
3B	59	;	;	
3C	60	<	<	Shift
3D	61	=	=	
3E	62	>	>	Shift
3F	63	?	?	Shift
40	64	@	@	Shift
41	65	A	A	Note 4
42	66	B	B	Note 4
43	67	C	C	Note 4
44	68	D	D	Note 4
45	69	E	E	Note 4
46	70	F	F	Note 4
47	71	G	G	Note 4
48	72	H	H	Note 4
49	73	I	I	Note 4
4A	74	J	J	Note 4

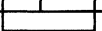
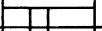
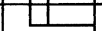
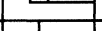
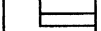
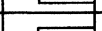
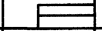
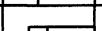
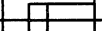
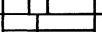
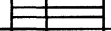
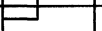
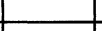
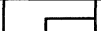
Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
4B	75	K	K	Note 4
4C	76	L	L	Note 4
4D	77	M	M	Note 4
4E	78	N	N	
4F	79	O	O	Note 4
50	80	P	P	Note 4
51	81	Q	Q	Note 4
52	82	R	R	Note 4
53	83	S	S	Note 4
54	84	T	T	Note 4
55	85	U	U	Note 4
56	86	V	V	Note 4
57	87	W	W	Note 4
58	88	X	X	Note 4
59	89	Y	Y	Note 4
5A	90	Z	Z	Note 4
5B	91	[	[	
5C	92	\	\	Note 4
5D	93	]	]	
5E	94	^	^	Shift
5F	95	_	_	Shift
60	96	`	`	
61	97	a	a	Note 5
62	98	b	b	Note 5
63	99	c	c	Note 5
64	100	d	d	Note 5
65	101	e	e	Note 5
66	102	f	f	Note 5

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
67	103	g	g	Note 5
68	104	h	h	Note 5
69	105	i	i	Note 5
6A	106	j	j	Note 5
6B	107	k	k	Note 5
6C	108	l	l	Note 5
6D	109	m	m	Note 5
6E	110	n	n	Note 5
6F	111	o	o	Note 5
70	112	p	p	Note 5
71	113	q	q	Note 5
72	114	r	r	Note 5
73	115	s	s	Note 5
74	116	t	t	Note 5
75	117	u	u	Note 5
76	118	v	v	Note 5
77	119	w	w	Note 5
78	120	x	x	Note 5
79	121	y	y	Note 5
7A	122	z	z	Note 5
7B	123	{	{	Shift
7C	124	¦	¦	Shift
7D	125	}	}	Shift
7E	126	~	~	Shift
7F	127	△	Ctrl-	

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
80	128	Ç	Alt 128	Note 6
81	129	ü	Alt 129	Note 6
82	130	é	Alt 130	Note 6
83	131	â	Alt 131	Note 6
84	132	ä	Alt 132	Note 6
85	133	à	Alt 133	Note 6
86	134	å	Alt 134	Note 6
87	135	ç	Alt 135	Note 6
88	136	ê	Alt 136	Note 6
89	137	ë	Alt 137	Note 6
8A	138	è	Alt 138	Note 6
8B	139	ï	Alt 139	Note 6
8C	140	î	Alt 140	Note 6
8D	141	ì	Alt 141	Note 6
8E	142	Ä	Alt 142	Note 6
8F	143	Å	Alt 143	Note 6
90	144	É	Alt 144	Note 6
91	145	æ	Alt 145	Note 6
92	146	Æ	Alt 146	Note 6
93	147	ô	Alt 147	Note 6
94	148	ö	Alt 148	Note 6
95	149	ò	Alt 149	Note 6
96	150	û	Alt 150	Note 6
97	151	ù	Alt 151	Note 6
98	152	ÿ	Alt 152	Note 6
99	153	Ö	Alt 153	Note 6
9A	154	Ü	Alt 154	Note 6

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
9B	155	¢	Alt 155	Note 6
9C	156	£	Alt 156	Note 6
9D	157	¥	Alt 157	Note 6
9E	158	Pt	Alt 158	Note 6
9F	159	ƒ	Alt 159	Note 6
A0	160	á	Alt 160	Note 6
A1	161	í	Alt 161	Note 6
A2	162	ó	Alt 162	Note 6
A3	163	ú	Alt 163	Note 6
A4	164	ñ	Alt 164	Note 6
A5	165	Ñ	Alt 165	Note 6
A6	166	a	Alt 166	Note 6
A7	167	o	Alt 167	Note 6
A8	168	¿	Alt 168	Note 6
A9	169	⌐	Alt 169	Note 6
AA	170	¬	Alt 170	Note 6
AB	171	½	Alt 171	Note 6
AC	172	¼	Alt 172	Note 6
AD	173	¡	Alt 173	Note 6
AE	174	<<	Alt 174	Note 6
AF	175	>>	Alt 175	Note 6
B0	176	⁙	Alt 176	Note 6
B1	177	▒	Alt 177	Note 6
B2	178	▓	Alt 178	Note 6
B3	179		Alt 179	Note 6
B4	180		Alt 180	Note 6
B5	181		Alt 181	Note 6
B6	182		Alt 182	Note 6

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
B7	183		Alt 183	Note 6
B8	184		Alt 184	Note 6
B9	185		Alt 185	Note 6
BA	186		Alt 186	Note 6
BB	187		Alt 187	Note 6
BC	188		Alt 188	Note 6
BD	189		Alt 189	Note 6
BE	190		Alt 190	Note 6
BF	191		Alt 191	Note 6
C0	192		Alt 192	Note 6
C1	193		Alt 193	Note 6
C2	194		Alt 194	Note 6
C3	195		Alt 195	Note 6
C4	196		Alt 196	Note 6
C5	197		Alt 197	Note 6
C6	198		Alt 198	Note 6
C7	199		Alt 199	Note 6
C8	200		Alt 200	Note 6
C9	201		Alt 201	Note 6
CA	202		Alt 202	Note 6
CB	203		Alt 203	Note 6
CC	204		Alt 204	Note 6
CD	205		Alt 205	Note 6
CE	206		Alt 206	Note 6
CF	207		Alt 207	Note 6
D0	208		Alt 208	Note 6

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
D1	209		Alt 209	Note 6
D2	210		Alt 210	Note 6
D3	211		Alt 211	Note 6
D4	212		Alt 212	Note 6
D5	213		Alt 213	Note 6
D6	214		Alt 214	Note 6
D7	215		Alt 215	Note 6
D8	216		Alt 216	Note 6
D9	217		Alt 217	Note 6
DA	218		Alt 218	Note 6
DB	219		Alt 219	Note 6
DC	220		Alt 220	Note 6
DD	221		Alt 221	Note 6
DE	222		Alt 222	Note 6
DF	223		Alt 223	Note 6
EO	224	α	Alt 224	Note 6
E1	225	β	Alt 225	Note 6
E2	226	Γ	Alt 226	Note 6
E3	227	π	Alt 227	Note 6
E4	228	Σ	Alt 228	Note 6
E5	229	σ	Alt 229	Note 6
E6	230	μ	Alt 230	Note 6
E7	231	τ	Alt 231	Note 6
E8	232	Φ	Alt 232	Note 6
E9	233	θ	Alt 233	Note 6
EA	234	Ω	Alt 234	Note 6
EB	235	δ	Alt 235	Note 6

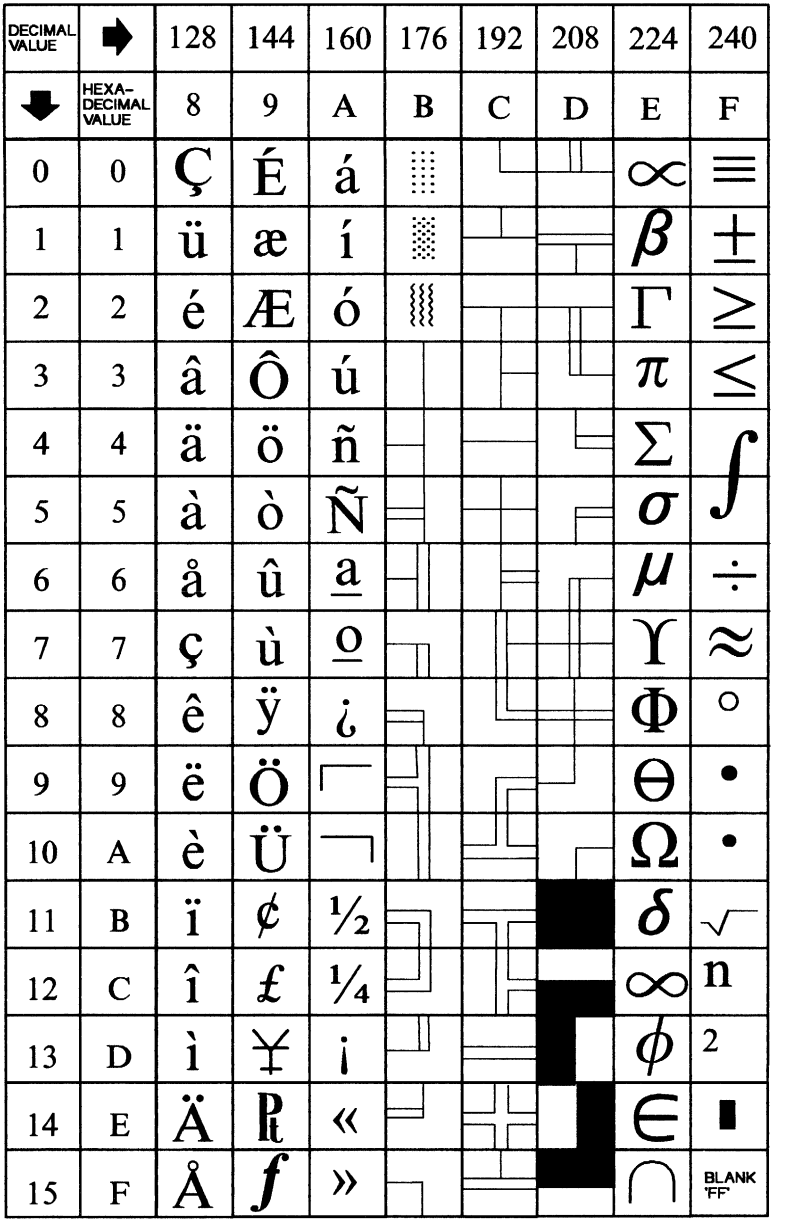
Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
EC	236	∞	Alt 236	Note 6
ED	237	φ	Alt 237	Note 6
EE	238	∈	Alt 238	Note 6
EF	239	∩	Alt 239	Note 6
F0	240	≡	Alt 240	Note 6
F1	241	±	Alt 241	Note 6
F2	242	≥	Alt 242	Note 6
F3	243	≤	Alt 243	Note 6
F4	244	⌠	Alt 244	Note 6
F5	245	⌡	Alt 245	Note 6
F6	246	÷	Alt 246	Note 6
F7	247	≈	Alt 247	Note 6
F8	248	°	Alt 248	Note 6
F9	249	∙	Alt 249	Note 6
FA	250	·	Alt 250	Note 6
FB	251	√	Alt 251	Note 6
FC	252	n	Alt 252	Note 6
FD	253	2	Alt 253	Note 6
FE	254	■	Alt 254	Note 6
FF	255	BLANK	Alt 255	Note 6

Notes:

1. Asterisk (*) can be typed by pressing the * key or, in the shift state, pressing the 8 key.
2. Period (.) can be typed by pressing the . key or, in the shift or Num Lock state, pressing the Del key.
3. Numeric characters 0-9 can be typed by pressing the numeric keys on the top row of the keyboard or, in the shift or Num Lock state, pressing the numeric keys in the keypad portion of the keyboard.
4. Uppercase alphabetic characters (A-Z) can be typed by pressing the character key in the shift state or the Caps Lock state.
5. Lowercase alphabetic characters (a-z) can be typed by pressing the character key in the normal state or in Caps Lock and shift state combined.
6. The three digits are typed on the numeric keypad while holding the Alt key depressed. Character codes 001-255 may be entered in this fashion (with Caps Lock activated, character codes 97-122 display uppercase).

Quick Reference

DECIMAL VALUE	➡	0	16	32	48	64	80	96	112
⬇	HEXA-DECIMAL VALUE	0	1	2	3	4	5	6	7
0	0	BLANK (NULL)	►	BLANK (SPACE)	0	@	P	‘	p
1	1	☺	◄	!	1	A	Q	a	q
2	2	☻	↕	"	2	B	R	b	r
3	3	♥	!!	#	3	C	S	c	s
4	4	♦	¶	$	4	D	T	d	t
5	5	♣	§	%	5	E	U	e	u
6	6	♠	▬	&	6	F	V	f	v
7	7	•	↨	′	7	G	W	g	w
8	8	◘	↑	(	8	H	X	h	x
9	9	○	↓	)	9	I	Y	i	y
10	A	◙	→	*	:	J	Z	j	z
11	B	♂	←	+	;	K	[	k	{
12	C	♀	∟	,	<	L	\	l	¦
13	D	♪	↔	—	=	M	]	m	}
14	E	♫	▲	.	>	N	^	n	~
15	F	☼	▼	/	?	O	_	o	△


DECIMAL VALUE
HEXA-DECIMAL VALUE
128 144 160 176 192 208 224 240
8 9 A B C D E F
0 0 Ç É á ∝ ≡
1 1 ü æ í β ±
2 2 é Æ ó Γ ≥
3 3 â Ô ú π ≤
4 4 ä ö ñ Σ ∫
5 5 à ò Ñ σ
6 6 å û a μ ÷
7 7 ç ù o Υ ≈
8 8 ê ÿ ¿ Φ °
9 9 ë Ö Θ •
10 A è Ü Ω •
11 B ï ¢ ½ δ √
12 C î £ ¼ ∞ n
13 D ì ¡ φ 2
14 E Ä « ∈
15 F Å » ∩
BLANK 'FF'

Notes:

Section 9. Compatibility

Notes:

Introduction

This section discusses the major differences between the systems in the IBM Personal Computer and Personal System/2 product lines. Also included are programming considerations that must be taken into account when designing application programs for the IBM Personal Computer and Personal System/2 products.

System Board

The Model 80 system board uses an 80386 microprocessor and has provisions to install an optional 80387 math coprocessor. The Model 80 is generally compatible with applications written for 8087 and 80287 Math Coprocessors. However, if a math coprocessor instruction other than FINIT, FSTSW, or FSTCW is executed without the 80387 present, the 80386 microprocessor waits indefinitely for a response from the 80387. This causes the system to stop processing without providing an error indication. To prevent this problem, software should check for the presence of the 80387 before executing math coprocessor instructions. Math coprocessor compatibility is discussed in "Math Coprocessor Compatibility" on page 3-10.

The 80386 system microprocessor and general architecture of the Model 80 have created some fundamental differences between it and other systems. These differences must be taken into consideration when designing programs exclusively for the Model 80 or programs compatible across the IBM Personal Computer and Personal System/2 product lines. Programming considerations are discussed in "Application Guidelines" on page 9-6.

Diskette Drives and Controller

The following figure shows the read, write, and format capabilities for each type of diskette drive used by the Model 80.

Diskette Drive Type	160/180K Mode	320/360K Mode	1.44M Mode	720K Mode
5.25-Inch Diskette Drive:				
Single Sided (48 TPI)	R W F	---	---	---
Double Sided (48 TPI)	R W F	R W F	---	---
3.5-Inch Diskette Drive:				
720KB Drive	---	---	---	R W F
1.44MB Drive	---	---	R W F	R W F
R-Read W-Write F-Format				

Figure 9-1. Diskette Drive Read, Write, and Format Capabilities

Notes:

1. 5.25-inch diskettes designed for the 1.2M mode cannot be used in either a 160/180K or a 320/360K diskette drive.
2. 3.5-inch diskettes designed for the 1.44M mode cannot be used in a 720K diskette drive.

Warning: 32-bit operations to the video subsystem can cause a diskette overrun in the 1.44M mode because data width conversions may require more than 12 microseconds. If an overrun occurs, BIOS returns an error code and the operation should be retried.

Copy Protection

The following methods of copy protection may not work on systems using the 5.25-inch high capacity diskette drive or the 3.5-inch 1.44M diskette drive.

- Bypassing BIOS Routines
  - Track Density: The 5.25-inch high capacity diskette drive records tracks at a density of 96 tracks per inch (TPI). This drive has to double-step in the 48 TPI mode, which is performed by BIOS.

- Data Transfer Rate: BIOS selects the proper data transfer rate for the media being used.
- Diskette Parameters Table: Copy protection, which creates its own Diskette Parameters Table may not work on these drives.

- Diskette Drive Controls
  - Rotational Speed: The time between two events on a diskette is a function of the controller.
  - Access Time: Diskette BIOS routines must set the track-to-track access time for the different types of media used in the drives.
  - Diskette Change Signal: Copy protection may not be able to reset this signal.
- Write Current Control—Copy protection that uses write current control will not work because the controller selects the proper write current for the media being used.

Fixed Disk Drives and Controller

Reading from and writing to the fixed disk drive is initiated in the same way as with IBM Personal Computer products; however, new functions are supported. Detailed information about specific fixed disk drives and fixed disk adapters is available in separate technical references.

Application Guidelines

Use the following information to develop application programs for the IBM Personal Computer and Personal System/2 products. Whenever possible, BIOS should be used as an interface to hardware in order to provide maximum compatibility and portability of applications across systems.

Hardware Interrupts

Hardware interrupts are level-sensitive for systems using the Micro Channel architecture while systems using the Personal Computer type I/O channel design have edge-sensitive hardware interrupts. On edge-sensitive interrupt systems, the interrupt controller clears its internal interrupt-in-progress latch when the interrupt routine sends an End-of-Interrupt (EOI) command to the controller. The EOI is sent whether the incoming interrupt request to the controller is active or inactive.

In level-sensitive systems, the interrupt-in-progress latch is readable at an I/O address bit position. This latch is read during the interrupt service routine and may be reset by the read operation or may require an explicit reset.

Note: Designers may want to limit the number of devices sharing an interrupt level for performance and latency considerations.

The interrupt controller on level-sensitive systems requires the interrupt request to be inactive at the time the EOI is sent; otherwise, a "new" interrupt request will be detected and another microprocessor interrupt caused.

To avoid this problem, a level-sensitive interrupt handler must clear the interrupt condition (usually by a Read or Write to an I/O port on the device causing the interrupt). After clearing the interrupt condition, a JMP $+2 should be executed prior to sending the EOI to the interrupt controller. This ensures that the interrupt request is removed prior to re-enabling the interrupt controller. Another JMP $+2 should be executed after sending the EOI, but prior to enabling the interrupt through the Set Interrupt Enable Flag (STI) command.

In the level-sensitive systems, hardware prevents the interrupt controllers from being set to the edge-sensitive mode.

Hardware interrupt IRQ9 is defined as the replacement interrupt level for the cascade level IRQ2. Program interrupt sharing should be implemented on IRQ2, interrupt hex 0A. The following processing occurs to maintain compatibility with the IRQ2 used by IBM Personal Computer products:

1. A device drives the interrupt request active on IRQ2 of the channel.
2. This interrupt request is mapped in hardware to IRQ9 input on the second interrupt controller.
3. When the interrupt occurs, the system microprocessor passes control to the IRQ9 (interrupt hex 71) interrupt handler.
4. This interrupt handler performs an end of interrupt (EOI) to the second interrupt controller and passes control to IRQ2 (interrupt hex 0A) interrupt handler.
5. This IRQ2 interrupt handler, when handling the interrupt, causes the device to reset the interrupt request prior to performing an EOI to the master interrupt controller that finishes servicing the IRQ2 request.

Software Interrupts

With the advent of software interrupt sharing, software interrupt routines must daisy chain interrupts. Each routine must check the function value and if it is not in the range of function calls for that routine, it must transfer control to the next routine in the chain. Because software interrupts are initially pointed to 0:0, before daisy chaining, it is necessary to check for this case. If the next routine is pointed to 0:0 and the function call is out of range, the appropriate action is to set the carry flag and do a RET 2 to indicate an error condition.

High-Level Language Considerations

The IBM-supported languages of IBM C, BASIC, FORTRAN, COBOL, and Pascal are the best choices for writing compatible programs.

If a program uses specific features of the hardware, that program may not be compatible with all IBM Personal Computer and Personal System/2 products. Specifically, the use of assembler language subroutines or hardware-specific commands (In, Out, Peek, Poke, ...) must follow the assembler language rules. See "Assembler Language Programming Considerations" on page 9-8.

Any program that requires precise timing information should obtain it through an operating system or language interface; for example, TIME$ in BASIC. If greater precision is required, the assembler techniques in "Assembler Language Programming Considerations" are available. The use of programming loops may prevent a program from being compatible with other IBM Personal Computer products, IBM Personal System/2 products, and software.

Assembler Language Programming Considerations

This section describes fundamental differences between the systems in the Personal Computer and Personal System/2 product lines that may affect program development.

Opcodes

The following opcodes work differently on systems using either the 80286 or 80386 microprocessor than they do on systems using the 8088 or 8086 microprocessor.

- PUSH SP

  The 80286 and 80386 microprocessors push the current stack pointer; the 8088 and 8086 microprocessors push the new stack pointer, that is, the value of the stack pointer after the PUSH SP instruction is completed.

- Single step interrupt (when TF = 1) on the interrupt instruction (Opcode hex CC, CD):

The 80286 and 80386 microprocessors do not perform a single-step interrupt on the INT instruction; the 8088 and 8086 microprocessors do perform a single-step interrupt on the INT instruction.

- The divide error exception (interrupt 0):

  The 80286 and 80386 microprocessors push the CS:IP of the instruction that caused the exception; the 8088 and 8086 microprocessors push the CS:IP of the instruction following the instruction that caused the exception.

- Shift counts for the 80286 and 80386 microprocessors:

  Shift counts are masked to 5 bits. Shift counts greater than 31 are treated mod 32. For example, a shift count of 36 shifts the operand four places.

- LOCK prefix:

  When the LOCK prefix is used with an instruction, the system microprocessor executes the entire instruction before allowing interrupts. If a Repeat String Move instruction is locked, interrupts may be disabled for a long duration.

  The 8088, 8086, and 80286 microprocessors allow the LOCK prefix to be used with most instructions. However, the 80386 microprocessor restricts the use of LOCK to the following instructions:

  - Bit Test and Set Memory, Register/Immediate
  - Bit Test and Reset Memory, Register/Immediate
  - Bit Test and Complement Memory, Register/Immediate
  - XCHG Register, Memory
  - XCHG Memory, Register
  - ADD, OR, ADC, SBB, Memory, Register/Immediate
  - AND, SUB, XOR Memory, Register/Immediate
  - NOT, NEG, INC, DEC Memory.

  An undefined opcode trap (INT 6) is generated if the LOCK prefix is used in the 80386 environment with an instruction not listed.

  When the 80286 is operating in the virtual memory mode, the LOCK prefix is IOPL-sensitive. Since the 80386 restricts the use

of the LOCK prefix to a specific set of instructions, the LOCK prefix is not IOPL-sensitive in the 80386 environment.

- Multiple lockout instructions:

  There are several microprocessor instructions that, when executed, lock out external bus signals. DMA requests are not honored during the execution of these instructions. Consecutive instructions of this type prevent DMA activity from the start of the first instruction to the end of the last instruction. To allow for necessary DMA cycles, as required by the diskette controller in a multitasking system, multiple lock-out instructions must be separated by a JMP SHORT $+2.

- Back-to-back I/O commands:

  Back-to-back I/O commands to the same I/O ports do not permit enough recovery time for some I/O adapters. To ensure enough time, a JMP SHORT $+2 must be inserted between IN/OUT instructions to the same I/O adapters.

  Note: MOV AL,AH type instruction does not allow enough recovery time. An example of the correct procedure follows:

```
OUT  IO_ADD,AL
JMP  SHORT $+2
MOV  AL,AH
OUT  IO_ADD,AL
```

- I/O commands followed by an STI instruction:

  I/O commands followed immediately by an STI instruction do not permit enough recovery time for some system board and channel operations. To ensure enough time, a JMP SHORT $+2 must be inserted between the I/O command and the STI instruction.

  Note: MOV AL,AH type instruction does not allow enough recovery time. An example of the correct procedure follows:

```
OUT  IO_ADD,AL
JMP  SHORT $+2
MOV  AL,AH
STI
```

- NT bit and IOPL bits:

  When the 80286 is operating in the Real Address Mode, the NT and IOPL bits in the flag register cannot be changed; the bits are zero.

  The 80386 allows the NT bit and the IOPL bits to be modified by POP stack into flags, and other instructions, while operating in the Real Address Mode. This has no effect on the Real Address Mode operation. However, upon entering Protected Mode operation, the NT bit should be cleared to prevent erroneous execution of the IRET instruction. If NT is set, the IRET attempts to perform a task switch to the previous task.

- Overlap of OUT and following instructions:

  The 80386 has a delayed write to memory and delayed output-to-I/O capability. It is possible for the actual output cycle to I/O devices to occur after the completion of instructions following the Out instruction. Under certain conditions, this may cause some programs to behave in an undesirable manner. For example, an interrupt handler routine may output an EOI command to the interrupt controller to drop the interrupt request. If the interrupt handler has an STI instruction following the output instruction, the 80386 may re-enable interrupts before the interrupt controller drops the interrupt request. This could cause the interrupt routine to be reentered.

  To avoid this problem, either of the following procedures may be used:

  - Place a JMP SHORT $+2 instruction between the OUT instruction and the STI instruction, or
  - Read back the status from the interrupt controller before executing the STI instruction.

- Math Coprocessor Instructions:

In 80386-based systems, the mode of the microprocessor and math coprocessor are tightly coupled. This is not the case for 80286-based systems. The 80286-based systems require the math coprocessor FSETPM instruction to be executed to enable the 80287 to operate in the Protected Mode. The 80287 remains in the Protected Mode until it is reset.

The mode of the 80287 determines the format in which the math coprocessor state information is saved by the FSTENV and FSAVE instructions. In the Protected Mode, the instruction and data operand pointers are saved as selector/offset pairs; in the Real Address Mode, the physical address and opcode are saved.

If the FSETPM instruction is encountered in the 80386 environment, it is ignored. The formatting is performed by the 80386, which internally maintains the instruction and data operand pointers. The Real Address Mode format image is saved when the 80386 is operating in the Real Address Mode or Virtual 8086 Mode. The Protected Mode format is used otherwise.

- Use of 32-bit registers and the 32-bit addressing mode:

  It is possible to use the 32-bit registers and 32-bit addressing mode in all operating modes of the 80386 through the use of the operand-size prefix or address-size prefix.

  In a multitasking environment, extreme care must be taken to avoid conflicts with other tasks that use extended registers. If the operating system saves the extended 32-bit registers and new segment registers in the task context save area, conflicts will be avoided; if the operating system does not provide this function, another method must be implemented.

  One possible method is to disable the interrupts while using the extended registers. The extended registers should be saved before use and restored immediately after use while the interrupts are still disabled. The time that interrupts are disabled should be kept as short as possible.

- Operand Alignment:

  When multiple bus cycles are required to transfer a multibyte logical operand (for example, a word operand beginning at an

address not evenly divisible by 2), the 80386 transfers the highest order bytes first.

This characteristic may affect adapters with memory-mapped I/O that require or assume that sequential memory accesses are made to the memory I/O ports.

This problem may be avoided by using a REP MOVB(yte) instead of a REP MOVSW(ord)

80286 Anomalies

In the Protected Mode, when any of the null selector values (0000H, 0001H, 0002H, 0003H) are loaded into the DS or ES registers with a MOV or POP instruction or a task switch, the 80286 always loads the null selector 0000H into the corresponding register.

If a coprocessor (80287) operand is read from an "executable and readable" and conforming (ERC) code segment, and the coprocessor operand is sufficiently near the segment limit that the second or subsequent byte lies outside the limit, no protection exception #9 will be generated.

The following describes the operation of all 80286 parts:

- Instructions longer than 10 bytes (instructions using multiple redundant prefixes) generate exception #13 (General Purpose Exception) in both the Real Address Mode and Protected Mode.
- If the second operand of an ARPL instruction is a null selector, the instruction generates an exception #13.

80386 Anomalies

The following describes anomalies that apply to the B-1 stepping level of the 80386 microprocessor.

80386 Real Address Mode Operation

- FSAVE/FSTENV opcode field incorrect:

  The opcode of some numeric instructions is saved incorrectly in the FSAVE/FSTENV format image when the 80386 is operating in the Real Address Mode or Virtual 8086 Mode.

The power-on self-test (POST) code in the system ROM enables hardware interrupt 13 and sets up its vector (INT 75H) to point to a math coprocessor exception routine in ROM. Any time this routine is executed as a result of an exception, it repairs the opcode field by performing the following sequence:

1. Clears the 'busy' signal latch
2. Executes FNSTENV (save image on stack)
3. Extracts instruction pointer from FSTENV memory image
4. Skips over prefix bytes until opcode is found
5. Inserts correct opcode information in the memory image
6. Executes FLDENV to restore the corrected opcode field
7. Writes the EOI command to the interrupt controller
8. Transfers control to the address pointed to by the NMI handler.

Any math coprocessor application containing an NMI handler should require its NMI handler to read the status of the coprocessor to determine if the NMI was caused by the coprocessor. If the interrupt was not generated by the coprocessor, control should be passed to the original NMI handler.

Applications do not require any modification for this errata because the BIOS exception routine repairs the opcode field after exceptions. However, if a debugger is used to display the math coprocessor state information, the opcode field will contain an incorrect value for some math coprocessor instructions.

- Single stepping repeated MOVS:

  If a repeated MOVS instruction is executed when single-stepping is turned on (TF = 1 in the EFLAGS register), a single-step interrupt is taken after two move steps on the 80386 microprocessor. The 8088, 8086, and 80286 microprocessors take a single-step interrupt after every iteration step. However, for the 80386, if a data breakpoint is encountered on the first iteration of a repeated MOVS, the data break is not taken until after the second iteration. Data breakpoints encountered on the second and subsequent iterations stop immediately after the step causing a break.

- Wrong register size for string instructions:

One of the (E)CX, (E)SI, or (E)DI registers will not be updated properly if certain string and loop instructions are followed by instructions that either:

- Use a different address size (that is, either the string instruction or the following instruction uses an address size prefix), or
- Reference the stack (such as PUSH/POP/CALL/RET) and the "B" bit in the SS descriptor is different from the address size used by the instructions.

The size of the register (16 bits or 32 bits) is taken from the instruction following the string instruction rather than from the string instruction itself. This could result in one of the following conditions:

- Only the lower 16 bits of a 32-bit instruction updated (if the 32-bit string instruction was followed by an instruction using a memory operand addressed with a 16-bit address).
- All 32 bits of a register updated rather than just the lower 16 bits.

The following is a list of the instructions and the affected registers:

Instruction	Register
REP MOVS	(E)SI
MOVS	(E)DI
STOS	(E)DI
INS	(E)DI
REP INS	(E)CX

Notes:

1. A 32-bit effective address size specified with a string instruction indicates that the 32-bit ESI and EDI registers should be used for forming addresses, and the 32-bit ECX register should be used as the count register.
2. A 32-bit operand size on a repeated string move (MOVS) should be used only if the compiler or programmer can guarantee that the strings do not overlap destructively. An 8-bit or 16-bit MOVS has a predictable effect when the strings overlap destructively.

Figure 9-2. String Instruction/Register Size Mismatch

The problem only occurs if instructions with different address sizes are mixed together, or if a code segment of one size is used with a stack segment of the other size.

To avoid this problem, add a NOP after each of the instructions listed in Figure 9-2 on page 9-15 and ensure that the NOP has the same address size as the string/loop instruction. If necessary, an address size prefix hex 67 may precede the NOP instruction.

- Wrong ECX update with REP INS:

  ECX (or CX in a 16-bit address size) is not updated correctly in the case of a REP INS[1] followed by an early start instruction[2]. After executing any repeat-prefixed instruction, the contents of ECX is supposed to be 0, but in the case of an REP INS instruction, ECX is not updated correctly and its contents becomes hex FFFFFFFF for 32-bit address size operations and hex 0FFFF for 16-bit address size operations. INS is still executed the correct number of times and EDI is updated properly.

  To avoid this problem, one of the following procedures may be used:

  - Insert an explicit MOV ECX,0 (or MOV CX,0) instruction after any REP INS instruction. This ensures that the contents of ECX or CX is 0.
  - Do not rely on the count in ECX (or CX) after a REP INS instruction but instead, move a new count into ECX (or CX) before using it again.

- Test register access fails:

  Accessing the Translate Lookaside Buffer (TLB) test registers, TR6 and TR7, may not function properly.

  Avoid using test registers TR6 and TR7 to test the TLB.

---

1 REP INS refers to any input string instruction with a repeat prefix.

2 An early start instruction refers to PUSH, POP, or memory reference instructions.

80286 Compatible Protected Mode Operation

- Math coprocessor Save/Restore environment operands:

  If either of the last two bytes of an FSAVE/FRSTOR or FSTENV/FLDENV is not accessible, the instruction cannot be restarted. An FNINIT instruction must be issued to the math coprocessor before any other math coprocessor instruction can be executed. This problem arises only in demand-paged systems, or demand-segmented systems that increase the segment size on demand.

- Wrap-around math coprocessor operands:

  The 80386 architecture does not permit a math coprocessor operand, or any other operand, to wrap around the end of a segment. If such an instruction is issued in a protected segmented system, and the operand starts and ends in valid parts of a segment, but passes through an inaccessible region of the segment, the math coprocessor may be put in an indeterminate state. Under these conditions, a FCLEX or FINIT must be sent before any other math coprocessor instruction is issued.

- Load Segment Limit instruction cannot precede PUSH/POP:

  If the instruction executed immediately after a Load Segment Limit (LSL) instruction does a stack operation, the value of (E)SP may be incorrect after the operation.

  Note: Stack operations resulting from non-instruction sources, such as exceptions or interrupts following the LSL, do not corrupt (E)SP.

  To avoid this problem, make sure that the instruction following an LSL instruction is never one that does a push to or pop off the stack. This includes PUSH, POP, RET, CALL, ENTER, and other such instructions. This can be achieved by always following an LSL instruction with a NOP instruction. Even if a forbidden instruction is used, (E)SP may be updated correctly since the problem is data-dependent and only occurs if the LSL operation succeeds (that is, sets the ZF flag).

- LSL/LAR/VERR/VERW malfunction with a NULL selector:

  An LSL, LAR, VERR, or VERW executed with a NULL selector (that is, bits 15 through 2 of the selector set to 0) operates on the descriptor at entry 0 of the Global Descriptor Table (GDT) instead of unconditionally clearing the ZF flag.

  This problem can be avoided by filling in the "NULL descriptor" (that is, the descriptor at entry 0 of the GDT) with all zeroes, which is an invalid descriptor type.

  The access to the "NULL descriptor" is made but fails since the descriptor has an invalid type. The failure is reported with ZF cleared, which is the desired behavior.

80386 Extended Protected Mode Operation

The following problems exist for operation in the Virtual 8086 Mode.

- Task switch to Virtual 8086 Mode does not set prefetch limit:

  The 80386 prefetch unit limit is not updated when doing a task switch to the Virtual 8086 Mode. This may cause an incorrect segment limit violation to be reported if the microprocessor instruction fetches the segment limit that existed before the task switch.

  This problem can be avoided by using an IRET with the appropriate items on the stack to start the Virtual 8086 task in place of the task switch method.

- FAR jump near page boundary in Virtual 8086 Mode:

  When paging is enabled in the Virtual 8086 Mode, and a direct FAR jump (opcode EA) instruction is located at the end of a page (or within 16 bytes of the end), and the next page is not cached in the TLB internal to the 80386, the FAR jump instruction leaves the prefetcher limit at the "end" of the old code segment instead of setting it at the "end" of the new code segment. This can allow execution off the end of the new segment to trigger a segment limit violation, or cause a spurious GP fault if the old and new segments overlap and a prefetch crosses the old segment limit.

There is no way to detect code "walking off" the end of a code segment. However, the spurious GP fault can be avoided by simply performing an IRET back to the instruction causing the fault. The IRET will set the prefetch limit correctly, providing the exception handler has the ability to determine a spurious GP fault from a "real" GP fault.

The following problems exist for operations with paging enabled.

- Coprocessor operands:

  To avoid having a nonstartable instruction involving math coprocessor operands in demand-paged systems, ensure the operands do not cross page boundaries. This can be accomplished by aligning math coprocessor operands in 128-byte boundaries within a segment, and aligning the start of segments on 128-byte physical boundaries.

- Page fault error code on stack is not reliable:

  When a page fault (exception 14) occurs, the three defined bits in the error code can be unreliable if a certain sequence of prefetches occurred at the same time.

  Although the page-fault error code pushed onto the page-fault handler stack is sometimes unreliable, the page-fault linear address stored in register CR2 is always correct. The page-fault handler should refer to the page-fault linear address in register CR2 to access the corresponding page table entry and thereby determine whether the page fault was due to a page-not-present condition or a usage violation.

- I/O relocated in paged systems:

  When paging is enabled (PG = 1 in CR0), accessing I/O addresses in the range hex 00001000 to hex 0000FFFF, or accessing a 80387 math coprocessor using ESC instructions (I/O addresses hex 800000F8 to hex 800000FF) can generate incorrect I/O addresses on A12 through A31 if the I/O address is the same as a memory linear address that is mapped by the TLB.

  The physical address corresponding to the memory linear address mapped by the TLB is ANDed with the I/O address, causing the I/O address to be incorrect in most cases.

A suggested method for handling normal I/O addresses between hex 00000 and hex 0FFFF is as follows: The operating system is required to map the lowest (first) 64K of linear address space to 16 pages, which are defined such that bits 12 through 15 of the linear and physical addresses are equal. This requires that the pages be aligned on a 64K physical boundary (the physical address associated with the first page has address bits 15 through 0 equal to 0).

A suggested method for handling the math coprocessor I/O addresses requires that the memory page at linear address hex 80000000 always be marked "not present" so it cannot be cached in the TLB. This may be accomplished in one of the following ways:

- Require the operating system to handle a 4K hole in the linear address space at the 2G boundary.
- Restrict the linear address space to a 2G maximum instead of 4G. No segments will have a linear address above the 2G boundary.

- Spurious page level protection fault:

  This problem only occurs when the page table and the directory entries that map the stacks for the inner levels of a task are marked as supervisor access only, and an external bus HOLD comes during the cycle that pops (E)SP off the stack during an inter-level RET or IRET.

  This problem can be avoided by marking the pages that map the inner level stacks (level 0, 1, and 2) to permit user read access. The segmentation protection mechanism can be used to prevent user access to the linear addresses containing these stacks, if required.

ROM BIOS and Operating System Function Calls

For maximum portability, programs should perform all I/O operations through operating system function calls. In environments where the operating system does not provide the necessary programming interfaces, programs should access the hardware through ROM BIOS function calls, if permissible.

- In some environments, program interrupts are used for access to these functions. This practice removes the absolute addressing from the program. Only the interrupt number is required.
- The coprocessor detects six different exception conditions that can occur during instruction execution. If the appropriate exception mask within the coprocessor is not set, the coprocessor sets the 'error' signal. This 'error' signal generates a hardware interrupt 13 (IRQ 13) causing the 'busy' signal to be held in the busy state. The 'busy' signal can be cleared by an 8-bit I/O Write command to address hex 00F0 with bits D0 through D7 equal to 0.

  The power-on self-test code in the system ROM enables hardware IRQ 13 and sets up its vector to point to a routine in ROM. The ROM routine clears the 'busy' signal latch and then transfers control to the address pointed to by the NMI vector. This maintains code compatibility across the IBM Personal Computer and Personal System/2 product lines. The NMI handler reads the status of the coprocessor to determine if the NMI was caused by the coprocessor. If the interrupt was not caused by the coprocessor, control is passed to the original NMI handler.
- In systems using the 80286 or 80386 microprocessor, IRQ 9 is redirected to INT hex 0A (hardware IRQ 2). This ensures that hardware designed to use IRQ 2 will operate in these systems. See "Hardware Interrupts" on page 9-6 for more information.
- The system can mask hardware sensitivity. New devices can change the ROM BIOS to accept the same programming interface on the new device.
- In cases where BIOS provides parameter tables, such as for video or diskette, a program can substitute new parameter values by building a new copy of the table and changing the vector to point to that table. However, the program should copy the current table, using the current vector, and then modify those locations in the table that need to be changed. In this way, the program does not inadvertently change any values that should be left the same.
- The Diskette Parameters Table pointed to by INT hex 1E consists of 11 parameters required for diskette operation. It is recommended that the values supplied in ROM be used. If it becomes necessary to modify any of the parameters, build another parameter block and modify the address at INT hex 1E (0:78) to point to the new block.

The parameters were established to allow:

- Some models of the IBM Personal Computer to operate both the 5.25-inch high capacity diskette drive (96 tracks per inch) and the 5.25-inch double-sided diskette drive (48 tracks per inch).
- Some models of the Personal System/2 to operate both the 3.5-inch 1.44M diskette drive and the 3.5-inch 720KB diskette drive.

The Gap Length Parameter is not always retrieved from the parameter block. The gap length used during diskette read, write, and verify operations is derived from within diskette BIOS. The gap length for format operations is still obtained from the parameter block.

Note: Special considerations are required for format operations. Refer to the diskette section of the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for the required details.

If a parameter block contains a head settle time parameter value of 0 milliseconds, and a write or format operation is being performed, the following minimum head settle times are enforced.

Drive Type	Head Settle Time
5.25-Inch Diskette Drives:	
Double Sided (48 TPI)	20 milliseconds
High Capacity (96 TPI)	15 milliseconds
3.5-Inch Diskette Drives:	
720K	20 milliseconds
1.44M	15 milliseconds

Figure 9-3. Write and Format Head Settle Time

Read and verify operations use the head settle time provided by the parameter block.

If a parameter block contains a motor start wait parameter of less than 500 milliseconds (1 second for a Personal Computer product) for a write or verify operation, diskette BIOS enforces a minimum time of 500 milliseconds (1 second for a Personal Computer product). Read and write operations use the motor start time provided by the parameter block.

- Programs may be designed to reside on both 5.25-inch or 3.5-inch diskettes. Since not all programs are operating-system dependent, the following procedure can be used to determine the type of media inserted into a diskette drive.

  1. Verify Track 0, Head 0, Sector 1 (1 sector): This allows diskette BIOS to determine if the format of the media is a recognizable type.

     If the verify operation fails, issue the reset function (AH=0) to diskette BIOS and try the operation again. If another failure occurs, the media needs to be formatted or is defective.

  2. Verify Track 0, Head 0, Sector 16 (1 sector).

     If the verify operation fails, either a 5.25-inch (48 TPI) or 3.5-inch 720KB diskette is installed. The type can be determined by verifying Track 78, Head 1, Sector 1 (1 sector). A successful verification of Track 78 indicates a 3.5-inch 720KB diskette is installed; a verification failure indicates a 5.25-inch (48 TPI) diskette is installed.

     Note: Refer to the *DOS Technical Reference* for the File Allocation Table parameters for single-sided and double-sided diskettes.

  3. Read the diskette controller status in BIOS starting with address 40:42. The fifth byte defines the head that the operation ended with. If the operation ended with head 1, the diskette is a 5.25-inch high capacity (96 TPI) diskette; if the operation ended with head 0, the diskette is a 3.5-inch 1.44M diskette.

Hardware Compatibility

The Personal System/2 products maintain many of the interfaces used by the IBM Personal Computer AT. In most cases command and status organization of these interfaces is maintained.

The functional interfaces for the Model 80 are compatible with the following interfaces:

- The Intel 8259 interrupt controllers (without edge triggering)
- The Intel 8253 timers driven from 1.190 MHz (timer 0 and 2 only)

- The Intel 8237 DMA controller-address/transfer counters, page registers and status fields only. The Command and Request registers are not supported. The rotate and mask functions are not supported. The Mode register is partially supported.
- The NS16450 serial port
- The Intel 8088, 8086, and 80286 microprocessors
- The Intel 8272 diskette drive controller (level-sensitive interrupt)
- The Motorola MC146818 Time of Day Clock command and status (CMOS reorganized)
- The Intel 8042 keyboard port at address hex 0060
- Display modes supported by the IBM Monochrome Display and Printer Adapter, the IBM Color/Graphics Monitor Adapter, and the IBM Enhanced Graphics Adapter
- The parallel printer ports (Parallel 1, Parallel 2, and Parallel 3) in compatibility mode
- Generally compatible with the Intel 80287 and 8087 math coprocessors (See "Math Coprocessor Compatibility" on page 3-10 for limitations).

Multitasking Provisions

The BIOS contains a feature to assist multitasking implementation. "Hooks" are provided for a multitasking dispatcher. Whenever a busy (wait) loop occurs in the BIOS, a hook is provided for the program to break out of the loop. Also, whenever BIOS services an interrupt, a corresponding wait loop is exited, and another hook is provided. Thus a program can be written that employs the bulk of the device driver code. The following is valid only in the Real Address Mode and must be taken by the code to allow this support.

- The program is responsible for the serialization of access to the device driver. The BIOS code is not reentrant.
- The program is responsible for matching corresponding Wait and Post calls.

Warning: 32-bit operations to the video subsystem can cause a diskette overrun in the 1.44M mode because data width conversions may require more than 12 microseconds. If an overrun occurs, BIOS returns an error code and the operation should be retried.

Interfaces

There are four interfaces to be used by the multitasking dispatcher:

Startup: First, the startup code hooks interrupt hex 15. The dispatcher is responsible to check for function codes of AH = hex 90 or 91. The "Wait" and "Post" sections describe these codes. The dispatcher must pass all other functions to the previous user of interrupt hex 15. This can be done by a JMP or a CALL. If the function code is hex 90 or 91, the dispatcher should do the appropriate processing and return by the IRET instruction.

Serialization: It is up to the multitasking system to ensure that the device driver code is used serially. Multiple entries into the code can result in serious errors.

Wait: Whenever the BIOS is about to enter a busy loop, it first issues an interrupt hex 15 with a function code of hex 90 in AH. This signals a wait condition. At this point, the dispatcher should save the task status and dispatch another task. This allows overlapped execution of tasks when the hardware is busy. The following is an outline of the code that has been added to the BIOS to perform this function.

```
MOV AX, 90XXH          ; wait code in AH and
                       ; type code in AL
INT 15H                ; issue call
JC  TIMEOUT            ; optional: for time-out or
                       ; if carry is set, time-out
                       ; occurred
NORMAL TIMEOUT LOGIC ; normal time-out
```

Post: Whenever the BIOS has set an interrupt flag for a corresponding busy loop, an interrupt hex 15 occurs with a function code of hex 91 in AH. This signals a Post condition. At this point, the dispatcher should set the task status to "ready to run" and return to the interrupt routine. The following is an outline of the code added to BIOS that performs this function.

```
MOV AX, 91XXH          ; post code AH and
                       ; type code AL
INT 15H                ; issue call
```

Classes

The following types of wait loops are supported:

- The class for hex 0 to 7F is serially reusable. This means that for the devices that use these codes, access to the BIOS must be restricted to only one task at a time.
- The class for hex 80 to BF is reentrant. There is no restriction on the number of tasks that can access the device.
- The class for hex C0 to FF is noninterrupt. There is no corresponding interrupt for the wait loop. Therefore, it is the responsibility of the dispatcher to determine what satisfies this condition to exit from the loop.

Function Code Classes

Type Code (AL)	Description
00H->7FH	Serially reusable devices; the operating system must serialize access
80H->0BFH	Reentrant devices; ES:BX is used to distinguish different calls (multiple I/O calls are allowed simultaneously).
0C0H->0FFH	Wait-only calls; there is no complementary Post for these waits--these are time-out only. Times are function-number dependent.

Function Code Assignments: The following are specific assignments for the Model 50, Model 60, and Model 80 BIOS. Times are approximate. They are grouped according to the classes described under "Function Code Classes."

Type Code (AL)	Time Out	Description
00H	Yes (12 seconds)	Fixed Disk
01H	Yes (2 seconds)	Diskette
02H	No	Keyboard
0FCH	Yes	Fixed Disk Reset
0FDH	Yes (500-ms Read/Write)	Diskette Motor Start
0FEH	Yes (20 seconds)	Printer

Figure 9-4. Functional Code Assignments

The asynchronous support has been omitted. The serial and parallel controllers generate interrupts, but BIOS does not support them in the interrupt mode. Therefore, the support should be included in the multitasking system code if that device is to be supported.

Time-Outs

To support time-outs properly, the multitasking dispatcher must be aware of time. If a device enters a busy loop, it generally should remain there for a specific amount of time before indicating an error. The dispatcher should return to the BIOS wait loop with the carry bit set if a time-out occurs.

Machine-Sensitive Programs

Programs can select machine-specific features, but they must first identify the machine and model type. IBM has defined methods for uniquely determining the specific machine type. The location of the machine model bytes can be found through interrupt 15 function code (AH) = hex C0. The model bytes for the Model 80 are shown in the following figure.

Model Byte	Sub-Model Byte	Product Name
F8	00	Model 80

Figure 9-5. Machine Model Bytes

See the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for a listing of model bytes for other IBM products.

Glossary

This glossary includes terms and definitions from the *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems*, GC20-1699.

μ. Prefix micro; 0.000,001.

μs. Microsecond; 0.000,001 second.

A. Ampere.

ac. Alternating current.

accumulator. A register in which the result of an operation is formed.

active high. Designates a signal that has to go high to produce an effect. Synonymous with positive true.

active low. Designates a signal that has to go low to produce an effect. Synonymous with negative true.

adapter. An auxiliary device or unit used to extend the operation of another system.

address bus. One or more conductors used to carry the binary-coded address from the processor throughout the rest of the system.

algorithm. A finite set of well-defined rules for the solution of a problem in a finite number of steps.

all points addressable (APA). A mode in which all points of a displayable image can be controlled by the user.

alphameric. Synonym for alphanumeric.

alphanumeric (A/N). Pertaining to a character set that contains letters, digits, and usually other characters, such as punctuation marks. Synonymous with alphameric.

alternating current (ac). A current that periodically reverses its direction of flow.

American National Standard Code for Information Interchange (ASCII). The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information exchange between data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

ampere (A). The basic unit of electric current.

A/N. Alphanumeric.

analog. (1) Pertaining to data in the form of continuously variable physical quantities. (2) Contrast with digital.

AND. A logic operator having the property that if P is a statement, Q is

a statement, R is a statement,..., then the AND of P, Q, R,...is true if all statements are true, false if any statement is false.

AND gate. A logic gate in which the output is 1 only if all inputs are 1.

APA. All points addressable.

arbitration. A process by which bus masters and DMA slaves compete for ownership of the channel in a prioritized fashion.

ASCII. American National Standard Code for Information Interchange.

assemble. To translate a program expressed in an assembler language into a computer language.

assembler. A computer program used to assemble.

assembler language. A computer-oriented language whose instructions are usually in one-to-one correspondence with computer instructions.

asynchronous transmission. (1) Transmission in which the time of occurrence of the start of each character, or block of characters, is arbitrary; once started, the time of occurrence of each signal representing a bit within a character, or block, has the same relationship to significant instants of a fixed time frame. (2) Transmission in which each information character is individually transmitted (usually timed by the use of start elements and stop elements).

audio frequencies. Frequencies that can be heard by the human ear (approximately 15 hertz to 20,000 hertz).

BASIC. Beginner's all-purpose symbolic instruction code.

basic input/output system (BIOS). The feature that provides the level control of the major I/O devices, and relieves the programmer from concern about hardware device characteristics.

baud. (1) A unit of signaling speed equal to the number of discrete conditions or signal events per second. For example, one baud equals one bit per second in a train of binary signals, one-half dot cycle per second in Morse code, and one 3-bit value per second in a train of signals each of which can assume one of eight different states. (2) In asynchronous transmission, the unit of modulation rate corresponding to one unit of interval per second; that is, if the duration of the unit interval is 20 milliseconds, the modulation rate is 50 baud.

beginner's all-purpose symbolic instruction code (BASIC). A programming language with a small repertoire of commands and a simple syntax, primarily designed for numeric applications.

binary. (1) Pertaining to a selection, choice, or condition that has two possible values or states. (2) Pertaining to a fixed radix numeration system having a radix of 2.

binary digit. (1) In binary notation, either of the characters 0 or 1. (2) Synonymous with bit.

binary notation. Any notation that uses two different characters, usually the binary digits 0 and 1.

binary synchronous communications (BSC). A uniform procedure, using a standardized set of control characters and control character sequences for synchronous transmission of binary-coded data between stations.

BIOS. Basic input/output system.

bit. Synonym for binary digit

bits per second (bps). A unit of measurement representing the number of discrete binary digits transmitted by a device in one second.

block. (1) A string of records, a string of words, or a character string formed for technical or logic reasons to be treated as an entity. (2) A set of things, such as words, characters, or digits, treated as a unit.

block-check character (BCC). In cyclic redundancy checking, a character that is transmitted by the sender after each message block and is compared with a block-check character computed by the receiver to determine if the transmission was successful.

bps. Bits per second.

BSC. Binary synchronous communications.

bus master. A device that arbitrates for ownership of the channel and, upon winning ownership, issues address and control signals to a slave.

buffer. (1) An area of storage that is temporarily reserved for use in performing an input/output operation, into which data is read or from which data is written. Synonymous with I/O area. (2) A portion of storage for temporarily holding input or output data.

bus. One or more conductors used for transmitting signals or power.

byte. (1) A sequence of eight adjacent binary digits that are operated upon as a unit. (2) A binary character operated upon as a unit. (3) The representation of a character.

C. Celsius.

capacitor. An electronic circuit component that stores an electric charge.

CAS. Column address strobe.

cathode ray tube (CRT). A vacuum tube in which a stream of electrons is projected onto a fluorescent screen producing a luminous spot. The location of the spot can be controlled.

cathode ray tube display (CRT display). (1) A CRT used for displaying data. For example, the electron beam can be controlled to form alphanumeric data by use of a dot matrix. (2) Synonymous with monitor.

CCITT. International Telegraph and Telephone Consultative Committee.

Celsius (C). A temperature scale. Contrast with Fahrenheit (F).

central processing unit (CPU). Term for processing unit.

channel. A path along which signals can be sent; for example, data channel, output channel.

character generator. (1) In computer graphics, a functional unit that converts the coded representation of a graphic character into the shape of the character for display. (2) In word processing, the means within equipment for generating visual characters or symbols from coded data.

character set. (1) A finite set of different characters upon which agreement has been reached and that is considered complete for some purpose. (2) A set of unique representations called characters. (3) A defined collection of characters.

characters per second (cps). A standard unit of measurement for the speed at which a printer prints.

check key. A group of characters, derived from and appended to a data item, that can be used to detect errors in the data item during processing.

clipping. In computer graphics, removing parts of a display image that lie outside a window.

closed circuit. A continuous unbroken circuit; that is, one in which current can flow. Contrast with open circuit.

CMOS. Complementary metal oxide semiconductor.

code. (1) A set of unambiguous rules specifying the manner in which data may be represented in a discrete form. Synonymous with coding scheme. (2) A set of items, such as abbreviations, representing the members of another set. (3) To represent data or a computer program in a symbolic form that can be accepted by a data processor. (4) Loosely, one or more computer programs, or part of a computer program.

collector. An element in a transistor toward which current flows.

column address strobe (CAS). A signal that latches the column addresses in a memory chip.

compile. (1) To translate a computer program expressed in a problem-oriented language into a computer-oriented language. (2) To prepare a machine-language program from a computer program written in another programming language by making use of the overall logic structure of the program, or generating more than one computer instruction for each symbolic statement, or both, as well as performing the function of an assembler.

complement. A number that can be derived from a specified number by

subtracting it from a second specified number.

complementary metal oxide semiconductor (CMOS). A logic circuit family that uses very little power. It works with a wide range of power supply voltages.

computer. A functional unit that can perform substantial computation, including numerous arithmetic operations or logic operations, without human intervention during a run.

computer instruction code. A code used to represent the instructions in an instruction set. Synonymous with machine code.

computer program. A sequence of instructions suitable for processing by a computer.

computer word. A word stored in one computer location and capable of being treated as a unit.

configuration. (1) The arrangement of a computer system or network as defined by the nature, number, and the chief characteristics of its functional units. More specifically, the term configuration may refer to a hardware configuration or a software configuration. (2) The devices and programs that make up a system, subsystem, or network.

conjunction. Synonym for AND operation.

contiguous. Touching or joining at the edge or boundary; adjacent.

control character. A character whose occurrence in a particular context initiates, modifies, or stops a control operation.

control operation. An action that affects the recording, processing, transmission, or interpretation of data; for example, starting or stopping a process, carriage return, font change, rewind, and end of transmission.

control storage. A portion of storage that contains microcode.

coordinate space. In computer graphics, a system of Cartesian coordinates in which an object is defined.

cps. Characters per second.

CPU. Central processing unit.

CRC. Cyclic redundancy check.

CRT. Cathode ray tube.

CRT display. Cathode ray tube display.

CTS. Clear to send. Associated with modem control.

cursor. (1) In computer graphics, a movable marker that is used to indicate a position on a display. (2) A displayed symbol that acts as a marker to help the user locate a point in text, in a system command, or in storage. (3) A movable spot of light on the screen of a display device, usually indicating where the next character is to be entered, replaced, or deleted.

cyclic redundancy check (CRC). (1) A redundancy check in which the check key is generated by a

cyclic algorithm. (2) A system of error checking performed at both the sending and receiving station after a block-check character has been accumulated.

cylinder. (1) The set of all tracks with the same nominal distance from the axis about which the disk rotates. (2) The tracks of a disk storage device that can be accessed without repositioning the access mechanism.

data. (1) A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by human or automatic means. (2) Any representations, such as characters or analog quantities, to which meaning is, or might be assigned.

data transmission. Synonym for transmission.

dB. Decibel.

dc. Direct current.

decibel. (1) A unit that expresses the ratio of two power levels on a logarithmic scale. (2) A unit for measuring relative power.

decoupling capacitor. A capacitor that provides a low impedance path to ground to prevent common coupling between circuits.

Deutsche Industrie Norm (DIN). (1) German Industrial Norm. (2) The committee that sets German dimension standards.

digit. (1) A graphic character that represents an integer; for example, one of the characters 0 to 9. (2) A symbol that represents one of the non-negative integers smaller than the radix. For example, in decimal notation, a digit is one of the characters 0 to 9.

digital. (1) Pertaining to data in the form of digits. (2) Contrast with analog.

DIN. Deutsche Industrie Norm.

DIN connector. One of the connectors specified by the DIN committee.

DIP. Dual in-line package.

DIP switch. One of a set of small switches mounted in a dual in-line package.

direct current (dc). A current that always flows in one direction.

direct memory access (DMA). A method of transferring data between main storage and I/O devices that does not require processor intervention.

disable. To stop the operation of a circuit or device.

disabled. Pertaining to a state of a processing unit that prevents the occurrence of certain types of interruptions. Synonymous with masked.

disk. Loosely, a magnetic disk.

diskette. A thin, flexible magnetic disk and a semirigid protective jacket, in which the disk is permanently enclosed. Synonymous with flexible disk.

diskette drive. A device for storing data on and retrieving data from a diskette.

display. (1) A visual presentation of data. (2) A device for visual presentation of information on any temporary character imaging device. (3) To present data visually. (4) See cathode ray tube display.

display attribute. In computer graphics, a particular property that is assigned to all or part of a display; for example, low intensity, green color, blinking status.

display element. In computer graphics, a basic graphic element that can be used to construct a display image; for example, a dot, a line segment, a character.

display image. In computer graphics, a collection of display elements or display groups that are represented together at any one time in a display space.

display surface. In computer graphics, that medium on which display images may appear; for example, the entire screen of a cathode ray tube.

DMA. Direct memory access.

DMA controller. A device that does NOT arbitrate for ownership of the channel but tracks the arbitration process. If programmed to support the winning arbitration level, the DMA controller issues address and control signals so as to move data to or from the DMA slave that won ownership of the bus through arbitration.

DMA slave. A device that arbitrates for ownership of the channel and, upon winning ownership, does NOT issue address and control signals. Rather, it responds to the control signals from a DMA controller by executing the specified Read or Write command.

dot-matrix character generator. In computer graphics, a character generator that generates character images composed of dots.

DSR. Data set ready. Associated with modem control.

DTR. Data terminal ready, associated with modem control.

dual in-line package (DIP). A widely used container for an integrated circuit. DIPs have pins in two parallel rows. The pins are spaced 1/10 inch apart. See also DIP switch.

duplex. (1) In data communication, pertaining to a simultaneous two-way independent transmission in both directions. (2) Contrast with half-duplex.

duty cycle. In the operation of a device, the ratio of on time to idle time. Duty cycle is expressed as a decimal or percentage.

dynamic memory. RAM using transistors and capacitors as the memory elements. This memory requires a refresh (recharge) cycle every few milliseconds. Contrast with static memory.

edge connector. A terminal block with a number of contacts attached

to the edge of a printed-circuit board to facilitate plugging into a foundation circuit.

EIA. Electronic Industries Association.

electromagnet. Any device that exhibits magnetism only while an electric current flows through it.

enable. To initiate the operation of a circuit or device.

end of block (EOB). A code that marks the end of a block of data.

end of file (EOF). An internal label, immediately following the last record of a file, signaling the end of that file. It may include control totals for comparison with counts accumulated during processing.

error checking and correction (ECC). The detection and correction of all single-bit errors, plus the detection of double-bit and some multiple-bit errors.

ESC. The escape character.

escape character (ESC). A code extension character used, in some cases, with one or more succeeding characters to indicate by some convention or agreement that the coded representations following the character or the group of characters are to be interpreted according to a different code or according to a different coded character set.

ETB. End-of-transmission-block.

ETX. End-of-text.

F. Fahrenheit.

Fahrenheit (F). A temperature scale. Contrast with Celsius (C).

falling edge. Synonym for negative-going edge.

FCC. Federal Communications Commission.

fetch. To locate and load a quantity of data from storage.

field. (1) In a record, a specified area used for a particular category of data. (2) In a data base, the smallest unit of data that can be referred to.

FIFO (first-in-first-out). A queuing technique in which the next item to be retrieved is the item that has been in the queue for the longest time.

fixed disk drive. In the IBM Personal Computer, a unit consisting of nonremovable magnetic disks, and a device for storing data on and retrieving data from the disks.

flag. (1) Any of various types of indicators used for identification. (2) A character that signals the occurrence of some condition, such as the end of a word. (3) Deprecated term for mark.

flexible disk. Synonym for diskette.

flip-flop. A circuit or device containing active elements, capable of assuming either one of two stable states at a given time.

font. A family or assortment of characters of a given size and style;

for example, 10 point Press Roman medium.

foreground. (1) In multiprogramming, the environment in which high-priority programs are executed. (2) On a color display screen, the characters as opposed to the background.

format. The arrangement or layout of data on a data medium.

frame. (1) In SDLC, the vehicle for every command, every response, and all information that is transmitted using SDLC procedures. Each frame begins and ends with a flag. (2) In data transmission, the sequence of contiguous bits bracketed by and including beginning and ending flag sequences.

g. Gram.

G. (1) Prefix giga; 1,000,000,000. (2) When referring to computer storage capacity, 1,073,741,824. (1,073,741,824 = 2 to the 30th power.)

gate. (1) A combinational logic circuit having one output channel and one or more input channels, such that the output channel state is completely determined by the input channel states. (2) A signal that enables the passage of other signals through a circuit.

Gb. 1,073,741,824 bytes.

giga (G). Prefix 1,000,000,000.

gram (g). A unit of weight (equivalent to 0.035 ounces).

graphic. A symbol produced by a process such as handwriting, drawing, or printing.

graphic character. A character, other than a control character, that is normally represented by a graphic.

hardware. (1) Physical equipment used in data processing, as opposed to programs, procedures, rules, and associated documentation. (2) Contrast with software.

head. A device that reads, writes, or erases data on a storage medium; for example, a small electromagnet used to read, write, or erase data on a magnetic disk.

hertz (Hz). A unit of frequency equal to one cycle per second.

hex. Common abbreviation for hexadecimal.

hexadecimal. (1) Pertaining to a selection, choice, or condition that has 16 possible different values or states. These values or states are usually symbolized by the ten digits 0 through 9 and the six letters A through F. (2) Pertaining to a fixed radix numeration system having a radix of 16.

highlighting. In computer graphics, emphasizing a given display group by changing its attributes relative to other display groups in the same display field.

housekeeping. Operations or routines that do not contribute directly to the solution of the problem but do contribute directly to the operation of the computer.

Hz. Hertz

image. A fully processed unit of operational data that is ready to be transmitted to a remote unit; when loaded into control storage in the remote unit, the image determines the operations of the unit.

immediate instruction. An instruction that contains within itself an operand for the operation specified, rather than an address of the operand.

index register. A register whose contents may be used to modify an operand address during the execution of computer instructions.

indicator. (1) A device that may be set into a prescribed state, usually according to the result of a previous process or on the occurrence of a specified condition in the equipment, and that usually gives a visual or other indication of the existence of the prescribed state, and that may in some cases be used to determine the selection among alternative processes; for example, an overflow indicator. (2) An item of data that may be interrogated to determine whether a particular condition has been satisfied in the execution of a computer program; for example, a switch indicator, an overflow indicator.

inhibited. (1) Pertaining to a state of a processing unit in which certain types of interruptions are not allowed to occur. (2) Pertaining to the state in which a transmission control unit or an audio response unit cannot accept incoming calls on a line.

initialize. To set counters, switches, addresses, or contents of storage to 0 or other starting values at the beginning of, or at prescribed points in, the operation of a computer routine.

input/output (I/O). (1) Pertaining to a device or to a channel that may be involved in an input process, and, at a different time, in an output process. In the English language, "input/output" may be used in place of such terms as "input/output data," "input/output signal," and "input/output terminals," when such usage is clear in a given context. (2) Pertaining to a device whose parts can be performing an input process and an output process at the same time. (3) Pertaining to either input or output, or both.

instruction. In a programming language, a meaningful expression that specifies one operation and identifies its operands, if any.

instruction set. The set of instructions of a computer, of a programming language, or of the programming languages in a programming system.

intensity. In computer graphics, the amount of light emitted at a display point.

interface. A device that alters or converts actual electrical signals between distinct devices, programs, or systems.

interleave. To arrange parts of one sequence of things or events so that they alternate with parts of one or more other sequences of the same

nature and so that each sequence retains its identity.

interrupt. (1) A suspension of a process, such as the execution of a computer program, caused by an event external to that process, and performed in such a way that the process can be resumed. (2) In a data transmission, to take an action at a receiving station that causes the transmitting station to terminate a transmission. (3) Synonymous with interruption.

I/O. Input/output.

irrecoverable error. An error that makes recovery impossible without the use of recovery techniques external to the computer program or run.

k. Prefix kilo; 1000.

K. When referring to storage capacity, 1024. (1024 = 2 to the 10th power.)

Kb. 1024 bytes.

kg. Kilogram; 1000 grams.

kHz. Kilohertz; 1000 hertz.

kilo (k). Prefix 1000

kilogram (kg). 1000 grams.

kilohertz (kHz). 1000 hertz

latch. (1) A simple logic-circuit storage element. (2) A feedback loop in sequential digital circuits used to maintain a state.

least-significant digit. The rightmost digit. See also low-order position.

LED. Light-emitting diode.

light-emitting diode (LED). A semiconductor device that gives off visible or infrared light when activated.

load. In programming, to enter data into storage or working registers.

look-up table (LUT). (1) A technique for mapping one set of values into a larger set of values. (2) In computer graphics, a table that assigns a color value (red, green, blue intensities) to a color index.

low power Schottky TTL. A version (LS series) of TTL giving a good compromise between low power and high speed. See also transistor-transistor logic and Schottky TTL.

LUT. Look-up table.

m. (1) Prefix milli; 0.001. (2) Meter.

M. (1) Prefix mega; 1,000,000. (2) When referring to computer storage capacity, 1,048,576. (1,048,576 = 2 to the 20th power.)

mA. Milliampere; 0.001 ampere.

machine code. The machine language used for entering text and program instructions onto the recording medium or into storage and which is subsequently used for processing and printout.

machine language. (1) A language that is used directly by a machine. (2) Deprecated term for computer instruction code.

magnetic disk. (1) A flat circular plate with a magnetizable surface layer on which data can be stored by magnetic recording. (2) See also diskette.

mark. A symbol or symbols that indicate the beginning or the end of a field, of a word, of an item of data, or of a set of data such as a file, a record, or a block.

mask. (1) A pattern of characters that is used to control the retention or elimination of portions of another pattern of characters. (2) To use a pattern of characters to control the retention or elimination of portions of another pattern of characters.

masked. Synonym for disabled.

matrix. (1) A rectangular array of elements, arranged in rows and columns, that may be manipulated according to the rules of matrix algebra. (2) In computers, a logic network in the form of an array of input leads and output leads with logic elements connected at some of their intersections.

Mb. 1 048 576 bytes.

mega (M). Prefix 1,000,000.

megahertz (MHz). 1,000,000 hertz.

memory. Term for main storage.

meter (m). A unit of length (equivalent to 39.37 inches).

MFM. Modified frequency modulation.

MHz. Megahertz; 1,000,000 hertz.

micro (μ). Prefix 0.000,001.

microcode. (1) One or more microinstructions. (2) A code, representing the instructions of an instruction set, implemented in a part of storage that is not program-addressable.

microinstruction. (1) An instruction of microcode. (2) A basic or elementary machine instruction.

microprocessor. An integrated circuit that accepts coded instructions for execution; the instructions may be entered, integrated, or stored internally.

microsecond (μs). 0.000,001 second.

milli (m). Prefix 0.001.

milliampere (mA). 0.001 ampere.

millisecond (ms). 0.001 second.

mnemonic. A symbol chosen to assist the human memory; for example, an abbreviation such as "mpy" for "multiply."

mode. (1) A method of operation; for example, the binary mode, the interpretive mode, the alphanumeric mode. (2) The most frequent value in the statistical sense.

modem (modulator-demodulator). A device that converts serial (bit by bit) digital signals from a business machine (or data communication

equipment) to analog signals that are suitable for transmission in a telephone network. The inverse function is also performed by the modem on reception of analog signals.

modified frequency modulation (MFM). The process of varying the amplitude and frequency of the 'write' signal. MFM pertains to the number of bytes of storage that can be stored on the recording media. The number of bytes is twice the number contained in the same unit area of recording media at single density.

modulation. The process by which some characteristic of one wave (usually high frequency) is varied in accordance with another wave or signal (usually low frequency). This technique is used in modems to make business-machine signals compatible with communication facilities.

module. (1) A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading. (2) A packaged functional hardware unit designed for use with other components.

monitor. Synonym for cathode ray tube display (CRT display).

most-significant digit. The leftmost (non-zero) digit. See also high-order position.

ms. Millisecond; 0.001 second.

multiplexer. A device capable of interleaving the events of two or more activities, or capable of distributing the events of an interleaved sequence to the respective activities.

n. Prefix nano; 0.000 000 001.

NAND. A logic operator having the property that if P is a statement, Q is a statement, R is a statement,..., then the NAND of P, Q ,R,... is true if at least one statement is false, false if all statements are true.

NAND gate. A gate in which the output is 0 only if all inputs are 1.

nano (n). Prefix 0.000,000,001.

nanosecond (ns). 0.000,000,001 second.

non-return-to-zero (inverted) recording (NRZI). Deprecated term for non-return-to-zero change-on-ones recording.

NOR. A logic operator having the property that if P is a statement, Q is a statement, R is a statement,..., then the NOR of P, Q, R,... is true if all statements are false, false if at least one statement is true.

NOR gate. A gate in which the output is 0 only if at least one input is 1.

NOT. A logical operator having the property that if P is a statement, then the NOT of P is true if P is false, false if P is true.

ns. Nanosecond; 0.000 000 001 second.

NUL. The null character.

null character (NUL). A control character that is used to accomplish media-fill or time-fill, and that may be inserted into or removed from, a sequence of characters without affecting the meaning of the sequence; however, the control of the equipment or the format may be affected by this character.

odd-even check. Synonym for parity check.

offline. Pertaining to the operation of a functional unit without the continual control of a computer.

one-shot. A circuit that delivers one output pulse of desired duration for each input (trigger) pulse.

open circuit. (1) A discontinuous circuit; that is, one that is broken at one or more points and, consequently, cannot conduct current. Contrast with closed circuit. (2) Pertaining to a no-load condition; for example, the open-circuit voltage of a power supply.

open collector. A switching transistor without an internal connection between its collector and the voltage supply. A connection from the collector to the voltage supply is made through an external (pull-up) resistor.

operand. (1) An entity to which an operation is applied. (2) That which is operated upon. An operand is usually identified by an address part of an instruction.

operating system. Software that controls the execution of programs; an operating system may provide services such as resource allocation, scheduling, input/output control, and data management.

OR. A logic operator having the property that if P is a statement, Q is a statement, R is a statement,..., then the OR of P, Q, R,...is true if at least one statement is true, false if all statements are false.

OR gate. A gate in which the output is 1 only if at least one input is 1.

output. Pertaining to a device, process, or channel involved in an output process, or to the data or states involved in an output process.

output process. (1) The process that consists of the delivery of data from a data processing system, or from any part of it. (2) The return of information from a data processing system to an end user, including the translation of data from a machine language to a language that the end user can understand.

overcurrent. A current of higher than specified strength.

overrun. Loss of data because a receiving device is unable to accept data at the rate it is transmitted.

overvoltage. A voltage of higher than specified value.

parallel. (1) Pertaining to the concurrent or simultaneous operation of two or more devices, or to the concurrent performance of two or more activities.
(2) Pertaining to the concurrent or simultaneous occurrence of two or more related activities in multiple devices or channels. (3) Pertaining

to the simultaneity of two or more processes. (4) Pertaining to the simultaneous processing of the individual parts of a whole, such as the bits of a character and the characters of a word, using separate facilities for the various parts. (5) Contrast with serial.

parameter. (1) A variable that is given a constant value for a specified application and that may denote the application. (2) A name in a procedure that is used to refer to an argument passed to that procedure.

parity bit. A binary digit appended to a group of binary digits to make the sum of all the digits either always odd (odd parity) or always even (even parity).

parity check. (1) A redundancy check that uses a parity bit. (2) Synonymous with odd-even check.

PEL. Picture element.

personal computer. A small home or business computer that has a processor and keyboard and that can be connected to a television or some other monitor. An optional printer is usually available.

picture element (PEL). The smallest displayable unit on a display.

polling. (1) Interrogation of devices for purposes such as to avoid contention, to determine operational status, or to determine readiness to send or receive data. (2) The process whereby stations are invited, one at a time, to transmit.

port. An access point for data entry or exit.

POS. programmable option select.

power supply. A device that produces the power needed to operate electronic equipment.

printed circuit. A pattern of conductors (corresponding to the wiring of an electronic circuit) formed on a board of insulating material.

printed-circuit board. A usually copper-clad plastic board used to make a printed circuit.

priority. A rank assigned to a task that determines its precedence in receiving system resources.

processing unit. A functional unit that consists of one or more processors and all or part of internal storage.

processor. (1) In a computer, a functional unit that interprets and executes instructions. (2) A functional unit, a part of another unit such as a terminal or a processing unit, that interprets and executes instructions. (3) Deprecated term for processing program. (4) See microprocessor.

program. (1) A series of actions designed to achieve a certain result. (2) A series of instructions telling the computer how to handle a problem or task. (3) To design, write, and test computer programs.

programmable option select (POS). A method by which device options

are initialized in a device-resident register by the power-on self-test. The use of this method replaces the setting of switches or jumpers.

programming language. (1) An artificial language established for expressing computer programs. (2) A set of characters and rules with meanings assigned prior to their use, for writing computer programs.

programming system. One or more programming languages and the necessary software for using these languages with particular automatic data-processing equipment.

propagation delay. (1) The time necessary for a signal to travel from one point on a circuit to another. (2) The time delay between a signal change at an input and the corresponding change at an output.

protocol. (1) A specification for the format and relative timing of information exchanged between communicating parties. (2) The set of rules governing the operation of functional units of a communication system that must be followed if communication is to be achieved.

pulse. A variation in the value of a quantity, short in relation to the time schedule of interest, the final value being the same as the initial value.

RAM. Random access memory. Read/write memory.

random access memory (RAM). Read/write memory.

RAS. In the IBM Personal Computer, row address strobe.

raster. In computer graphics, a predetermined pattern of lines that provides uniform coverage of a display space.

read. To acquire or interpret data from a storage device, from a data medium, or from another source.

read-only memory (ROM). A storage device whose contents cannot be modified. The memory is retained when power is removed.

read/write memory. A storage device whose contents can be modified. Also called RAM.

recoverable error. An error condition that allows continued execution of a program.

redundancy check. A check that depends on extra characters attached to data for the detection of errors. See cyclic redundancy check.

register. (1) A storage device, having a specified storage capacity such as a bit, a byte, or a computer word, and usually intended for a special purpose. (2) A storage device in which specific data is stored.

RGBI. Red-green-blue-intensity.

rising edge. Synonym for positive-going edge.

ROM. Read-only memory.

ROM/BIOS. The ROM resident basic input/output system, which provides the level control of the

major I/O devices in the computer system.

row address strobe (RAS). A signal that latches the row address in a memory chip.

RS-232C. A standard by the EIA for communication between computers and external equipment.

RTS. Request to send. Associated with modem control.

run. A single continuous performance of a computer program or routine.

saturation. In computer graphics, the purity of a particular hue. A color is said to be saturated when at least one primary color (red, green, or blue) is completely absent.

Schottky TTL. A version (S series) of TTL with faster switching speed, but requiring more power. See also transistor-transistor logic and low power Schottky TTL.

SDLC. Synchronous Data Link Control.

sector. That part of a track or band on a magnetic drum, a magnetic disk, or a disk pack that can be accessed by the magnetic heads in the course of a predetermined rotational displacement of the particular device.

serial. (1) Pertaining to the sequential performance of two or more activities in a single device. In English, the modifiers serial and parallel usually refer to devices, as opposed to sequential and consecutive, which refer to processes. (2) Pertaining to the sequential or consecutive occurrence of two or more related activities in a single device or channel. (3) Pertaining to the sequential processing of the individual parts of a whole, such as the bits of a character or the characters of a word, using the same facilities for successive parts. (4) Contrast with parallel.

setup. (1) In a computer that consists of an assembly of individual computing units, the arrangement of interconnections between the units, and the adjustments needed for the computer to operate. (2) The preparation of a computing system to perform a job or job step. Setup is usually performed by an operator and often involves performing routine functions, such as mounting tape reels. (3) The preparation of the system for normal operation.

short circuit. A low-resistance path through which current flows, rather than through a component or circuit.

signal. A variation of a physical quantity, used to convey data.

sink. A device or circuit into which current drains.

slave. A device that recognizes its address on the channel and responds to control signals by executing the specified Read or Write command.

software. (1) Computer programs, procedures, and rules concerned with the operation of a data processing system. (2) Contrast with hardware.

source. The origin of a signal or electrical energy.

square wave. An alternating or pulsating current or voltage whose waveshape is square.

start bit. (1) A signal to a receiving mechanism to get ready to receive data or perform a function. (2) In a start-stop system, a signal preceding a character or block that prepares the receiving device for the reception of the code elements.

start-of-text (STX). A transmission control character that precedes a text and may be used to terminate the message heading.

stop bit. (1) A signal to a receiving mechanism to wait for the next signal. (2) In a start-stop system, a signal following a character or block that prepares the receiving device for the reception of a subsequent character or block.

storage. (1) A storage device. (2) A device, or part of a device, that can retain data. (3) The retention of data in a storage device. (4) The placement of data into a storage device.

strobe. An instrument that emits adjustable-rate flashes of light. Used to measure the speed of rotating or vibrating objects.

STX. Start-of-text.

symbol. (1) A conventional representation of a concept. (2) A representation of something by reason of relationship, association, or convention.

synchronization. The process of adjusting the corresponding significant instants of two signals to obtain the desired phase relationship between these instants.

synchronous transmission. (1) Data transmission in which the time of occurrence of each signal representing a bit is related to a fixed time frame. (2) Data transmission in which the sending and receiving devices are operating continuously at substantially the same frequency and are maintained, by means of correction, in a desired phase relationship.

syntax. (1) The relationship among characters or groups of characters, independent of their meanings or the manner of their interpretation and use. (2) The structure of expressions in a language. (3) The rules governing the structure of a language. (4) The relationships among symbols.

text. In ASCII and data communication, a sequence of characters treated as an entity if preceded and terminated by one STX and one ETX transmission control character, respectively.

time-out. (1) A parameter related to an enforced event designed to occur at the conclusion of a predetermined elapsed time. A time-out condition can be cancelled by the receipt of an appropriate time-out cancellation signal. (2) A time interval allotted for certain operations to occur; for example, response to polling or addressing before system operation is interrupted and must be restarted.

track. (1) The path or one of the set of paths, parallel to the reference edge on a data medium, associated with a single reading or writing component as the data medium moves past the component. (2) The portion of a moving data medium such as a drum, or disk, that is accessible to a given reading head position.

transistor-transistor logic (TTL). A popular logic circuit family that uses multiple-emitter transistors.

translate. To transform data from one language to another.

transmission. (1) The sending of data from one place for reception elsewhere. (2) In ASCII and data communication, a series of characters including headings and text. (3) The dispatching of a signal, message, or other form of intelligence by wire, radio, telephone, or other means. (4) One or more blocks or messages. For BSC and start-stop devices, a transmission is terminated by an EOT character. (5) Synonymous with data transmission.

TTL. Transistor-transistor logic.

V. Volt.

vector. In computer graphics, a directed line segment.

video. Computer data or graphics displayed on a cathode ray tube, monitor, or display.

virtual space. In computer graphics, a space in which the coordinates of the display elements are expressed in terms of user coordinates.

volt. The basic practical unit of electric pressure. The potential that causes electrons to flow through a circuit.

W. Watt.

watt. The practical unit of electric power.

window. (1) In computer graphics, a predefined part of the virtual space. (2) In computer graphics, the visible area of a viewplane mapped into a viewport.

word. (1) A character string or a bit string considered as an entity. (2) See computer word.

write. To make a permanent or transient recording of data in a storage device or on a data medium.

write precompensation. The varying of the timing of the head current from the outer tracks to the inner tracks of the diskette to keep a constant 'write' signal.

Notes:

Bibliography

- Microprocessor and Peripheral Handbook
  - INTEL Corporation.*210844.001*
- Introduction to the iAPX 286
  - INTEL Corporation.*210308.001*
- iAPX 286 Operating Systems Writer's Guide
  - INTEL Corporation.*121960.001*
- iAPX 286 Programmer's Reference Manual
  - INTEL Corporation.*210498.001*
- iAPX 286 Hardware Reference Manual
  - INTEL Corporation.*210760.001*
- Numeric Processor Extension Data Sheet
  - INTEL Corporation.*210920*
- 80287 Support Library Reference Manual
  - INTEL Corporation.*122129*
- 80386 Hardware Reference Manual
  - INTEL Corporation.*231732.001*
- Introduction to the 80386
  - INTEL Corporation.*231252.001*
- 80386 Programmer's Reference Manual
  - INTEL Corporation.*230985.001*
- 80386 System Software Writer's Guide
  - INTEL Corporation.*231499*
- National Semiconductor Corporation. *NS16450*
- Motorola Microprocessor's Data Manual
  - Motorola Inc. *Series B*

Notes:

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Numerics

Printed in the
United States of America

References in this
publication to IBM
products or services do not
imply that IBM intends
to make them available
outside the United States.

84X1508