Section 2. Micro Channel Architecture

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- or 16-bit I/O transfers within a 64K range. A memory address width of 24 bits allows 8- or 16-bit memory transfers within 16M (K = 1024; M = 1,048,576).
- 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 two types of channel connectors:

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

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. 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 2-90.

The following is a diagram of the channel connectors.

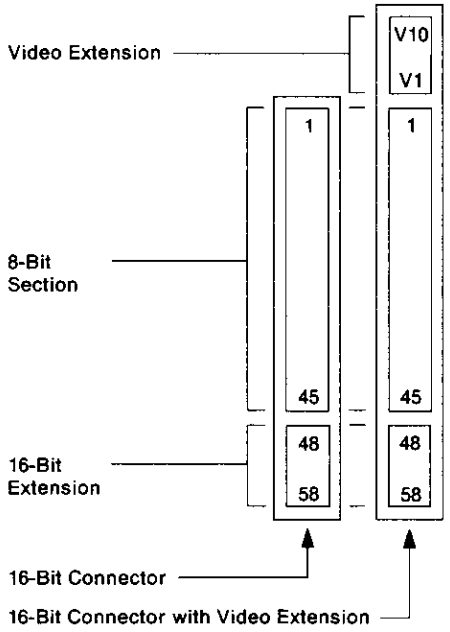


Figure 2-1. Micro Channel Connectors

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 driven with a tri-state driver.

-CD DS 16 (n): -Card Data Size 16: This line is driven by a 16-bit memory, I/O, or DMA slave to provide an indication on the channel of a 16-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: When active, this line indicates that if a memory cycle is in progress, the address is less than 16M. This signal must be included in all memory address decodes. MADE 24 is driven with a tri-state driver.

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 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 SFBDK 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 3.0 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 LSB and MSB 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 connector slots 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 cards. 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 (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, 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.

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

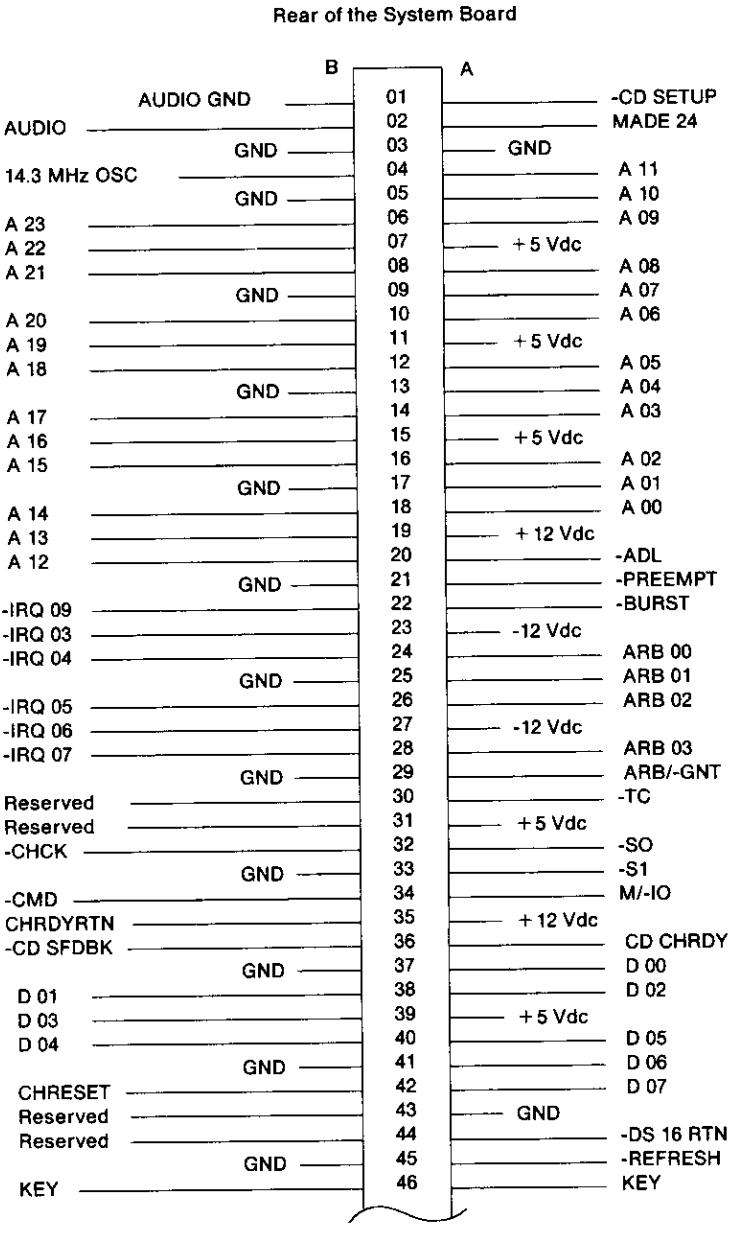


Figure 2-3. Channel Connector Voltage and Signal Assignments (8-Bit Section)

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

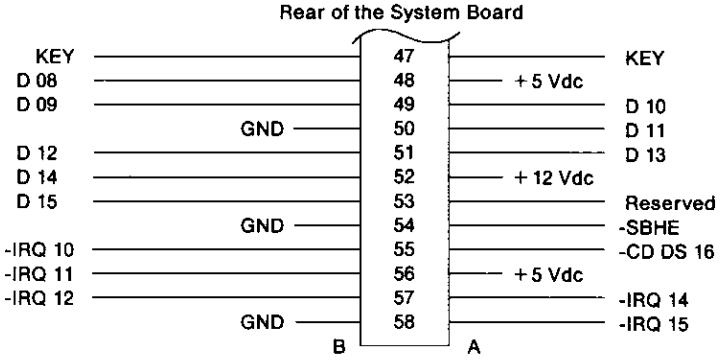


Figure 2-4. Micro Channel Connector 16-Bit Extension Voltage and Signal Assignments

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.

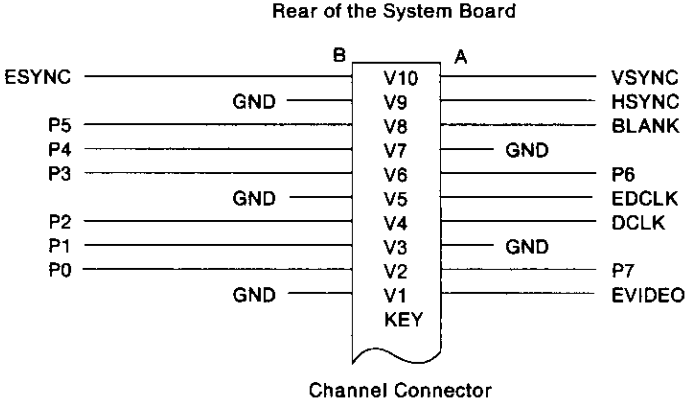


Figure 2-5. Auxiliary Video Connector

Channel Signal Groups (16-Bit)

The following figure lists the 16-bit 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)

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 2-6. Driver/Receiver Requirements and Options

The following notes apply to the driver and receiver options listed on page 2-18.

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 2-7 on page 2-20 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 2-7. 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 uA.

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 2-34) 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 2-8. 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.

Warning: The Model 50 and Model 60 system boards do not support 16-bit POS I/O operations to 8-bit POS registers. Using 16-bit I/O instructions on 8-bit POS registers will cause erroneous data to be written to or read from the registers. Only 8-bit transfers are supported for setup operations.

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 2-9. 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 2-10. 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.

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 2-11. 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.

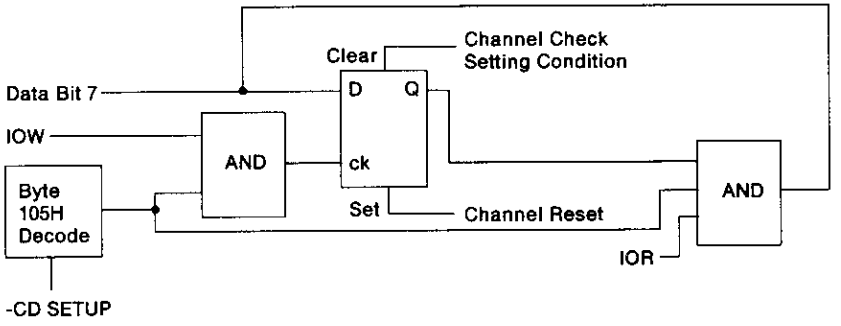


Figure 2-12. Channel Check Active Indicator

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.

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

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 2-13. 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 0C0000 to 0DFFFF.

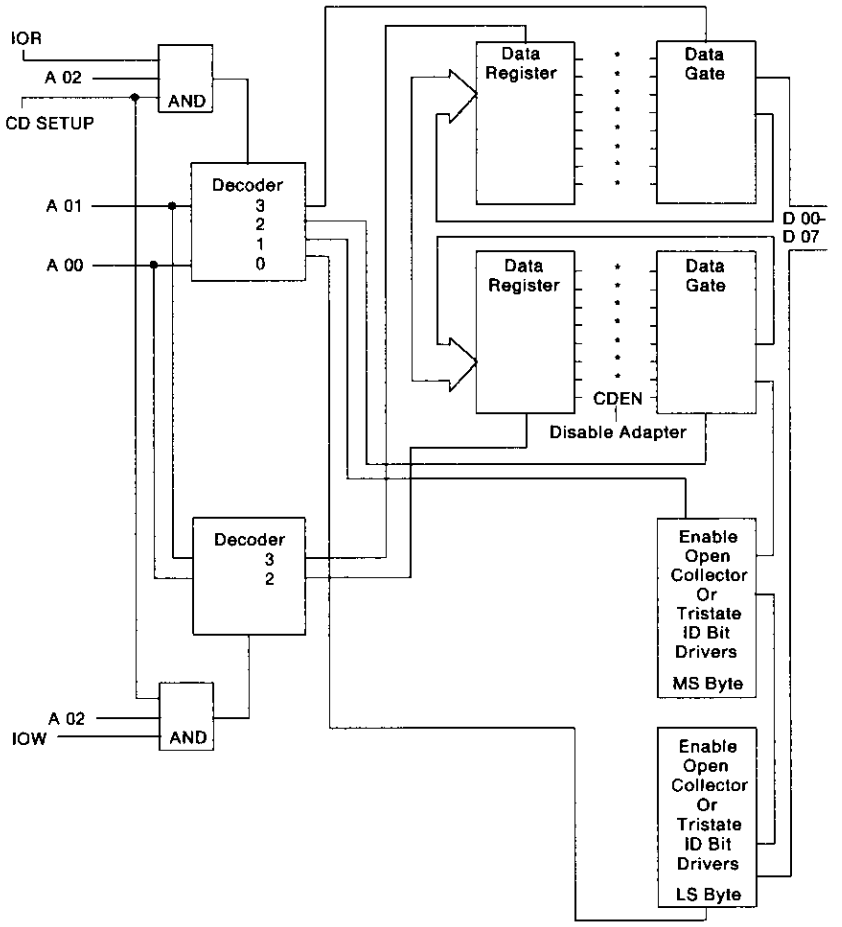


Figure 2-14. Typical Adapter Implementation of POS

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.

Subaddressing is used to load an address translator RAM on the IBM Personal System/2 80286 Memory Expansion Option during the POST. This address translator is used in memory relocation and is supported only by the POST.

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.

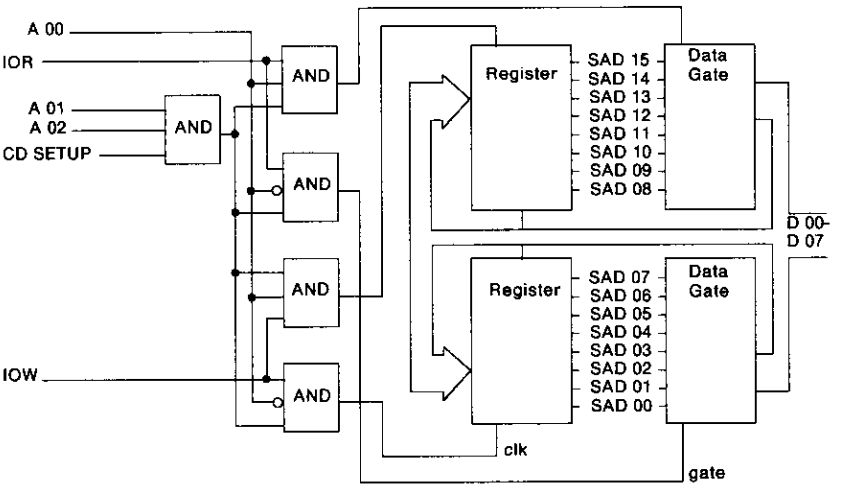


Figure 2-15. Subaddressing POS Extension Example

During the POST, RAM is loaded with a translate table that relocates all defects to the top of memory using the POS subaddress function.

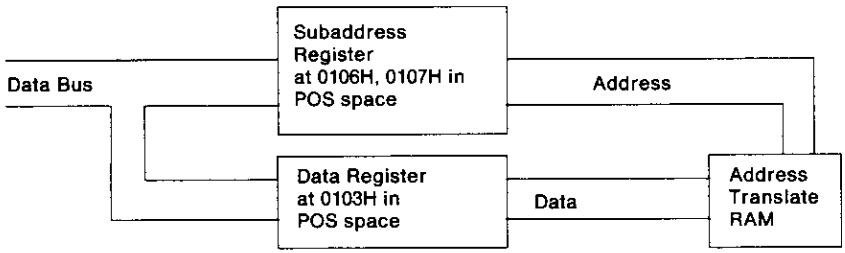


Figure 2-16. Relocation of Defects

After the POST, the RAM translates the system address bus to good blocks of storage. Memory fragmented by defects can be recovered by the Automatic Configuration utility. For more information on the Automatic Configuration utility see "System Configuration Utilities" on page 2-34.

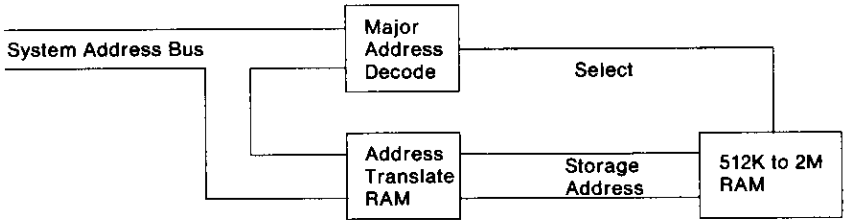


Figure 2-17. System Address Translation

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, 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 2-18. 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]=XXXXO1XXb int 3`

Figure 2-19. 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 2-20. 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 2-21. 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 2-22. 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 2-23. 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 2-45.

```
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.

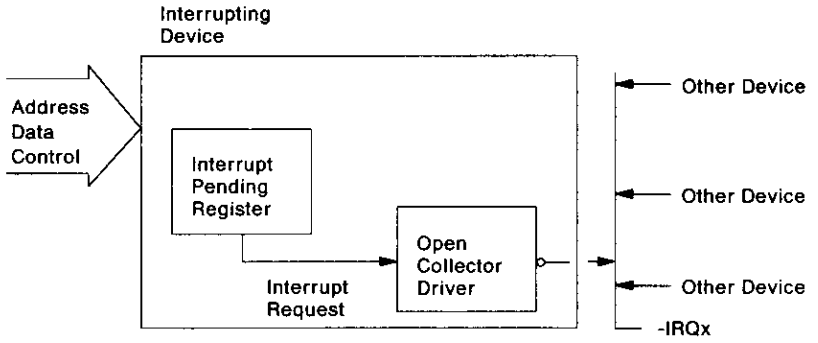


Figure 2-24. Typical Adapter Interrupt Sharing Implementation

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 2-25. 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:

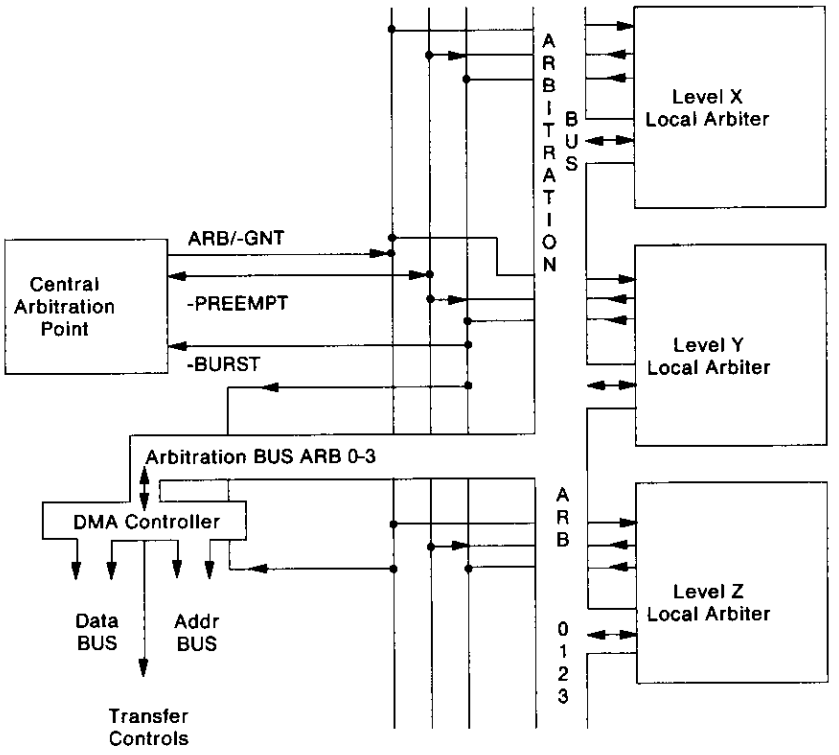


Figure 2-26. Central Arbitration

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.

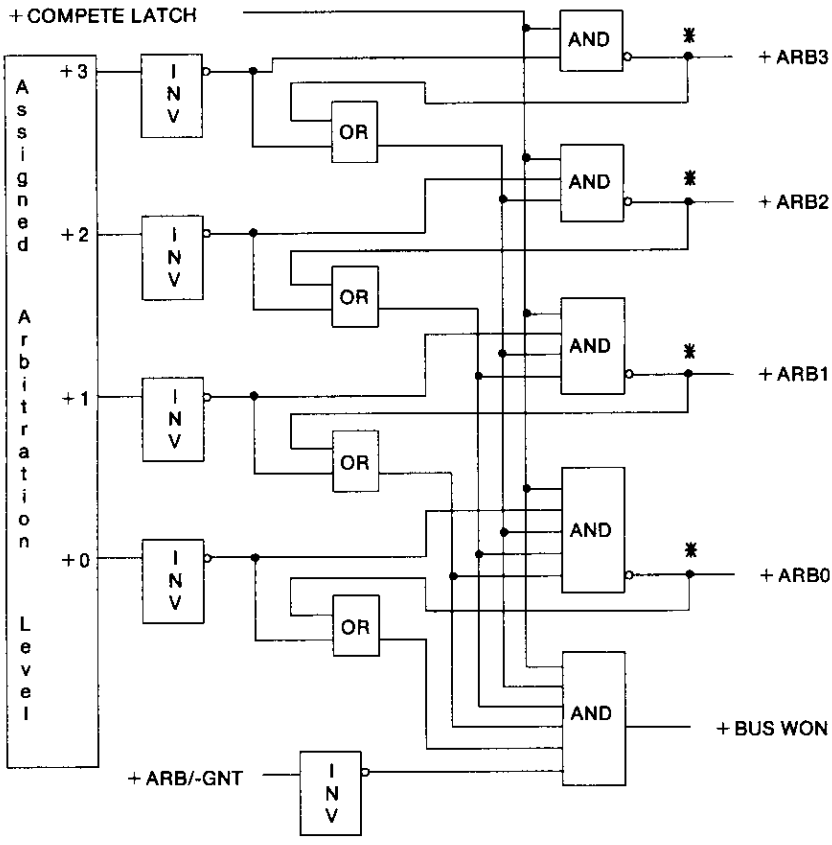


Figure 2-27. Local Arbiter Example

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:

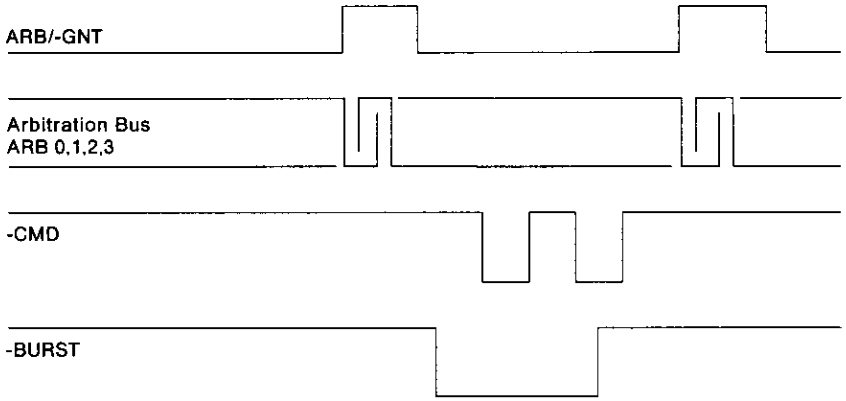


Figure 2-28. Burst Mode Timing

Preemption

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

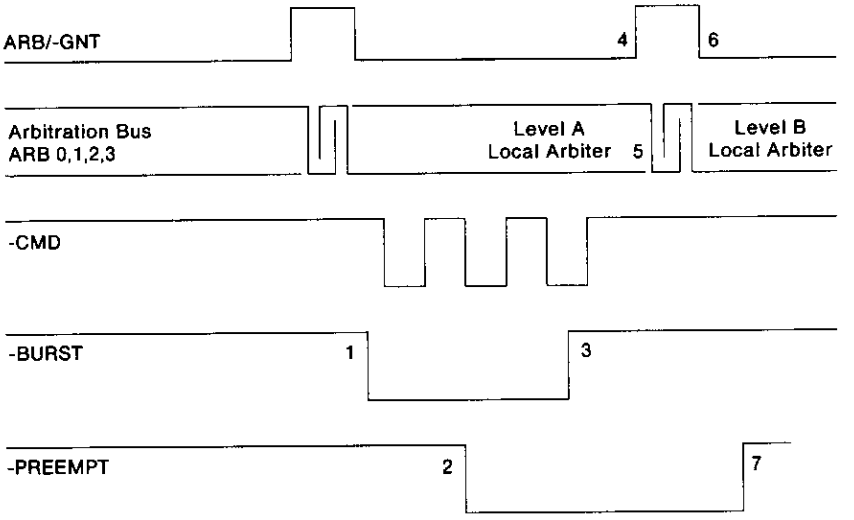


Figure 2-29. Preempt Timing

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 2-30. 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 2-31. 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 2-32. 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 600 nanoseconds, as opposed to 300 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.

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. 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).
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.
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:

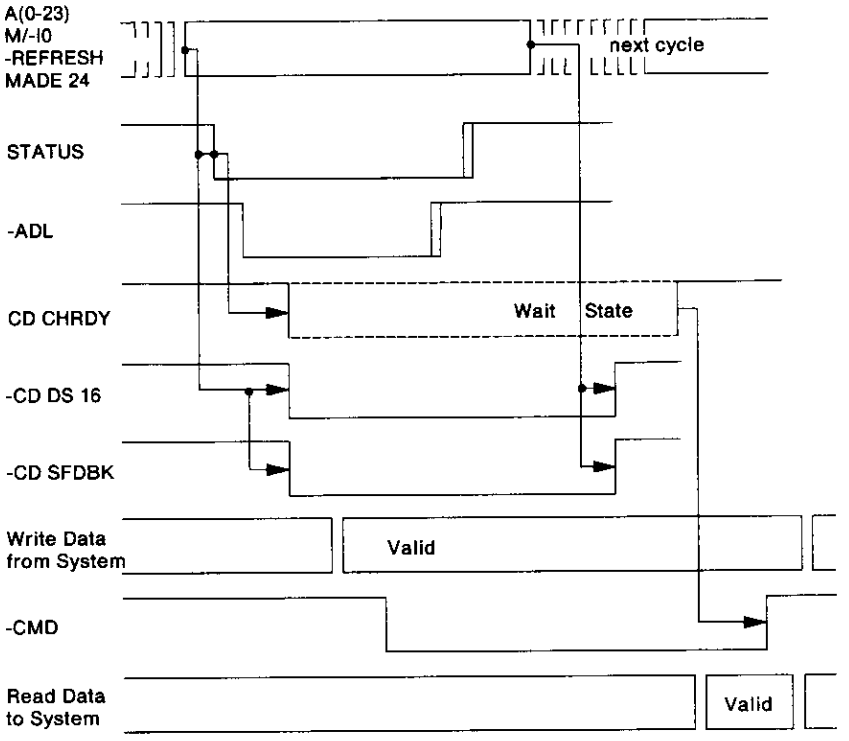


Figure 2-33. Overview of the Basic Transfer Cycle

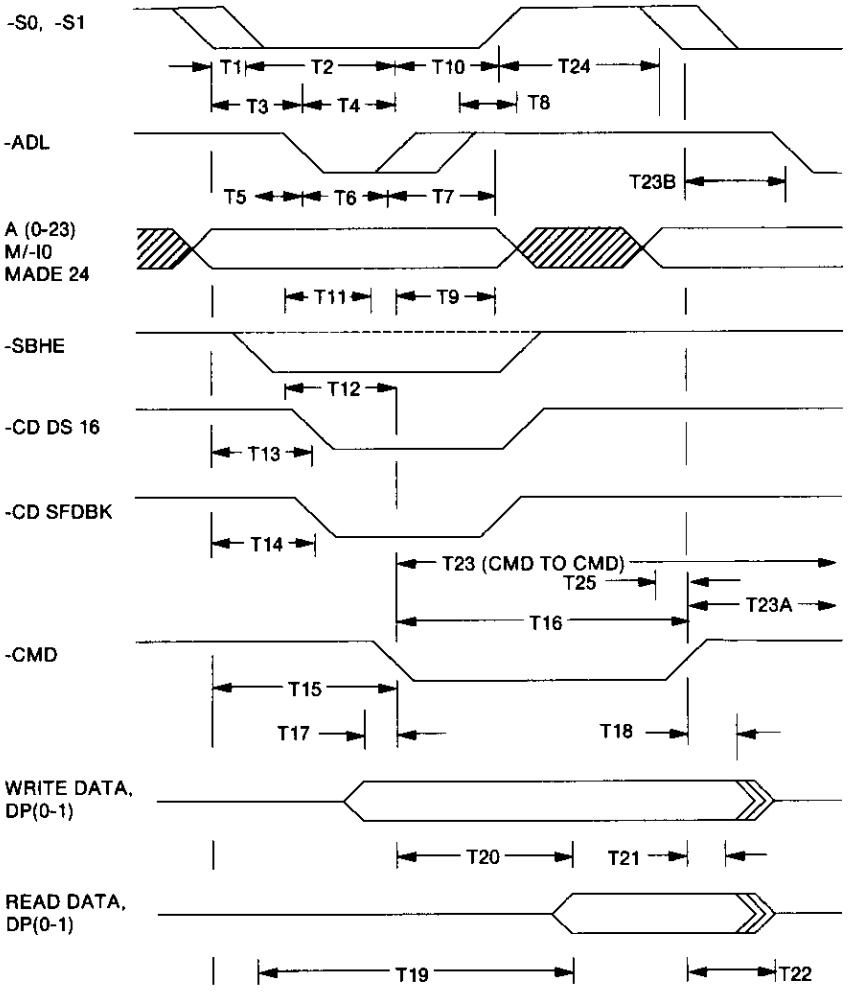
I/O and Memory Cycle

The I/O and memory cycle timing diagrams are shown on the following pages. They 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 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	

Figure 2-34. I/O and Memory Default Cycle (200 nanoseconds minimum)

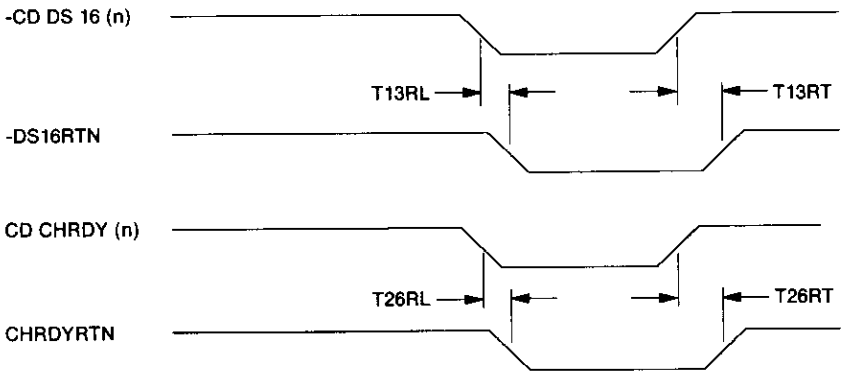
Notes:

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 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.

5. Model 50 and Model 60 automatically extend all default cycles to synchronous extended cycles. Adapter designs should support the 200 nanosecond default cycle to assist portability to other systems or drive CD CHRDY regardless of the system synchronous extension cycle.

Default Cycle Return Signals



Timing Parameter		Min/Max	Note
T13RL	-CD DS 16 (n) low to -DS 16 RTN low	- / 20 ns	1
T13RT	-CD DS 16 (n) high to -DS 16 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 2-35. 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. CHRDYRTN becomes active 40 nanoseconds maximum after -ADL becomes active.
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.

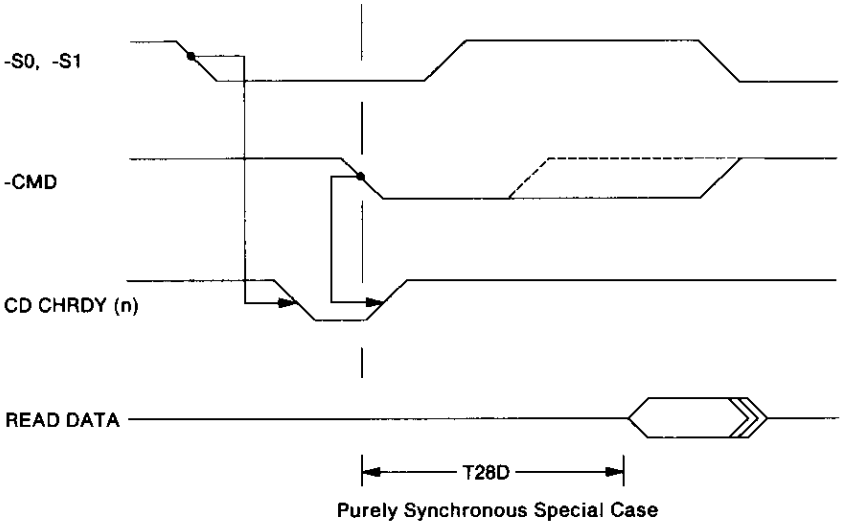
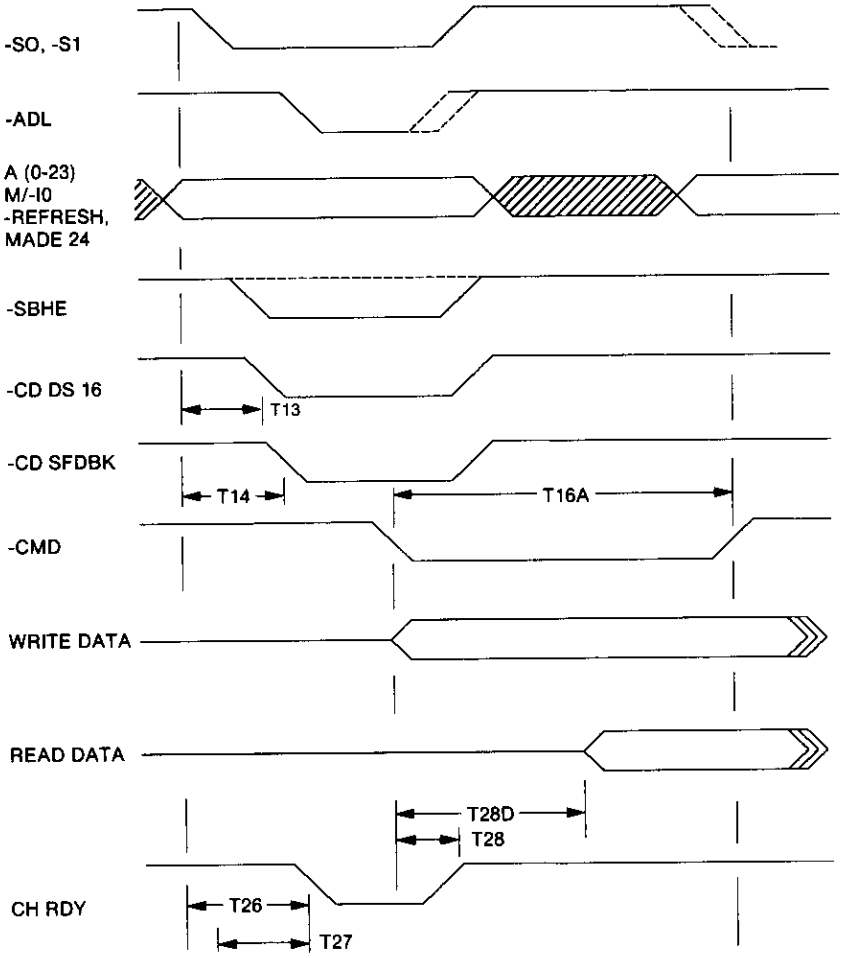


Figure 2-36. Timing Sequence for the Synchronous Special Case of Extended Cycle

Synchronous Extended Cycle (300 nanoseconds minimum - Special Case)



	Timing Parameter	Min/Max	Note
T13	-CD DS 16 (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 2-37. 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.

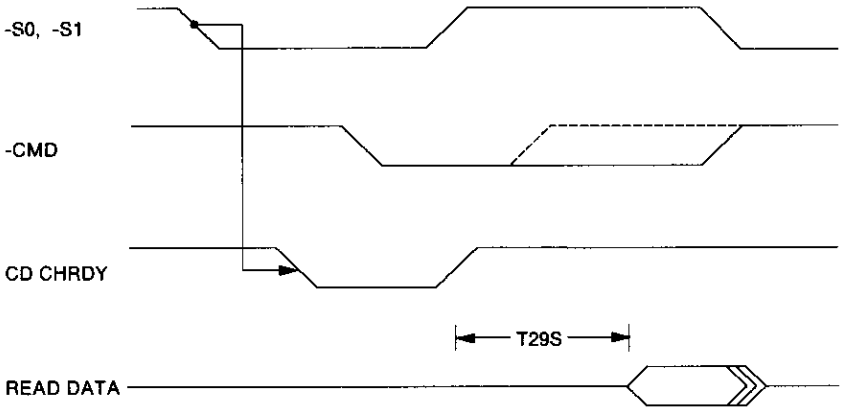
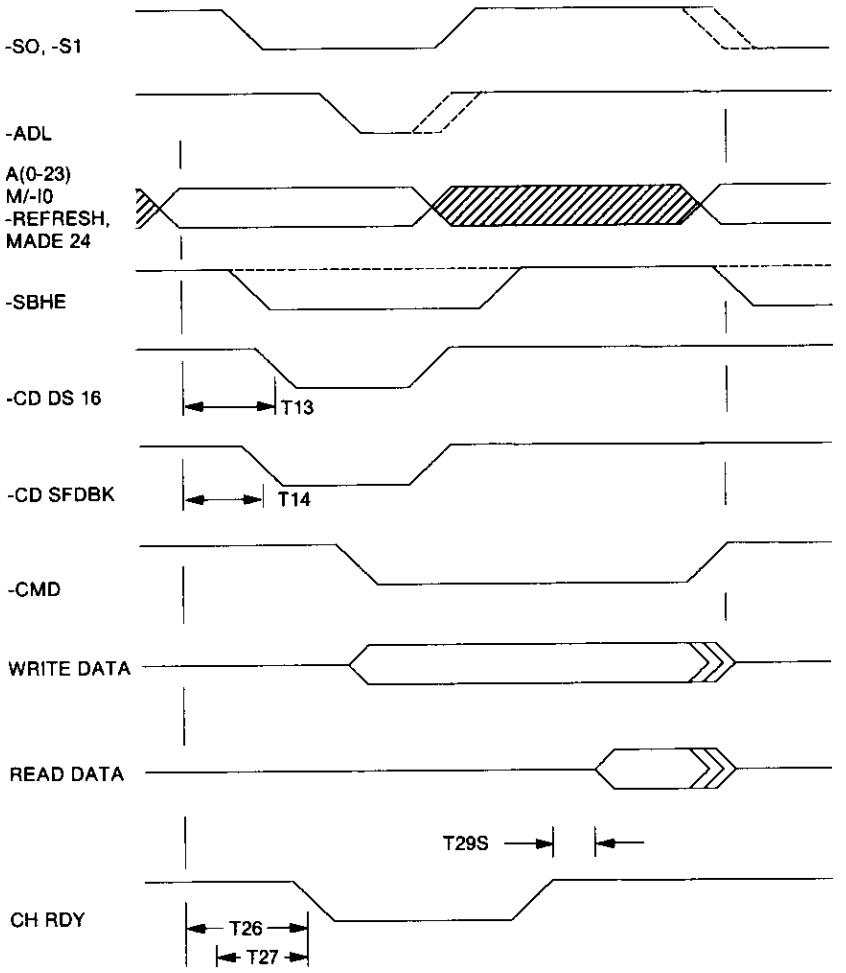


Figure 2-38. Timing Sequence for the Asynchronous Extended Cycle (General Case)

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



	Timing Parameter	Min/Max	Note
T13	-CD DS 16 (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.

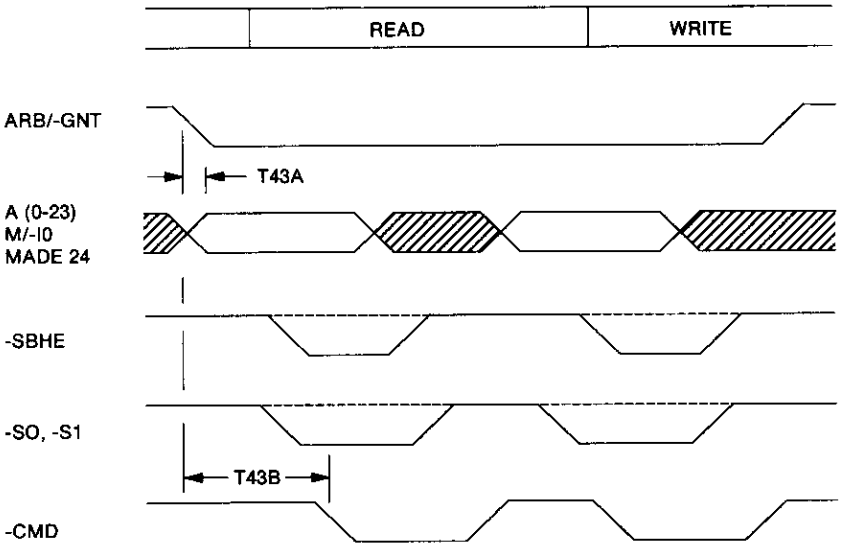
Figure 2-39. Asynchronous Extended Cycle (≥300 nanoseconds minimum - General Case)

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

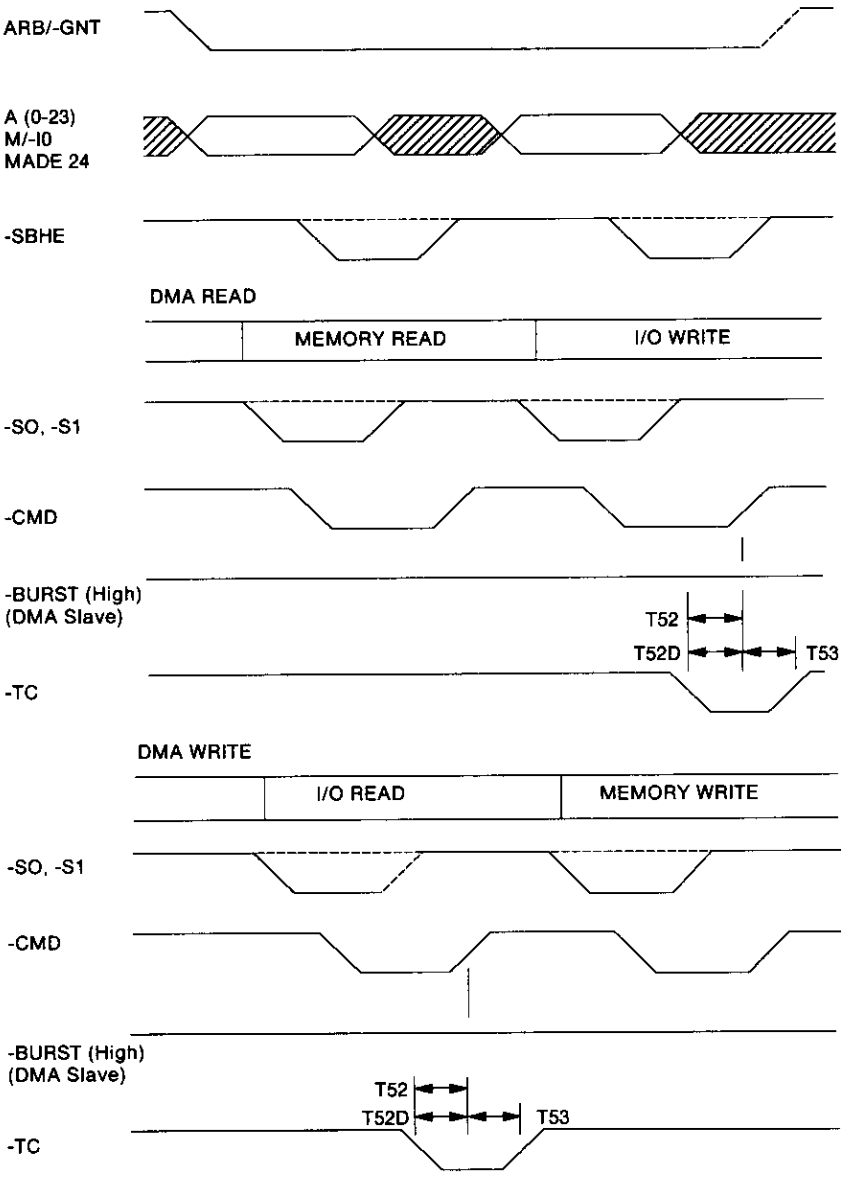


	Timing Parameter	Min/Max	Note
T43A	ADDR valid from ARB/-GNT low	0 / - ns	
T43B	-CMD active from ARB/-GNT low	115 / - ns	

Figure 2-40. 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, -CD SFDBK and CD CHRDY, if it places an address on the bus within 30 nanoseconds after the grant. However, if the controller places the address on the bus 30 nanoseconds after the grant, this 30 nanoseconds allowance is not needed.

Single DMA Transfer (DMA Controller Controlled)



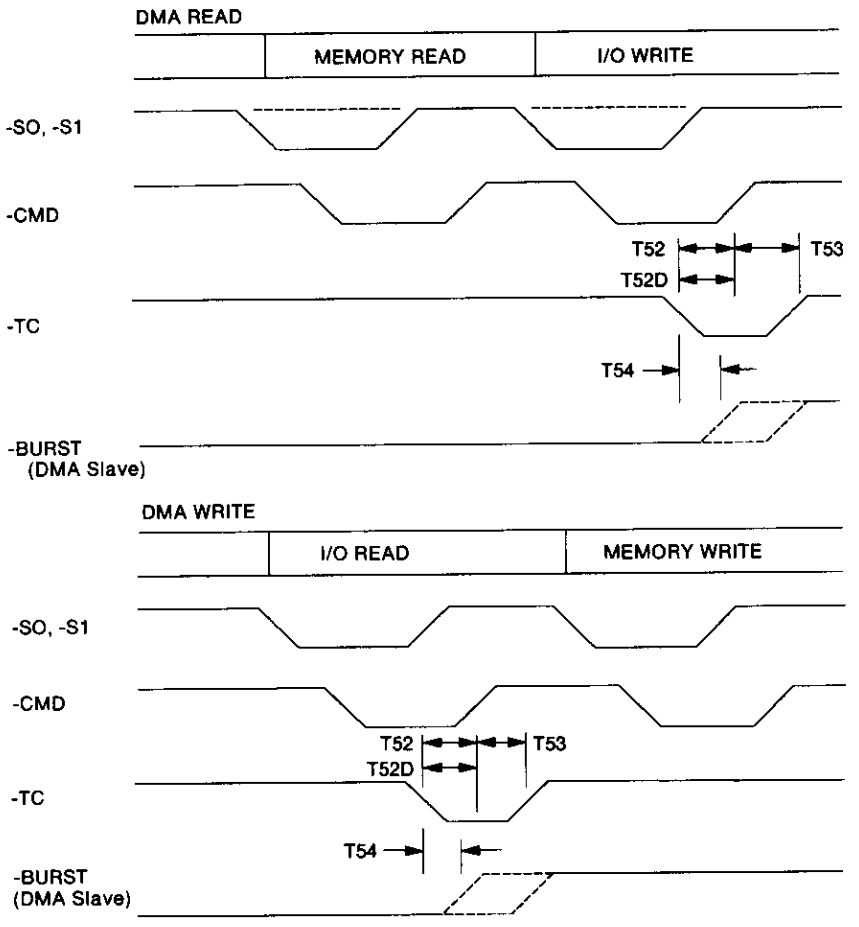
	Timing Parameter	Min/Max	Note
T52	-TC setup to -CMD inactive	30 / - ns	
T52D	-TC setup to -CMD inactive	15 / - ns	2
T53	-TC hold to -CMD inactive	10 / - ns	

Figure 2-41. 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)



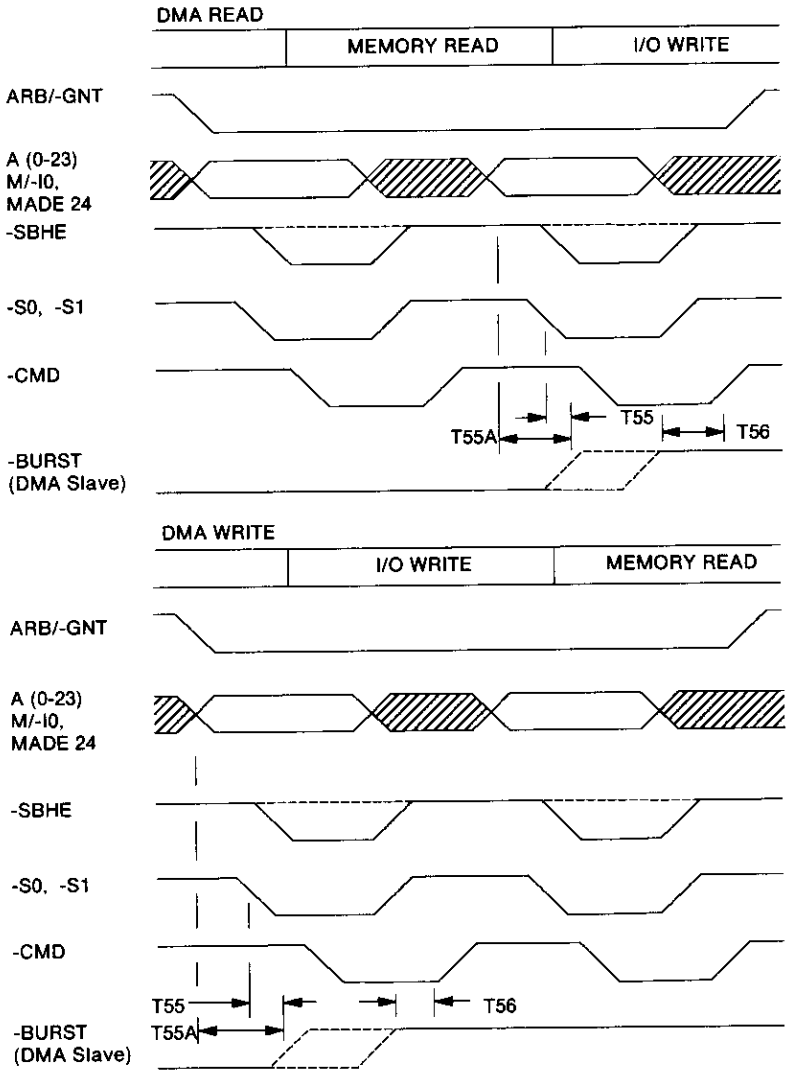
	Timing Parameter	Min/Max	Note
T52	-TC setup to -CMD inactive	30 / - ns	
T52D	-TC setup to -CMD inactive	15 / - ns	2
T53	-TC hold from -CMD inactive	10 / - ns	
T54	-BURST released by the DMA slave from -TC active	- / 30 ns	

Figure 2-42. 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)



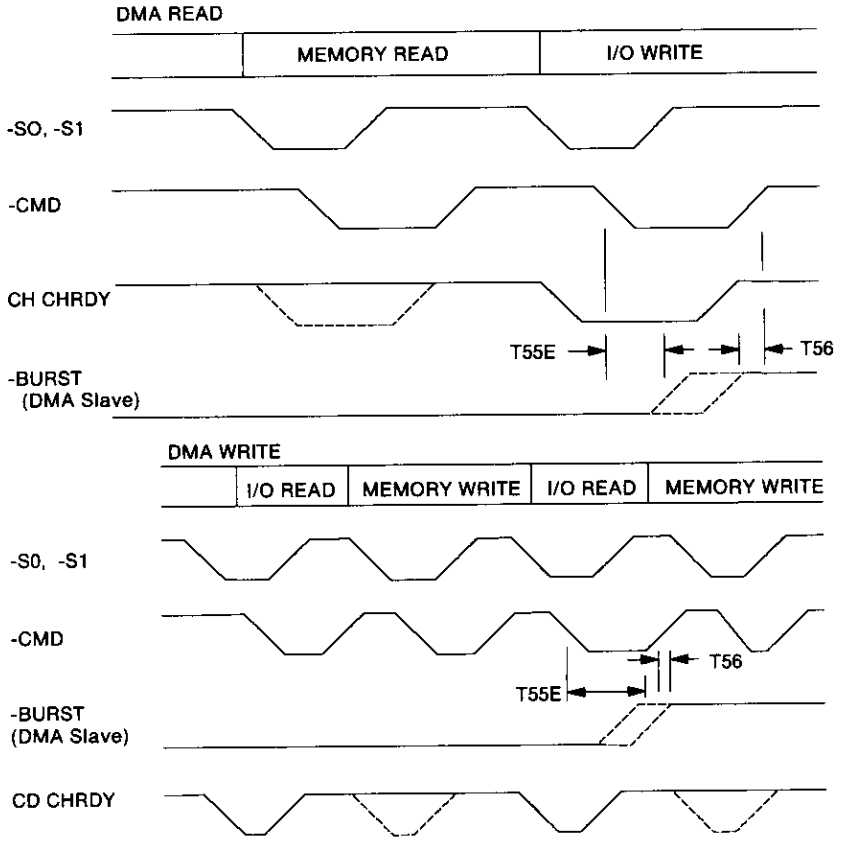
	Timing Parameter	Min/Max	Note
T55	-BURST released by the DMA slave from the last I/O Status active (Default cycle only)	- / 40 ns	2
T55A	-BURST released by the DMA slave from the last I/O ADDRESS valid (Default cycle only)	- / 70 ns	2
T56	-BURST inactive (high) setup to -CMD inactive	35 / - ns	3

Figure 2-43. 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 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)



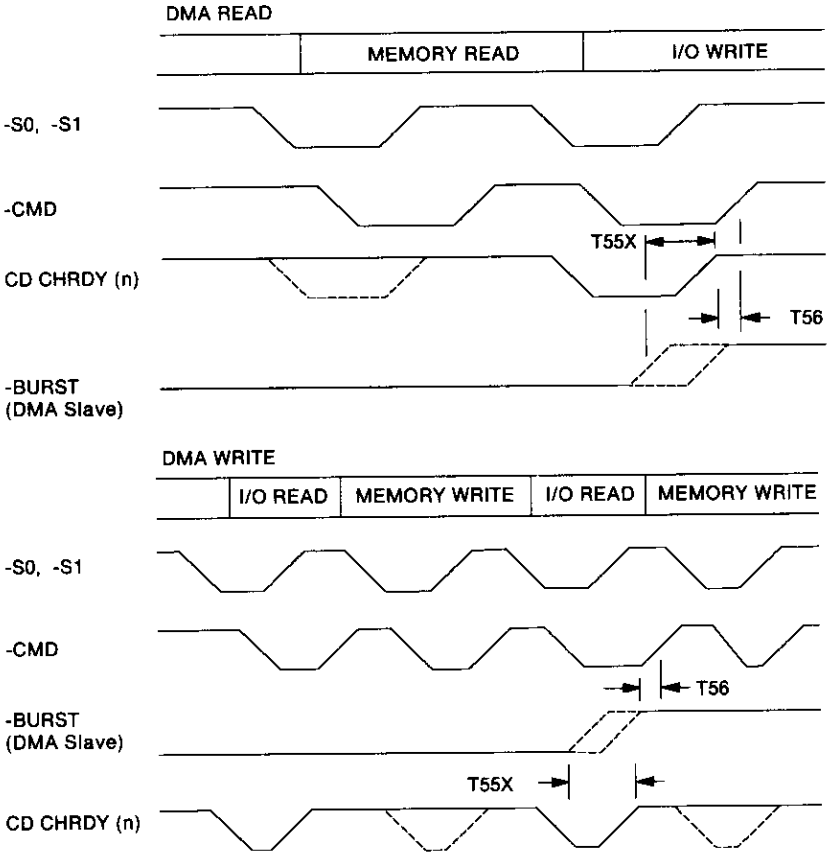
	Timing Parameter	Min/Max	Note
T55E	-BURST released by the DMA slave from the last -CMD active (Extended cycles only)	- / 80 ns	
T56	-BURST inactive (high) setup to -CMD inactive	35 / - ns	2

Figure 2-44. 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 2-45. Burst DMA Transfer (DMA Slave Terminated - Asynchronous Extended Cycle ≥300 nanoseconds)

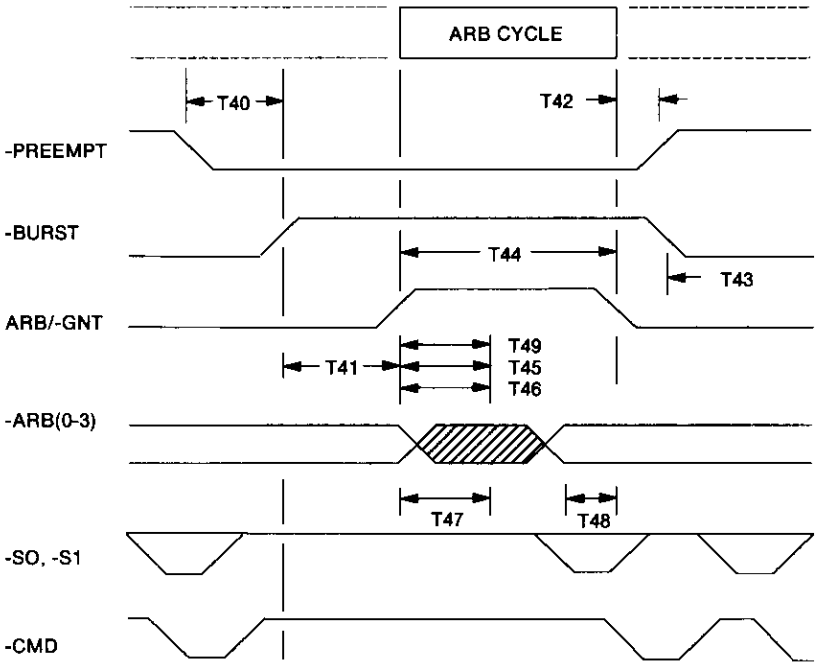
Notes:

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 the prevent 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 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.

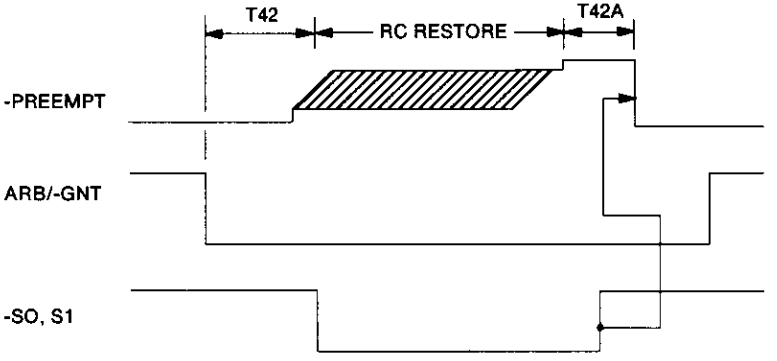
Arbitration Timing

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 2-46. Arbitration Cycle

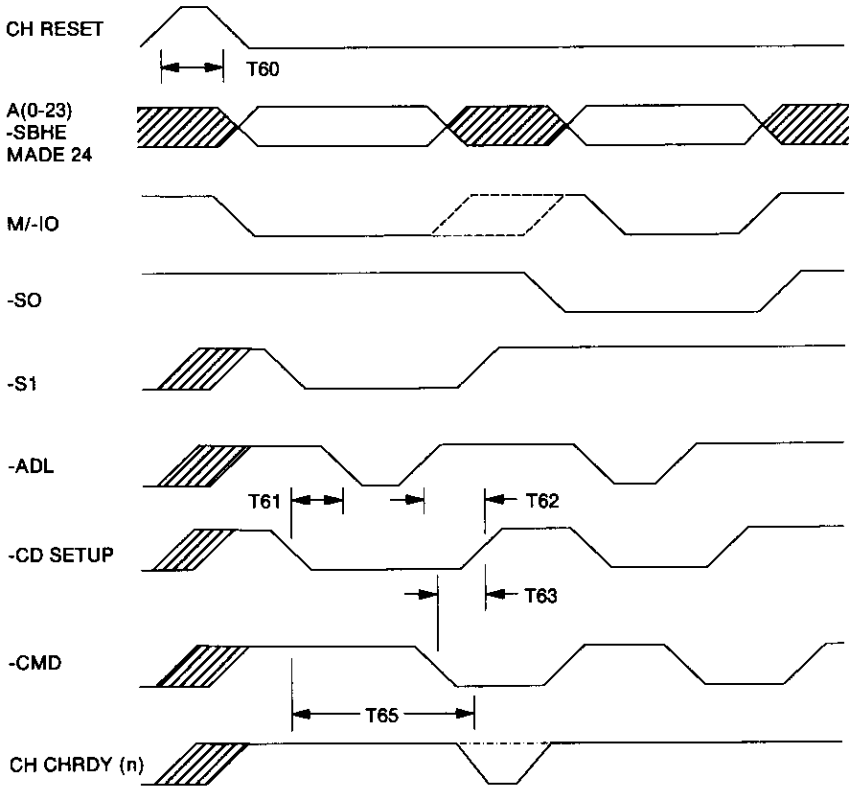
Notes:

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 resistor capacitor 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



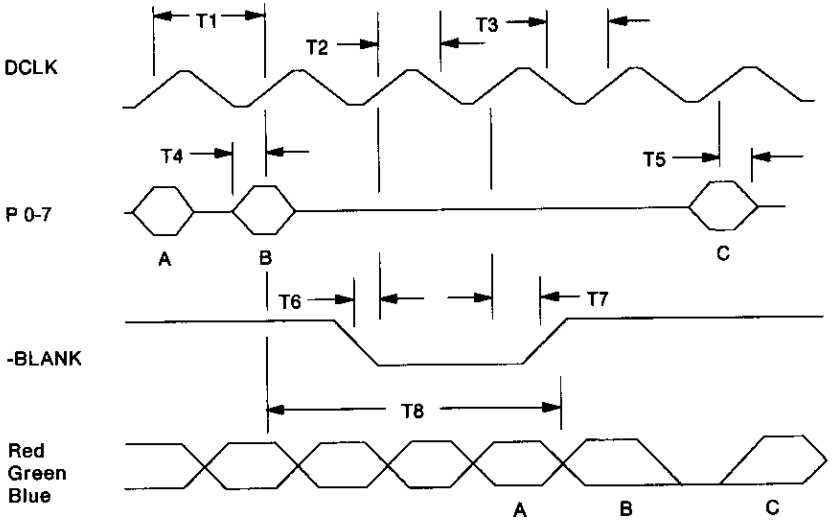
	Timing Parameter	Min/Max	Note
T60	CHRESET active (high) pulse width	100 / - ms	
T61	-CD SETUP (n) active (low) to -ADL active (low)	15 / - ns	
T62	-CD SETUP (n) hold from -ADL inactive (high)	25 / - ns	
T63	-CD SETUP (n) hold from -CMD active (low)	30 / - ns	
T65	CD CHRDY (n) inactive (low) from -CD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

Figure 2-47. 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 2-48. Auxiliary Video Connector Timing (DAC Signals)

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

Adapter Design

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

The system board provides two types of connectors for the channel:

- 8- or 16-bit connector
- 8- or 16-bit connector with video extension.

Plated connector contacts are not required for signals not used by an adapter. One position on the channel has the video extension connector. The video extension signal connector is used for display-only applications. See "Channel Definition" on page 2-4 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.

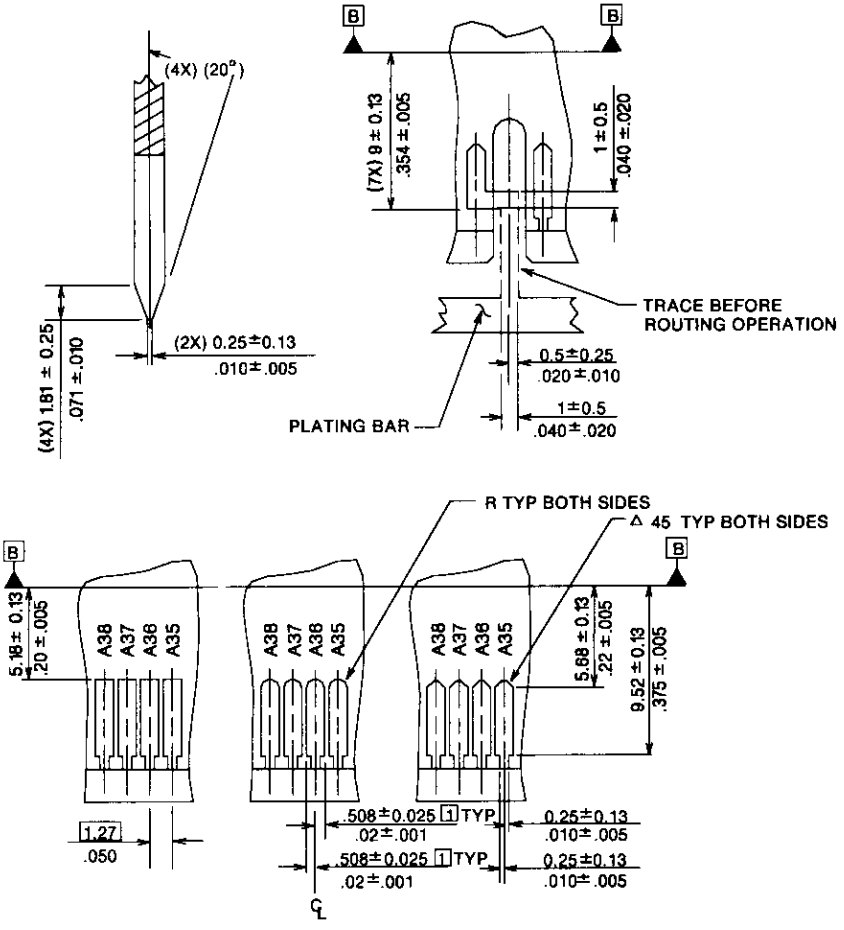


Figure 2-49. Connector (Common Detail)

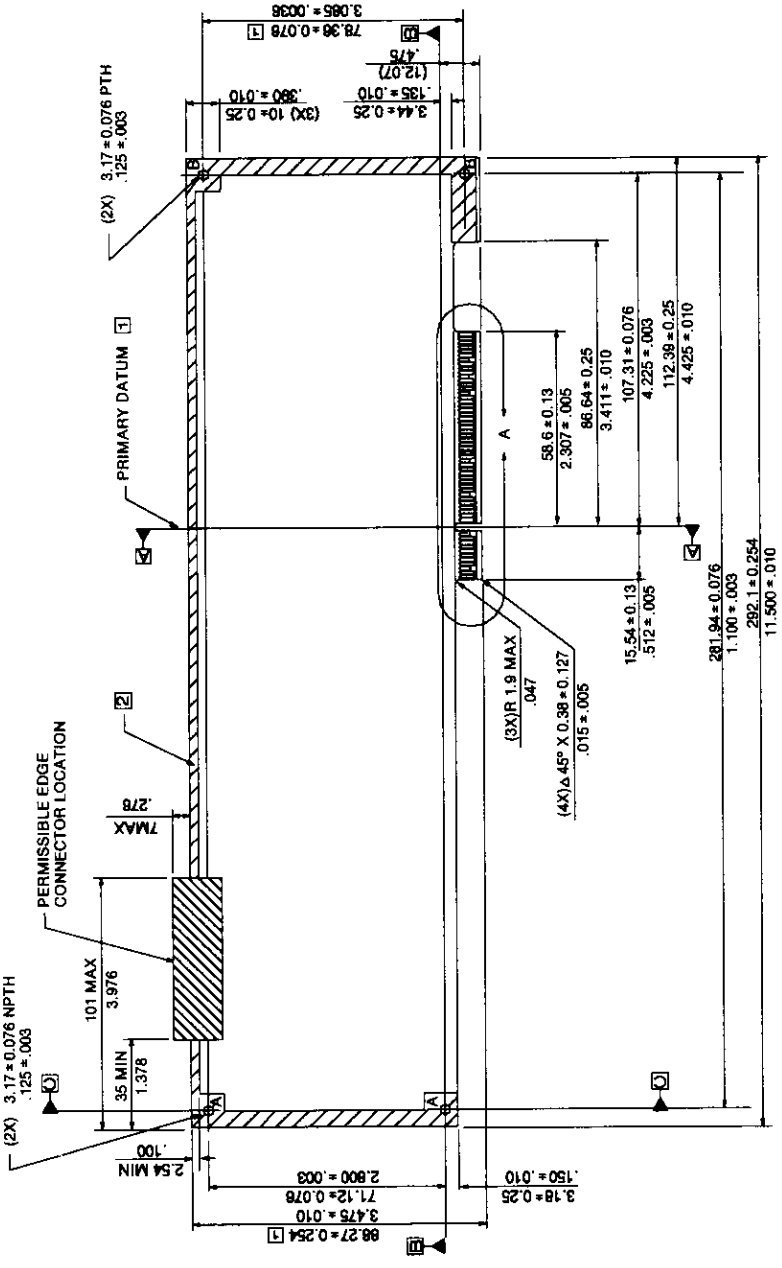


Figure 2-50. Adapter Dimensions (8- or 16-Bit)

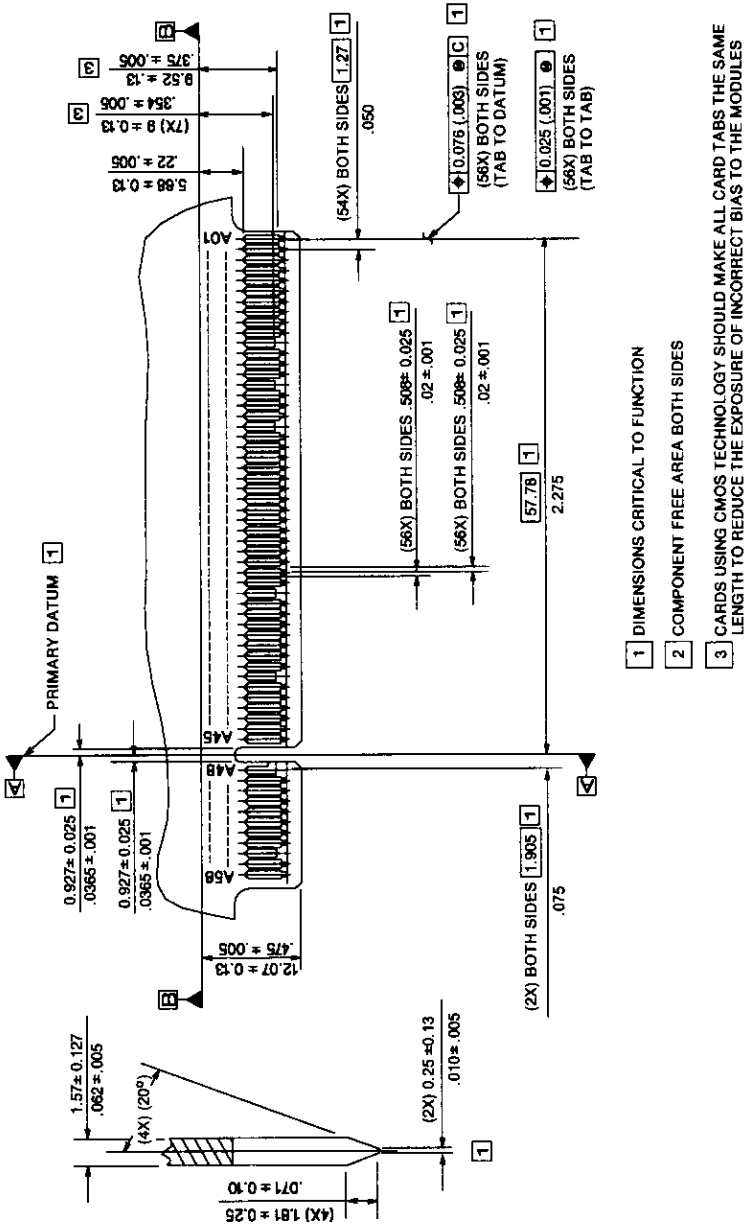


Figure 2-51. Connector Dimensions (8- or 16-Bit)

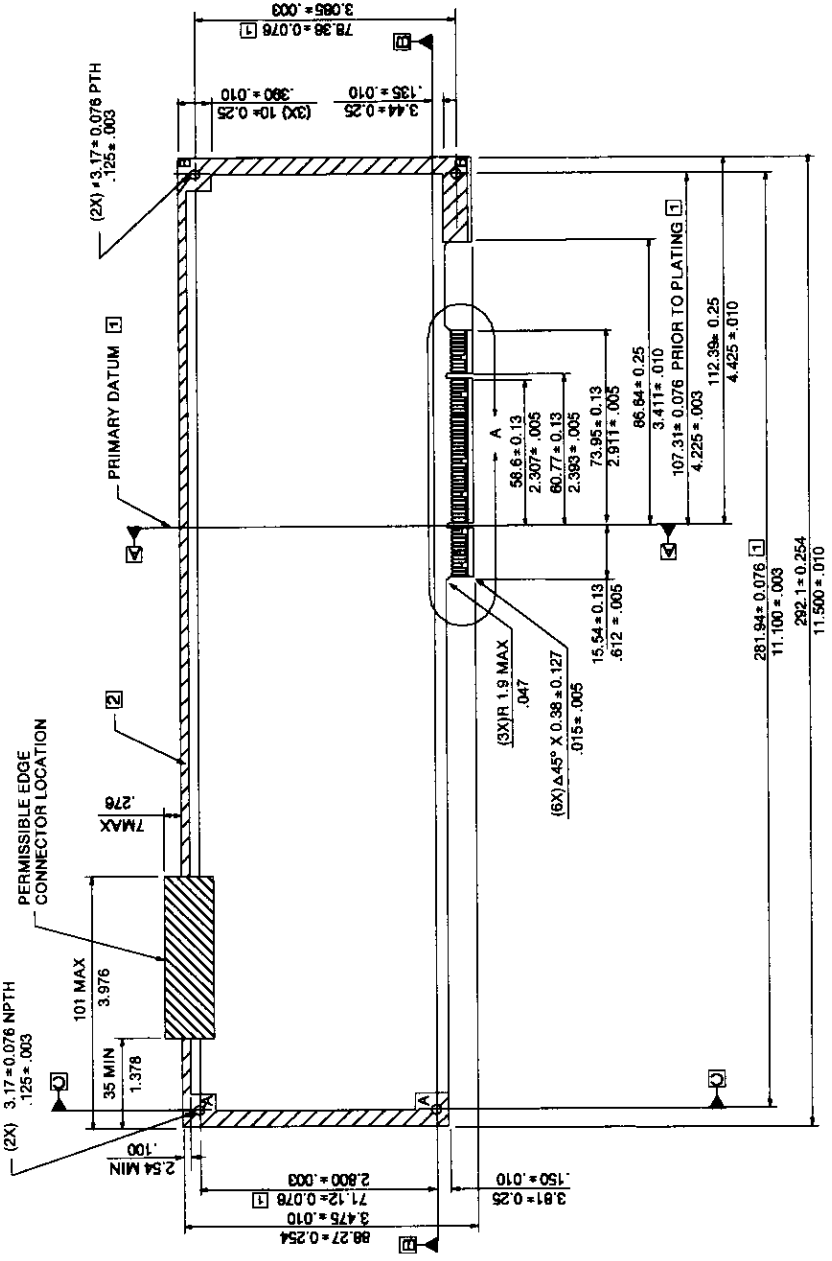


Figure 2-52. Adapter Dimensions (8- or 16-Bit with Video Extension)

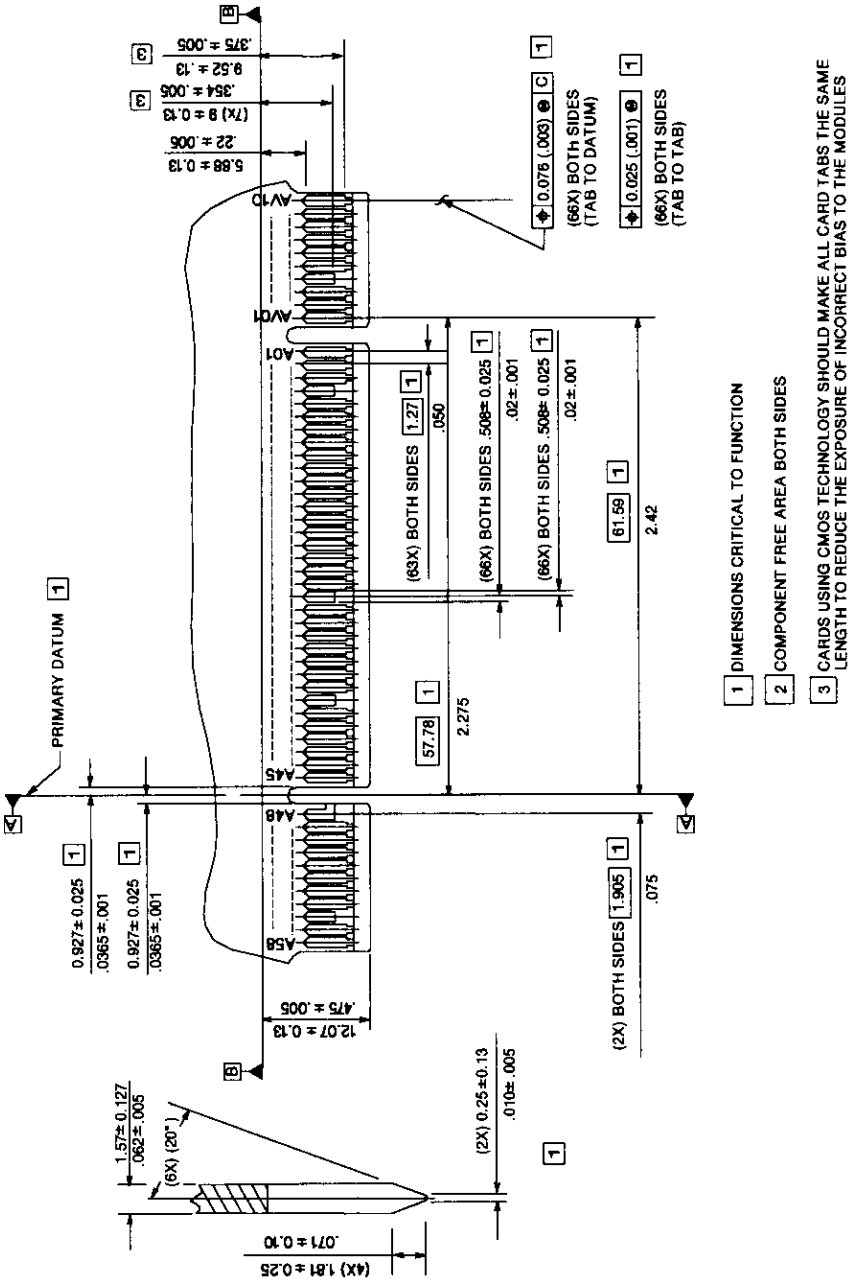
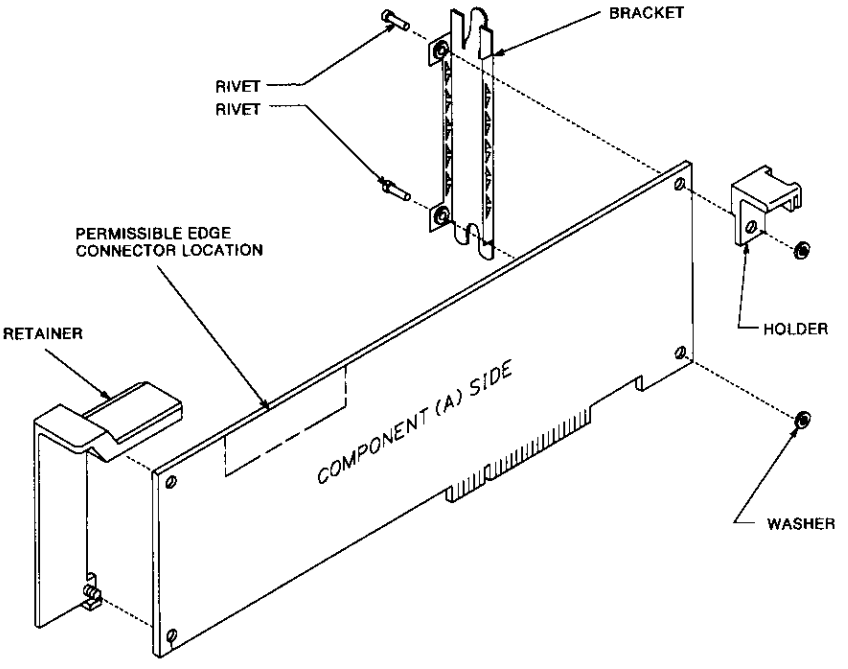


Figure 2-53. Connector Dimensions (8- or 16-Bit with Video Extension)



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

Figure 2-54. Typical Adapter Assembly

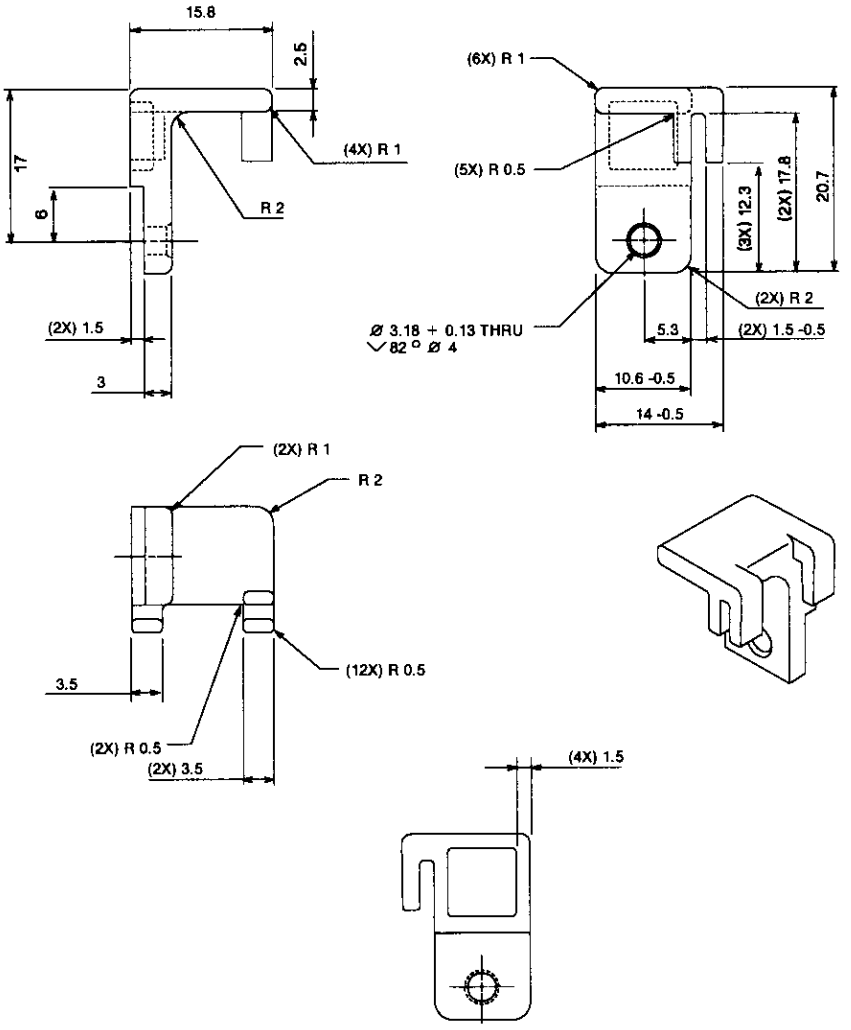


Figure 2-55. Adapter Holder

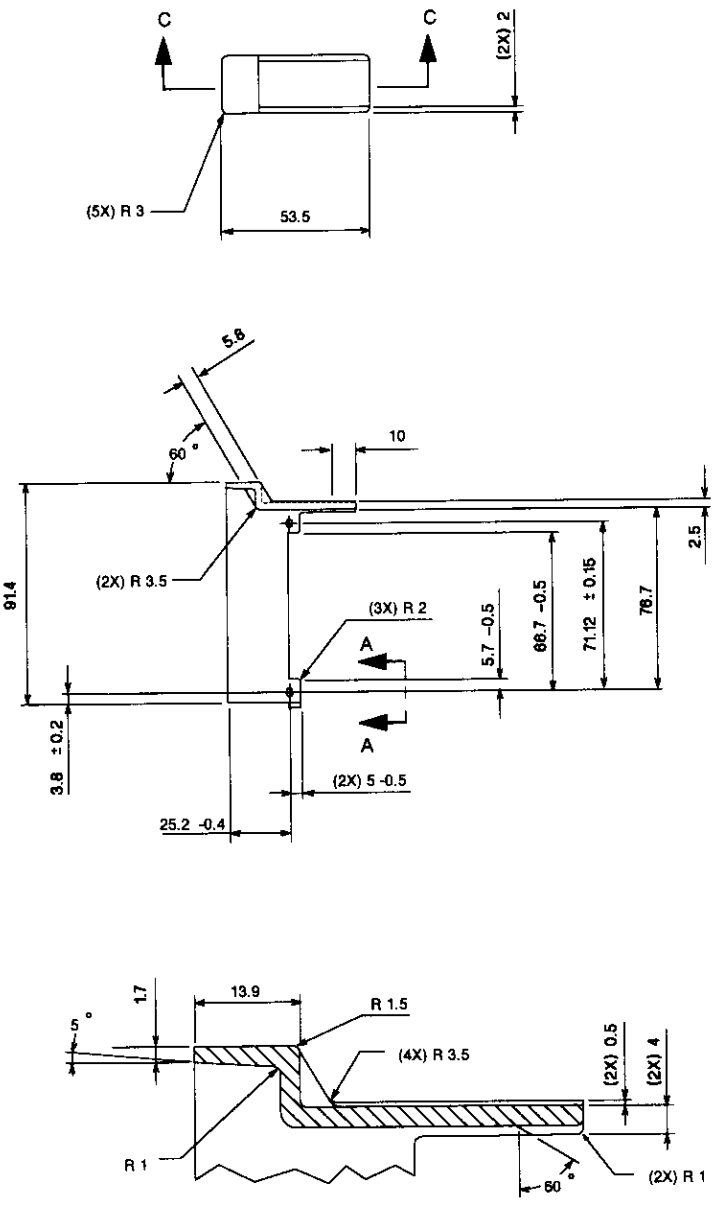


Figure 2-56. Adapter Retainer (Part 1 of 2)

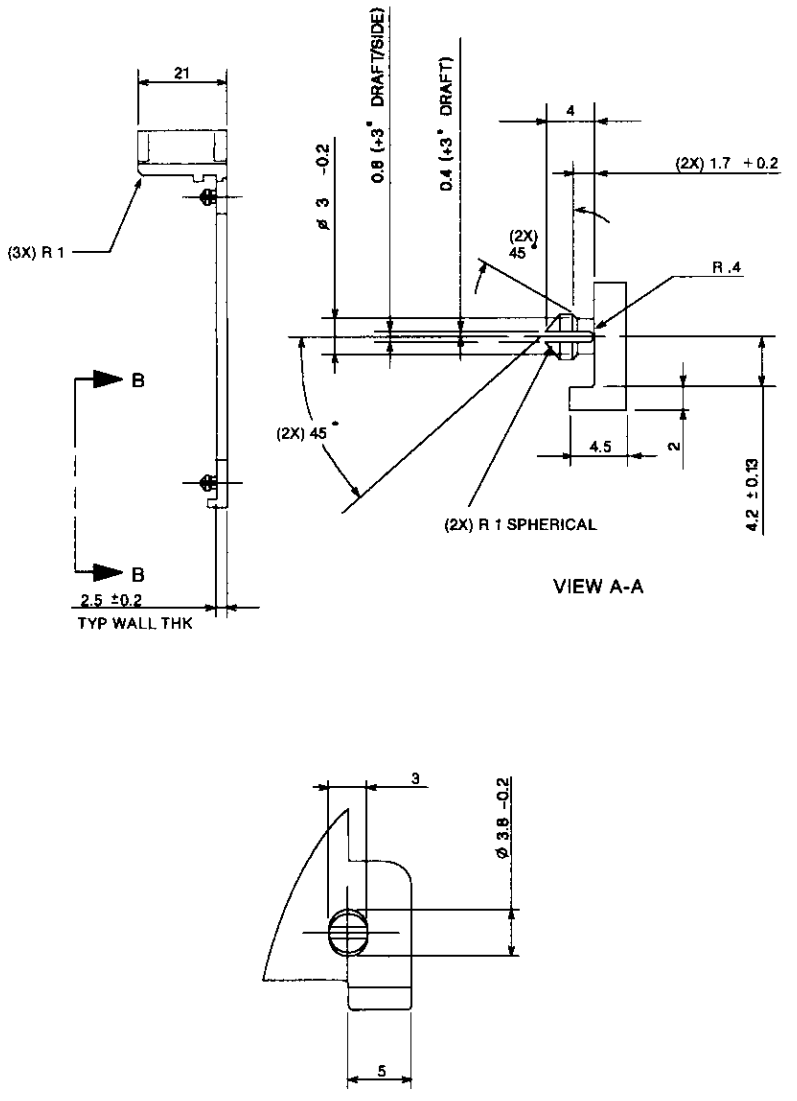
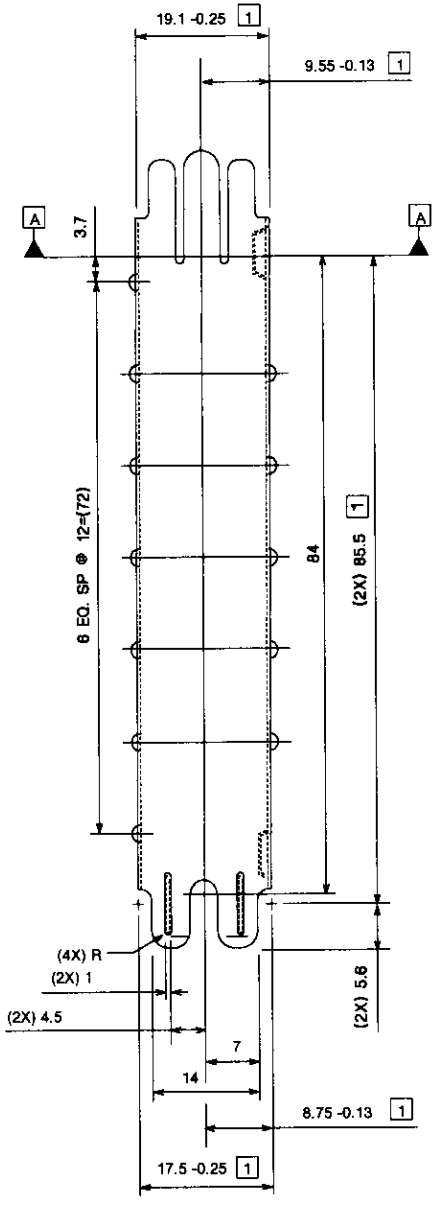


Figure 2-57. Adapter Retainer (Part 2 of 2)



1 DIMENSION IS CRITICAL TO FUNCTION.

Figure 2-58. Adapter Bracket (Part 1 of 4)

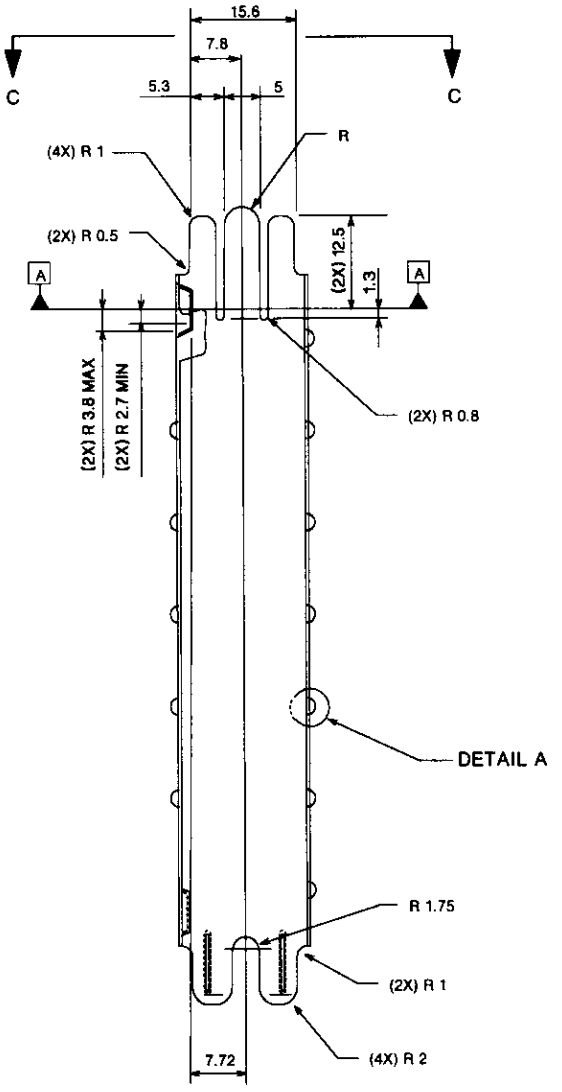


Figure 2-59. Adapter Bracket (Part 2 of 4)

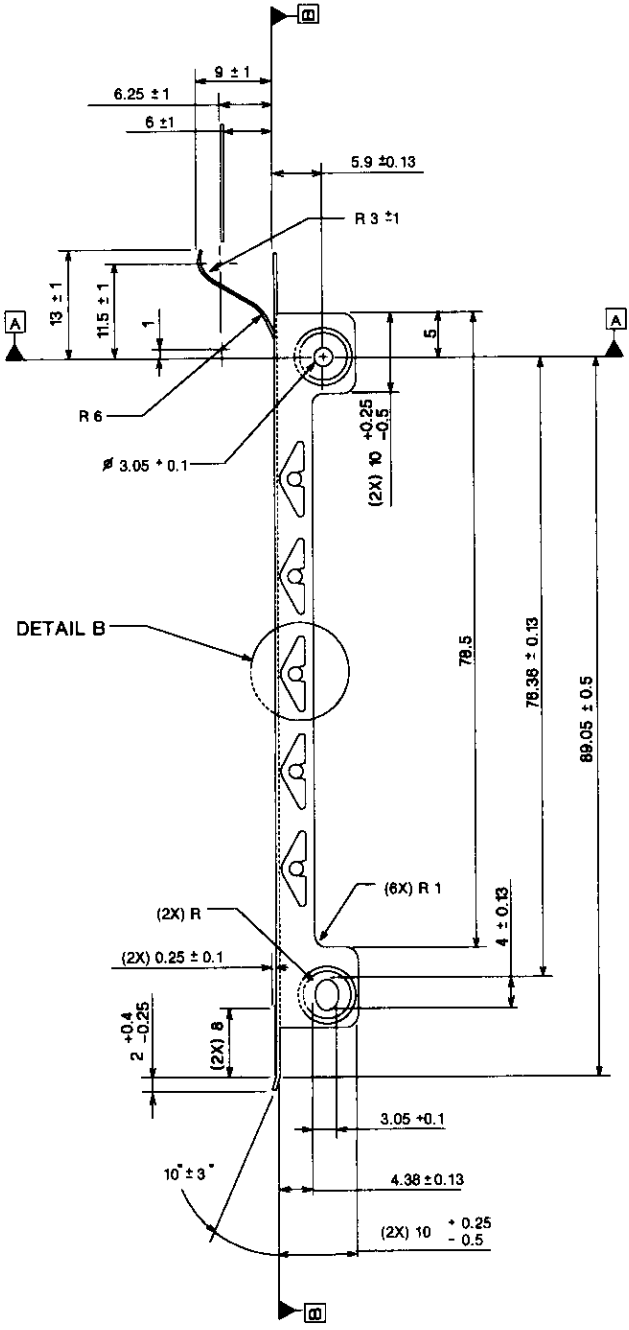
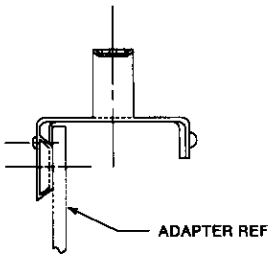
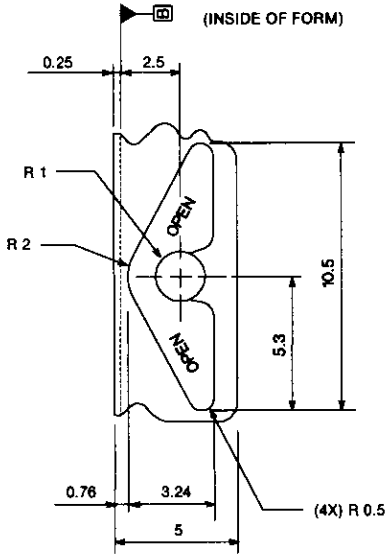


Figure 2-60. Adapter Bracket (Part 3 of 4)

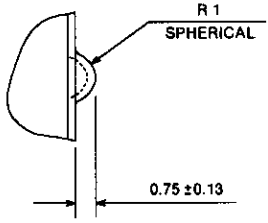


VIEW C-C



DETAIL B

SCALE 10/1
(12X)



DETAIL A

SCALE 10/1
(12X)

Figure 2-61. 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	Typical Current Per Connector	Maximum Statistical Current * Per Connector
+5.0 Vdc	1.4 A	1.6 A
+12.0 Vdc	0.100 A	0.175 A
-12.0 Vdc	0.040 A	0.040 A

Figure 2-62. Channel Load Current

* Maximum Statistical Current =

$$(I_{TC_1} + \cdots + I_{TC_n}) + \sqrt{((I_{MC_1} - I_{TC_1})^2 + \cdots + (I_{MC_n} + I_{TC_n})^2)}$$

Where:

- I_{MC} is the maximum current for each component on a given adapter.
- I_{TC} is the typical current for each component on a given adapter and the sum of all I_{TC} equals I_{TA}.

Note: If I_{MC} or I_{TC} is not available, estimate by using: $I_{TC} = 0.7 \times I_{MC}$

The total channel current is also determined in a statistical manner.
Total Channel Current =

$$(I_{TA_1} + \cdots + I_{TA_n}) + \sqrt{((I_{MA_1} - I_{TA_1})^2 + \cdots + (I_{MA_n} - I_{TA_n})^2)}$$

Where:

- I_{MA} is a maximum statistical current for a given adapter.
- I_{TA} is the typical current for a given adapter.

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, B33, B37, B41, A43, B45, B50, B54, B58	N/A
+5.0 Vdc	A7, A11, A15, A31, A39, A48, A56	+5% -4.5%
+12.0 Vdc	A19, A35, A52	+5% -4.5%
-12.0 Vdc	A23, A27	+10% -9.5%
Ground (Auxiliary Video)	BV1, AV3, BV5, AV7, BV9	N/A

Figure 2-63. 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 2-90. 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 2-58 on page 2-100.
- 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 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	Reserved
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 80FF	Video
FFFF	Device Not Attached

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

Figure 2-64. 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, 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.
- The only memory adapters supported by Model 50 and Model 60 POST are the IBM Personal System/2 80286 Memory Expansion Option and any other memory adapters equivalent in function and identical in design to the IBM Personal System/2 80286 Memory Expansion Option.

Notes: