
Micro Channel Architecture

Description	1
Micro Channel Participants	2
Masters	2
Slaves	3
Micro Channel Structure	5
Addressing Model	6
Micro Channel Buses and Signals	7
Arbitration Bus and Associated Signals	7
Address Bus and Associated Signals	9
Address Signal Combinations	12
Data Bus and Associated Signals	13
Interrupt Signals	18
Other Micro Channel Signals	19
Channel Signal Groups (Digital)	20
Matched-Memory Extension Signal Descriptions	23
Auxiliary Video Extension Signal Descriptions	23
Micro Channel Architecture Base and Extensions	24
32-Bit Connector and Extensions	25
16-Bit Connector and Extensions	26
Micro Channel Procedures	27
Arbitration	27
Channel Ownership	27
Central Arbitration Control Point	28
Local Arbiters	30
Burst Transfer	32
Preemption	33
Fairness Feature and the Inactive State	34
Arbitration Timing	35
Data Transfer	39
Basic Transfer	40
Basic Transfer Procedure	40
Default Cycle Timing (200 ns Minimum)	43
Synchronous-Extended Cycle Timing (300 ns Minimum)	47
Asynchronous-Extended Cycle Timing (≥ 300 ns)	50
Data Steering	53
Data Steering Control	54
Master Replication of Data Bytes	55

Channel Support	55
Address Bus Translator	55
Central Data Steering	56
TR 32	56
Central Steering Logic	56
Central Translator Logic	56
-BE(0-3)	56
Streaming Data Transfer	58
Streaming Data Procedure (16- or 32-Bit)	59
Streaming Data Procedure (64-Bit)	61
Streaming Data Transfer Rates	63
Address Bus Management	63
Address Boundary Alignment	64
Streaming Data Cycles	66
64-Bit Streaming Data Cycles	68
Starting the Cycle	70
Deferring the Start of the Cycle	70
Pacing the Data	72
Terminating the Cycle	74
Streaming Data Timing	77
Direct Memory Access Procedure	83
Read and Write Operations	83
DMA Slave Selection	84
Address During DMA Operation	84
Transfer Data Length	84
DMA Procedure	84
Single Transfer	85
Burst Transfer (Multiple Transfer Cycles)	86
DMA Terminating Conditions	86
Use of -BURST	87
DMA Timings	88
Single DMA Transfer (DMA-Controller Controlled)	88
Burst DMA Transfer (DMA-Controller Terminated)	90
Burst DMA Transfer (DMA-Slave Terminated - Default Cycle, 200 ns)	91
Burst DMA Transfer (DMA-Slave Terminated - Synchronous-Extended Cycle, 300 ns)	94
Burst DMA Transfer (DMA Slave Terminated - Asynchronous-Extended Cycle, ≥ 300 ns)	96
Interrupt Procedures	98
Refresh	100
System Configuration	101
Multiple Card Adapters	102

Adapter Configuration Data	103
Required Configuration Fields	103
Adapter ID Field	104
Card Enable Field	104
Channel Check Field	104
Conditionally-Required Configuration Fields	104
Adapter I/O Address Select Field	105
Adapter ROM/RAM Address Select Field	105
Arbitration Level Field	105
Auto-Incrementing Enable Field	105
Channel Check Exception Reporting Field	106
Channel Check Status Indicator Field	106
Data and Address Parity Enable Field	106
Device I/O Address Assignment	106
Matched Memory Enable Field	107
Selected Feedback Return Exception Enable Field	107
Streaming Data Enable Field	108
Optional Configuration Fields	108
Option Select Data Field	108
Fairness Feature Enable Field	108
POS Subaddress Extension Field	108
Status Information Field	111
System Configuration Procedure	112
Address Bus During Setup Cycles	112
Card Enable Field During Setup Cycles	113
Channel Check Status During Setup Cycles	113
-CD SETUP During Setup Cycles	113
CD CHRDY During Setup Cycles	113
System Configuration Timing Diagrams	114
Exception Condition Reporting and Handling	118
Exception Conditions	118
Channel Data and Address Parity Exceptions	119
Data Parity	120
Read Operations - Basic Transfer	120
Read Operations - Streaming Data Transfer	122
Write Operations - Basic Transfer	122
Write Operations - Streaming Data Transfer	124
Data Parity Exception Handling	126
Bus Master	126
System Master	126
DMA Controller	126
Slaves	126
Address Parity	127

Channel Non-Parity Exceptions	128
Invalid -DS 16 RTN and -DS 32 RTN Combination	128
Selected Feedback Return Exception	129
Master and Slave Dependent Exceptions	130
Channel Timeout Exception	130
Exception Reporting	131
Exception Reporting using -CHCK	131
Synchronous -CHCK	132
Asynchronous -CHCK	137
Exception Reporting Using Interrupts	138
Exception Handling	138
Bus Master	138
System Master	138
DMA Controller	139
Slaves	139
Select Feedback Exception Handling	140
Bus Master	140
System Master	140
DMA Controller	140
Channel Reset	141
Connector Definition	142
Micro Channel Connector	144
Matched-Memory Extension	148
Auxiliary Video Extension	149
Base Video Extension	149
Adapter Physical Specifications	150
Adapter Size	151
Connector Spacing	151
Power	152
Voltage Regulation	153
Index	154

Figures

1.	Master-Slave Relationships	4
2.	Micro Channel Structure	5
3.	I/O and Memory Address Space	6
4.	Address Bus Support - Memory Cycle	12
5.	Address Bus Support - I/O Cycle	12
6.	Port Size of Selected Slave	16
7.	Streaming Data Rates	17
8.	I/O and Memory Transfer Controls	17
9.	Driver and Receiver Requirements and Options	21
10.	Signal Groups, Driver Types, and Loading Specifications	22
11.	The 32-Bit Connector and the Optional Extensions	25
12.	The 16-Bit Connector and the Optional Extensions	26
13.	Arbitration Block Diagram	29
14.	Local Arbiter Example	31
15.	Burst Transfer	32
16.	Preempted Burst Transfer	33
17.	Exiting from the Inactive State	35
18.	Arbitration Cycle	36
19.	Arbitration Cycle Timing Specifications	36
20.	First Cycle After Grant	38
21.	First-Cycle-After-Grant Timing Specifications	38
22.	Basic Transfer Procedure - Overview	42
23.	Default Cycle (200 ns Minimum)	44
24.	Default Cycle (200 ns Minimum) Timing Specifications	44
25.	Synchronous-Extended Cycle (300 ns Minimum)	48
26.	Synchronous-Extended Cycle Timing Specifications (300 ns Minimum)	48
27.	Asynchronous-Extended Cycle Timing Specifications (≥ 300 ns)	50
28.	Asynchronous Extended Cycle (≥ 300 ns)	51
29.	Steering Control	53
30.	Steering Responsibility (Basic Transfer, DMA, and Setup Procedures)	54
31.	Steering Responsibility (Streaming Data Procedure)	55
32.	Data Bus Steering Implementation	57
33.	Streaming Data Transfer Rates	63
34.	Data Block Example	65
35.	Streaming Data Signal Combinations	66
36.	16- and 32-Bit Streaming Data Cycle Overview	67

37.	64-Bit Streaming Data Cycle Overview	68
38.	Streaming Data Deferred Start Overview	71
39.	CD CHRDY and Data Pacing	73
40.	Master-Terminated Streaming Data Cycle	74
41.	Master-Terminated Streaming Data Cycle (Slave Not Ready)	75
42.	Slave-Terminated Streaming Data Cycle	76
43.	Streaming Data Cycle	77
44.	Streaming Data Cycle - Deferred Start	78
45.	Streaming Data Transfer Timing Specifications	78
46.	Streaming Data Clocking (16- and 32-Bit)	80
47.	Streaming Data Transfer Timing Specifications (16- and 32-Bit)	80
48.	Streaming Data Clocking (64-Bit)	82
49.	Streaming Data Transfer Timing Specifications (64-Bit)	83
50.	Single DMA Transfer (DMA-Controller Controlled)	88
51.	Single DMA Transfer (DMA-Controller Controlled) Timing Specifications	89
52.	Burst DMA Transfer (DMA-Controller Terminated)	90
53.	Burst DMA Transfer (DMA-Controller Terminated) Timing Specifications	90
54.	Burst DMA Transfer (DMA-Slave Terminated - Default Cycle, 200 ns)	92
55.	Burst DMA Transfer (DMA-Slave Terminated - Default Cycle, 200 ns) Timing Specifications	92
56.	Burst DMA Transfer (DMA-Slave Terminated - Synchronous-Extended Cycle)	94
57.	Burst DMA Transfer (DMA-Slave Terminated - Synchronous-Extended Cycle, 300 ns) Timing Specifications	95
58.	Burst DMA Transfer (DMA-Slave Terminated - Asynchronous-Extended Cycle, ≥ 300 ns)	96
59.	Burst DMA Transfer (DMA-Slave Terminated - Asynchronous-Extended Cycle, ≥ 300 ns) Timing Specifications	97
60.	Typical Adapter Interrupt Sharing Implementation	98
61.	POS I/O Address Space	103
62.	Subaddressing Model	110
63.	POS I/O Address Decode	112
64.	Setup Cycle Overview	114
65.	System Configuration Timing	115
66.	System Configuration Timing Specifications	116
67.	Data Parity Checking Read Cycle - Basic Transfer	121
68.	Data Parity Checking Read Timing Specifications	121
69.	Data Parity Read Cycle - Streaming Data Transfer	122

70.	Data Parity Checking Write Cycle - Basic Transfer	123
71.	Data Parity Checking Write Timing Specifications	123
72.	Data Parity Write Cycle - Streaming Data Transfer	125
73.	Data Parity Checking Write Timing Specifications	125
74.	Address Parity	127
75.	Address Parity Timing Specifications	127
76.	Channel-Check - Default Cycle	133
77.	Channel-Check Timing Specifications - Default Cycle	133
78.	Channel-Check - Extended Cycle	134
79.	Channel-Check Timing Specifications - Extended Cycle	134
80.	Channel Check - Streaming Data Read Cycle	135
81.	Channel-Check Timing Specifications - Streaming Data Read Cycle	135
82.	Channel Check - Streaming Data Write Cycle	136
83.	Channel-Check Timing Specifications - Streaming Data Write Cycle	136
84.	Micro Channel Bus and Signal Status after a CHRESET	141
85.	Micro Channel Connectors	143
86.	Micro Channel Connector	145
87.	Micro Channel Connector - continued	146
88.	Micro Channel Connector - continued	147
89.	Channel Connector Voltage and Signal Assignments (Matched-Memory Extension)	148
90.	Auxiliary Video Extension	149
91.	Base Video Extension	150
92.	Minimum current available for the Type 3 adapter.	152
93.	Minimum current available for the Type 5 adapter	152
94.	Channel Voltage Regulation	153

Notes:

Description

The Micro Channel™ architecture consists of an address bus, a data bus, an arbitration bus, a set of interrupt signals, and support signals. It uses synchronous and asynchronous procedures for data transfer between memory, I/O devices, and a controlling master. The controlling master can be a DMA controller, the system master (system processor), or a bus master. The features of the Micro Channel architecture are:

- I/O data transfers of 8-, 16-, 24-, or 32-bits within a 64KB¹ address space (16-bit address width).
- Memory data transfers of 8-, 16-, 24-, or 32-bits within a 16MB¹ (24-bit address width) or 4GB¹ (32-bit address width) address space.
- An arbitration procedure that enables up to 15 devices and the system master to bid for control of the channel.
- A basic transfer procedure that allows data transfers between masters and slaves.
- A direct memory access (DMA) procedure that supports multiple DMA channels. Additionally, this procedure allows a device to transfer data in bursts.
- An optional streaming data procedure that provides a faster data-transfer rate than the basic transfer procedure and allows 64-bit data transfers.
- Address- and data-parity enable and detect procedures.
- Interrupt sharing on all levels.
- A flexible system-configuration procedure that uses programmable registers.

Micro Channel is a trademark of the IBM Corporation

¹ KB equals 1024 bytes; MB equals 1,048,576 bytes; GB equals 1,073,741,824 bytes.

- An adapter interface to the channel using:
 - A 16-bit connector with a 24-bit address bus and a 16-bit data bus
 - A 32-bit connector with a 32-bit address bus and a 32-bit data bus
 - An optional matched-memory extension
 - An optional video extension.
- Support for audio signal transfer (audio voltage-sum node).
- Support for both synchronous and asynchronous data transfer.
- An exception condition reporting procedure.
- Improved electromagnetic characteristics.

Micro Channel Participants

All Micro Channel participants are either masters or slaves. There are three types of masters and three types of slaves. (See Figure 1 on page 4).

An adapter can incorporate either a master function, a slave function, or a combination of both. For example, an adapter might be designed to operate primarily as a DMA slave. However, it would also respond to certain I/O read and I/O write operations from the system master, making it an I/O slave. If the adapter contains RAM or ROM that is in the memory address space, it would be a memory slave when that memory was accessed.

Masters

A master is a participant that drives the address bus and data transfer control signals that cause data transfer to or from a slave.

The channel supports up to 16 masters. The central arbitration control point and the Micro Channel arbitration procedure control ownership of the channel. The central arbitration control point grants ownership at the end of the arbitration procedure to highest-priority requester.

The three types of masters are:

- **System master**

A system master controls and manages the system configuration. It arbitrates for use of the channel. A system master can also be a default master. The default master is the master that owns the channel when no other master requires the channel.

The system master supports data transfers with an I/O slave or a memory slave.

- **Bus master**

A bus master arbitrates for use of the channel. A bus master supports data transfers with an I/O slave or a memory slave.

- **DMA controller**

A DMA controller does not initiate arbitration for the channel; a DMA slave initiates and completes the arbitration procedure. The DMA controller monitors the arbitration bus to detect the DMA slave's arbitration level.

A DMA controller supports data transfers with DMA slaves and memory slaves.

Slaves

A slave is a participant that sends and receives data under the control of a master. The slave responds to signals that are driven by the master. A slave is selected by the controlling master using Micro Channel procedures.

The channel supports 8-, 16- and 32-bit data bus size in a slave. The data port size (8-, 16-, or 32-bits) describes the maximum width of the data transfer. For example, a 32-bit slave can support 8-, 16-, 24-, or 32-bit data transfers. During a data transfer cycle, an 8-bit data port on a 32-bit slave acts like an 8-bit slave.

The three types of slaves are:

- I/O slave

I/O slaves are selected by their address within the I/O address space.

- Memory slave

Memory slaves are selected by their address within the memory address space.

- DMA slave

A DMA slave is selected by arbitration or, optionally, by its address within the I/O address space. A DMA slave initiates a request for the channel and arbitrates for the channel with arbitrating masters and other DMA slaves. The DMA controller provides the DMA slave with the address and data transfer signals required for data transfers. During a DMA write operation, the DMA slave provides the data, and the memory slave stores it. During a DMA read operation, the memory slave provides the data, and the DMA slave stores it.

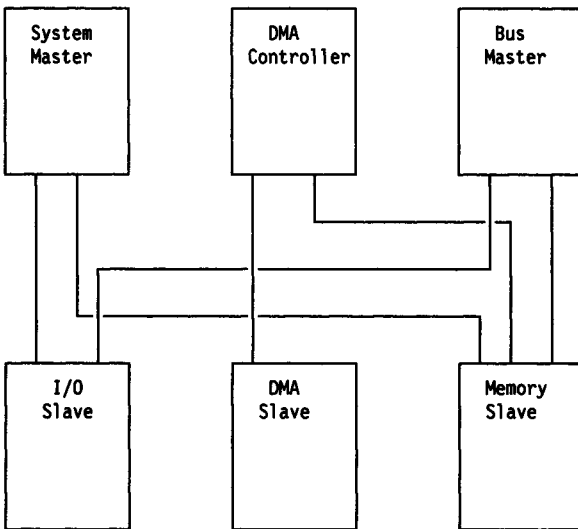
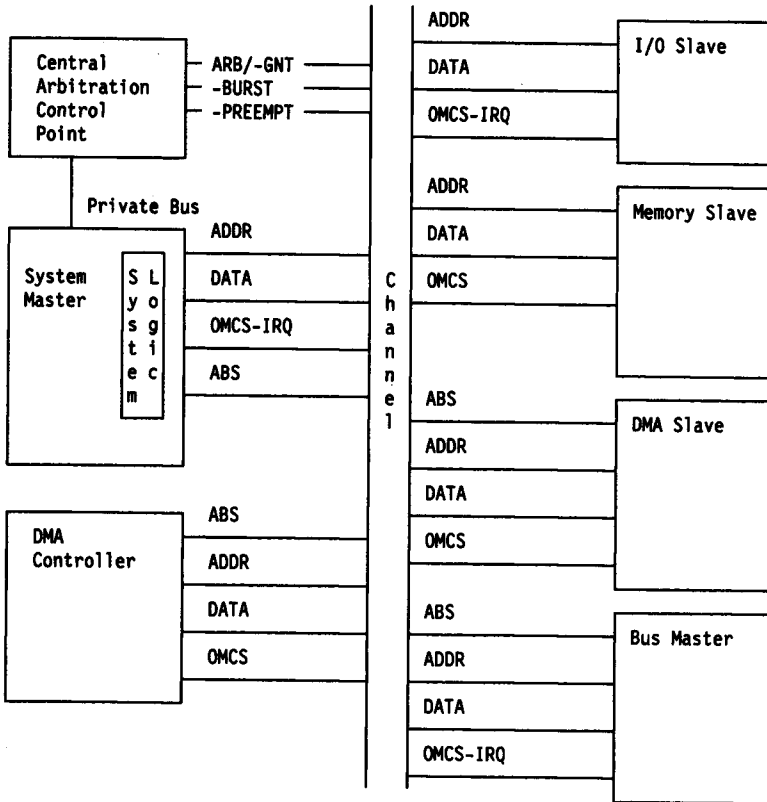


Figure 1. Master-Slave Relationships

Micro Channel Structure

The following shows the participants, the central arbitration control point, and the Micro Channel buses and signals that they drive or receive. "Micro Channel Buses and Signals" on page 7 describes the buses and signals of the Micro Channel architecture.



- ABS - Arbitration Bus and Associated Signals
- ADDR - Address Bus and Associated Signals
- DATA - Data Bus and Associated Signals
- IRQ - Interrupt Request Signals
- OMCS - Other Micro Channel Signals

Figure 2. Micro Channel Structure

Figure 9 on page 21 shows which Micro Channel participant drives and receives each bus and signal, and specifies the requirements for each driver and receiver.

Addressing Model

The Micro Channel addressing model consists of a memory address space and an I/O address space. During an I/O cycle, the 64KB I/O address space is addressed by the low-order 16 bits of the address bus. During a memory cycle, the memory address space is addressed by the address bus.

The signals used for an address of up to 4GB are MADE 24, M/-IO, and A0 through A31 of the address bus. An address of up to 16MB uses A0 through A23 plus MADE 24 and M/-IO.

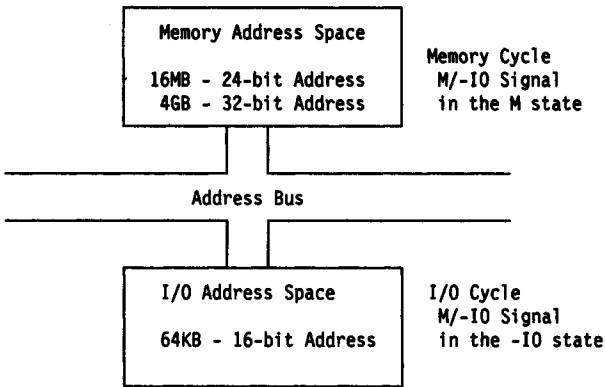


Figure 3. I/O and Memory Address Space

Micro Channel Buses and Signals

The Micro Channel architecture consists of:

- An arbitration bus and associated signals
- An address bus and associated signals
- A data bus and associated signals
- Interrupt signals
- Other Micro Channel signals
- Optional extensions for:
 - Matched memory extension signals
 - Video extension signals.

Throughout this document, a minus sign (-) in front of a signal name indicates that the signal is active when it is at a low-voltage level. When no minus sign appears, the signal is active when it is at a high-voltage level. For example, -CMD specifies the signal is active low. Also, ARB/-GNT is in the ARB state when it is at a high level and is in the -GNT state when it is at a low level.

Arbitration Bus and Associated Signals

The arbitration bus and associated signals allow arbitrating participants (the system master, bus masters, and DMA slaves) to request and gain ownership of the channel. The resolution of multiple arbitration requests results in granting ownership of the channel to the highest-priority requester. See "Arbitration" on page 27 for more information.

ARB0 – ARB3: Arbitration Bus: These signals make up the arbitration bus. They are driven by the system master, bus masters, and DMA slaves to present their arbitration level when requesting ownership of the channel. ARB0 (least significant) through ARB3 (most-significant) support up to 16 arbitration levels.

The highest value of the arbitration bus (hex F) has the lowest priority, and the lowest value (hex 0) has the highest priority. To

participate in the arbitration procedure, an arbitrating participant must present its arbitration level immediately after the rising edge of ARB/-GNT. All arbitrating participants monitor the arbitration bus, and those with lower priority arbitration levels withdraw them by not driving less-significant arbitration bits.

The arbitration level of the highest-priority requester is valid on the arbitration bus after a settling time. After the channel is granted to the highest-priority requester, that requester continues to drive its arbitration level on the bus.

ARB/-GNT: Arbitrate/-Grant: Only the central arbitration control point drives this signal. The negative-to-positive transition of ARB/-GNT initiates an arbitration cycle. When in the ARB state, this signal indicates an arbitration cycle is in progress. When in the -GNT state, this signal indicates the acknowledgment from the central arbitration control point to the arbitrating participants and the DMA controller that channel ownership has been granted. This signal is driven to the ARB state by the central arbitration control point following the end of transfer (EOT). (See page 28 for a description of the end of transfer.)

Note: The system master can perform data transfers during arbitration (ARB/-GNT in the ARB state).

-BURST: -Burst: This signal is optionally driven by the winning arbitrating participant or the DMA controller after ARB/-GNT is driven to the -GNT state. This signal indicates to the central arbitration control point that the controlling master will use the channel for one or more consecutive data transfer cycles. This type of data transfer is called *burst transfer*.

-PREEMPT: -Preempt: This signal is used by arbitrating participants to request use of the channel through arbitration. Any arbitrating participant that requires ownership of the channel drives -PREEMPT active, causing an arbitration cycle to occur. When a participant is granted control of the channel, it stops driving -PREEMPT. All arbitrating participants that have not been granted ownership keep their requests pending by continuing to drive -PREEMPT active. All masters and DMA slaves that use burst transfer must receive -PREEMPT.

Address Bus and Associated Signals

The address bus and the associated signals are used by the controlling master to assert the memory address or the I/O address (M-/IO), to enable a slave to latch the address and status signals (-S0, -S1), and to indicate that the memory address is greater than 16MB.

A0 – A23: Address Bits 0 through 23: These lines, along with A24 through A31, make up the address bus. These lines are driven by the controlling master to address memory, I/O slaves, and, optionally, DMA slaves. A0 is the least-significant bit and A23 is the most-significant bit. These 24 address lines allow access of up to 16MB of memory. Only the lower 16 address lines (A0 through A15) are for I/O operations, and all 16 lines must be decoded by the I/O slave. A0 through A23 are generated by the controlling master. Valid addresses, generated by the controlling master, are unlatched on the channel and, if required, must be latched by the slaves using either the leading or trailing edge of -ADL or the leading edge of -CMD.

A24 – A31: Address Bits 24 through 31: These lines, along with A0 through A23 are driven by the controlling master to address memory attached to the channel. A0 is the least-significant bit and A31 is the most-significant bit. These additional address lines allow access of up to 4GB of memory. A24 through A31 are generated by the controlling master. Valid addresses, generated by the controlling master, are unlatched on the channel and, if required, must be latched by the slaves using either the leading or trailing edge of -ADL or the leading edge of -CMD.

Note: A0 - A31 are used to transfer data during a 64-bit streaming data cycle.

-ADL: -Address Decode Latch: This signal, driven by the controlling master, is provided as a convenient way for the slave to latch valid address decodes and status bits.

-APAREN: -Address Parity Enable: This optional signal is driven active by a master when the master places an address on the bus. This signal indicates to a slave that the address parity signals are valid.

APAR0 – APAR3: Address Parity Bits 0 through 3: These optional signals are driven by a master when an address is placed on the address bus. These signals represent the odd parity of the address bits on the address bus during both read and write operations. (Odd parity is the condition where the total number of 1s in a byte of data, including the parity bit, is odd.)

APAR(0) represents the odd parity of A(0-7).

APAR(1) represents the odd parity of A(8-15).

APAR(2) represents the odd parity of A(16-23).

APAR(3) represents the odd parity of A(24-31).

During both read and write operations, a master generates a parity bit for each valid address byte, and the receiving slave optionally performs the parity checking to ensure the integrity of the address.

Note: APAR0 - APAR3 represent data parity during 64-bit streaming data cycle when -DPAREN is active.

APAR(0) represents the odd parity of D(32-39).

APAR(1) represents the odd parity of D(40-47).

APAR(2) represents the odd parity of D(48-55).

APAR(3) represents the odd parity of D(56-63).

-CD SFDBK (n): -Card Selected Feedback: This signal is driven active by the selected slave as a positive acknowledgement of the slave's selection. The (n) indicates this signal is unique to each channel connector (one independent signal per connector). This signal is unlatched and driven by the selected slave. The slave does not drive -CD SFDBK during the configuration procedure (-CD SETUP active).

Note: Memory that contains diagnostic code must not drive -CD SFDBK during the diagnostic operation.

MADE 24: Memory Address Enable 24: This signal is driven by the controlling master and decoded by all memory slaves, regardless of the size of their address-space. When this signal is active, A24 - A31 are undefined.

- A master driving only A0 - A23 drives MADE 24 active.
- A master driving A0 - A31 drives MADE 24:
 - Active when all bits in A24 - A31 are 0
 - Inactive when any bit in A24 - A31 is 1.

M/-IO: Memory/-Input Output: This signal is driven by the controlling master and decoded by all slaves. This signal selects a memory cycle or an I/O cycle. When this signal is in the M state, a memory cycle is selected. When this signal is in the -IO state, an I/O cycle is selected.

-SFDBKRTN: Selected Feedback Return: This optional signal is generated by the system logic from the AND of the -CD SFDBK(n) signals being driven by slaves. This signal is a positive acknowledgement to the master from the slave that the slave is at the address specified by the master. Masters that support address parity must receive this signal.

Address Signal Combinations

The following tables show the valid addressing signal combinations for:

- the address bus (A0 - A23 and A24 - A31)
- M/-IO
- MADE 24
- -APAR(0-3).

Addressing Signals	16-Bit Connector	32-Bit Connector	
	24-Bit Address	< 16MB Address	≥ 16MB Address
M/-IO	1	1	1
MADE 24	1	1	0
A0 - A23	Valid	Valid	Valid
APAR0 - APAR2 (Optional)	N/A	Valid	Valid
A24 - A31	N/A	X	Valid
APAR3 (Optional)	N/A	X	Valid
X - Don't Care N/A - Not applicable			

Figure 4. Address Bus Support - Memory Cycle

Addressing Signals	16-Bit Connector	32-Bit Connector
M/-IO	0	0
MADE 24	X	X
A0 - A15	Valid	Valid
APAR0 - APAR1 (Optional)	N/A	Valid
X - Don't Care N/A - Not applicable		

Figure 5. Address Bus Support - I/O Cycle

Data Bus and Associated Signals

The data bus is used to transfer either 8, 16, 24, or 32 bits of data. The associated signals indicate the amount of data transferred by the master in a single transfer cycle, the size of the slave's data port, and the type (read or write) of the data transfer.

D0 – D15: Data Bits 0 through 15: These lines, along with D16 - D31, make up the data bus. The data bus is driven by any master or slave that is transferring data. These lines (D0 - D15) provide data bits 0 through 15. D0 is the least-significant bit; D15 is the most-significant bit. The 16-bit transfers from the controlling master to an 8-bit slave are converted by the controlling master to two 8-bit transfers, and are transmitted on lines D0 through D7. An 8-bit slave must use D0 through D7 to communicate with the controlling master.

D16 – D31: Data Bits 16 through 31: These lines, along with D0 - D15, make up the data bus. The data bus is driven by any master or slave that is transferring data. These lines (D16 - D31) provide data bits 16 through 31. D0 is the least-significant bit; D31 is the most-significant bit. The 32-bit transfers from the controlling master to an 8-bit slave are converted to four 8-bit transfers by the controlling master, and are transmitted on lines D0 through D7. The 32-bit transfers from the controlling master to a 16-bit slave are converted to two 16-bit transfers by the controlling master, and are transmitted on lines D0 through D15.

-BE0 – -BE3: -Byte Enable 0 through 3: These signals are used during data transfers with 32-bit slaves to indicate which data bytes are valid on the data bus. Data transfers of 8, 16, 24, or 32 contiguous bits are controlled by -BE0 through -BE3 during transfers involving 32-bit slaves only. These signals are driven by the controlling master when TR 32 is inactive, and by the central translator logic (for those operations involving a 16-bit master with a 32-bit slave) when TR 32 is active. These signals are not latched on the bus and, if required, are latched by 32-bit slaves. (See "Channel Support" on page 55 for additional information.)

-CD DS 16 (n): -Card Data Size 16: This signal is driven by 16-bit and 32-bit slaves to indicate a 16-bit or 32-bit data port at the location addressed. The (n) indicates this signal is unique to each channel connector (one independent signal per connector). This signal is derived from a valid unlatched address decode. -CD DS 16 is not driven by 8-bit slaves and is inactive for an 8-bit data port.

-CD DS 32 (n): -Card Data Size 32: This signal, along with -CD DS 16, is driven by 32-bit slaves to indicate a 32-bit data port at the location addressed. The (n) indicates this signal is unique to a channel connector position (one independent signal per connector). -CD DS 32 is derived from a valid unlatched address decode. -CD DS 32 is inactive for an 8- or 16-bit data port.

CD CHRDY (n): Channel Ready: This signal is normally active (ready) and is driven inactive (not ready) by a slave to allow additional time to complete a channel cycle. The (n) indicates this signal is unique to each channel connector (one independent signal per connector).

During a read cycle, a slave ensures that data is valid within the time specified after releasing the signal to a ready state. The slave also holds the data long enough for the controlling master to sample the data. A slave can also use this signal during a write cycle if more time is needed to store the data. This signal is initially driven from a valid unlatched address decode and status active.

CHRDYRTN: Channel Ready Return: This signal is the AND of CD CHRDY (n). It is driven by the system logic. If all slaves drive CD CHRDY active, this signal is active. CHRDYRTN allows the controlling master to monitor the ready information.

-CMD: -Command: This signal is driven by the controlling master and is used to define when data on the data bus is valid. The trailing edge of this signal indicates the end of the bus cycle. This signal indicates to the slave how long data is valid. During write cycles, the data is valid as long as -CMD is active. During read cycles, the data is valid after the leading edge, but before the trailing edge, of -CMD and is held on the bus until after -CMD goes inactive. Slaves can latch address and status information with the leading edge of -CMD.

-DPAREN: -Data Parity Enable: This optional signal is driven active by the participant when data is placed on the data bus. This signal indicates that the data parity signals are valid.

Note: APAR(0) - APAR(3) represent data parity during 64-bit streaming data cycles when -DPAREN is active.

DPAR0 – DPAR1: Data Parity Bits 0 and 1: These optional signals are driven by the participant when data is placed on the data bus. These signals represent the odd parity of the data bits on the data bus during both read and write operations. (Odd parity is the condition where the total number of 1s in a byte of data, including the parity bit, is odd.)

DPAR(0) represents the odd parity of D(0-7).

DPAR(1) represents the odd parity of D(8-15).

During write operations, a master generates a parity bit for each data byte being transferred, and the receiving slave optionally performs the parity checking to ensure the integrity of the data. During read operations, a slave generates a parity bit for each valid data byte, and the receiving master performs the parity checking to ensure the integrity of the data.

DPAR2 – DPAR3: Data Parity Bits 2 and 3: These optional signals are driven by the participant when data is placed on the data bus. These signals represent the odd parity of the data bits on the data bus during both read and write operations.

DPAR(2) represents the odd parity of D(16-23).

DPAR(3) represents the odd parity of D(24-31).

During write operations, a master generates a parity bit for each data byte being transferred, and the receiving slave optionally performs the parity checking to ensure the integrity of the data. During read operations, a slave generates a parity bit for each valid data byte, and the receiving master performs the parity checking to ensure the integrity of the data.

-DS 16 RTN: -Data Size 16 Return: This signal is driven by the system logic. This signal is the AND of -CD DS 16 (n) from each channel connector. If any slave drives its -CD DS 16 active, this signal is active. This signal allows the controlling master to monitor the information about the selected slave's data port size (see Figure 6).

-DS 32 RTN: -Data Size 32 Return: This signal is driven by the system logic. This signal is the AND of -CD DS 32 (n) from each channel connector. If any slave drives its -CD DS 32 active, this signal is active. This signal allows the controlling master to monitor the information about the selected slave's data port size.

-DS 16 RTN	-DS 32 RTN	Data Port Size
1	1	8-Bit Data Port
1	0	Not Valid
0	1	16-Bit Data Port
0	0	32-Bit Data Port

Figure 6. Port Size of Selected Slave

-MSDR: -Multiplexed Streaming Data Request: This signal is driven by a slave to indicate to the controlling master that the slave is capable of 64-bit streaming data transfers.

-SBHE: -System Byte High Enable: This signal is driven by the controlling master to indicate and enable transfers of data on D8 - D15. It is used with A0 to distinguish between high-byte transfers (D8 - D15) and low-byte transfers (D0 - D7) and double-byte (16-bit) transfers to 16-bit data ports. All 16-bit slaves receive this signal.

-SD STROBE: -Streaming Data Strobe: The controlling master and the slave use this signal to clock data on and off the data bus. This optional signal is driven by the controlling master. This signal also indicates to the slave that the controlling master supports streaming data.

-SDR(0): -Streaming Data Request 0: This optional signal is driven by a slave to indicate to the controlling master that the slave is capable of streaming data, and also indicates the maximum clocking rate the slave supports.

-SDR(1): -Streaming Data Request 1: This optional signal is driven by a slave to indicate to the controlling master that the slave is capable of streaming data, and also indicates the maximum clocking rate the slave supports.

-SDR(0)	-SDR(1)	Decoded Streaming Rate
1	1	Basic Transfer Cycle
0	1	10 MHz maximum (100 ns minimum cycle)
1	0	Reserved
0	0	Reserved

Figure 7. Streaming Data Rates

-S0, -S1: -Status 0, -Status 1: These status signals are driven by the controlling master to indicate the start of a data transfer cycle and also define the type of data transfer. When used with M/I-O, memory read or write cycles are distinguished from I/O read or write cycles. These signals are latched by the slave, as required, using the leading edge of -CMD, or the leading or trailing edge of -ADL.

Data is transferred to or from the data bus based on -CMD and a latched decode of the address, the status lines (-s0 exclusive-OR -s1), and M/I-O.

Slaves must support a full decode of -s0 and -s1. The following table shows the states of M/I-O, -s0, and -s1 in decoding I/O and memory read and write commands.

M/I-O	-S0	-S1	Function
0	0	0	Reserved
0	0	1	I/O Write Command
0	1	0	I/O Read Command
0	1	1	Inactive
1	0	0	Reserved
1	0	1	Memory Write Command
1	1	0	Memory Read Command
1	1	1	Inactive

Figure 8. I/O and Memory Transfer Controls

An I/O write command instructs an I/O slave to receive the data from the data bus. An I/O read command instructs an I/O slave to drive its data onto the data bus.

A memory write command instructs a memory slave to receive the data from the data bus. A memory read command instructs a memory slave to drive its data onto the data bus.

-TC: -Terminal Count: This signal is driven by the DMA controller and provides a pulse during a read or write command to the DMA slave to indicate that the terminal count of the current DMA channel has been reached. This indicates to the DMA slave that this is the last cycle to be performed. -TC is driven active on the channel during DMA operations only.

TR 32: Translate 32: This signal is driven inactive by 32-bit controlling masters and received by the central translator logic. The signal indicates to the central translator logic that the controlling master is performing data steering. TR 32 can also be received by any 32-bit slave. See "Data Steering" on page 53 for information about the central translator logic.

Interrupt Signals

-IRQ 3 – 7, -IRQ 9 – 12, and -IRQ 14 – 15: -Interrupt Request: An interrupt request is generated when an I/O slave drives one of the 'interrupt request' signals low. These signals make up the set of interrupt signals. The polarity of these signals makes it possible for multiple slaves to concurrently share the same interrupt level.

Other Micro Channel Signals

AUDIO: Audio Sum Node: This analog signal is the sum of all the audio signals being driven. It is used to drive audio signals from an adapter to the system audio output and between adapters. The frequency response of the audio signal is 50 Hz to 10 kHz \pm 3 dB. The maximum signal amplitude is 2.5 Vac peak-to-peak, at a dc offset of 0.0 Vdc \pm 50 millivolts. The noise level is limited to a maximum of 50 millivolts peak-to-peak.

AUDIO GND: Audio Ground: This is the analog ground return signal for the audio subsystem.

-CD SETUP (n): -Card Setup: This signal is driven by system logic to individually select channel connectors. The (n) indicates this signal is unique to each channel connector (one independent signal per connector). When this signal is activated, a specific channel connector is selected, and, if an adapter is present, access is gained to configuration data at that connector. The adapter ID and configuration data is obtained by an I/O read during a setup cycle; the configuration data is stored by an I/O write during a setup cycle.

-CHCK: -Channel Check: This signal is driven active by a slave to report an exception condition, and optionally, it can be driven by a master.

CHRESET: Channel Reset: This signal is generated by the system logic to reset or initialize all adapters at power-on or when a low voltage condition is detected. The system can also activate this signal under program control.

OSC: Oscillator: This signal is a high-speed clock driven by the system logic, with a frequency of 14.31818 MHz \pm 0.01%. The high-level pulse width (more than 2.3 Vdc) and the low-level pulse width (less than 0.8 Vdc) must not be less than 20 nanoseconds each.

-REFRESH: -Refresh: This signal is driven by the system logic and is used to indicate that a memory refresh operation is in progress. Memory slaves that do not need to perform refresh operations do not need to receive this signal.

Channel Signal Groups (Digital)

The following figure lists the digital Micro Channel signals and shows what type of driver or receiver is required. The 'audio' and 'audio ground' signals are analog signals. (For further information about these signals, see page 19.)

When a Micro Channel adapter combines the functions of bus masters and slaves, the signals driven and received by the adapter are a superset of the selected functions.

For example:

- An adapter that combines a slave function with a bus master function must drive and receive all signals required by a slave and all signals required by a bus master.
- An adapter that combines an I/O slave function with a memory slave or DMA slave function, must drive and receive all signals required by both slaves.

Note: All adapters must support setup.

The central arbitration control point, central steering logic, and central translator logic are packaged within the system logic. The system logic also includes circuitry for memory refresh, the oscillator, and channel reset.

| The signals for the 16-bit adapter are a subset of the signals for the
| 32-bit adapter.

Signal Name	System Master	DMA Cntl'r	Bus Master	DMA Slave	Memory Slave	I/O Slave	Driver Type
	D/R	D/R	D/R	D/R	D/R	D/R	Signal Group
A(0-15)	D/-	D/-	D/-	-/0	-/R	-/R	TS (1)
A(16-23)	D/-	D/-	D/-	-/-	-/R	-/-	TS (1)
A(24-31)	*/-	*/-	*/-	-/-	-/*	-/-	TS (1)
-ADL	D/-	D/-	D/-	-/0	-/0	-/0	TS (1)
APAR(0-3)	0/-	0/-	0/-	-/0	-/0	-/0	TS (2)
-APAREN	0/-	0/-	0/-	-/0	-/0	-/0	TS (2)
ARB(0-3)	0/0	-/R	D/R	D/R	-/-	-/-	OC (5)
ARB/-GNT	D/-	-/R	-/R	-/R	-/-	-/-	BD (4)
-BE(0-3)	*/-	*/-	*/-	-/*	-/*	-/*	TS (1)
-BURST	0/R	D/R	0/-	0/-	-/0	-/0	OC (5)
CD CHRDY (n)	-/R	-/-	-/-	0/-	0/-	0/-	TP (3)
-CD DS 16 (n)	-/R	-/-	-/-	\$/-	\$/-	\$/-	TP (3)
-CD DS 32 (n)	-/*	-/-	-/-	*/-	*/-	*/-	TP (3)
-CD SETUP (n)	D/-	-/0	-/0	-/0	-/0	-/0	TP (8)
-CD SFDBK (n)	-/R	-/-	-/-	D/-	D/-	D/-	TP (3)
-CHCK	-/R	0/R	0/R	0/-	0/-	0/-	OC (6)
CHRDYRTN	D/-	-/R	-/R	-/-	-/-	-/-	BD (4)
CHRESET	D/-	-/R	-/R	-/R	-/R	-/R	BD (4)
-CMD	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
D(0-7)	D/R	D/R	D/R	D/R	D/R	D/R	TS (2)
D(8-15)	D/R	D/R	D/R	\$\$	\$\$	\$\$	TS (2)
D(16-31)	**	**	**	**	**	**	TS (2)
DPAR(0)	0/0	0/0	0/0	0/0	0/0	0/0	TS (2)
DPAR(1)	0/0	0/0	0/0	0/0	0/0	0/0	TS (2)
DPAR(2-3)	0/0	0/0	0/0	0/0	0/0	0/0	TS (2)
-DPAREN	0/0	0/0	0/0	0/0	0/0	0/0	TS (2)
-DS 16 RTN	D/-	-/R	-/R	-/-	-/-	-/-	BD (4)
-DS 32 RTN	*/-	-/*	-/*	-/-	-/-	-/-	BD (4)
-IRQ (N)	0/R	0/-	0/0	0/-	0/-	0/-	OC (6)
M/-IO	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
MADE 24	D/-	D/-	D/-	-/-	-/R	-/-	TS (1)
-MSDR	-/0	-/0	-/0	0/-	0/-	0/-	TS (1)
OSC	D/-	-/0	-/0	-/0	-/0	-/0	CD (7)
-PREEMPT	D/R	-/-	D/0	D/0	-/-	-/-	OC (5)
-REFRESH	D/-	-/-	-/-	-/-	-/0	-/-	TS (1)
-S0, -S1	D/*	D/-	D/-	-/R	-/R	-/R	TS (1)
-SBHE	D/*	D/-	D/-	-/#	-/#	-/#	TS (1)
-SFDBKRTN	0/-	-/0	-/0	-/-	-/-	-/-	BD (4)
-SD STROBE	0/-	0/-	0/-	-/0	-/0	-/0	TS (1)
-SDR(0)	-/0	-/0	-/0	0/-	0/-	0/-	TS (1)
-SDR(1)	-/0	-/0	-/0	0/-	0/-	0/-	TS (1)
-TC	-/-	D/-	-/-	-/R	-/-	-/-	TS (1)
TR 32	*/R	*/-	*/-	-/0	-/0	-/0	TS (1)

- KEY -

D = Drive Enabled	OC = Open Collector	# = Required for 16-bit devices
0 = Optional	TS = Tri-State	\$ = Required for 16- or 32-bit devices
R = Receive Enabled	TP = Totem Pole	
- = Not Implemented	BD = Bus Driver	* = Required for 32-bit devices
CD = Clock Driver	N = IRQ(9-12,14,15,3-7)	@ = -CD SETUP is received by all adapters

Figure 9. Driver and Receiver Requirements and Options

The following table and notes apply to the driver and receiver options listed on page 21.

Signal Group	Driver Type	Sinking Capacity	Capacitive Loading	Total Capacitance
1	Tri-State	24 mA	20 pF	240 pF
2	Tri-State	24 mA	20 pF	240 pF
3	Totem Pole	8 mA	20 pF	40 pF
4	Bus Driver	24 mA	20 pF	240 pF
5	Open Collector	24 mA	15 pF	200 pF
6	Open Collector	24 mA	20 pF	240 pF
7	Clock Driver	24 mA	15 pF	200 pF
8	Totem Pole or Tri-State	8 mA	50 pF	Not Applicable

Figure 10. Signal Groups, Driver Types, and Loading Specifications

Notes:

1. Pull-up resistors are provided by system logic for all signals other than -SDR0, -SDR1, -MSDR, and -SD STROBE. Adapters that receive these signals must provide a pull-up resistor for each signal received. (A minimum of 20 kΩ pulled to 5 Vdc is recommended.)
2. The maximum loading current is 1.6 mA per channel connector, except signal group 5. The maximum loading current of group 5 is 1.0 mA per channel connector.
3. 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 of the tri-state device to control the output.
4. For good signal quality, the net length on the adapter for any control signal driven or received on the adapter should not exceed 2 inches.

Matched-Memory Extension Signal Descriptions

The matched-memory extension can, in some cases, improve the data transfer capabilities between the system master and channel-resident memory. The matched-memory signal-sequence and signal-timing specifications are system dependent. For more information, see the technical manuals for the systems that support the matched-memory signals.

Auxiliary Video Extension Signal Descriptions

| The auxiliary video extension allows optional video adapters to share video resources with the base video subsystem. The base video can be on the system board or on an adapter with the base video extension. When the base video subsystem is on the system board, support of the auxiliary video extension is direct. When the base video subsystem is on an adapter, support of the auxiliary video extension is through the base video extension.

| The signals and signal timings for the auxiliary and base video extensions are system dependent. For information on these signals, see the technical references for your specific system.

| The auxiliary video extension is shown in Figure 90 on page 149.
| The base video extension is shown in Figure 91 on page 150.

Micro Channel Architecture Base and Extensions

| Figure 11 on page 25 shows the buses and signals available for use
| with the 32-bit connector. The optional extensions signals are shown
| at the top of the figure and the optional connector pins at the bottom
| of the figure.

| Figure 12 on page 26 shows the buses and signals available for use
| with the 16-bit connector. The optional extensions signals are shown
| at the top of the figure and the optional connector pins at the bottom
| of the figure. The 16-bit connector is a subset of the 32-bit connector.

“Connector Definition” on page 142 contains the physical specifications for the connectors.

32-Bit Connector and Extensions

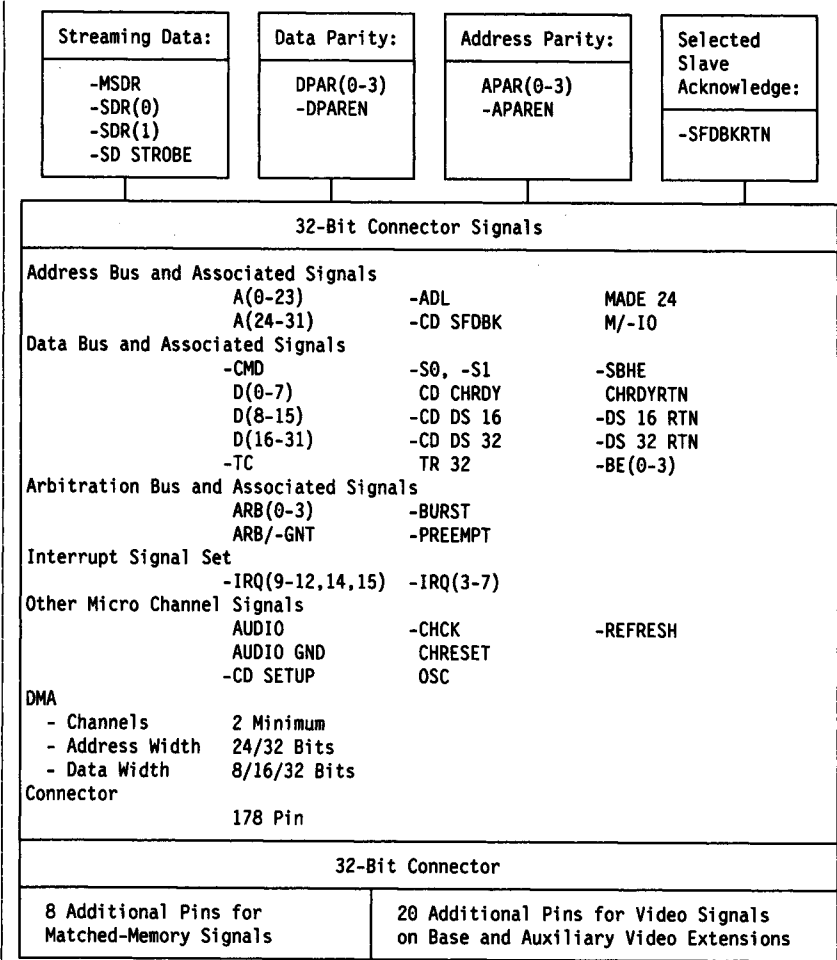


Figure 11. The 32-Bit Connector and the Optional Extensions

Micro Channel Procedures

The Micro Channel procedures, as well as data steering, are described in the following pages.

Arbitration

The arbitration procedure provides the functions to resolve multiple requests for control of the channel. Arbitration consists of local arbiters, the arbitration bus and the associated signals, and a central arbitration control point.

The arbitration bus is a parallel bus that consists of four lines ARB(0-3), supporting up to 16 arbitration levels. These arbitration levels are hex 0 through hex F; level hex 0 is the highest priority, level hex F is the lowest priority. The master with level hex F is always the default master.

If no -PREEMPT is driven, the arbitration level defaults to hex F and the default master gains control of the channel.

Channel Ownership

Channel ownership is controlled by the central arbitration control point based on prioritized arbitration of up to 16 devices. These arbitrating devices can be the system master, bus masters, or DMA slaves. If either a bus master or the system master wins the arbitration, it controls the bus and becomes the controlling master. If a DMA slave wins the arbitration, the supporting DMA controller becomes the controlling master.

If a system master is also the default master, the system master controls the Micro Channel when the channel is not controlled by any other master.

Central Arbitration Control Point

The central arbitration control point controls arbitration cycles using three arbitration signals: -PREEMPT, ARB/-GNT, and -BURST.

Arbitrating participants (local arbiters) drive -PREEMPT active to request use of the channel. An arbiter can activate -PREEMPT asynchronously. However, an arbiter must have -PREEMPT activated *before* the beginning of an arbitration cycle to participate in that cycle. If an arbiter drives -PREEMPT active when an arbitration cycle is in progress, it must wait for the *next* arbitration cycle before driving its arbitration level onto the arbitration bus.

The central arbitration control point initiates an arbitration cycle by driving ARB/-GNT to the ARB state. The requesting local arbiters then drive their 4-bit arbitration level onto the arbitration bus. When a local arbiter sees a more-significant bit low on the arbitration bus (other than those driven low by itself), it stops driving its less-significant bits onto the arbitration bus. The local arbiter driving the lowest arbitration level (highest priority) thereby wins control of the channel when ARB/-GNT goes to the -GNT state.

Arbitrating participants with multiple data transfers to perform signal the central arbitration control point by driving -BURST active. -BURST is held active until all transfers have been completed or until another participant drives -PREEMPT active. When preempted, the controlling master or DMA slave must release the channel and signal the end of transfer within 7.8 microseconds.

The end of transfer (EOT) is indicated by the trailing edge of -S₀, -S₁, -BURST, or -CMD, whichever occurs last. When the central arbitration control point detects an EOT condition, it drives ARB/-GNT to the ARB state, initiating the next arbitration cycle.

When ARB/-GNT is in the ARB state, bus masters and DMA controllers disable all of their channel drivers except -PREEMPT and ARB(0-3). Slaves, including bus masters and DMA controllers responding as slaves, must be capable of driving the channel when addressed.

When a master gains control of the channel, it is required to initiate a transfer cycle. If the master must abort the transfer cycle, it activates -S(0,1) with the minimum pulse width of T_{2A}, and does *not* activate -ADL and -CMD (see Figure 24 on page 45 for the T_{2A} timing specification).

The T2A time allows the central arbitration control point to recognize EOT and allows adapters to exit the inactive state.

During exception conditions, the central arbitration control point can gain control of the channel by driving ARB/-GNT to the ARB state independently of an EOT condition.

Note: While ARB/-GNT is in the ARB state, the system master can perform data transfers and system logic can perform refresh operations. Selected slaves must respond independently of the state of ARB/-GNT

The following is a block diagram of the interaction between the central arbitration control point and the local arbiters.

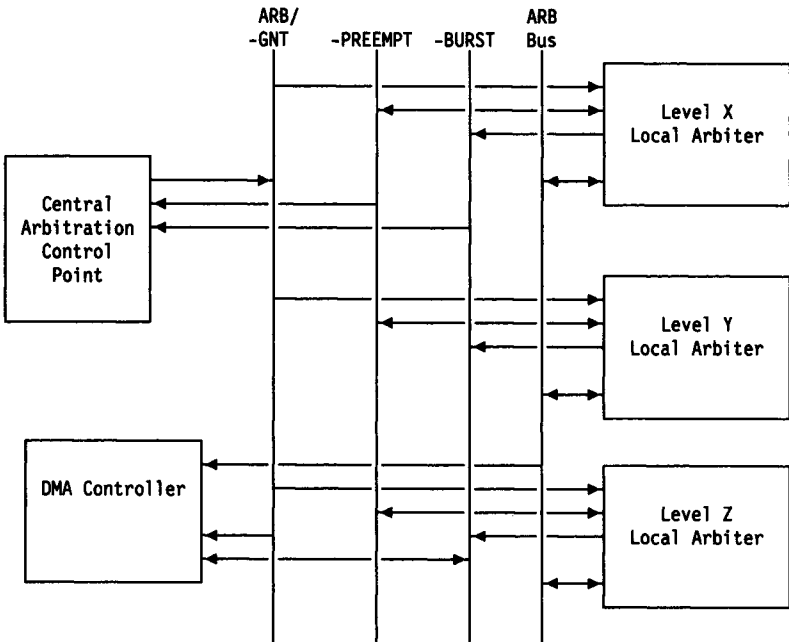


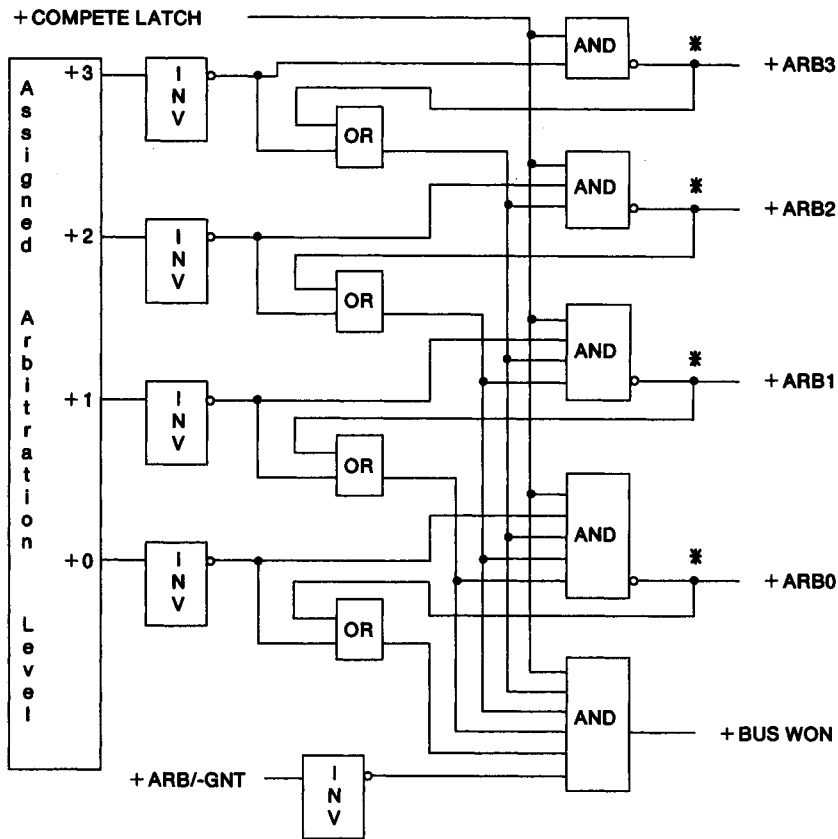
Figure 13. Arbitration Block Diagram

Local Arbiters

Participants requesting the use of the channel implement logic that allows all competing participants to recognize the winner. This logic is known as a local arbiter. An arbitrating participant competes for control of the channel only if it has driven -PREEMPT active before ARB/-GNT goes to the ARB state. A competing local arbiter drives its arbitration level onto the arbitration bus, then compares its arbitration level, on a bit-by-bit basis, with the value appearing on the arbitration bus beginning with the most significant bit, ARB3. If the competing local arbiter detects a mismatch on one of the bits, it immediately stops driving its less-significant bits. If the local arbiter subsequently recognizes a match on that bit, it resumes driving its less-significant bits until another mismatch is detected. The following is an example of an arbitration cycle.

1. Two local arbiters with arbitration levels 1010 and 0101 (hex A and 5) compete for the channel. Both local arbiters drive their arbitration levels on the bus, which now appears as 0000.
2. The first local arbiter (1010) detects a mismatch on ARB3 and stops driving all less-significant arbitration bits (ARB2 and ARB0 in this case).
3. The second local arbiter (0101) detects a mismatch on ARB2 and stops driving its less-significant arbitration bit (ARB1, in this case). The arbitration bus now shows 0111.
4. The second local arbiter now sees a match on ARB2 and resumes driving ARB1 of the arbitration bus.
5. The arbitration bus now shows a value of 0101, and the second local arbiter wins the arbitration cycle.

The following is a simplified example of a local arbiter using combinational logic. If ARB/-GNT goes to the ARB state, +BUS WON is not true.



* Open-Collector Driven

Figure 14. Local Arbiter Example

Burst Transfer

To use burst transfer, a controlling master or a DMA slave activates -BURST after being granted the channel. -BURST is held active until after the start of the last cycle of the burst transfer. The following diagram shows a burst transfer.

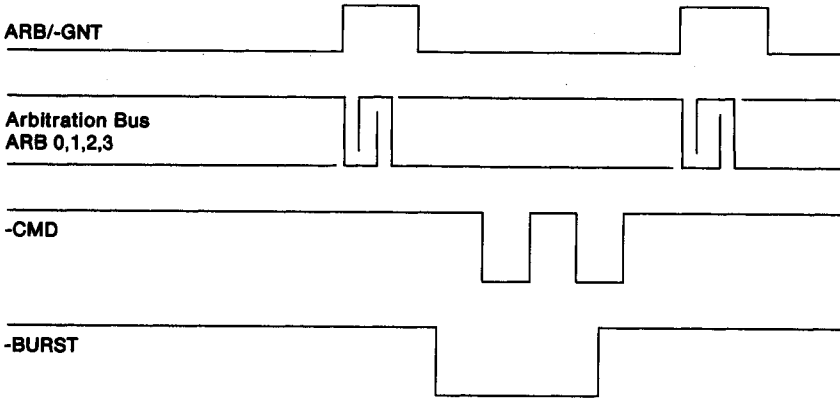


Figure 15. Burst Transfer

Preemption

When an arbitrating participant requires the channel, it activates -PREEMPT. The following diagram shows -PREEMPT occurring during a burst transfer.

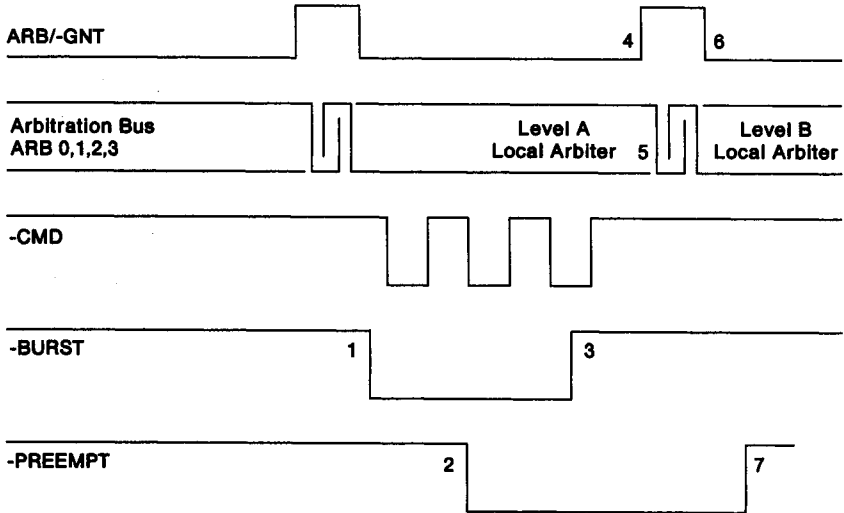


Figure 16. Preempted Burst Transfer

The preempted burst transfer sequence is as follows:

1. Participant A has control of the channel and has driven -BURST active.
2. Participant B drives -PREEMPT active.
3. Participant A, still in control of the channel, recognizes -PREEMPT active and signals the EOT condition within 7.8 microseconds. Participant A does not participate in the next arbitration cycle if its fairness feature is enabled. (See "Fairness Feature and the Inactive State" on page 34.)
4. When the central arbitration control point recognizes the EOT, it drives ARB/-GNT to the ARB state.
5. The arbitration cycle begins.
6. When ARB/-GNT is in the -GNT state, Participant B gains control of the channel.
7. Participant B, the preempting participant, removes -PREEMPT in response to gaining control of the channel.

Arbitrating masters should maximize use of the channel within the 7.8 microsecond limit from -PREEMPT going active. If a master or a DMA slave *does not* release the channel by presenting EOT within 7.8 microseconds after -PREEMPT is driven active, a channel time-out can occur, creating an exception condition.

Fairness Feature and the Inactive State

| All bursting participants must support the fairness feature. Fairness
| forces the bursting participant to equitably share the channel time
| with the other participants. An optional configuration field in
| programmable option select (POS) can be used to disable the
| fairness feature.

When a bursting participant with control of the channel senses an active -PREEMPT, it must release the channel within the specified time (even if the fairness feature is disabled). If its fairness feature is enabled, the participant must enter the *inactive state* unless -PREEMPT was inactive while that participant owned the channel. If -PREEMPT was inactive while that participant had control, it is not required to enter the inactive state but must release the channel. In the inactive state, the bursting participant will not drive -PREEMPT active or participate in arbitration.

To exit the inactive state, the arbitrating participant must wait for
| -PREEMPT to go inactive. The participant then waits for a specified
| time (T42B) after the trailing edge of status (status active to inactive)
| before driving -PREEMPT active.

The following figure shows exiting from inactive state.

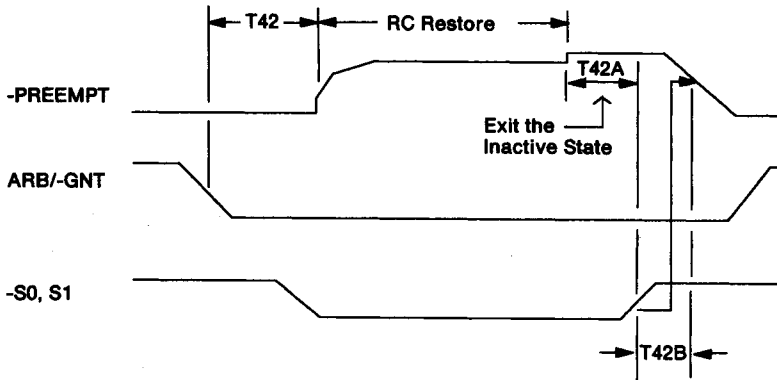


Figure 17. Exiting from the Inactive State

Note: The participant exits the inactive state independently of the state of ARB/-GNT.

Arbitration Timing

The following figures show the signal sequences and the timing specifications of the arbitration procedure.

After ARB/-GNT goes to the -GNT state and until EOT, the data transfer procedures are used to transfer data. See “Basic Transfer” on page 40, “Streaming Data Transfer” on page 58, and “DMA Procedure” on page 84 for the signal sequences and timing specifications.

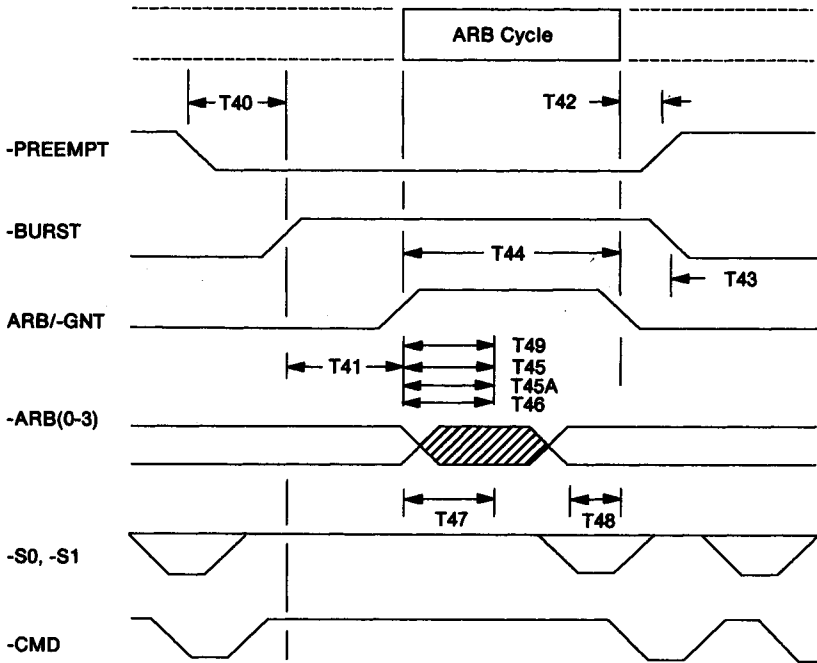


Figure 18. Arbitration Cycle

Timing Parameter	Min/Max	Note	
T40	-PREEMPT active to EOT (channel release)	0 / 7.8 μ s	1
T41	ARB/-GNT in the ARB state from EOT (channel release)	30 / - ns	6
T42	-PREEMPT inactive from ARB/-GNT in -GNT state	0 / 50 ns	8
T42A	-PREEMPT inactive to status inactive	20 / - ns	5
T42B	-PREEMPT active from status inactive (Compete for the channel after the inactive state)	10 / - ns	
T43	-BURST active from ARB/-GNT in -GNT state	- / 50 ns	4
T44	ARB/-GNT in ARB state	100 / - ns	2
T45	Driver turn-on delay from ARB/-GNT in ARB state	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 in ARB state	0 / 50 ns	3
T47	Driver turn-off delay from higher priority line	0 / 50 ns	3
T48	Arbitration bus stable before ARB/-GNT in -GNT state	10 / - ns	
T49	Tri-state drivers from ARB/-GNT in ARB state	- / 50 ns	7

Figure 19. Arbitration Cycle Timing Specifications

Notes:

1. This parameter applies to all masters and DMA slaves driving -BURST active. This parameter ensures that time-critical processes (such as refresh) can gain control of the channel resources. The value of the parameter is a requirement for the adapter designer and not a channel timeout requirement.
2. The value shown applies only to special case implementation involving the central arbitration control point and is provided for pulse width consideration only.
3. T45, T45A, T46 and T47 must be satisfied by the ARB(0-3) drivers of all arbitrating participants.
4. This parameter applies to all DMA slaves and bus masters when they are granted control of the channel.
5. This parameter is met indirectly by meeting T43D and T42 and includes the RC restore time on the bus, assuming no other participant is driving -PREEMPT active during this time. This delay allows participants to detect inactive -PREEMPT and subsequently exit the inactive state.
6. Because no maximum is specified, each non-default master must maintain the EOT on the channel until ARB/-GNT is in the ARB state.
7. This timing applies only to the controlling master as it is releasing control of the channel.
8. The RC restore of -PREEMPT will only occur at this time if no other participant is driving -PREEMPT active. If the system master performs cycles when ARB/-GNT is in the ARB state, -PREEMPT might be released while ARB/-GNT is in the ARB state.

The following are the timing parameters for the first cycle after ARB/-GNT goes to the -GNT state.

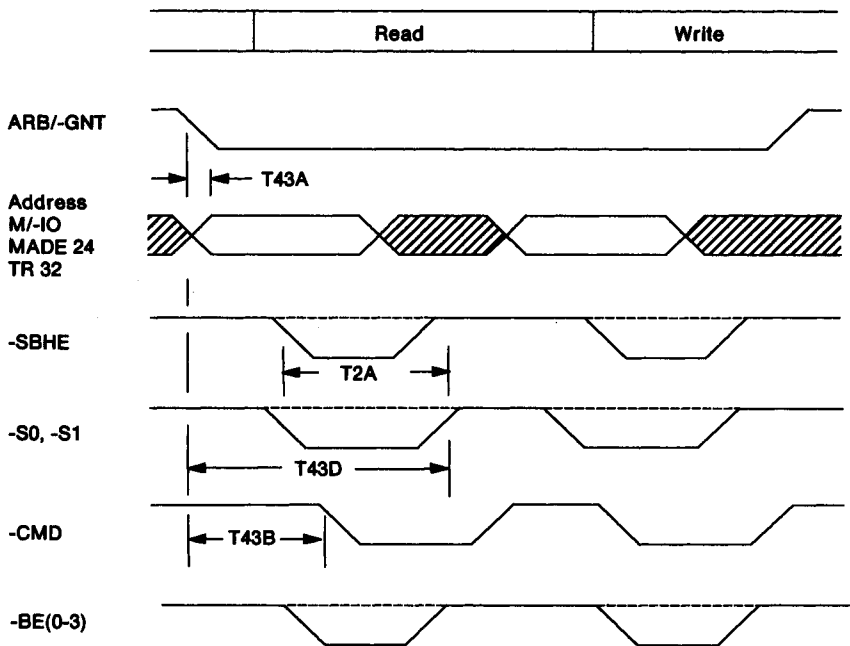


Figure 20. First Cycle After Grant

Timing Parameter	Min/Max	Note	
T2A	Status pulse width of an aborted cycle	85 / - ns	
T43A	Address bus valid from ARB/-GNT in the -GNT state	0 / - ns	
T43B	-CMD active from ARB/-GNT in the -GNT state	115 / - ns	
T43C	DMA controller delay	30 / - ns	1
T43D	Status inactive from ARB/-GNT in the -GNT state	145 / - ns	

Figure 21. First-Cycle-After-Grant Timing Specifications

Notes:

1. A DMA controller must allow a minimum time (T43C) after the grant for a slave to generate an internal acknowledgment that it has been selected. During the first cycle, the DMA controller must allow this additional time before sampling $-\text{DS } 16/32 \text{ RTN}$ and CHRDYRTN if it places an address on the address bus within 30 nanoseconds after grant. However, if the DMA controller drives the address 30 nanoseconds after grant, the additional delay is not needed.

Data Transfer

The data transfer procedures are used to transfer data between a controlling master and the selected slave. The three types of data transfer procedures are:

- Basic Transfer
- Streaming Data Transfer
- DMA Transfer.

Each procedure defines the signal sequences and the signal timing specifications used in the procedure. Data transfer occurs only during the data transfer cycle defined by the procedure. A cycle is one or more data transfers, occurring during the time between the leading edge of -CMD and the trailing edge of -CMD. The data transfer cycle address selects either an I/O-address-space address or a memory-address-space address, under the control of M-/IO.

A controlling master or a DMA slave can drive -BURST, to indicate to the central arbitration control point that one or more data transfer cycles will be used before the EOT occurs. Refer to "Burst Transfer" on page 32 for more information.

This section describes each procedure, the signal sequence for each procedure, and the signal timing specifications.

The system-specific technical manuals contain the signal sequence and timing specifications for matched-memory signals and auxiliary-video signals.

Exception condition handling and reporting descriptions are contained in "Exception Condition Reporting and Handling" on page 118.

Basic Transfer

The three types of basic transfer cycles are:

- Default Cycle
- Synchronous Extended Cycle
- Asynchronous Extended Cycle.

Basic Transfer Procedure

All masters and DMA controllers transfer data with the same control sequence. The following covers the case of a write cycle. The signals appear in the following sequence:

1. The master drives the address bus, MADE 24, M/I-O, -APAREN, APAR(0-3) and -REFRESH (if applicable) valid, beginning the cycle.
2. In response to an unlatched decode of address, MADE 24, and M/I-O, the selected slave returns:
 - -CD SFDBK
 - -CD DS 16 (if the addressed data port is capable of 16-bit operations)
 - -CD DS 16 and -CD DS 32 (if the addressed data port is capable of 32-bit operations)
3. The master drives -S0 active (-S1 inactive).
4. In response to an unlatched decode of address, MADE 24, M/I-O, and status, the selected slave drives CD CHRDI inactive if the cycle is to be extended.
5. The master drives -ADL active. A slave can latch decodes of address, status, and M/I-O at the leading or trailing edge of -ADL. A DMA slave can also be selected by a latched decode of the same signals, using the arbitration level in place of the address.
6. The master drives the write data onto the data bus and, if appropriate, drives -DPAREN active and DPAR(0-3) valid.
7. The master drives -CMD active and -ADL inactive. A slave must latch the decodes of address, status, and M/I-O with the leading edge of -CMD if they were not latched with -ADL.

8. If CD CHRDY has been driven inactive, the cycle is extended until CD CHRDY is driven active. The signal must not be held inactive longer than specified.
9. The status signals become inactive and the address bus can change in preparation for the next cycle.
10. In response to an address change, the slave updates the state of:
 - -CD SFDBK
 - -CD DS 16
 - -CD DS 32.
11. The status signals and M/IO can become valid in preparation for the next cycle.
12. -CMD goes inactive and the selected slave receives the data from the data bus. The data transfer cycle is complete.

Note: The changes to the address bus and status (-s0, -s1) can be overlapped with the preceding cycle to minimize the impact of the memory access time.

The sequence for the basic transfer procedure is as follows.

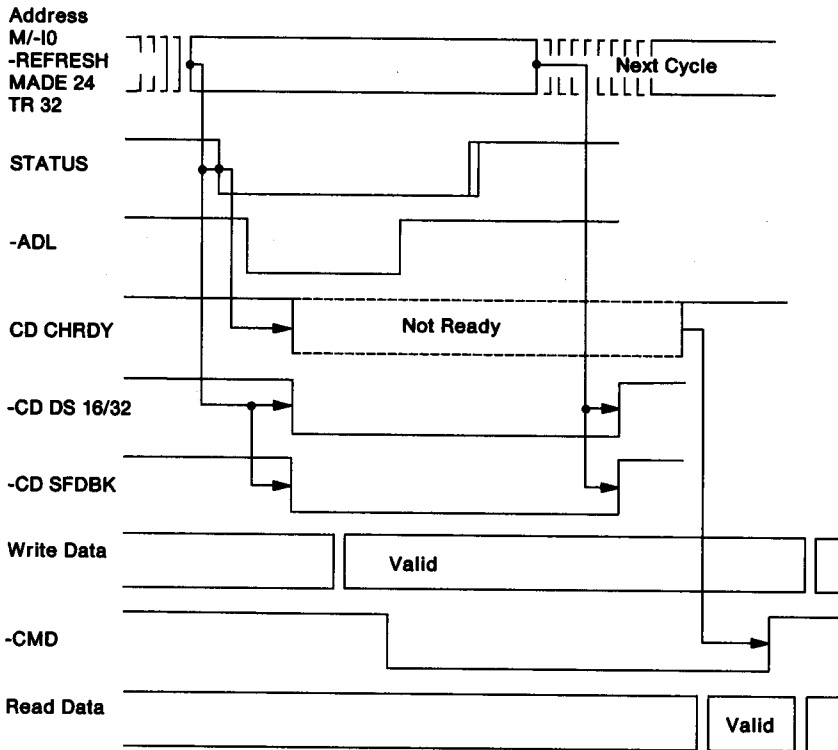


Figure 22. Basic Transfer Procedure - Overview

Whether a default, a synchronous-extended, or an asynchronous-extended cycle is performed depends on how a slave uses CD CHRDY.

- A default cycle occurs when a slave does not drive CD CHRDY inactive.
- A synchronous-extended cycle occurs when a slave drives CD CHRDY inactive, then releases CD CHRDY synchronously within the specified time after the leading edge of -CMD. The slave provides the read data within the specified time from -CMD active.
- An asynchronous-extended cycle occurs when a slave drives the CD CHRDY inactive, then releases CD CHRDY asynchronously.

If a master begins a transfer cycle but must abort that cycle, -ADL and -CMD must *not* be activated, and -S0, -S1 must be activated with a minimum pulse width T2A (see Figure 24 on page 45 for the T2A timing specification). -BURST, if active, is deactivated with status deactivation. Slaves must not latch CD CHRDY inactive until after the appearance of an active -ADL signal, and if the cycle is aborted, must release CD CHRDY with the deactivation of status. The selected slave degrades the data bus and data transfer control signals when it is no longer selected (status and -CMD are inactive).

Default Cycle Timing (200 ns Minimum)

This section provides the specifications for the timing parameters for the I/O and memory default transfer cycle.

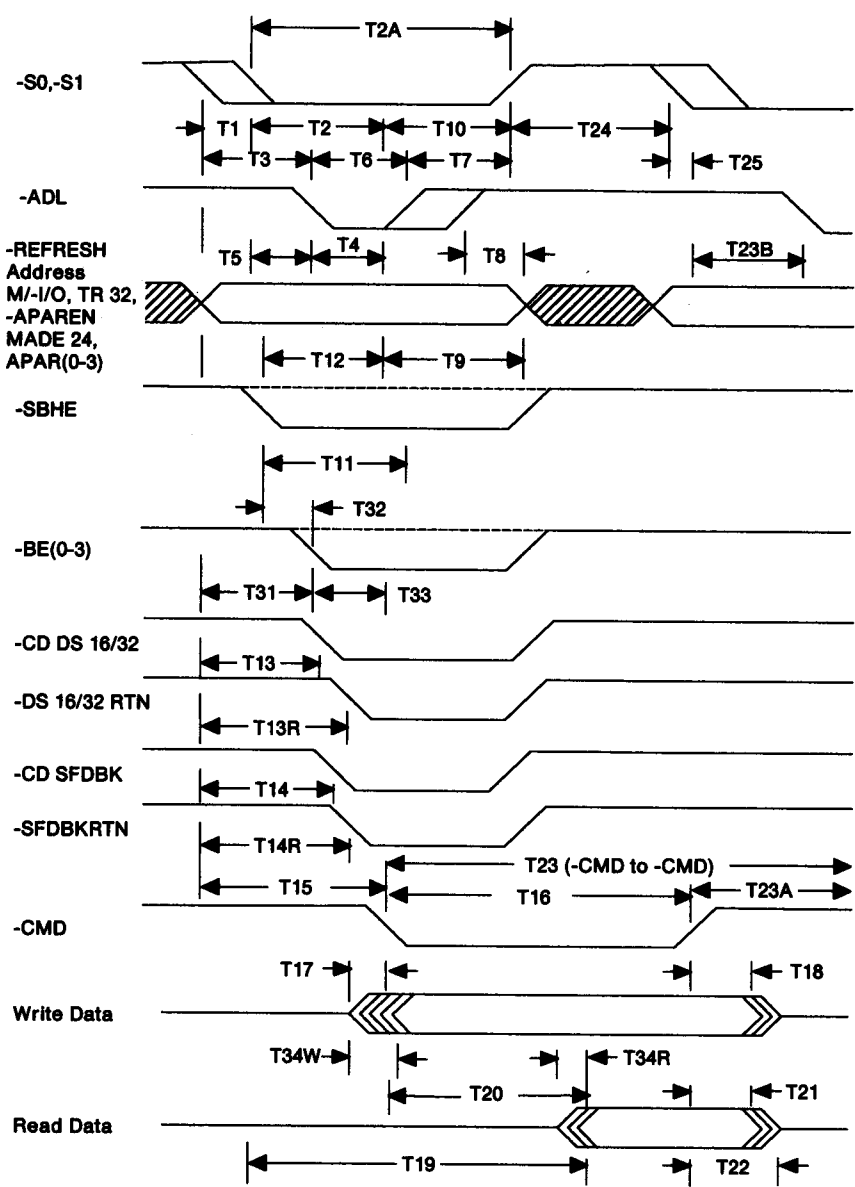


Figure 23. Default Cycle (200 ns Minimum)

	Timing Parameter	Min/Max	Note
T1	Status active from Address, M/-IO, -APAREN, -REFRESH, MADE 24, TR 32, and APAR(0-3) valid	10 / - ns	13
T2	-CMD active from status active	55 / - ns	4
T2A	Status pulse width of an aborted cycle	85 / - ns	10
T3	-ADL active from Address, M/-IO, -APAREN, -REFRESH, MADE 24, TR 32, and APAR(0-3) valid	45 / - ns	
T4	-ADL active to -CMD active	40 / - ns	
T5	-ADL active from status active	12 / - ns	
T6	-ADL pulse width	40 / - ns	
T7	Status hold from -ADL inactive	25 / - ns	4,11
T8	Address, M/-IO, -REFRESH, -SBHE, -BE(0-3), MADE 24, TR 32, APAR(0-3), -APAREN hold from -ADL inactive	25 / - ns	4,13
T9	Address, M/-IO, -REFRESH, -SBHE, -BE(0-3), MADE 24, TR 32, APAR(0-3), -APAREN hold from -CMD active	30 / - ns	2,3,4, 13
T10	Status hold from -CMD active	30 / - ns	4,11
T11	-SBHE setup to -ADL inactive	40 / - ns	4
T12	-SBHE setup to -CMD active	40 / - ns	4
T13	-CD DS 16/32 active from Address Bus and M/-IO valid	- / 55 ns	2,6
T13R	-DS 16/32 RTN active from Address Bus and M/-IO valid	- / 75 ns	2
T14	-CD SFDBK active from Address Bus and M/-IO valid	- / 60 ns	1,2
T14R	-SFDBKRTN active from Address Bus and M/-IO valid	- / 80 ns	2
T15	-CMD active from Address valid	85 / - ns	4
T16	-CMD pulse width	90 / - ns	
T17	Write Data setup to -CMD active	0 / - ns	5,12
T18	Write Data hold from -CMD inactive	30 / - ns	5
T19	Status active to Read Data valid (Access Time)	- / 125 ns	5,8
T20	Read Data valid from -CMD active	- / 60 ns	5
T21	Read Data hold from -CMD inactive	0 / - ns	5
T22	Read data bus tri-state from -CMD inactive	- / 40 ns	5
T23	-CMD active to next -CMD active	190 / - ns	
T23A	-CMD inactive to next -CMD active	80 / - ns	
T23B	-CMD inactive to next -ADL active	40 / - ns	
T24	Next status active from status inactive	30 / - ns	
T25	Next status active to -CMD inactive	- / 20 ns	
T31	-BE(0-3) active from Address valid (32-bit masters only)	- / 40 ns	
T32	-BE(0-3) active from -SBHE, A0, A1 active	- / 30 ns	
T33	-BE(0-3) active to -CMD active	10 / - ns	
T34R	Centralized steering logic Read Data buffer delay	0 / 15 ns	9
T34W	Centralized steering logic Write Data buffer delay	0 / 15 ns	9

Figure 24. Default Cycle (200 ns Minimum) Timing Specifications

Notes:

1. Slaves drive -CD SFDBK when selected by either arbitration or address. The slave does not drive -CD SFDBK active when selected by -CD SETUP.
2. The master must hold the address bus and associated signals valid until -DS 16 RTN, -DS 32 RTN, and -SFDBKRTN are sampled, in addition to meeting T8 and T9. The slave must continue to drive -CD DS 16/32 and -CD SFDBK as long as its address is valid.
3. A master can deactivate status and address concurrently. This can result in changes on the address bus while status is still active at the slave. CD CHRDY can change states during changes on the address bus.
4. It is recommended that slaves use transparent latches to latch information with the leading edge of -CMD, or with the leading or trailing edge of -ADL.
5. When data parity is supported, the implemented data parity signals *must be* active and stable on the bus for the data to be considered valid.
6. -CD DS 16/32 and -CD SFDBK must be driven by unlatched address decodes because the next address can be driven valid in the current cycle.
7. When address parity is supported, the implemented address parity signals *must be* active and stable for the address to be considered valid.
8. T19 is a valid timing parameter *only* when T2 is less than 65 nanoseconds.
9. The centralized steering delay applies when a 16-bit master accesses a 32-bit data port. The slave must compensate for the centralized steering delay, as required.
10. Masters must provide this minimum timing for the central arbitration control point and for inactive state exit. Other participants must not depend on this minimum during an abort cycle. (See "Central Arbitration Control Point" on page 28 for more information.)
11. A bus master must deactivate -S(0,1) as soon as possible after the hold time specified by T7 and T10, and prior to the deactivation of

-CMD. Additional rules apply for extended cycles and streaming data cycles.

- 12. If data parity is used, see T96 timing in Figure 71 on page 123.**
- 13. TR 32 can be held inactive between cycles by a 32-bit master. TR 32 timings are provided to define when it must be valid on the first and last cycle.**

Synchronous-Extended Cycle Timing (300 ns Minimum)

A synchronous-extended cycle occurs when a slave drives CD CHRDY inactive, then releases CD CHRDY synchronously within the specified time after the leading edge of -CMD.

The timing sequence is shown in the following figure.

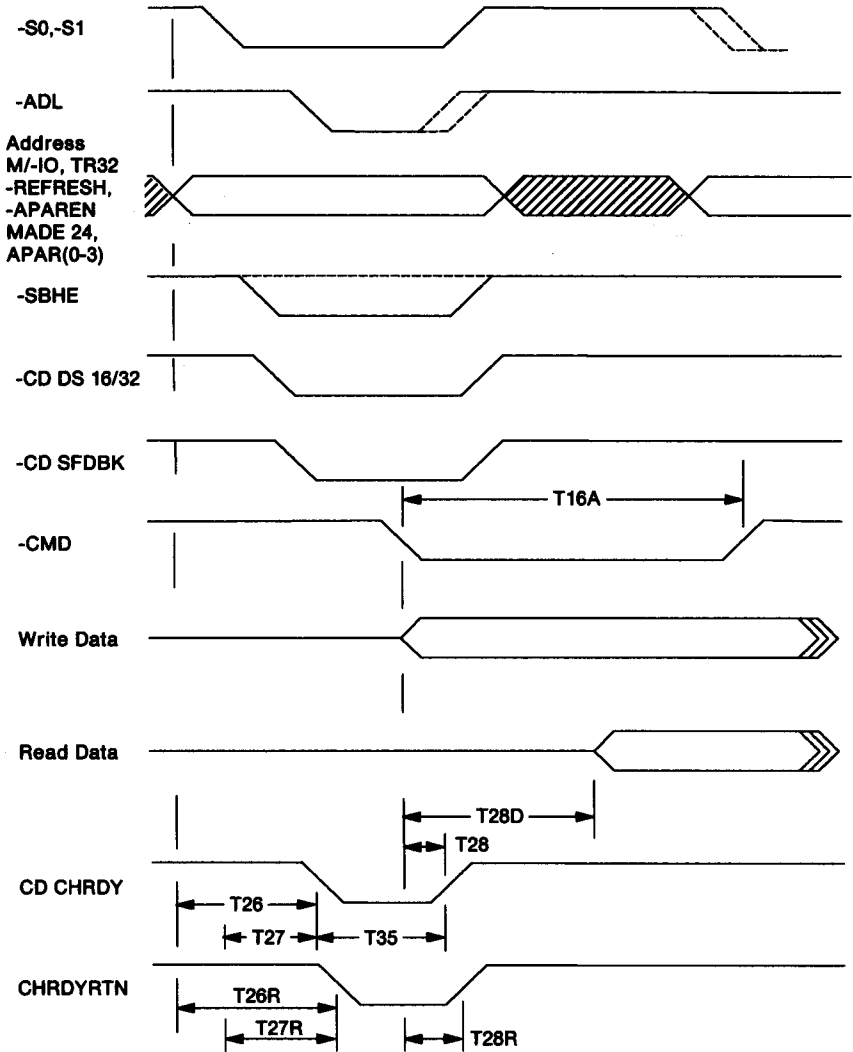


Figure 25. Synchronous-Extended Cycle (300 ns Minimum)

	Timing Parameter	Min/Max	Note
T16A	-CMD pulse width	190 / - ns	
T26	CD CHRDY inactive from Address Bus valid	- / 60 ns	3
T26R	CHRDYRTN inactive from Address Bus valid	- / 80 ns	3
T27	CD CHRDY inactive from status valid and previous -CMD inactive	0 / 30 ns	3,4
T27R	CHRDYRTN inactive from status valid and previous -CMD inactive	0 / 50 ns	3,4
T28	CD CHRDY active from -CMD active	0 / 30 ns	1
T28R	CHRDYRTN active from -CMD active	0 / 50 ns	
T28D	Read Data valid from -CMD active (used with T28)	0 / 150 ns	
T35	CD CHRDY active from CD CHRDY inactive	0 / 3.5 μ s	5

This table shows only the parameters that are additional to those of the default cycle. All other parameters are the same as the default cycle.

Figure 26. Synchronous-Extended Cycle Timing Specifications (300 ns Minimum)

Notes:

1. CD CHRDY is released by a slave performing a 300 nanoseconds extended cycle within the specified time after the leading edge of -CMD.
2. A master can deactivate status and address concurrently. This can result in changes on the address bus while status is still active at the slave. CD CHRDY can change states during changes on the address bus.
3. T27 and T27R apply *only* when T1 is greater than 30 nanoseconds. T26 and T26R applies *only* when T1 is less than or equal to 30 nanoseconds.
4. Masters are responsible for controlling their own bus activity. When a master uses overlapped cycles, it *must* be capable of handling (or ignoring) CD CHRDY from the next slave while terminating the cycle with the current slave. Once the current cycle is terminated, the master must act appropriately to an inactive CD CHRDY signal. Masters must also recognize and handle any added delay in a slaves deactivation of CD CHRDY caused by overlapped cycles.
5. A bus master must deactivate -S0,-S1 as specified in the default cycle, and this deactivation must be independent of CHRDYRTN. Slaves should not depend on the removal of -S0,-S1 for the deactivation of CD CHRDY.

- When address parity is supported, the address parity signals (APAR 0-3) *must* be active and stable for the address to be considered valid.

Asynchronous-Extended Cycle Timing (≥ 300 ns)

An asynchronous-extended cycle occurs when a slave drives CD CHRDY inactive, then releases CD CHRDY asynchronously. The timing sequence is shown in the following figures.

Timing Parameter	Min/Max	Note
T16A -CMD pulse width	190 / - ns	
T26 CD CHRDY inactive from Address Bus valid	- / 60 ns	3
T26R CHRDRYRTN inactive from Address Bus valid	- / 80 ns	3
T27 CD CHRDY inactive from status valid and previous -CMD inactive	0 / 30 ns	3,4
T27R CHRDRYRTN inactive from status valid and previous -CMD inactive	0 / 50 ns	3,4
T29S Read Data from slave valid from CD CHRDY active	- / 60 ns	1
T29M Read Data valid to master from CHRDRYRTN active	- / 60 ns	1
T29A -CMD inactive from CHRDRYRTN active	60 / - ns	
T35 CD CHRDY active from CD CHRDY inactive	0 / 3.5 μ s	5

This table shows only the parameters that are additional to those of the default cycle. All other parameters are the same as the default cycle.

Figure 27. Asynchronous-Extended Cycle Timing Specifications (≥ 300 ns)

Notes:

- CD CHRDY is released asynchronously by a slave performing a ≥ 300 -nanosecond cycle. The slave must present the read data within the time specified after the release of CD CHRDY.
- A master can deactivate status and address concurrently. This can result in changes on the address bus while status is still active at the slave. CD CHRDY can change states during changes on the address bus.
- T27 and T27R apply *only* when T1 is greater than 30 nanoseconds. T26 and T26R apply *only* when T1 is less than or equal to 30 nanoseconds.

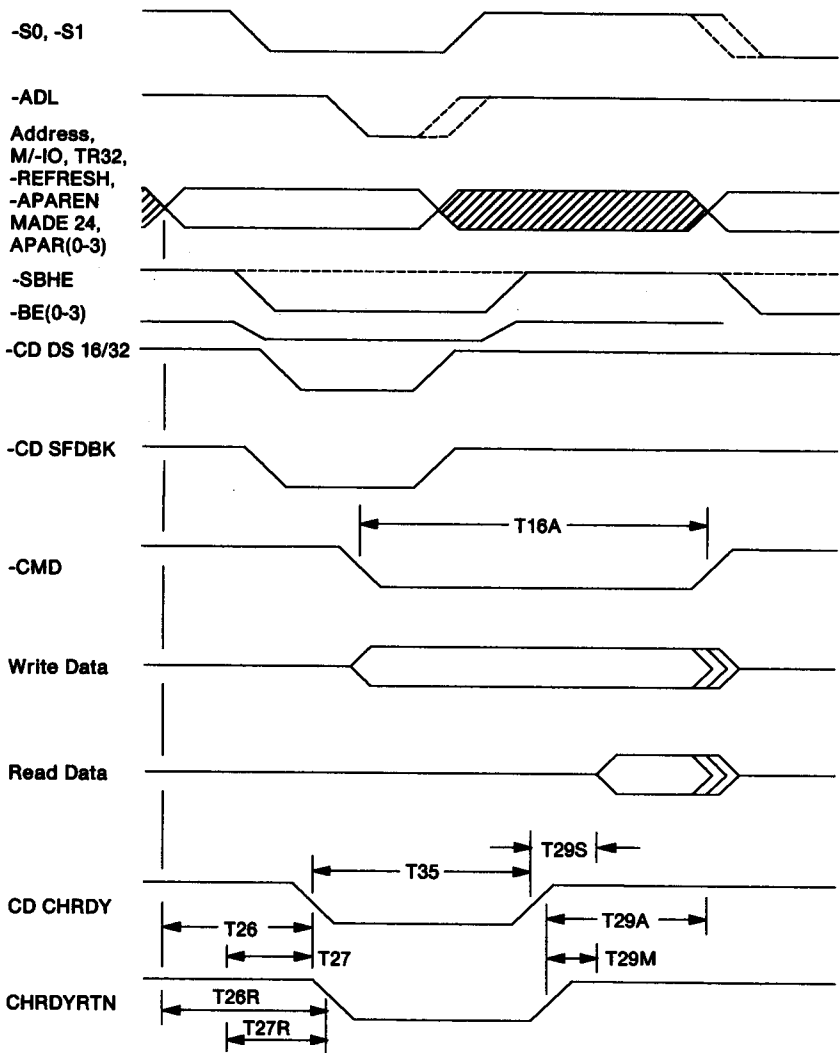


Figure 28. Asynchronous Extended Cycle (≥ 300 ns)

4. Masters are responsible for controlling their own bus activity. When a master uses overlapped cycles, it *must* be capable of handling (or ignoring) CD CHRDY from the next slave while terminating the cycle with the current slave. Once the current cycle is terminated, the master must act appropriately to an inactive CD CHRDY signal. Masters must also recognize and handle the potential added delay in a slave's deactivation of CD CHRDY due to overlapped cycles.
5. A bus master must deactivate -S0, -S1 as specified in the default cycle, and this deactivation must be independent of CHRDYRTN. Slaves should not have dependencies on removal of -S0, -S1 for the deactivation of CD CHRDY.
6. When address parity is supported, the implemented address parity signals (APAR 0-3) *must* be active and stable for the address to be considered valid.

Data Steering

During data transfer procedures, data can be transferred between a 16-bit or 32-bit master, and an 8-, 16-, or 32-bit slave. The following table shows the relationship between the data bus, associated signals, and the number of cycles required for the combinations of data transfer and steering.

		S32/32-BIT DATA PORT -CD DS 32 ACTIVE -CD DS 16 ACTIVE								S16/16-BIT DATA PORT -CD DS 32 INACTIVE -CD DS 16 ACTIVE								S8/8-BIT DATA PORT -CD DS 32 INACTIVE -CD DS 16 INACTIVE							
DATA	SIZE	BE3	BE2	BE1	BE0	D3	D2	D1	D0	SBHE	A1	A0	D3	D2	D1	D0	SBHE	A1	A0	D3	D2	D1	D0		
M32	4 BYTE	0	0	0	0	D	C	B	A	0	0	0	-	B	A	-	0	0	0	-	-	-	-	A	
TR										0	1	0	-	D	C	-	0	1	0	-	-	-	-	B	
																	1	0	0	-	-	-	-	C	
																	1	1	0	-	-	-	-	D	
M32	3 BYTE	1	0	0	0	-	C	B	A	0	0	0	-	B	A	-	0	0	0	-	-	-	-	A	
TR										1	1	0	-	-	C	-	0	1	0	-	-	-	-	B	
																	1	0	0	-	-	-	-	C	
																	1	1	0	-	-	-	-	D	
																	1	0	1	-	-	-	-	B	
																	1	0	0	-	-	-	-	C	
																	1	1	0	-	-	-	-	D	
																	1	1	1	-	-	-	-	A	
																	1	0	1	-	-	-	-	B	
																	1	1	0	-	-	-	-	C	
																	1	1	1	-	-	-	-	D	
M16	2 BYTE	1	1	0	0	-	-	B	A	0	0	0	-	B	A	-	0	0	0	-	-	-	-	A	
TR										0	1	0	-	D	C	-	0	1	0	-	-	-	-	B	
																	1	0	0	-	-	-	-	C	
																	1	1	0	-	-	-	-	D	
																	1	0	1	-	-	-	-	A	
																	1	1	0	-	-	-	-	B	
																	1	0	1	-	-	-	-	C	
																	1	1	1	-	-	-	-	D	
M16	1 BYTE	1	1	1	0	-	-	-	A	1	0	0	-	-	A	-	0	0	0	-	-	-	-	A	
TR										0	0	1	-	B	-	0	1	0	0	-	-	-	-	B	
																	1	0	0	-	-	-	-	C	
																	1	1	0	-	-	-	-	D	
																	1	0	1	-	-	-	-	A	
																	1	1	0	-	-	-	-	B	
																	1	0	1	-	-	-	-	C	
																	1	1	1	-	-	-	-	D	

Figure 29. Steering Control

Notes:

1. Masters are responsible for generating and controlling the extra cycles needed to complete the transfer.
2. If TR 32 = 1, the central translator logic drives -BE(0-3) signals. The central steering logic steers data bus byte D(0-7) to or from data bus byte D(16-23) and data bus byte D(8-15) to or from data bus byte D(24-31).
3. If TR 32 = 0, the 32-bit master (M32) drives -BE(0-3) signals.
4. A -BE0, -BE1, -BE2, or -BE3 signal equal to 0 indicates a valid data byte in the corresponding data bus byte D(0-7), D(8-15), D(16-23), or D(24-31).
5. A 1 denotes the most-positive logic state of the signal. A 0 denotes the least-positive logic state of the signal.
6. Combinations of -SBHE, A1, A0 not shown in the table are not permitted.

Data Steering Control

The following table shows which participant is responsible for data-steering control during basic transfer, DMA, and setup procedures.

	S32	S16	S8
M16	Central Steering	No Steering	M16
M32	No Steering	M32	M32
	M32 - 32-bit Master	S32 - 32-bit Slave	
	M16 - 16-bit Master	S16 - 16-bit Slave	
		S8 - 8-bit Slave	

Figure 30. Steering Responsibility (Basic Transfer, DMA, and Setup Procedures)

The following table shows which participant is responsible for data-steering control during streaming data procedures.

	S32	S16
M16	S32	No Steering
M32	No Steering	M32
	M32 - 32-bit Master	S32 - 32-bit Slave
	M16 - 16-bit Master	S16 - 16-bit Slave

Figure 31. Steering Responsibility (Streaming Data Procedure)

Master Replication of Data Bytes

The following are the requirements for master replication of data bytes during a write operation:

- A 16 or 32-bit master transferring one byte of data on D(8-15) replicates D(8-15) on D(0-7).
- A 32-bit master transferring two bytes of data on D(8-15) and D(16-23) or three bytes of data on D(8-15), D(16-23), and D(24-31) replicates D(8-15) on D(0-7).
- A 32-bit master transferring one byte of data on D(16-23) replicates D(16-23) on D(0-7).
- A 32-bit master transferring one byte of data on D(24-31) replicates D(24-31) on D(0-7) and D(8-15).
- A 32-bit master transferring two bytes of data on D(16-23) and D(24-31) replicates D(16-23) on D(0-7) and replicates D(24-31) on D(8-15).

Channel Support

The 32-bit data bus requires unique system logic to permit 16-bit masters to communicate with slaves with 32-bit data ports.

Address Bus Translator

A 32-bit slave uses signals -BE(0-3) as part of its address instead of A0, A1, and -SBHE. A 16-bit master does not provide -BE(0-3); the central translation logic generates them when a 16-bit master has control of the channel.

Central Data Steering

A 32-bit slave writes data to and reads data from data bits 0 through 31. A 16-bit master does not use data bits 16 through 31; the central steering logic must cross data over from the low 16 data lines (D₀ through D₁₅) to the high data lines (D₁₆ through D₃₁) and back at the appropriate times. The 32-bit slave must compensate for the added delay.

TR 32

TR 32 is driven inactive by 32-bit masters only. When TR 32 is active, it is used by:

- The central translator logic to drive -BE(0-3)
- The central steering logic to perform data steering.

When TR 32 is active, 32-bit slaves can use it to detect that the controlling master is not a 32-bit master and compensate for additional delay attributable to the central steering logic.

Central Steering Logic

Central steering logic uses A1, -SBHE, -S(0,1), -CD DS 16, and -CD DS 32 to steer data in support of 16-bit masters communicating with 32-bit slaves.

Central Translator Logic

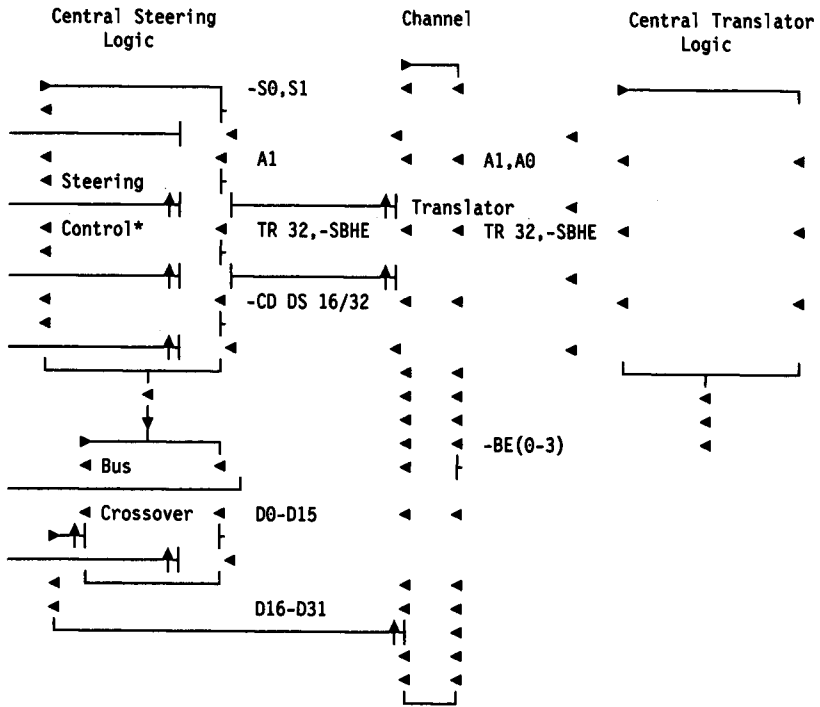
Central translator logic translates A0, A1, and -SBHE to -BE(0-3), when TR 32 is active to support transfers between 16-bit masters and 32-bit slaves.

-BE(0-3)

Signals -BE(0-3) are:

- Driven by a 32-bit controlling master.
- Created by the central translator logic when a 16-bit master is the controlling master.
- Used by 32-bit slaves only.

The following block diagram shows the implementation of data steering.



* For 16-bit devices to 32-bit devices

Figure 32. Data Bus Steering Implementation

Streaming Data Transfer

The streaming data procedure provides performance improvements over basic transfer procedures for block transfers, and supports data-transfer rates of up to 80MB per second. It provides for the transfer of a data block by using a single address followed by multiple 16-, 32-, or 64-bit data transfers within a single streaming data cycle. The data transfers are clock-synchronous and incorporate automatic speed matching between the controlling master and slave.

The streaming data procedure can be used for high speed data transfer between a controlling master and the selected slave. The streaming data procedure is transparent to devices that are not selected. Streaming data participants **must** support the basic transfer procedure to operate with nonstreaming-data participants.

Special rules apply to streaming data devices to ensure compatibility with nonstreaming-data participants. These include:

- A streaming-data-enable bit must be supported in the adapter POS space.

Note: Streaming data participants must support a fully-functional operation with nonstreaming-data participants, independent of the state of the Streaming Data Enable field.

- After CHRESET, the default state of the streaming-data-enable bit is disabled.
- Both 16- and 32-bit streaming data transfers must start on four-byte address boundaries.
- A 64-bit streaming data transfer must be aligned on an eight-byte address boundary.

Data is validated by using CD CHRDY to indicate ready or not ready, so that the slave has the ability to pace the data transfer for 16- or 32-bit transfers. Either the master or slave can terminate the cycle at any time.

Implementing and using streaming data transfer is optional to masters. Implementing and indicating streaming data transfer capabilities is optional to slaves.

Streaming Data Procedure (16- or 32-Bit)

A 16- or 32-bit streaming data transfer cycle is initiated as either a 16- or 32-bit basic data transfer cycle, and all rules associated with those procedures apply. The following is an example of a 16- or 32-bit streaming data write cycle. The signals appear in the following sequence:

1. The master drives the address bus, MADE 24, M/-IO, -APAREN, and APAR(0-3) valid to begin the cycle.
2. In response to an unlatched decode of address, MADE 24, and M/-IO, the selected slave returns:
 - -CD SFDBK
 - -CD DS 16 (if the addressed data port is capable of 16-bit operations)
 - -CD DS 16 and -CD DS 32 (if the addressed data port is capable of 32-bit operations).
3. The master drives -S0 active (-S1 inactive).
4. In response to an unlatched decode of address, MADE 24, M/-IO, and status, the selected slave drives CD CHRDY inactive, if the cycle is to be extended.
5. A 32-bit master (or the central translator logic for a 16-bit master in a 32-bit system) drives the byte enable signals. -BE(0-3) is driven to binary 0011 by the central translator logic, or -BE(0-3) is driven to binary 0000 by a 32-bit master.

Note: Because 32-bit operations are aligned on four-byte boundaries, A0, A1, and -SBHE equal a binary 000.

6. The master drives -ADL valid. A slave can latch decodes of address, status, and M/-IO at the leading or trailing edge of -ADL.
7. The slave drives -SDR(0,1) active.

Note: When the slave **does not** drive -SDR(0,1) active (indicating that it is defaulting to basic transfer procedures), the controlling master will drive -S(0,1) inactive after the hold time specified by timing parameter T7 and T10 (see Figure 24 on page 45).

8. The master drives the write data onto the bus and, if supported, drives -DPAREN active and DPAR(0-3) valid.

9. The master drives -CMD active and -ADL inactive. A slave must latch the decodes of address, status, and M/-IO with the leading edge of -CMD if they were not latched with -ADL.
10. If CD CHRDY has been driven inactive, the cycle is extended until CD CHRDY is driven active. CD CHRDY must not be held inactive longer than specified.
11. The controlling master indicates the ability to perform streaming data by starting -SD STROBE.

Note: If CHRDYRTN is active, -SD STROBE can be activated concurrently with -CMD. -SD STROBE is used by both the controlling master and slave to transfer data, with data being clocked on and off the bus on the falling clock edge and clocked off the bus on the next falling clock edge. The operation proceeds with new data being placed on the bus every time -SD STROBE makes a high-to-low transition.

12. The master drives status inactive.
13. The slave drives -SDR(0,1) inactive.
14. The address bus can change in preparation for the next cycle.
15. In response to an address change, the slave updates the state of:
 - -CD SFDBK
 - -CD DS 16
 - -CD DS 32.
16. The status signals and M/-IO can become valid in preparation for the next cycle.
17. The master stops driving -SD STROBE.
18. On the last transfer, the controlling master drives -CMD inactive, and the selected slave gates the data off the bus. The transfer cycle is complete.

Streaming Data Procedure (64-Bit)

A 64-bit streaming data transfer cycle is initiated as a 32-bit basic transfer cycle, and all the rules associated with this procedure apply. The following is an example of a 64-bit streaming data write cycle. The signals appear in the following sequence:

1. The master drives the address bus, MADE24, M/-IO, -APAREN, and APAR(0-3) valid to begin the cycle.
2. In response to an unlatched decode of address, MADE24, and M/-IO, the selected slave returns:
 - -CD SFDBK
 - -CD DS 16
 - -CD DS 32.

3. The master drives -S0 active (-S1 inactive).

4. In response to an unlatched decode of address, MADE24, M/-IO, and status, the selected slave drives CD CHRDY inactive, if the cycle is to be extended.

5. A 32-bit master drives -BE(0-3) to binary 0000 (64-bit streaming data transfers are not performed by 16-bit masters).

Note: Because 64-bit operations are aligned on an eight-byte boundary, A0, A1, and -SBHE equal binary 000.

6. The master drives -ADL active. A slave can latch decodes of address, status, and M/-IO at the leading or trailing edge of -ADL.

7. The slave drives -SDR(0,1) to indicate its transfer speed capability and drives -MSDR active to indicate 64-bit capability.

Note: When the slave does not drive -SDR(0,1), and -MSDR active (indicating that it is defaulting to a basic transfer procedure), the controlling master drives -S(0,1) inactive after the hold time specified by timing parameters T7 and T10, and prior to driving -CMD active (see Figure 24 on page 45.)

8. The master drives the write data onto the data bus and, if supported, drives -DPAREN active and DPAR(0-3) valid.

9. The master drives -CMD active and -ADL inactive. A slave must latch the decodes of address, status, and M/-IO with the leading edge of -CMD if they were not latched with -ADL.

10. If CD CHRDY has been driven inactive, the cycle is extended until CD CHRDY is driven active. CD CHRDY must not be held inactive longer than specified.
11. The controlling master indicates the ability to perform streaming data transfers by activating -SD STROBE. The 64-bit streaming data transfer capability is indicated by driving -BE(0-3) inactive, driving the four most-significant bytes of the eight byte transfer on the address bus, and, if -DPAREN is active, driving APAR(0-3) valid. -APAREN is driven inactive.

Note: -SD STROBE is used by both the controlling master and the slave to transfer data; data is clocked on and off the address bus and the data bus on the falling clock edge and clocked off the address bus and the data bus on the next falling clock edge. The operation proceeds with the new data being placed on the address bus and the data bus each time -SD STROBE makes a high-to-low transition.
12. Both the address bus and data bus are used for transferring data for the remainder of the cycle.
13. The status signals are driven inactive on the last high-to-low transition of -SD STROBE. The slave drives -SDR(0,1) inactive.
14. In response to data on the address bus, other slaves can drive -CD SFDBK, -CD DS 16, -CD DS 32, and CD CHRDY valid during the data transfer. The master should not monitor the state of these signals during the data transfer.
15. On the last transfer, the controlling master drives -CMD inactive and the slave gates the data off the address bus and the data bus. The cycle is complete.

Streaming Data Transfer Rates

The amount of data transferred per -SD STROBE period is constant during streaming data transfer. The number of bytes transferred in one transfer cycle are listed below.

Streaming Data Master Size	16-Bit Slave	32-Bit Slave	64-Bit Slave
16-Bit	2	2	2
32-Bit	2	4	4
64-Bit	2	4	8

Figure 33. Streaming Data Transfer Rates

Note: When a slave requests a streaming data transfer, it must be prepared to support a minimum of two transfers. When a master initiates a streaming data transfer, it must be prepared to support a minimum of one transfer.

Address Bus Management

During streaming data transfers, the address on the address bus is neither incremented nor decremented. The controlling master and the slave must provide memory-address-space address management and I/O-address-space address management as follows:

- Memory-address-space addresses

The controlling master must keep a count of the number of bytes transferred, so that if early termination of the block transfer occurs, the master can assert the correct address (original address plus the number of bytes transferred) at the start of the next data transfer procedure.

The slave must manage the address, to ensure that the data being clocked off the bus is gated into the correct memory addresses.

- I/O-address-space addresses

If early termination of the block transfer occurs, the master must present the same (original) address on the address bus when restarting the streaming data procedure.

The slave must accommodate the streaming data transfer to the same I/O address.

During 64-bit streaming data transfers (after the address is removed from the address bus), the controlling master must ignore CHRDRYRTN, -DS 16 RTN, -DS 32 RTN, and -SFDBKRTN. Slaves (except the selected slave) are decoding the address bus and responding as if selected.

Address Boundary Alignment

If the starting address is not a four-byte address boundary and a 16- or 32-bit streaming data procedure is going to be used, then the basic transfer procedure must be used to transfer data until the address has the correct boundary alignment. The controlling master must execute the basic transfer procedure to obtain address boundary alignment.

A slave can indicate streaming data capability, by driving -SDR(0,1) active, without checking address boundary alignment. Then the controlling master and the slave can use the streaming data procedure. The basic transfer procedure might also be required to complete the data transfer, even if the initial data address was a four-byte address boundary.

If the starting address is not on an eight-byte address boundary and a 64-bit streaming transfer procedure is going to be used basic transfer cycles are used to transfer data until the address has a correct boundary alignment. The basic transfer procedure might also be needed to complete the data transfer, even if the initial data address was a eight-byte address boundary.

The following is an example of a basic transfer procedure occurring before and after a 16- or 32-bit streaming data procedure. The controlling master should drive -BURST active to indicate that one or more consecutive data transfer cycles will occur. Figure 34 on page 65 shows the data block to be transferred.

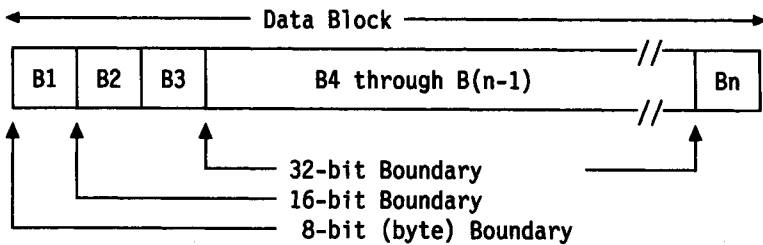


Figure 34. Data Block Example

In the above example, the data transfer procedures used to achieve address boundary alignment between a controlling master and the selected slave are:

- Byte 1 (B1) is transferred using a basic transfer procedure.
- Bytes 2 (B2) and 3 (B3) are transferred using a basic transfer procedure.

The data address is now on a 32-bit boundary.

- Bytes B4 through B(n-1) are transferred using the streaming data procedure. Each period of -SD STROBE transfers two or four bytes of data. The streaming data procedure continues until all data, except byte Bn, is transferred.
- The streaming data procedure is terminated.
- Byte Bn, the last byte is transferred using a basic transfer procedure.
- The session between the controlling master and the slave is terminated.

Streaming Data Cycles

A master can perform streaming data transfers at 10 MHz when $\text{-SDR}(0,1)$ is a binary 00, 01, or 10, and can perform the transfer at less than the requested width. $\text{-SDR}(1)$ is reserved and slaves should not activate it. The valid streaming data signal combinations from the selected slave are shown in the following table.

-SDR 0 1	-CD DS 16/32	-MSDR	Type of Cycle Requested by Slave
0 0	X X	X	Reserved
0 1	0 0	0	64-bit Streaming Data Transfer
0 1	0 0	1	32-bit Streaming Data Transfer
0 1	0 1	1	16-bit Streaming Data Transfer
1 0	X X	X	Reserved
1 1	X X	X	Basic Transfer

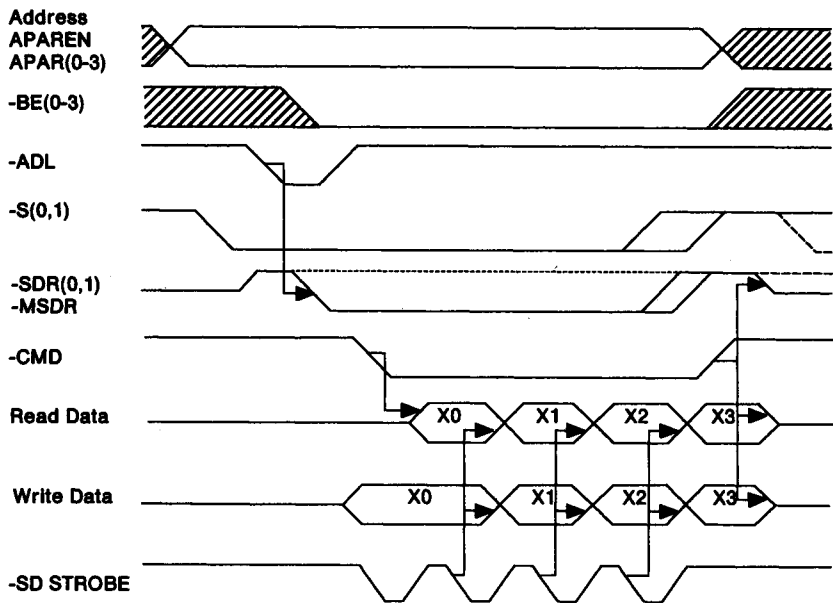
Figure 35. Streaming Data Signal Combinations

Both 16- and 32-bit masters can use streaming data procedures with either 16- or 32-bit slaves. However, because the central translator logic does not perform data steering during a streaming data transfer, a 32-bit slave must perform the steering when streaming with a 16-bit master. (The 32-bit master does the steering for all data transfers with slaves having a smaller data port.)

“Streaming Data Procedure (16- or 32-Bit)” on page 59 and “Streaming Data Procedure (64-Bit)” on page 61 contain the general signal sequence overview.

The following describe how:

- Masters and slaves select streaming data or basic transfer procedures, and the number of bits in each transfer.
- Deferred cycles are performed.
- Data is paced during streaming data procedures.
- Streaming data can be terminated by either the master or the slave.



X0, X1, X2, and X3 represent a data transfer of 16 or 32 bits.

Figure 36. 16- and 32-Bit Streaming Data Cycle Overview

64-Bit Streaming Data Cycles

The 64-bit streaming data transfer can only be used between 64-bit streaming masters and 64-bit streaming slaves. An example of a 64-bit cycle is shown in Figure 37. The 64-bit streaming transfer is similar to the streaming data transfer shown in Figure 36 on page 67, but uses both the data bus and the address bus to achieve the 64-bit data transfer width.

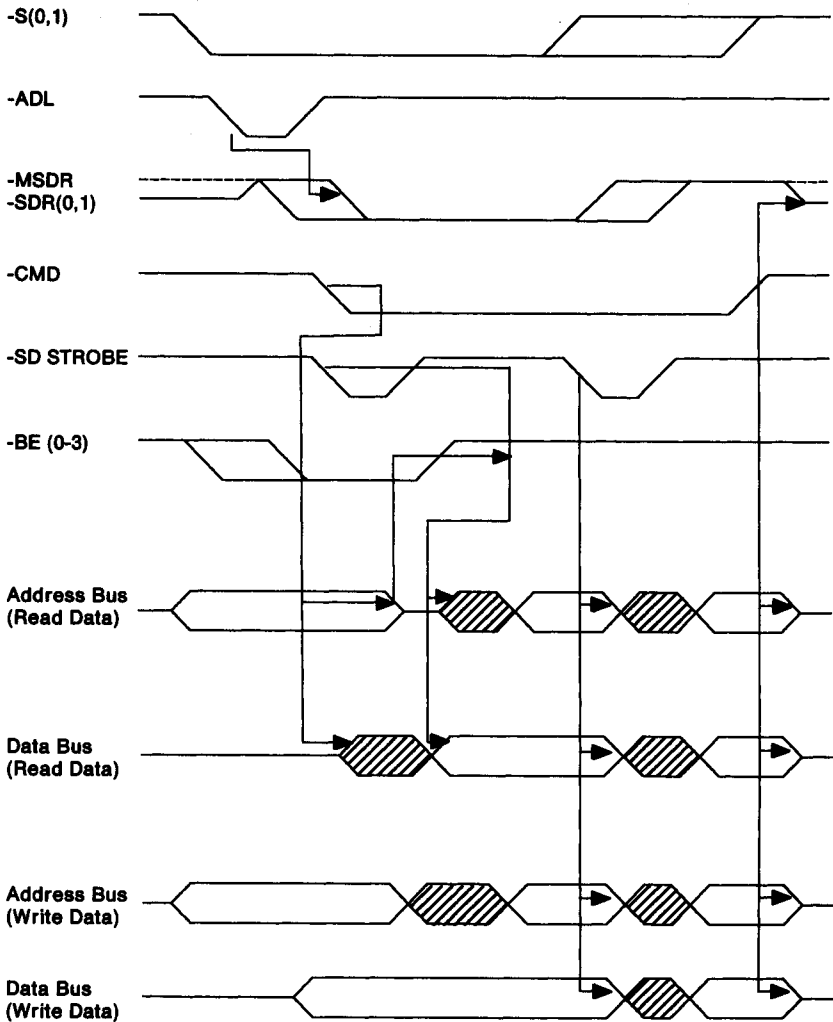


Figure 37. 64-Bit Streaming Data Cycle Overview

The 64-bit cycle begins as a 32-bit basic transfer cycle. The selected slave responds by driving $-CD DS 16$, $-CD DS 32$, $-SDR(0,1)$ and $-MSDR$ valid to indicate that it supports 64-bit streaming operations. $-BE(0-3)$ are driven inactive by the master to indicate a 64-bit streaming-data procedure will be used. During a transfer to the slave, the master starts $-SD STROBE$ and gates the data onto the data and address buses. During a transfer to the master, the master tri-states the address bus after driving $-CMD$ active, and, after the trailing edge of $-BE(0-3)$, the slave gates the data onto the data and address buses. The least significant byte is $D(0-7)$ and the most significant byte is $A(24-31)$.

The 64-bit streaming transfer then proceeds like a 32-bit streaming data transfer.

If $-DPAREN$ has been driven active, the parity bits for data bytes zero through three are contained in $DPAR(0-3)$, the parity bits for data bytes four through seven are contained in $APAR(0-3)$.

Deferred start of the 64-bit streaming transfer is supported, but data pacing is not available during 64-bit data streaming transfers.

Starting the Cycle

Streaming masters can drive **-SD STROBE** active at the same time as **-CMD** to minimize the overhead in starting the streaming data cycle. The **T70** time plus propagation delay for the signals to and from the slave can delay the sample point of **-SDR(0,1)** and **-MSDR** until after **-CMD** and **-SD STROBE** have been activated. Therefore, if a slave indicates it is capable of performing a streaming data transfer, the master must perform at least one streaming data transfer. Requiring one streaming data transfer prevents the slave from confusing a 32-bit basic transfer with a 64-bit streaming data transfer.

Deferring the Start of the Cycle

During a streaming data transfer, the slave may need more time at the start of the cycle to prepare for the transfer. The slave can defer the start of the transfer by driving **CD CHRDY** inactive. The controlling master delays the start of **-SD STROBE** until after **CHRDYRTN** goes active.

If **-SDR(0,1)** is not active within the time specified (**T70A**), the controlling master continues the cycle using basic transfer procedures. All specifications for the basic transfer procedure apply.

If **-SDR(0,1)** is active, after **CHRDYRTN** becomes active, the streaming data procedure can be invoked as shown in Figure 38 on page 71.

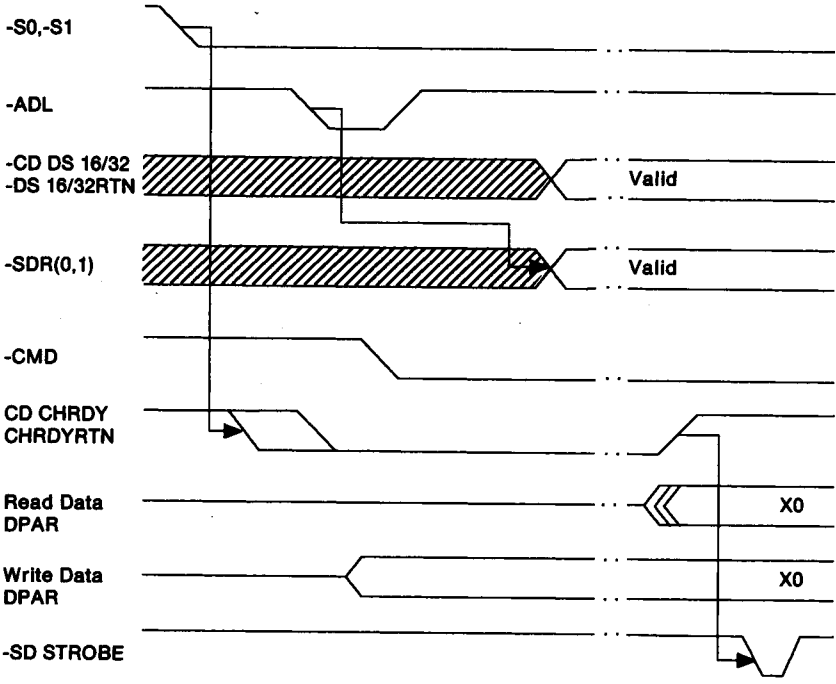


Figure 38. Streaming Data Deferred Start Overview

Pacing the Data

After a 16- or 32-bit streaming data cycle is started, CD CHRDY and CHRDYRTN pace the data transfer, allowing slaves to introduce momentary pauses in the data transfer. CD CHRDY is clocked simultaneously with data and makes state transitions following the high-to-low transition of -SD STROBE. On each falling edge, the controlling master must inspect the state of CHRDYRTN. When CHRDYRTN is inactive during a read operation, it indicates that the slave did not have valid data in this clock period and that the transfer must be repeated. When CHRDYRTN is inactive during a write operation, it indicates the slave did not accept the data and that the transfer must be repeated.

Note: Data pacing by the slave is not supported during 64-bit streaming data cycles.

An example of data pacing using CD CHRDY is shown in Figure 39 on page 73. The example shows two idle (CD CHRDY inactive) clock periods, one in the middle of the data transfer, and one at the end. If the slave has driven CD CHRDY inactive, it must not deactivate -SDR(0,1).

When -MSDR is active, the slave cannot pace the transfer. The slave can use CD CHRDY only to delay the initiation of 64-bit transfers. Slaves that cannot meet the timing specifications for the 64-bit transfer must terminate the streaming data cycle. After -SD STROBE is being driven, the master ignores CHRDYRTN.

In this example, as the controlling source of CD CHRDY, the slave internally clocks its bus interface registers with the inversion of -SD STROBE. At the end of the cycle, the bus interface registers are clocked when -CMD goes high. The internal interface clock of the controlling master is conditional on CHRDYRTN being active.

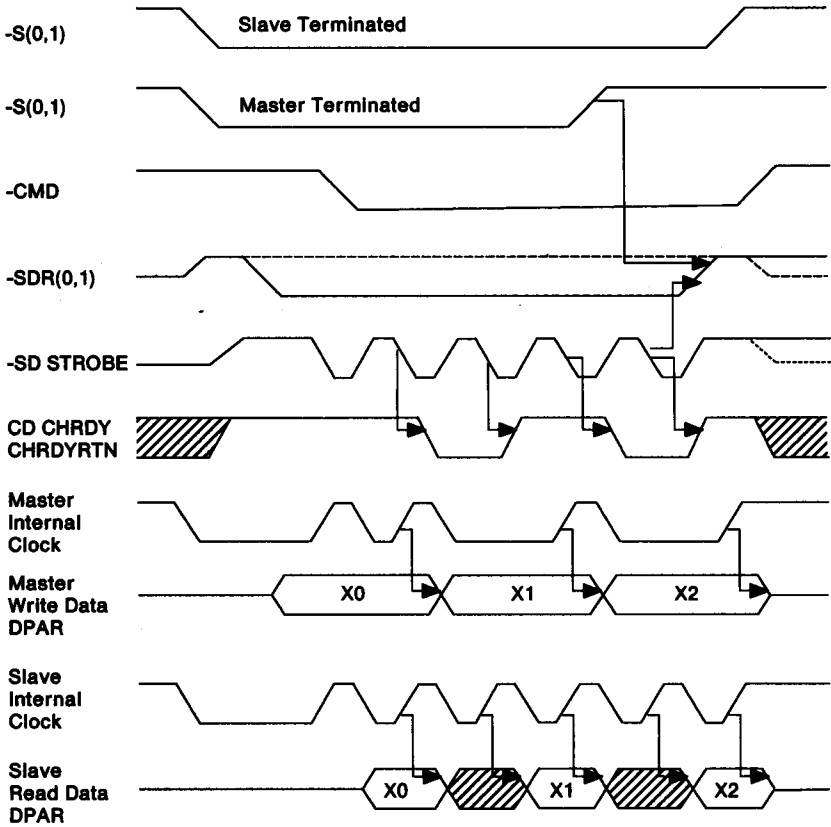


Figure 39. CD CHRDY and Data Pacing

To perform data transfer pacing, streaming masters can introduce momentary pauses in the data transfer by extending -SD STROBE. Data is transferred on the falling edge of -SD STROBE. The strobe period can be extended at either the high or low level, or both. This is done asynchronously to the edges of -SD STROBE.

Whenever the slave is not-ready (CD CHRDY inactive), the master must continue to pulse -SD STROBE. The actual data transfer will occur on the falling edge of -SD STROBE once the slave becomes ready by driving CD CHRDY active.

Terminating the Cycle

Either the controlling master or the slave can terminate the streaming data procedure. If -PREEMPT has become active, the controlling master terminates the streaming data procedure within 7.8 microseconds and releases the channel at EOT. If the fairness feature is active, the preempted master enters the inactive state. See "Fairness Feature and the Inactive State" on page 34 for more information.

Figure 40 shows streaming data cycle termination by the controlling master. Figure 41 on page 75 shows streaming data cycle termination by the controlling master when the slave is not ready. Figure 42 on page 76 shows streaming data cycle termination by a streaming slave.

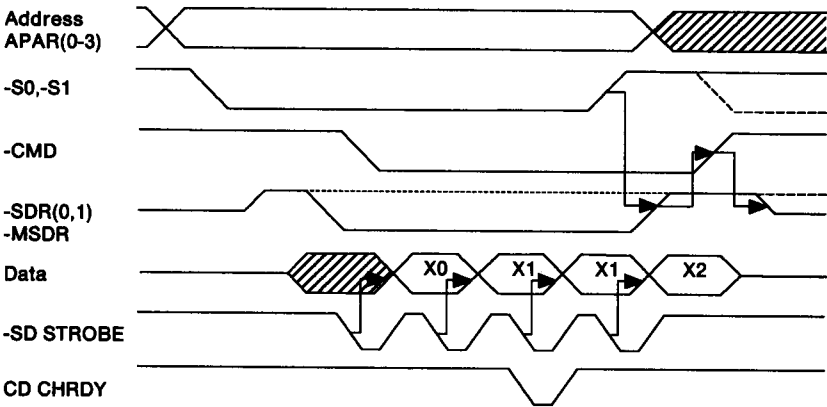


Figure 40. Master-Terminated Streaming Data Cycle

The controlling master can terminate a streaming data cycle by deactivating -S(0,1) on the data transfer before the desired stopping point. This can be done concurrently with the high-to-low transition of -SD STROBE. If -SDR(0,1) is inactive, the controlling master drives -CMD inactive at the point where the next -SD STROBE would have been. The slave **must not** drive -SDR(0,1) inactive while driving CD CHRDY inactive during 16- or 32-bit transfers. If -SDR(0,1) is active and CD CHRDY is inactive, it indicates that the slave is not ready, and the controlling master cannot terminate the cycle.

Figure 41 shows streaming data cycle termination by the master, when the slave is not ready. The master continues to pulse -SD STROBE until it detects the deactivation of -SDR(0,1), which implies that CD CHRDY has been activated. Following the slaves deactivation of -SDR(0,1), the master proceeds to terminate the streaming data cycle. The slave must drive -SDR(0,1), according to T71, on the last -SD STROBE falling edge, similar to a slave-terminated cycle.

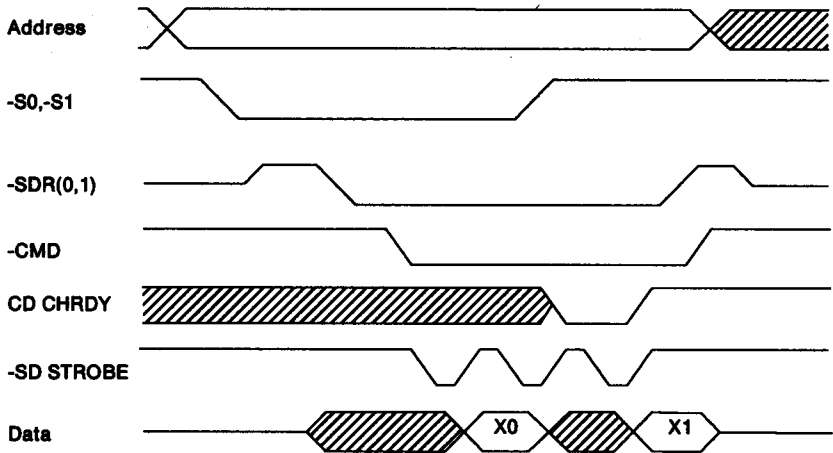


Figure 41. Master-Terminated Streaming Data Cycle (Slave Not Ready)

Figure 42 shows how a streaming data cycle can be terminated by a slave. However, if the master has not transferred all of its data, the master can immediately initiate an additional data transfer procedure.

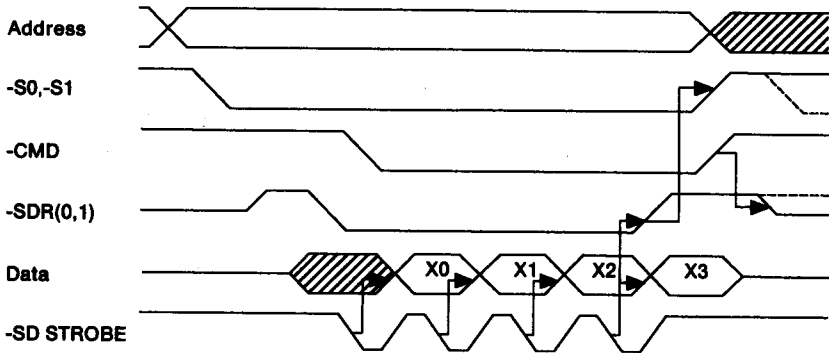


Figure 42. Slave-Terminated Streaming Data Cycle

The slave can terminate a streaming data cycle by driving $\text{-SDR}(0,1)$ inactive following the last desired high-to-low transition of the -SD STROBE . The slave must not drive $\text{-SDR}(0,1)$ inactive while CD CHRDY is inactive. The controlling master stops -SD STROBE and deactivates -S0 , -S1 and -CMD at the point where the next -SD STROBE would have been.

A slave that does not drive CD CHRDY inactive (in response to slave selection) must not drive $\text{-SDR}(0,1)$ or -MSDR inactive with the intention of executing a slave-terminated single-cycle operation. This ensures that the master has adequate time to sample the state of $\text{-SDR}(0,1)$ before it is deactivated. The slave must be able to execute **two or more** data transfers with the master. The slave can sample TR32 to determine if two or four bytes will be transferred with each -SD STROBE . A master can perform a single streaming-data cycle operation. A master that performs a single streaming-data cycle must make sure that it meets the basic timing requirements (T10) as well as all master-terminated streaming-data requirements, including T71A.

When -CMD is driven inactive, the slave tri-states -SDR(0,1), -MSDR and, if performing a read operation, the data bus. The termination sequence results in restoring the streaming data signals to their inactive state prior to their being tri-stated, eliminating uncertainties about the state of these signals in succeeding channel cycles.

Streaming Data Timing

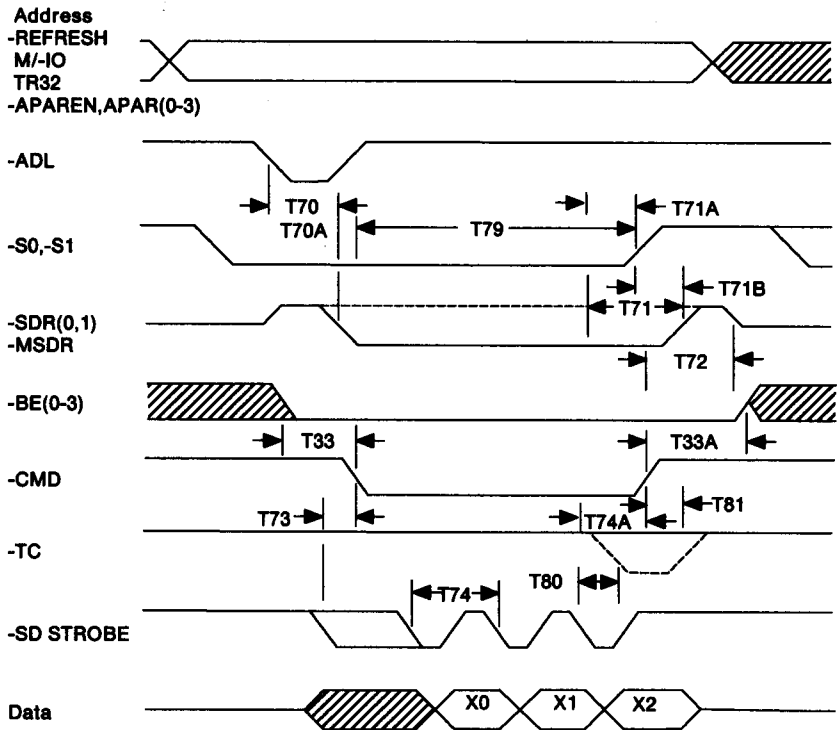


Figure 43. Streaming Data Cycle

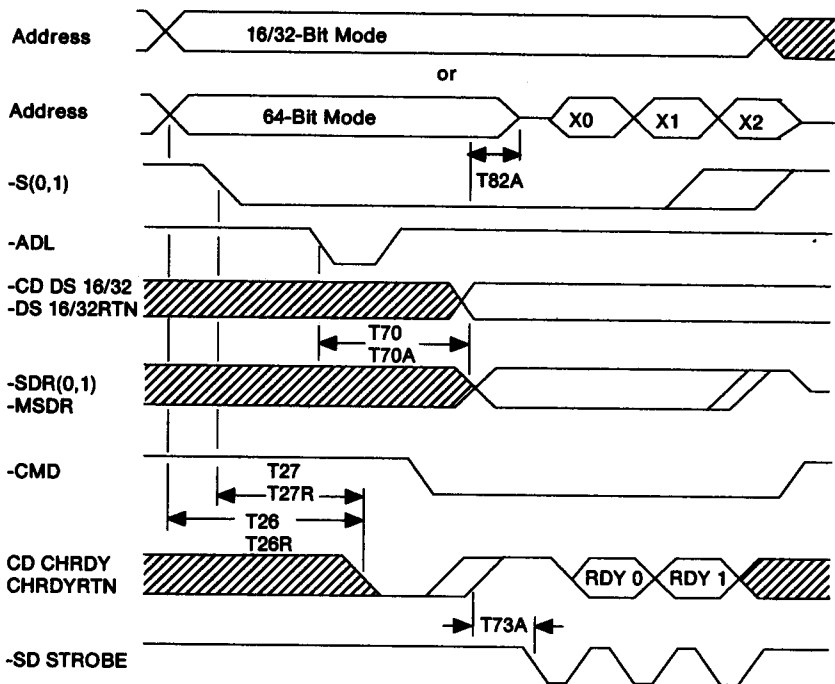


Figure 44. Streaming Data Cycle - Deferred Start

Note: CD CHRDY is not supported during 64-bit streaming data transfers after the address has been removed.

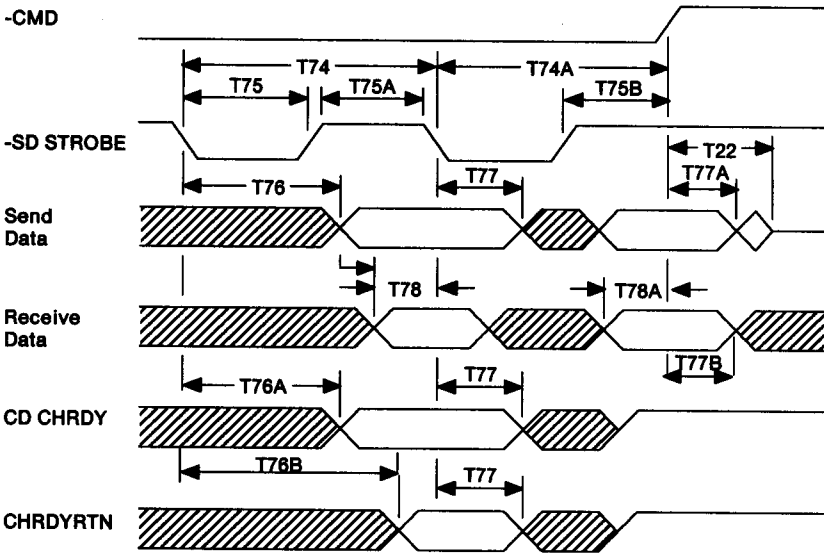
	Timing Parameter	Min/Max	Note
T22	Data bus tri-state from -CMD inactive	0 / 40 ns	
T33	-BE(0-3) active to -CMD active	10 / - ns	
T33A	-BE(0-3) hold from -CMD inactive	10 / - ns	
T70	-SDR(0,1) and -MSDR valid from -ADL active (slave only)	0 / 40 ns	7
T70A	-SDR(0,1) and -MSDR valid from -ADL active (masters only, for deferred cycle use)	0 / 115 ns	
T71	-SDR(0,1) and -MSDR inactive from last -SD STROBE fall (slave terminated cycle)	0 / 40 ns	
T71A	-S0, -S1 inactive from last -SD STROBE fall (master terminated cycle)	- / 10 ns	4,1
T71B	-SDR(0,1) and -MSDR inactive from -S0, -S1 inactive (master terminated; slave ready)	0 / 40 ns	
T72	-SDR(0,1) and -MSDR tri-state from -CMD inactive (master-terminated cycle)	0 / 40 ns	
T73	-SD STROBE active to -CMD active	- / 10 ns	3
T73A	-SD STROBE active from CHRDYRTN active	0 / - ns	
T74	-SD STROBE period	100 / - ns	
T74A	-CMD inactive from last -SD STROBE fall	100 / - ns	
T79	Status inactive from -CMD active (master-terminated cycle)	- / 7.8 μ s	6
T80	-TC active from last -SD STROBE fall	- / 60 ns	1
T81	-TC hold from -CMD inactive	10 / - ns	
T82A	A(0-31) tri-stated from CD CHRDY active	0 / - ns	5

Figure 45. Streaming Data Transfer Timing Specifications

Notes:

1. Streaming masters can increase T71A and T80 by the amount that the -SD STROBE clock period exceeds T74 minimum.
2. For streaming data cycles that default to the basic transfer cycles, the rules for deactivation of -s0, -s1 apply, as specified by the basic transfer procedure.
3. -SD STROBE can be driven active concurrently or after -CMD active, and can become active just before -CMD active due to driver skew and loading.
4. -s0, -s1 are deactivated concurrently with the negative transition of -SD STROBE. Normal skew from circuits and net loading can result in -s0, -s1 being deactivated prior to the negative transition of -sd STROBE in order to meet the T71A timing requirement.

5. This timing applies only to the 64-bit streaming data transfer.
6. The master must signal termination of the streaming data transfer within 7.8 microseconds from -CMD active. However, if -PREEMPT goes active, the master can continue to hold the channel for 7.8 microseconds from -PREEMPT .
7. A valid address decode must be completed before driving these signals.



See notes for definition of Send and Receive Data

Figure 46. Streaming Data Clcking (16- and 32-Bit)

	Timing Parameter	Min/Max	Note
T74	-SD STROBE period	100 / - ns	
T74A	-CMD inactive from last -SD STROBE fall	100 / - ns	
T75	-SD STROBE active	35 / - ns	
T75A	-SD STROBE inactive	35 / - ns	
T75B	-SD STROBE inactive to -CMD inactive	35 / - ns	
T76	Send Data valid from -SD STROBE fall	- / 60 ns	1
T76A	CD.CHRDY valid from -SD STROBE fall	3 / 25 ns	
T76B	CHRDYRTN valid from -SD STROBE fall	3 / 45 ns	
T77	Send Data and CD CHRDY hold from -SD STROBE fall	10 / - ns	
T77A	Write data hold from -CMD inactive	11 / - ns	
T77B	Read data hold from -CMD inactive	7 / - ns	
T78	Receive Data valid before -SD STROBE fall	25 / - ns	
T78A	Receive Data valid before -CMD inactive	25 / - ns	

Figure 47. Streaming Data Transfer Timing Specifications (16- and 32-Bit)

Notes:

1. Streaming masters can increase T76 by the amount that the -sd STROBE clock period exceeds T74 minimum.
2. To perform data transfer pacing, streaming masters can introduce momentary pauses in the data transfer by extending the active or inactive levels of the streaming data strobe.
3. Send Data refers to write data at the connector for the master and read data at the connector for the slave. Receive Data refers to read data at the connector for the master and write data at the connector for the slave.
4. The value of (T74 minus T76 minus T78) represents system timing margins to accommodate bus skew, propagation delays, and other signal transmission delays.

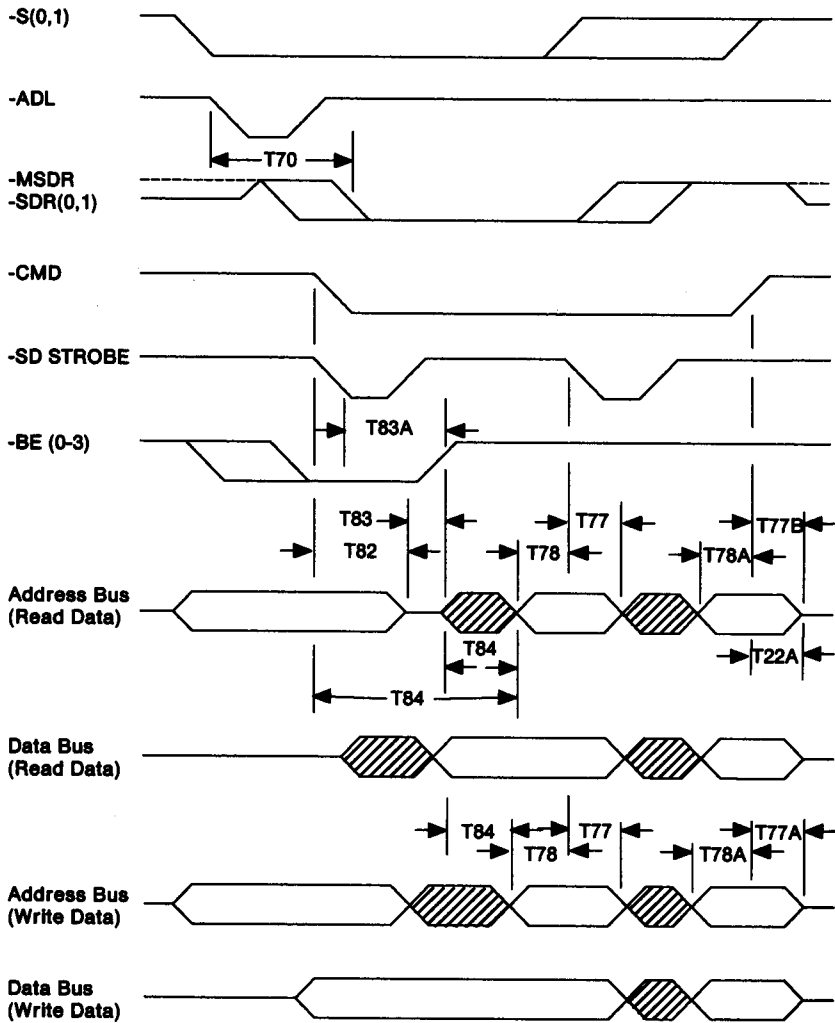


Figure 48. Streaming Data Clcking (64-Bit)

	Timing Parameter	Min/Max	Note
T22A	A(0-31) as D(32-63) tri-state from -CMD inactive	0 / 40 ns	2
T82	A(0-31) tri-stated from -CMD active	30 / - ns	1
T83	-BE(0-3) inactive from A(0-31) tri-stated	0 / - ns	3
T83A	-BE(0-3) inactive from -SD STROBE active	20 / - ns	1
T84	A(0-31) valid as D(32-63) from -BE(0-3) inactive and -SD STROBE active	0 / 60 ns	4

Figure 49. Streaming Data Transfer Timing Specifications (64-Bit)

Notes:

1. This timing applies only to a master during a read operation.
2. This timing applies only to a slave during a read operation.
3. During write operations, the master must meet the setup and hold times for the 16- and 32-bit streaming data transfer.
4. Streaming masters can increase T84 by the amount that the -sd STROBE clock period exceeds T74 minimum.

Direct Memory Access Procedure

A direct memory access (DMA) operation is a specialized sequence of basic transfer procedure. It consists of a single transfer or a burst transfer between a DMA slave and a memory slave. The following lists only those definitions, signal sequences, and timing specifications that are different from or additional to those of a basic transfer cycle.

Read and Write Operations

During a DMA write operation, the DMA slave provides the data and the memory slave stores it. During a DMA read operation, the memory slave provides the data and the DMA slave stores it. If the slave is unable to provide or store data within the time specified for a normal operation, it extends the cycle by driving CD CHRDY inactive. To ensure the read data is valid, the slaves storing the data sample the data lines with reference to the trailing edge of the -CMD.

DMA Slave Selection

A DMA slave is selected by either its arbitration level or optionally by its I/O address, but not by both. The default selection is by decode of the arbitration level, status (-s0 exclusive-OR -s1), and an I/O cycle (M/-IO in the -IO state). Selection by I/O address is by decode of the I/O address (A0-A15), status (-s0, -s1), and an I/O cycle (M/-IO in the -IO state). When selection by I/O address is implemented for a DMA slave, a bit located in either configuration-data space or I/O address space is used to control the method of selection. All DMA slaves drive -CD SDBK during the I/O portion of the DMA cycle for both selection methods.

Address During DMA Operation

During a memory cycle, the DMA controller drives the memory address on the address bus. During an I/O cycle, the DMA controller must drive a known I/O address on the bus. The I/O address must be that of the DMA slave, or hex FFFC or 0000.

Transfer Data Length

Normally, the DMA transfer length is an integral number of transfers of a width determined by the DMA slave's data port width. The data port address for a 16-bit DMA slave must be on an even address boundary; the address for a 32-bit DMA slave must be on a four-byte address boundary. The burden of supporting odd-length transfers is the responsibility of the DMA slave.

DMA Procedure

The two types of DMA procedures are:

- Single Transfer
- Burst Transfer.

Single Transfer

During a data transfer, the requesting DMA slave initiates the DMA operation by having its arbiter drive -PREEMPT active and arbitrate for the channel. On the low-to-high transition edge of ARB/-GNT (start of arbitration cycle), the DMA slave's priority level is placed on the arbitration bus. The DMA slave competes for use of the channel as defined in the arbitration procedure.

On the high-to-low transition edge of ARB/-GNT the DMA controller monitors the arbitration bus to determine if its DMA slave gained control of the channel. A match initiates a serial DMA operation in which the DMA controller drives the address lines during both cycles, as well as driving -S0, -S1, -CMD, and other control signals. The DMA controller also monitors -BURST, finds it inactive, and then activates -BURST (either on the channel, or through the system logic) to notify the central arbitration control point that consecutive cycles are required.

The DMA controller also manages the storage of data during the DMA read or DMA write operations for the consecutive cycles required by serial DMA transfers. Data count for data gated on or off the bus by the DMA slave is also maintained by the DMA controller.

Notes:

1. The DMA controller might be required to perform multiple cycles for the data transfer to the memory slave, if the data port width of the memory slave is less than that of the DMA slave.
2. DMA slaves using the single-transfer-type DMA procedure *will not* receive more data than is required to support their data port width (for example, 2 bytes to a 16-bit port).
3. Consecutive transfers can be made with the DMA slave (as opposed to the serial sequences shown in the remainder of this section). The memory read or write cycles might not appear on the channel.

Burst Transfer (Multiple Transfer Cycles)

The method for transferring a block of data is similar to a single transfer DMA cycle, except that the local arbiter for the requesting DMA slave activates -BURST to request continued channel ownership for a burst operation. The central arbitration control point will not drive ARB/-GNT to the ARB state while -BURST is active. As in single-transfer DMA operations, the DMA controller drives the address, -S0, -S1, -CMD, and data transfer signals, as well as manages addresses and data transfer counts. The DMA controller also drives -BURST (either on the channel or through the system logic) jointly with the DMA slave for DMA write cycles. It does not drive -BURST active for burst DMA read cycles.

DMA Terminating Conditions

The DMA transfer operation is terminated by the EOT. At this time, a DMA slave can again request DMA service by arbitrating for use of the channel.

The terminating conditions for DMA transfers are:

- During a single transfer, the DMA controller can terminate the DMA transfer by releasing -BURST.
- During burst transfers, either the DMA slave or the DMA controller can terminate the DMA operation.
 - The DMA slave terminates the transfer by driving -BURST inactive. The DMA slave will terminate the transfer when it reaches its buffer limit, or after -PREEMPT is driven active by another arbitrating participant. Arbitrating slaves should maximize use of the channel within the 7.8 microsecond time limit.
 - The DMA controller terminates the transfer by driving -TC active when the transfer count goes to zero. When the DMA slave receives -TC, it drives -BURST inactive.

Use of -BURST

When a bursting DMA slave wins the arbitration for the channel, it drives -BURST active. During default cycles, the DMA slave drives -BURST inactive when it detects status active for the last I/O cycle of this burst transfer. However, on sensing -SBHE and/or -BE(0-3), the DMA slave might discover that it has one more cycle to perform. If so, it redrives -BURST active to continue the transfer. See “Burst DMA Transfer (DMA-Slave Terminated - Default Cycle, 200 ns)” on page 91.

During extended cycles, the DMA slave drives -BURST inactive within the specified time after the leading edge of -CMD.

If a burst transfer is terminated by the DMA controller, the DMA slave drives -BURST inactive within the specified time after the leading edge of -TC.

For DMA write cycles using burst transfers, the DMA controller drives -BURST active at the leading edge of -CMD for each I/O read and drives it inactive when it detects status active for the next memory write.

DMA Timings

The following are the timing parameters for the DMA procedure.

Single DMA Transfer (DMA-Controller Controlled)

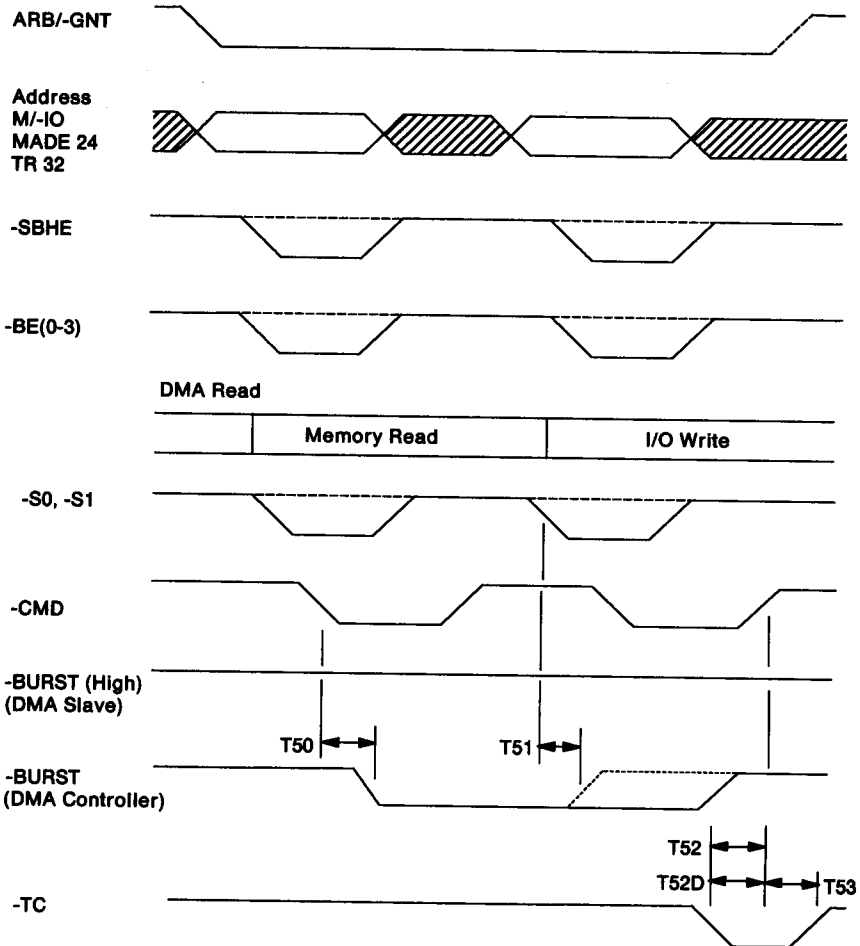


Figure 50 (Part 1 of 2). Single DMA Transfer (DMA-Controller Controlled)

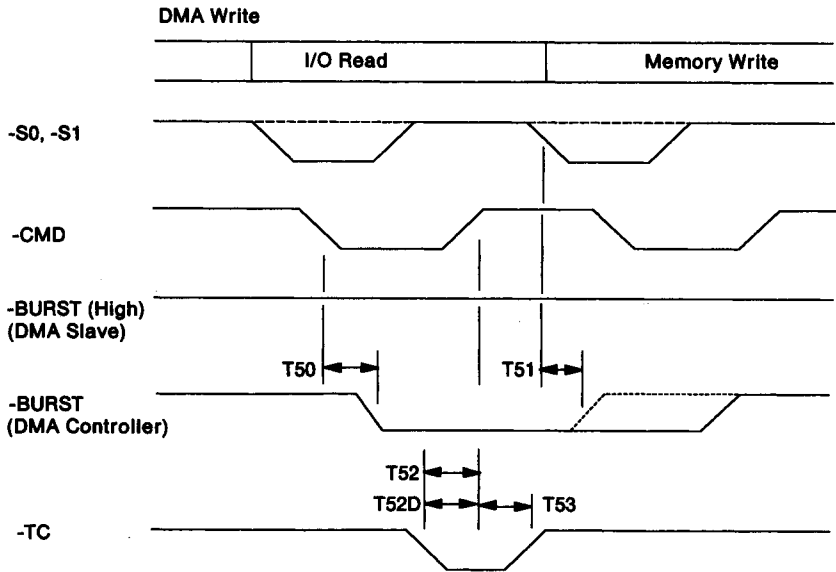


Figure 50 (Part 2 of 2). Single DMA Transfer (DMA-Controller Controlled)

Timing Parameter		Min/Max	Note
T50	-BURST activated by the DMA controller from the first -CMD active	- / 30 ns	
T51	-BURST released by the DMA controller from the last status active	- / 40 ns	
T52	-TC setup to -CMD inactive	30 / - ns	2
T52D	-TC setup to -CMD inactive	15 / - ns	2
T53	-TC hold from -CMD inactive	10 / - ns	3

Figure 51. Single DMA Transfer (DMA-Controller Controlled) Timing Specifications

Notes:

1. Only the timing parameters that are additional to those specified for the basic transfer cycle are included here.
2. T52D applies to participants operating at 200-nanosecond cycles *only*. Devices operating at cycle rates slower than 200 nanoseconds *must* meet T52.
3. For streaming data procedures, T80 and T81 also apply. (See "Streaming Data Timing" on page 77).

Burst DMA Transfer (DMA-Controller Terminated)

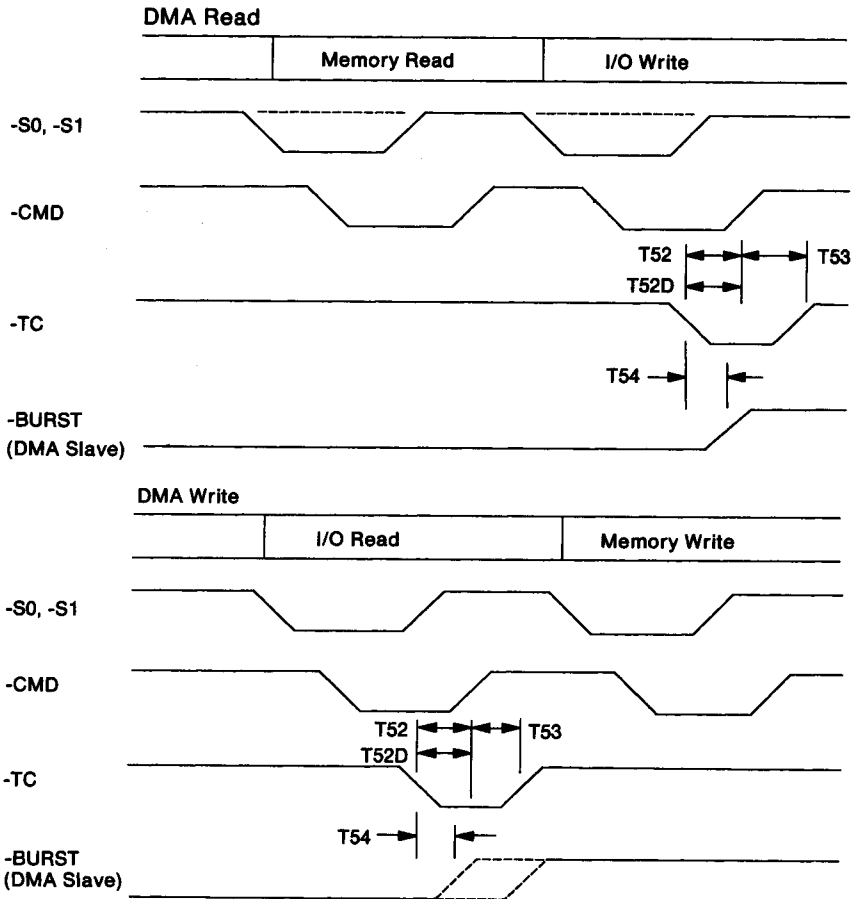


Figure 52. Burst DMA Transfer (DMA-Controller Terminated)

Timing Parameter	Min/Max	Note
T52 -TC setup to -CMD inactive	30 / - ns	1
T52D -TC setup to -CMD inactive	15 / - ns	1
T53 -TC hold from -CMD inactive	10 / - ns	3
T54 -BURST released by the DMA slave from -TC active	- / 30 ns	

Figure 53. Burst DMA Transfer (DMA-Controller Terminated) Timing Specifications

Notes:

1. T52D applies to devices operating at 200-nanosecond cycles *only*. Devices operating at cycle rates slower than 200 nanoseconds *must* meet T52.
2. During a DMA read, the DMA controller must not initiate another cycle after issuing -TC, even if the DMA slave is not fast enough to drive -BURST inactive before the end of -CMD.
3. For the streaming data procedure, T80 and T81 also apply. (See "Streaming Data Timing" on page 77).

Burst DMA Transfer (DMA-Slave Terminated - Default Cycle, 200 ns)

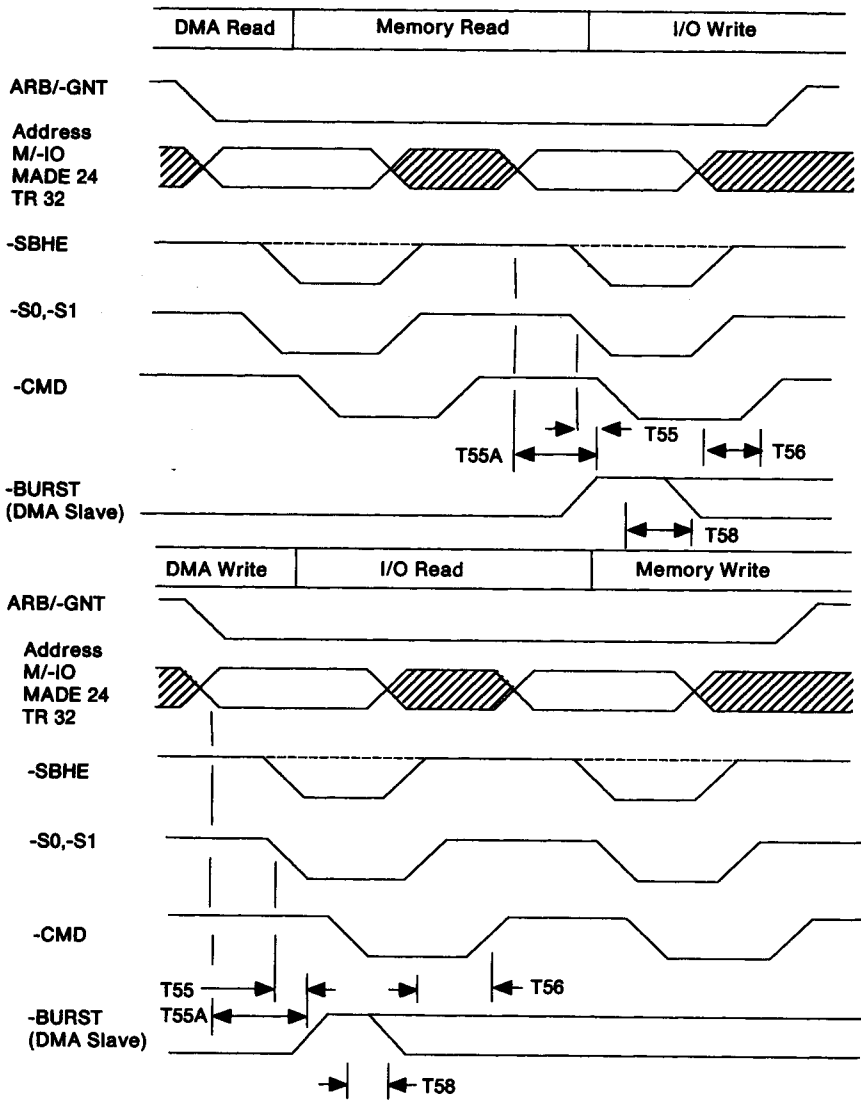


Figure 54. Burst DMA Transfer (DMA-Slave Terminated - Default Cycle, 200 ns)

	Timing Parameter	Min/Max	Note
T55	-BURST released by the DMA slave from the last I/O cycle status active (default cycle only)	- / 40 ns	1
T55A	-BURST released by the DMA slave from I/O address valid for the last I/O cycle (default cycle only)	- / 70 ns	1
T56	-BURST inactive to -CMD inactive	35/ - ns	2
T58	-BURST redrive from -CMD active	- / 30 ns	1

Figure 55. Burst DMA Transfer (DMA-Slave Terminated - Default Cycle, 200 ns) Timing Specifications

Notes:

1. After releasing -BURST and on receiving -SBHE and/or -BE(0-3), if the DMA slave determines that it requires another cycle, it can redrive -BURST active.
2. The setup time for -BURST inactive to the end of -CMD (T56) 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 time (T55/T55A controlled by the DMA slave) and the -BURST RC restoration time. The RC restoration time must not exceed 70 nanoseconds. T56 is the same for default and extended cycles.

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

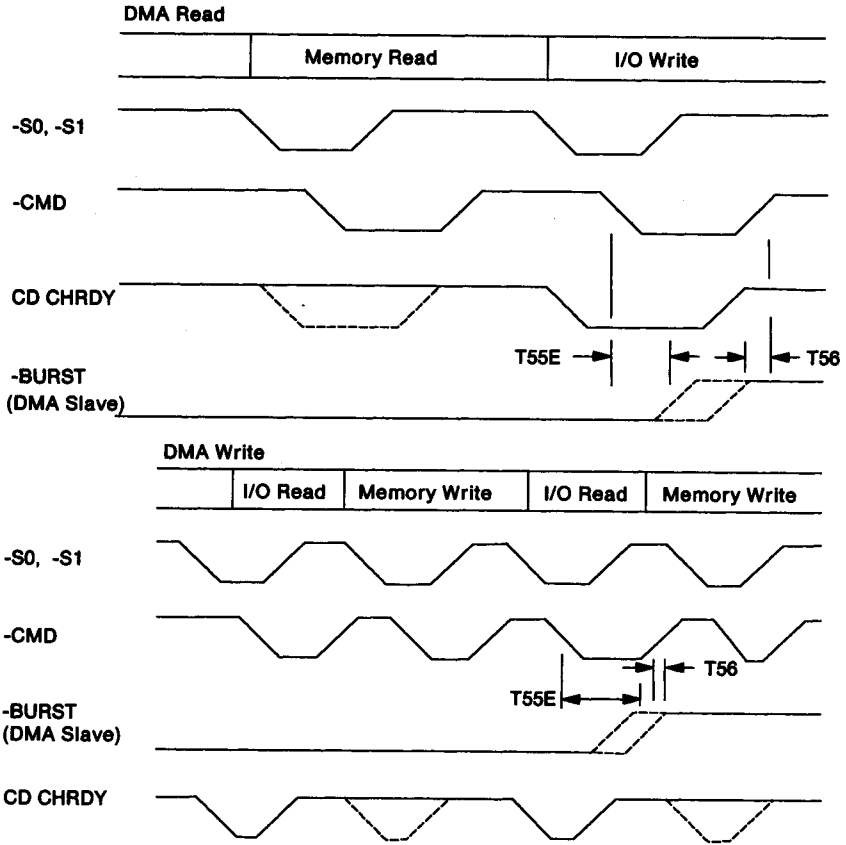


Figure 56. Burst DMA Transfer (DMA-Slave Terminated - Synchronous-Extended Cycle)

Timing Parameter		Min/Max	Note
T55E	-BURST released by the DMA slave from the last I/O -CMD active (extended cycles only)	- / 80 ns	2
T56	-BURST inactive to -CMD inactive	35 / - ns	1

Figure 57. Burst DMA Transfer (DMA-Slave Terminated - Synchronous-Extended Cycle, 300 ns) Timing Specifications

Notes:

1. The setup time for -BURST inactive to the end of -CMD (T56) 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 time (T55E controlled by the DMA slave) and the -BURST RC restoration time. The RC restoration time must not exceed 70 nanoseconds. T56 is the same for default and extended cycles.
2. -BURST cannot be redriven.

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

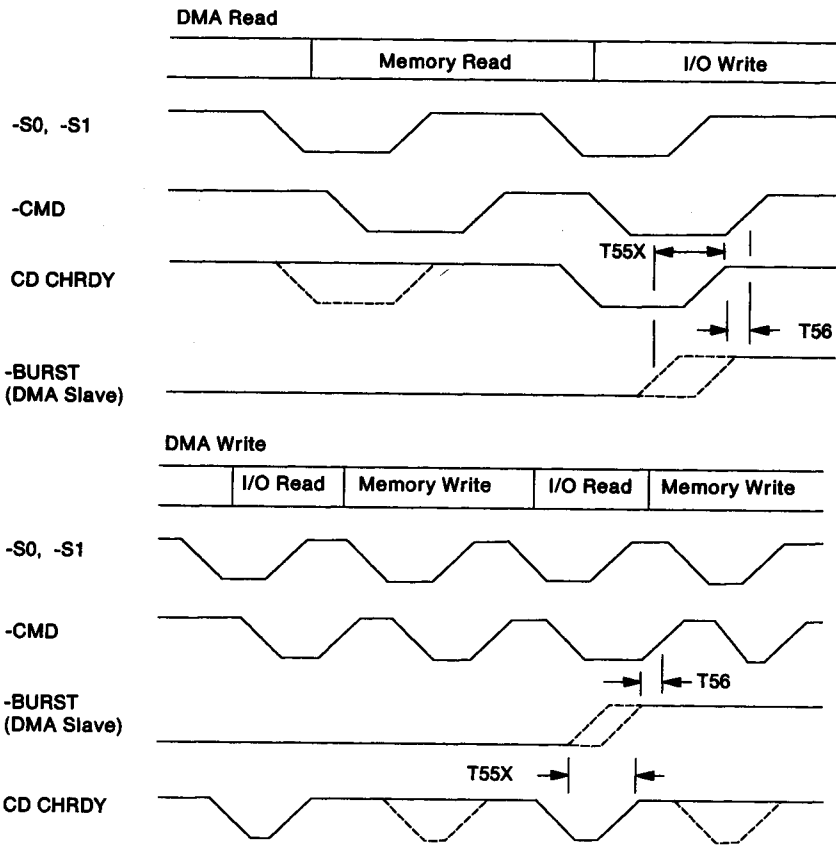


Figure 58. Burst DMA Transfer (DMA-Slave Terminated - Asynchronous-Extended Cycle, ≥ 300 ns)

	Timing Parameter	Min/Max	Note
T55X	-BURST released by the DMA slave before CD CHRDY active (asynchronous-extended cycles only)	50 / - ns	2
T56	-BURST inactive to -CMD inactive	35 / - ns	1

Figure 59. Burst DMA Transfer (DMA-Slave Terminated - Asynchronous-Extended Cycle, ≥ 300 ns) Timing Specifications

Notes:

1. The setup time for -BURST inactive to the end of -CMD (T56) 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 time (T55X controlled by the DMA slave) and the -BURST RC restoration time. The RC restoration time must not exceed 70 nanoseconds. T56 is the same for default and extended cycles.
2. -BURST cannot be redriven.

Interrupt Procedures

This section describes how all devices use interrupts. The procedure for sharing interrupts uses a request for interrupt services that is detected by the level of the interrupt request signal (level sensitive). These procedures involve the interaction between the hardware and an interrupt service routine.

To initiate an interrupt request, a device drives its $-IRQ(N)$ active (N represents the assigned interrupt level for the device) and holds the signal active until it is reset by the interrupt service routine.

Each device provides an interrupt-pending bit within its address space. This bit is set by the device when it has an interrupt request pending and is reset by the interrupt handler when the interrupt request is serviced.

Note: The drivers for the interrupt signal must be disabled when the device is disabled.

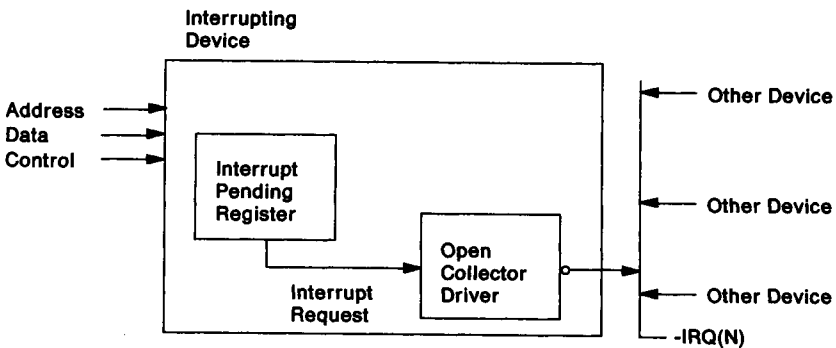


Figure 60. Typical Adapter Interrupt Sharing Implementation

An example of the sequence of the hardware and interrupt service routine interaction is shown in the following.

Hardware Operation

1. An interrupt condition causes the hardware to drive -IRQ(N) active and sets an interrupt-pending latch, which can be read by software.

Software Operation

2. A master begins executing code at the beginning of the appropriate chain of interrupt handlers.
3. The interrupt handler reads the interrupt-pending latch of the first device in the chain. If the latch is not set, the next device in the chain is tested. When the reporting device is detected, the handler executes the service routine.
4. The interrupt service routine operates the device hardware.

Hardware Operation

5. The device hardware resets the interrupt-pending latch and the interrupt request because of interrupt-service-routine actions.

Software Operation

6. The interrupt service routine completes the interrupt processing.

Hardware Operation

7. If another interrupt is pending (-IRQ(N) driven active by another device), the sequence starts again at 1.

Refresh

The timing of the refresh operation is performed by system logic. **-REFRESH** is driven active during a memory-read operation to indicate that a refresh cycle is in progress. The refresh cycle is a basic transfer default cycle with **-REFRESH** active. The timing for **-REFRESH** is the same as that of the address bus.

Note: Adapters with registers that have destructive read-outs and are mapped into the memory address space should include **-REFRESH** in their address decode.

During a refresh operation, the nine low-order address bits are incremented for each refresh cycle to address the next byte to be refreshed. (A0 changes after each refresh cycle; A8 changes after 256 cycles.)

The address bits, A31 through A9, might not change, but these bits will be stable during the refresh cycle.

The refresh operation can occur in periodic intervals or in short bursts of multiple operations. The refresh rate is a minimum of 128 refresh cycles every 2 milliseconds.

Note: Time periods between refresh cycles can vary and should not be used as a timing mechanism.

Regardless of whether it decodes **-REFRESH** or not, a memory slave can perform a memory read operation during a refresh cycle and drive its data onto the data bus, provided its memory address is on the address bus.

Memory slaves that do not need to perform refresh operations do not need to receive this signal.

System Configuration

The adapter configuration procedure consists of:

- Adapter ID and adapter configuration data
- Programmable option select (POS) registers
- Nonvolatile memory (for system configuration data)
- Signal timing specifications.

The system master configures a system using setup cycles. The system configuration procedure supports the identification of the adapters that reside within a system (by reading the adapter IDs) and the configuration of those adapters (by writing configuration data). Configuration data consists of the following programmable fields:

- Required Fields
 - Adapter ID - the two byte adapter ID
 - Card Enable - a bit that enables or disables the output drivers on the adapter
 - Channel Check - a bit that indicates if the adapter has driven -CHCK active
- Conditionally Required Fields
 - Adapter I/O Address Select - if two or more of the same type of adapter can be installed in a system
 - Adapter ROM/RAM Address Select - if the adapter uses part of the memory address space at hex 000C0000 to 000DFFFF
 - Arbitration level - if the adapter is an arbitrating participant
 - Auto-Incrementing Enable - if the adapter uses POS Registers 6 and 7 as an automatically-incremented register
 - Channel Check Exception Reporting - if the adapter supports synchronous -CHCK for channel non-parity or slave-dependent exceptions
 - Channel Check Status Indicator - if the adapter supports -CHCK
 - Data and Address Parity Enable - if the adapter supports data or address parity

- Device I/O Address Assignment - if the total I/O-address-space required for a type of adapter is greater than 32 bytes
- Matched Memory Enable - if the adapter supports the matched-memory extension
- Selected Feedback Return Exception Enable - if the bus master receives -SFDBKRTN
- Streaming Data Enable - if the adapter supports streaming data procedures
- Optional Fields
 - Option Select Data - customized, adapter-unique configuration data
 - Fairness Enable - optional if the adapter drives -BURST
 - POS Subaddress Extension
 - Status Information - contains the channel-check status or a pointer to the channel-check status.

See the specific-system technical manuals for system configuration information.

Multiple Card Adapters

Multiple-card adapters are adapters that consist of a primary card and one or more secondary cards. The secondary cards are installed in the channel connectors adjacent to the primary card and can be on either, or both, sides of the primary card. The number of secondary cards is limited only by the number of connectors available.

The multiple-card adapter provides only one set of POS registers. Only the primary card responds to -CD SETUP.

The multiple-card adapter operates as a single adapter, and is completely enabled or disabled by the single set of POS registers.

Adapter Configuration Data

After initialization, the system must be able to access the configuration data written into adapter POS registers.

Adapters must always respond to setup cycles. After initialization, the ability to change the configuration data in the POS registers for an adapter is adapter dependent.

Figure 61 shows both the address and field definition. When its -CD SETUP is active, the adapter must provide access to its configuration data. This data is contained in its POS registers and is divided into multiple device-unique, read/write fields.

The following table shows the organization of the I/O address space used for POS operations. Setup functions respond to these addresses only when -CD SETUP is active.

Address (Hex)	Function
XXX0	POS Register 0 - Adapter Identification Byte (Low Byte)
XXX1	POS Register 1 - Adapter Identification Byte (High Byte)
XXX2	POS Register 2 - Option Select Data Byte 1 Bit 0 is designated as Card Enable.
XXX3	POS Register 3 - Option Select Data Byte 2
XXX4	POS Register 4 - Option Select Data Byte 3
XXX5	POS Register 5 - Option Select Data Byte 4 Bit 7 is designated as channel check. Bit 6 is designated as channel-check-status indicator.
XXX6	POS Register 6 - Subaddress Extension (Low Byte)
XXX7	POS Register 7 - Subaddress Extension (High Byte)

Figure 61. POS I/O Address Space

Bits 6 and 7 of POS Register 5 and bit 0 of POS Register 2 are defined by the architecture; all other bits within POS Registers 2 through 5 are available for adapter use.

Required Configuration Fields

The following configuration fields are required for all adapters.

Note: When CHRESET is driven active, adapters must set all fields to the default state.

Adapter ID Field

The Adapter ID field is a two-byte, read-only field contained in POS Register 0 and 1. Each adapter is assigned a unique two-byte adapter ID that is used for adapter identification. An adapter ID of hex 0000 indicates that the adapter is not ready; if an adapter is not present, the adapter ID will be read as hex FFFF.

Card Enable Field

The card enable field is a one-bit, read/write field; it is bit 0 in POS Register 2. When this field is set to 0, the adapter disables the Micro Channel interface except when responding to its setup cycles. When this field is set to 1, the adapter enables all of its Micro Channel interface. The default state of this field is 0.

Channel Check Field

The Channel Check field is a one-bit, read/write field that indicates the adapter has driven -CHCK active; it is bit 7 of POS Register 5. The adapter sets this field to 0 to indicate it has an exception condition, and the system master resets this field to 1 to reset the adapter after handling the error condition. If the adapter has driven -CHCK active, it must set this field to 0 except for address parity exceptions. If the system master resets this field to 1 and the adapter is still driving -CHCK active, the adapter must drive -CHCK inactive.

Note: If this field is set to 0 during a setup cycle, the adapter should not drive -CHCK active.

Conditionally-Required Configuration Fields

The following are the configuration fields that are required when the adapter supports a specified function.

The Channel Check Status Indicator field has a specific location and state because the system master must handle the channel-check condition. Other conditionally-required fields do not have a specific location within POS address space or a specific state.

Note: When CHRESET is driven active, adapters must set all fields to the default state.

Adapter I/O Address Select Field

The Adapter I/O Address Select field is a one- to three-bit field that selects the adapter address decode from the possible address ranges. These bits allow I/O address resolution for up to eight devices of one type in a system. However, if the total I/O address space needed to support the maximum number of devices exceeds 32 bytes, use the Device I/O Address Assignment field (see page 106).

Adapter ROM/RAM Address Select Field

The Adapter ROM/RAM Address Select field is a four-bit, read/write field that determines the starting address of the adapter ROM/RAM address space for the adapter. The adapter address space is defined as the memory address space from hex 000C0000 to 000DFFFF. The four bits represent address bits A16 through A13, which corresponds to 16 consecutive 8KB blocks.

Note: Adapters that require more than 16KB in this memory space can cause addressing conflicts that cannot be resolved during system configuration.

Arbitration Level Field

The Arbitration Level field is a four-bit, read/write field that selects the arbitration level. All adapters that use the arbitration procedure (bursting and nonbursting) must support a selectable arbitration level.

Auto-Incrementing Enable Field

The Auto Incrementing Enable field is a one-bit, read/write field that enables or disables the automatic incrementing of POS Registers 6 and 7. The default state for this field is disable, which causes the adapter to not perform auto-incrementing. When enabled, it causes POS Registers 6 and 7 to automatically increment for each access to POS Register 3 or 4 (see "POS Subaddress Extension Field" on page 108).

Channel Check Exception Reporting Field

The Channel Check Exception Reporting field is a one-bit, read/write field that selects either synchronous or asynchronous -CHCK exception reporting for channel non-parity or slave-dependent exception conditions. When enabled, these exceptions are reported with synchronous -CHCK. When disabled, these exceptions are reported by asynchronous -CHCK. The default state for this field is disabled (see "Exception Reporting" on page 131 for more information on exception reporting).

Channel Check Status Indicator Field

The Channel Check Status Indicator field is a one-bit, read-only field that indicates if channel-check status is available; it is bit 6 of POS Register 5. The default state of this field is 1. This field is required if the Channel Check field is supported by the adapter.

When this field is a 1, it indicates that status for the channel-check condition is not available. When the field is 0 and the Channel Check field is 0, it indicates that POS Registers 6 and 7 contain the channel-check status or a pointer to the status. (If an adapter does not support -CHCK, bit 6 of POS register 5 can be used for other configuration information.)

Data and Address Parity Enable Field

The Data and Address Parity Enable field is a one-bit, read/write field that is required if the adapter supports address or data parity. This field enables address or data parity on the channel. The default state for this field is disabled. While disabled, the device does not drive either -DPAREN or -APAREN active, and does not check parity from the channel. This field can be a two-bit field to individually enable data and address parity.

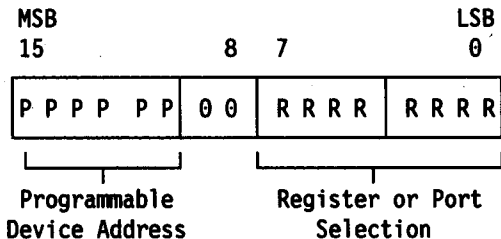
Device I/O Address Assignment

The Device I/O Address Assignment field is a six-bit field that selects the I/O address of the adapter. The six bits represent bits 15 through 10 of the I/O address as shown in the following figure. All I/O devices with a total I/O-address-space requirement greater than 32 bytes must support this field.

The total I/O-address-space requirement for a particular type of device is:

The number of bytes of I/O address space required by one device multiplied by the maximum number of devices of the same type supported in one system.

The I/O address assigned to the device is:



Address bits 15, 14, and 13 must be in POS registers 2 through 5. Address bits 12, 11, and 10 can be in POS registers 2 through 5 or in the POS extension. The address bits in the POS extension must default to 1 at power on.

Matched Memory Enable Field

The Matched Memory Enable field is a one-bit, read/write field that enables or disables support for the matched-memory signals. The default state of this field is disable, which causes the memory adapter to use the default cycle of the Basic Transfer procedures. When the field is in the enable state, the adapter will support the matched-memory signals.

Selected Feedback Return Exception Enable Field

The Selected Feedback Return Exception Enable field is a one-bit, read/write field that selects whether or not a master reports select-feedback-return exceptions. When the field is in the enable state, the adapter reports select-feedback-return exceptions. When the field is in the disable state, the adapter does not report a select-feedback-return exception. The default state of this field is disable.

Streaming Data Enable Field

The Streaming Data Enable field is a one-bit, read/write field that enables support for streaming data procedures. The default state of this field is disable, which causes the adapter to use the basic data transfer procedures. Adapters that support streaming data procedures must support this bit. When disabled, a slave must tri-state its streaming data signals, and a master must tri-state `-SD STROBE`. When enabled, the streaming data procedure is supported.

Optional Configuration Fields

The following configuration fields are optional.

Note: When `CHRESET` is driven active, adapters must set all fields to the default state.

Option Select Data Field

The Option Select Data field is four bytes of adapter unique information in POS Registers 2 through 5 that allow developers to customize configuration data for the specific adapter. Bit 0 of POS Register 2 and bits 7 and 6 of POS Register 5 are defined by the architecture; all other bits within the Option Select Data field are available for adapter configuration.

Fairness Feature Enable Field

The Fairness Feature Enable field is a one-bit, read/write field that enables or disables the fairness feature. The default state for this bit is enabled (see "Fairness Feature and the Inactive State" on page 34).

POS Subaddress Extension Field

The POS Subaddress Extension field is an optional, two-byte field that provides a method to extend the POS register space. It is contained in POS Registers 6 and 7 and is used as an index for all setup-cycle accesses to POS Registers 3 and 4. This field allows the subaddressing of up to 128KB of additional information for each adapter.

The POS extension is an extension to POS Registers 3 and 4 and is made up of two one-byte registers. An index of hex 0000 results in

setup cycles accessing POS Register 3 and 4. All adapters supporting the POS extension must set the index to hex 0000 when CHRESET is driven active. The default value for this field is hex 0000.

Adapters supporting POS extension can use POS Registers 6 and 7 as an index register that is automatically incremented for each access to POS Register 3 or 4. Adapters supporting this feature use the Auto-Incrementing field to enable the feature.

When bits 6 and 7 in POS Register 5 are set to 0, reading POS Registers 6 and 7 will return the channel-check-status information. This information can be status or a pointer to the status.

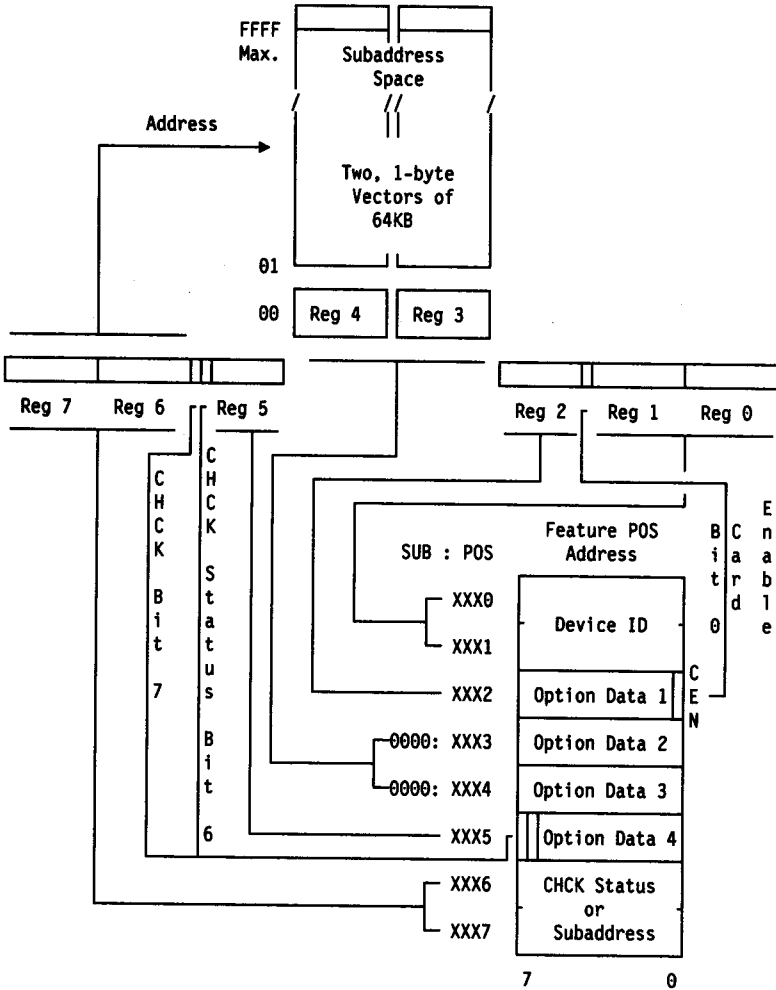


Figure 62. Subaddressing Model

The following is an example of how subaddressing could be used to store or access data in the POS extension.

- POS Register 6 is set to hex 01
- POS Register 7 is set to hex 00
- Access to the POS extension is through POS Register 4
- POS-extension data is in consecutive locations
- Extended POS addressing starts at hex 0001, unless specified by the adapter description file.

Note: The amount of POS-extension information that can be supported is system dependent.

Status Information Field

The Status Information field is contained in POS Registers 6 and 7. The field is used by an adapter to indicate its channel-check status or as a pointer to channel-check status.

System Configuration Procedure

The system configuration procedure uses setup cycles to transfer adapter ID and adapter configuration data between the system master and adapters. The system configuration procedure is similar to the basic transfer procedure, except for the following:

- An adapter is selected by an active **-CD SETUP** (not by an address decode or arbitration level).
- An adapter must not drive **-CHCK** active because of a setup cycle.
- A valid POS register address is driven on the address bus; only address bits **A(0-2)** are decoded by the adapter.
- **M/-IO** must be in the **-IO** state during the setup cycle, and adapters must decode **M/-IO** in the **-IO** state.
- The adapter does not drive **-CD SFDBK** active.
- All configuration data transfers are single-byte (8-bit) transfers. All transfers to an 8-bit data port occur on **D(0-7)**. If a 16-bit or 32-bit slave drives **-CD DS 16** or **-CD DS 32**, the 8-bit transfer occurs on the data lines indicated by **-SBHE**, **A0**, and **A1**, or by **-BE(0-3)** (see Figure 29 on page 53).
- The minimum setup cycle is 300 nanoseconds; adapters can extend the cycle by driving **CD CHRDY** inactive.

Address Bus During Setup Cycles

During a setup cycle, a valid POS-register address is driven on the address bus; only address bits **A(0-2)** are decoded by the adapter. The address driven on the bus must not be the I/O address of any adapter.

The following figure shows the organization of the address space used by POS during setup.

-CD SETUP	M/-IO	Address			Function
		A2	A1	A0	
0	0	0	0	0	Adapter ID (Low Byte)
0	0	0	0	1	Adapter ID (High Byte)
0	0	0	1	0	Option Select Data Byte 1
0	0	0	1	1	Option Select Data Byte 2
0	0	1	0	0	Option Select Data Byte 3
0	0	1	0	1	Option Select Data Byte 4
0	0	1	1	0	Subaddress Extension/Channel Check Status (Low Byte)
0	0	1	1	1	Subaddress Extension/Channel Check Status (High Byte)

Figure 63. POS I/O Address Decode

Card Enable Field During Setup Cycles

An adapter selected by -CD SETUP must enable its drivers, regardless of the state of the Card Enable field.

Channel Check Status During Setup Cycles

When -CHCK is driven active, the Channel Check field indicates that the adapter drove -CHCK active (except for address parity). The Channel Check, Channel Check Status Indicator, and Status Information fields (and channel-check status, if the field is used as a pointer) must remain unchanged until the system master resets the Channel Check field to 1 or until CHRESET is driven active.

-CD SETUP During Setup Cycles

During a setup cycle, only one adapter at a time is selected. The adapter is selected by an active -CD SETUP, not by an address decode or arbitration level.

CD CHRDY During Setup Cycles

CD CHRDY can be used to extend the setup cycle, as in the basic transfer procedure (see "Asynchronous-Extended Cycle Timing (≥ 300 ns)" on page 50 for usage of CD CHRDY in an extended cycle).

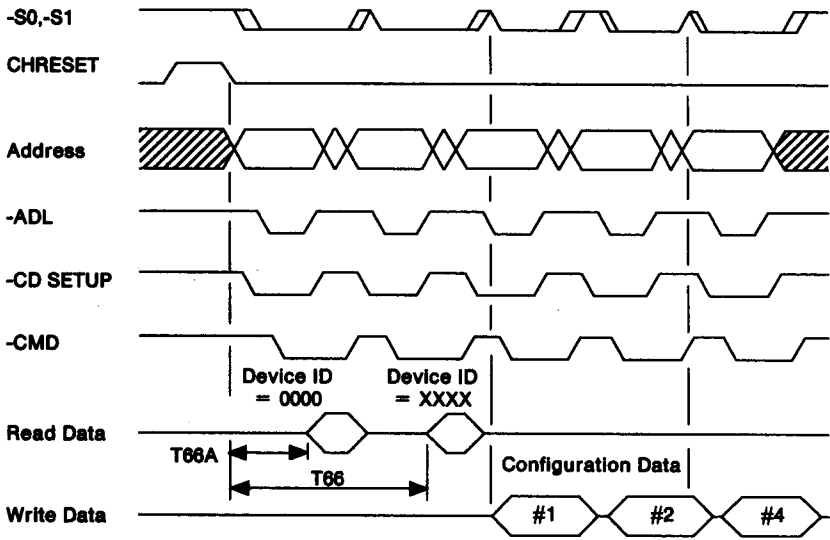


Figure 64. Setup Cycle Overview

System Configuration Timing Diagrams

The following provides the specification for the signal timings used for the system configuration procedure.

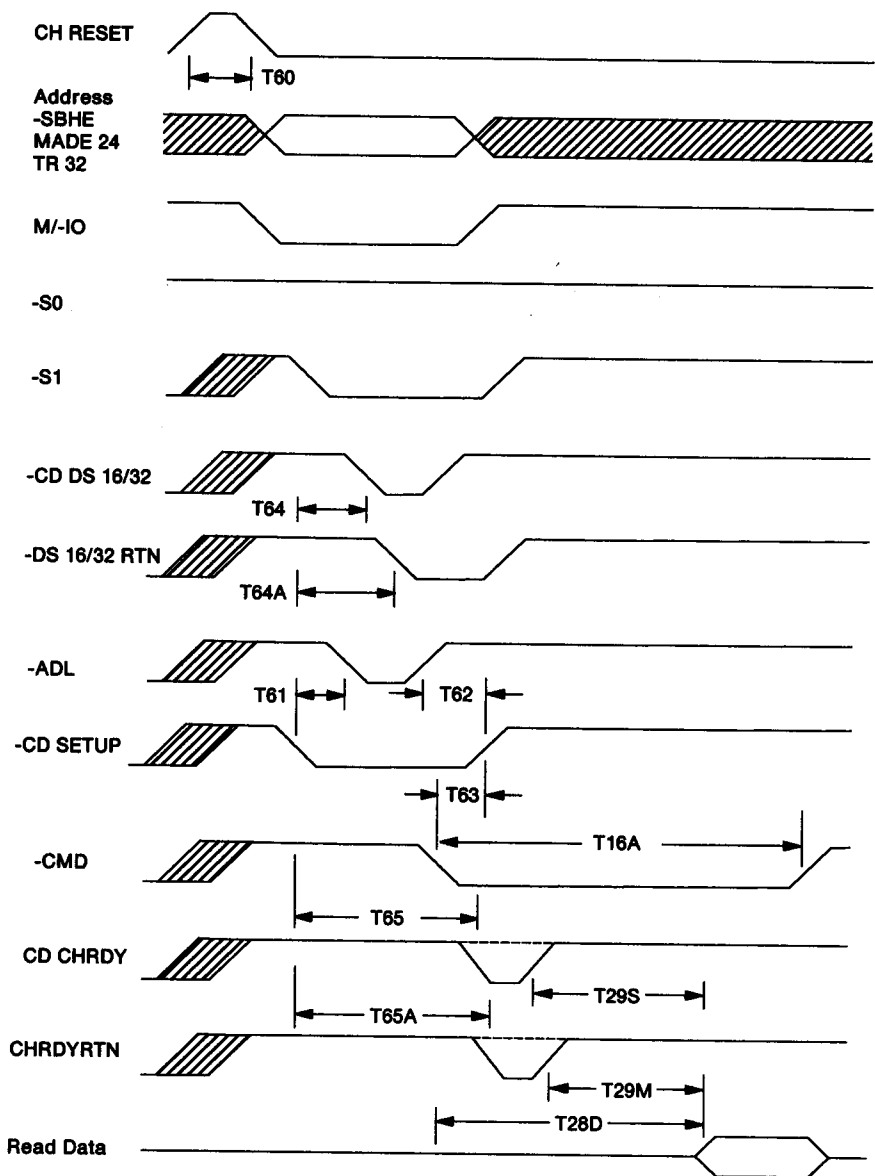


Figure 65. System Configuration Timing

	Timing Parameter	Min/Max	Note
T16A	-CMD Pulse width	190 / - ns	1
T28D	Read data valid from -CMD active	- / 160 ns	1
T29S	Read data from slave valid from CD CHRDY active	- / 60 ns	1
T29M	Read data valid to master from CHRDYRTN active	- / 60 ns	1
T60	CHRESET active pulse width	100 / - ms	
T61	-CD SETUP active to -ADL active	15 / - ns	
T62	-CD SETUP hold from -ADL inactive	25 / - ns	
T63	-CD SETUP hold from -CMD active	30 / - ns	
T64	-CD DS 16 active from -CD SETUP active	- / 25 ns	
T64A	-DS 16 RTN active from -CD SETUP active	- / 45 ns	
T64	-CD DS 32 active from -CD SETUP active	- / 25 ns	5
T64A	-DS 32 RTN active from -CD SETUP active	- / 45 ns	5
T65	CD CHRDY inactive from -CD SETUP active and Status active	- / 100 ns	3
T65A	CHRDYRTN inactive from -CD SETUP active and Status active	- / 120 ns	
T66	Adapter ID valid from trailing edge of CHRESET	- / 1 s	4
T66A	Adapter ID access from trailing edge of CHRESET	1 / - ms	4

Figure 66. System Configuration Timing Specifications

Notes:

1. See "Synchronous-Extended Cycle Timing (300 ns Minimum)" on page 47 and "Asynchronous-Extended Cycle Timing (≥ 300 ns)" on page 50 for specific details. The following are the timing parameters for the system configuration procedure that are different or additional to those specified for the basic transfer procedure.
2. The minimum setup cycle is 300 nanoseconds. A valid non-card-selecting address must be present on the address bus during setup. During setup, the address on the address bus must not be an I/O address of any adapter.
3. A slave can extend the setup cycle beyond 300 nanoseconds with CD CHRDY.

4. All adapters must present their adapter ID within 1 second after CHRESET is driven inactive. If an adapter is not ready to present its complete adapter ID before T66 time, it must present an ID of hex 0000. If an adapter continues to present an ID of hex 0000 for more than 1 second after CHRESET, it is considered defective.
5. Signals and timings only supported by the 32-bit connector.

Exception Condition Reporting and Handling

An exception condition is an event (including errors) that disrupts normal processing by the master. Resolving an exception usually requires system intervention. Exception conditions are reported to the controlling master or the system master using either -CHCK or an interrupt. Methods for reporting exceptions are described in "Exception Reporting" on page 131.

Note: An adapter whose master function detects exception conditions reports the condition through its I/O slave function by driving -IRQ active.

Exception Conditions

The following types of exception conditions are supported:

- Channel data and address parity exceptions
- Channel non-parity exceptions
- Master- and slave-dependent exceptions
- Channel-timeout exceptions.

Descriptions of each type are included in the following sections.

Channel Data and Address Parity Exceptions

Data and address parity support are optional features on the channel. This type of exception can occur during read or write operations. All participants that support data and address parity checking provide a POS bit (Data and Address Parity Enable field) to enable and disable this feature. The default state of this field is disable. (See "Data and Address Parity Enable Field" on page 106.)

When the Data and Address Parity Enable field is in the disable state, slaves disable checking data and address parity and do not activate -DPAREN during a data-transfer read cycle. Masters disable activation of -DPAREN and -APAREN and do not check data parity on the channel.

Note: Slaves do not disable the driving of -CHCK for non-parity exceptions (independent of the Data and Address Parity Enable field) and masters do not disable the detection of -CHCK.

The description of data and address parity support follows.

Data Parity

Data parity support is optional and allows the use of parity and non-parity adapters on the channel. For exceptions to be detected and reported, both the master and slave involved in the operation must support data parity. `-DPAREN` is activated by masters on write operations and by slaves on read operations. (`-DPAREN` is activated by the data source only, and indicates the support of data parity.) The data parity bits, `DPAR (0-3)`, are provided at the same time as data on both write and read cycles, when parity is supported by the master and slave. Only the data bytes used during the data transfer need to have correct parity. For 64-bit streaming, the address parity bits are used during data transfers. `-DPAREN` is used to indicate that parity is valid on the address bus during 64-bit streaming data transfers. `-APAREN` remains inactive or is deactivated during the data portion of the transfer.

Read Operations - Basic Transfer

A read cycle with parity checking is shown in Figure 67 on page 121. It is a basic-transfer read cycle that adds data parity checking. During a read cycle, slaves supporting data parity drive `-DPAREN` active with the data following the activation of `-CMD`. When `-CMD` is deactivated, slaves must drive `-DPAREN` inactive then tri-state it. This ensures that `-DPAREN` is inactive before a write cycle begins. If a slave fails to restore `-DPAREN`, adjacent read and write cycles issued from a controlling master that does not support data parity could cause a slave that supports data parity to erroneously assume the write cycle included parity.

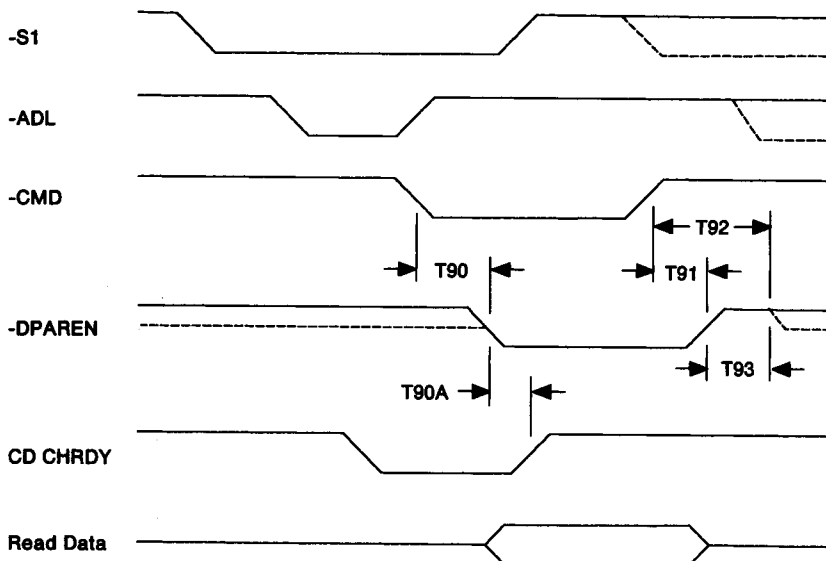


Figure 67. Data Parity Checking Read Cycle - Basic Transfer

Timing Parameter	Min/Max	Note	
T90	-DPAREN active from -CMD active	0 / 60 ns	1
T90	-DPAREN active from -CMD active	0 / 160 ns	2
T90	-DPAREN active from -CMD active	0 / - ns	3
T90A	-DPAREN active to CD CHRDY active	- / 0 ns	3
T91	-DPAREN inactive from -CMD inactive	0 / 40 ns	4
T92	-DPAREN tri-stated from -CMD inactive	0 / 80 ns	4
T93	-DPAREN tri-stated from -DPAREN inactive	20 / - ns	

Figure 68. Data Parity Checking Read Timing Specifications

Notes:

1. Default cycle timing only.
2. Synchronous-extended cycle timing only.
3. Asynchronous-extended cycle timing only.
4. -DPAREN must be driven inactive before tri-stating.

Masters supporting data parity must check data parity on read operations, if -DPAREN is activated. If the master detects a parity error, the slave is not informed that the error occurred.

Read Operations - Streaming Data Transfer

Timing of -DPAREN for streaming-data read operations is shown in the following figure.

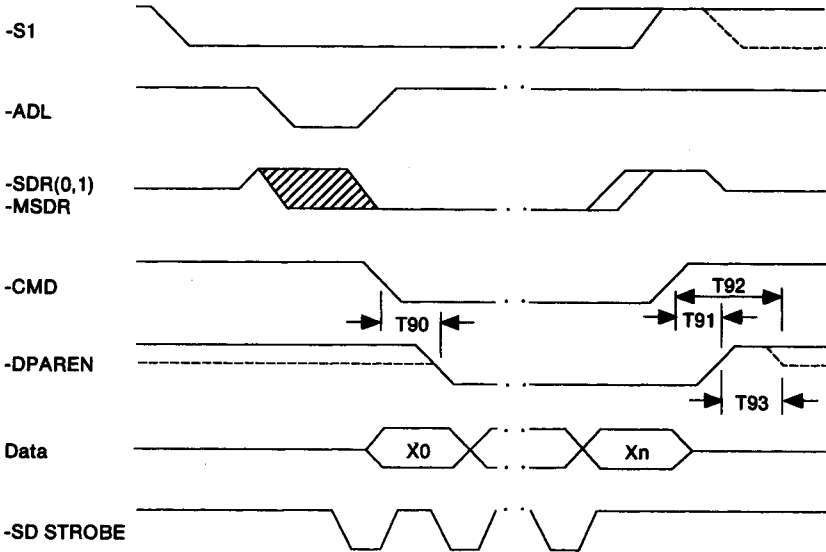


Figure 69. Data Parity Read Cycle - Streaming Data Transfer

Notes:

1. See Figure 68 on page 121 for timing specifications not listed here. For Streaming Data Read Cycles (when CD CHRDY is not driven inactive) T90 is the same as T90 for the basic-transfer cycle.

Write Operations - Basic Transfer

The timing of -DPAREN during write cycles is shown in Figure 70 on page 123. The figure is a basic-transfer write cycle but includes data parity. A master that supports data parity activates -DPAREN during write cycles to indicate that parity is associated with the data. To aid in parity checking at the slave, masters supporting parity are required to provide the data setup time with respect to -CMD. See Figure 70 on page 123 for -DPAREN and T96 timing specifications.

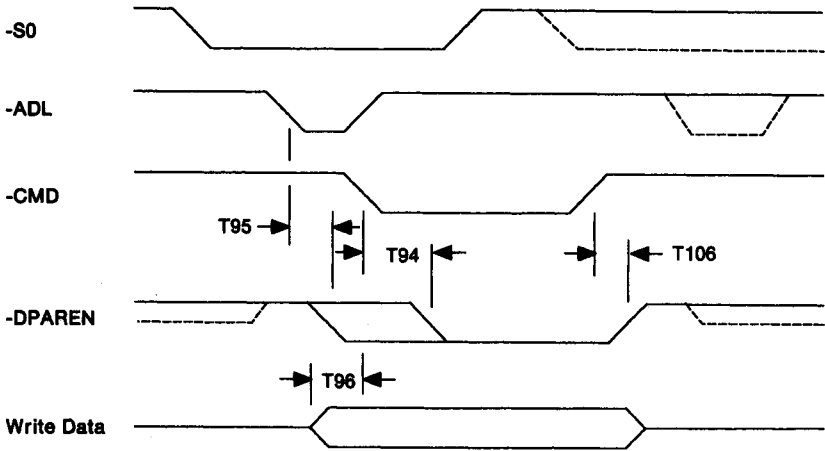


Figure 70. Data Parity Checking Write Cycle - Basic Transfer

Timing Parameter	Min/Max	Note
T94	- / 20 ns	1
T95	40 / - ns	2,4
T96	15 / - ns	2
T106	20 / - ns	3

Figure 71. Data Parity Checking Write Timing Specifications

Notes:

1. Controlling masters can drive -DPAREN active concurrently with -CMD during write operations. T94 specifies the maximum allowable skew.
2. Applies only to masters supporting data parity.
3. When a write cycle follows a basic-transfer write cycle, the master can optionally hold -DPAREN active between the cycles. When a write cycle follows a streaming data write cycle, the master must drive -DPAREN inactive between cycles. If a master does not hold -DPAREN active between cycles, it must drive -DPAREN inactive following the write cycle. The master must ensure -DPAREN has had time to be restored to the inactive state before sampling its parity check logic during the read cycle. (The master should allow for a 90 ns RC recovery of -DPAREN without an active restore.)

4. Applies only for a master executing a write cycle preceded by a read cycle.
5. See Figure 68 on page 121 for timing specifications not listed here.

Slaves that support data parity, check data parity when -DPAREN is active. In the event of a data parity error during a write, the slave activates -CHCK during the cycle to indicate to the master that an error occurred. Bit 7 in POS Register 5 is set to 0 in the slave when it activates -CHCK. See "Exception Reporting using -CHCK" on page 131 for more information. Slaves do not post an interrupt as a result of detecting a parity error. The slave determines what to do with the data. When sending data to a DMA slave, the DMA controller drives -TC active after detecting -CHCK. Alternatively, the DMA controller can drive -TC active on a subsequent operation with the DMA slave. When the DMA slave detects -TC, it terminates the data transfer and interrupts the system master.

Write Operations - Streaming Data Transfer

The timing of -DPAREN for streaming-data write operations is shown in Figure 72 on page 125. -DPAREN must remain active longer at the end of the cycle. The last data word is not valid until just before the rise of -CMD, and additional time is required for the slave to respond with -CHCK, if a parity error occurred. Slaves use -DPAREN to gate -CHCK onto the channel on streaming-data write operations, and the master must hold -DPAREN active until it can successfully sample -CHCK. Masters that support streaming data transfers with data parity drive -BURST on each transfer and keep it active to delay EOT. The EOT delay prevents conflicts between -DPAREN of the controlling master and the system master, which can execute cycles when ARB/GNT is in the arbitration state.

The participant receiving the data checks parity on the bytes that are signaled as valid by the data transfer control signals A0, -SBHE or -BE(0-3). The participant driving the data bus (and address bus for 64 bit streaming data) includes parity with the valid data bytes. When the master replicates data bytes for steering purposes, the replicated bytes contain valid parity also. When the system provides data steering the steering includes parity.

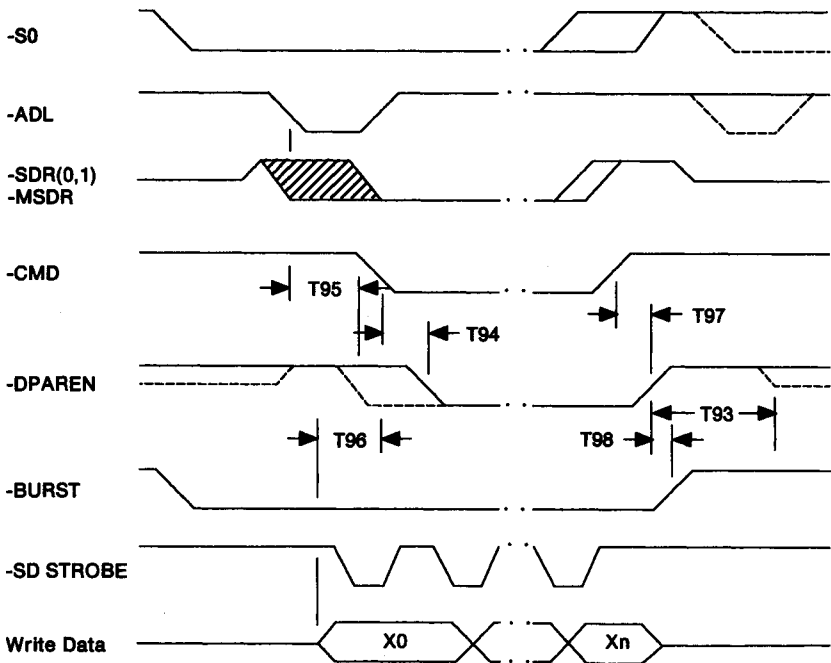


Figure 72. Data Parity Write Cycle - Streaming Data Transfer

Timing Parameter	Min/Max	Note	
T97	-DPAREN hold from -CMD inactive	40 / - ns	1
T98	-BURST inactive from -DPAREN inactive	0 / -	

Figure 73. Data Parity Checking Write Timing Specifications

Notes:

1. Masters must drive -DPAREN inactive before tri-stating it. (See T93 in Figure 67 on page 121).
2. See Figure 67 on page 121 and Figure 70 on page 123 for timing specifications not listed here.

Data Parity Exception Handling

The following is a summary of how participants handle data-parity exception conditions.

Bus Master

During a read cycle a master that supports data parity checking and detects a parity error should do one of the following:

- Suspend the data transfer and post an interrupt reporting a read data parity exception. The master determines what to do with the data.
- Complete the data transfer and post an interrupt reporting a read data parity exception. The master determines what to do with the data.

System Master

When a system master owns the channel and is enabled to support data parity, it must handle data parity errors.

DMA Controller

When a DMA controller that supports data parity detects a data parity error during a read cycle, it should record read data parity exception status and, if the DMA slave is selected, drive -TC active.

Alternatively, the DMA controller can drive -TC active on a subsequent operation with the DMA slave.

The system master reads the DMA controller and DMA slave exception status at the end of the DMA operation to determine if the operation completed successfully.

Slaves

When a selected slave detects a parity error during a write cycle, it should activate -CHCK and set bit 7 in POS Register 5 to 0 to indicate that an exception condition occurred (see "Exception Reporting using -CHCK" on page 131). The slave determines what to do with the data.

Address Parity

Address parity support is optional and allows the use of parity and non-parity adapters on the channel. For the exception to be detected and reported, both the master and slave involved in the operation must support address parity.

A controlling master that supports address parity drives -APAREN active to indicate that the parity is associated with the address. When a slave that supports address parity finds -APAREN active and an address parity error is detected, the slave is not selected. The slave includes address parity checking in its address decode. The following figure shows address parity timing requirements.

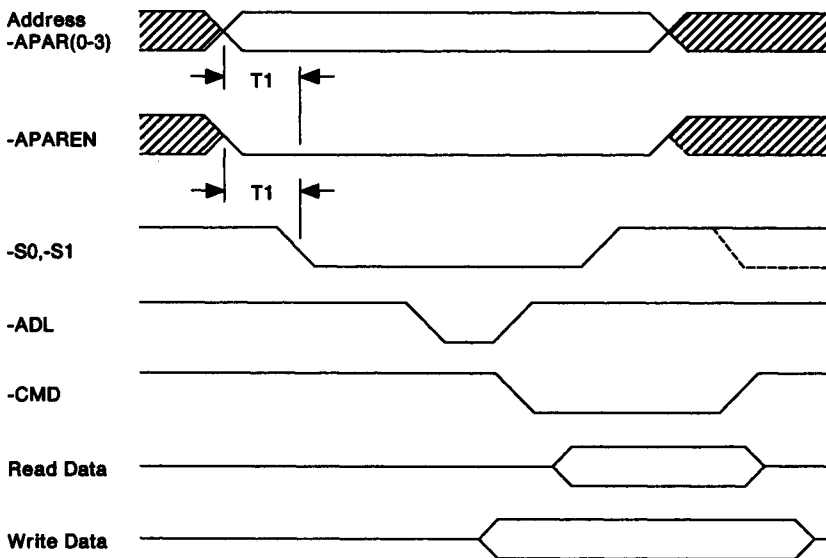


Figure 74. Address Parity

Timing Parameter	Min/Max	Note
T1	10 / - ns	Status active from Address Bus, M/-IO, -APAREN, and -REFRESH valid

Figure 75. Address Parity Timing Specifications

-APAREN and address parity bits are valid during the same interval that the address is valid (see Figure 23 on page 44). Only those address bytes used in the data-transfer cycle need to have correct parity. (See data parity for use of address parity bits during 64-bit streaming-data transfers).

Slaves that support address parity must include parity checking in their address decode logic so that, when parity errors occur, they do not become selected and do not drive -CD SFDBK. Slaves check address parity for all data transfers that have -APAREN active. Slaves that support address parity check each address and drive -CHCK active when an address parity error is detected. I/O slaves check parity on all I/O addresses, and memory slaves check parity on all memory addresses. They do not set the channel check status bit in POS Register 5.

Note: A slave treating the address bus as a command should ensure address parity is good before allowing the command to execute.

Channel Non-Parity Exceptions

These exceptions are channel exceptions other than data and address parity exceptions. These exceptions can be signaled by the current slave or detected by the master.

Invalid -DS 16 RTN and -DS 32 RTN Combination

If the system master or a bus master detects an inactive -DS 16 RTN and an active -DS 32 RTN, it suspends the data transfer and posts an interrupt, indicating with interrupt status that the operation was not valid. Detecting the invalid operation is optional.

The DMA controller drives -TC active during the cycle of the detected exception when the DMA slave is selected, and records the status of the invalid operation. Alternatively, the DMA controller can drive -TC active on a subsequent operation with the DMA slave. The timings for -TC are listed in Figure 45 on page 78, Figure 51 on page 89, and Figure 53 on page 90.

Selected Feedback Return Exception

Bus masters monitor -SFDBKRTN under control of a Selected Feedback Return Exception Enable field. Masters that do not support -SFDBKRTN do not need this POS bit. When the Selected Feedback Return Exception Enable field is enabled and -SFDBKRTN is inactive, the bus master sets appropriate status, suspends the data transfer and interrupts the system master.

The DMA controller monitors -SFDBKRTN under control of a Selected Feedback Return Exception Enable field. DMA controllers that do not support -SFDBKRTN do not need this POS bit. When the Selected Feedback Return Exception Enable field is enabled and -SFDBKRTN is inactive, the DMA controller sets the appropriate status and, if a DMA slave is selected, drives -TC active for that cycle. Alternatively, the DMA controller can drive -TC active on a subsequent operation with the DMA slave.

Systems can monitor the -CD SFDBK signals, or -SFDBKRTN, under control of the Selected Feedback Return Exception Enable field. When the Selected Feedback Return Exception Enable field is enabled and a -CD SFDBK is not returned, the system monitor should set appropriate status and generate an interrupt.

These types of exceptions can occur during either read or write operations.

Note: Masters must record the state of -SFDBKRTN when a channel check is detected regardless of the state of the Select Feedback Return Exception Enable field.

Master and Slave Dependent Exceptions

Master- and slave-dependent exceptions are not related to the channel but to the participants. Reporting of this type of exception is specific to each participant and to the specific exception condition.

This type of exception can occur during either read or write operations. A channel check can also occur. Therefore, a master must monitor -CHCK for exceptions during both read and write operations. This is required regardless of whether address parity or data parity are supported.

Master- and DMA slave-dependent exceptions should be reported with an interrupt. However, certain slave-dependent exceptions might require using -CHCK to report the exception. Examples of exceptions that could require using -CHCK are:

- Page fault
- Internal parity error (known bad data without an interrupt procedure to report the error)
- Command queue overflow
- Catastrophic exception condition.

Channel Timeout Exception

If a controlling master or DMA slave fails to release the channel within 7.8 microseconds of -PREEMPT going active, the central arbitration control point can drive ARB/-GNT to the ARB state. In this state, the controlling master degrades the address and data buses and all data transfer signals.

Note: Reporting and handling of this type of exception is system dependent.

Exception Reporting

A slave normally reports exceptions with synchronous -CHCK. A master normally reports exceptions with an interrupt and termination of current bus ownership. Exception reporting using -CHCK is described in the following information.

Exception Reporting using -CHCK

Exceptions detected by a slave or channel monitor are usually reported by activating a synchronous -CHCK. If -CHCK is activated, the system master or a bus master suspends processing and generates an interrupt. This interrupt invokes a utility program to handle the exception condition.

Any adapter that activates -CHCK sets bit 7 in POS Register 5 to a 0, to indicate the source of the exception in all cases except address parity errors. If multiple conditions can set -CHCK, the adapter can optionally set status information indicating the cause of the exception. The status or a pointer to it should be placed in POS Registers 6 and 7. Bit 6 in POS Register 5 is then set to 0 to indicate that status information is available. The system master can reset the -CHCK indication by setting a logical 1 into bit 7 in POS Register 5. Setting the bit to 0 will produce unpredictable results. Adapters that do not support -CHCK must ensure bit 7 in POS Register 5 is interpreted as a logical 1.

Synchronous and asynchronous reporting of exceptions using the -CHCK signal are supported. Both reporting methods are described in the following sections.

Synchronous -CHCK

Synchronous -CHCK is signaled by a slave or channel monitor with a pulse on -CHCK. The conditions that are reported with synchronous -CHCK are:

- Data parity errors
- Address parity errors
- Channel non-parity exceptions
- Slave-dependent exceptions.

Note: Slaves or channel monitors that report channel non-parity or slave-dependent exceptions with synchronous -CHCK must support the Channel Check Exception Reporting field (see "System Configuration" on page 101 for a description of this field).

Figure 76 shows a synchronous-exception presentation for the basic transfer cycles.

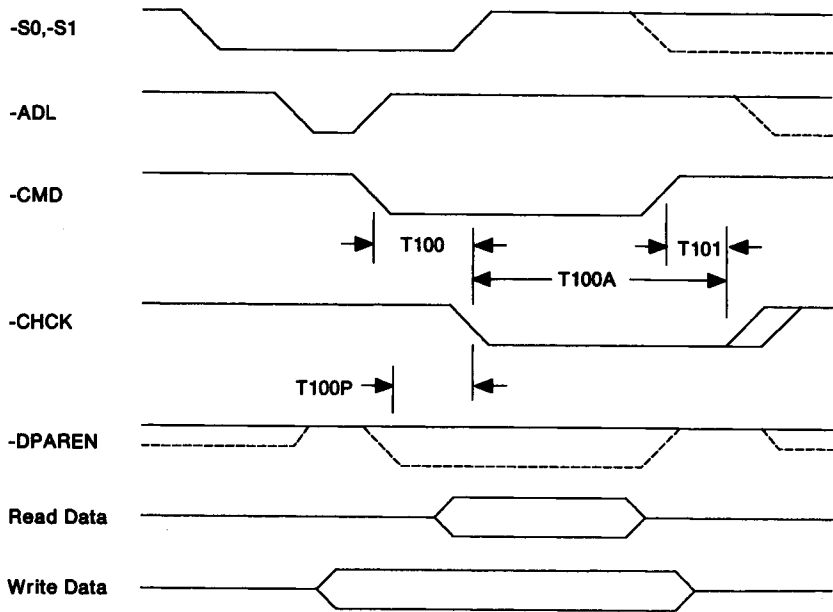


Figure 76. Channel-Check - Default Cycle

Timing Parameter	Min/Max	Note
T100 -CHCK active from -CMD active	0 / 50 ns	1
T100A -CHCK pulse width	50 / - ns	1
T100P -CHCK active from -CMD and -DPAREN active	0 / 50 ns	2
T101 -CHCK disabled from -CMD inactive	0 / 40 ns	

Figure 77. Channel-Check Timing Specifications - Default Cycle

Notes:

1. Slave exception reporting for exceptions other than channel data parity.
2. Activation of -CHCK because of a write-data parity error is timed from -CMD and -DPAREN active, whichever occurs last.
3. Using synchronous -CHCK to report an exception during a transfer that is caused by a condition other than those listed is timed from the falling edge of -CMD.

In a default cycle, the slave activates -CHCK from the activation of -CMD or -CMD and -DPAREN. If the slave requires additional time, it can extend the cycle by deactivating CD CHRDY. Figure 78 shows an extended bus cycle.

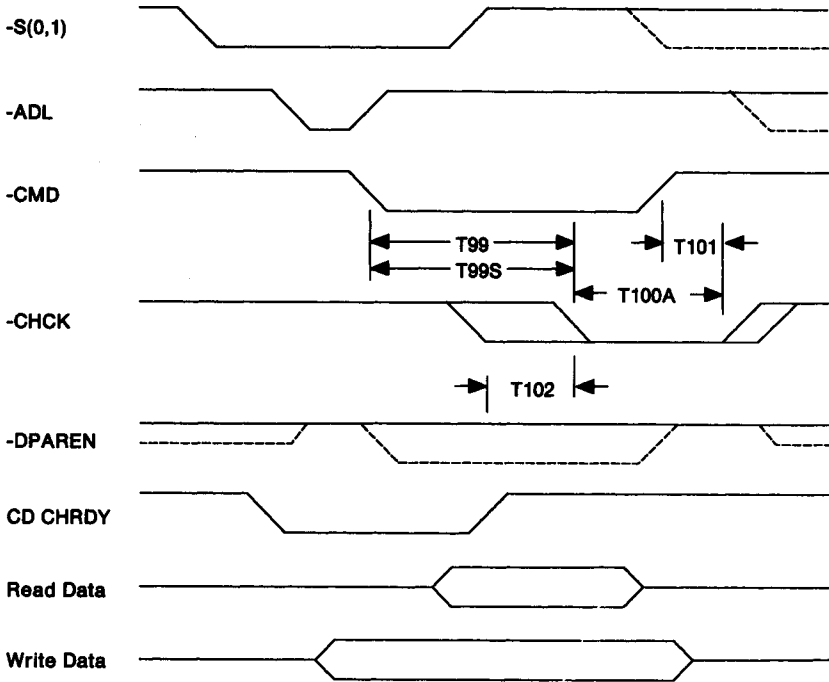


Figure 78. Channel-Check - Extended Cycle

Timing Parameter	Min/Max	Note	
T99	-CHCK active from -CMD active	0 / - ns	2
T99S	-CHCK active from -CMD active	0 / 150 ns	1
T102	-CHCK active from CD CHRDY active	- / 50 ns	2

Figure 79. Channel-Check Timing Specifications - Extended Cycle

Notes:

1. For a synchronous-extended cycle only.
2. For an asynchronous-extended cycle and streaming data.
3. See Figure 77 on page 133 for timing specifications not listed here.

Presentation of exceptions for streaming data cycles is similar to a basic transfer cycle. Since exceptions can also occur on any of the data cycles, -CHCK can be activated at any time during the cycle. During read operations, -CHCK is gated by -CMD and is required to be valid coincident with the final data word (Figure 80). During streaming data write operations, -CHCK is gated by -DPAREN and this signal is held active beyond the rise of -CMD to allow time for -CHCK to be activated on the channel (see Figure 72 on page 125).

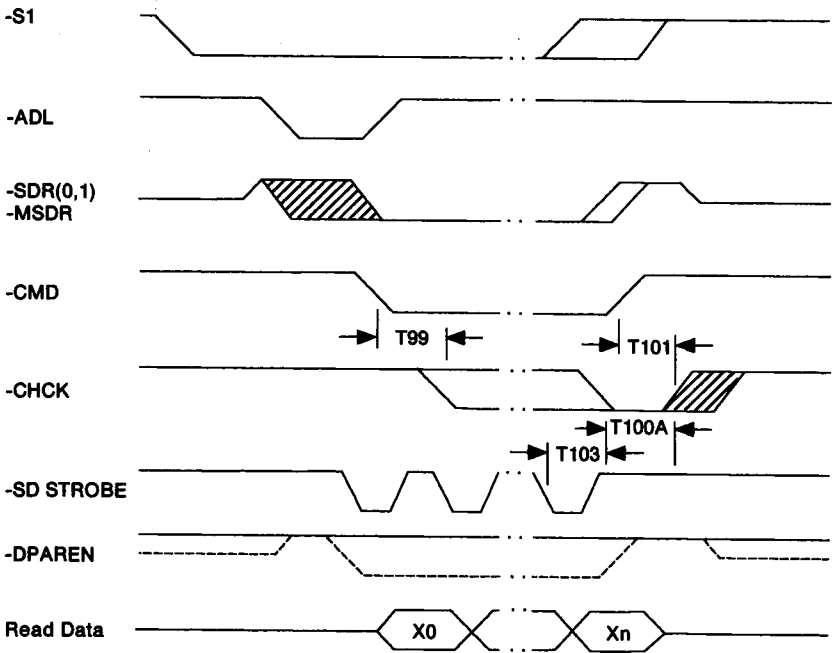


Figure 80. Channel Check - Streaming Data Read Cycle

Timing Parameter	Min/Max
T103 -CHCK active from last -SD STROBE active	- / 50 ns

Figure 81. Channel-Check Timing Specifications - Streaming Data Read Cycle

Note: See Figure 77 on page 133 and Figure 79 on page 134 for timing specifications not listed here.

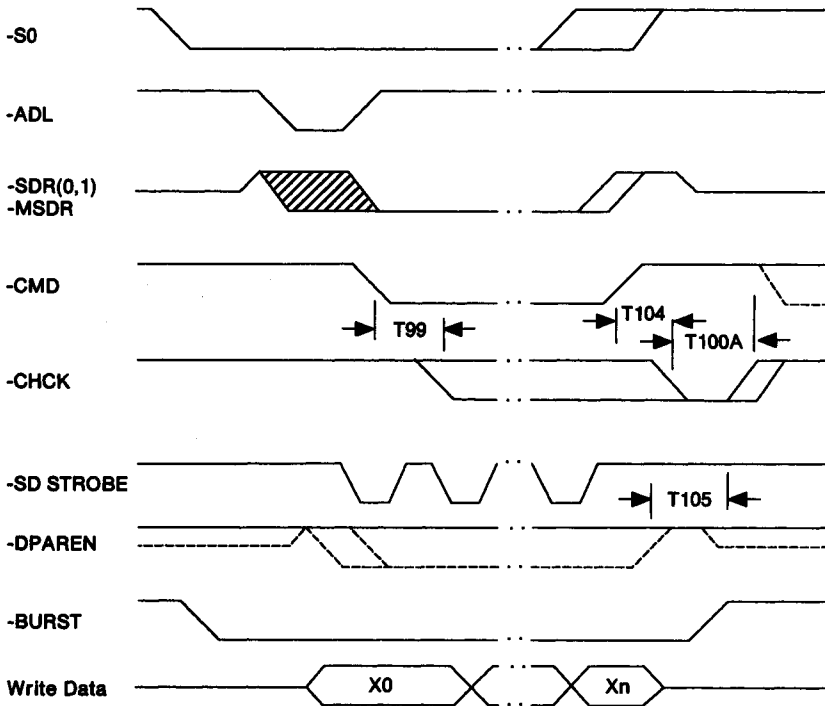


Figure 82. Channel Check - Streaming Data Write Cycle

Timing Parameter	Min/Max
T104 -CHCK active from -CMD inactive	- / 40 ns
T105 -CHCK disabled from -DPAREN inactive	0 / 40 ns

Figure 83. Channel-Check Timing Specifications - Streaming Data Write Cycle

Note: T100A, T101, and T103 apply to -CHCK for all conditions other than data parity exceptions.

Asynchronous -CHCK

An asynchronous -CHCK is indicated by driving -CHCK active asynchronously to a data transfer. -CHCK remains active until reset by the system master. The adapter that activates an asynchronous -CHCK sets bit 7 in POS Register 5 to a 0. The asynchronous -CHCK is reset by the system master setting bit 7 in POS Register 5, to a 1 (in the adapter that activated -CHCK). The time that asynchronous -CHCK remains active is system master dependent.

Because asynchronous exceptions are latched, -CHCK is detected during the reading of the adapter configuration registers to determine the source of the exception. System masters should ignore -CHCK while collecting status associated with the signaling of -CHCK.

| The conditions that can be reported with asynchronous -CHCK are:

- | • Channel non-parity exceptions
- | • Slave-dependent exceptions.

| If the exception cannot be reported with an interrupt or a synchronous -CHCK and the condition would otherwise go undetected, the exception can be reported with an asynchronous -CHCK. **Using asynchronous -CHCK can cause momentary suspension of all I/O activity and could require a system reset.**

Exception Reporting Using Interrupts

If an interrupt is used to report an exception, status must be supplied to indicate the type of exception being reported.

- The minimum status for bus masters is:
 - Data parity error during a read operation
 - -CHCK state
 - -SFDBKRTN state (if received).
- The minimum status for slaves is:
 - Data parity error during a write operation
 - Internal error.

Exception Handling

The following describes how participants handle -CHCK conditions.

Bus Master

The bus master monitors -CHCK when it is the controlling master. When the bus master detects -CHCK, it should do one of the following:

- Suspend the data transfer and post an interrupt.
- Complete the data transfer and post an interrupt.

Data sent by the bus master after it detects -CHCK might not be received. Data received by the bus master after it detects -CHCK might not be valid.

System Master

The system master monitors -CHCK when it is the controlling master. If -CHCK is detected, the system master will handle the exception condition. The system master can monitor -CHCK at all other times also. The action taken is system dependent.

The method for handling exceptions is software dependent and will include setting the -CHCK status bit to 1 in participants that have activated -CHCK, if exception recovery is part of the action taken.

DMA Controller

A DMA controller monitors -CHCK when it is the controlling master. When it detects -CHCK active, the DMA controller should record channel-check status, record the state of -SFDBKRTN, and, if the DMA slave is selected, drive -TC active. Alternatively, the DMA controller can drive -TC active on a subsequent operation with the DMA slave.

The system master reads the DMA controller and DMA slave exception status, when the end of the DMA operation is signaled, to determine if the operation completed successfully.

Slaves

When a slave activates -CHCK, it must set bit 7 in POS Register 5 to a 0 (except for address parity checking). The system master, upon resolving the -CHCK exception, will reset the bit to a 1.

Select Feedback Exception Handling

A -SFDBKRTN exception occurs when the controlling master does not detect -SFDBKRTN active (or -CD SFDBK for system masters). Reporting of this exception is optional. However, all masters that receive -SFDBKRTN must record its state when an exception occurs. Reporting -SFDBKRTN exceptions is controlled by the Select Feedback Return Exception Enable field. If the master does not receive the signal, the POS field and the monitoring of the signal is not required.

Note: All masters that support address parity must receive -SFDBKRTN.

The following describes how participants handle a -SFDBKRTN exception.

Bus Master

A bus master that is enabled to report a -SFDBKRTN exception and does not receive a -SFDBKRTN should do one of the following:

- Suspend the data transfer and post an interrupt
- Complete the data transfer and post an interrupt.

The status provided with the interrupt includes the lack of -SFDBKRTN. Data sent by the bus master after it detects the lack of -SFDBKRTN might not be received. Data received by the bus master after it detects the lack of -SFDBKRTN might not be valid.

System Master

Reporting a -SFDBKRTN exception and handling the exception is system dependent.

DMA Controller

A DMA controller that is enabled to report a -SFDBKRTN exception and receives no -SFDBKRTN should record status and, if the DMA slave is selected, drive -TC active. Alternatively, the DMA controller can drive -TC active on a subsequent operation with the DMA slave.

At the end of the DMA operation the system master reads the DMA controller and DMA slave status to determine if the operation completed successfully.

Channel Reset

CHRESET is used to force all system functions and channel participants to a defined state. The following situations cause a channel reset.

- A programmed-controlled channel-reset request
- A system power-on reset, which occurs:
 - When the system is powered on
 - When a low-voltage condition is detected at the input of the power supply.

CHRESET is active for a minimum amount of time, after all voltages in the system are within operating limits. See Figure 66 on page 116 for the T60 timing specifications.

After a channel reset or a system power-on reset, the state of the Micro Channel buses and signals are as follows:

Buses and Signals	System Power-On Reset		Channel Reset	
	Adapter	System	Adapter	System
Address Bus and Associated Signals	Tri-state	Initial	Tri-state	Unknown
Data Bus and Associated Signals	Tri-state	Initial	Tri-state	Unknown
Arbitration Bus and Associated Signals	Tri-state	Initial	Tri-state	Unknown
IRQ signal	Inactive	Initial	Inactive	Initial
Channel Check Signal	Inactive	Initial	Inactive	Initial
All Others	Inactive	Initial	Inactive	Unknown

Figure 84. Micro Channel Bus and Signal Status after a CHRESET

Connector Definition

The channel provides all signal, power, and ground signals to adapters through 50-mil channel connectors.

The channel provides two basic types of connectors:

- 16-bit, which support 8- and 16-bit operations
- 32-bit, which support 8-, 16-, 24-, and 32-bit operations.

Pins 01 through 45 support 8-bit operations. Pins 46 and 47 are keys (physical notch). Pins 48 through 58 provide additional power and signals to support 16-bit operations. For 32-bit adapters that function in 16-bit connectors, pins 59 and 60 are keys; for 32-bit adapters that cannot function in 16-bit connectors, pins 59 and 60 are reserved. Pins 61 through 89 are used with pins 01 through 58 to support 32-bit operations.

The power and ground pins on side A of each connector are offset from side B by 2 pins, and every fourth pin on either side of each connector is an ac ground.

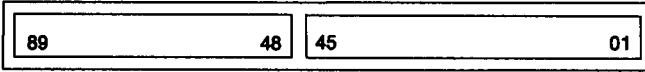
Video and matched-memory extensions to the basic 16- and 32-bit connectors are implemented on a system-by-system basis. For more information, see the technical reference for the specific system.

Note: Adapter designs should not extend the card-edge connector beyond the basic 16- or 32-bit connector unless the signals provided by the extension are used by the adapter.

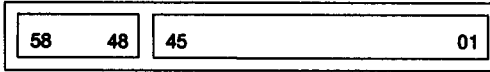
All of the logic signal lines are transistor-transistor logic (TTL) compatible.

The following figure shows the basic types of channel connectors with optional extensions.

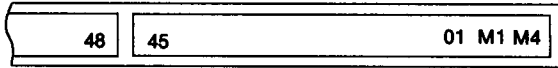
32-Bit Connector



16-Bit Connector

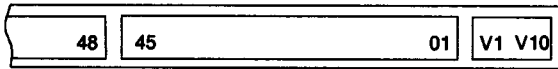


16 or 32-Bit Connector



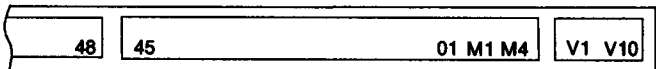
Matched Memory Extension (Optional)

16 or 32-Bit Connector



Auxiliary Video Extension (Optional)

16 or 32-Bit Connector



Base Video Extension (Optional)

Figure 85. Micro Channel Connectors

Warning: Any signals shown or described as “Reserved” should not 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.

Micro Channel Connector

| The following three figures show the signals and the voltages
| assigned to the 32-bit channel connector. The 16-bit connector is a
| subset of the 32-bit connector consisting of pins 1 through 58. A key
| is provided at pin locations 46 and 47 for mechanical alignment.

Rear of the System Board

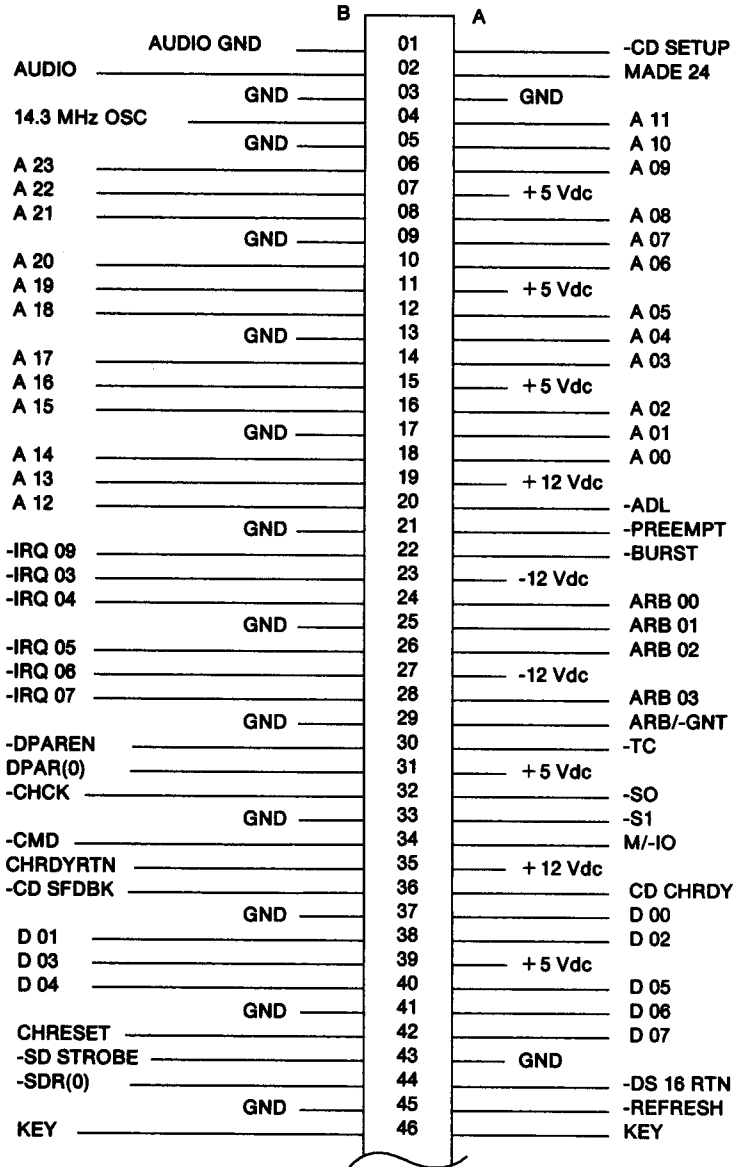


Figure 86. Micro Channel Connector

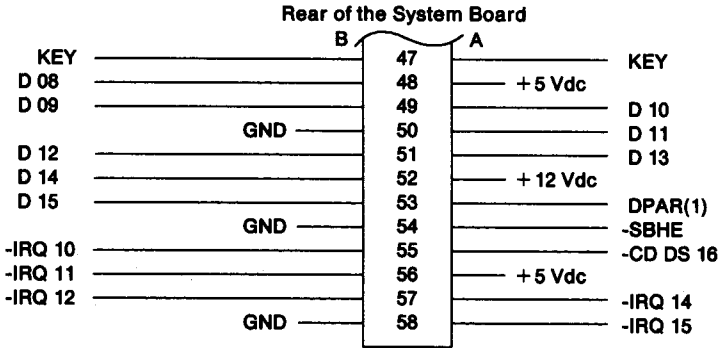


Figure 87. Micro Channel Connector - continued

Rear of the System Board

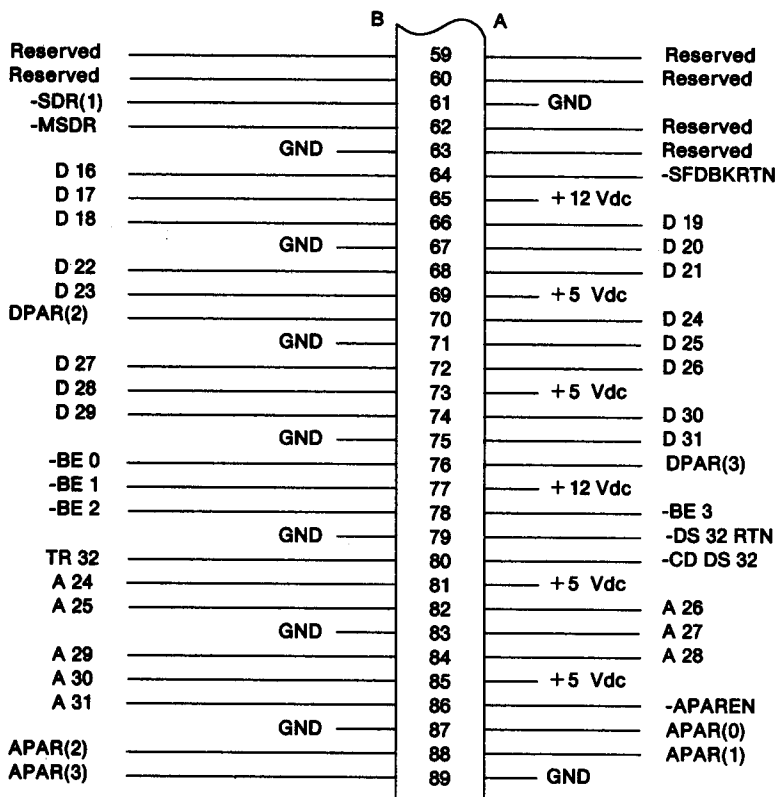


Figure 88. Micro Channel Connector - continued

Matched-Memory Extension

This extension provides additional signals to accommodate matched-memory cycles. The following figure shows a connector with a typical set of matched-memory signals. Refer to the system-specific technical reference for the system you are dealing with for further information.

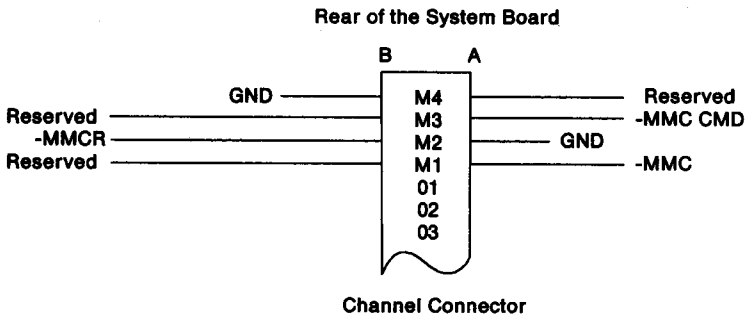


Figure 89. Channel Connector Voltage and Signal Assignments (Matched-Memory Extension)

Auxiliary Video Extension

This extension provides a video adapter with access to resources of the base video subsystem and allows it to extend or supplement the functions provided.

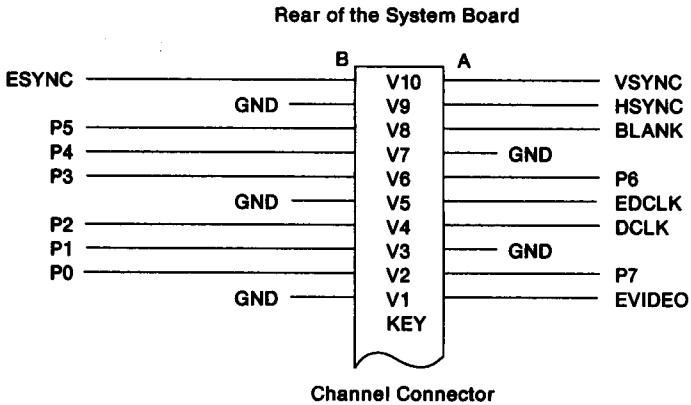
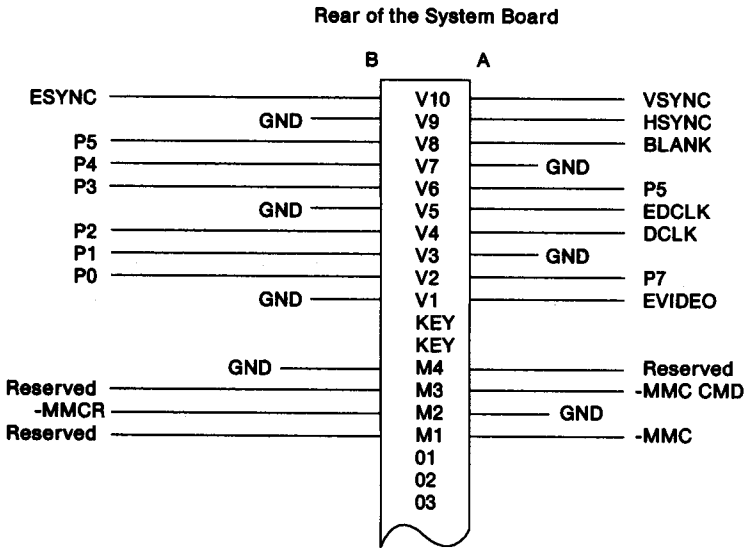


Figure 90. Auxiliary Video Extension

Base Video Extension

This extension is for adapters that provide the base video subsystem. Only systems without a base video subsystem on the system board have a connector with this extension.



Notes:

1. The KEY between M4 and V1 is a physical key.
2. Support of the matched-memory signals (M1 - M4) is optional.

Figure 91. Base Video Extension

Adapter Physical Specifications

The following are the physical specifications for Micro Channel adapters; these adapters can be Type 3, Type 3 half-card, or Type 5. The type number identifies the dimensions of the adapter. Systems specify the adapter type that they support.

Systems that specify support for the Type 3 adapter do not provide the space or the current capacity to accommodate Type 5 adapters (but can accommodate Type 3 half-card adapters). Systems that specify support for the Type 5 adapter support the Type 3, Type 3 half-card, and Type 5 adapters.

| Adapter Size

| The following are overall dimensions for Micro Channel adapters.
| Additional and more detailed information (plus tolerances) is
| available in Micro Channel Adapter Design.

| Type 3 Adapter

- | • Card height - 3.475 in.
- | • Card plus retainer length - 12.3 in.

| Type 3 Half-card Adapter

| The Type 3 half card adapter does not require a retainer.

- | • Card height - 3.475 in.
- | • Card plus retainer length - 6.35 in.

| Type 5 Adapter

| The Type 5 adapter does not require a retainer.

- | • Card height - 4.825 in.
- | • Card length - 13.1 in.

Connector Spacing

- Minimum Spacing from Card Center to Card Center - 0.85 in.
(Maximum spacing is system dependent.)
- Maximum Component Height (Measured from card surface.)
 - A side (without electrical insulation) = 0.6 in
 - A side (with electrical insulation) = 0.675 in
 - B side = 0.105 in.

Electrical insulation is also required if the card bows in excess of 0.025 inch. Measure the bow at a point 0.1 inch from the top of the card and 5.75 inches from the connector end (rear) of the card.

Power

The following figures list the minimum available load current for each voltage present on each connector. Some systems can provide more current than the amount specified below. For more information, refer to the system-specific technical manual.

Supply Voltage	Minimum Available Current Per 16-Bit Connector	Minimum Available Current Per 32-Bit Connector
+ 5.0 Vdc	1.6 A	2.0 A
+12.0 Vdc	0.175 A	0.175 A
-12.0 Vdc	0.040 A	0.040 A

Figure 92. Minimum current available for the Type 3 adapter.

Supply Voltage	Minimum Available Current Per 16-Bit Connector	Minimum Available Current Per 32-Bit Connector
+ 5.0 Vdc	3.12 A	3.12 A
+12.0 Vdc	0.25 A	0.25 A
-12.0 Vdc	0.10 A	0.10 A

Figure 93. Minimum current available for the Type 5 adapter

The following formulas can be used to determine maximum *statistical* current values.

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

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 =

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

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,	N/A
	B33, B37, B41, A43, B45, B50, B54, B58	N/A
	A61, B63, B67, B71, B75, B79, B83, B87, A89, BM4*, AM2*	N/A
+ 5.0 Vdc	A7, A11, A15, A31, A39, A48, A56, A69, A73, A81, A85	+5% -4.5%
+12.0 Vdc	A19, A35, A52, A65, A77,	+5% -4.5%
-12.0 Vdc	A23, A27	+9.5% -10%
		(-10.86 to -13.2 Vdc)
Ground (Auxiliary Video)	BV1, AV3, BV5, AV7, BV9	N/A

* These pins are in the Matched-Memory Extension.

Figure 94. 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), or the drop due to distribution within the adapter.

Index

A

adapter configuration data 103
adapter ID field 104
adapter I/O address select field 105
adapter ROM/RAM address select field 105
adapters, multiple card 102
address boundary alignment, streaming transfer 64
address bus signals 9
address bus translator 55
address decode latch signal 9
address management, streaming data 63
address parity 127
address parity enable field 106
address parity enable signal 9
address parity signals 10
address space, POS 103
addressing model 6
ADL 9
APAREN 9
APAR0 – APAR3 10
arbiter logic 31
arbiter mismatch 30
arbitration 27
arbitration block diagram 29
arbitration bus 7
arbitration cycle 35
arbitration level field 105
arbitration timing 35
arbitration/grant signal 8
ARB0 - ARB3 7
ARB/-GNT 8
architecture extensions 26
asynchronous -CHCK 137
asynchronous-extended cycle 50

AUDIO 19
AUDIO GND 19
audio ground 19
audio sum node signal 19
auto-increment enable field 105
auxiliary video extension 149
auxiliary video signal 23
A0 - A31 9

B

base architecture, 16-bit 26
base architecture, 32-bit 25
base video extension 149
basic transfer 40
BE signals 55, 56
BE0 - BE3 13
burst DMA transfer 90
burst mode 39
burst signal 8
burst transfer 8, 32, 86
burst transfer, preempted 33
bus master 3
bus, arbitration 7
byte enable 55
byte enable signals 13

C

card data size 16 signal 14
card data size 32 signal 14
card enable field 104
card selected feedback signal 10
card setup signal 19
CD CHRDY 14
CD DS 16 14
CD DS 32 14
CD SETUP 19
CD SFDBK 10

- central arbitration control point 28
- central data steering 56
- central steering logic 56
- central translator logic 56
- channel check field 104
- channel check signal 19
- channel connector definition 142
- channel connector diagram 143
- channel connector voltages and signal assignments 144
- channel connectors 142
- channel definition 7
- channel load current 152
- channel ownership 27
- channel ready return signal 14
- channel ready signal 14
- channel reset 141
- channel reset signal 19
- channel signal groups 20
- channel support 55
- channel time-out 34
- channel timeout exception 130
- CHCK 19
- CHCK field 104
- CHRDYRTN 14
- CHRESET 19
- CHRESET signal 141
- CMD 14
- command signal 14
- conditionally-required fields 104
- configuration data 103
- configuration fields 103
- configuration, system 101
- connectors, channel 142
- connector, auxiliary video extension 149
- connector, base video extension 149
- connector, matched-memory extension 148
- connector, 16 bit 144
- connector, 32 bit 144

- controlling master 27
- cycle, arbitration 35
- cycle, asynchronous-extended 50
- cycle, default 43
- cycle, synchronous-extended 48

D

- data and address parity enable field 119
- data bus signals 13
- data bus signals (0 - 31) 13
- data bus steering 57
- data bus steering control 54
- data parity 120
- data parity enable field 106
- data parity enable signal 15
- data parity exception handling 126
- data parity read operations 120
- data parity signals (0 - 3) 15
- data parity write operations 122
- data port 3
- data replication, data bus 55
- data size 16 return signal 16
- data size 32 return signal 16
- data steering 53
- data transfer cycle 39
- default cycle 43
- default master 27
- definition, channel 7
- description, micro channel 1
- device I/O address assignment field 106
- direct memory access (DMA) procedure 83
- DMA controller 3
- DMA cycle termination 86
- DMA slave 4
- DMA use of -BURST 87
- DPAREN 15
- DPAR0 - DPAR3 15
- DS 16 RTN 16
- DS 32 RTN 16

D0 - D31 13

E

end of transfer 28, 86
EOT 28
error condition 118
exception condition reporting 118
exception conditions 118
exception handling 138
exception reporting 106, 131
exception reporting using external interrupt 138
exception, channel time-out 34
exiting inactive state 35
extensions, architecture 26

F

fairness enable field 108
fairness feature 34
first cycle after grant 37
formula, maximum statistical current 152
formula, total channel current 153

H

handling, exception condition 118

I

inactive state 34
inactive state, exiting 35
interrupt 98
interrupt signals 18
IRQ 14-15 18
IRQ 3-7 18
IRQ 9-12 18
I/O address decode 113
I/O slave 4

L

load current 152
local arbiters 28, 30

M

MADE 24 10
masters 2
master, controlling 27
master, default 27
matched memory enable field 107
matched-memory connector 148
matched-memory signal 23
maximum statistical current formula 152
memory address enable 24
 signal 10
memory slave 4
memory/-input output signal 11
micro channel description 1
MSDR 16
multiple card adapters 102
multiple data transfers 28
multiplexed streaming data request signal 16
M/-IO 11

O

option select data fields 108
optional configuration fields 108
OSC 19
oscillator signal 19
ownership, channel 27

P

parity errors 119
POS I/O address selection 112
POS I/O address space 103
POS registers 103
power on reset 141

power specifications 152
PREEMPT 8, 33
preempt signal 8
preempted burst transfer 33
preemption 33
primary cards 102

R

REFRESH 19
refresh signal 19
regulation, voltage 153
reporting procedure, exception
 condition 118
required configuration fields 103
reserved signals 143
reset 141

S

SBHE 16
SD STROBE 16
SDR(0,1) 16
secondary cards 102
select feedback exception 140
selected feedback return exception
 enable field 107
selected feedback return signal 11
setup cycle timing 114
SFDBKRTN 11
signal descriptions 7
signal groups 20
signals
 -ADL 9
 -APAREN 9
 -BE0 - -BE3 13
 -BURST 8
 -CD DS 16 14
 -CD DS 32 14
 -CD SETUP 19
 -CD SFDBK 10
 -CHCK 19
 -CMD 14
 -DPAREN 15

signals (*continued*)

-DS 16 RTN 16
-DS 32 RTN 16
-MSDR 16
-PREEMPT 8
-REFRESH 19
-SD STROBE 16
-SDR(0,1) 16
-SFDBKRTN 11
-S0, -S1 17
-TC 18
address bus 9
APAR0 - APAR3 10
ARB0 - ARB3 7
ARB/-GNT 8
AUDIO 19
AUDIO GND 19
auxiliary video 23
CD CHRDY 14
CHRDYRTN 14
CHRESET 19
data bus 13
DPAR0 - DPAR3 15
D0 - D31 13
IRQ 14-15 18
IRQ 3-7 18
IRQ 9-12 18
MADE 24 10
matched-memory 23
M/-IO 11
OSC 19
TR 32 18
signals, address bus 9
signals, arbitration 7
signals, interrupt 18
signals, reserved 143
single DMA transfer 88
single transfer 85
slaves 3
specifications, power 152
status field 106
status information field 111

- status signals 17
- steering control, data bus 54
- streaming data address management 63
- streaming data cycle termination 74
- streaming data enable field 108
- streaming data request signals 16
- streaming data request 1 signal 17
- streaming data strobe signal 16
- streaming data timing specifications 77
- streaming data transfer 58
- streaming transfer, address boundary alignment 64
- synchronous -CHCK 132
- synchronous-extended cycle 48
- system byte high enable signal 16
 - refid.-SBHE 16
- system configuration 101
- system configuration fields 103
- system configuration procedure 112
- system configuration timing 114
- system master 3
- S0, S1 17

T

- TC 18
- terminal count signal 18
- termination, DMA cycle 86
- termination, streaming data 74
- timing diagrams, system configuration 114
- timing, first cycle after grant 37
- total channel current formula 153
- TR 32 18
- TR 32 signal 56
- translate 32 signal 18
- translator, address bus 55

V

- voltage regulation 153

Numerics

- 16-bit base architecture 26
- 16-bit connectors 142
- 32-bit base architecture 25
- 32-bit connectors 142