

- [Log in / create account](#)
- [Connectors](#)
- [Computers](#)
- [Cables](#)
- [Adapters](#)
- [Circuits](#)
- [Tables](#)
- [RND](#)

ISA

From HwB

ISA=Industry Standard Architecture

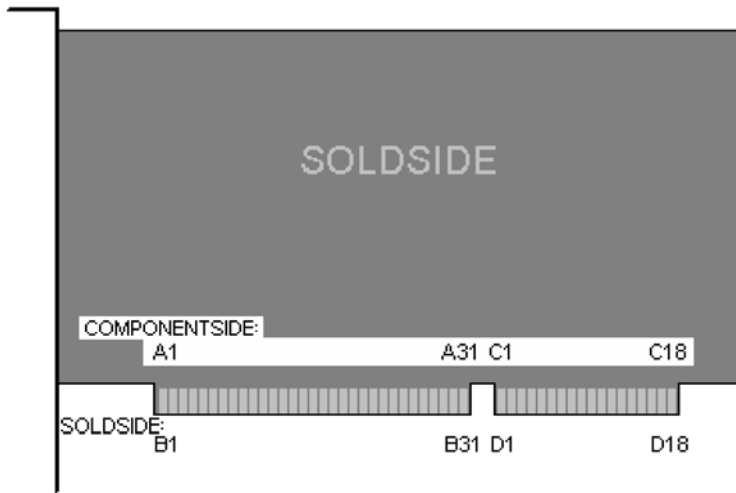
Developed by IBM.

Contents

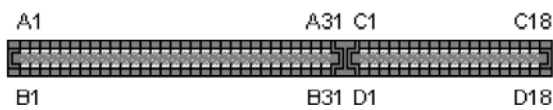
- 1 Pinout
- 2 Technical
 - 2.1 Physical Design
 - 2.1.1 8-bit card
 - 2.1.2 16-bit card
 - 2.2 Signal Descriptions
 - 2.2.1 +5, -5, +12, -12
 - 2.2.2 AEN
 - 2.2.3 BALE
 - 2.2.4 BCLK
 - 2.2.5 DACK_x
 - 2.2.6 DRQ_x
 - 2.2.7 IOCS16
 - 2.2.8 I/O CH CK
 - 2.2.9 I/O CH RDY
 - 2.2.10 IOR
 - 2.2.11 IOW
 - 2.2.12 IRQ_x
 - 2.2.13 LA_{xx}
 - 2.2.14 MASTER
 - 2.2.15 MEMCS16
 - 2.2.16 MEMR
 - 2.2.17 MEMW
 - 2.2.18 NOWS
 - 2.2.19 OSC
 - 2.2.20 REFRESH
 - 2.2.21 RESET
 - 2.2.22 SA0-SA19
 - 2.2.23 SBHE
 - 2.2.24 SD0-SD16
 - 2.2.25 SMEMR
 - 2.2.26 SMEMW
 - 2.2.27 T/C
 - 2.3 8 Bit Memory or I/O Transfer Timing Diagram (4 wait states shown)
 - 2.4 16 Bit Memory or I/O Transfer Timing Diagram (1 wait state shown)
 - 2.5 Shortening or Lengthening the bus cycle
 - 2.6 I/O Port Addresses
 - 2.7 DMA Read and Write
 - 2.8 Slave DMA Controller
 - 2.9 Master DMA Controller
 - 2.10 Single Transfer Mode
 - 2.11 Block Transfer Mode
 - 2.12 Demand Transfer Mode
 - 2.13 Interrupts on the ISA bus
 - 2.14 Bus Mastering

- 3 Contributions
- 4 Sources

Pinout



62+36 PIN EDGE CONNECTOR MALE at the card.



62+36 PIN EDGE CONNECTOR FEMALE at the computer.

Pin	Name	Dir	Description
A1	/I/O CH CK	←	I/O channel check; active low=parity error
A2	D7	↔	Data bit 7
A3	D6	↔	Data bit 6
A4	D5	↔	Data bit 5
A5	D4	↔	Data bit 4
A6	D3	↔	Data bit 3
A7	D2	↔	Data bit 2
A8	D1	↔	Data bit 1
A9	D0	↔	Data bit 0
A10	I/O CH RDY	←	I/O Channel ready, pulled low to lengthen memory cycles
A11	AEN	→	Address enable; active high when DMA controls bus
A12	A19	→	Address bit 19
A13	A18	→	Address bit 18

A14	A17	➡	Address bit 17
A15	A16	➡	Address bit 16
A16	A15	➡	Address bit 15
A17	A14	➡	Address bit 14
A18	A13	➡	Address bit 13
A19	A12	➡	Address bit 12
A20	A11	➡	Address bit 11
A21	A10	➡	Address bit 10
A22	A9	➡	Address bit 9
A23	A8	➡	Address bit 8
A24	A7	➡	Address bit 7
A25	A6	➡	Address bit 6
A26	A5	➡	Address bit 5
A27	A4	➡	Address bit 4
A28	A3	➡	Address bit 3
A29	A2	➡	Address bit 2
A30	A1	➡	Address bit 1
A31	A0	➡	Address bit 0
B1	GND		Ground
B2	RESET	➡	Active high to reset or initialize system logic
B3	+5V		+5 VDC
B4	IRQ2	←	Interrupt Request 2
B5	-5VDC		-5 VDC
B6	DRQ2	←	DMA Request 2
B7	-12VDC		-12 VDC
B8	/NOWS	←	No WaitState
B9	+12VDC		+12 VDC
B10	GND		Ground
B11	/SMEMW	➡	System Memory Write
B12	/SMEMR	➡	System Memory Read
B13	/IOW	➡	I/O Write
B14	/IOR	➡	I/O Read
B15	/DACK3	➡	DMA Acknowledge 3
B16	DRQ3	←	DMA Request 3
B17	/DACK1	➡	DMA Acknowledge 1

B18	DRQ1	←	DMA Request 1
B19	/REFRESH	↔	Refresh
B20	CLOCK	→	System Clock (67 ns, 8-8.33 MHz, 50% duty cycle)
B21	IRQ7	←	Interrupt Request 7
B22	IRQ6	←	Interrupt Request 6
B23	IRQ5	←	Interrupt Request 5
B24	IRQ4	←	Interrupt Request 4
B25	IRQ3	←	Interrupt Request 3
B26	/DACK2	→	DMA Acknowledge 2
B27	T/C	→	Terminal count; pulses high when DMA term. count reached
B28	ALE	→	Address Latch Enable
B29	+5V		+5 VDC
B30	OSC	→	High-speed Clock (70 ns, 14.31818 MHz, 50% duty cycle)
B31	GND		Ground
C1	SBHE	↔	System bus high enable (data available on SD8-15)
C2	LA23	↔	Address bit 23
C3	LA22	↔	Address bit 22
C4	LA21	↔	Address bit 21
C5	LA20	↔	Address bit 20
C6	LA18	↔	Address bit 19
C7	LA17	↔	Address bit 18
C8	LA16	↔	Address bit 17
C9	/MEMR	↔	Memory Read (Active on all memory read cycles)
C10	/MEMW	↔	Memory Write (Active on all memory write cycles)
C11	SD08	↔	Data bit 8
C12	SD09	↔	Data bit 9
C13	SD10	↔	Data bit 10
C14	SD11	↔	Data bit 11
C15	SD12	↔	Data bit 12
C16	SD13	↔	Data bit 13
C17	SD14	↔	Data bit 14
C18	SD15	↔	Data bit 15
D1	/MEMCS16	←	

			Memory 16-bit chip select (1 wait, 16-bit memory cycle)
D2	/IOCS16	←	I/O 16-bit chip select (1 wait, 16-bit I/O cycle)
D3	IRQ10	←	Interrupt Request 10
D4	IRQ11	←	Interrupt Request 11
D5	IRQ12	←	Interrupt Request 12
D6	IRQ15	←	Interrupt Request 15
D7	IRQ14	←	Interrupt Request 14
D8	/DACK0	→	DMA Acknowledge 0
D9	DRQ0	←	DMA Request 0
D10	/DACK5	→	DMA Acknowledge 5
D11	DRQ5	←	DMA Request 5
D12	/DACK6	→	DMA Acknowledge 6
D13	DRQ6	←	DMA Request 6
D14	/DACK7	→	DMA Acknowledge 7
D15	DRQ7	←	DMA Request 7
D16	+5 V		
D17	/MASTER	←	Used with DRQ to gain control of system
D18	GND		Ground

Notes:

- Direction is Motherboard relative ISA-Cards.
- B8 was /CARD SLCDTD on the XT. Card selected, activated by cards in XT's slot J8

Technical

This file is designed to give a basic overview of the bus found in most IBM clone computers, often referred to as the XT or AT bus. The AT version of the bus is upwardly compatible, which means that cards designed to work on an XT bus will work on an AT bus. This bus was produced for many years without any formal standard. In recent years, a more formal standard called the ISA bus (Industry Standard Architecture) has been created, with an extension called the EISA (Extended ISA) bus also now as a standard. The EISA bus extensions will not be detailed here.

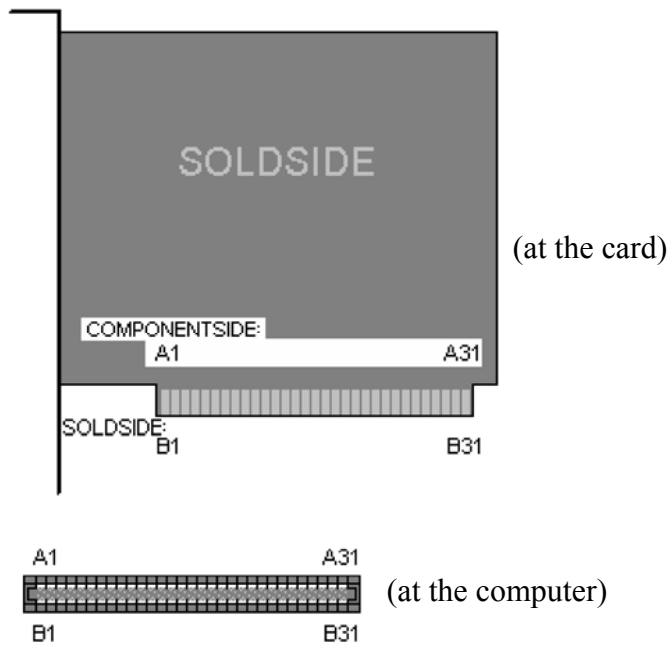
This file is not intended to be a thorough coverage of the standard. It is for informational purposes only, and is intended to give designers and hobbyists sufficient information to design their own XT and AT compatible cards.

Physical Design

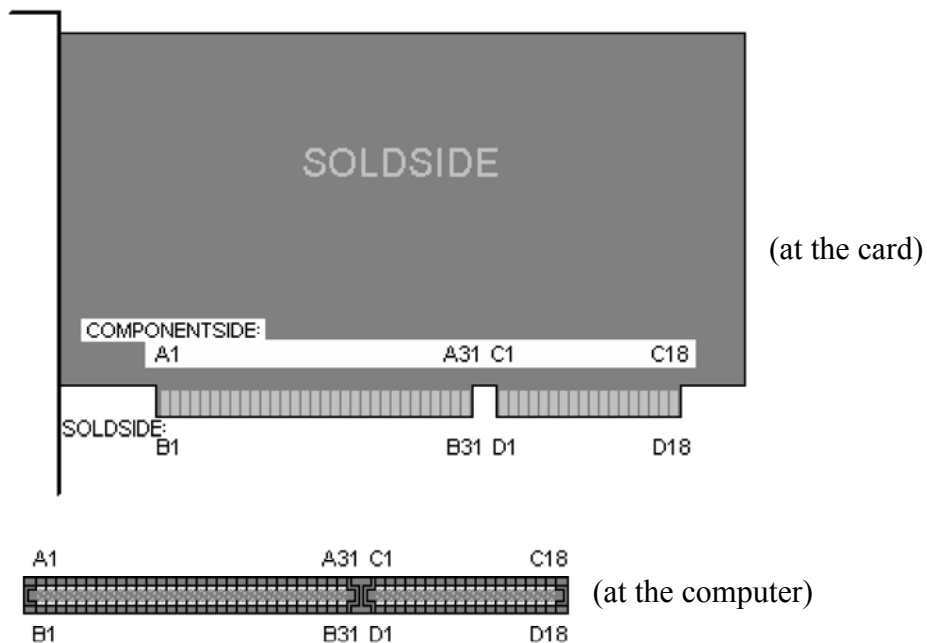
ISA cards can be either 8-bit or 16-bit. 8-bit cards only uses the first 62 pins and 16-bit cards uses all 98

pins. Some 8-bit cards uses some of the 16-bit extension pins to get more interrupts.

8-bit card



16-bit card



Signal Descriptions

+5, -5, +12, -12

Power supplies. -5 is often not implemented.

AEN

Address Enable. This is asserted when a DMAC has control of the bus. This prevents an I/O device from responding to the I/O command lines during a DMA transfer. When AEN is active, the DMA Controller has control of the address bus as the memory and I/O read/write command lines.

BALE

Bus Address Latch Enable. The address bus is latched on the rising edge of this signal. The address on the SA bus is valid from the falling edge of BALE to the end of the bus cycle. Memory devices should latch the LA bus on the falling edge of BALE. Some references refer to this signal as Buffered Address Latch Enable, or just Address Latch Enable (ALE). The Buffered-Address Latch Enable is used to latch SA0-19 on the falling edge. This signal is forced high during DMA cycles.

BCLK

Bus Clock, 33% Duty Cycle. Frequency Varies. 4.77 to 8 MHz typical. 8.3 MHz is specified as the maximum, but many systems allow this clock to be set to 12 MHz and higher.

DACKx

DMA Acknowledge. The active-low DMA Acknowledge 0 to 3 and 5 to 7 are the corresponding acknowledge signals for DRQ 0-3, 5-7.

DRQx

DMA Request. These signals are asynchronous channel requests used by I/O channel devices to gain DMA service. DMA request channels 0-3 are for 8-bit data transfer. DAM request channels 5-7 are for 16-bit data transfer. DMA request channel 4 is used internally on the system board. DMA requests should be held high until the corresponding DACK line goes active. DMA requests are serviced in the following priority sequence:

High: DRQ 0, 1, 2, 3, 5, 6, 7 Lowest

IOCS16

I/O size 16. Generated by a 16 bit slave when addressed by a bus master. The active-low I/O Chip Select 16 indicates that the current transfer is a 1 wait state, 16 bit I/O cycle. Open Collector.

I/O CH CK

Channel Check. A low signal generates an NMI. The NMI signal can be masked on a PC, externally to the processor (of course). Bit 7 of port 70(hex) (enable NMI interrupts) and bit 3 of port 61 (hex) (recognition of channel check) must both be set to zero for an NMI to reach the cpu. The I/O Channel Check is an active-low signal which indicates that a parity error exists in a device on the I/O channel.

I/O CH RDY

Channel Ready. Setting this low prevents the default ready timer from timing out. The slave device may then set it high again when it is ready to end the bus cycle. Holding this line low for too long (15 microseconds, typical) can prevent RAM refresh cycles on some systems. This signal is called IOCHRDY (I/O Channel Ready) by some references. CHRDY and NOWS should not be used simultaneously. This may cause problems with some bus controllers. This signal is pulled low by a memory or I/O device to lengthen memory or I/O read/write cycles. It should only be held low for a minimum of 2.5 microseconds.

IOR

The I/O Read is an active-low signal which instructs the I/O device to drive its data onto the data bus, SD0-SD15.

IOW

The I/O Write is an active-low signal which instructs the I/O device to read data from the data bus, SD0-SD15.

IRQx

Interrupt Request. IRQ2 has the highest priority. IRQ 10-15 are only available on AT machines, and are higher priority than IRQ 3-7. The Interrupt Request signals which indicate I/O service attention. They are prioritized in the following sequence: Highest IRQ 9(2),10,11,12,14,3,4,5,6,7

LAXx

Latchable Address lines. Combine with the lower address lines to form a 24 bit address space (16 MB) These unlatched address signals give the system up to 16 MB of address ability. They are valid when "BALE" is high.

MASTER

16 bit bus master. Generated by the ISA bus master when initiating a bus cycle. This active-low signal is used in conjunction with a DRQ line by a processor on the I/O channel to gain control of the system. The I/O processor first issues a DRQ, and upon receiving the corresponding DACK, the I/O processor may assert MASTER, which will allow it to control the system address, data and control lines. This signal should not be asserted for more than 15 microseconds, or system memory may be corrupted due to the lack of memory refresh activity.

MEMCS16

The active-low Memory Chip Select 16 indicates that the current data transfer is a 1 wait state, 16 bit data memory cycle.

MEMR

The Memory Read is an active-low signal which instructs memory devices to drive data onto the data bus SD0-SD15. This signal is active on all memory read cycles.

MEMW

The Memory Write is an active-low signal which instructs memory devices to store data present on the data bus SD0-SD15. This signal is active on all memory write cycles.

NOWS

No Wait State. Used to shorten the number of wait states generated by the default ready timer. This causes the bus cycle to end more quickly, since wait states will not be inserted. Most systems will ignore NOWS if CHRDY is active (low). However, this may cause problems with some bus controllers, and both signals should not be active simultaneously.

OSC

Oscillator, 14.31818 MHz, 50% Duty Cycle. Frequency varies. This was originally divided by 3 to provide the 4.77 MHz cpu clock of early PCs, and divided by 12 to produce the 1.19 MHz system clock. Some references have placed this signal as low as 1 MHz (possibly referencing the system clock), but most modern systems use 14.318 MHz.

This frequency (14.318 MHz) is four times the television colorburst frequency. Refresh timing on many PC's is based on OSC/18, or approximately one refresh cycle every 15 microseconds. Many modern motherboards allow this rate to be changed, which frees up some bus cycles for use by software, but also can cause memory errors if the system RAM cannot handle the slower refresh rates.

REFRESH

Refresh. Generated when the refresh logic is bus master. This active-low signal is used to indicate a memory refresh cycle is in progress. An ISA device acting as bus master may also use this signal to initiate a refresh cycle.

RESET

This signal goes low when the machine is powered up. Driving it low will force a system reset. This signal goes high to reset the system during powerup, low line-voltage or hardware reset. ??????????????

SA0-SA19

System Address Lines, tri-state. The System Address lines run from bit 0 to bit 19. They are latched on to the falling edge of "BALE".

SBHE

System Bus High Enable, tri-state. Indicates a 16 bit data transfer. The System Bus High Enable indicates high byte transfer is occurring on the data bus SD8-SD15. This may also indicate an 8 bit transfer using the upper half of the bus data (if an odd address is present).

SD0-SD16

System Data lines, or Standard Data Lines. They are bidirectional and tri-state. On most systems, the data lines float high when not driven. These 16 lines provide for data transfer between the processor, memory and I/O devices.

SMEMR

System Memory Read Command line. Indicates a memory read in the lower 1 MB area. This System Memory Read is an active-low signal which instructs memory devices to drive data onto the data bus SD0-SD15. This signal is active only when the memory address is within the lowest 1MB of memory address space.

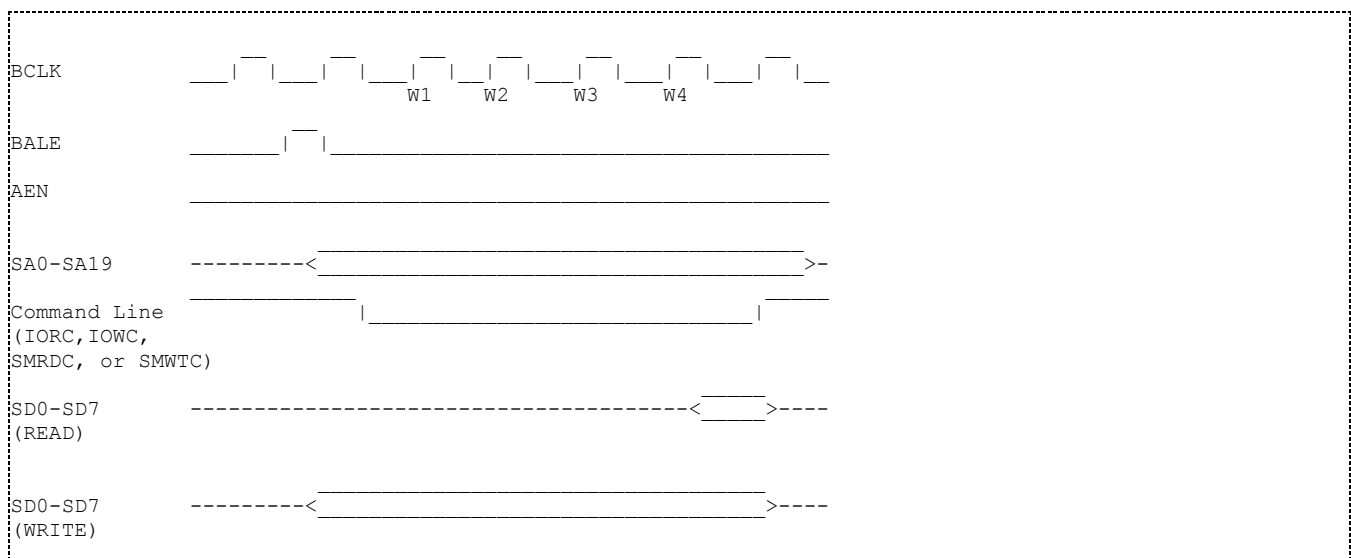
SMEMW

System Memory Write Command line. Indicates a memory write in the lower 1 MB area. The System Memory Write is an active-low signal which instructs memory devices to store data preset on the data bus SD0-SD15. This signal is active only when the memory address is within the lowest 1MB of memory address space.

T/C

Terminal Count. Notifies the cpu that that the last DMA data transfer operation is complete. Terminal Count provides a pulse when the terminal count for any DMA channel is reached.

8 Bit Memory or I/O Transfer Timing Diagram (4 wait states shown)



Note: W1 through W4 indicate wait cycles.

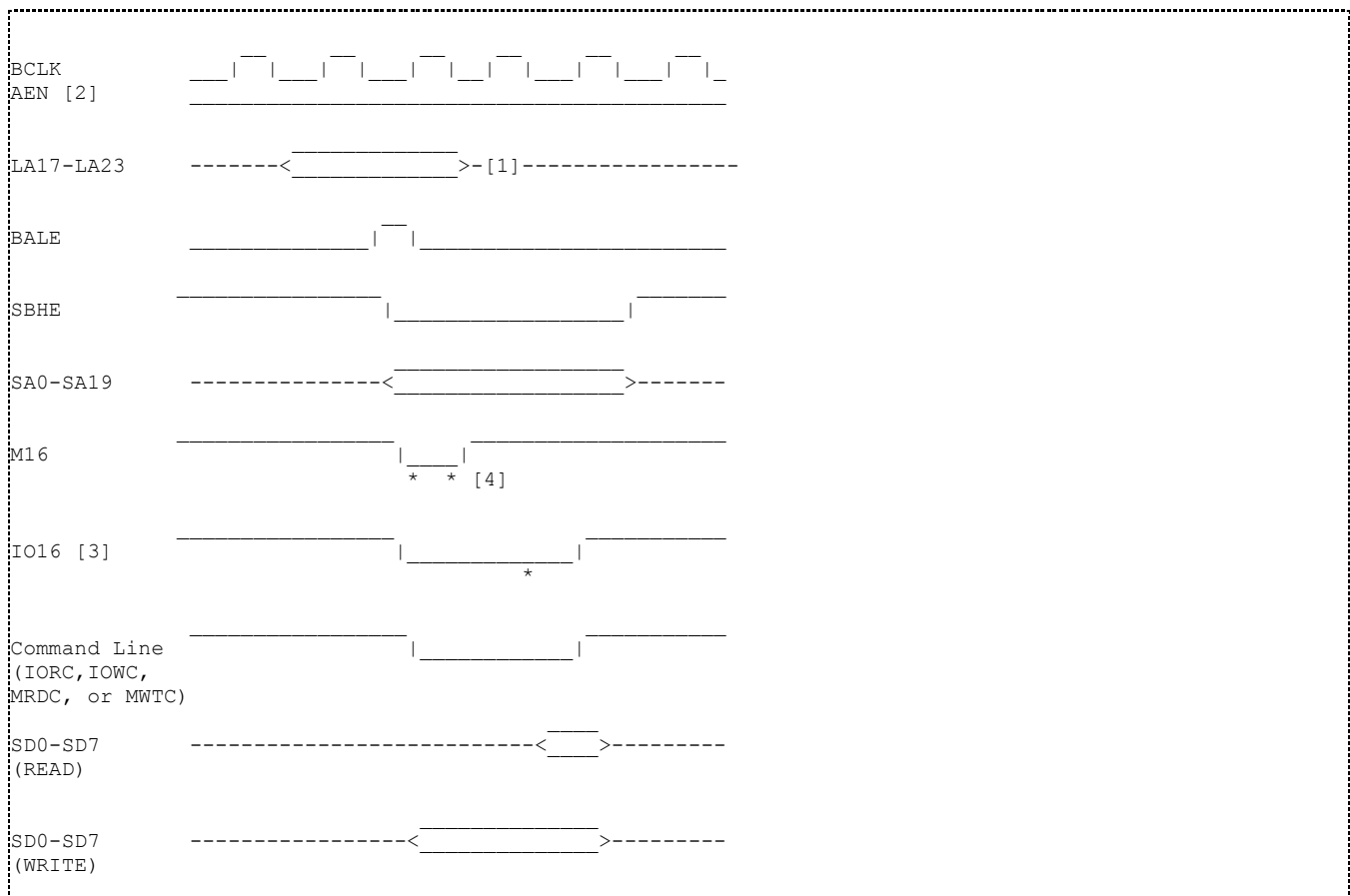
BALE is placed high, and the address is latched on the SA bus. The slave device may safely sample the address during the falling edge of BALE, and the address on the SA bus remains valid until the end of the transfer cycle. Note that AEN remains low throughout the entire transfer cycle.

The command line is then pulled low (IORC or IOWC for I/O commands, SMRDSC or SMWTC for memory commands, read and write respectively). For write operations, the data remains on the SD bus for the remainder of the transfer cycle. For read operations, the data must be valid on the falling edge of the last cycle.

NOWS is sampled at the midpoint of each wait cycle. If it is low, the transfer cycle terminates without further wait states. CHRDY is sampled during the first half of the clock cycle. If it is low, further wait cycles will be inserted.

The default for 8 bit transfers is 4 wait states. Some computers allow the number of default wait states to be changed.

16 Bit Memory or I/O Transfer Timing Diagram (1 wait state shown)



An asterisk (*) denotes the point where the signal is sampled.

[1] The portion of the address on the LA bus for the NEXT cycle may now be placed on the bus. This is used so that cards may begin decoding the address early. Address pipelining must be active.

[2] AEN remains low throughout the entire transfer cycle, indicating that a normal (non-DMA) transfer is occurring.

[3] Some bus controllers sample this signal during the same clock cycle as M16, instead of during the first wait state, as shown above. In this case, IO16 needs to be pulled low as soon as the address is decoded, which is before the I/O command lines are active.

[4] M16 is sampled a second time, in case the adapter card did not active the signal in time for the first sample (usually because the memory device is not monitoring the LA bus for early address information, or is waiting for the falling edge of BALE).

16 bit transfers follow the same basic timing as 8 bit transfers. A valid address may appear on the LA bus prior to the beginning of the transfer cycle. Unlike the SA bus, the LA bus is not latched, and is not valid for the entire transfer cycle (on most computers). The LA bus should be latched on the falling edge of BALE. Note that on some systems, the LA bus signals will follow the same timing as the SA bus. On either type of system, a valid address is present on the falling edge of BALE.

I/O adapter cards do not need to monitor the LA bus or BALE, since I/O addresses are always within the address space of the SA bus.

SBHE will be pulled low by the system board, and the adapter card must respond with IO16 or M16 at the appropriate time, or else the transfer will be split into two separate 8 bit transfers. Many systems expect IO16 or M16 before the command lines are valid. This requires that IO16 or M16 be pulled low as soon as the address is decoded (before it is known whether the cycle is I/O or Memory). If the system is starting a memory cycle, it will ignore IO16 (and vice-versa for I/O cycles and M16).

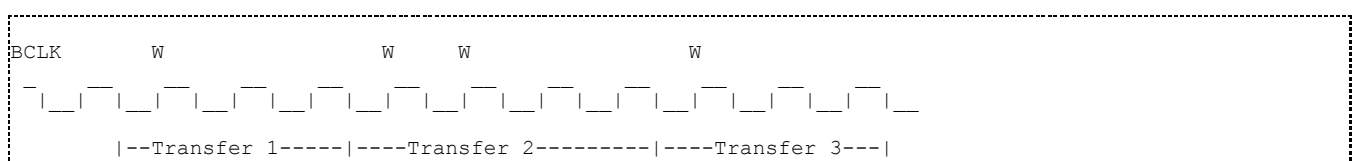
For read operations, the data is sampled on the rising edge of the last clock cycle. For write operations, valid data appears on the bus before the end of the cycle, as shown in the timing diagram. While the timing diagram indicates that the data needs to be sampled on the rising clock, on most systems it remains valid for the entire clock cycle.

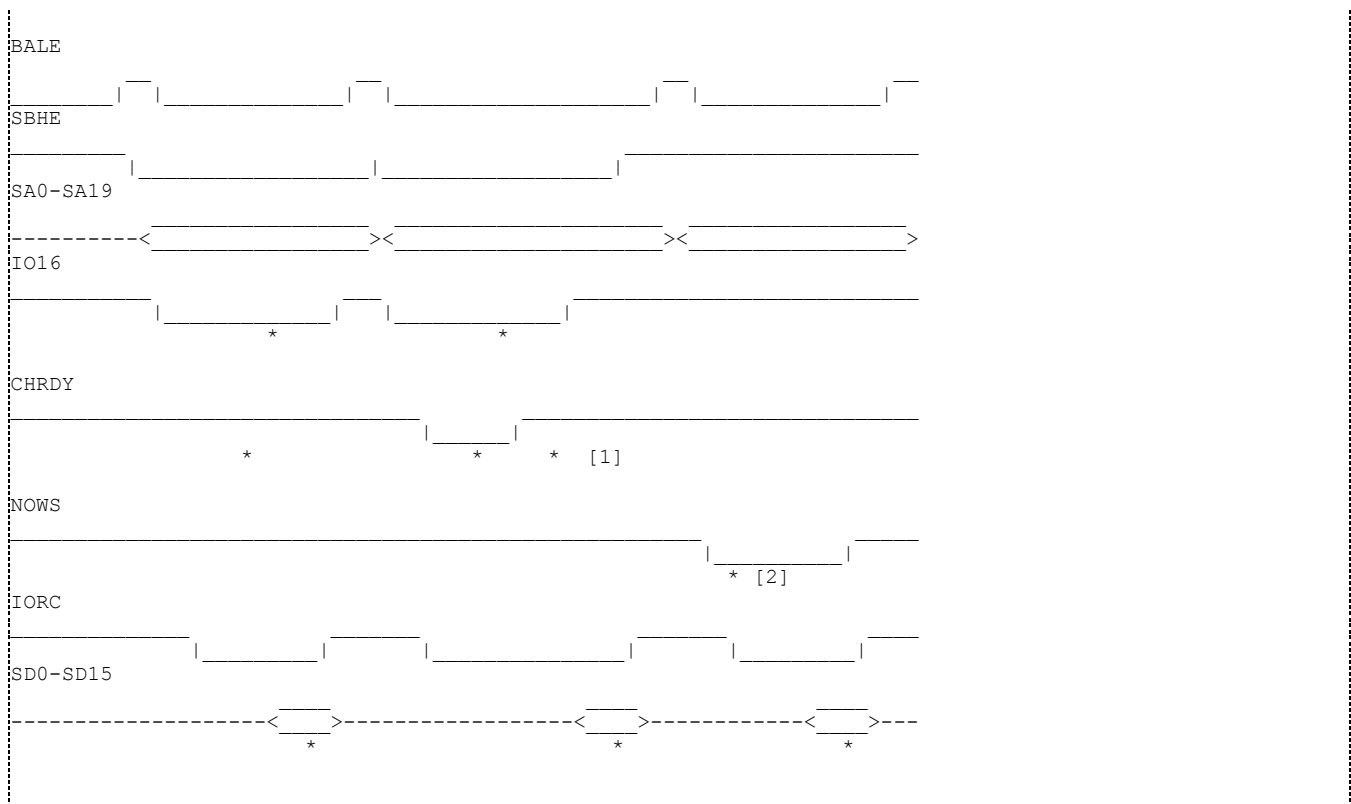
The default for 16 bit transfers is 1 wait state. This may be shortened or lengthened in the same manner as 8 bit transfers, via NOWS and CHRDY. Many systems only allow 16 bit memory devices (and not I/O devices) to transfer using 0 wait states (NOWS has no effect on 16 bit I/O cycles).

SMRDC/SMWTC follow the same timing as MRDC/MWTC respectively when the address is within the lower 1 MB. If the address is not within the lower 1 MB boundary, SMRDC/SMWTC will remain high during the entire cycle.

It is also possible for an 8 bit bus cycle to use the upper portion of the bus. In this case, the timing will be similar to a 16 bit cycle, but an odd address will be present on the bus. This means that the bus is transferring 8 bits using the upper data bits (SD8-SD15).

Shortening or Lengthening the bus cycle





An asterisk (*) denotes the point where the signal is sampled.

W=Wait Cycle

This timing diagram shows three different transfer cycles. The first is a 16 bit standard I/O read. This is followed by an almost identical 16 bit I/O read, with one wait state inserted. The I/O device pulls CHRDY low to indicate that it is not ready to complete the transfer (see [1]). This inserts a wait cycle, and CHRDY is again sampled. At this second sample, the I/O device has completed its operation and released CHRDY, and the bus cycle now terminates. The third cycle is an 8 bit transfer, which is shortened to 1 wait state (the default is 4) by the use of NWS.

I/O Port Addresses

Note: Only the first 10 address lines are decoded for I/O operations. This limits the I/O address space to address 3FF (hex) and lower. Some systems allow for 16 bit I/O address space, but may be limited due to some I/O cards only decoding 10 of these 16 bits.

Port (hex)	Port Assignments
000-00F	DMA Controller
010-01F	DMA Controller (PS/2)
020-02F	Master Programmable Interrupt Controller (PIC)
030-03F	Slave PIC
040-05F	Programmable Interval Timer (PIT)
060-06F	Keyboard Controller

070-071	Real Time Clock
080-083	DMA Page Register
090-097	Programmable Option Select (PS/2)
0A0-0AF	PIC #2
0C0-0CF	DMAC #2
0E0-0EF	reserved
0F0-0FF	Math coprocessor, PCJr Disk Controller
100-10F	Programmable Option Select (PS/2)
110-16F	AVAILABLE
170-17F	Hard Drive 1 (AT)
180-1EF	AVAILABLE
1F0-1FF	Hard Drive 0 (AT)
200-20F	Game Adapter
210-217	Expansion Card Ports
220-26F	AVAILABLE
278-27F	Parallel Port 3
280-2A1	AVAILABLE
2A2-2A3	clock
2B0-2DF	EGA/Video
2E2-2E3	Data Acquisition Adapter (AT)
2E8-2EF	Serial Port COM4
2F0-2F7	Reserved
2F8-2FF	Serial Port COM2
300-31F	Prototype Adapter, Periscope Hardware Debugger
320-32F	AVAILABLE
330-33F	Reserved for XT/370
340-35F	AVAILABLE
360-36F	Network
370-377	Floppy Disk Controller
378-37F	Parallel Port 2
380-38F	SDLC Adapter
390-39F	Cluster Adapter
3A0-3AF	reserved
3B0-3BF	Monochrome Adapter

3BC-3BF	Parallel Port 1
3C0-3CF	EGA/VGA
3D0-3DF	Color Graphics Adapter
3E0-3EF	Serial Port COM3
3F0-3F7	Floppy Disk Controller
3F8-3FF	Serial Port COM1

Soundblaster cards usually use I/O ports 220-22F.

Data acquisition cards frequently use 300-31F.

DMA Read and Write

The ISA bus uses two DMA controllers (DMAC) cascaded together. The slave DMAC connects to the master DMAC via DMA channel 4 (channel 0 on the master DMAC). The slave therefore gains control of the bus through the master DMAC. On the ISA bus, the DMAC is programmed to use fixed priority (channel 0 always has the highest priority), which means that channel 0-4 from the slave have the highest priority (since they connect to the master channel 0), followed by channels 5-7 (which are channel 1-3 on the master).

The DMAC can be programmed for read transfers (data is read from memory and written to the I/O device), write transfers (data is read from the I/O device and written to memory), or verify transfers (neither a read or a write - this was used by DMA CH0 for DRAM refresh on early PCs).

Before a DMA transfer can take place, the DMA Controller (DMAC) must be programmed. This is done by writing the start address and the number of bytes to transfer (called the transfer count) and the direction of the transfer to the DMAC. After the DMAC has been programmed, the device may activate the appropriate DMA request (DRQx) line.

Slave DMA Controller

I/O	Port
0000	DMA CH0 Memory Address Register Contains the lower 16 bits of the memory address, written as two consecutive bytes.
0001	DMA CH0 Transfer Count Contains the lower 16 bits of the transfer count, written as two consecutive bytes.
0002	DMA CH1 Memory Address Register
0003	DMA CH1 Transfer Count
0004	DMA CH2 Memory Address Register
0005	DMA CH2 Transfer Count
0006	DMA CH3 Memory Address Register

0007	DMA CH3 Transfer Count
0008	<p>DMAC Status/Control Register</p> <p>Status (I/O read) bits 0-3: Terminal Count, CH 0-3</p> <ul style="list-style-type: none"> bits 4-7: Request CH0-3 <p>Control (write)</p> <ul style="list-style-type: none"> bit 0: Mem to mem enable (1 = enabled) bit 1: ch0 address hold enable (1 = enabled) bit 2: controller disable (1 = disabled) bit 3: timing (0 = normal, 1 = compressed) bit 4: priority (0 = fixed, 1 = rotating) bit 5: write selection (0 = late, 1 = extended) bit 6: DRQx sense asserted (0 = high, 1 = low) bit 7: DAKn sense asserted (0 = low, 1 = high)
0009	<p>Software DRQn Request</p> <ul style="list-style-type: none"> bits 0-1: channel select (CH0-3) bit 2: request bit (0 = reset, 1 = set)
000A	<p>DMA mask register</p> <ul style="list-style-type: none"> bits 0-1: channel select (CH0-3) bit 2: mask bit (0 = reset, 1 = set)
000B	<p>DMA Mode Register</p> <ul style="list-style-type: none"> bits 0-1: channel select (CH0-3) bits 2-3: 00 = verify transfer, 01 = write transfer, 10 = read transfer, 11 = reserved bit 4: Auto init (0 = disabled, 1 = enabled) bit 5: Address (0 = increment, 1 = decrement) bits 6-7: 00 = demand transfer mode, 01 = single transfer mode, 10 = block transfer mode, 11 = cascade mode
000C	<p>DMA Clear Byte Pointer</p> <p>Writing to this causes the DMAC to clear the pointer used to keep track of 16 bit data transfers into and out of the DMAC for hi/low byte sequencing.</p>
000D	DMA Master Clear (Hardware Reset)
000E	DMA Reset Mask Register - clears the mask register
000F	<p>DMA Mask Register</p> <ul style="list-style-type: none"> bits 0-3: mask bits for CH0-3 (0 = not masked, 1 = masked)

0081	DMA CH2 Page Register (address bits A16-A23)
0082	DMA CH3 Page Register
0083	DMA CH1 Page Register
0087	DMA CH0 Page Register
0089	DMA CH6 Page Register
008A	DMA CH7 Page Register
008B	DMA CH5 Page Register

Master DMA Controller

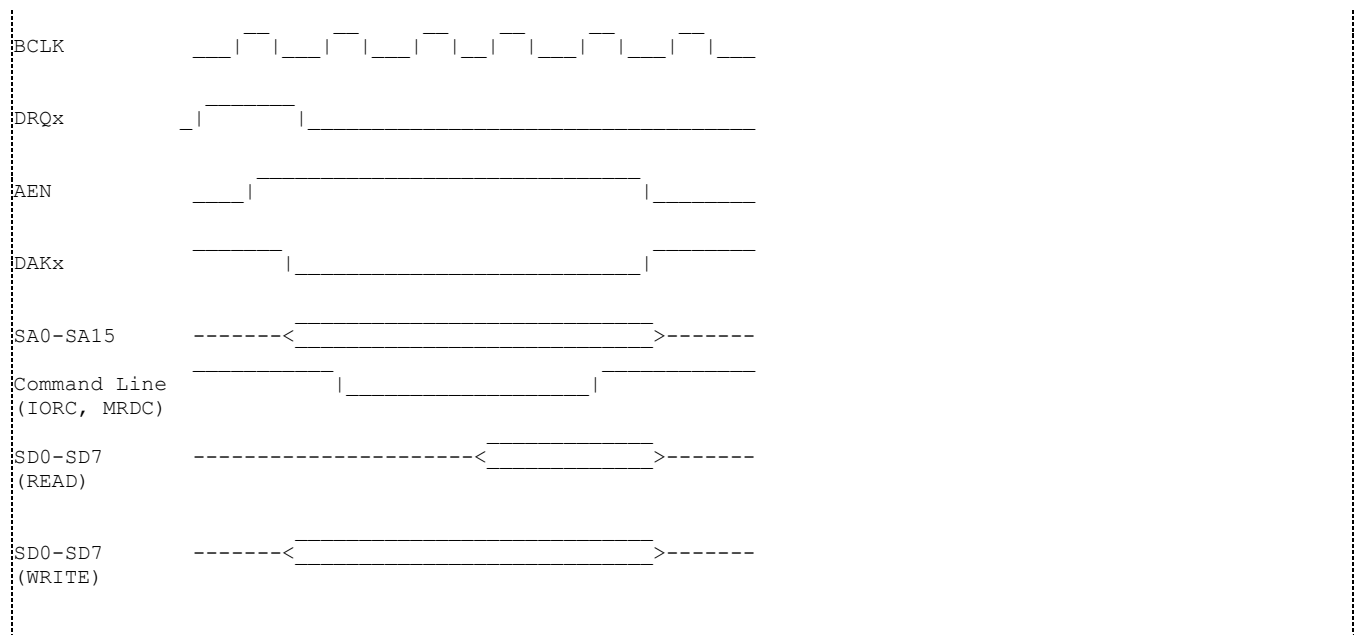
I/O	Port
00C0	DMA CH4 Memory Address Register Contains the lower 16 bits of the memory address, written as two consecutive bytes.
00C2	DMA CH4 Transfer Count Contains the lower 16 bits of the transfer count, written as two consecutive bytes.
00C4	DMA CH5 Memory Address Register
00C6	DMA CH5 Transfer Count
00C8	DMA CH6 Memory Address Register
00CA	DMA CH6 Transfer Count
00CC	DMA CH7 Memory Address Register
00CE	DMA CH7 Transfer Count
00D0	<p>DMAC Status/Control Register</p> <p>Status (I/O read):</p> <ul style="list-style-type: none"> ▪ bits 0-3: Terminal Count, CH 4-7 ▪ bits 4-7: Request CH4-7 <p>Control (write):</p> <ul style="list-style-type: none"> ▪ bit 0: Mem to mem enable (1 = enabled) ▪ bit 1: ch0 address hold enable (1 = enabled) ▪ bit 2: controller disable (1 = disabled) ▪ bit 3: timing (0 = normal, 1 = compressed) ▪ bit 4: priority (0 = fixed, 1 = rotating) ▪ bit 5: write selection (0 = late, 1 = extended) ▪ bit 6: DRQx sense asserted (0 = high, 1 = low) ▪ bit 7: DAKn sense asserted (0 = low, 1 = high)

00D2	Software DRQn Request <ul style="list-style-type: none"> bits 0-1: channel select (CH4-7) bit 2: request bit (0 = reset, 1 = set)
00D4	DMA mask register <ul style="list-style-type: none"> bits 0-1: channel select (CH4-7) bit 2: mask bit (0 = reset, 1 = set)
00D6	DMA Mode Register <ul style="list-style-type: none"> bits 0-1: channel select (CH4-7) bits 2-3: 00 = verify transfer, 01 = write transfer, 10 = read transfer, 11 = reserved bit 4: Auto init (0 = disabled, 1 = enabled) bit 5: Address (0 = increment, 1 = decrement) bits 6-7: 00 = demand transfer mode, 01 = single transfer mode, 10 = block transfer mode, 11 = cascade mode
00D8	DMA Clear Byte Pointer Writing to this causes the DMAC to clear the pointer used to keep track of 16 bit data transfers into and out of the DMAC for hi/low byte sequencing.
00DA	DMA Master Clear (Hardware Reset)
00DC	DMA Reset Mask Register - clears the mask register
00DE	DMA Mask Register <ul style="list-style-type: none"> bits 0-3: mask bits for CH4-7 (0 = not masked, 1 = masked)

Single Transfer Mode

The DMAC is programmed for transfer. The DMA device requests a transfer by driving the appropriate DRQ line high. The DMAC responds by asserting AEN and acknowledges the DMA request through the appropriate DAK line. The I/O and memory command lines are also asserted. When the DMA device sees the DAK signal, it drops the DRQ line.

The DMAC places the memory address on the SA bus (at the same time as the command lines are asserted), and the device either reads from or writes to memory, depending on the type of transfer. The transfer count is incremented, and the address incremented/decremented. DAK is de-asserted. The cpu now once again has control of the bus, and continues execution until the I/O device is once again ready for transfer. The DMA device repeats the procedure, driving DRQ high and waiting for DAK, then transferring data. This continues for a number of cycles equal to the transfer count. When this has been completed, the DMAC signals the cpu that the DMA transfer is complete via the TC (terminal count) signal.



Block Transfer Mode

The DMAC is programmed for transfer. The device attempting DMA transfer drives the appropriate DRQ line high. The motherboard responds by driving AEN high and DAK low. This indicates that the DMA device is now the bus master. In response to the DAK signal, the DMA device drops DRQ. The DMAC places the address for DMA transfer on the address bus. Both the memory and I/O command lines are asserted (since DMA involves both an I/O and a memory device). AEN prevents I/O devices from responding to the I/O command lines, which would not result in proper operation since the I/O lines are active, but a memory address is on the address bus. The data transfer is now done (memory read or write), and the DMAC increments/decrements the address and begins another cycle. This continues for a number of cycles equal to the DMAC transfer count. When this has been completed, the terminal count signal (TC) is generated by the DMAC to inform the cpu that the DMA transfer has been completed.

Note: Block transfer must be used carefully. The bus cannot be used for other things (like RAM refresh) while block mode transfers are being done.

Demand Transfer Mode

The DMAC is programmed for transfer. The device attempting DMA transfer drives the appropriate DRQ line high. The motherboard responds by driving AEN high and DAK low. This indicates that the DMA device is now the bus master. Unlike single transfer and block transfer, the DMA device does not drop DRQ in response to DAK. The DMA device transfers data in the same manner as for block transfers. The DMAC will continue to generate DMA cycles as long as the I/O device asserts DRQ. When the I/O device is unable to continue the transfer (if it no longer had data ready to transfer, for example), it drops DRQ and the cpu once again has control of the bus. Control is returned to the DMAC by once again asserting DRQ. This continues until the terminal count has been reached, and the TC signal informs the cpu that the transfer has been completed.

Interrupts on the ISA bus

Name	Interrupt	Description
NMI	2	Parity Error, Mem Refresh
IRQ0	8	8253 Channel 0 (System Timer)
IRQ1	9	Keyboard
IRQ2	A	Cascade from slave PIC
IRQ3	B	COM2
IRQ4	C	COM1
IRQ5	D	LPT2
IRQ6	E	Floppy Drive Controller
IRQ7	F	LPT1
IRQ8	F	Real Time Clock
IRQ9	F	Redirection to IRQ2
IRQ10	F	Reserved
IRQ11	F	Reserved
IRQ12	F	Mouse Interface
IRQ13	F	Coprocessor
IRQ14	F	Hard Drive Controller
IRQ15	F	Reserved

IRQ0,1,2,8, and 13 are not available on the ISA bus.

The IBM PC and XT had only a single 8259 interrupt controller. The AT and later machines have a second interrupt controller, and the two are used in a master/slave combination. IRQ2 and IRQ9 are the same pin on most ISA systems. Interrupts on most systems may be either edge triggered or level triggered. The default is usually edge triggered, and active high (low to high transition). The interrupt level must be held high until the first interrupt acknowledge cycle (two interrupt acknowledge bus cycles are generated in response to an interrupt request).

The software aspects of interrupts and interrupt handlers is intentionally omitted from this document, due to the numerous syntactical differences in software tools and the fact that adequate documentation of this topic is usually provided with development software.

Bus Mastering

An ISA device may take control of the bus, but this must be done with caution. There are no safety mechanisms involved, and so it is easily possible to crash the entire system by incorrectly taking control of the bus. For example, most systems require bus cycles for DRAM refresh. If the ISA bus master does not relinquish control of the bus or generate its own DRAM refresh cycles every 15 microseconds, the

system RAM can become corrupted. The ISA adapter card can generate refresh cycles without relinquishing control of the bus by asserting REFRESH. MRDC can be then monitored to determine when the refresh cycle ends.

To take control of the bus, the device first asserts its DRQ line. The DMAC sends a hold request to the cpu, and when the DMAC receives a hold acknowledge, it asserts the appropriate DAK line corresponding to the DRQ line asserted. The device is now the bus master. AEN is asserted, so if the device wishes to access I/O devices, it must assert MASTER16 to release AEN. Control of the bus is returned to the system board by releasing DRQ.

Contributions

- Joakim Ögren
- Niklas Edmundsson
- Mark Sokos (mailto:sokos@desupernet.net)
- Pieter Hollants (mailto:fxmts205@rz.uni-frankfurt.de)

Sources

- Mark Sokos ISA page (<http://users.supernet.com/sokos/isa.txt>)
- "ISA System Architecture, 3rd Edition" by Tom Shanley and Don Anderson ISBN 0-201-40996-8
- "Eisa System Architecture, 2nd Edition" by Tom Shanley and Don Anderson ISBN 0-201-40995-X
- "Microcomputer Busses" by R.M. Cram ISBN 0-12-196155-9
- HelpPC v2.10 Quick Reference Utility, by David Jurgens
- ZIDA 80486 Mother Board User's Manual, OPTi 486, 82C495sx

UNIQ6f7b3fe46a9b364a-keydata-00000000-QINU

Retrieved from "<http://www.hardwarebook.info/index.php?title=ISA&oldid=8429>"

Categories: Connector | Bus

- |
- Page |
- Discussion |
- View source |
- History |
- |
- What links here |
- Related changes |
- Special pages |

-
- Help |
 - About HwB |

-

Disclaimers