Dec Hex Bin
26 1A 00011010

ORG ; TWENTY-SIX

ISA, PC104 AND PCI BUSES

The x86 PC
assembly language, design, and interfacing
fifth edition
Prentice Hall

The x86 PC
assembly language, design, and interfacing
fifth edition

MUHAMMAD ALI MAZIDI
JANICE GILLISPIE MAZIDI
DANNY CAUSEY

OBJECTIVES

this chapter enables the student to:

- Describe the ISA bus signals for I/O interfacing.
- Describe the ISA bus signals for memory interfacing.
- Calculate I/O cycle time and bus bandwidth for the ISA bus.
- Define the meaning of the terms master, slave, bus arbitration, bus protocol, and bus bandwidth and describe their importance in PC design.
- Describe the evolution of bus architecture from ISA to PCI.
- List the limitations of the ISA bus.

OBJECTIVES (*cont*)

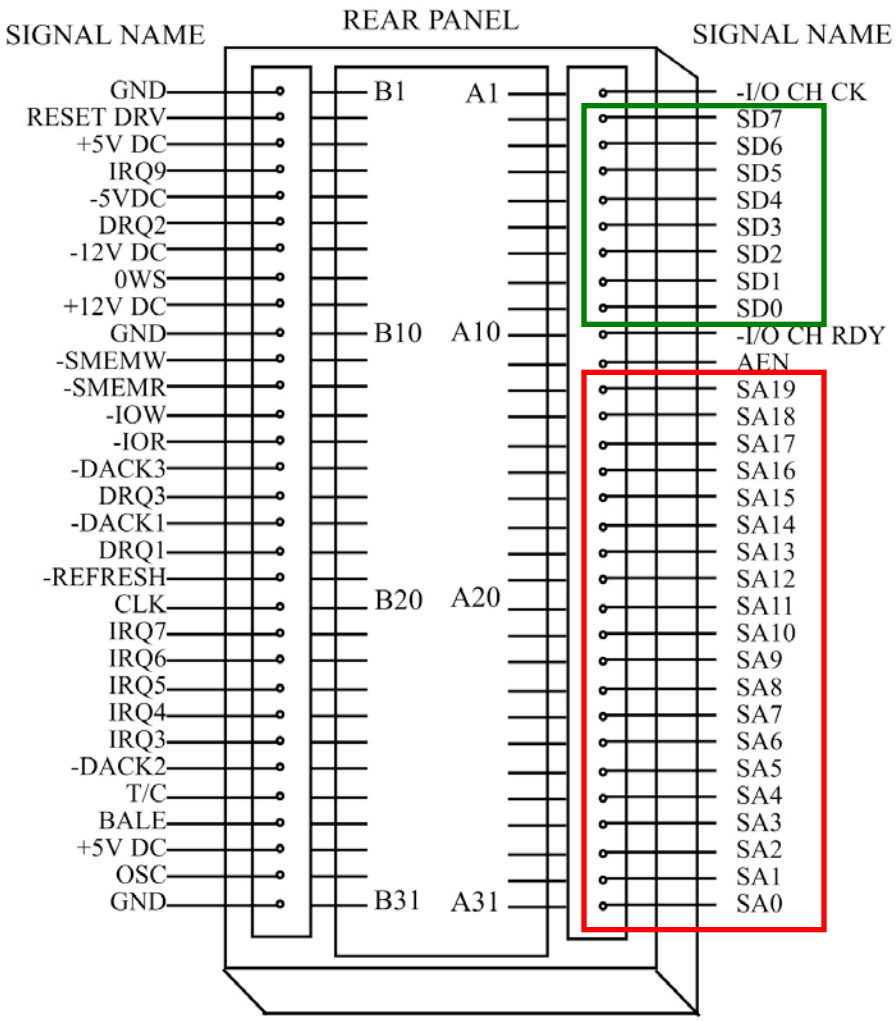
this chapter enables the student to:

- List the limitations of the ISA bus.
- List the major characteristics of PCI architecture.
- List the enhancements of the PCI bus over the ISA bus.
- Contrast and compare ISA and PCI in terms of bus bandwidth.
- Define the term local bus and describe its merits.
- List the major characteristics of the PCI local bus.

26.1: ISA BUS MEMORY SIGNALS

- In PCs with x86 microprocessors, signals for the ISA expansion slots are provided by the chipset.
  - The chipset makes sure signals for the ISA slot conform with the ISA bus standard regardless of CPU speed and data width.
- ISA bus specifications and timing for memory are precise and must be understood to design an ISA plug-in card with on-board memory.

26.1: ISA BUS MEMORY SIGNALS
address bus signals



SA0–SA19

(system address)

Provides address signals for the desired memory or I/O location.

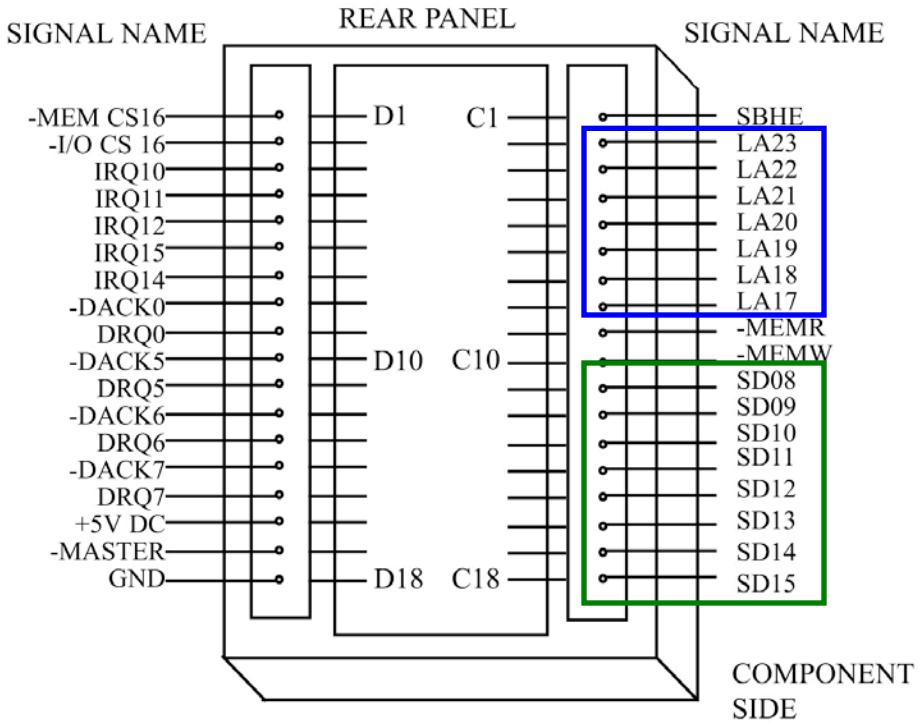
The chipset latches these signals & holds them valid throughout the bus cycle.

SD0–SD15

(system data bus)

Transfers data between CPU, memory & I/O devices.

26.1: ISA BUS MEMORY SIGNALS
address bus signals



LA17–LA23
(latchable address)

With **SA0–SA19**, gives access to 16M of memory space from the ISA slot.

These signals are latched by the board designed for the expansion slot.

SD0–SD15
(system data bus)

Transfers data between CPU, memory & I/O devices.

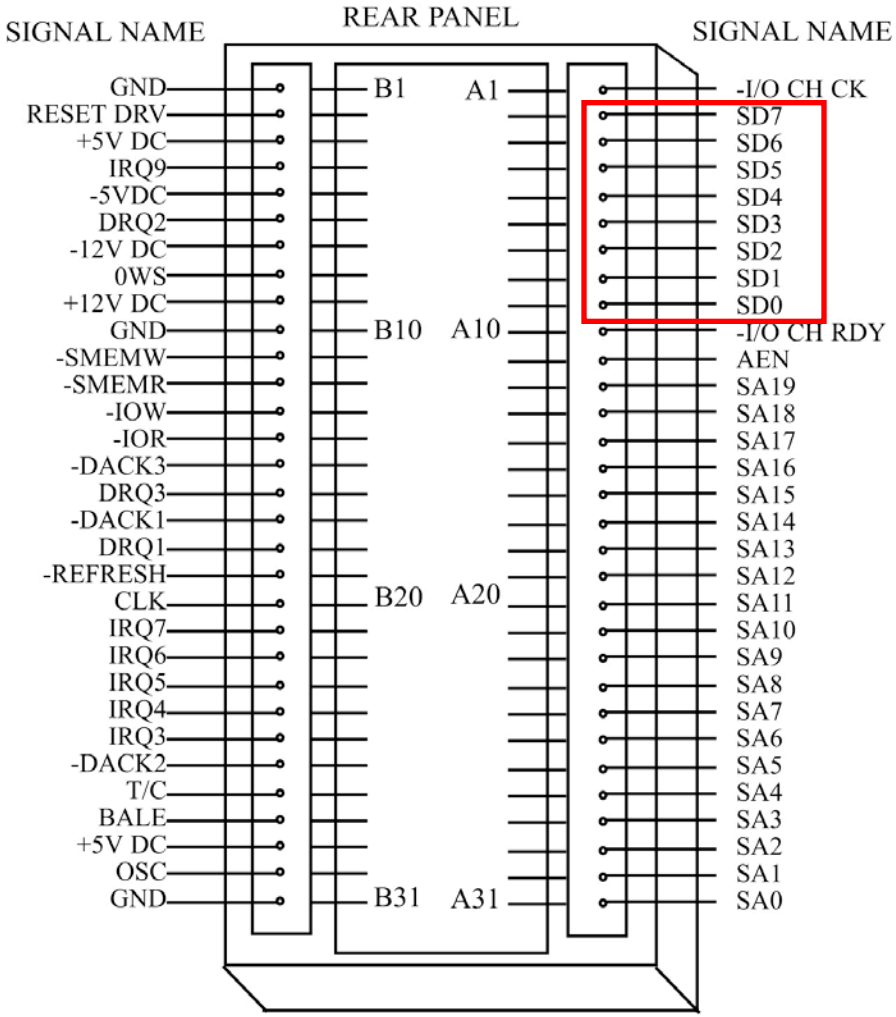
26.1: ISA BUS MEMORY SIGNALS
memory control signals

- **SBHE** (system byte high enable) - an *active-low* signal which indicates data is being transferred on the upper byte (D8–D15) of the data bus.
- **MEMW** and **MEMR** are used for 16M memory:
  - **MEMW** (memory write) - *active-low* control signal used to write data into the memory chip.
    - Connected to **WE** (write enable) pin of the memory chip.
  - **MEMR** (memory read) - *active-low* control signal used to read data from the memory chip.
    - Connected to **OE** (output enable) pin of the memory chip.

26.1: ISA BUS MEMORY SIGNALS
memory control signals

- **SMEMW** and **SMEMR** are used if the 1M memory 00000–FFFFFH is chosen:
  - **SMEMW** (system memory write) - an *active-low* control signal used to write data into a memory chip.
    - Connected to the WE pin of the memory chip.
    - Goes *low* when accessing addresses between 0 and FFFFFH (0 and 1M bytes).
  - **SMEMR** (system memory read) - an *active-low* control signal used to read data from the memory chip.
    - Connected to the **OE** pin of the memory chip.
    - Goes *low* when accessing addresses between 0 and FFFFFH (0 and 1M bytes).

26.1: ISA BUS MEMORY SIGNALS
memory control signals – MEMCS16

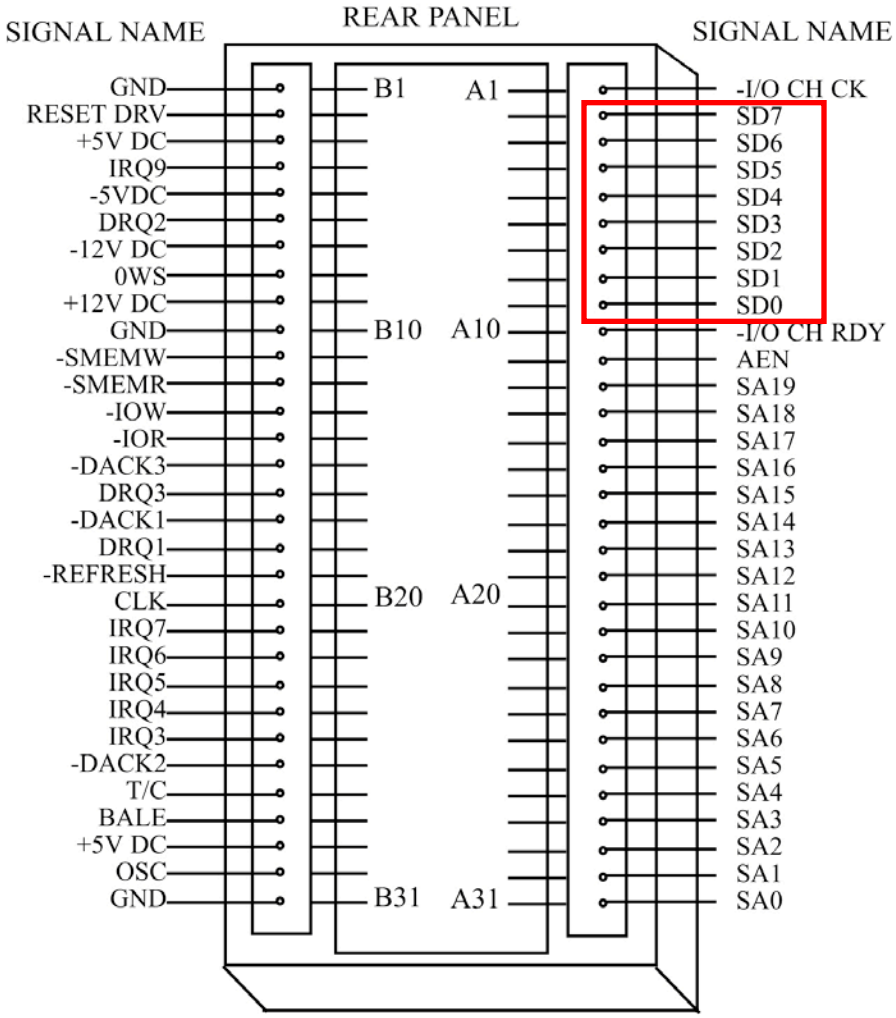


MEMCS16
(memory chip select 16) - an *active-low* input signal.

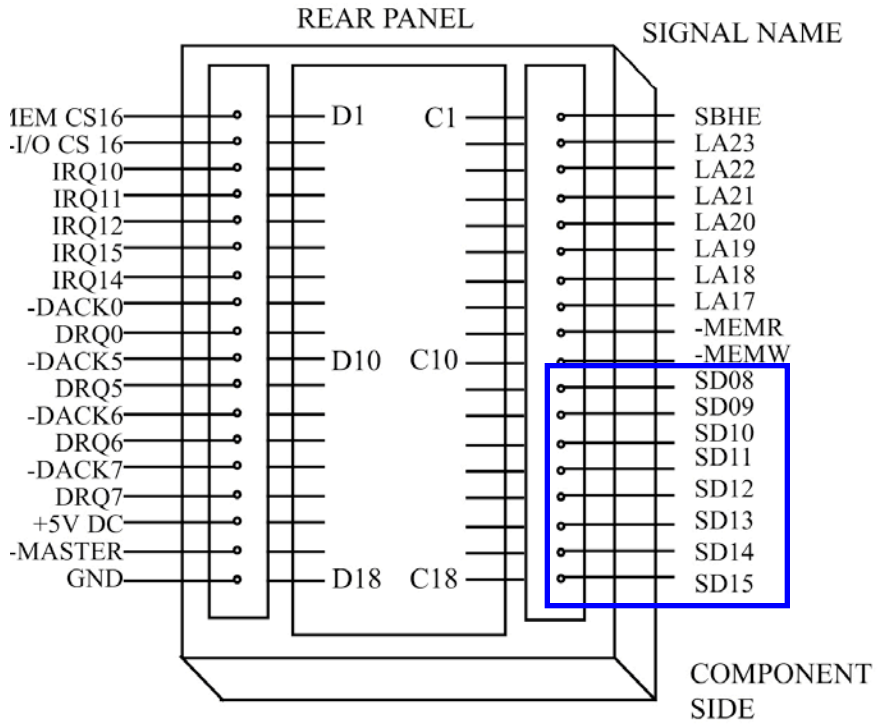
When *not* asserted, it indicates to the chipset that only **D0–D7** of the data bus is being used.

The 8-bit portion is the default mode, and it is achieved by doing nothing to this pin.

26.1: ISA BUS MEMORY SIGNALS
memory control signals - MEMCS16



When **MEMCS16** is asserted *low*, both the **low** & **high** byte of the data bus will be used.



26.1: ISA BUS MEMORY SIGNALS
16-bit memory timing for ISA bus

- The ISA bus allows interfacing of slow memories by inserting wait states into the memory cycle time.
  - Standard 8-bit data transfer has 4 WS in the memory read cycle time, making default memory cycle time 6 clocks.
  - Standard 16-bit data transfer uses 1 WS for a 3 clock read/write cycle time.

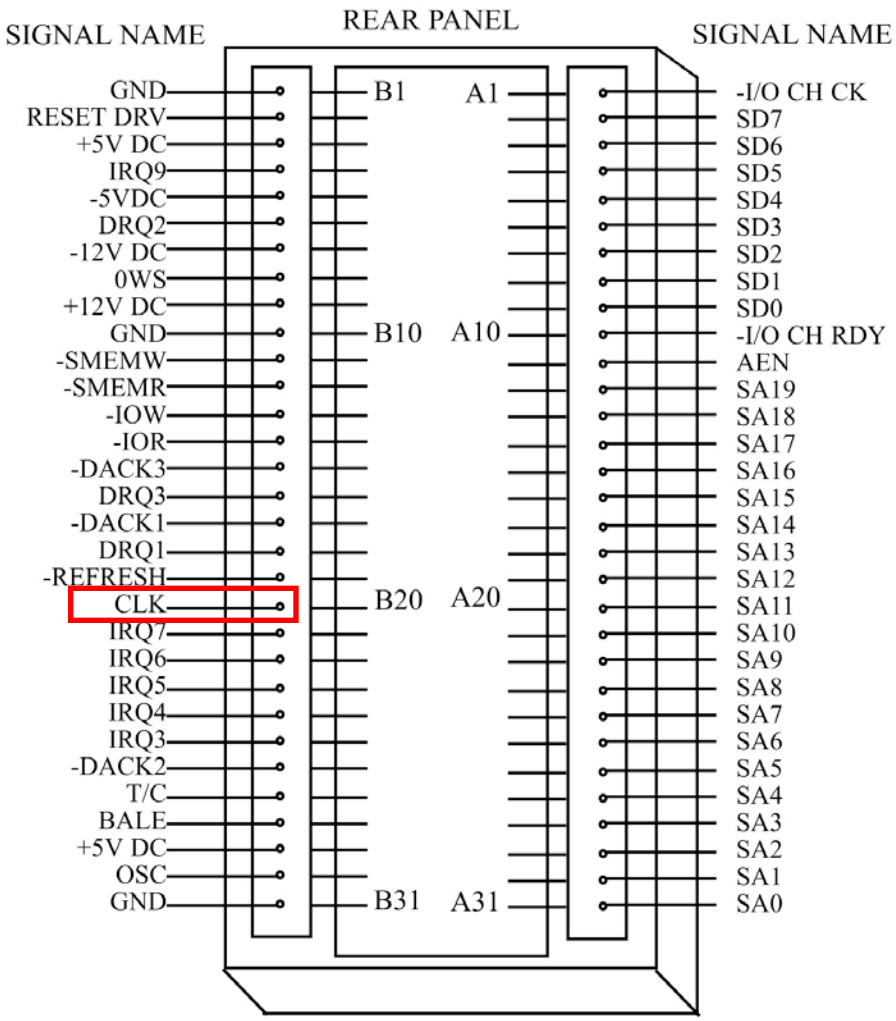
26.1: ISA BUS MEMORY SIGNALS
memory control signals

- **ZEROWS** (zero wait state) - *active-low* input signal.
  - Standard 16-bit ISA bus cycle time contains one WS unless **ZEROWS** is activated.
    - Activating this pin tells the CPU the present memory cycle can be completed with no wait state, a bus cycle time of 2 clocks.
  - Standard 8-bit ISA bus cycle time contains four WS unless **ZEROWS** is activated.
  - When both **MEMCS16** and **ZEROWS** are *high* (without asserting them *low*), the 8-bit data bus (D0–D7) is being used and the data transfer is completed in 6 clocks.
    - Sufficient for interfacing even slow ROMs to the ISA bus.
  - If **MEMCS**16 = 1 and **ZEROWS** = 0, the 8-bit memory cycle time has 1 WS instead of 4.

26.1: ISA BUS MEMORY SIGNALS
memory control signals

- **IOCHRDY** (IO Channel Ready) - an *active-low* input.
  - Driving it *low* asks the system to extend the standard ISA bus cycle time.
  - In response to the asserting of this signal, the system will insert wait states into the memory (or I/O) cycle time until it is deasserted.
    - Rarely used for memory interfacing since the standard memory cycle time of the ISA bus provides plenty of time.
  - The function of this pin is the opposite of **ZEROWS**.

26.1: ISA BUS MEMORY SIGNALS
memory control signals



SYSCLK (system clock) - an output clock providing 8 MHz standard ISA bus clock.

- 8 MHz clock results in a 125 ns (1/8 MHz = 125 ns) period, on which all memory I/O ISA bus is based.
- A zero WS read cycle time for a 16-bit bus takes 250 ns. (2 x 125)
- Standard 8-bit bus with its 4 WS will be 750 ns. (2 + 4) x 125 ns.
- 8-bit bus with **ZEROWS** asserted low has 1 WS, making its memory cycle time 375 ns. (2 + 1) x 125 ns

26.1: ISA BUS MEMORY SIGNALS
ISA bus timing for memory – 8-bit

- In the case of the standard 8-bit read/write cycle time, the chipset inserts 4 WS clocks the read cycle.

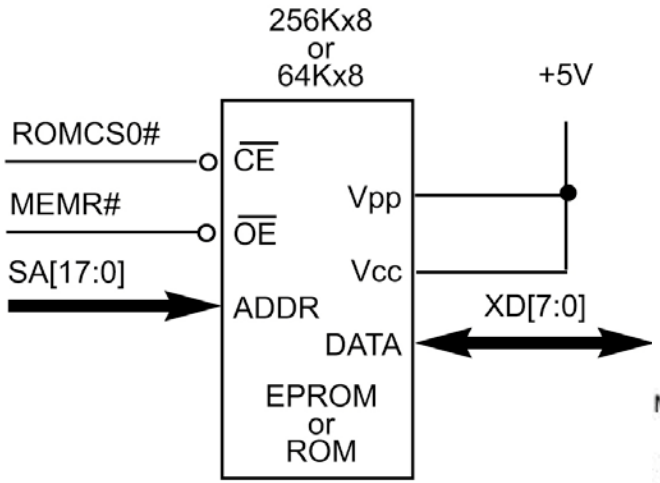


Figure 26-4 8-bit ROM Connection to ISA Bus

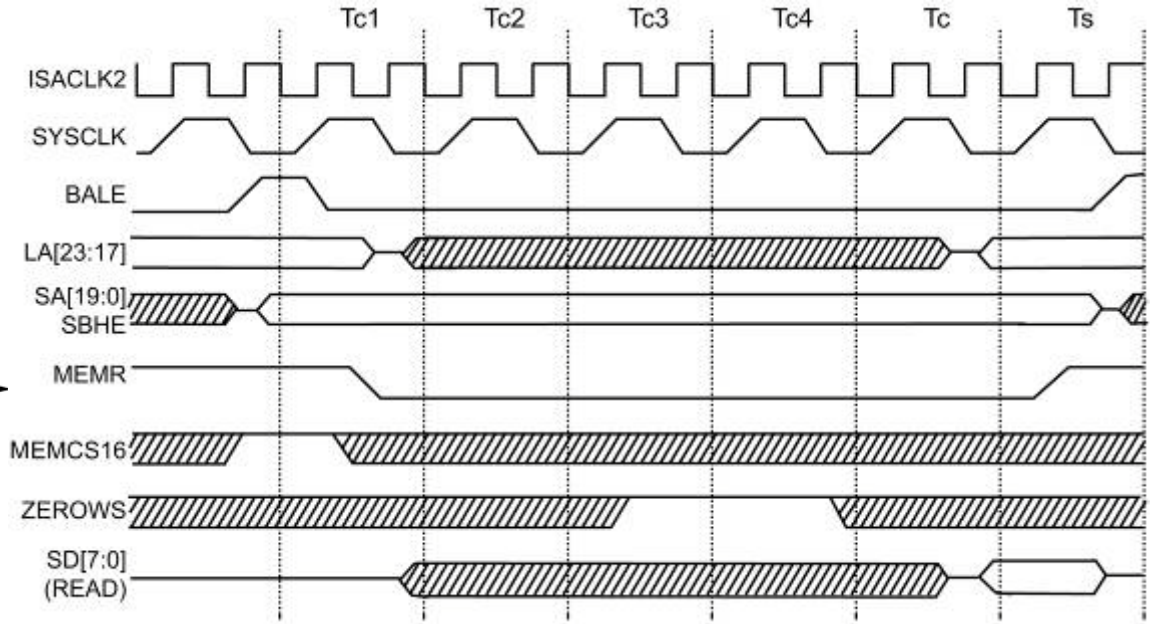


Figure 26-2 Standard 8-bit ISA Memory Read Cycle Time (4 WS)

26.1: ISA BUS MEMORY SIGNALS
ISA bus timing for memory – 8-bit

- Where **ZEROWS** is asserted *low* & **MEMCS16** = 1 the read/write cycle time has 1 WS for the 8-bit bus.

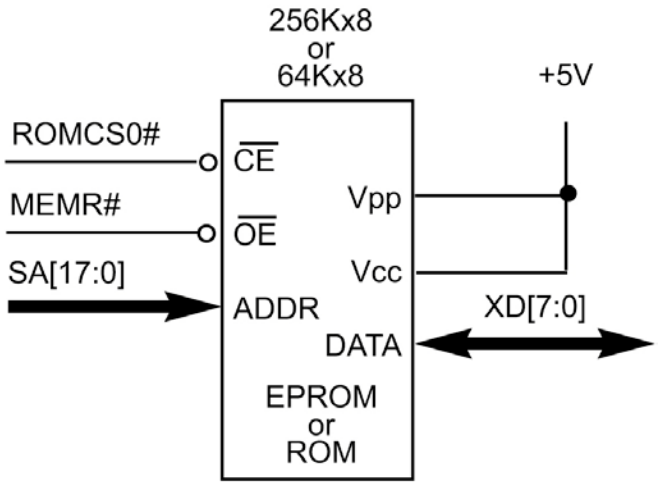


Figure 26-4 8-bit ROM Connection to ISA Bus

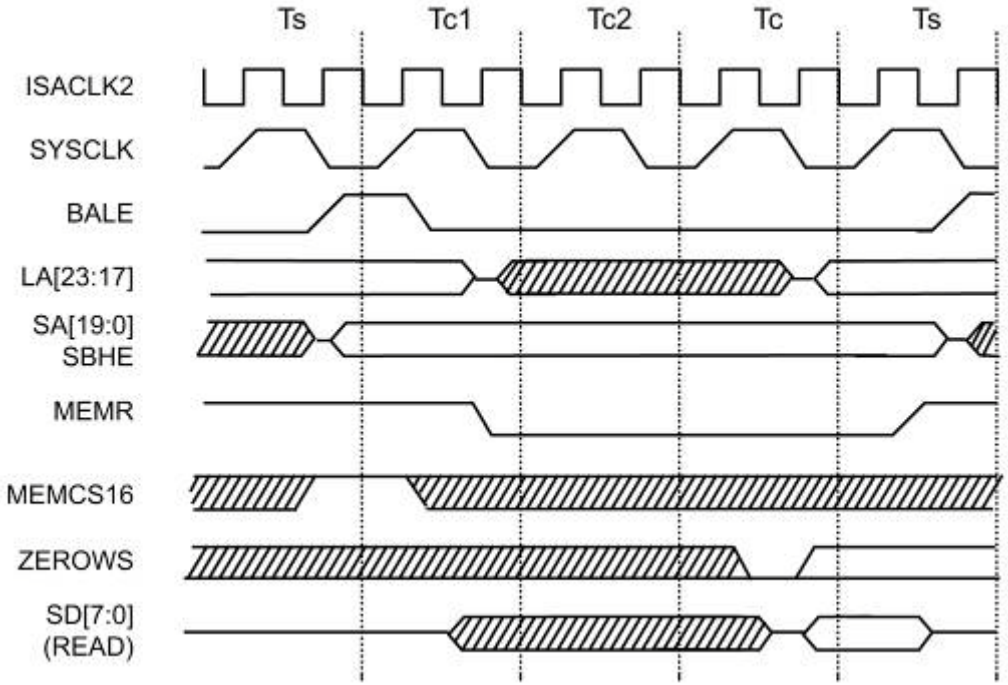


Figure 26-3 Zero WS 8-bit ISA Memory Read Cycle Time (1 WS)

26.1: ISA BUS MEMORY SIGNALS
ISA bus timing for memory – 16-bit

- A 16-bit data bus transfer is twice as fast as an 8-bit data bus transfer, requires twice the board space, and has a higher power consumption.

To perform the 16-bit data transfer using lines D0–D15, assert **MEMCS16** *low*.

The 16-bit data read cycle time for the ISA bus with zero WS is shown here.

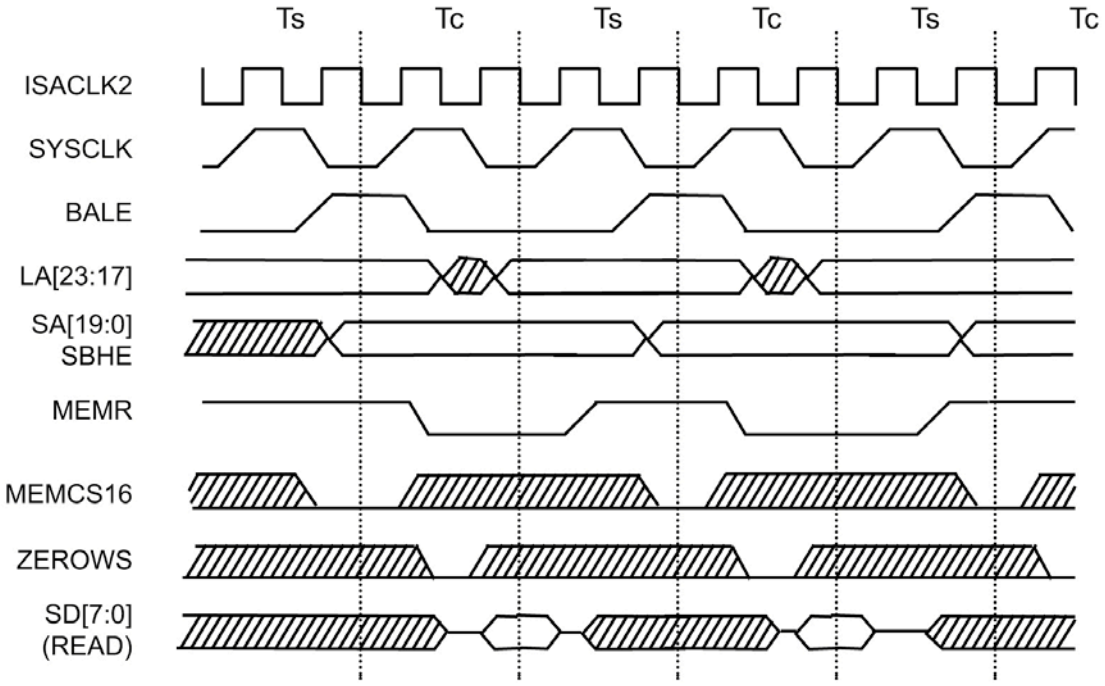


Figure 26-6 Zero WS 16-bit ISA Memory Read Cycle Time (0 WS)

26.1: ISA BUS MEMORY SIGNALS
ISA bus timing for memory - 16-bit

- In all 2-clock bus cycle CPUs and systems, the first clock is set aside for addresses, the second for data.
  - To get zero wait state bus activity, activate **ZEROWS**.
    - Because the standard ISA bus cycle contains one wait state.

26.1: ISA BUS MEMORY SIGNALS
ISA bus timing for memory - 16-bit

- Standard 16-bit ISA bus cycle timing with 1 WS.

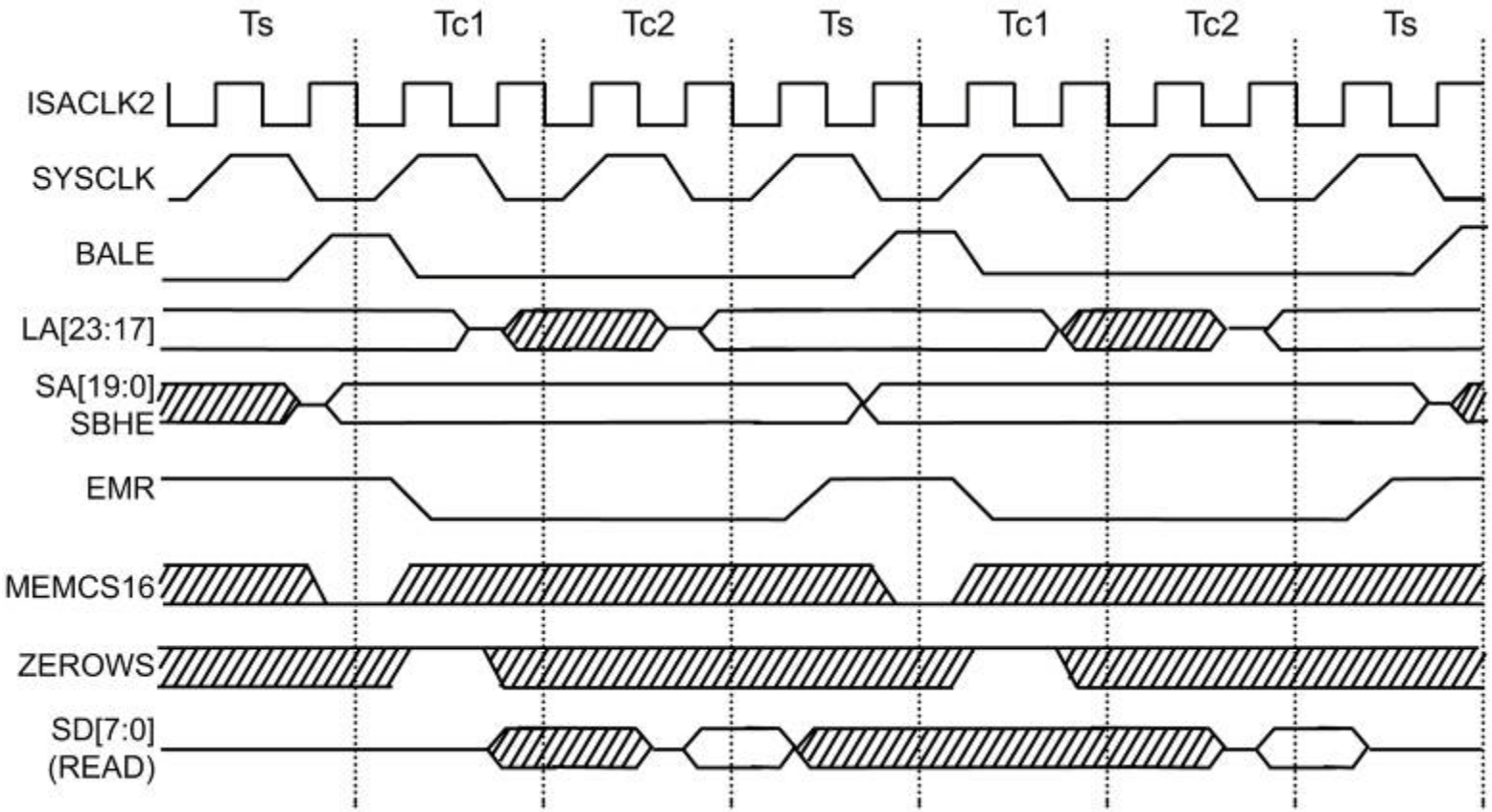


Figure 26-5 Standard 16-bit ISA Memory Read Cycle Time (1 WS)

26.1: ISA BUS MEMORY SIGNALS
ISA bus timing for memory – 16-bit

- The 16-bit ISA cycle with 0 WS.

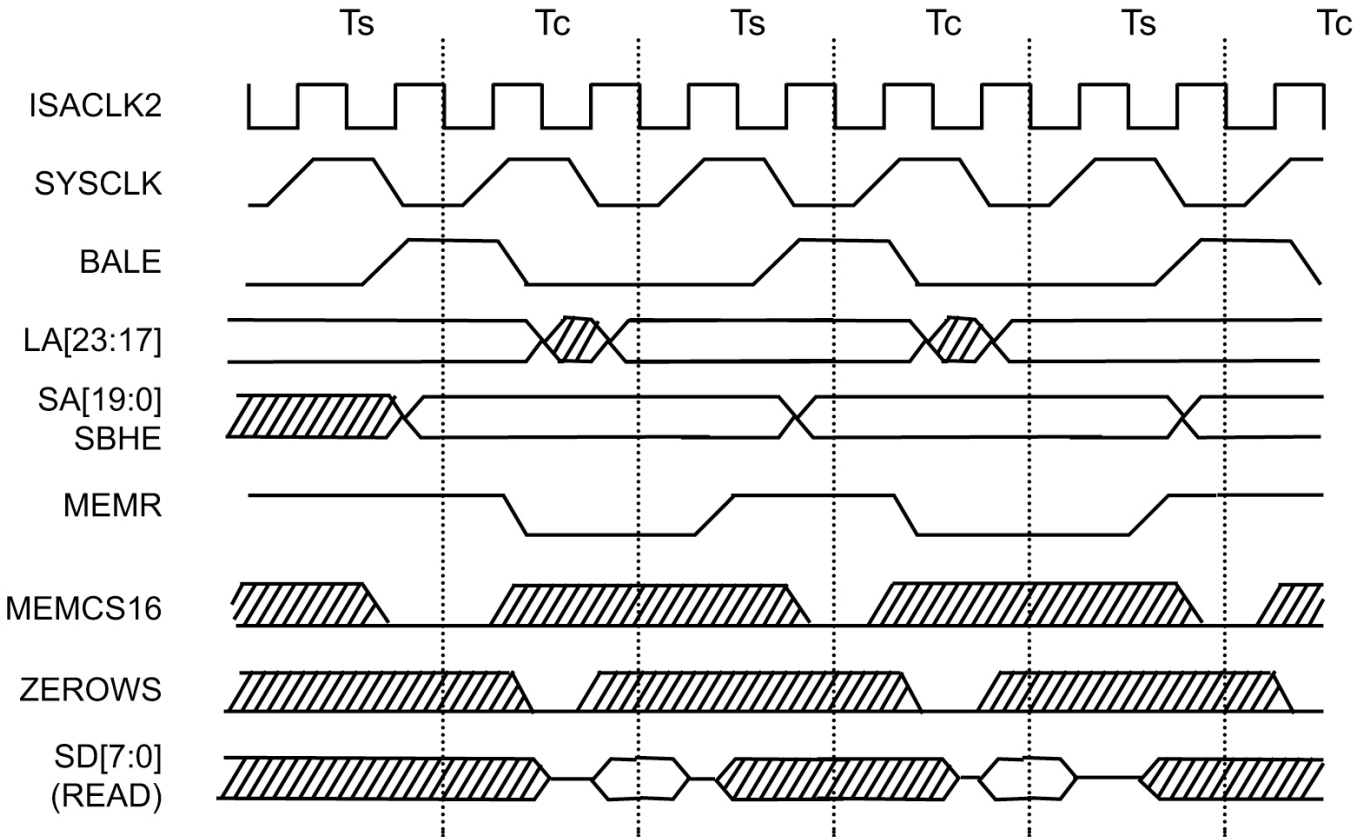


Figure 26-6 Zero WS 16-bit ISA Memory Read Cycle Time (0 WS)

26.1: ISA BUS MEMORY SIGNALS
ISA bus timing for memory – 16-bit

- Combining the effect of **MEMCS16** and **ZEROWS**.

Table 26-1: ISA Bus Memory Read/Write Cycle Time

MEMCS16	ZEROWS	Data Bus	Read Cycle Time
0	0	D0–D15	250 ns
0	1	D0–D15	375 ns
1	0	D0–D7	375 ns
1	1	D0–D7	750 ns

Example 26-1

Calculate the bus bandwidth of the ISA bus for (a) 0 WS, and (b) 1 WS. Assume that all the data transfers are 16-bit (D0–D15).

Solution:

Since the ISA bus speed is 8 MHz, we have 1/8 MHz = 125 ns as a clock period. The bus cycle time for zero wait state is 2 clocks; therefore, we have:

(a) ISA bus cycle time with 0 WS is 2 × 125 ns = 250 ns.
Bus bandwidth = 1/250 ns × 2 bytes = 8 megabytes/second.

(b) ISA bus cycle time with 1 WS is 250 ns + 125 ns = 375 ns.
Bus bandwidth = 1/375 ns × 2 bytes = 5.33 megabytes/second.

26.1: ISA BUS MEMORY SIGNALS
ISA bus timing for memory – 16-bit

- A major issue in 16-bit ISA bus interfacing is the problem of odd and even banks.

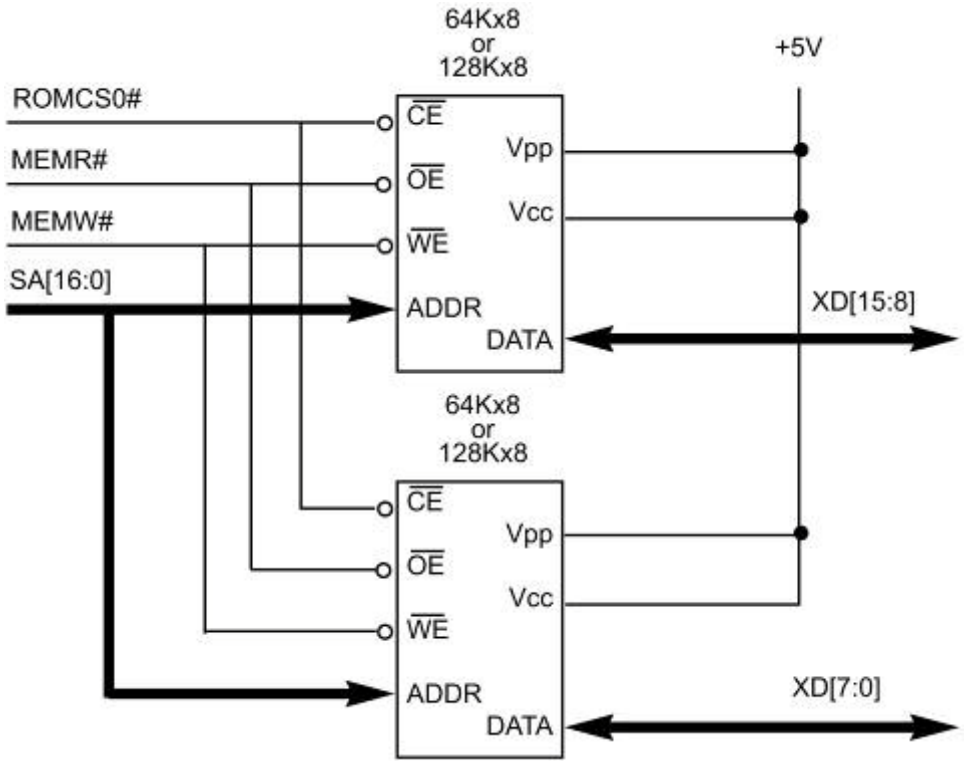


Figure 26-7 16-bit ISA Bus Connection to 2 ROMs

Assume an interface of two ROM chips, one connected to D0–D7, one to D8–D15.

The code or data must be divided into two parts and each part burned into one of the ROMs.

Many ROM burners, allow splitting the data into odd- and even-addressed bytes for 16-bit data systems.

26.1: ISA BUS MEMORY SIGNALS
DIMM and SIMM memory modules

- In the 1980s, PCs had sockets on the motherboard for DRAM chips to expand the memory.
  - With the 16-bit ISA bus, memory expansion boards became common, but were limited to the bus speed of the ISA slot, no matter how fast the memory.
    - This led to the idea of a memory module to expand.
- The problem of the lack of a standard connector. was solved with the introduction of SIP. (single in-line package).
  - SIMM (single in-line memory module) & DIMM (dual in-line memory module) were later introduced, and are now the dominant memory modules.

26.1: ISA BUS MEMORY SIGNALS
ROM duplicate and x86 PC memory map

- The memory map for the 1-megabyte memory range 00000 to FFFFFH is the same for all x86-based PCs.

  286 processors power up in real mode & fetch the first opcode from physical memory location FFFFF0H.

  CS = F000H, IP = FFF0H, and A20–A23 are all *high*.

  This leads to a physical address of FFFFF0H, 16 bytes below the top of FFFFFFH, the 16-mb maximum memory range of the 286.

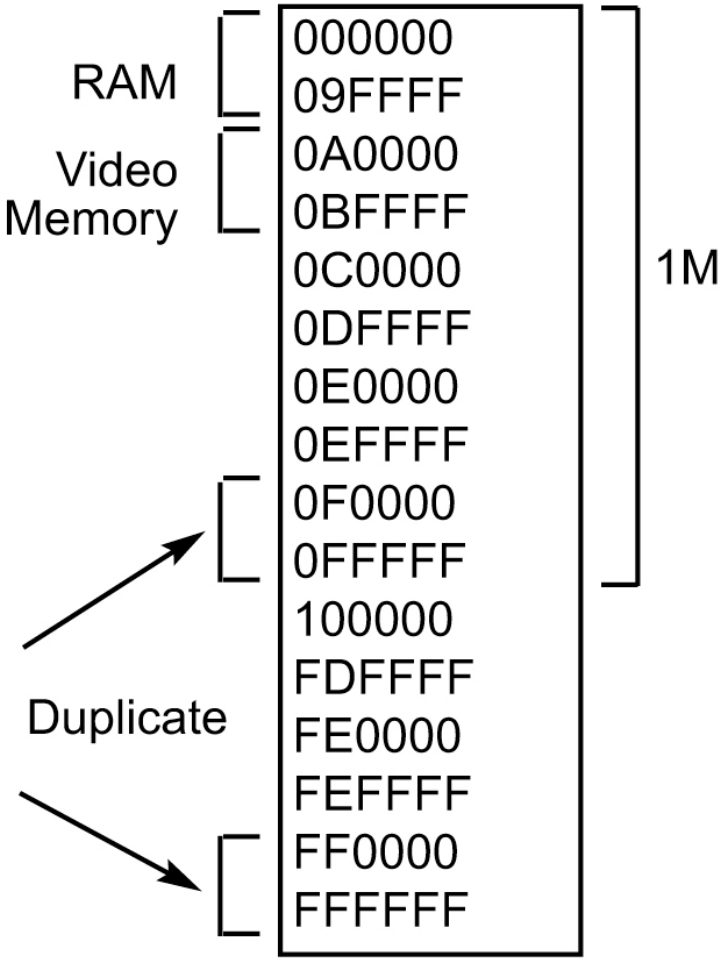


Figure 26-8 PC/AT Memory Map

26.1: ISA BUS MEMORY SIGNALS
ROM duplicate and x86 PC memory map

- The memory map for the 1-megabyte memory range 00000 to FFFFFH is the same for all x86-based PCs.

  After execution of the first opcode, pins A20–A23 all become 0s & will not be activated again unless the 286 is changed to protected mode.

  When the CPU wakes up in real mode at address FFFF0H, the first opcode is fetched from FFFFF0H because A20–A23 are all *high*.

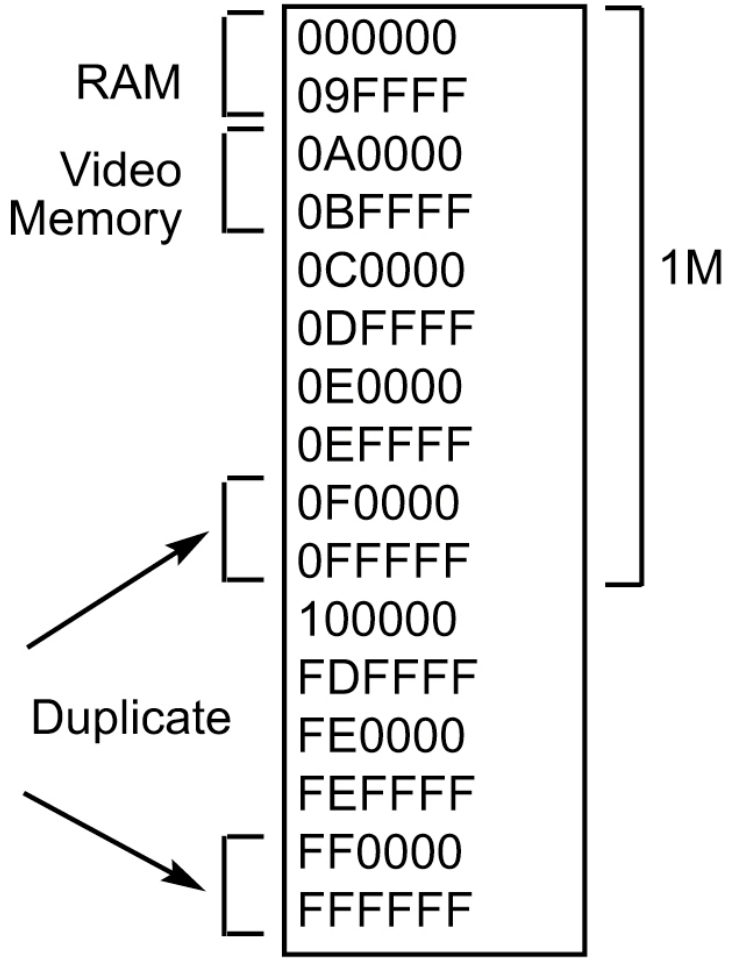


Figure 26-8 PC/AT Memory Map

26.1: ISA BUS MEMORY SIGNALS
ROM duplicate and x86 PC memory map

- There is an exact duplicate of ROM at addresses 0F0000–0FFFFF & FF0000–FFFFFF.
  - This allows access of BIOS ROM in both real & protected modes.

386/486 & Pentium® 32-bit address bus provides the memory space of 00000000 to FFFFFFFFH.

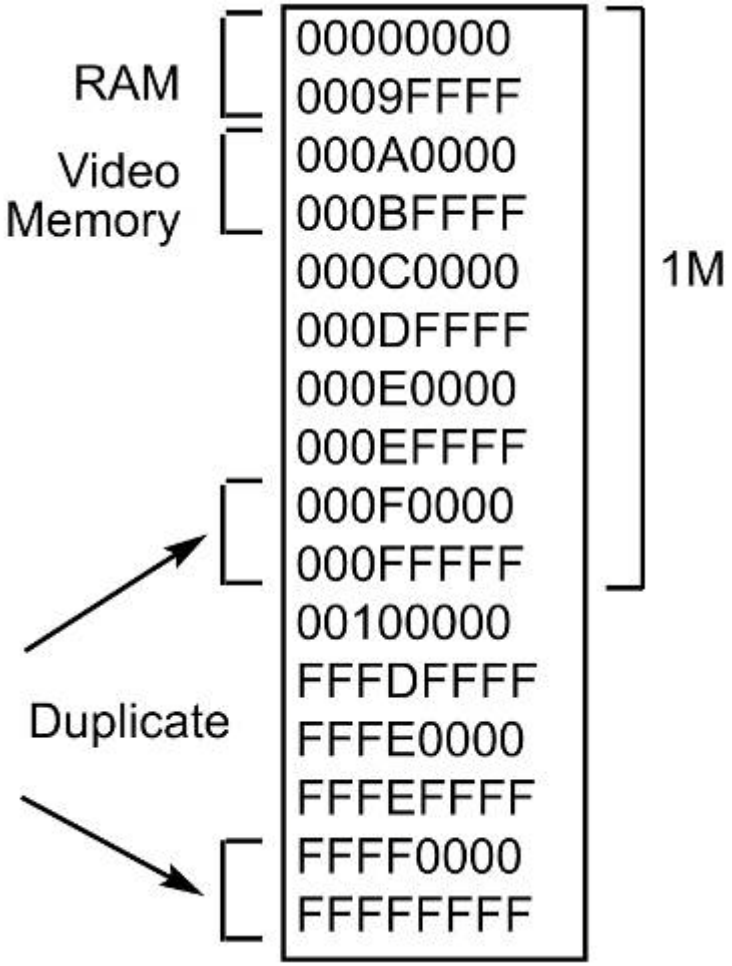


Figure 26-9a x86 PC Memory Map

26.1: ISA BUS MEMORY SIGNALS
ROM duplicate and x86 PC memory map

- This can be verified using the system tools software with Windows, under **Accessories**, **System Tools**, **System Information**.
  - Click on **Hardware Resources**, then click on **Memory**.

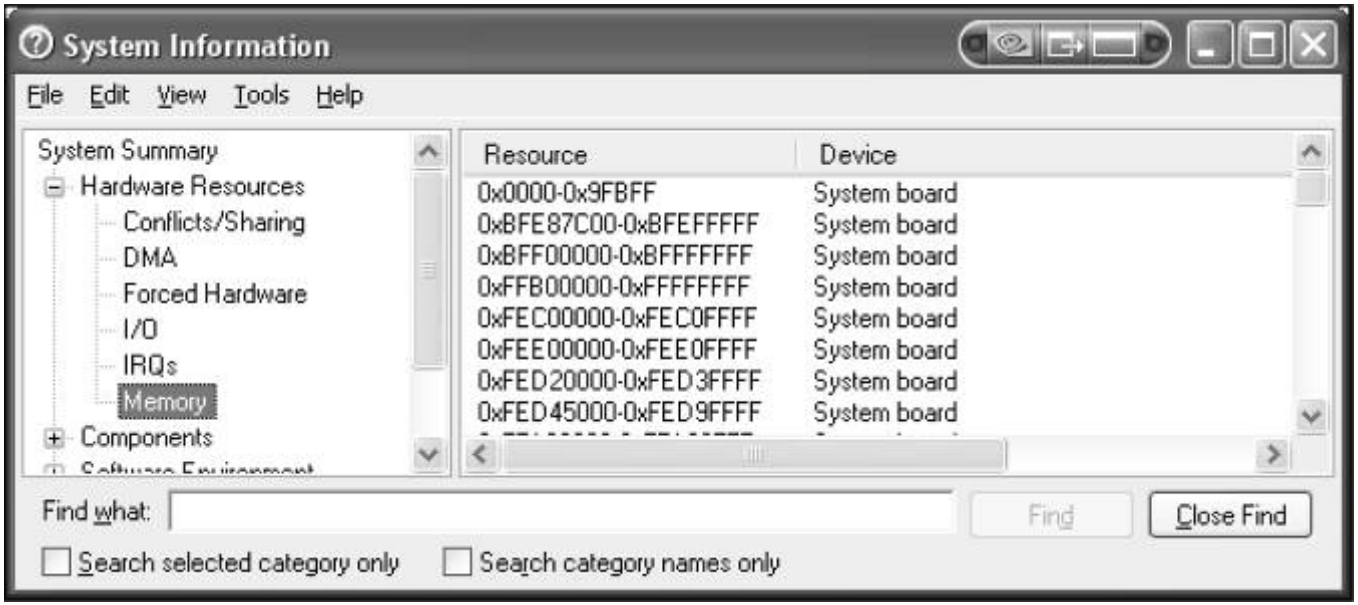


Figure 26-9b
x86 PC Memory Map from System Information Tool in Windows

26.1: ISA BUS MEMORY SIGNALS
shadow RAM

- Use System Tools to explore the system memory of Pentium® PCs, to see the duplicates of ROM in the RAM memory space.
- ROM access time is too slow for the 100 MHz bus speed, so to speed access time, its contents are copied into RAM and write protected.
  - This is called shadow RAM and provides the ROM's contents to CPU at a much faster speed than ROM.
- While the PC is on, the DRAM containing the ROM information is write protected & will not be corrupted.
  - When the PC is turned off, shadow RAM contents are lost.

26.2: I/O BUS TIMING IN ISA BUS
8-bit and 16-bit I/O in ISA bus

- The term *ISA computer* encompasses IBM PC/AT, PS, and any x86 PCs with AT-type expansion slots.
  - In communications between the x86 CPU and I/O ports, typically I/O devices are slow and cannot respond to the CPU's normal speed.
    - Wait states must be inserted into the I/O cycle.
  - When the CPU communicates with an ISA expansion slot, be it memory or a peripheral I/O port, it can use only an 8- or 16-bit data bus.
    - Even if the CPU is 32-bit, such as a 386, 486, or Pentium®.
  - ISA bus speed is limited to 8 MHz, which means inserting many wait states to access boards connected to ISA slots.
    - The chipset on the motherboard will do all these tasks.

26.2: I/O BUS TIMING IN ISA BUS
I/O signals of the ISA bus

- The following ISA slot pins are associated with I/O:
  - **SA0–SA9** (system address) - provides the I/O addresses, limited to 1024 ports.
  - **SD0–SD7** (system data bus) - provides the data path between the CPU and the I/O device.
  - **IOR** and **IOW** - *active-low,* connected to the read and write pins of I/O devices.
  - **ZEROWS** (zero wait state) - an input pin into the ISA bus and is *active-low*.
  - **IOCHRDY** (I/O channel ready) - an input signal into the ISA bus and *active-low*.

26.2: I/O BUS TIMING IN ISA BUS
I/O signals of the ISA bus

- **IOCS16** (I/O chip select 16) - an input into the ISA bus and is an *active-low* signal.
  - It informs the system that the I/O operation uses the entire 16-bit data bus D0–D15 and the present I/O cycle is a 16-bit data transfer.
    - If this input is not driven low, the ISA bus uses 8-bit data bus D0–D7.
  - On a 16-bit transfer to an 8-bit peripheral, if **IOCS16** is not pulled *low* the data transfer is performed in two consecutive I/O cycles.
    - Each transferring one byte at a time, which requires more time.

26.2: I/O BUS TIMING IN ISA BUS
I/O signals of the ISA bus

- **IOCS16** (I/O chip select 16) - an input into the ISA bus and is an *active-low* signal.
  - Add-in cards with a 16-bit data path use the **IOCS16** pin to instruct the motherboard not to convert a word transfer into a byte transfer.
  - This pin must be driven with an open collector or tri-state driver capable of sinking 20 mA.

Table 26-2: ISA Bus Data Transfer Summary

In 286/386/486/Pentium machines, memory (or I/O) cycle consists of 2 clocks in a zero-wait-state cycle.

Data Transfer Type (from Source to Destination)	Number of WS	Number of Clocks per Cycle
From 16-bit to 16-bit	1	3
From 8-bit to 8-bit	4	6
From 16-bit to 8-bit	10	12

26.2: I/O BUS TIMING IN ISA BUS
8-bit I/O timing & operation in ISA bus

- I/O operation of the ISA bus defaults to 8-bit and uses the D0–D7 data bus to transfer data between the I/O device and the CPU.

It completes the read (or write) cycle in 6 clocks if **ZEROWS** is not asserted *low*.

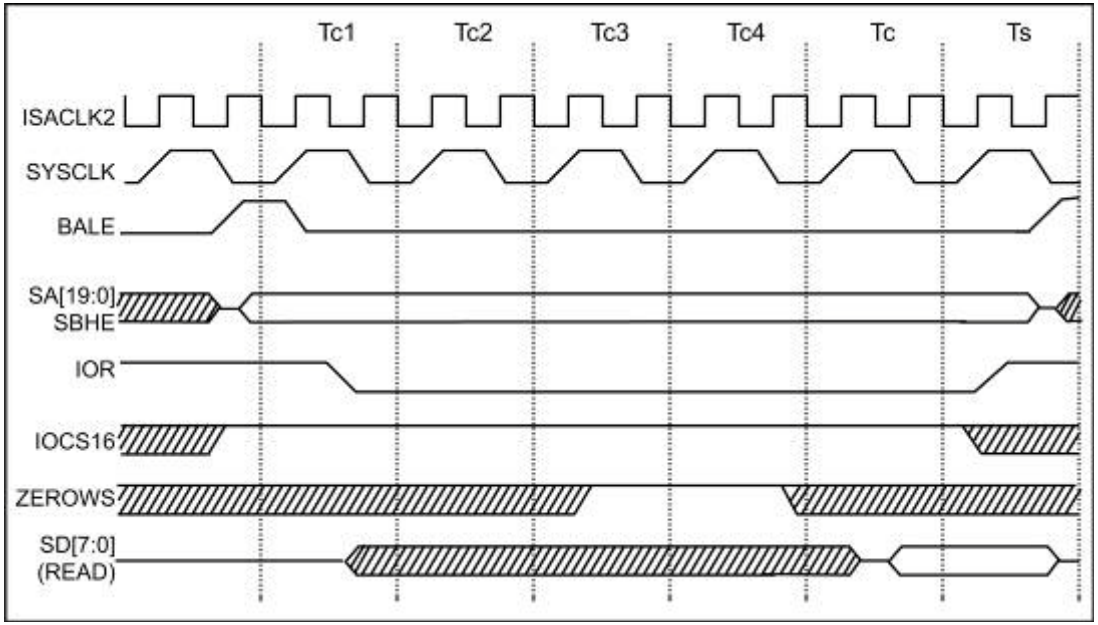


Figure 26-11
Standard 8-bit ISA
I/O Read Cycle Time (4 WS)

26.2: I/O BUS TIMING IN ISA BUS
8-bit I/O timing & operation in ISA bus

- I/O cycle time can be shortened by asserting the **ZEROWS** pin *low*, causing the I/O operation to be completed in 3 clocks instead of 6
  - The default is 4 WS unless **ZEROWS** is asserted.

The default 4 WS I/O cycle can be extended by driving the **IOCHRDY** pin *low*.

The extension happens as long as the pin is *low*.

Maximum extension is limited to 10 WS.

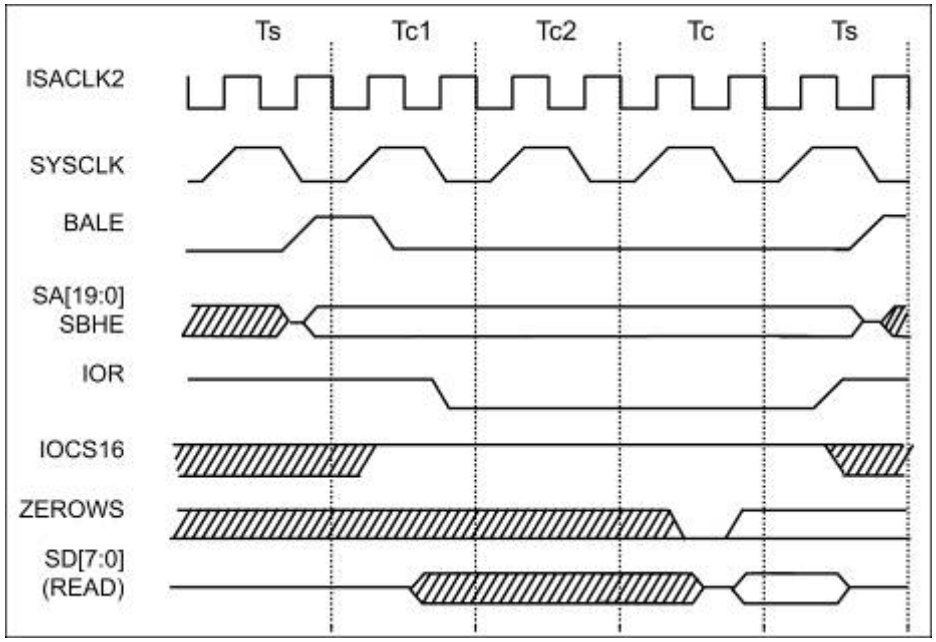


Figure 26-12 Zero WS 8-bit ISA I/O Read Cycle Time (1 WS)

26.2: I/O BUS TIMING IN ISA BUS
16-bit I/O operation & timing in ISA bus

- The 16-bit I/O port uses data bus D0–D15 to transfer data between the CPU and I/O devices.
  - The low byte uses D0–D7.
  - The high byte uses the D8–D15 data bus.
- The following is the 16-bit I/O instruction format:

	Inputting Data	Outputting Data
(1)	IN AX,port#	OUT port#,AX
(2)	MOV DX,port# IN AX,DX	MOV DX,port# OUT DX,AX

Use **AX** instead of **AL** in the 16-bit I/O.

26.2: I/O BUS TIMING IN ISA BUS
16-bit I/O operation & timing in ISA bus

- To use 16-bit data I/O, takes two port addresses, one for each byte.

```
MOV   AX,98F6H
OUT   40H,AX  ;send out AX to port 308H & 309H
```

F6H, the content of **AL**, goes to port address 40H while 98H, the content of **AH**, is transferred to port address 41H.

The low byte goes to the low port address, the high byte goes to the high port address.

26.2: I/O BUS TIMING IN ISA BUS
16-bit I/O operation & timing in ISA bus

- To use 16-bit data I/O, takes two port addresses, one for each byte.

```
MOV    DX,310H
MOV    AX,98F6H
OUT    DX,AX   ;send out AX to port 310H & 311H
```

Again the F6H is sent to port address 310H using data path D0–D7 and 98H goes to port address 311H using the D8–D15 data path.

26.2: I/O BUS TIMING IN ISA BUS
16-bit I/O timing & operation via ISA bus

- The 16-bit standard data read cycle time for the ISA bus with one WS is shown here:

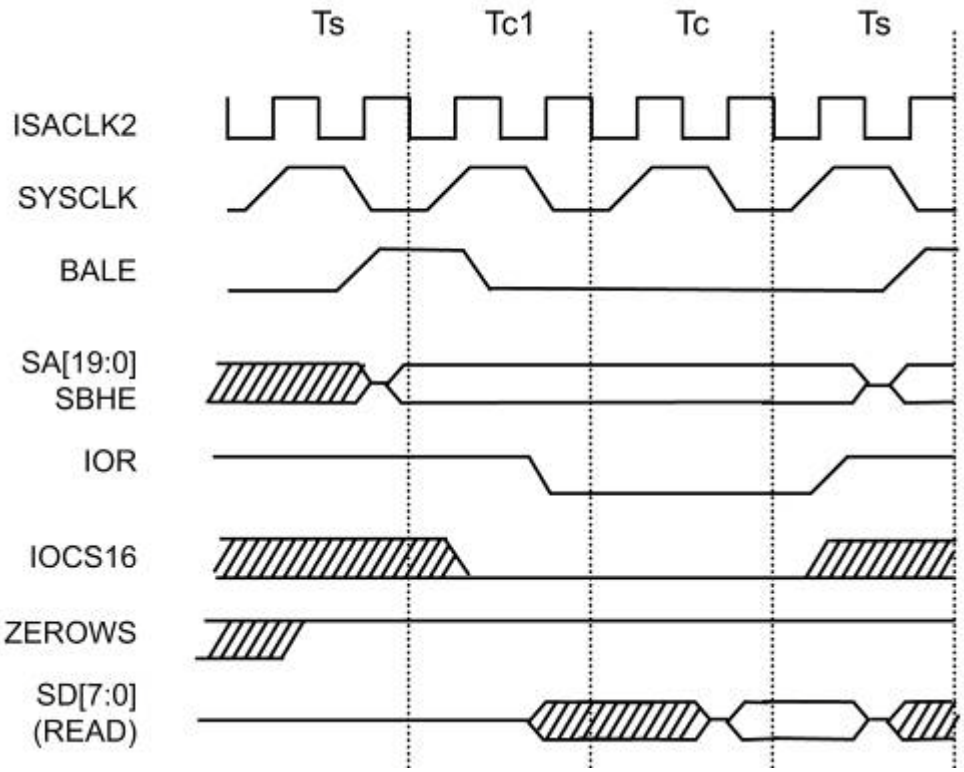


Figure 26-13 Standard 16-bit ISA I/O Read Cycle Time (1 WS)

Unlike memory, zero wait state bus activity for 16-bit I/O is not possible .

ZEROWS has no effect on 16-bit I/O operations, and the standard 16-bit ISA bus cycle timing is completed in 3 clocks.

16-bit I/O bus cycle time, can be extended it by asserting **IOCHRDY.**

26.2: I/O BUS TIMING IN ISA BUS
I/O bus bandwidth for ISA

I/O bus bandwidth is the rate of data transfer between CPU & I/O devices, dictated by bus speed and data bus width used.

Example 26-2

Find the ISA bus bandwidth for (a) 8-bit standard, (b) 8-bit with ZEROWS asserted, and (c) 16-bit standard.

Solution:

Since the ISA bus speed is 8 MHz, we have 1/8 MHz = 125 nanoseconds for the bus clock.

(a) The standard 8-bit I/O bus cycle for ISA uses 4 WS. Therefore, the bus cycle time is 6 clocks. Now cycle time = 6 × 125 ns = 750 and the bus bandwidth is 1/750 ns × 1 bytes = 1.33 megabytes/second.

(b) If ZEROWS is asserted in 8-bit I/O, we have 3 clocks for the I/O cycle time. Therefore, we have a cycle time of 3 × 125 = 375 ns and 1/375 ns × 1 byte = 2.66 megabytes/second for bus bandwidth.

(c) For 16-bit I/O data transfers we must assert the IOCS16 pin. The I/O cycle time is 3 clocks. Therefore, we have a bus bandwidth of 1 / (3 × 125) × 2 = 5.33 megabytes/second. For 16-bit I/O, we cannot assert ZEROWS to shorten the cycle time.

26.2: I/O BUS TIMING IN ISA BUS
8-bit peripherals to a 16-bit data bus

- Processors with a 16-bit data bus use odd and even byte spaces, done with the help of pins **A0** & **BHE**.

BHE	A0	
0	0	Even-addressed words (uses D0–D15)
0	1	Odd-addressed byte (uses D8–D15)
1	0	Even-addressed byte (uses D0–D7)

Because data for even-addressed ports is carried on data bus D0–D7 and data for odd-addressed ports is carried on D8–D15, port design is a challenging issue.

26.2: I/O BUS TIMING IN ISA BUS
8-bit peripherals to a 16-bit data bus

- Because data for even-addressed ports is carried on D0–D7 and data for odd-addressed ports is on D8–D15, port design is a challenging issue.

One Possible solution…

Use two separate PPI devices, such as the 8255, one for **odd**, the other for **even** addresses.

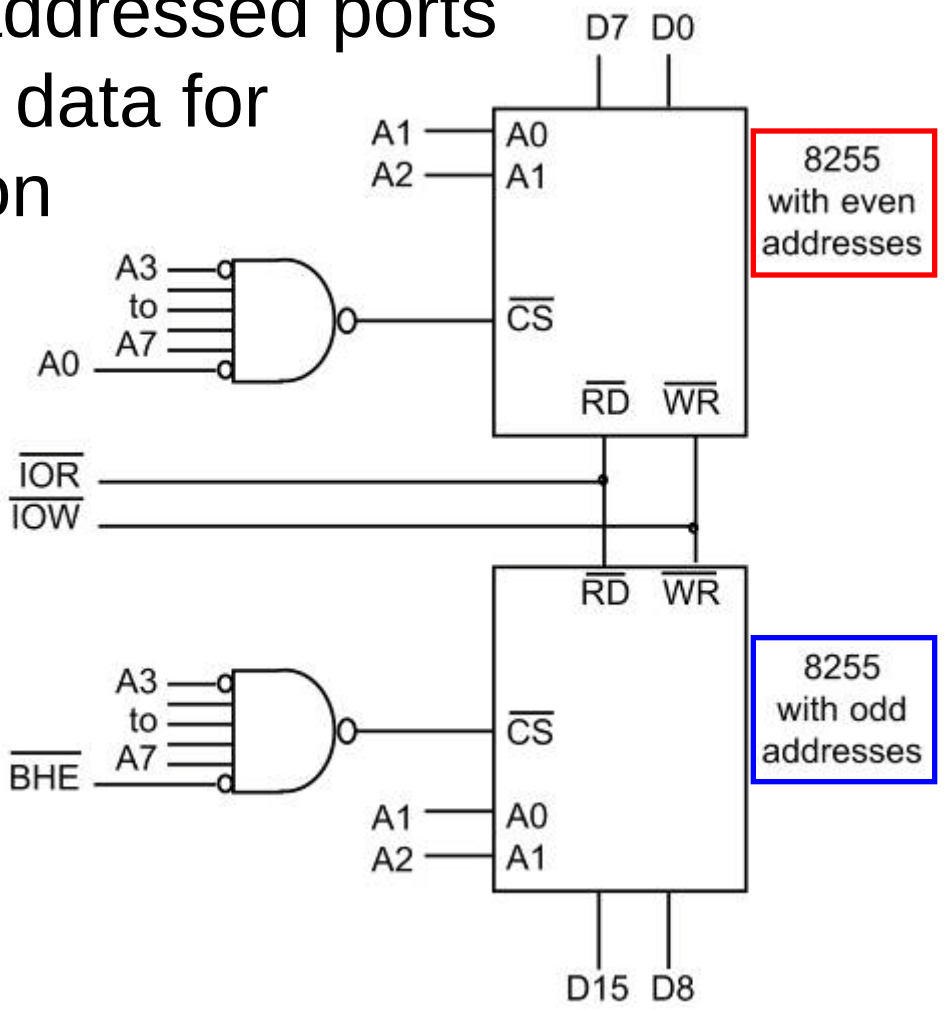


Figure 26-14
Odd and Even Ports with the 8255

26.2: I/O BUS TIMING IN ISA BUS
8-bit peripherals to a 16-bit data bus

The second solution…

Connect all 8-bit peripheral ports to data bus D0–D7.

What IBM PC/AT designers & all makers of the x86 ISA bus have done.

In such a design, one problem must be solved:

What happens when instructions such as "OUT 75H,AL" are executed?

This is the odd-addressed port, the data is provided on D8–D15, but the port is connected to D0–D7.

To solve this problem, a latch must grab the data from bus D8–D15 and provide it to D0–D7.

The latch responsible for this is called the *Hi/Lo byte copier*.

26.2: I/O BUS TIMING IN ISA BUS
limitations of the ISA bus

- The limitations of the ISA bus:
  - Data path is limited to 16 bits, unable to accommodate the 32-bit data bus of 386/486/ Pentium® processors.
  - 24-bit address bus limits the maximum memory accessible through the expansion slot to 16M.
    - Unable to accommodate 386/ Pentium® 32-bit address bus.
  - The ISA expansion slot is bulky, with a large surface contact, resulting in a massive capacitance & inductance load on each signal.
    - Accumulated capacitance/inductance associated, plus crosstalk, limits the working frequency to 8 MHz.

26.2: I/O BUS TIMING IN ISA BUS
limitations of the ISA bus

- The limitations of the ISA bus:
  - Since interrupts (IRQs) are edge triggered, each can be assigned only to a single device, with no interrupt sharing.
    - In high-frequency systems, edge-triggered interrupts can result in false activation due to a spike or noise on the IRQ input.
  - ISA DMA channels have a 16M address space limitation, due to the availability of the A0–A23 address bus.
    - 386 and higher systems cannot be used for DMA bus activity to transfer data to memory space located beyond 16M.

The combined effects of the limitations means performance of a 386/486/ Pentium® system is limited by its expansion slot and system design.

This led IBM and other PC makers to search for new solutions.

26.2: I/O BUS TIMING IN ISA BUS
the EISA bus

- IBM designed a whole new bus standard, radically different from ISA, called IBM Micro Channel.
  - Micro Channel was *not* made open architecture by IBM
    - It never became popular & was eventually discontinued.
- Other PC makers used a local bus or EISA bus.
  - EISA was in reality an *upgrade* of ISA, and while many ISA limitations were removed, EISA was still limited to 8 MHz, keeping it completely ISA compatible.
    - It had the A23–A31 address lines & D16–D31 data lines to accommodate the 386/486/Pentium®.
  - The low speed & limitations of the EISA bus led x86 PC makers to develop a whole new bus, called PCI.

26.2: I/O BUS TIMING IN ISA BUS
PC104 bus and embedded PC

- The ISA bus has found a new life in the form of the PC104 bus, so named for the 6 additional pins are added to the ISA, making it a 104-pin bus.
  - ISA has 62 pins for the A & B sides, 36 pins for the C & D sides, for a total of 98 pins.
    - Of the 6 additional pins, 5 are ground, 1 is the key.
  - Voltage and current characteristics of PC104 are the same as for ISA signals.
  - PC104 form-factor is much smaller than the original ISA and widely used in embedded systems, due to lower power consumption.

26.2: I/O BUS TIMING IN ISA BUS
PC104 bus and embedded PC

All ISA signals are the same as PC104, the only difference being the extra ground pins & the key pin

Table 26-3: PC104 Signals

	J2/P2			J1/P1	
Pin	**Row D**	**Row C**	**Pin**	**Row B**	**Row A**
0	**GND**	**GND**	1	GND	IOCHK*
1	MEMCS16*	SBHE*	2	RESET	SD7
2	IOCS16*	LA23	3	+5V	SD6
3	IRQ10	LA22	4	IRQ9	SD5
4	IRQ11	LA21	5	-5V	SD4
5	IRQ12	LA20	6	DRQ2	SD3
6	IRQ15	LA19	7	-12V	SD2
7	IRQ14	LA18	8	SRDY*	SD1
8	DACK0*	LA17	9	+12V	SD0
9	DRQ0	MEMR*	10	KEY	IOCHRDY
10					

See the complete table of PC104 signals on page 677 of your textbook.

26.3: PCI BUS
terminology – master and slave

- Many devices connected together communicate with each other via address, data & control buses.
  - When one device wishes to communicate with another, it sends an address to distinguish it from others.
    - It also sends a read or write signal to indicate its intention.
- A *master* device initiates & controls communication.
  - The responding device is called the *slave*.
    - In x86-based PCs, the CPU is an example of a master and memory is an example of a slave.

26.3: PCI BUS terminology – bus arbitration

- There is only one set of global address, data, and control buses available in a given system.
  - To access the buses, a master must ask permission of the central bus arbitrator & and wait for a response.
    - Requests by more than one master to use the buses must be arbitrated in an orderly fashion,
  - Depending on system design, the central arbitrator can assign access to each master according to a priority scheme or on a first-come-first-served basis.

26.3: PCI BUS terminology – bus protocol

- To coordinate activity, buses follow a strict set of timing and signal specifications, called *bus protocol*.
  - **Synchronous protocol** - bus activity is synchronized according to a central frequency—the timing of the central clock oscillator.
  - **Asynchronous protocol** - decides when it is ready and does not operate according to the central clock frequency.
    - x86 printer interfacing is an example.
- Asynchronous is used when there is a mismatch between the bus timing of the master and slave.
- Synchronous protocol generally has a higher rate of data transfer than asynchronous protocol.

26.3: PCI BUS
definition and merits of local bus

- While microprocessor performance is rapidly rising, buses are not keeping up.
  - Many high-performance supercomputers & mainframes use proprietary buses—nonstandard & expensive.
- Other manufacturers such as Compaq developed their own memory expansion modules.
  - ISA expansion slots are used for peripheral boards and, memory expansion was done by a specially designed slot on the motherboard.
  - Often advertised as dual-bus systems, these systems featured one bus for ISA cards and the other for memory.

26.3: PCI BUS
definition and merits of local bus

- The gap between CPU & expansion slot speed started to develop when 80286 exceeded 8 MHz, and with the introduction of graphical user interface (GUI) software such as Microsoft Windows.
  - Even a 16-bit video card plugged into the ISA expansion slot was not fast enough for the demand of the graphics software.
- Lack of a bus standard for video & cards such as disk controllers forced PC board designers to come up with what is called a local bus.
  - To access the system buses at the same speed as the microprocessor, or close to it.

26.3: PCI BUS
definition and merits of local bus

- Some PC manufacturers embedded the video card into the motherboard, bypassing the ISA slot.
- Use of a specially designed processor called a graphic processing unit (GPU) on the video board provided a major improvement in PC video.
  - Its limitation lies in the fact that if graphic data needs to be transferred from disk to video RAM (or vice versa), it must *still* go through the slow ISA bus.

26.3: PCI BUS
definition and merits of local bus

Table 26-4: Bus Bandwidth Requirements for Graphics and Real-Time Video

Graphics			
Resolution (pixels)	**Colors (bits/pixels)**	**Redraw Rate (updates/s)**	**Bandwidth (bytes/s)**
640 × 480	8	10	2.9M
1024 × 768	16	10	15M
1028 × 1024	24	10	37.5M
Real-Time Video			
Frame Size (Pixels)	**Colors (bits/s)**	**Frame Rate (frames/s)**	**Bandwidth (bytes/s)**
160 × 120	8	15	288K
320 × 240	24	15	3.5M
640 × 480	24	30	26.3M
1024 × 768	24	30	67.5M

26.3: PCI BUS
definition and merits of local bus

Table 26-4 and Example 26-3 explain why Intel led the push for a high-speed bus.

The resulting bus standard is called PCI (**Peripheral Component Interconnect**)

Example 26-3

Verify the bus bandwidth requirement for each of the following.
(a) 1024 × 768 resolution, 16 colors, 10 redraw rate
(b) 640 × 480 resolution, 24 colors, 30 frames per second

Solution:

(a) Bus bandwidth = 1024 × 768 × 16 × 10 = 125,829,120 bits/second = 15 megabytes/second
(b) Bus bandwidth = 640 × 480 × 24 × 30 = 221,184,000 bits/second = 26.3 megabytes/second

26.3: PCI BUS
PCI bus

- PCI was conceived as a specification standard for peripheral connections for Intel's high-performance microprocessors such as the 80486 and Pentium®.
  - Intel developed PCI local bus specifications, available free of charge to all PC & add-in board manufacturers.
- A local bus standard with the pin-out for expansion slot connections, PCI incorporates these features:
  - Burst mode data transfer; Level-triggered interrupts.
  - Bus mastering; Automatic configuration; High bandwidth.
  - A bridge which allows any kind of add-in card based on ISA to be plugged into the PCI local bus.

26.3: PCI BUS
PCI local bus characteristics

- Important characteristics of the PCI local bus:
  - Maximum speed of 33 MHz; 32- & 64-bit data paths.
  - Burst mode data transfer of 2-1-1-1 used by microprocessors such as the Pentium®.
  - Bus mastering, for multiprocessor implementation, where any number of microprocessors can become master and take control of the buses.
  - With implementation of a bus bridge, it supports the slow ISA bus.
    - Buffers in the bridge allow the processor to write into the buffer, leaving the task of handling ISA to the bridge.

26.3: PCI BUS
PCI local bus characteristics

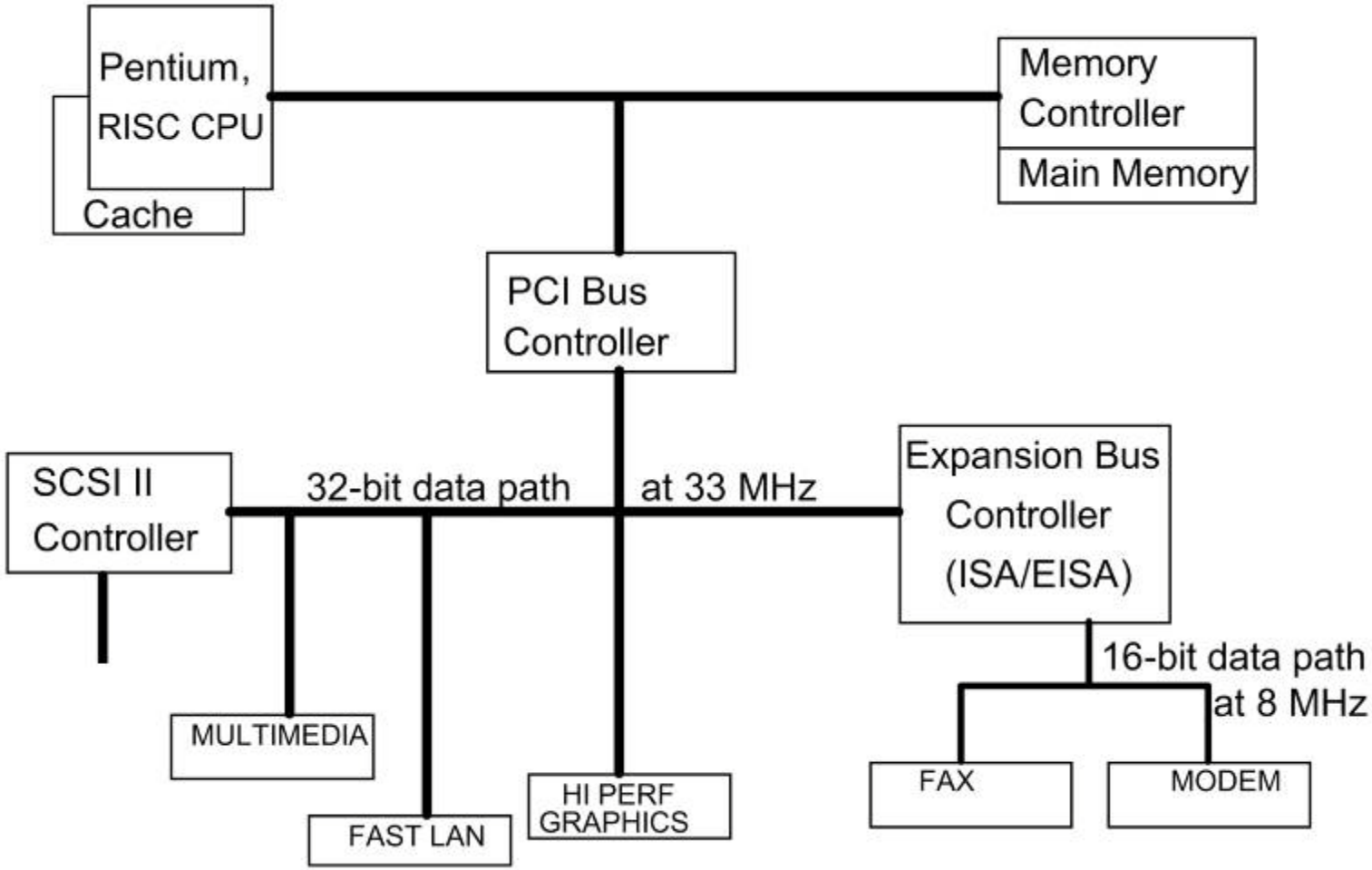


Figure 26-15 PCI Local Bus Architecture

26.3: PCI BUS
PCI local bus characteristics

The PCI local bus is processor independent.

It can be used with any microprocessor, not just the Intel x86.

For this reason, all companies have also supported the PCI & used it with their non-x86 processors.

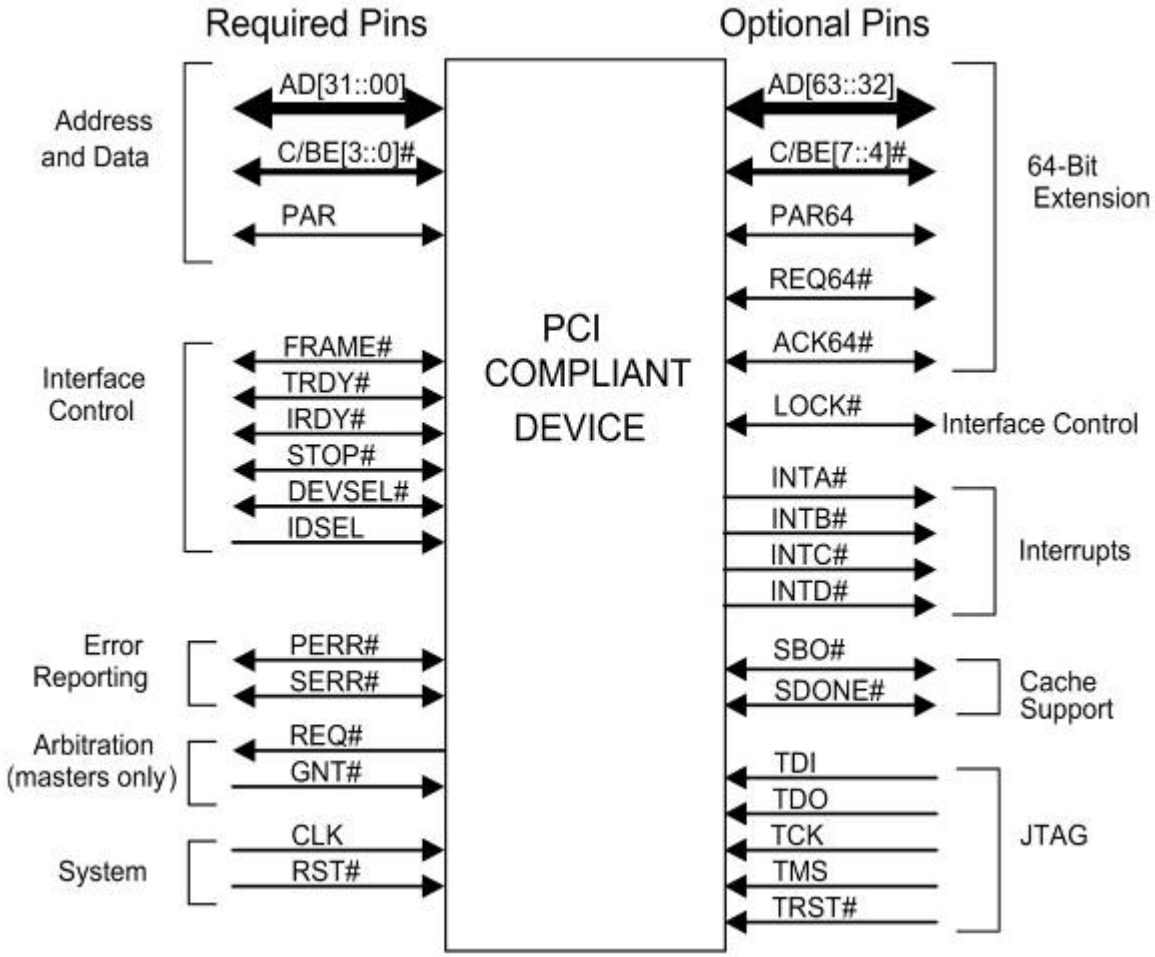


Figure 26-16 PCI Pin List

26.3: PCI BUS
PCI local bus characteristics

- More important PCI characteristics:
  - PCI supports both 5- and 3.3-V expansion cards.
    - Cutouts (keys) prevents users from plugging a card with one voltage into a motherboard with a different voltage.
  - It provides autoconfiguration capability.
    - A user can install a new add-in card without setting DIP switches, jumpers, and selecting the interrupt.
    - Level-triggered interrupts support interrupt sharing.
  - It supports up to 10 peripherals, some of which must be embedded into the motherboard.
  - Use of a highly refined connector with a small area of contact makes the PCI bus a high-frequency bus.

26.3: PCI BUS
PCI local bus characteristics

PCI has a ground or V_{cc} pin between every two signals to reduce crosstalk & radio-frequency emissions.

	5V Environment		3.3V Environment			5V Environment		3.3V Environment	
pin	Side B	Side A	Side B	Side A	pin	Side B	Side A	Side B	Side A
1	-12V	TRST#	-12V	TRST#	50	CONNECTOR KEY		Ground	Ground
2	TCK	+12V	TCK	+12V	51	CONNECTOR KEY		Ground	Ground
3	Ground	TMS	Ground	TMS	52	AD[08]	C/BE[0]#	AD[08]	C/BE[0]#
4	TDO	TDI	TDO	TDI	53	AD[07]	+3.3V	AD[07]	+3.3V
5	+5V	+5V	+5V	+5V	54	+3.3V	AD[06]	+3.3V	AD[06]
6	+5V	INTA#	+5V	INTA#	55	AD[05]	AD[04]	AD[05]	AD[04]
7	INTB#	INTC#	INTB#	INTC#	56	AD[03]	Ground	AD[03]	Ground
8	INTD#	+5V	INTD#	+5V	57	Ground	AD[02]	Ground	AD[02]
9	PRSNT1#	Reserved	PRSNT1#	Reserved	58	AD[01]	AD[00]	AD[01]	AD[00]
10	Reserved	+5V(I/O)	Reserved	+3.3V(I/O)	59	+5V(I/O)	+5V(I/O)	+3.3V(I/O)	+3.3V(I/O)
11	PRSNT2#	Reserved	PRSNT2#	Reserved	60	ACK64#	REQ64#	ACK64#	REQ64#
12	Ground	Ground	CONNECTOR KEY		61	+5V	+5V	+5V	+5V
13	Ground	Ground	CONNECTOR KEY		62	+5V	+5V	+5V	+5V
14	Reserved	Reserved	Reserved	Reserved		CONNECTOR KEY		CONNECTOR KEY	

See the entire PCI pinout chart on page 682 of your textbook.

26.3: PCI BUS
plug-and-play feature

- Microsoft & Intel worked together to equip the ISA bus with the autoconfiguration feature, often referred to as *plug and play*.
  - The feature can work completely only after the ISA cards and BIOS are equipped with autoconfiguration.
- Plug and play falls into three categories.
  - 1. Neither the motherboard BIOS nor the add-in card is equipped with the plug-and-play feature.
  - 2. The motherboard BIOS is equipped with plug and play, but the add-in card is not.
  - 3. Both the motherboard BIOS and the add-in card are equipped for plug and play.

26.3: PCI BUS
PCI connector

- Very few PCI signals match the signals of x86 microprocessors.
  - Because PCI is a *mezzanine bus*, meaning the controller sits between the CPU and the external bus connection.
    - In this way, any CPU can be used with the PCI bus.
- Address and data on the PCI bus are multiplexed, since the same pins are used for address and data.
  - In the first clock, the address is provided.
  - In the second clock, the data is provided.
- The PCI bus has a nonburst mode cycle time of 2 clocks, just like the Pentium®.

26.3: PCI BUS
PCI connector

- The PCI bus can be implemented for a 32-bit data bus or a 64-bit data bus.
- Every third pin is dedicated to ground or V_{CC}.
  - Eliminates crosstalk & allows 33 MHz bus frequencies.

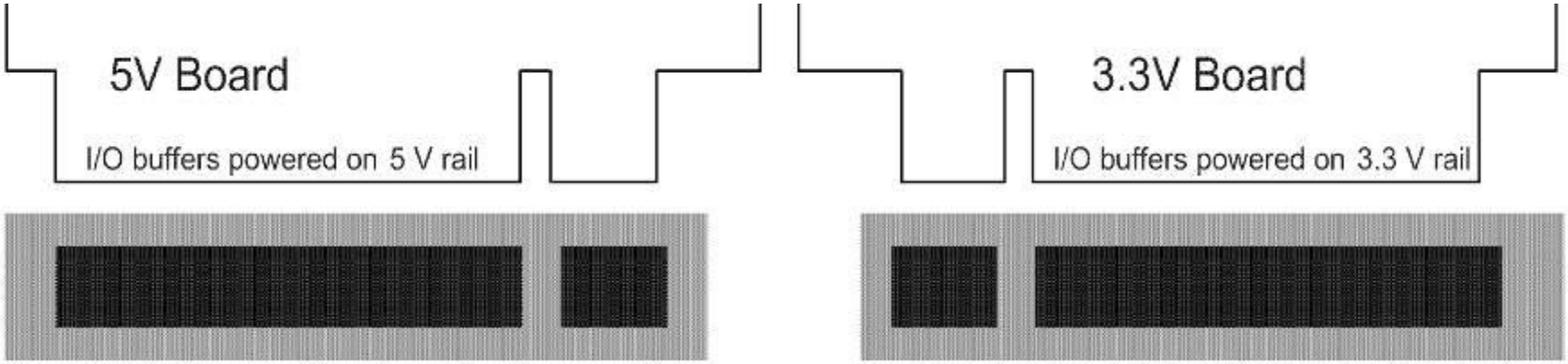


Figure 26-18 PCI Board Connectors

26.3: PCI BUS
PCI performance

- Table 26-5 provides the performance comparison of all the buses for non-burst mode data transfer.

Table 26-5: ISA, EISA, PCI Local Bus Bandwidth
In the bus bandwidth calculation, 2 clocks per memory cycle are assumed.

	ISA	EISA	PCI	PCI
Data path (bits)	16	32	32	64
Bus speed (MHz)	8	8.3	33	33
Bandwidth (megabytes/s)	8	16	66	133

26.3: PCI BUS
PCI performance

- Table 26-6 shows performance of buses and ports.

Table 26-6: Data Transfer Rate for Buses and Ports
In the bus bandwidth calculation , 2 clocks per memory cycle are assumed .

Bus/Port	Data Path bit	Maximum Bus Bandwidth
ISA (8 MHz)	16	8M bytes/second
PCI (33 MHz)	32	66M bytes/second
PCI (33 MHz)	64	133M bytes/second
PCI (66 MHz)	64	266M bytes/second
USB 2.0	1	480M bits/second
LPT	8	500K bytes/second
COM port	1	56K bits/second

26.3: PCI BUS
PCI performance

- The PCI local bus supports both single memory cycle and burst mode.
  - Single cycle takes 2 clocks to read or write a word of data.
  - In the first clock, the address is provided and in each subsequent clock, the data is accessed.
    - This makes it 2-1-1-1-1-1....

Example 26-4

Calculate the bus bandwidth of PCI for (a) single and (b) burst transfer, both on a 32-bit data path.

Solution:

PCI can work up to a maximum of 33 MHz. The clock period is 30 ns.
(a) For the single transfer, each transfer takes 2 clocks or a total of 60 ns to transfer 4 bytes (32 bits) of data. Therefore, bus bandwidth = (1/60 ns) × 4 bytes = 66.6 megabytes/second
(b) In burst mode, ignoring the overhead of the first clock for the address, it takes 1 clock or 30 ns to transfer 32-bit data. Therefore, bus bandwidth = (1/30 ns) × 4 bytes = 133 megabytes/second.

Dec Hex Bin
26 1A 00011010

ENDS ; TWENTY-SIX

The x86 PC

assembly language, design, and interfacing

fifth edition

MUHAMMAD ALI MAZIDI
JANICE GILLISPIE MAZIDI
DANNY CAUSEY

The x86 PC
assembly language, design, and interfacing
fifth edition

Prentice Hall