

# Computer Systems

## Lecture 7

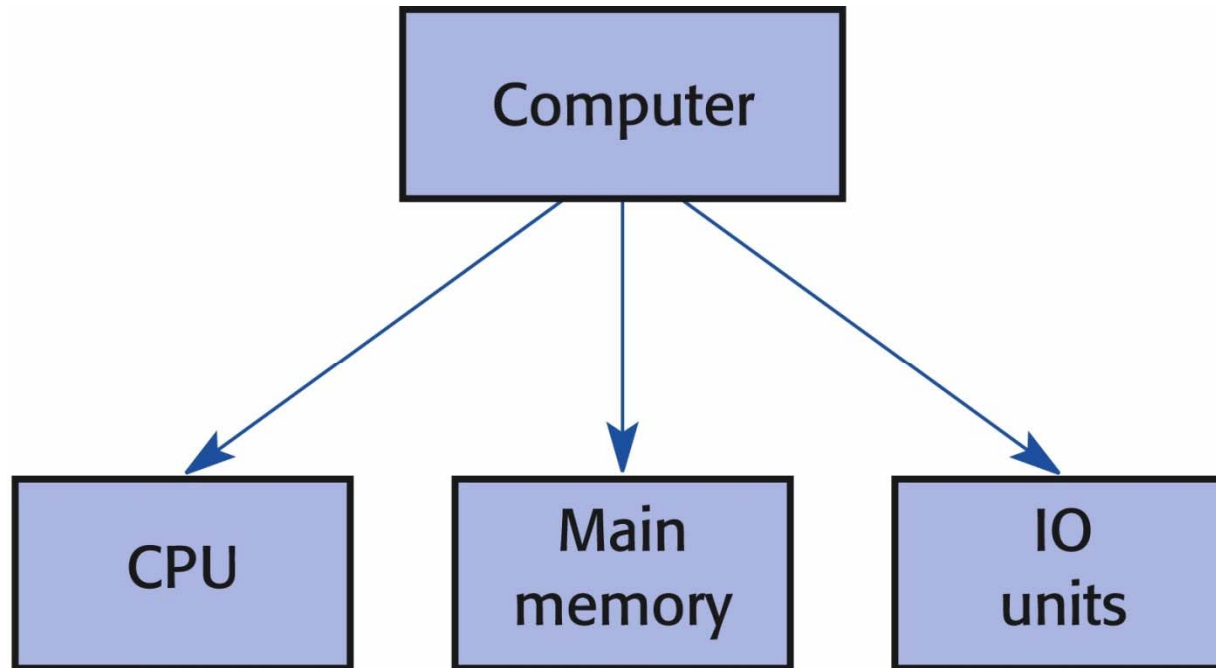
# Overview

- Principal components of a computer
- Motherboard
- Processor and Registers
- Coprocessors
- Buses
- Registers
- **Machine Cycle**
- **CISC & RISC**
- Output Hardware

# Reminder: The von Neumann Model

- The idea formulated by von Neumann (late 1940s):
  - The **computer** is a general-purpose machine controlled by an executable program.
  - A **program** is a list of instructions used to direct a task.
  - Both program and data are held in computer's **memory** (store) and both represented by **binary codes**.
  - A **processor** is an active part of the machine that executes the program instructions.

# The principal components of a computer

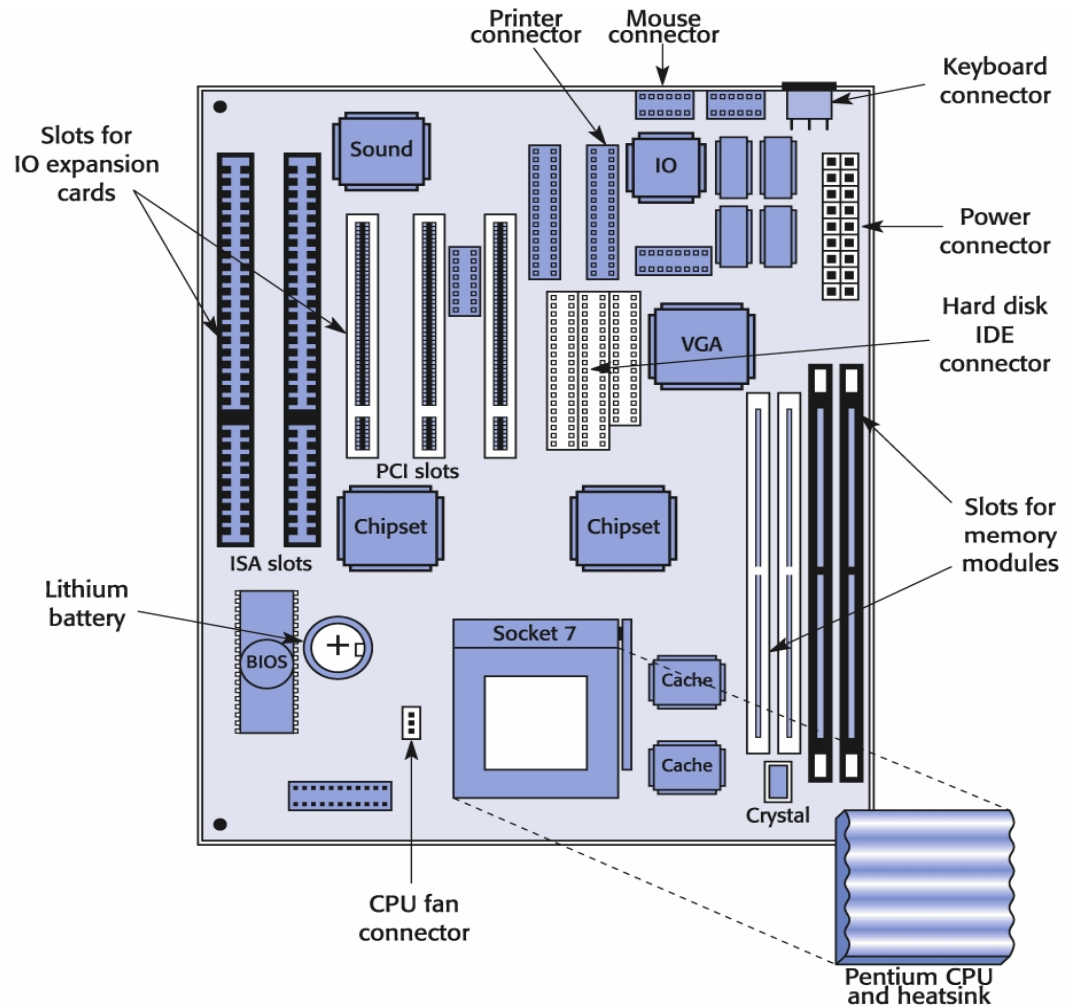


**Fig. 3.1** The principal components of a computer.

© Pearson Education 2001

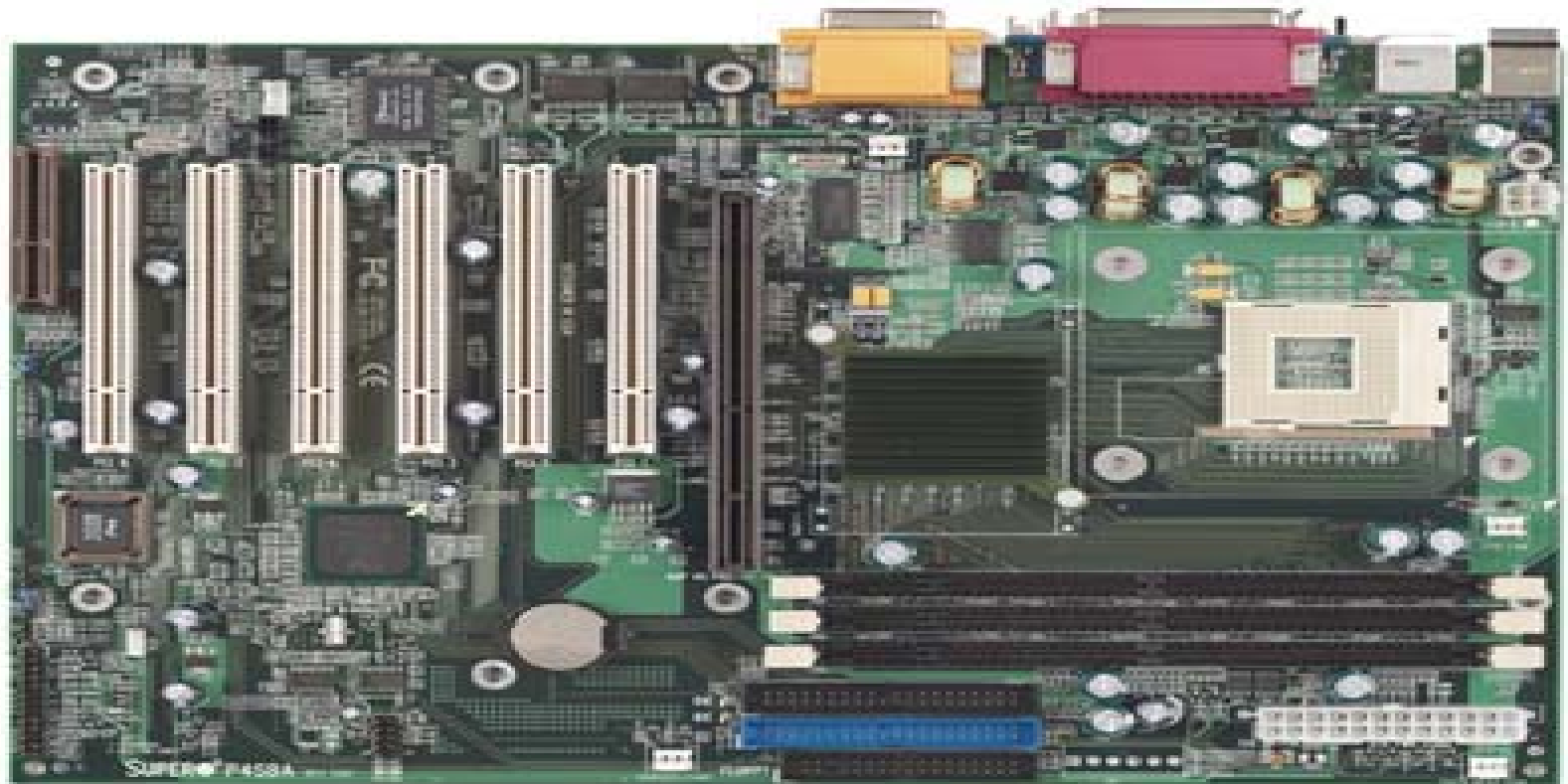
- These are the minimum set of components for a working digital computer.
- A PC motherboard often appears much more complicated.

# Motherboard



**Fig. 3.2** PC-At motherboard, showing the locations of the CPU, memory and IO card sockets.

# Pentium 4 Motherboard



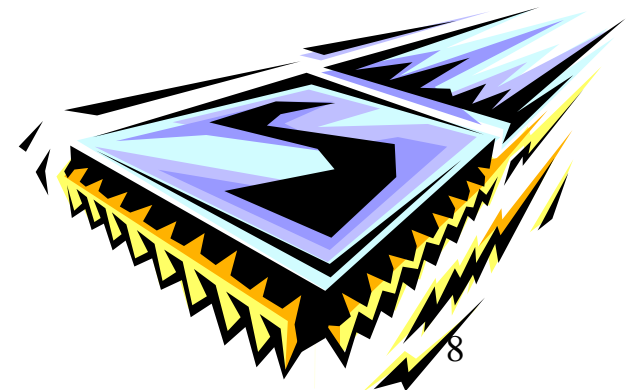
## Motherboard (cont.)

- Three principal subsystems:
  - CPU,
  - main memory, and
  - input-output units.
- Each of these is often made up of many components.
- How do they exchange data?

What is a computer processor? What does it consist of?

# Processor and Registers

- **Processor**
  - **arithmetic/logic unit (ALU)**
  - **control unit**: part of a CPU responsible for performing the machine cycle - fetch, decode, execute, store
- **Registers**
  - **Program counter (PC)**: contains the address of the next instruction to execute
  - **Instruction register (IR)**: part of a CPU control unit that stores an instruction

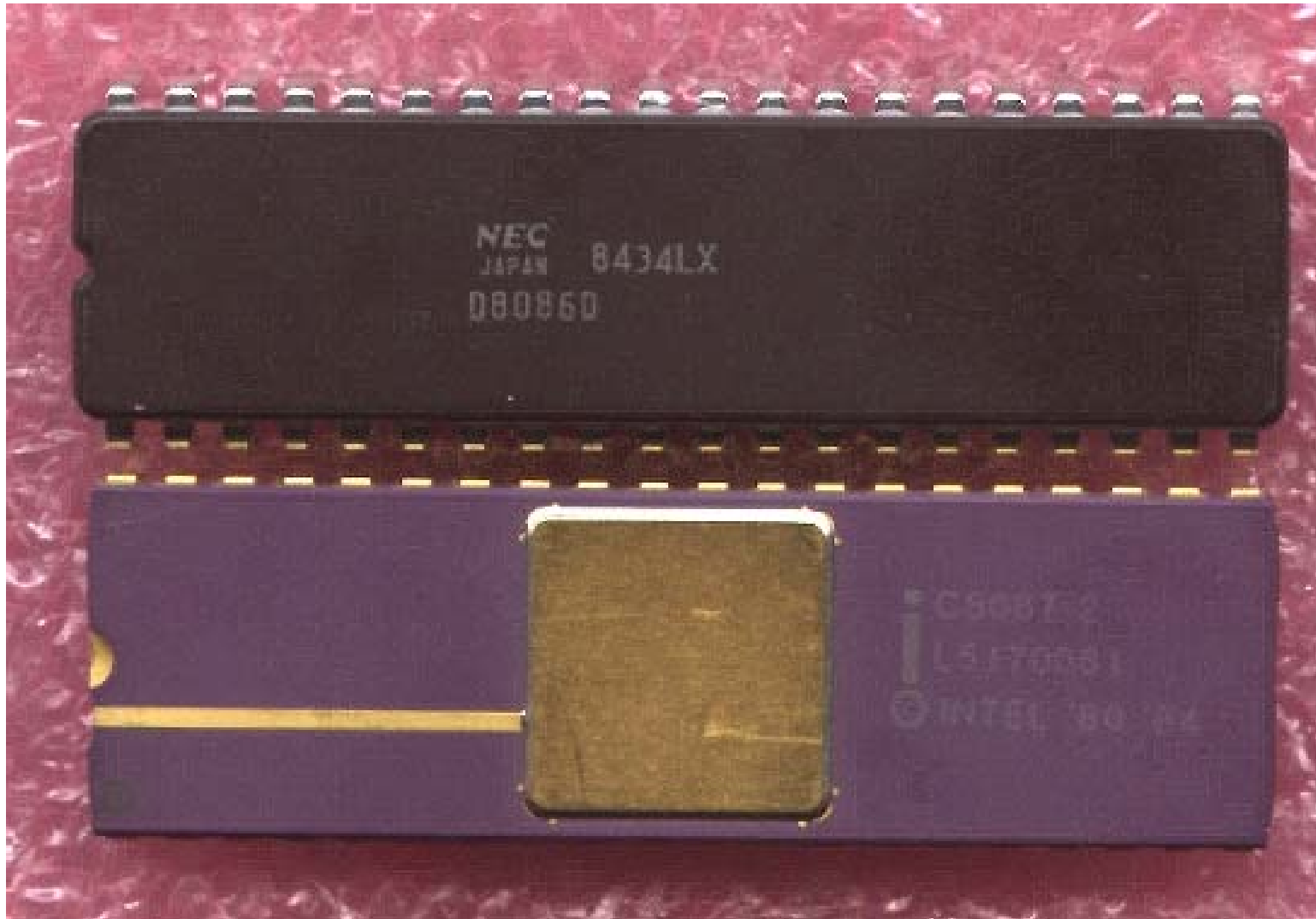




# Coprocessors: Assistants to the CPU

- Coprocessors: microprocessors performing specialized functions that CPU cannot perform or cannot perform as well and as quickly
  - math
  - graphics

# 8086 & 8087

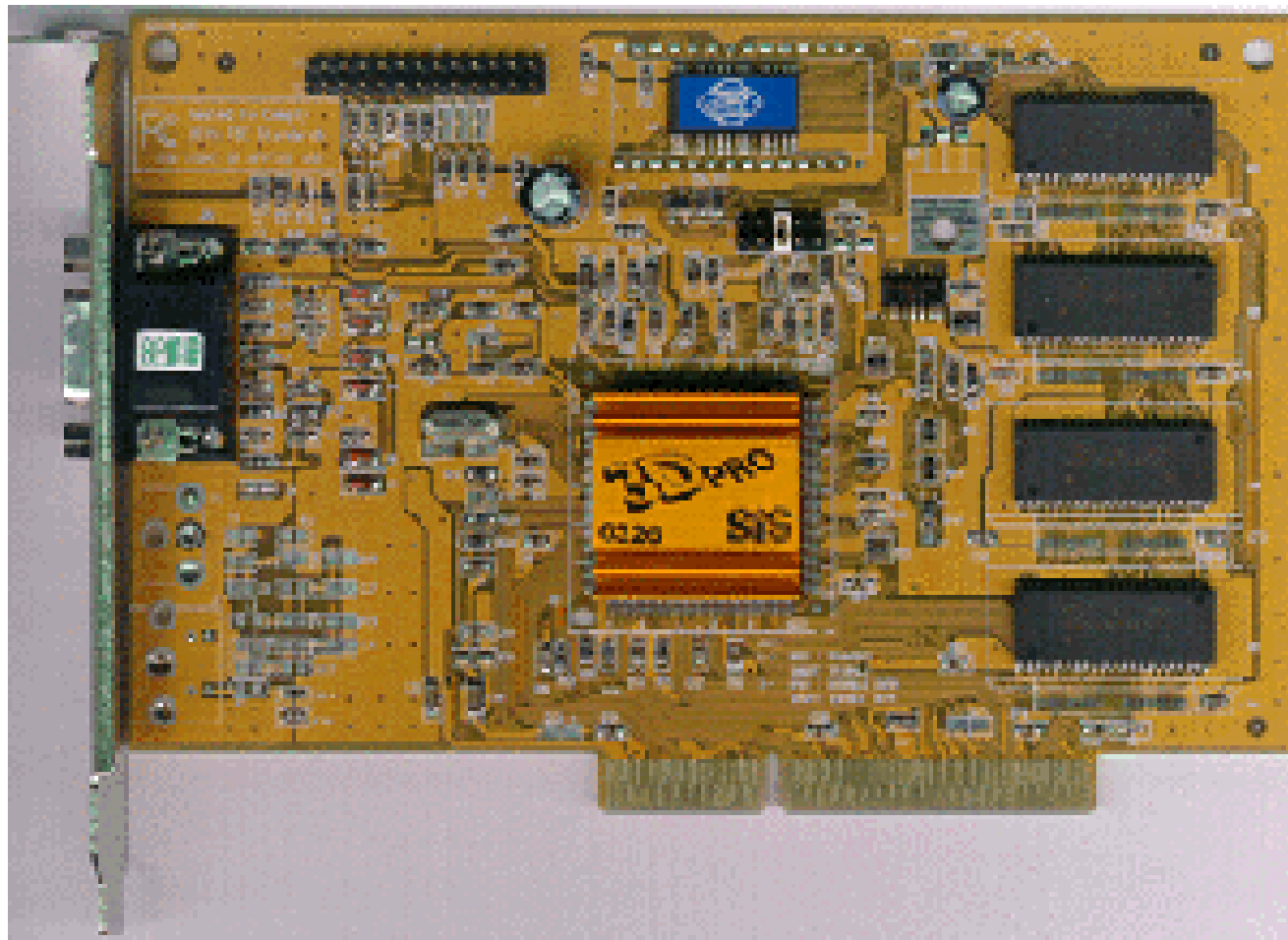


# Boards and Chips

- Circuit boards
- Use aluminum or copper to conduct electronic messages
- Chips of silicon
- Semiconductor

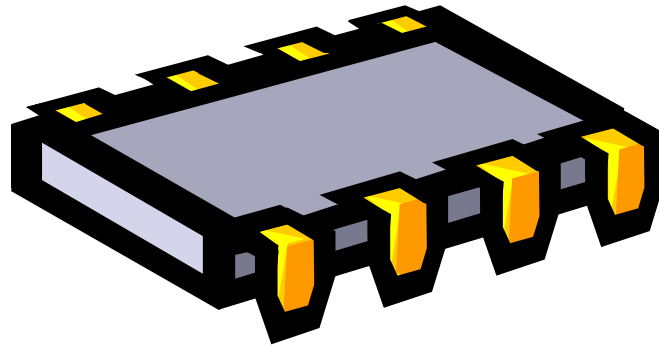


# VGA Board



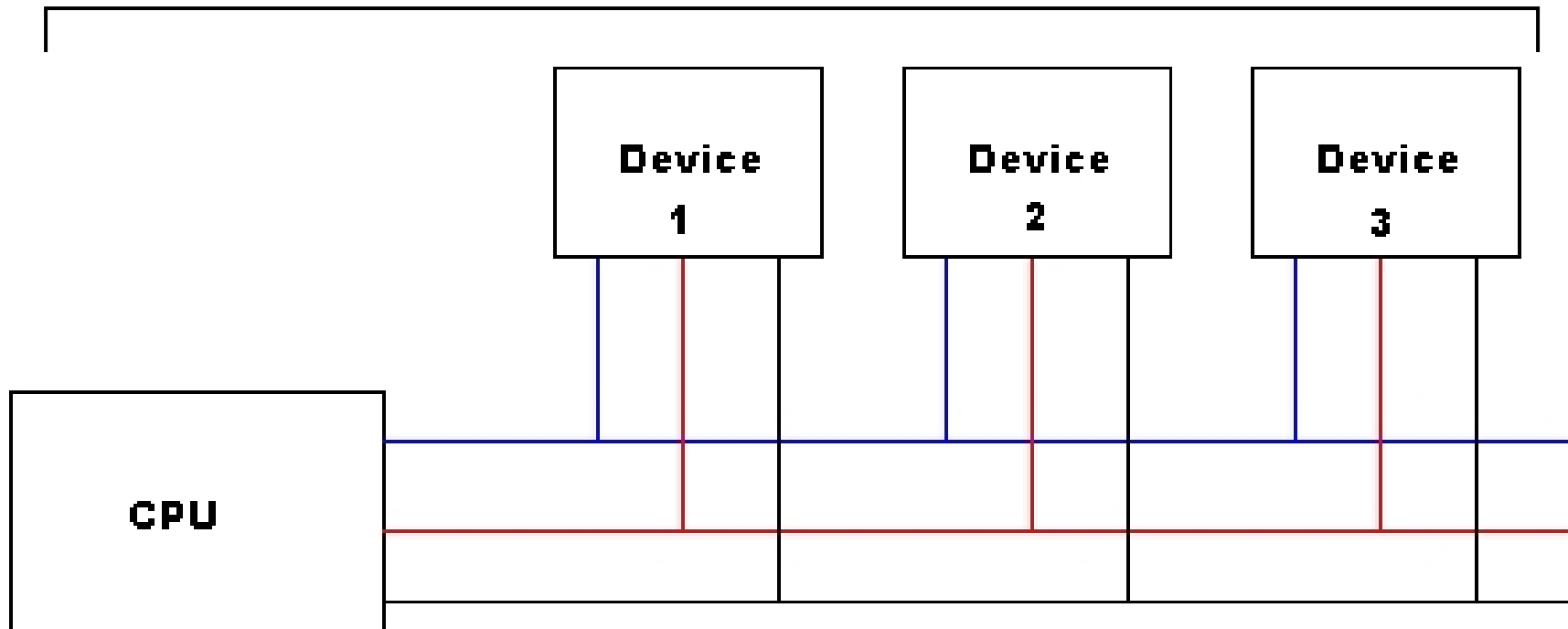
# Future

- PC on a chip

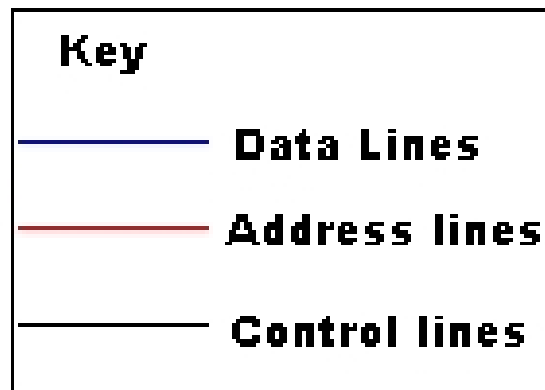


# How do they exchange data? via Buses

- On the motherboard, all the components are interconnected by **buses** (“signal highways”).
- A bus is a bundle of conductors, wires, or tracks.
- Typically, there are **address**, **data** and **control** buses, each including several signal lines.
  - Intel 8086: 20 shared address/data lines, and a further 17 lines for control.
  - Intel Pentium: data bus 64 lines, and the address bus 32 lines.



**Diagram shows how devices are attached to a generic bus.**



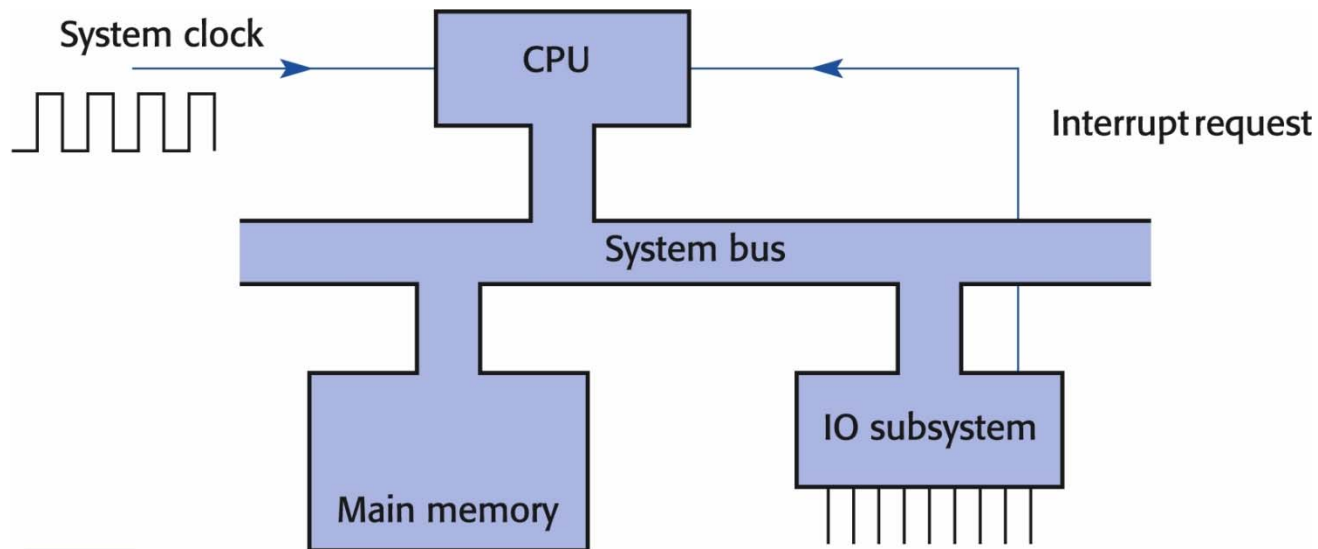
## Buses (cont.)

- Each hardware unit is connected to all these buses.
  - A simple way of building up complex systems in which each unit can communicate with each other.
  - Little disruption when plugging in new units and swapping out failed units.



# System interconnections

- The connected devices can have access to any signal line they require.



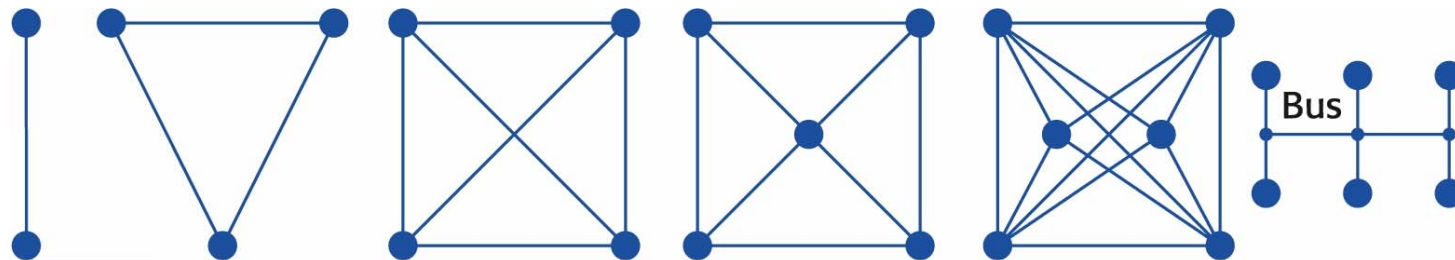
**Fig. 3.3** System interconnection schematic.

© PearsonEducation 2001

- Bus interconnection is often represented in diagrams as a wide pathway, rather than showing the individual wires.

# Bus vs. point-to-point connections

- An alternative scheme: point-to-point interconnection.
  - The number of pathways needed to link every possible pair of **n** units:  $n(n-1)/2$ .
  - Each pathway will still require a full-width data highway.
    - Could be 32 data lines and, say, 6 control lines.
  - The result will be a huge number of wires.



**Fig. 3.4** Point-to-point escalation compared with simple bus interconnection.

©Pearson Education 2001

## Bus vs. point-to-point connections (cont.)

- The number of wires required for a bus is much smaller than that for point-to-point connections.
- A bus can only transfer one item of data at a time, like a railway line.
  - Leads to a limit on the performance, termed the **Bus Bottleneck**.
  - It cannot be solved by simply increasing the speed of a processor.

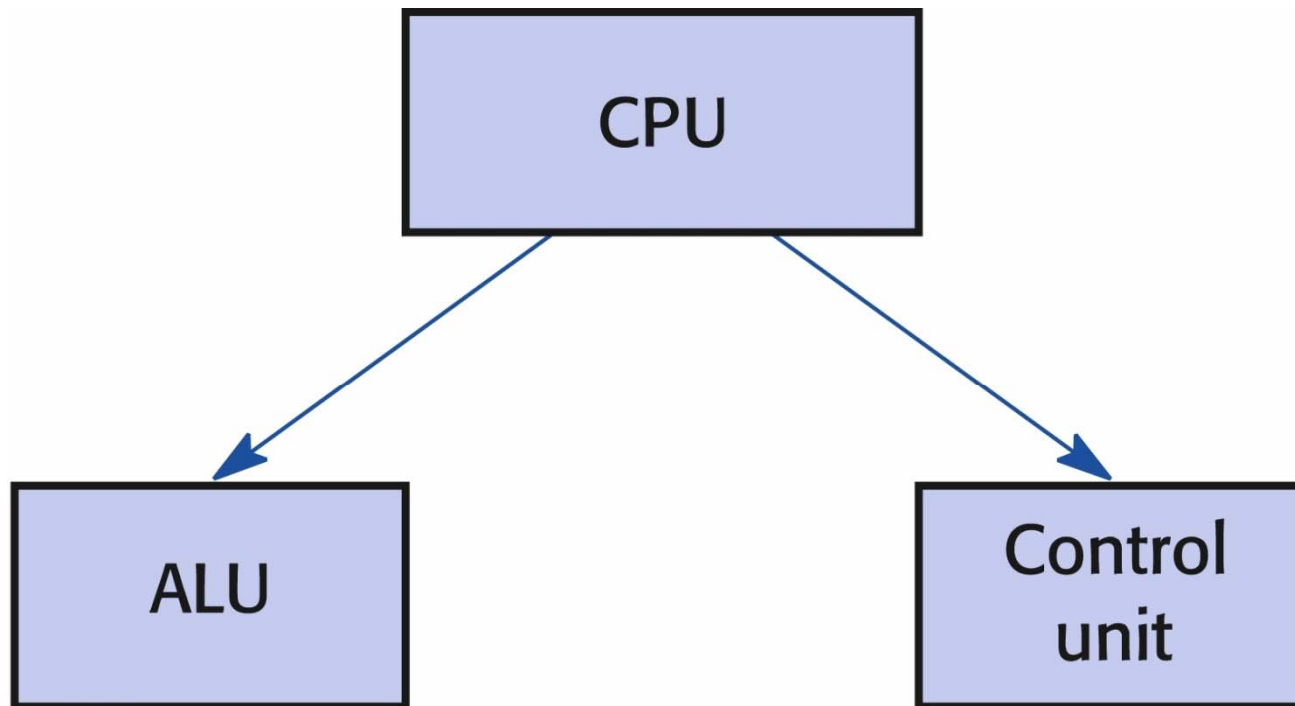
# Q&A

- Q1. What is the difference between a ‘general purpose machine’ and a ‘special purpose machine’ (‘CPU’ vs ‘coprocessor’)?

# Q&A

- Q2. assume there are 6 devices to be interconnected via 8 data lines (wires) plus 2 control lines (wires), how many wires will be needed if point-to-point connection scheme is used?

# Two parts of CPU



**Fig. 3.5** Further decomposition.

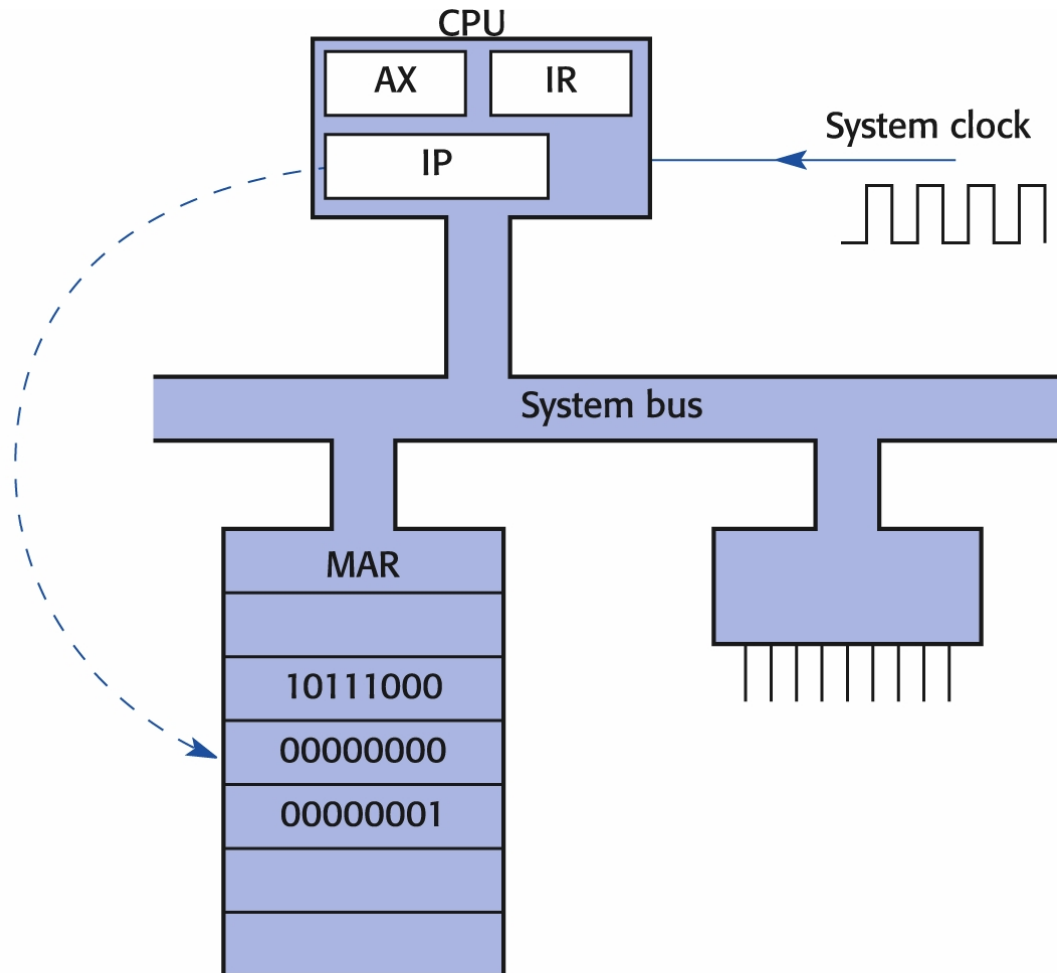
© Pearson Education 2001

- ALU: Arithmetic and Logic Unit.

# Registers

- CPU registers: small block of fast memory.
  - Temporarily store for data and address variables.
- Some CPU registers:
  - **Instruction Pointer (IP)** or Program Counter (PC).
    - Stores the address of the next instruction.
  - **Accumulator (AX, EAX in Pentium).**
    - General purpose data register.
  - **Instruction Register (IR).**
    - Stores the instruction that is being executed.
- **Memory address register (MAR).**
  - Temporarily holds address of the memory location during a bus transfer.
- **MBR**

# Registers (cont.)



**Fig. 3.6** Instruction pointer register (IP) points to the next instruction in memory.



# Computers speak this language:

## Instruction Set

- The collection of machine language instructions that a particular processor understands
- machine language instructions
  - instructions for a specific CPU
    - designed to be executed by a computer without being translated
    - Also called **machine code**
    - Operations like: ADD, SUB, INC, DEC, etc.

# How instructions are executed?

- The basic operation, known as the **fetch-execute cycle or machine cycle**.
  - The sequence whereby each instruction of the program is executed:
    - Read from the memory.
    - Decoded.
    - Executed.

# Machine Cycle

## I-cycle



Instruction fetch →

Instruction decode →

## E-cycle



Execute →

Store results →

# Machine cycle – refined

- Fetch the instruction from memory. This step brings the instruction into the *instruction register*, a circuit that holds the instruction so that it can be decoded and executed
- Decode the instruction
- [Read the effective address from memory if the instruction has an indirect address]
- Execute the instruction
- [Store the results]

# The fetch phase of the cycle (in principle)

- The address in IP register is copied onto the address bus and further to MAR register.
- IP is incremented ready for the next cycle. IP now points to the next location in the program memory.
- Memory selects location and copies the content onto the data bus.
- CPU copies the instruction code from the data bus into IR.
- Decoding of the instruction starts.

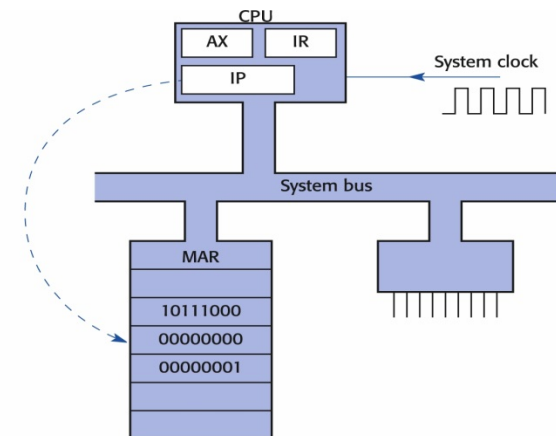
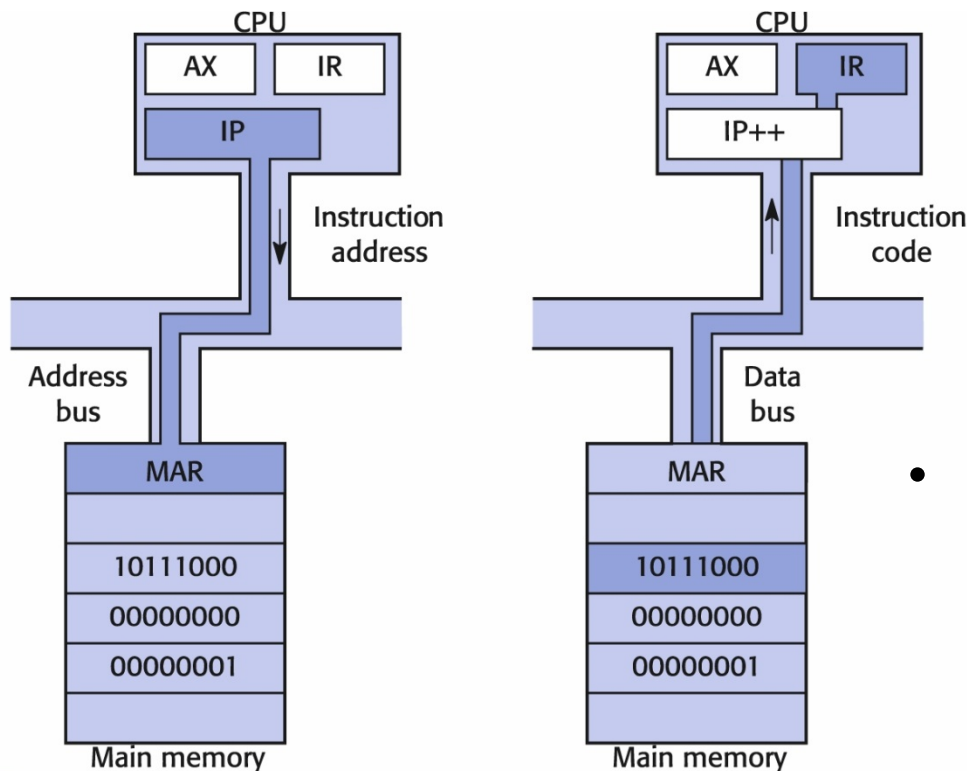


Fig. 3.6 Instruction pointer register (IP) points to the next instruction in memory.

# The fetch phase of the cycle (cont.)

- A Pentium instruction: **10111000 00000000 00000001**
- Assembly code: **MOV AX 0x100**



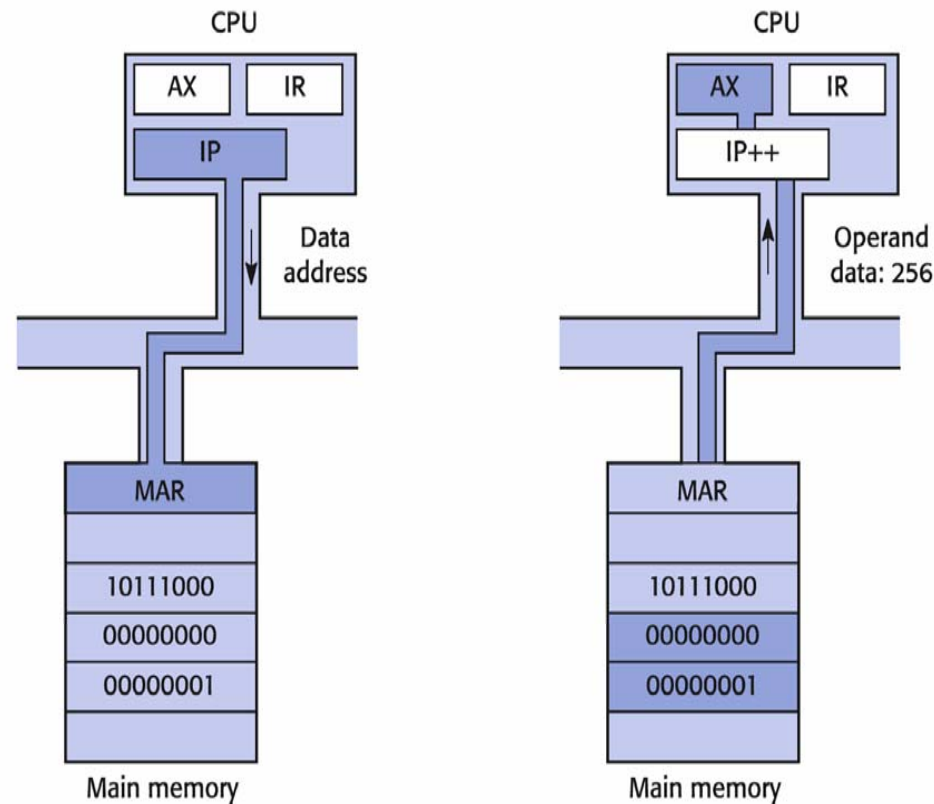
**Fig. 3.7** The fetch phase of the fetch-execute cycle.

- Note that the content of a memory cell is different from its address (not shown in the figure).

# The execution phase of the cycle

- Execute phase depends on the type of instruction.
- Example: the execution of **MOV AX, 256** instruction includes:
  - IP is copied to address bus and latched into memory.
  - IP is incremented.
  - The value selected in memory is copied onto the data bus.
  - CPU copies the value from the data bus into AX.

# The execution phase of the cycle



**Fig. 3.8** The execution phase of the fetch-execute cycle for `MOV AX, 256`.



# Q&A

- Q3. Name 2 registers that are always used during each instruction execution.

# CISC & RISC:

not all processors are designed equal

- CISC (“sisk”)
  - **complex instruction set**
  - most mainframes and PCs
- RISC (“risk”)
  - **reduced instruction set**
  - cheaper and faster
  - shift some work to software

# CISC vs RISC

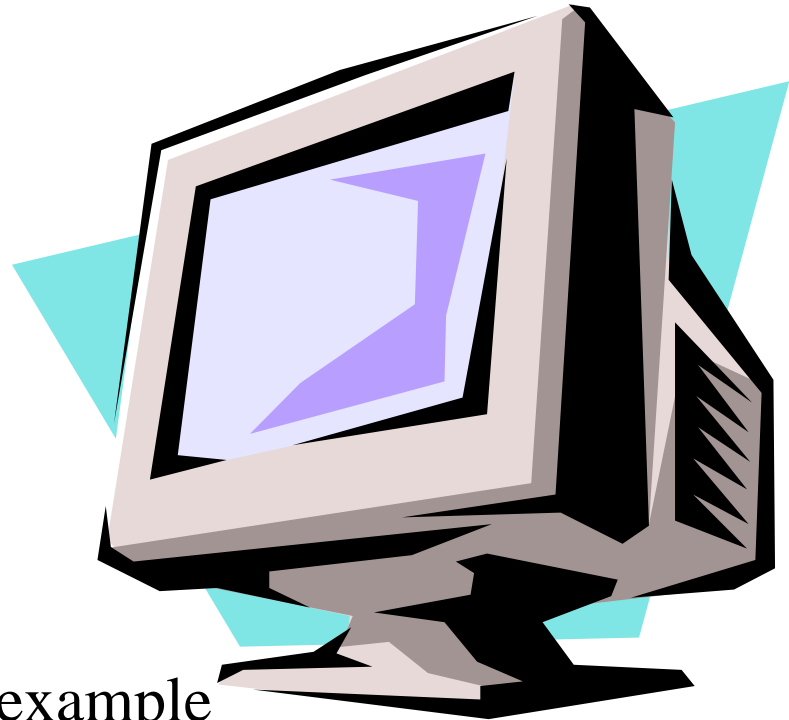
- In RISC an instruction usually consists of a single word but in CISC an instruction may be several words long, requiring several fetches

# RISC is faster because ...

- The vacated area of the chip can be used to accelerate the performance of more commonly used instructions (how?), rather than compensating for those rarely used instructions
- Easier to optimize the design
- Simplifies translation from high-level languages into the smaller instruction set that the hardware understands (Why?), resulting in more efficient programs

# Output Hardware

- Hardcopy output
  - graphics
  - letters
- Softcopy output
  - video
  - audio
    - music from MP3 for example



# Screen Clarity

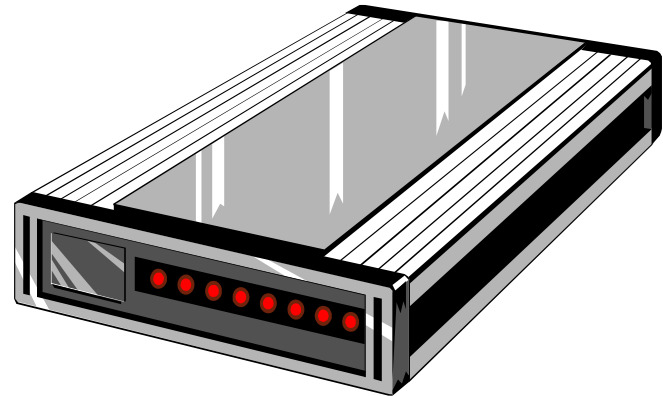
- Standard screen resolutions
  - 640 x 480
  - 800 x 600
  - 1024 x 768
  - 1280 x 1024
  - 1600 x 1200
  - in pixels ... ..

# Q&A

- How many pixels are rendered with 1024 x 768 screen resolution?

# Communications Hardware

- Facilitate networks
  - modems
  - hubs and other components of a network





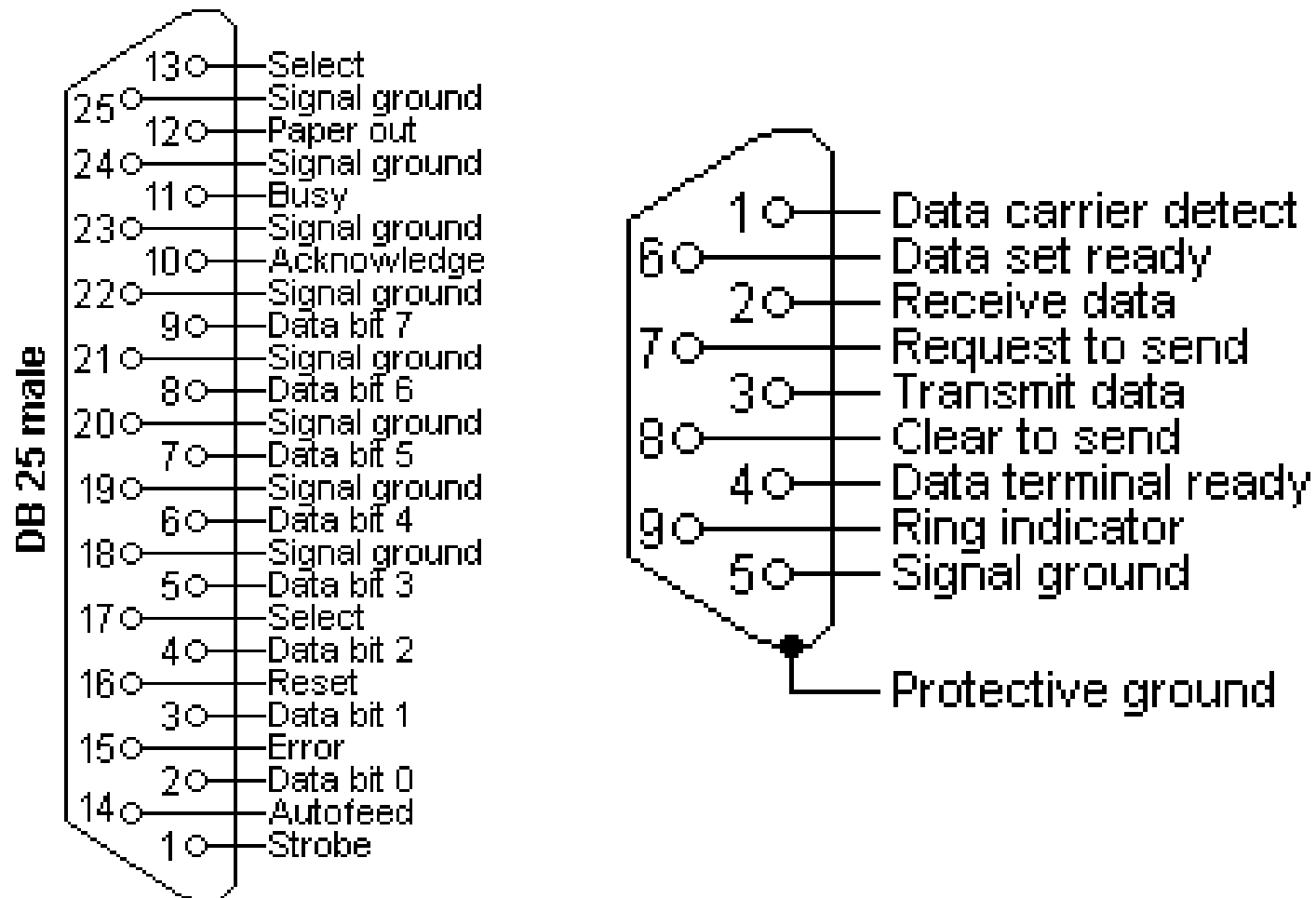
## Ports: connecting peripherals to the computers

- Parallel port (IEEE 1284)
  - printers, some scanners
- Serial port (RS-232)
  - modems, scanners, mice
- These now being replaced by ...

# USB (Universal Serial Bus)

- USB
  - industry standard developed in the mid-1990s that defines the cables, connectors and protocols used for connection, communication and power supply between computers and electronic devices
  - standardized the connection of computer peripherals, such as keyboards, pointing devices, digital cameras, printers, portable media players, disk drives and network adapters to PCs
  - replaced earlier interfaces, such as serial and parallel ports, as well as separate power chargers for portable devices

# Connectors



# What else is inside a computer?

## Power Supply

- Power supply
  - protected by power surge protector or
  - uninterruptible power supply unit (UPS)



# Q&A

- Name 2 typical applications run by coprocessors.
- Name 3 different types of bus in a computer system.
- List two reasons behind bus bottleneck.
- Highlight the 2 major components of a CPU.
- What tasks are performed during a machine cycle.
- Computers & printers are usually connected via what type of ports?

# Q&A

- RISC refers to
  - a. RAM that supports fewer instructions than do CISC chips.
  - b. instructions that support fewer codes than do CISC chips.
  - c. processors that support fewer instructions than do CISC chips.
  - d. coding schemes that are used as a back-up to CISC.

# Q&A

- Q: The computer's main processor follows its instructions to manipulate data into information.
- a. hardware
- b. CPU
- c. software
- d. Unicode

# Q&A

- Q: This type of hardware consists of devices that translate data into a form the computer can process.
- a. application
- b. input
- c. system
- d. None of the above is correct.



# Q&A

- Q. Name 4 registers that are always used during each instruction execution.

# Readings

- [Wil06] Chapter 3, sections 3.1 - 3.2.