Computer Organization

July 2016

Topics

- The Computer Level Hierarchy
- Structure & Function of Computer
- The Von Neumann Model
- Moore’s Law
- Terminology

The Computer Level Hierarchy

- Writing complex programs requires
  - A “divide and conquer” approach, where each program module solves a smaller problem

- Complex computer systems employ a similar technique through a series of virtual machine layers

Six Level Computer

LEVEL 5 — **Problem Oriented Language Level (High Language Level)**

Translation(Compiler)

LEVEL 4 — **Assembly Language Level**

Translation(Assembler)

LEVEL 3 — **Operating System Machine Level**

Partial Interpretation (Operating System)

LEVEL 2 — **Instruction Set Architecture Level**

Interpretation (Microprogram) / Direct Execution (Hardwired)

LEVEL 1 — **Micro Architecture Level**

Hardware

LEVEL 0 — **Digital Logic Level**

The Computer Level Hierarchy

- **Level 5: Problem Oriented (High-Level) Language Level**
  - Program execution and user interface level
  - The level with which we are most familiar
  - The level with which we interact when we write programs in languages such as C, Pascal, Lisp, and Java

- **Level 4: Assembly Language Level**
  - Acts upon assembly language produced from Level 5, as well as instructions programmed directly at this level

- **Level 3: Operating System Machine (System Software) Level**
  - Controls executing processes on the system
  - Protects system resources
  - Assembly language instructions often pass through Level 3 without modification

The Computer Level Hierarchy

- **Level 2: Instruction Set Architecture (ISA) Level**
  - Also known as the Machine Level
  - Consists of instructions that are particular to the architecture of the machine
  - Programs written in machine language need no compilers, interpreters, or assemblers

The Computer Level Hierarchy

- **Level 1: Control Level**
  - A *control unit* decodes and executes instructions and moves data through the system
  - A collection of 8 to 32 registers form a local memory and a circuit called an ALU capable of performing simple arithmetic operations are present
  - The registers are connected to the ALU to form a data path over which data flow
  - On some machines the operation of the data path is controlled by a program called a microprogram – *microprogrammed CU*
    - A microprogram is a program written in a low-level language that is implemented by the hardware
  - On other machines the data path is controlled directly by hardware – *hardwired CU*
    - Hardwired control units consist of hardware that directly executes machine instructions

The Computer Level Hierarchy

- **Level 0: Digital Logic Level**
  - Where find digital circuits (the chips)
  - Digital circuits consist of gates and wires
    - Each gate has one or more digital inputs and computes some simple function of the inputs such as AND or OR
  - These components implement the mathematical logic of all other levels
  - A small number of gates can be combined to form a 1-bit memory, 1-bit memories can be combined to form 16, 32, or 64 bit registers which can hold a single binary number

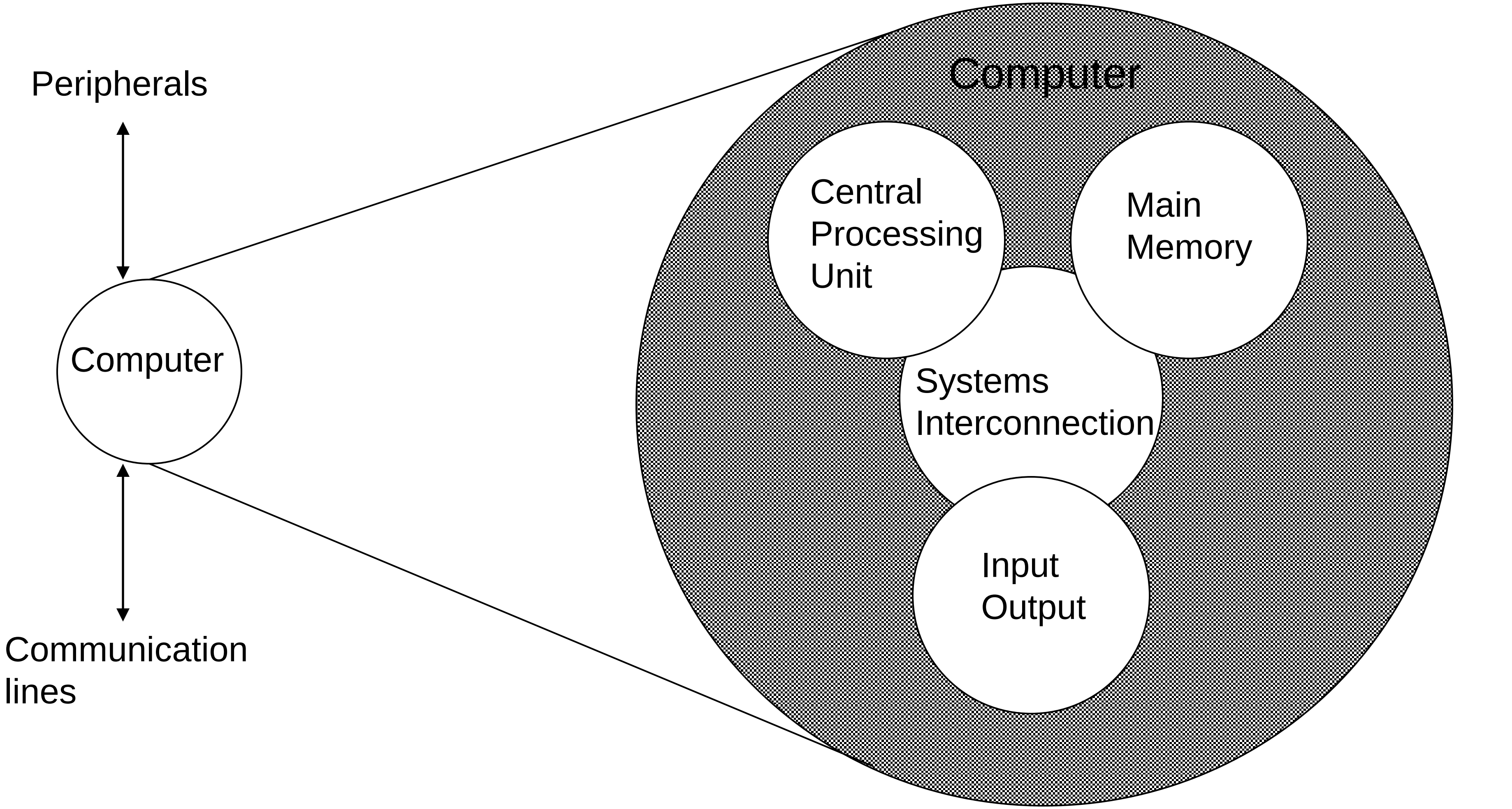
Structure & Function of Computer

- At the most basic level, a computer is a device consisting of :
  - A processor to interpret and execute programs
  - A memory to store both data and programs
  - A mechanism for transferring data to and from the outside world (Input/Output)

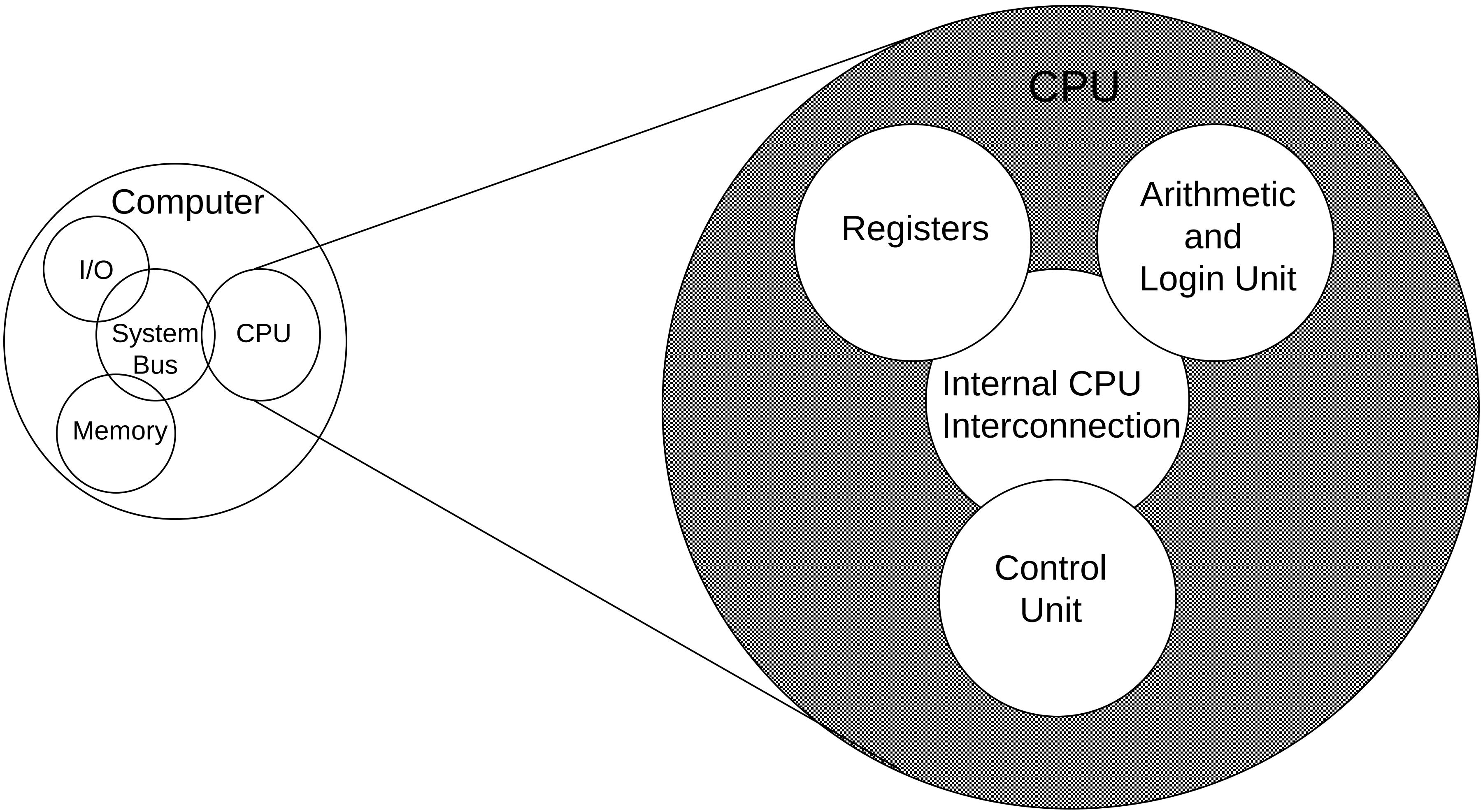
- Structure is the way in which components relate to each other

- Function is the operation of individual components as part of the structure

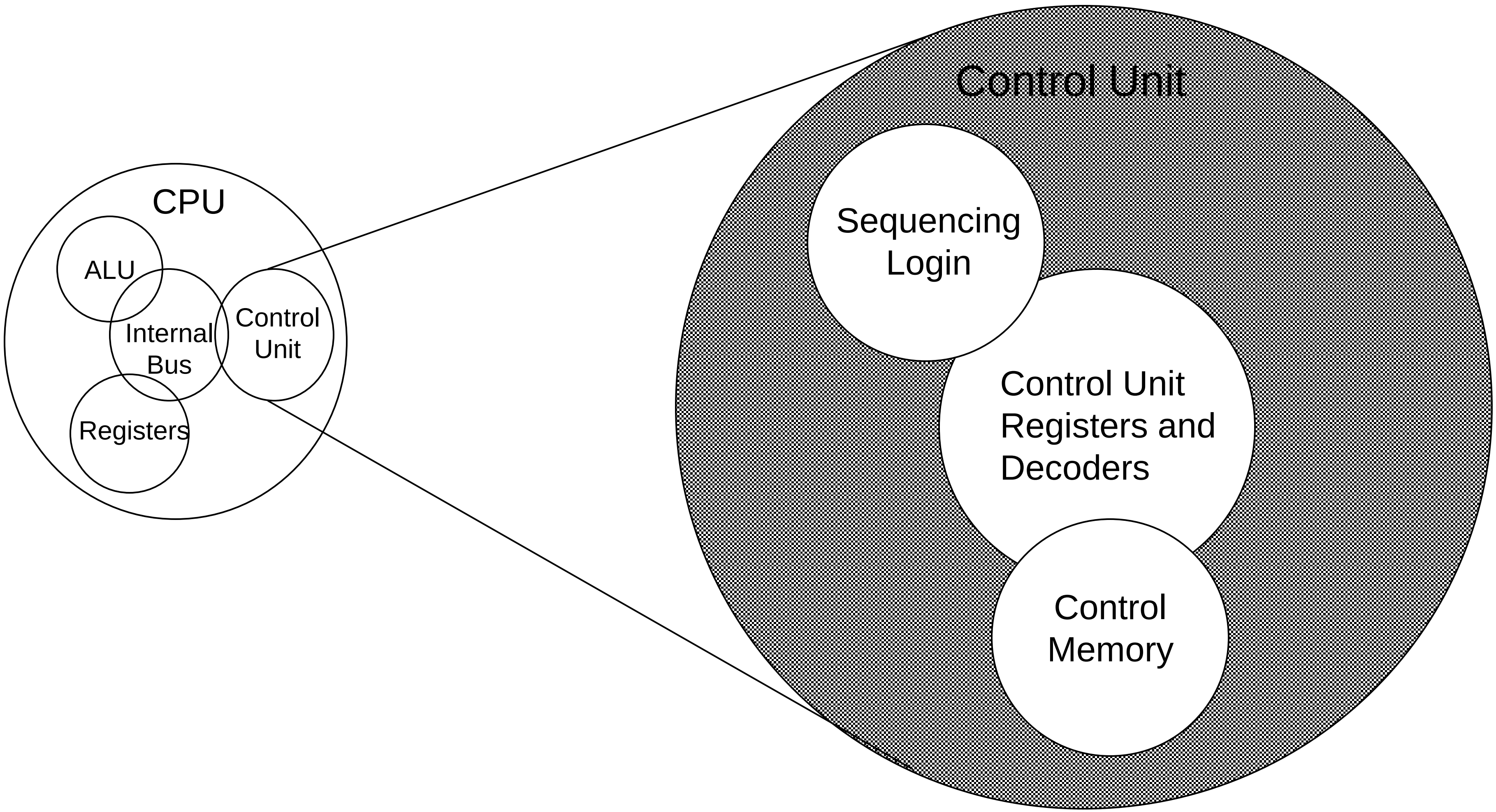
Structure - Top Level



Structure - The CPU



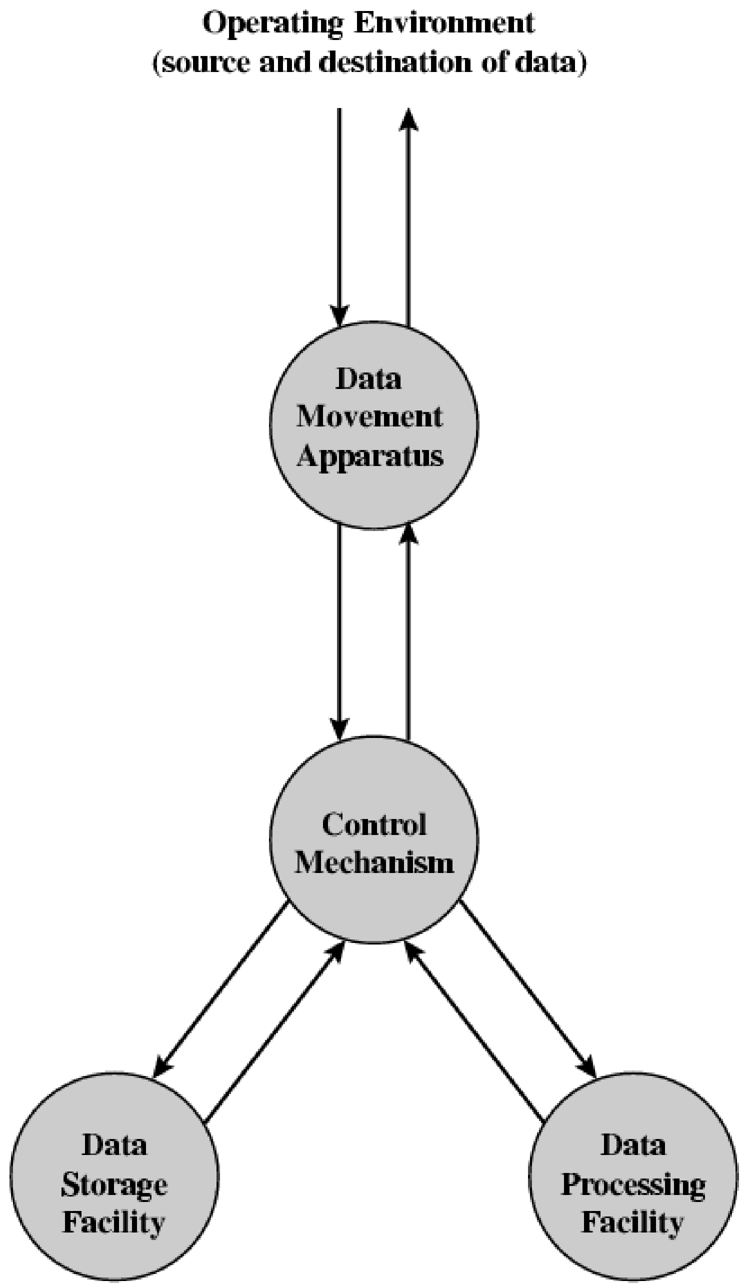
Structure - The Control Unit



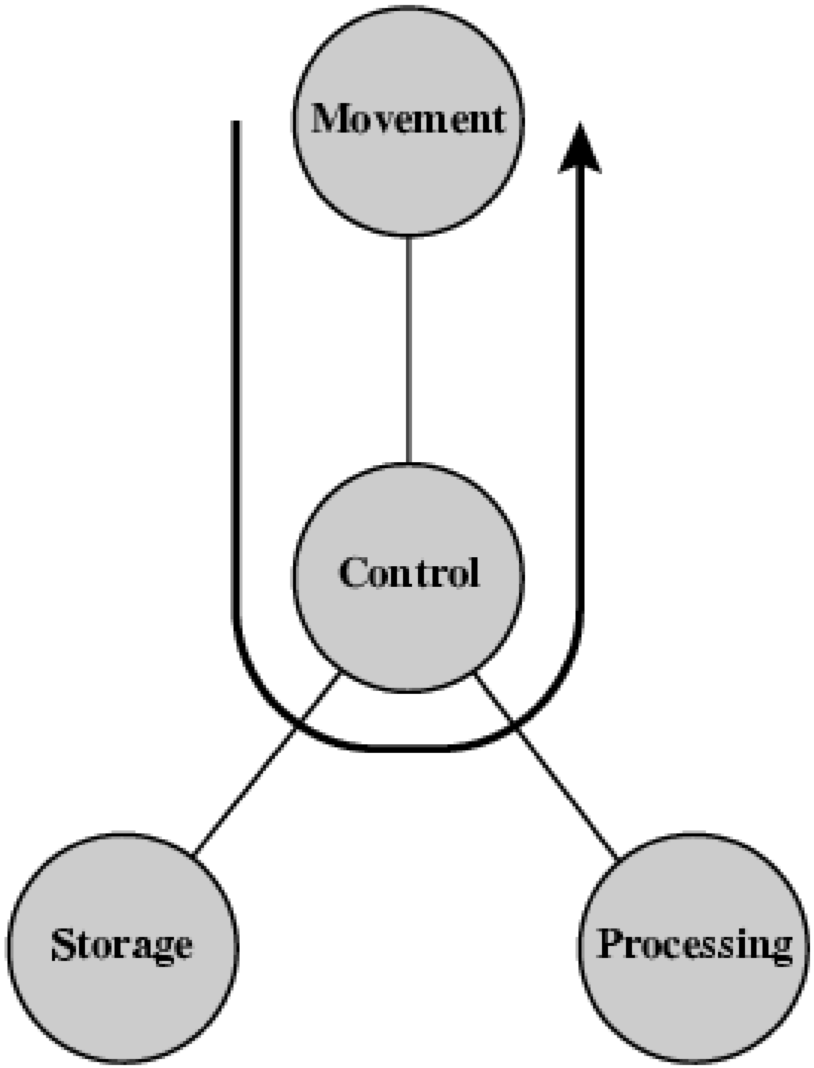
Computer Functions

- Data processing
- Data storage
- Data movement
- Control

Functional view

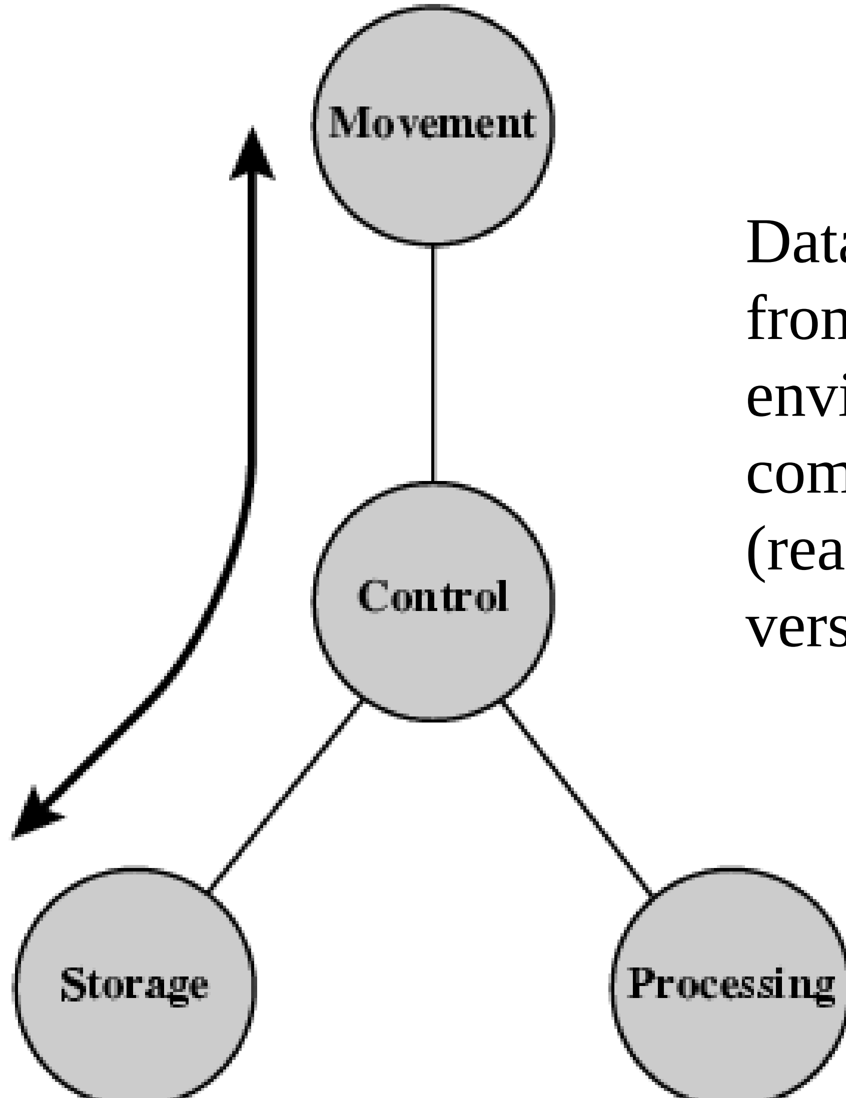


Operations (1) Data movement



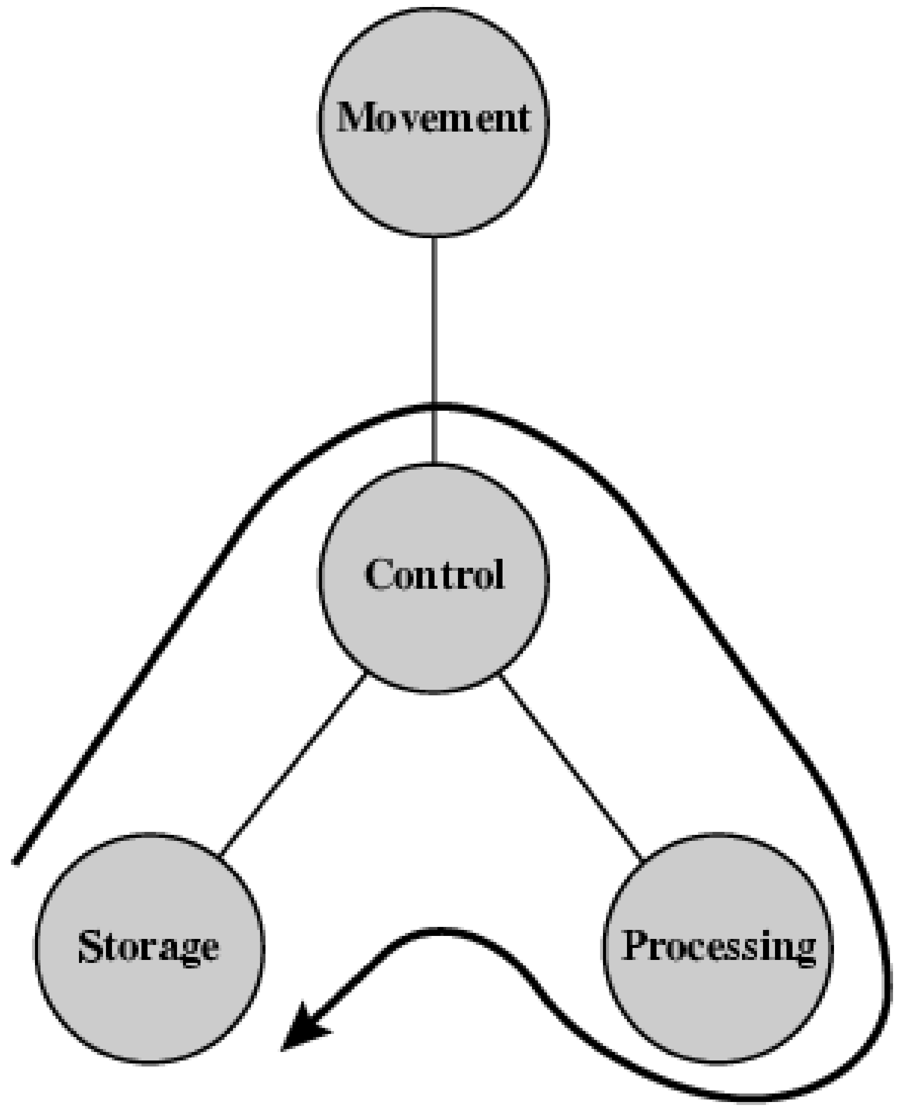
Simply transferring data from one peripheral or communications line to another.

Operations (2) Storage

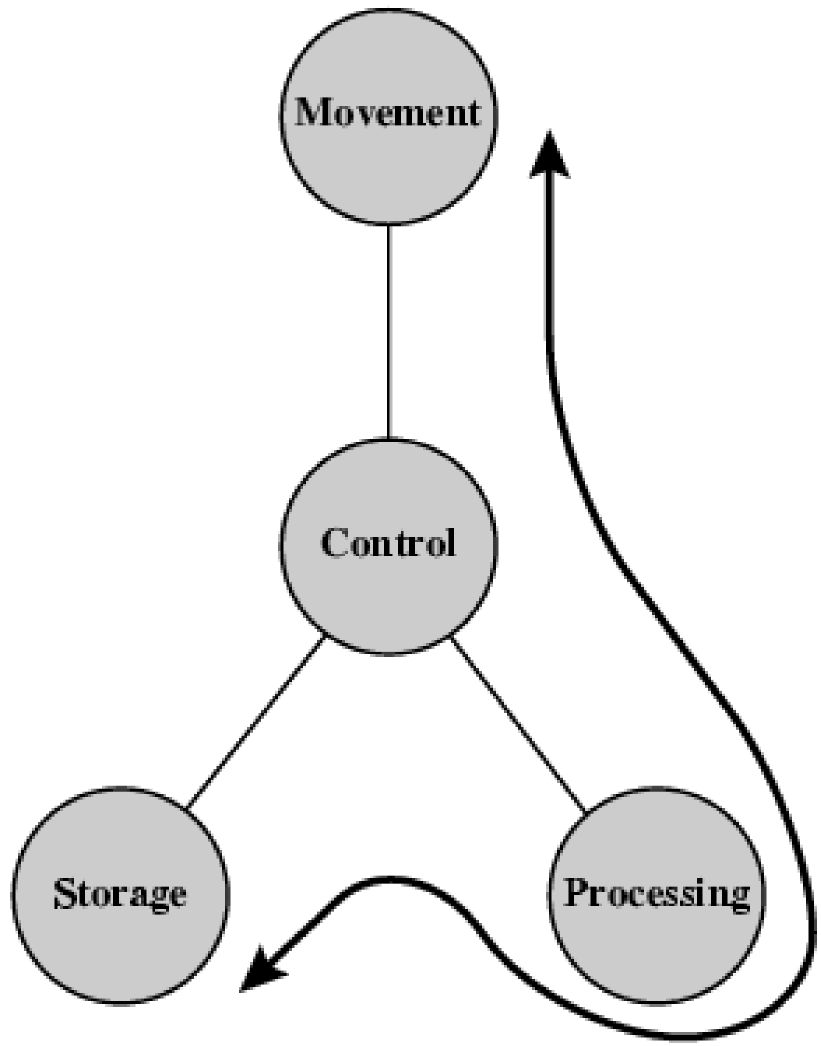


Data transferred from the external environment to computer storage (read) and vice versa (write).

Operation (3) Processing from/to storage



Operation (4)
Processing from storage to I/O

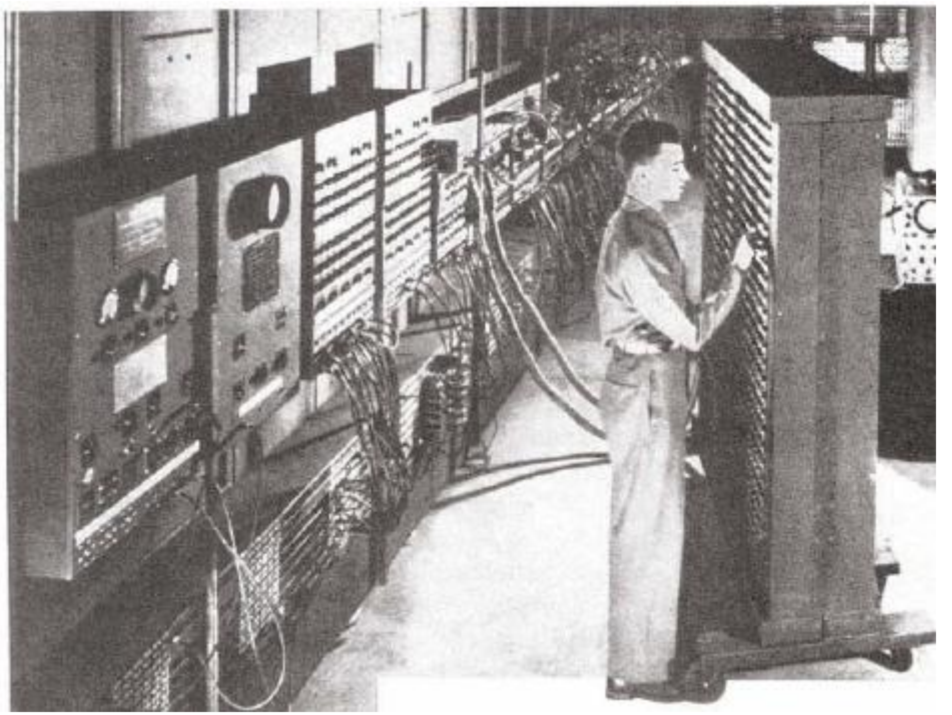


Basic Computer Model

- The von Neumann Model

Why The Von Neumann Model ?

- On the ENIAC, all programming was done at the digital logic level
- Programming the computer involved moving plugs and wires
- Configuring the ENIAC to solve a “simple” problem required many days labor by skilled technicians
- A different hardware configuration was needed to solve every unique problem type



The von Neumann Model

- The invention of stored program computers has been ascribed to a mathematician, John von Neumann

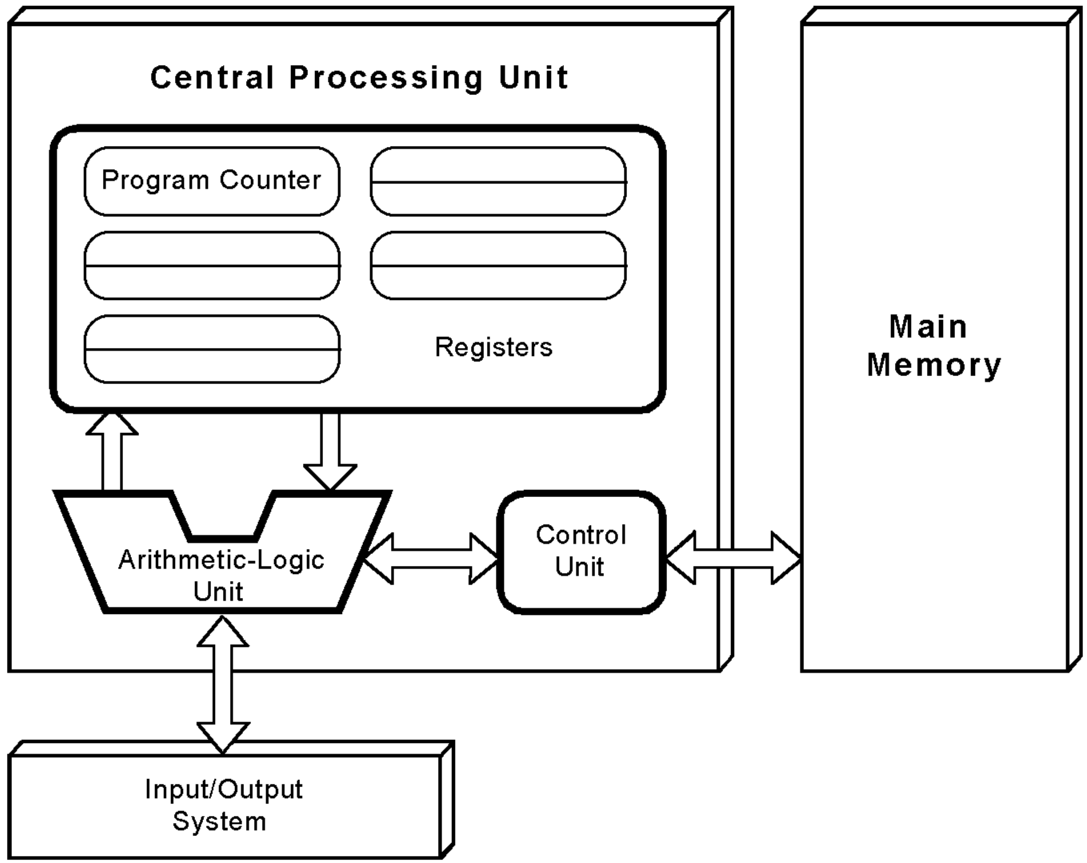
- Stored-program computers have become known as von Neumann Architecture systems

The von Neumann Model

Stored-program computers have the following characteristics:

- Three hardware systems:
  - A central processing unit (CPU)
  - A main memory system
  - An I/O system
- Provides the capacity to carry out sequential instruction processing

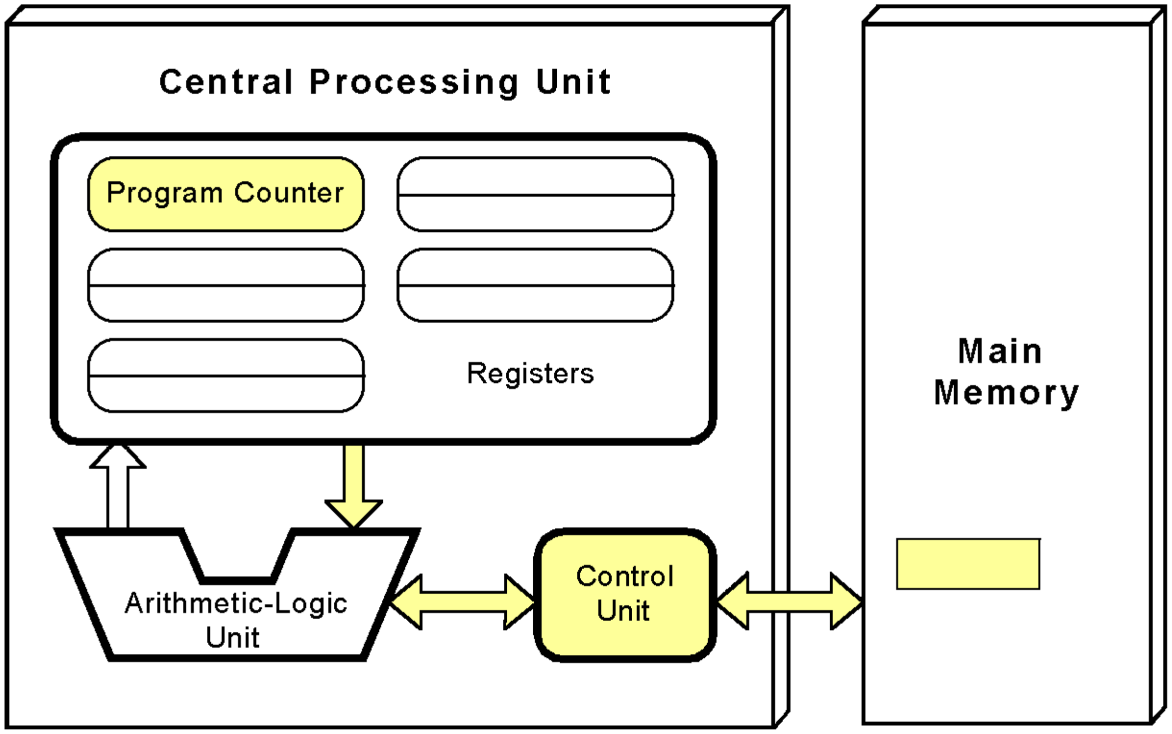
The von Neumann Model



Computers employ a <u>fetch-decode-execute</u> cycle to run programs

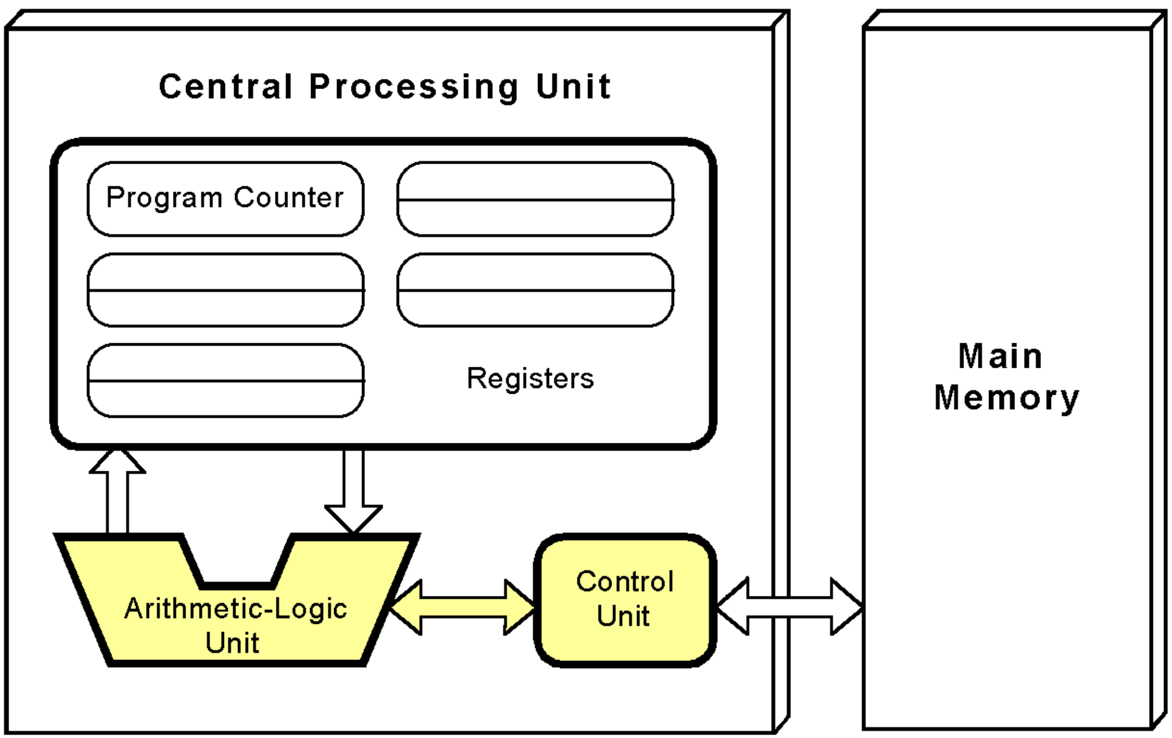
The von Neumann Model

- The control unit fetches the next instruction from memory using the program counter to determine where the instruction is located



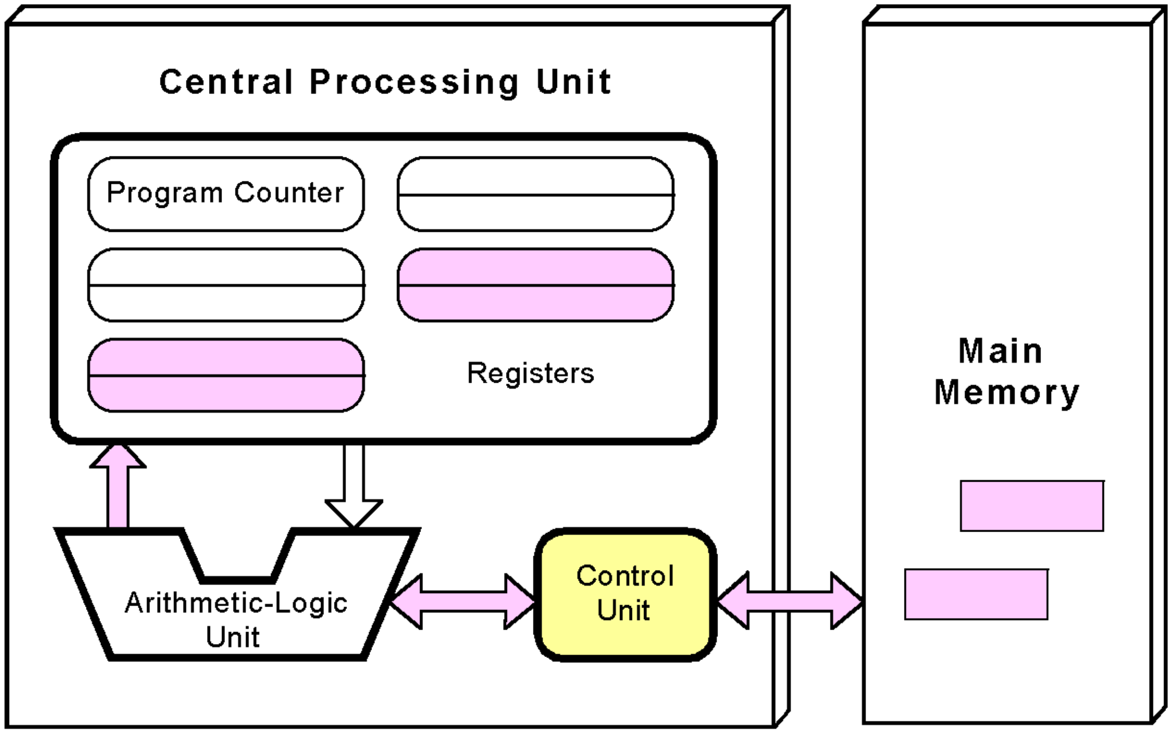
The von Neumann Model

- The instruction is decoded into a language that the ALU can understand



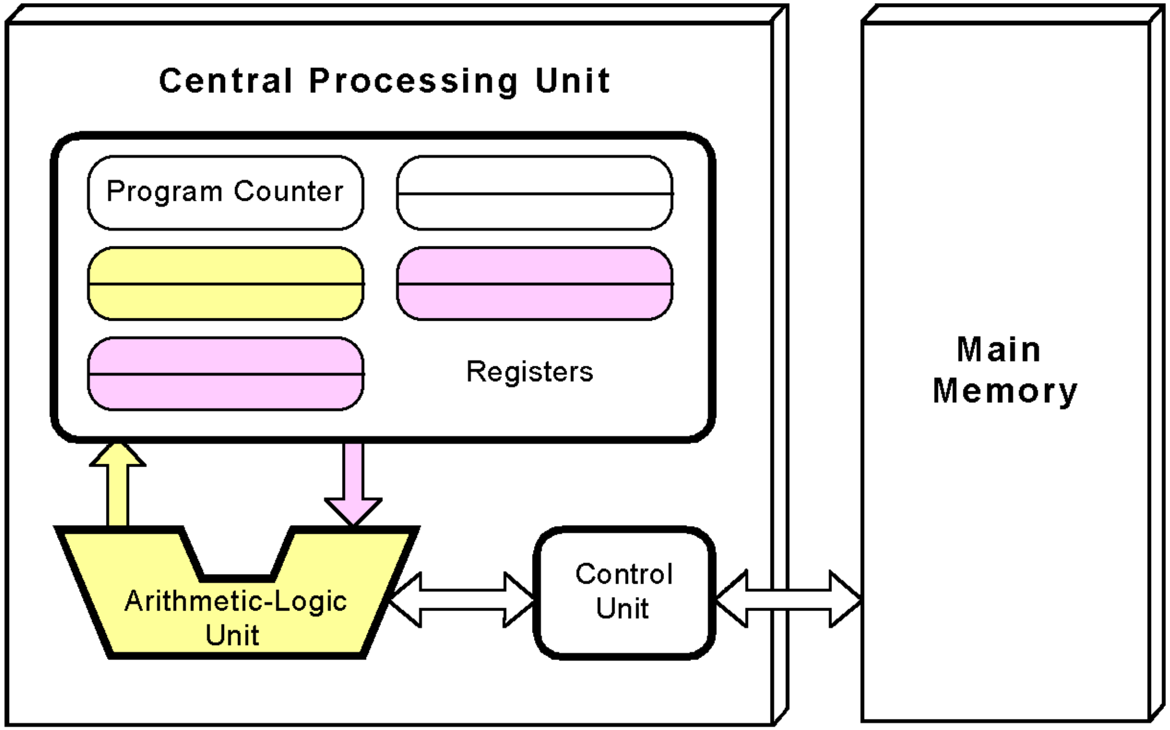
The von Neumann Model

- Any data operands required to execute the instruction are fetched from memory and placed into registers within the CPU



The von Neumann Model

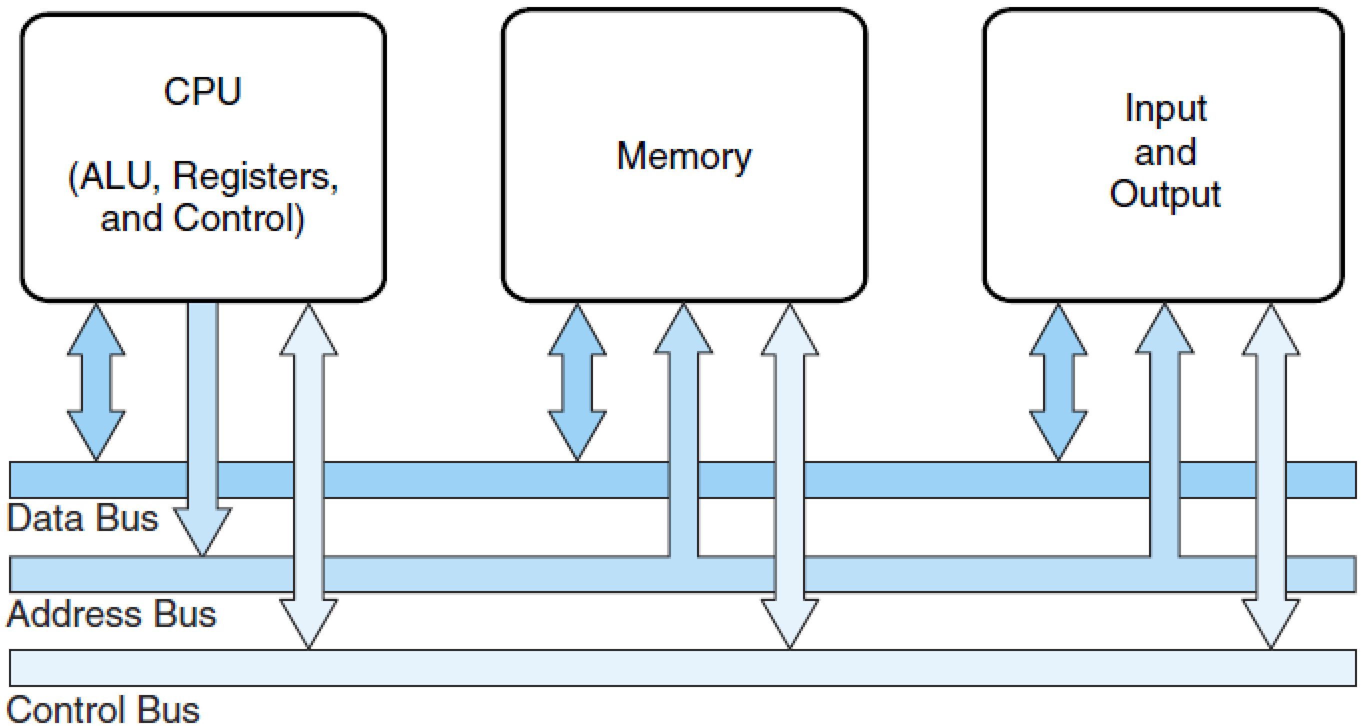
- The ALU executes the instruction and places results in registers or memory



The von Neumann Model

- Used the system bus
  - Data bus moves data from main memory to the CPU registers (and vice versa)
  - Address bus holds the address of the data that the data bus is currently accessing
  - Control bus carries the necessary control signals that specify how the information transfer is to take place

The von Neumann Model



The von Neumann Model

- A single data path between the CPU and main memory
  - This single path is known as the von Neumann bottleneck

Today's-von Neumann Models

- With many incremental improvements over the years:
  - Adding specialized buses
    - Programs and data stored in a slow-to-access storage medium, such as a HD, can be copied to a fast-access, volatile storage medium such as RAM prior to execution.
      - This architecture called the *system bus model*.
        - » The data bus moves data from main memory to the CPU registers (and vice versa).
        - » The address bus holds the address of the data that the data bus is currently accessing.
        - » The control bus carries the necessary control signals that specify how the information transfer is to take place.
  - Use of index registers addressing
  - Floating-point units
  - Cache memories
  - Use of interrupts
  - Asynchronous I/O
  - Concept of Virtual Memory
  - …

von Neumann Models

- Sufficient?
  - No, Computational power require departure from the classic von Neumann architecture

Non-von Neumann Models

- Adding processors is one approach
  - In the late 1960s, high-performance computer systems were equipped with dual processors to increase computational throughput
- In the 1970s supercomputer systems were introduced with 32 processors
- Supercomputers with 1,000 processors were built in the 1980s
- In 1999, IBM announced its Blue Gene system containing over 1 million processors

Non-von Neumann Models

- Parallel processing is only one method of providing increased computational power

- DNA computers, quantum computers and dataflow systems
  - May be the basis for the next generation of computers

Moore's Law

- Rapidly changing field:
  - Vacuum tube -> Transistor -> IC -> VLSI
  - Memory capacity
  - Processor speed (due to advances in technology <u>and</u> hardware organization)

Moore’s Law

- In 1965 by Gordon Moore, Intel founder

  “The density of transistors in an integrated circuit will double every year.”

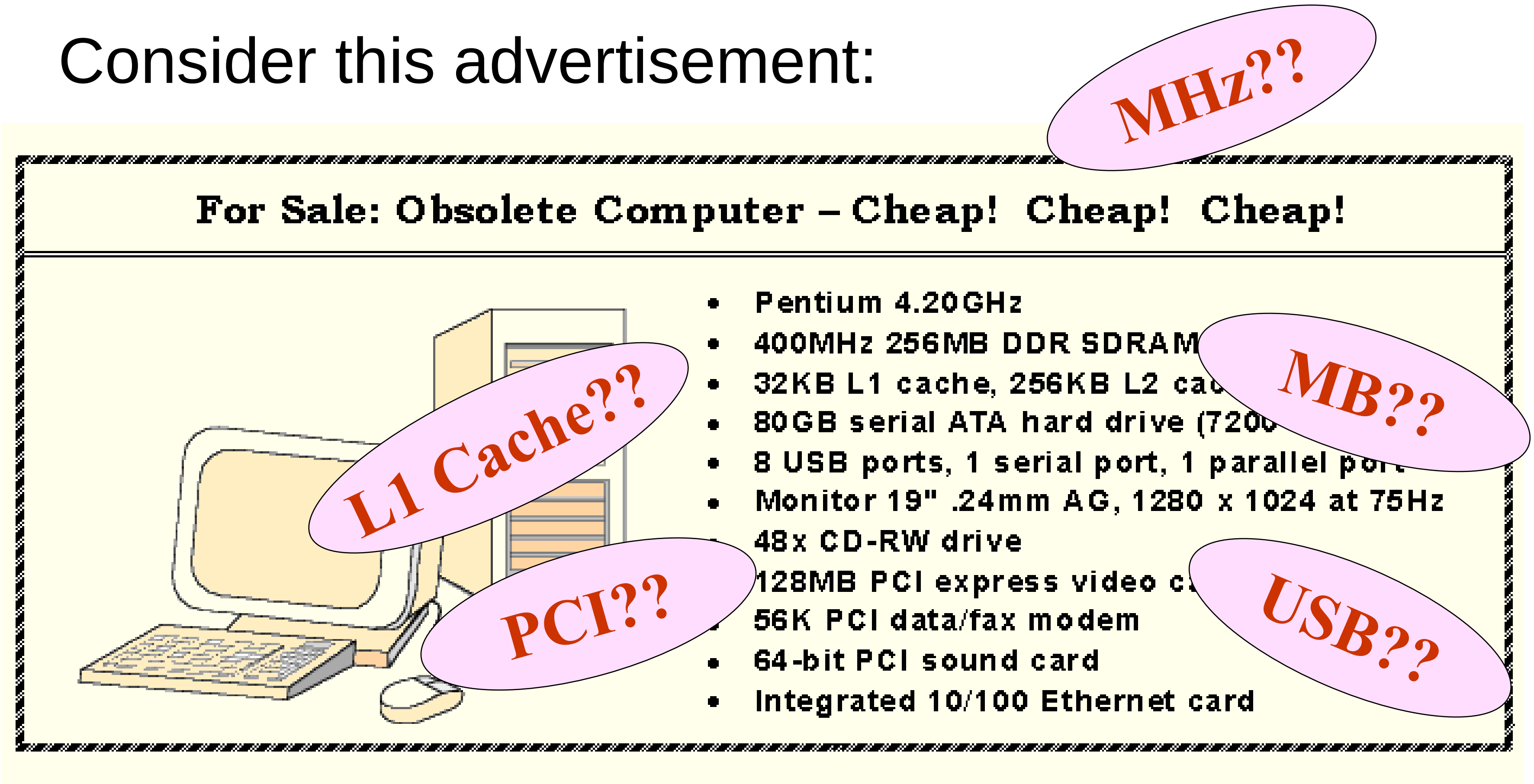
- Contemporary version:

  “The density of silicon chips doubles every 18 months.”

Terminology

An Example System

Consider this advertisement:



What does it all mean??

Summary

- Components of Computer
- The Computer Level Hierarchy
- Structure & Function of Computer
- Terminology

Next

[ISA]

Computer History

- Generation Zero:
  - Mechanical Calculating Machines (1642 - 1945)
  - Calculating Clock - Wilhelm Schickard (1592 - 1635).
  - Pascaline - Blaise Pascal (1623 - 1662).
  - Difference Engine - Charles Babbage (1791 - 1871), also designed but never built the Analytical Engine.
  - Punched card tabulating machines - Herman Hollerith (1860 - 1929).
- The First Generation:
  - Vacuum Tube Computers (1945 - 1953)
  - Electronic Numerical Integrator and Computer (ENIAC) by John Mauchly and J. Presper Eckertat the University of Pennsylvania, 1946
  - The IBM 650 first mass-produced computer. (1955).
- The Second Generation:
  - Transistorized Computers (1954 - 1965)
  - IBM 7094 (scientific) and 1401 (business)
  - Digital Equipment Corporation (DEC) PDP-1
  - Univac 1100
  - Control Data Corporation 1604.
  - . . . and many others.

Computer History

- The Third Generation:
  - Integrated Circuit Computers (1965 - 1980)
  - IBM 360
  - DEC PDP-8 and PDP-11
  - Cray-1 supercomputer
  - . . . and many others.
  - By this time, IBM had gained overwhelming dominance in the industry.
    - Computer manufacturers of this era were characterized as IBM and the BUNCH (Burroughs, Unisys, NCR, Control Data, and Honeywell).
- The Fourth Generation:
  - VLSI Computers (1980 - ????)
  - Very large scale integrated circuits (VLSI) have more than 10,000 components per chip.
  - Enabled the creation of microprocessors.
  - The first was the 4-bit Intel 4004.
  - Later versions, such as the 8080, 8086, and 8088 spawned the idea of “personal computing.”

An Example System

Measures of capacity and speed:

- Hertz = clock cycles per second (frequency)
  - 1MHz = 1,000,000Hz
  - Processor speeds are measured in MHz or GHz.

- Byte = a unit of storage
  - 1KB = 2^{10} = 1024 Bytes
  - 1MB = 2^{20} = 1,048,576 Bytes
  - Main memory (RAM) is measured in MB
  - Disk storage is measured in GB for small systems, TB for large systems.

  - Kilo- (K) = 1 thousand = 10^3 and 2^{10}
  - Mega- (M) = 1 million = 10^6 and 2^{20}
  - Giga- (G) = 1 billion = 10^9 and 2^{30}
  - Tera- (T) = 1 trillion = 10^{12} and 2^{40}
  - Peta- (P) = 1 quadrillion = 10^{15} and 2^{50}
  - ?

An Example System

Measures of time and space:

- Millisecond = 1 thousandth of a second
  - Hard disk drive access times are often 10 to 20 milliseconds.
- Nanosecond = 1 billionth of a second
  - Main memory access times are often 50 to 70 nanoseconds.
- Micron (micrometer) = 1 millionth of a meter
  - Circuits on computer chips are measured in microns.

- Milli- (m) = 1 thousandth = 10^{-3}
- Micro- (μ) = 1 millionth = 10^{-6}
- Nano- (n) = 1 billionth = 10^{-9}
- Pico- (p) = 1 trillionth = 10^{-12}
- Femto- (f) = 1 quadrillionth = 10^{-15}

An Example System

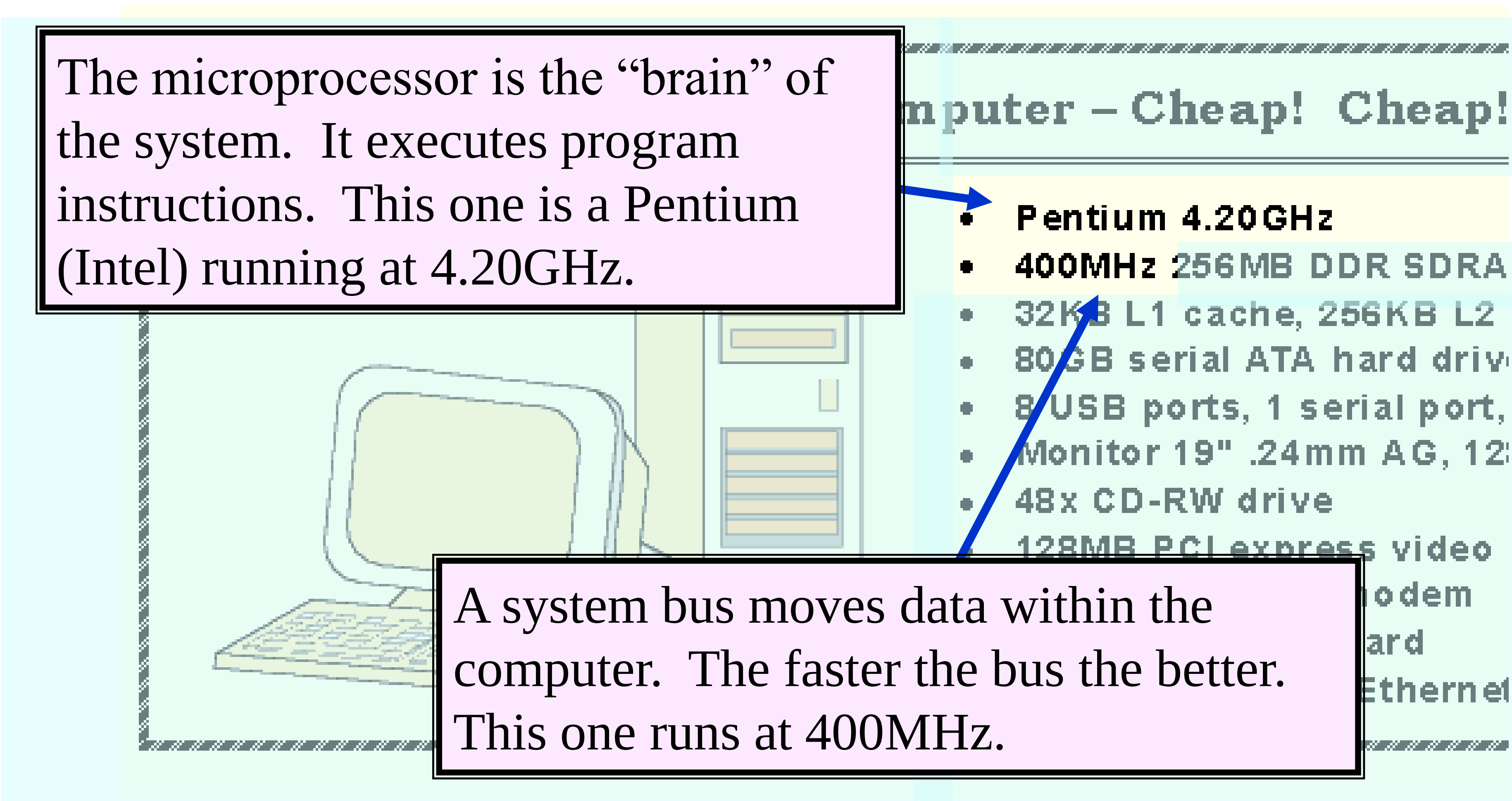
Cycle time is the reciprocal of clock frequency.

A bus operating at 133MHz has a cycle time of 7.52 nanoseconds:

$$133{,}000{,}000 \text{ cycles/second} = 7.52\text{ns/cycle}$$

Now back to the advertisement ...

An Example System



The microprocessor is the "brain" of the system. It executes program instructions. This one is a Pentium (Intel) running at 4.20GHz.

A system bus moves data within the computer. The faster the bus the better. This one runs at 400MHz.

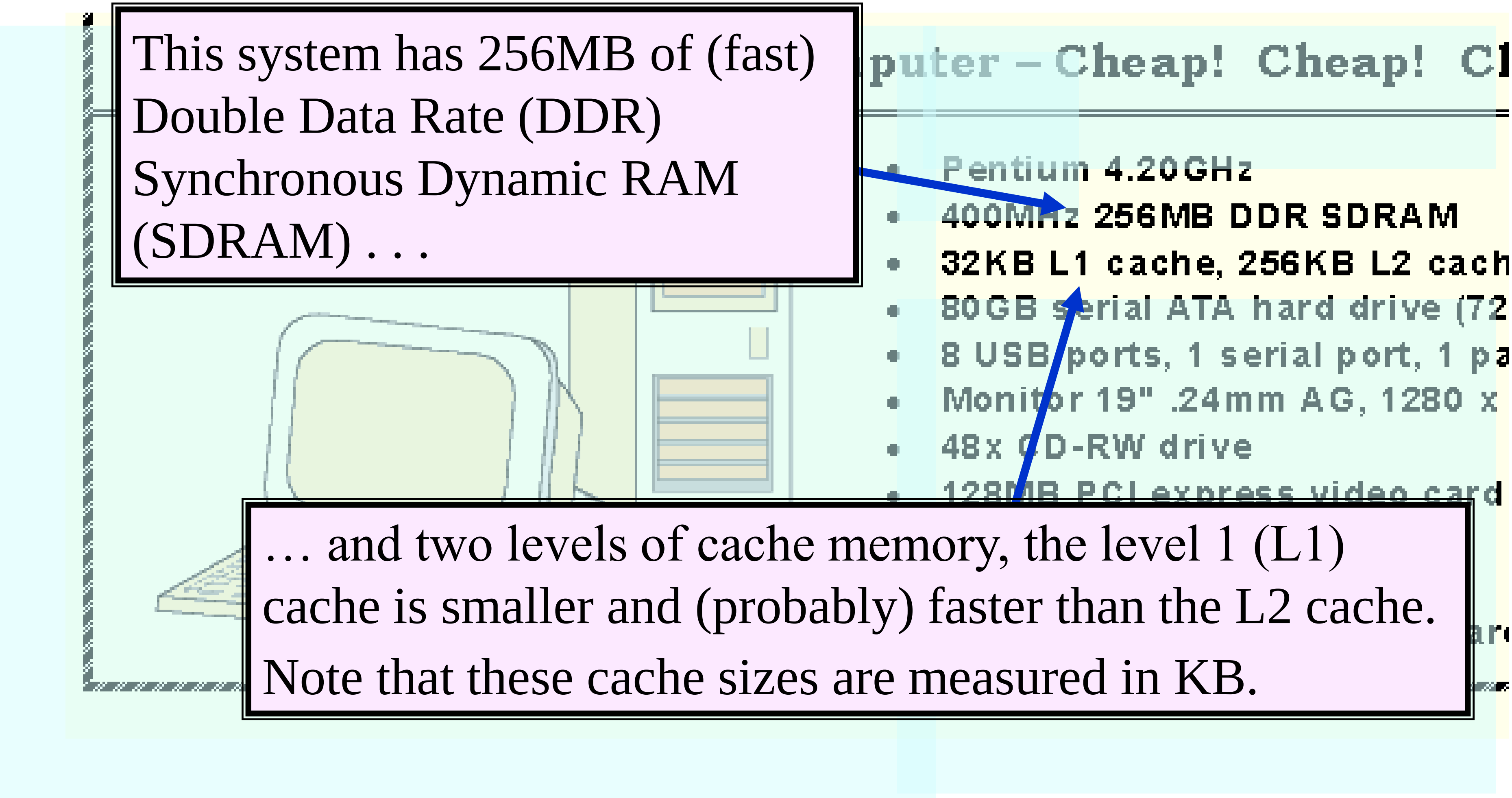
An Example System

- Large main memory capacity means you can run larger programs with greater speed than computers having small memories.

- RAM - Random Access Memory
  - Time to access contents is independent of its location.

- Cache is a type of temporary memory
  - that can be accessed faster than RAM.

An Example System

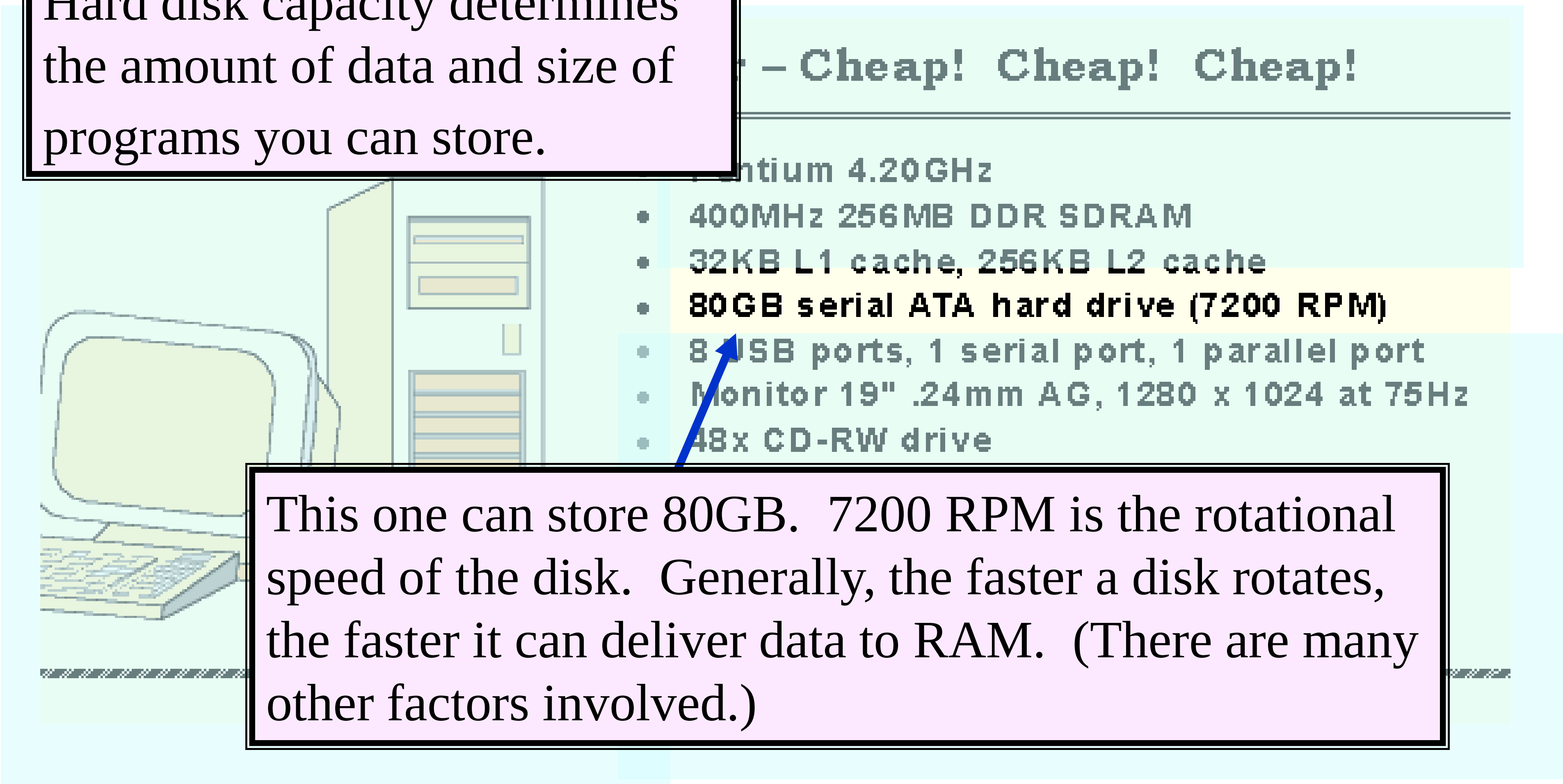


This system has 256MB of (fast) Double Data Rate (DDR) Synchronous Dynamic RAM (SDRAM) . . .

… and two levels of cache memory, the level 1 (L1) cache is smaller and (probably) faster than the L2 cache. Note that these cache sizes are measured in KB.

An Example System

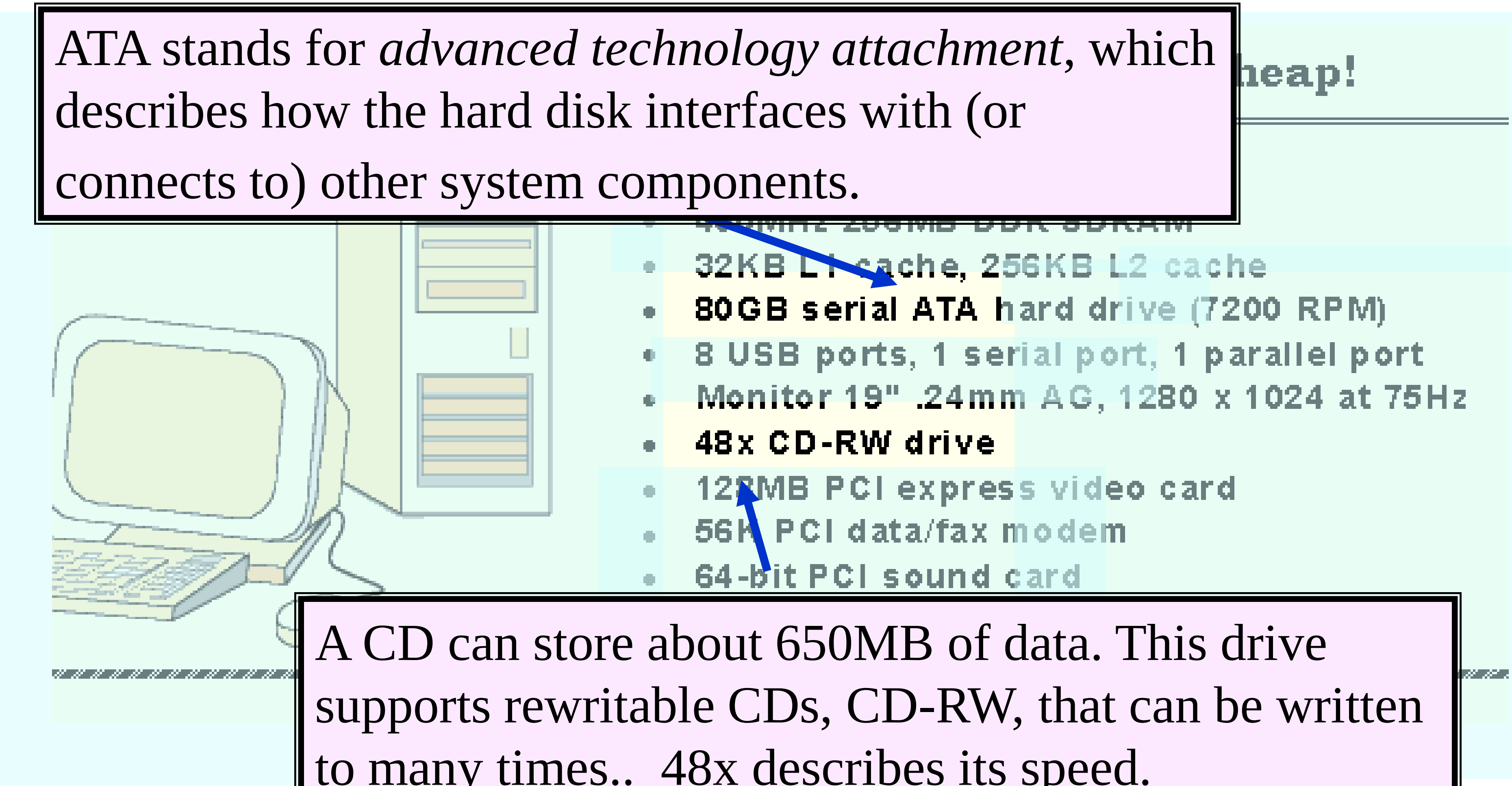
Hard disk capacity determines the amount of data and size of programs you can store.



This one can store 80GB. 7200 RPM is the rotational speed of the disk. Generally, the faster a disk rotates, the faster it can deliver data to RAM. (There are many other factors involved.)

An Example System

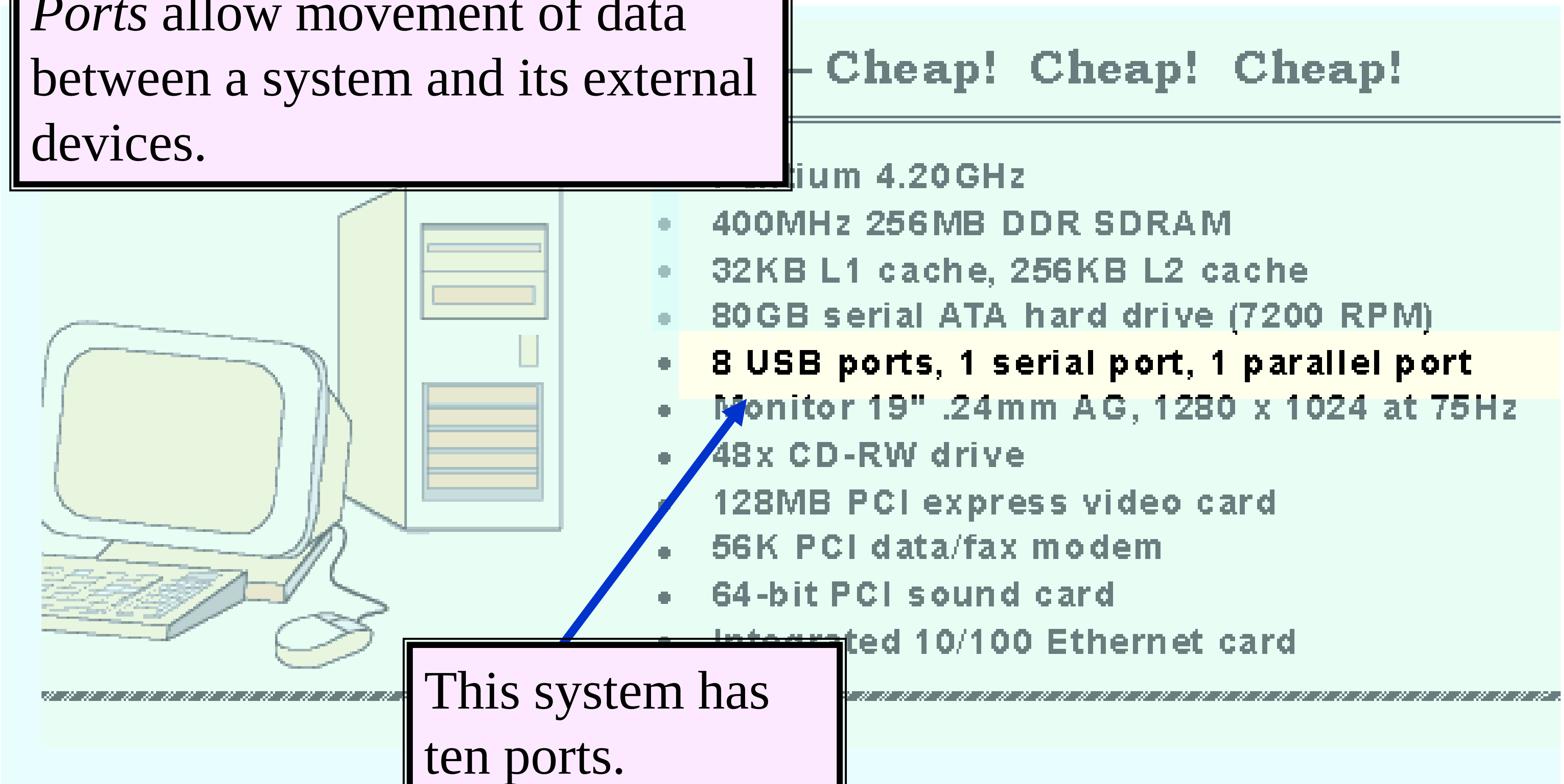
ATA stands for *advanced technology attachment*, which describes how the hard disk interfaces with (or connects to) other system components.



A CD can store about 650MB of data. This drive supports rewritable CDs, CD-RW, that can be written to many times.. 48x describes its speed.

An Example System

Ports allow movement of data between a system and its external devices.



This system has ten ports.

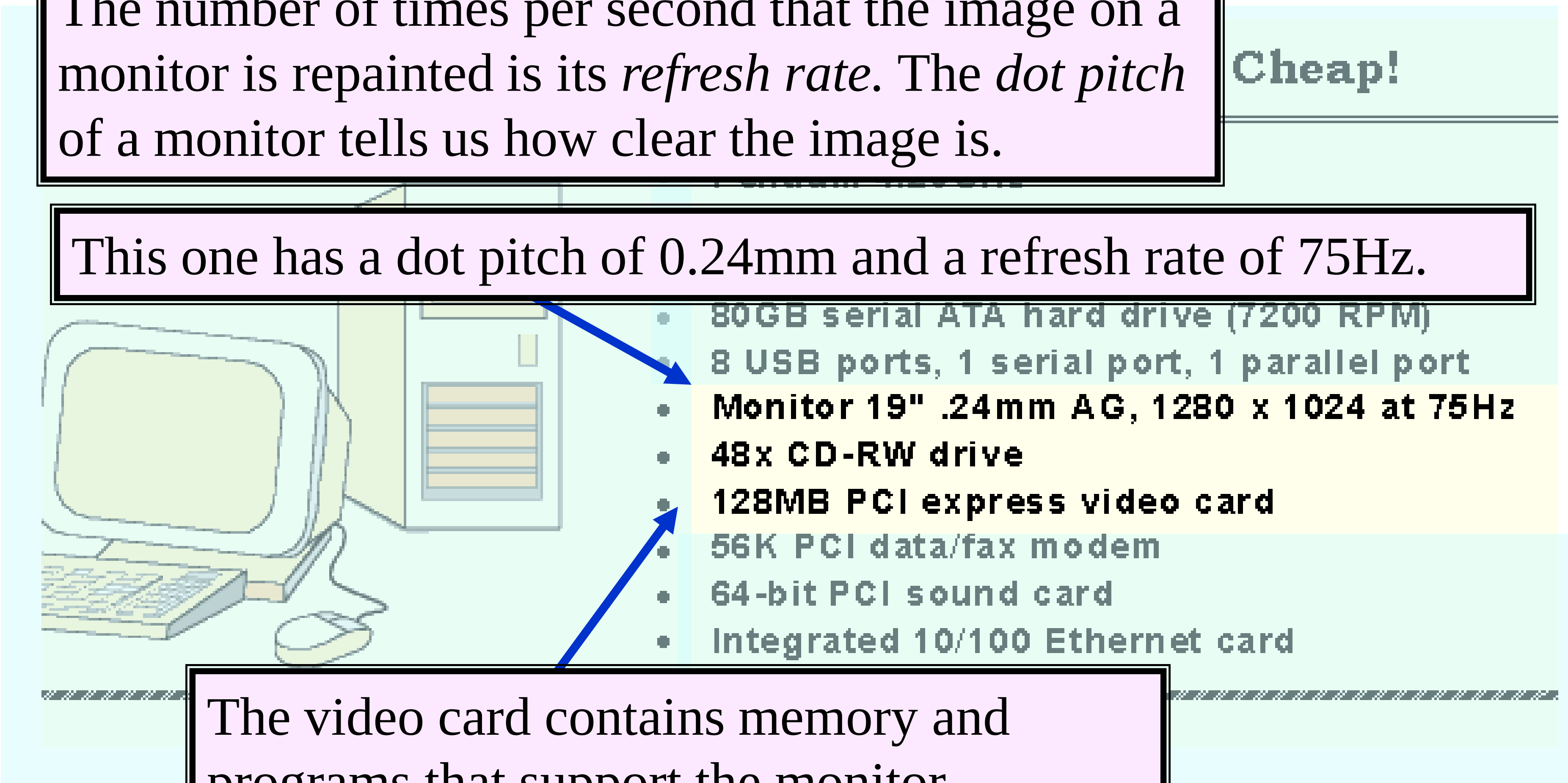
An Example System

- Serial ports send data as a series of pulses along one or two data lines.
- Parallel ports send data as a single pulse along at least eight data lines.
- USB, Universal Serial Bus, is an intelligent serial interface that is self-configuring. (It supports "plug and play.")

An Example System

The number of times per second that the image on a monitor is repainted is its *refresh rate*. The *dot pitch* of a monitor tells us how clear the image is.

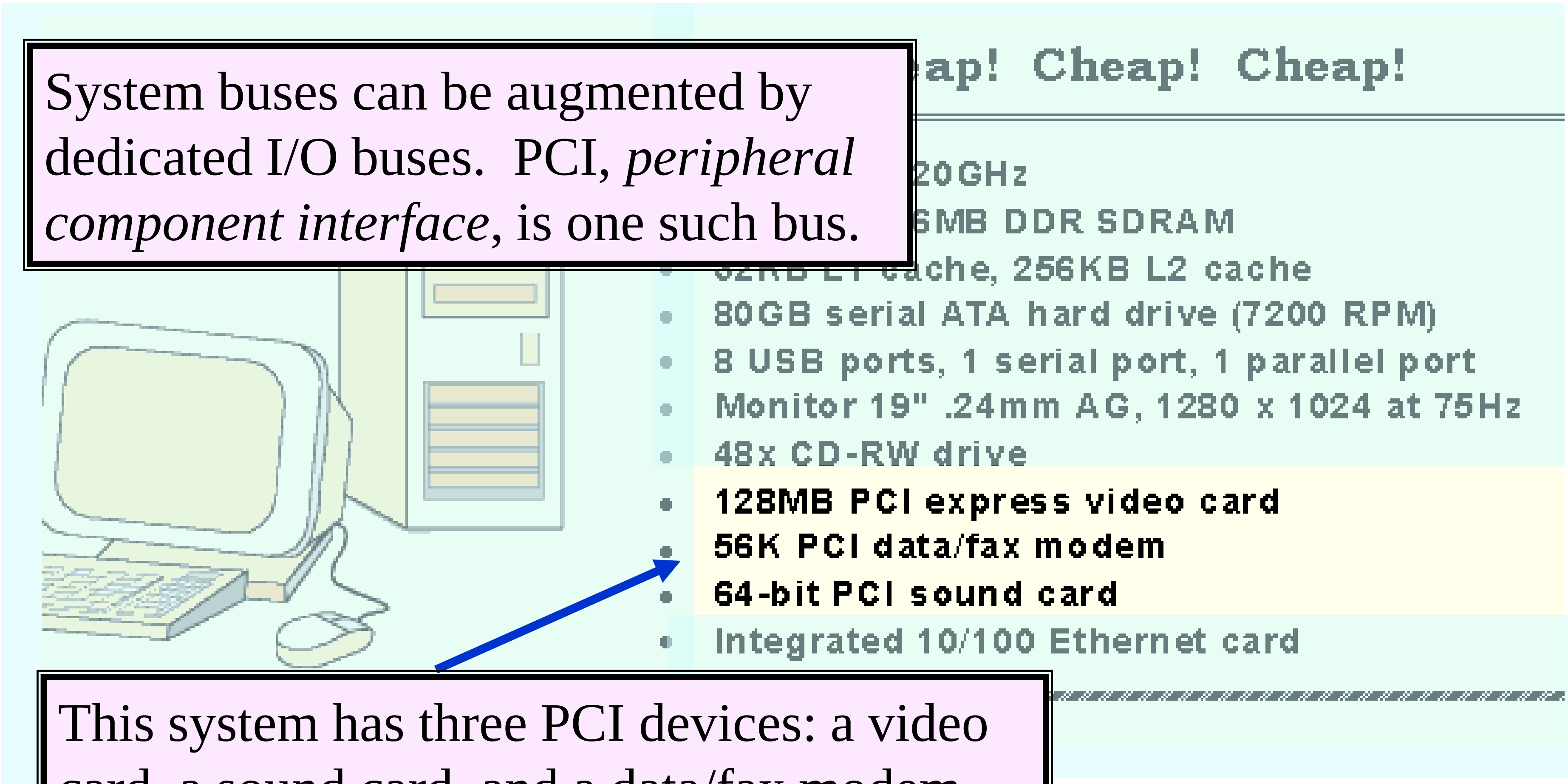
This one has a dot pitch of 0.24mm and a refresh rate of 75Hz.



The video card contains memory and programs that support the monitor.

An Example System

System buses can be augmented by dedicated I/O buses. PCI, *peripheral component interface*, is one such bus.



This system has three PCI devices: a video card, a sound card, and a data/fax modem.