Overview

Hamza Osman İLHAN

hoilhan@yildiz.edu.tr

YTU-CE / D037

Objectives

- Know the difference between computer organization and computer architecture.
- Understand units of measure common to computer systems.
- Appreciate the evolution of computers.
- Understand the computer as a layered system.
- Be able to explain the **von Neumann architecture** and the function of basic computer components.

Overview

Why study computer organization and architecture?

- Design better programs, including system software such as compilers, operating systems, and device drivers.
- Optimize program behavior.
- Evaluate (benchmark) computer system performance.
- Understand time, space, and price tradeoffs.

Overview

- Computer organization
  - Encompasses all physical aspects of computer systems.
  - E.g., circuit design, control signals, memory types.
  - *How does a computer work?*
- Computer architecture
  - Logical aspects of system implementation as seen by the programmer.
  - E.g., instruction sets, instruction formats, data types, addressing modes.
  - *How do I design a computer?*

Computer Components

- There is no clear distinction between matters related to computer organization and matters relevant to computer architecture.
- Principle of Equivalence of Hardware and Software:
  - ***Anything that can be done with software can also be done with hardware, and anything that can be done with hardware can also be done with software.****

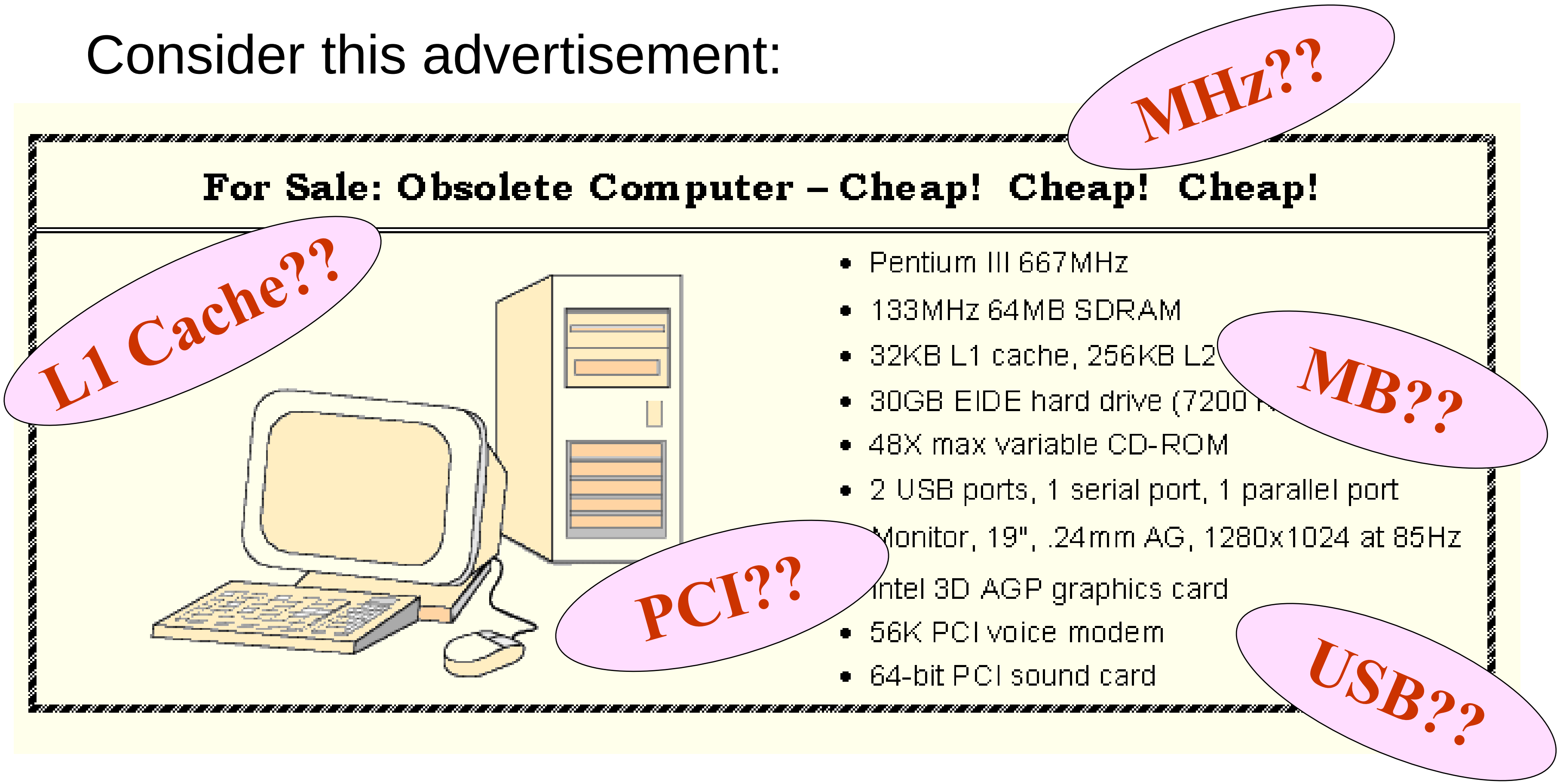
* Assuming speed is not a concern.

Computer Components

- At the most basic level, a computer is a device consisting of three pieces:
  - 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.

An example system

Consider this advertisement:



What does it all mean??

An example system

Measures of capacity and speed:

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

Whether a metric refers to a power of ten or a power of two typically depends upon what is being measured.

An example system

- 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.

An example system

Measures of time and space:

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

- 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.

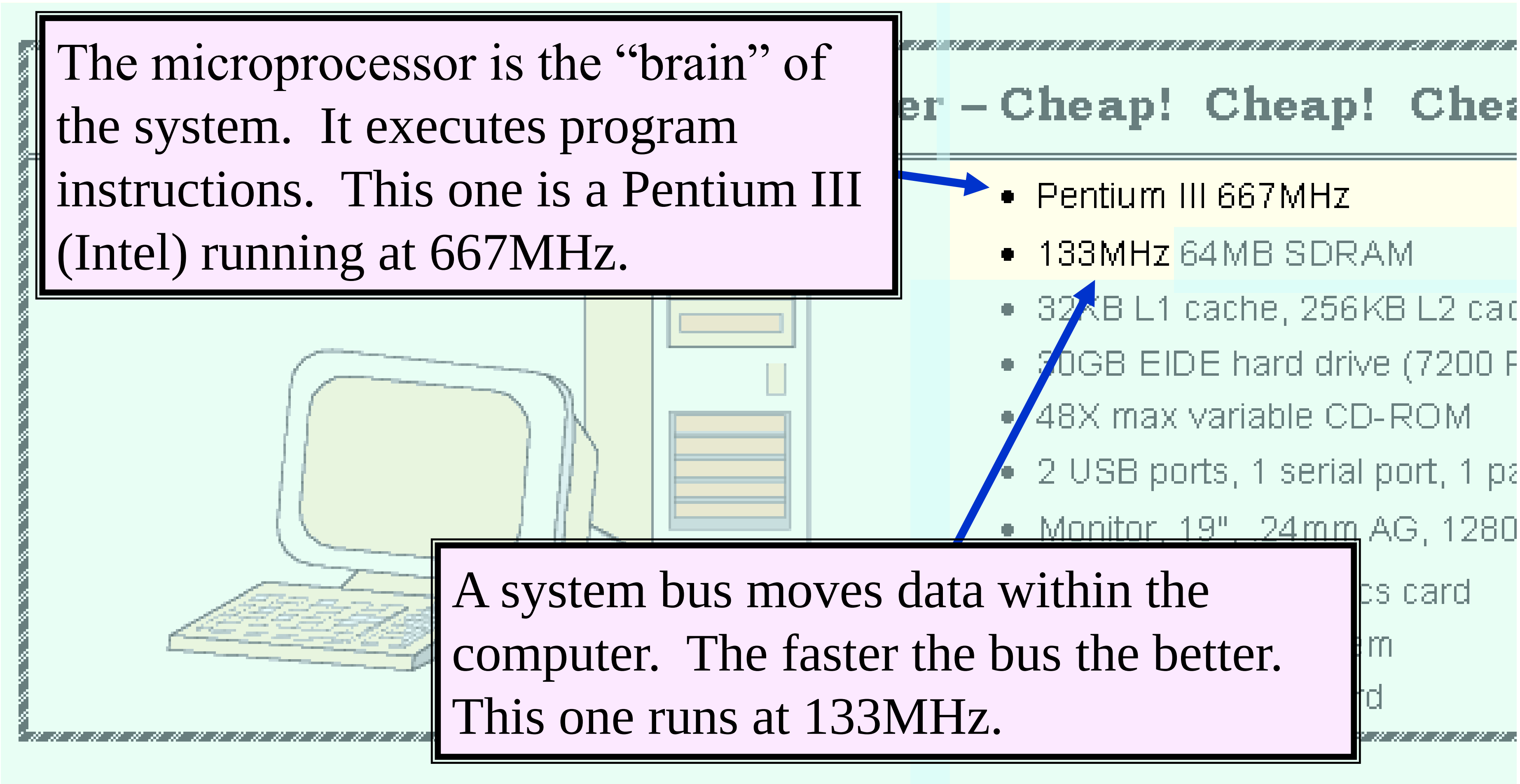
An example system

- We note that 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 III (Intel) running at 667MHz.

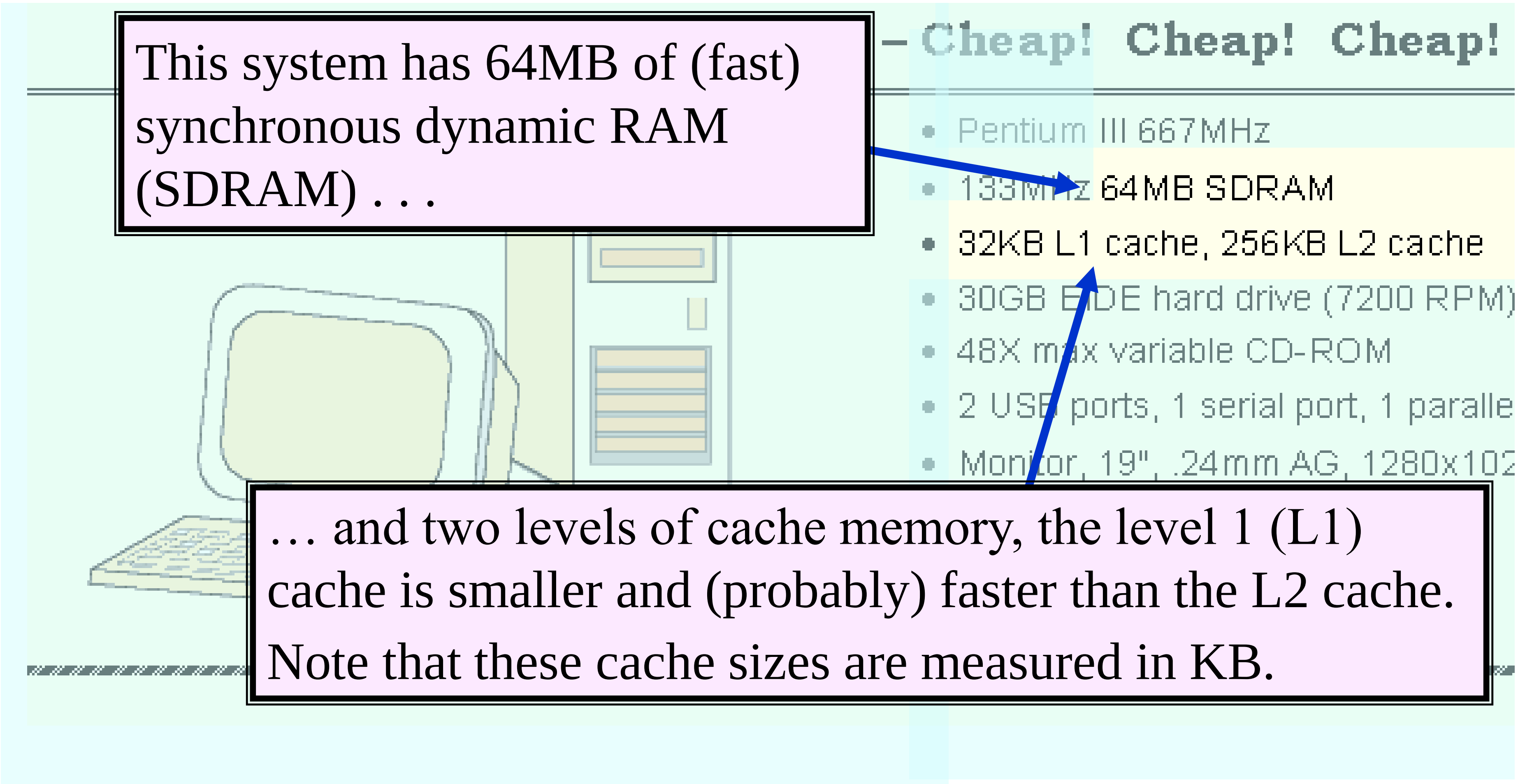
A system bus moves data within the computer. The faster the bus the better. This one runs at 133MHz.

An example system

- Computers with large main memory capacity can run larger programs with greater speed than computers having small memories.
- RAM is an acronym for *random access memory*. Random access means that memory contents can be accessed directly if you know its location.
- Cache is a type of temporary memory that can be accessed faster than RAM.

An example system

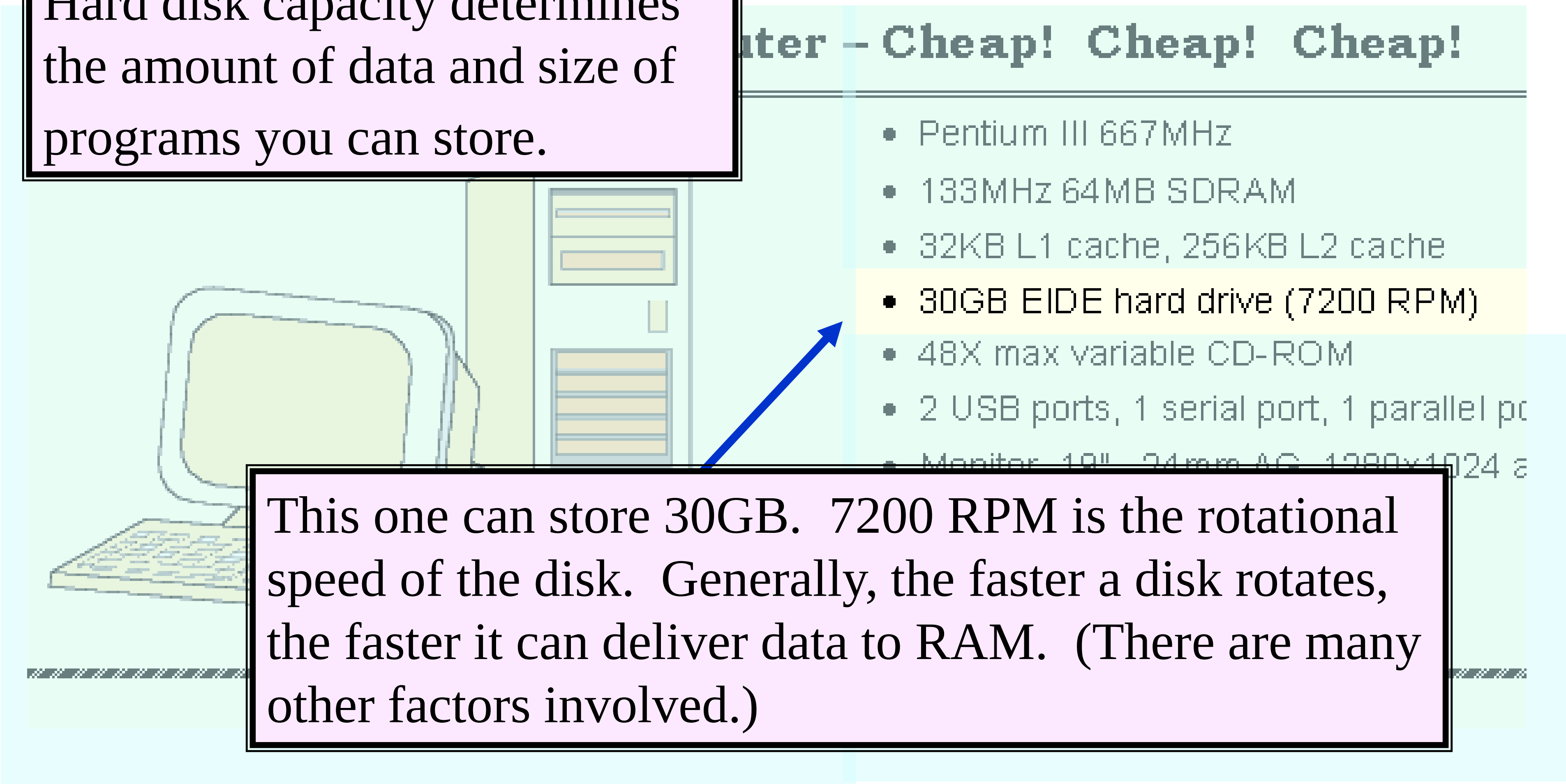
This system has 64MB of (fast) synchronous dynamic RAM (SDRAM) . . .



… 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

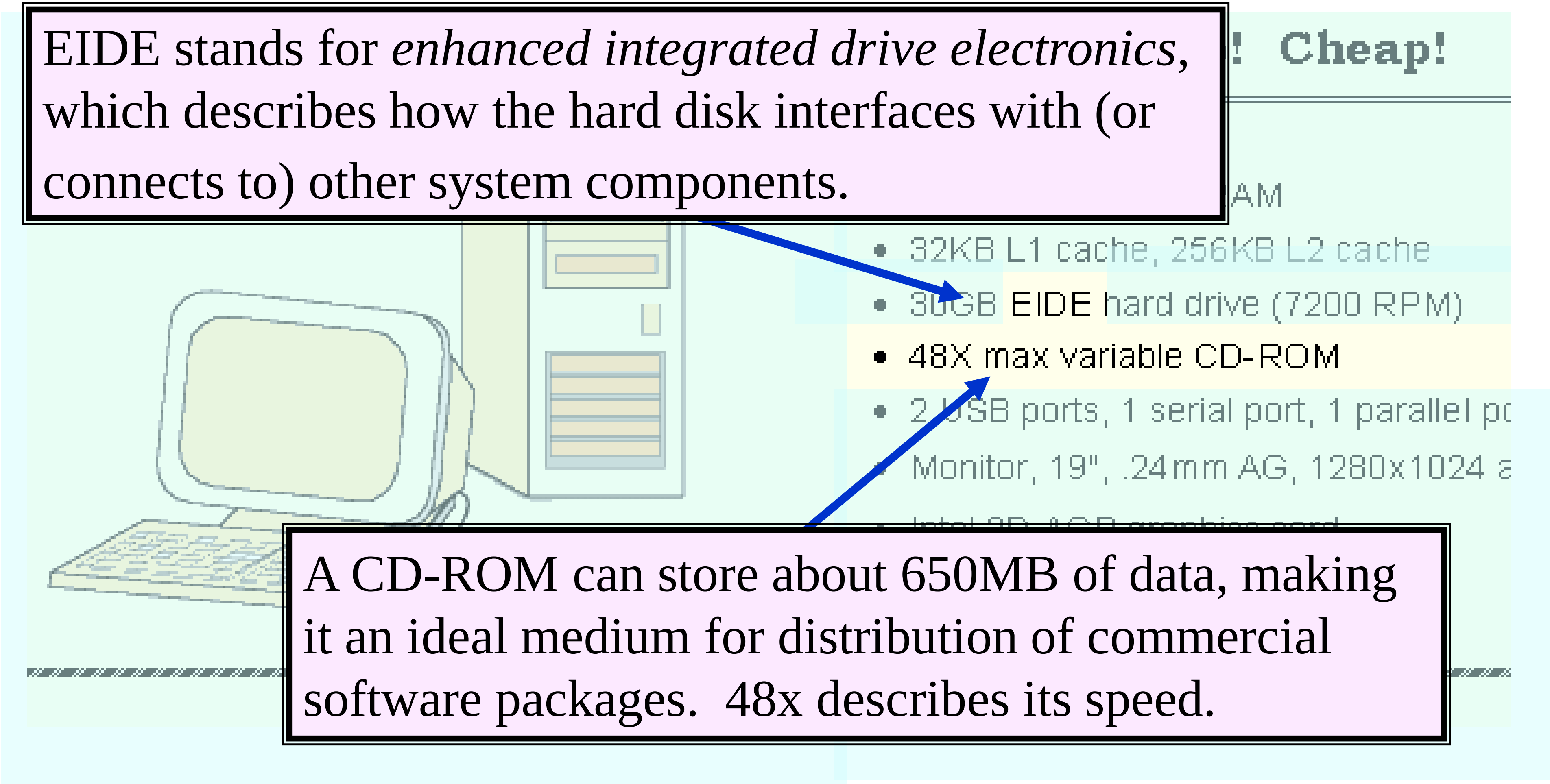
Hard disk capacity determines the amount of data and size of programs you can store.



This one can store 30GB. 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

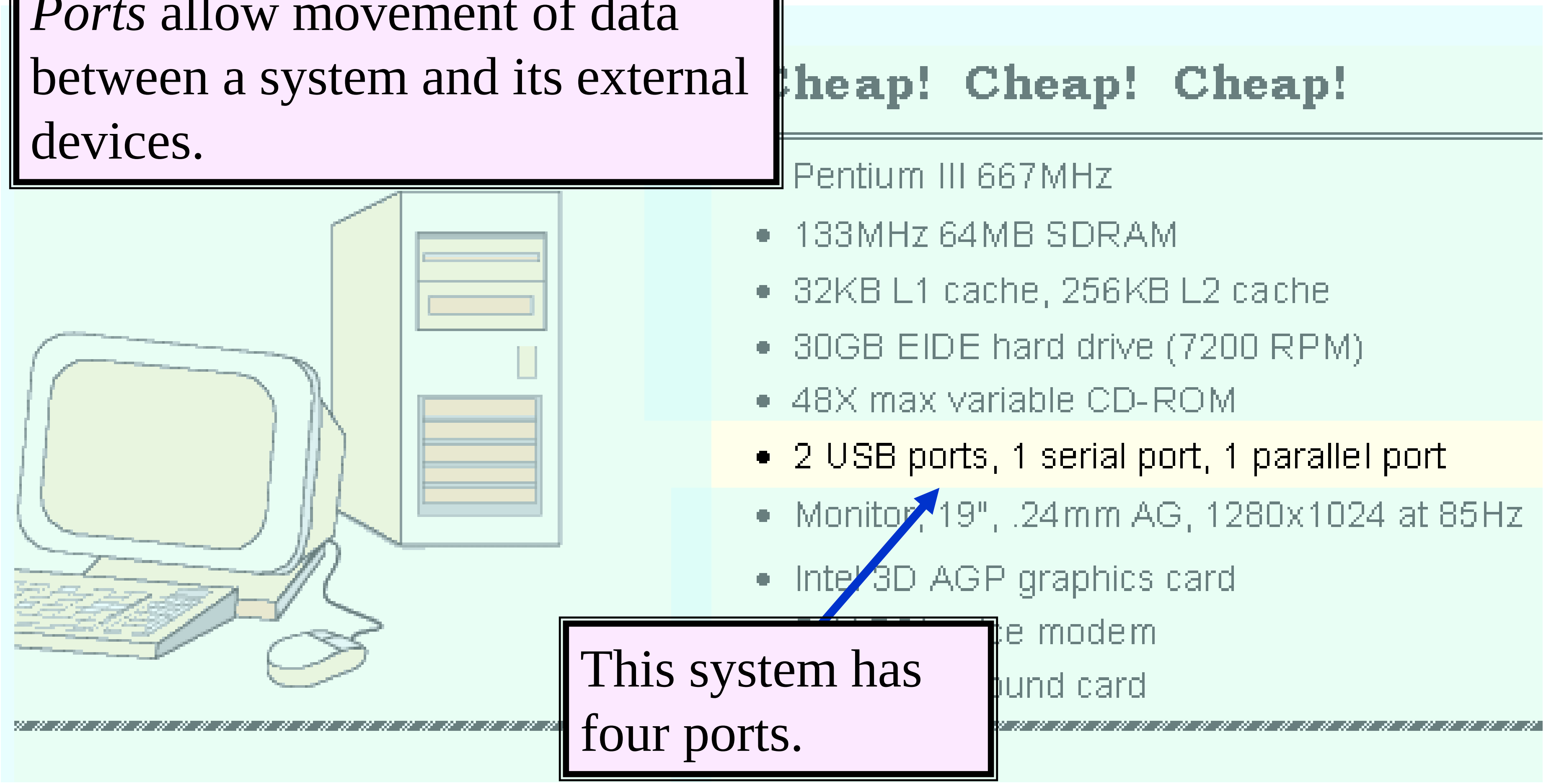
EIDE stands for *enhanced integrated drive electronics*, which describes how the hard disk interfaces with (or connects to) other system components.



A CD-ROM can store about 650MB of data, making it an ideal medium for distribution of commercial software packages. 48x describes its speed.

An example system

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



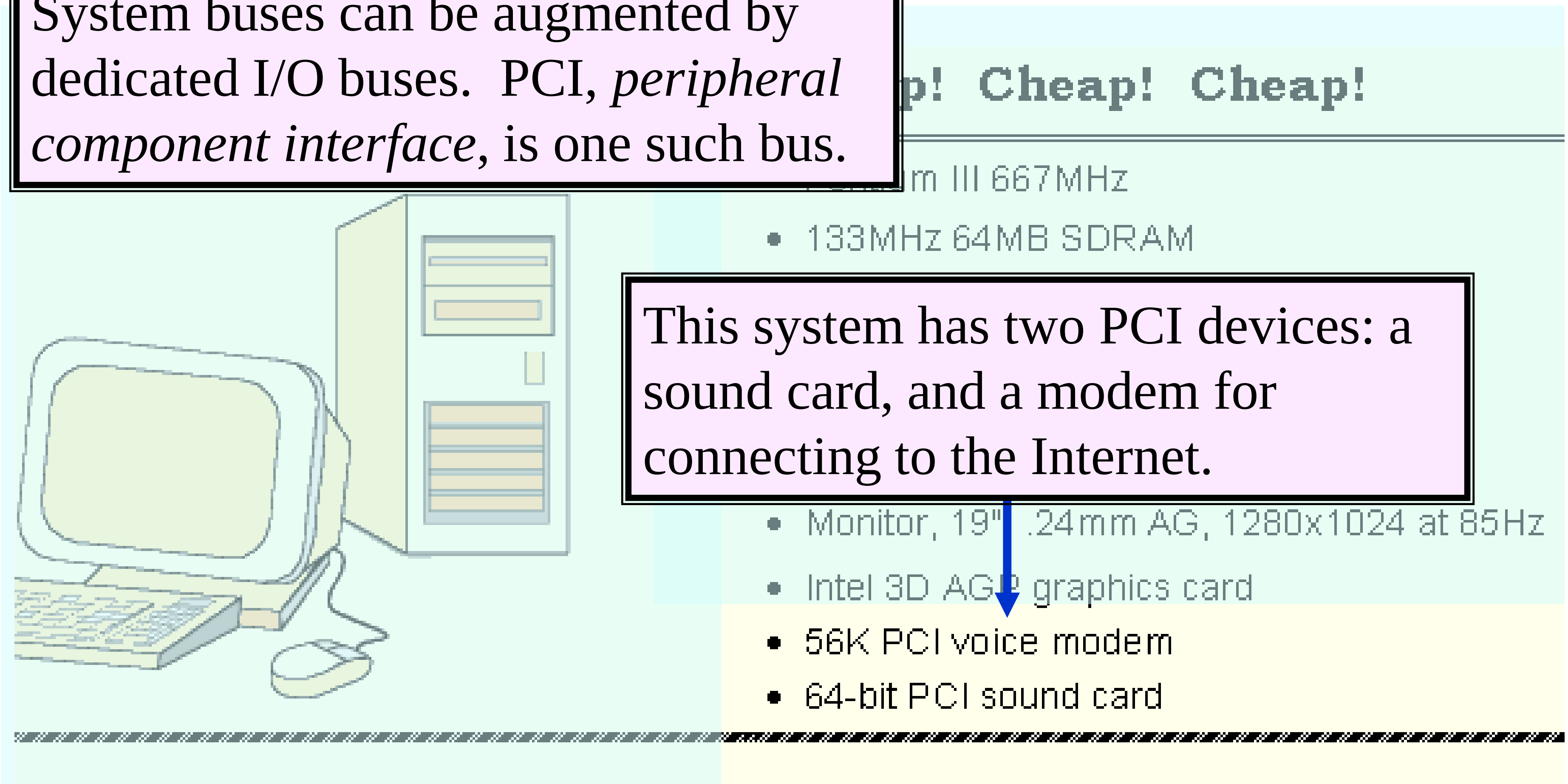
This system has four ports.

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

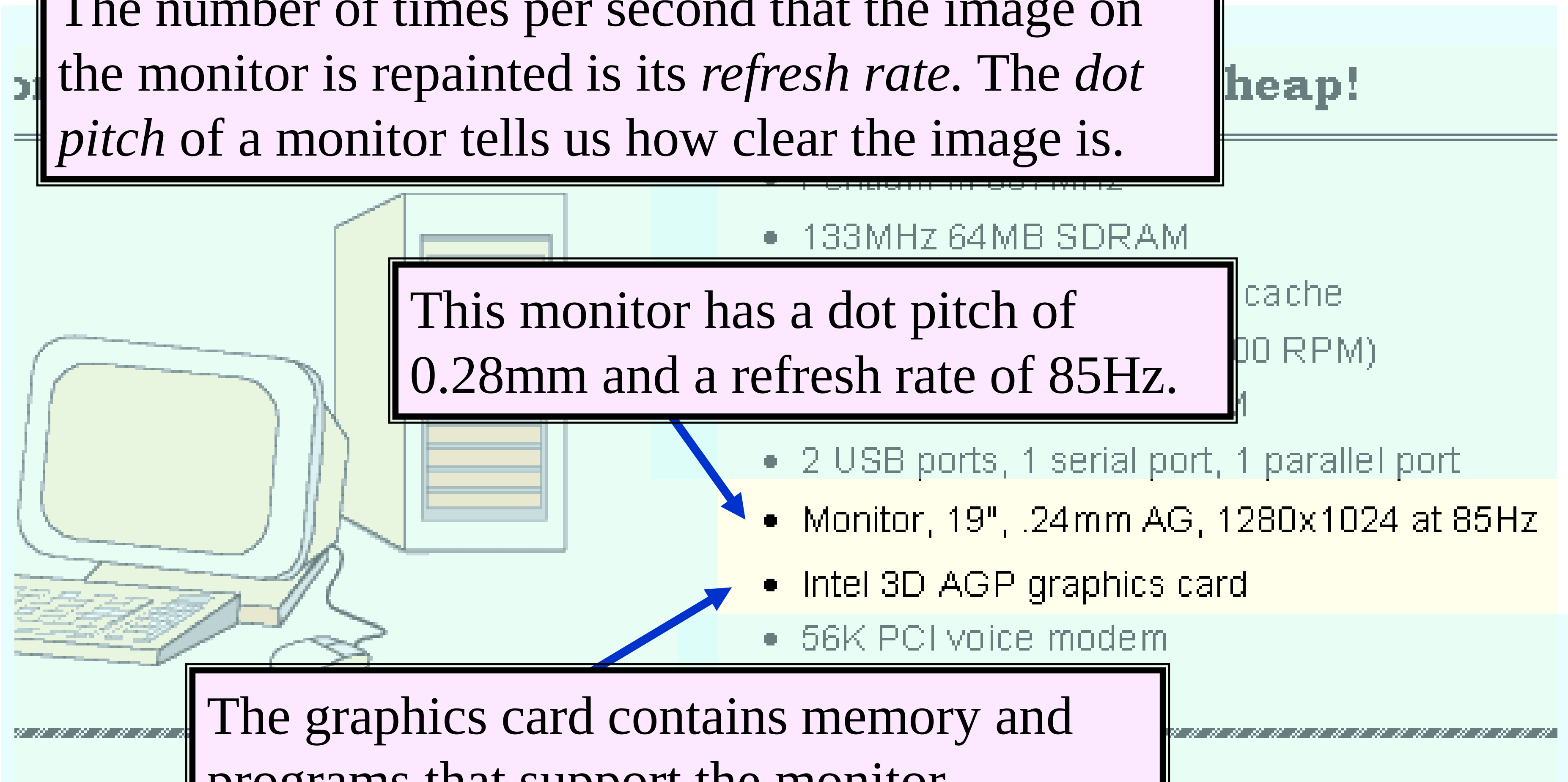
System buses can be augmented by dedicated I/O buses. PCI, *peripheral component interface*, is one such bus.



This system has two PCI devices: a sound card, and a modem for connecting to the Internet.

An example system

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



This monitor has a dot pitch of 0.28mm and a refresh rate of 85Hz.

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

An example system

Throughout the remainder of this book you will see how these components work and how they interact with software to make complete computer systems.

This statement raises two important questions:

What assurance do we have that computer components will operate as we expect?

And what assurance do we have that computer components will operate together?

Standards Organizations

- There are many organizations that set computer hardware standards-- to include the interoperability of computer components.
- Throughout this book, and in your career, you will encounter many of them.
- Some of the most important standards-setting groups are . . .

Standards Organizations

- The Institute of Electrical and Electronic Engineers (IEEE)
  - Promotes the interests of the worldwide electrical engineering community.
  - Establishes standards for computer components, data representation, and signaling protocols, among many other things.

Standards Organizations

- The International Telecommunications Union (ITU)
  - Concerns itself with the interoperability of telecommunications systems, including data communications and telephony.
- National groups establish standards within their respective countries:
  - The American National Standards Institute (ANSI)
  - The British Standards Institution (BSI)

Standards Organizations

- The International Organization for Standardization (ISO)
  - Establishes worldwide standards for everything from screw threads to photographic film.
  - Is influential in formulating standards for computer hardware and software, including their methods of manufacture.

Note: ISO is **not** an acronym. ISO comes from the Greek, *isos*, meaning “equal.”

Historical Development

- To fully appreciate the computers of today, it is helpful to understand how things got the way they are.
- The evolution of computing machinery has taken place over several centuries.
- In modern times computer evolution is usually classified into four generations according to the salient technology of the era.

We note that many of the following dates are approximate.

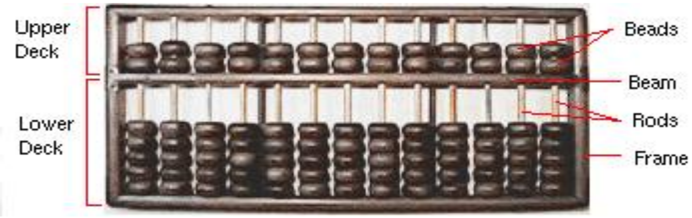
Historical Development

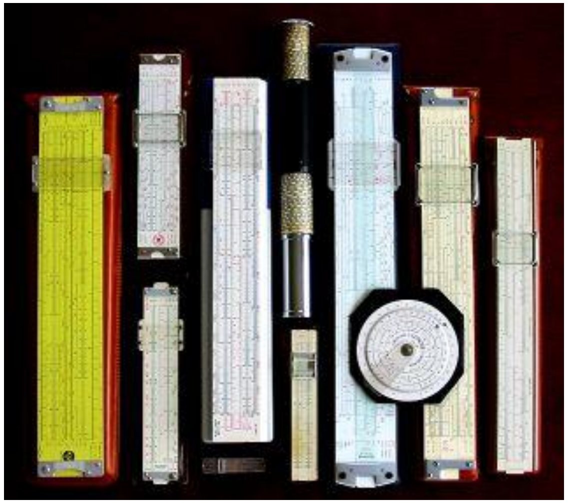
- Generation Zero: Mechanical Calculating Machines (1642 - 1945)
  - Calculating Clock - Wilhelm Schickard (1592 - 1635).
  - Pascaline - Blaise Pascal (1623 - 1662).
  - Difference Engine - Charles Babbage (1791 - 1871), also designed but never built the Analytical Engine.
  - Punched card tabulating machines - Herman Hollerith (1860 - 1929).

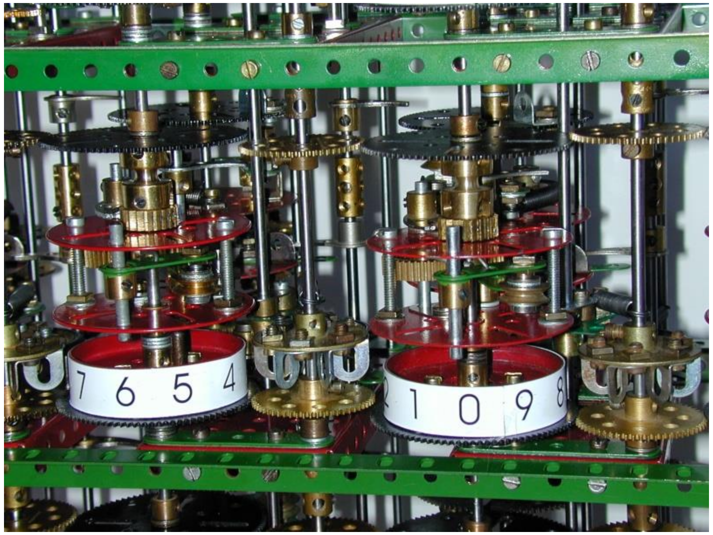
Hollerith cards were commonly used for computer input well into the 1970s.

Mechanical Brains

- Abacus
- Slide Rule
- Difference Engine
- Mechanical Calculators
- Differential Analyzer

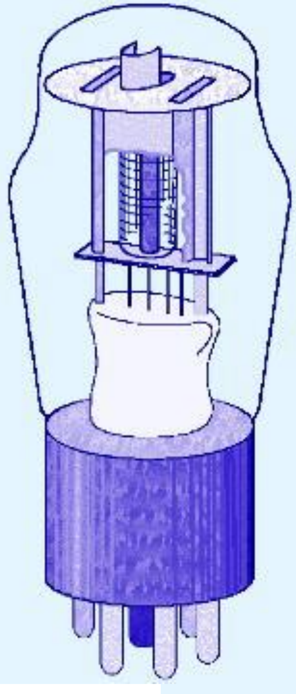






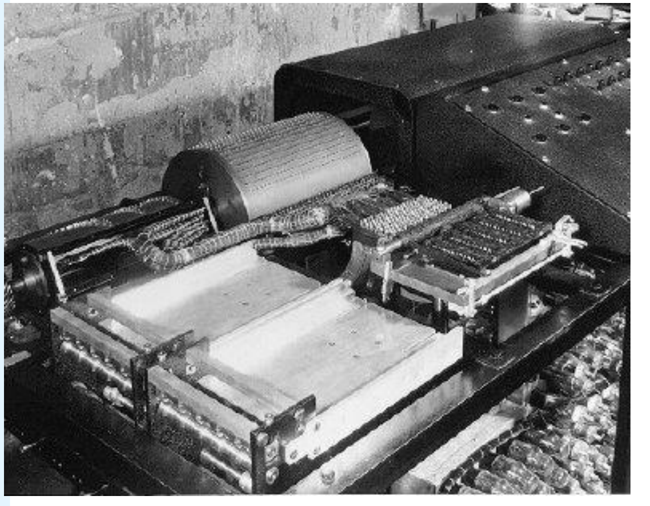
Historical Development

- The First Generation: Vacuum Tube Computers (1945 - 1953)



— Atanasoff Berry Computer (1937 - 1938) solved systems of linear equations.

– John Atanasoff and Clifford Berry of Iowa State University.



Historical Development

- The First Generation: Vacuum Tube Computers (1945 - 1953)
  - Electronic Numerical Integrator and Computer (ENIAC)
  - John Mauchly and J. Presper Eckert
  - University of Pennsylvania, 1946

The first *general-purpose* computer.

Historical Development

- The First Generation: Vacuum Tube Computers (1945 - 1953)

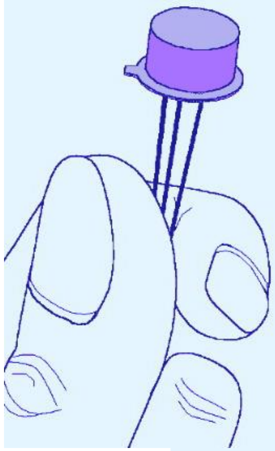
—IBM 650 (1955)

—Phased out in 1969.

The first *mass-produced* computer.

Historical Development

- The Second Generation: Transistorized Computers (1954 -
  - IBM 7094 (scientific) and 1401 (business)
  - Digital Equipment Corporation (DEC) PD 1
  - Univac 1100
  - . . . and many others.



DEC PDP-1

Historical Development

- The Third Generation: Integrated Circuit Computers (1965 - 1980)
  - —IBM 360
  - —DEC PDP-8 and PDP-11
  - —Cray-1 supercomputer
  - —. . . and many others.

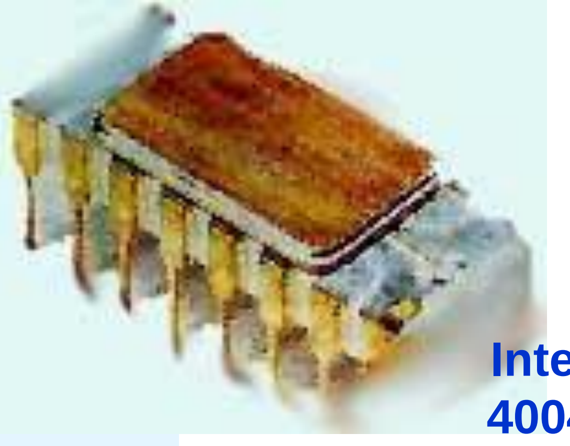
Historical Development

- 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 400



Intel 4004

Later versions, such as the 8080, 8086, and 8088 spawned the idea of "personal computing."

Historical Development

- Moore's Law (1965)
  - 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."

But this "law" cannot hold forever ...

Historical Development

- Rock's Law
  - Arthur Rock, Intel financier
  - "The cost of capital equipment to build semiconductors will double every four years."
  - In 1968, a new chip plant cost about $12,000.

At the time, $12,000 would buy a nice home in the suburbs.

An executive earning $12,000 per year was "making a very comfortable living."

Historical Development

- Rock's Law
  - In 2003, a chip plants under construction will cost over $2.5 billion.

    $2.5 billion is more than the gross domestic product of some small countries, including Belize, Bhutan, and the Republic of Sierra Leone.

  - For Moore's Law to hold, Rock's Law must fall, or vice versa. But no one can say which will give out first.

Architecture & Organization

- Architecture is those attributes visible to the programmer
  - Instruction set, number of bits used for data representation, I/O mechanisms, addressing techniques.
  - e.g. Is there a multiply instruction?
- Organization is how features are implemented
  - Control signals, interfaces, memory technology.
  - e.g. Is there a hardware multiply unit or is it done by repeated addition?

Architecture & Organization

- All Intel x86 family share the same basic architecture
- The IBM System/370 family share the same basic architecture

- This gives code compatibility
    - At least backwards
- Organization differs between different versions

Structure & Function

- Structure is the way in which components relate to each other
- Function is the operation of individual components as part of the structure

What is "Computer Architecture"

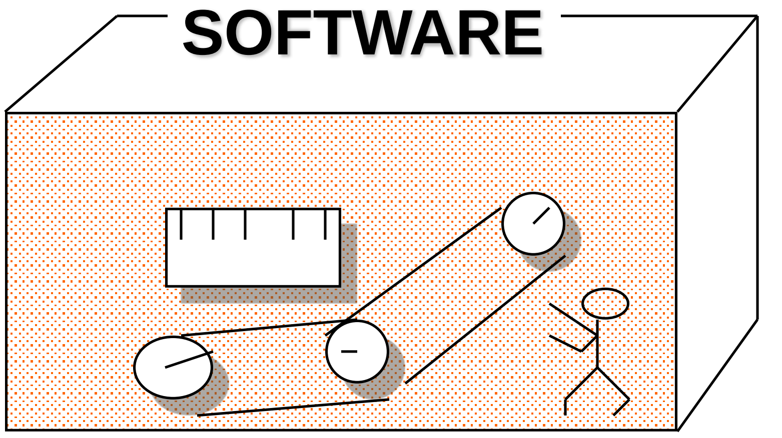
Computer Architecture =

Instruction Set Architecture +

Machine Organization

Instruction Set Architecture (subset of Computer Arch.)

... the attributes of a [computing] system as seen by the programmer, *i.e.* the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls the logic design, and the physical implementation. – Amdahl, Blaaw, and Brooks, 1964



-- **Organization of Programmable Storage**

-- **Data Types & Data Structures: Encodings & Representations**

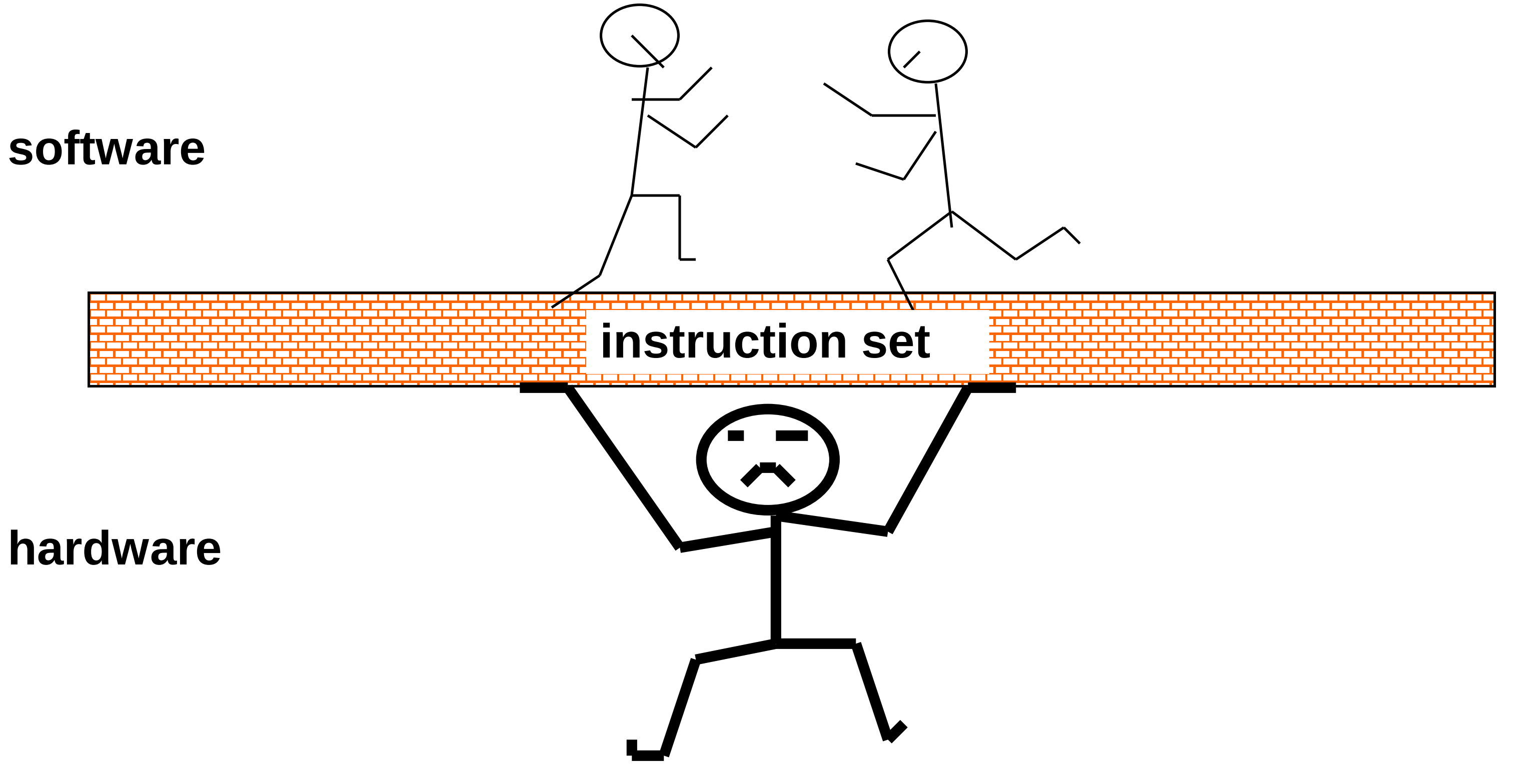
-- **Instruction Set**

-- **Instruction Formats**

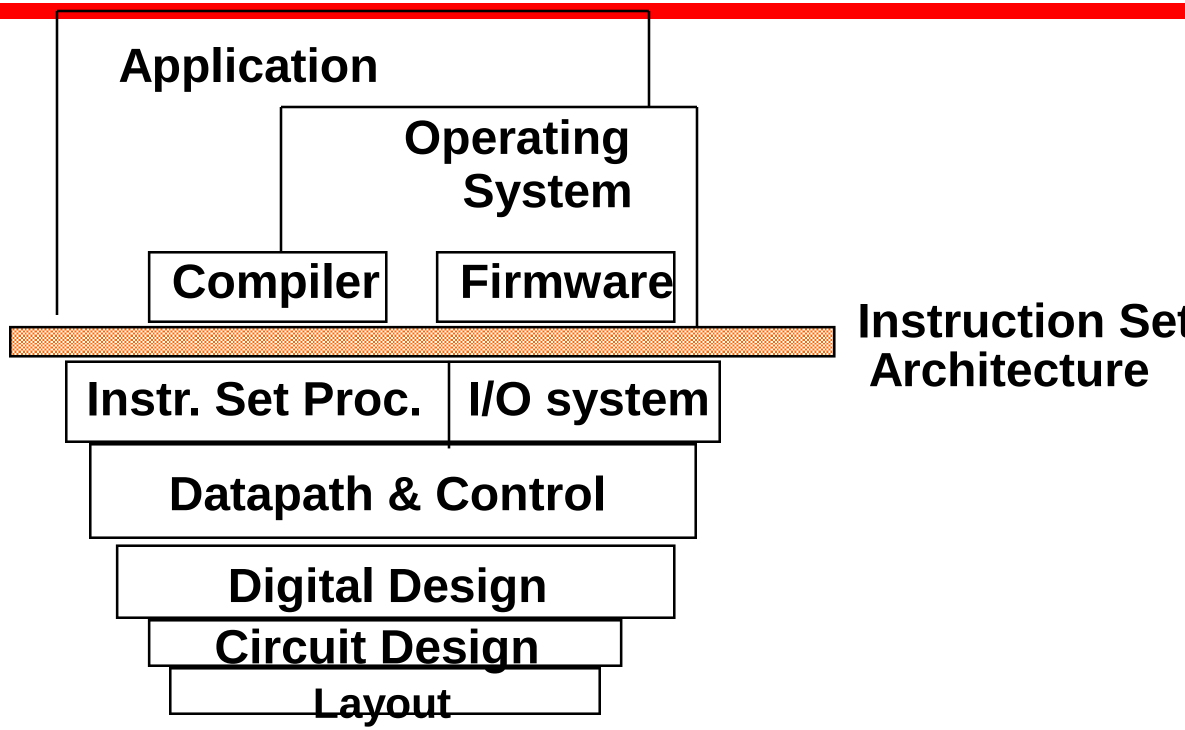
-- **Modes of Addressing and Accessing Data Items and Instructions**

-- **Exceptional Conditions**

The Instruction Set: a Critical Interface

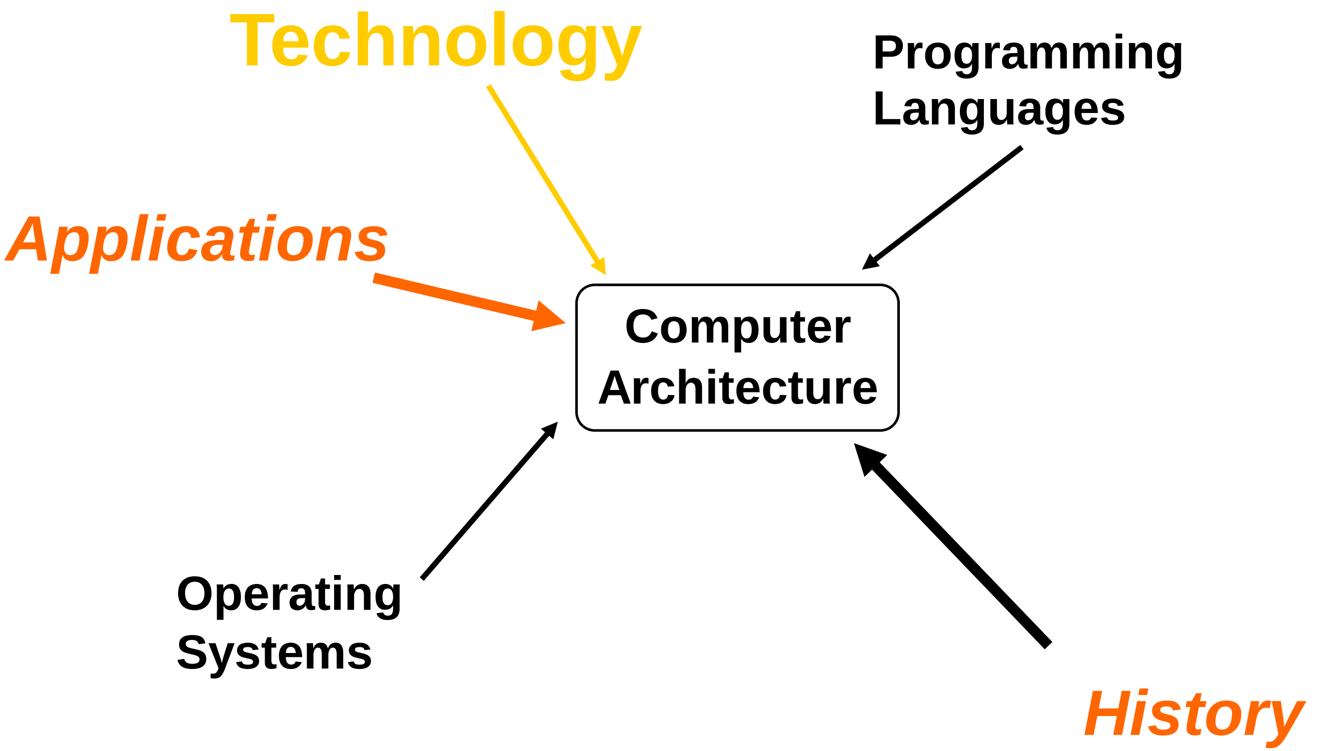


What is "Computer Architecture"?



- Coordination of many *levels of abstraction*
- Under a rapidly changing set of forces
- Design, Measurement, *and* Evaluation

Forces on Computer Architecture

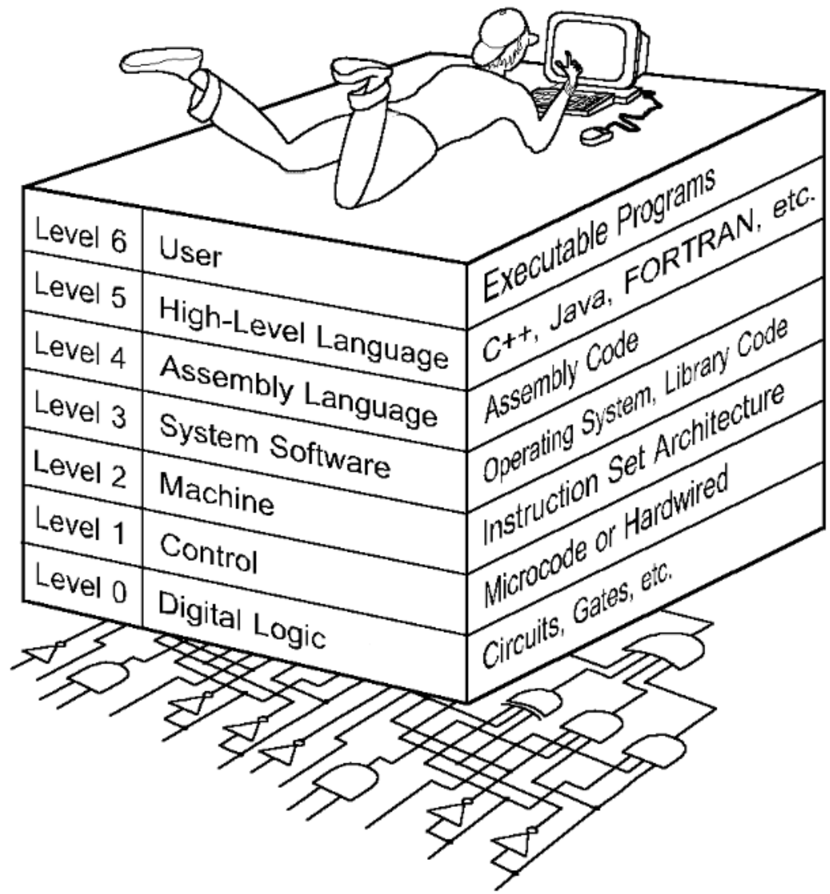


The Computer Level Hierarchy

- Computers consist of many things besides chips.
- Before a computer can do anything worthwhile, it must also use software.
- 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.

The Computer Level Hierarchy

- Each virtual machine layer is an abstraction of the level below it.
- The machines at each level execute their own particular instructions, calling upon machines at lower levels to perform tasks as required.
- Computer circuits ultimately carry out the work.



The Computer Level Hierarchy

- Level 6: The User Level
  - —Program execution and user interface level.
  - —The level with which we are most familiar.
- Level 5: High-Level Language Level
  - —The level with which we interact when we write programs in languages such as C, Pascal, Lisp, and Java.

The Computer Level Hierarchy

- Level 4: Assembly Language Level
  - Acts upon assembly language produced from Level 5, as well as instructions programmed directly at this level.
- Level 3: 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: Machine Level
  - Also known as the Instruction Set Architecture (ISA) 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.
  - Control units can be *microprogrammed* or *hardwired*.
  - A microprogram is a program written in a low-level language that is implemented by the hardware.
  - Hardwired control units consist of hardware that directly executes machine instructions.

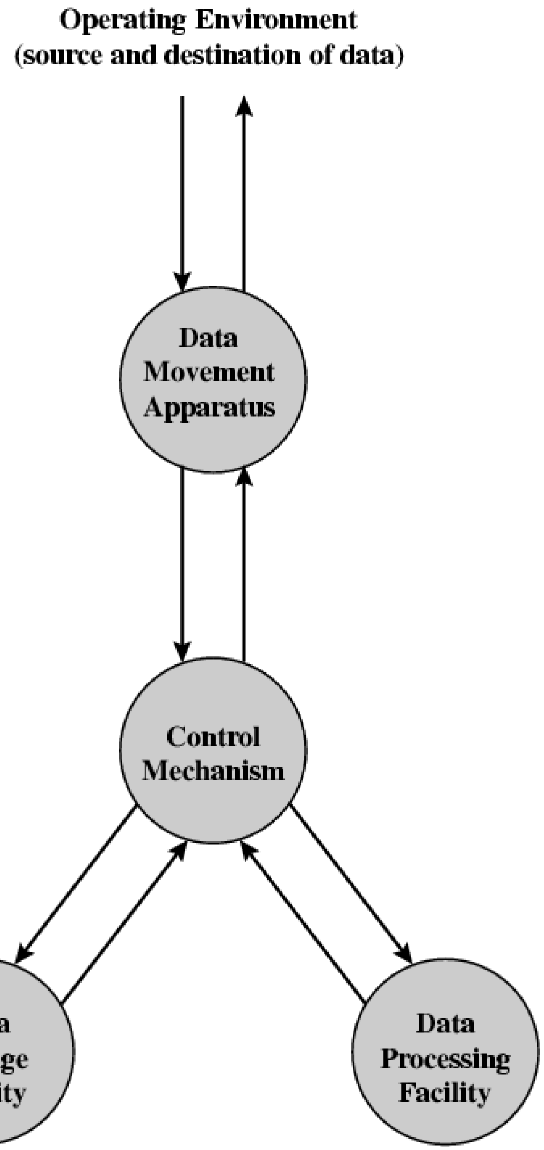
The Computer Level Hierarchy

- Level 0: Digital Logic Level
  - —This level is where we find digital circuits (the chips).
  - —Digital circuits consist of gates and wires.
  - —These components implement the mathematical logic of all other levels.

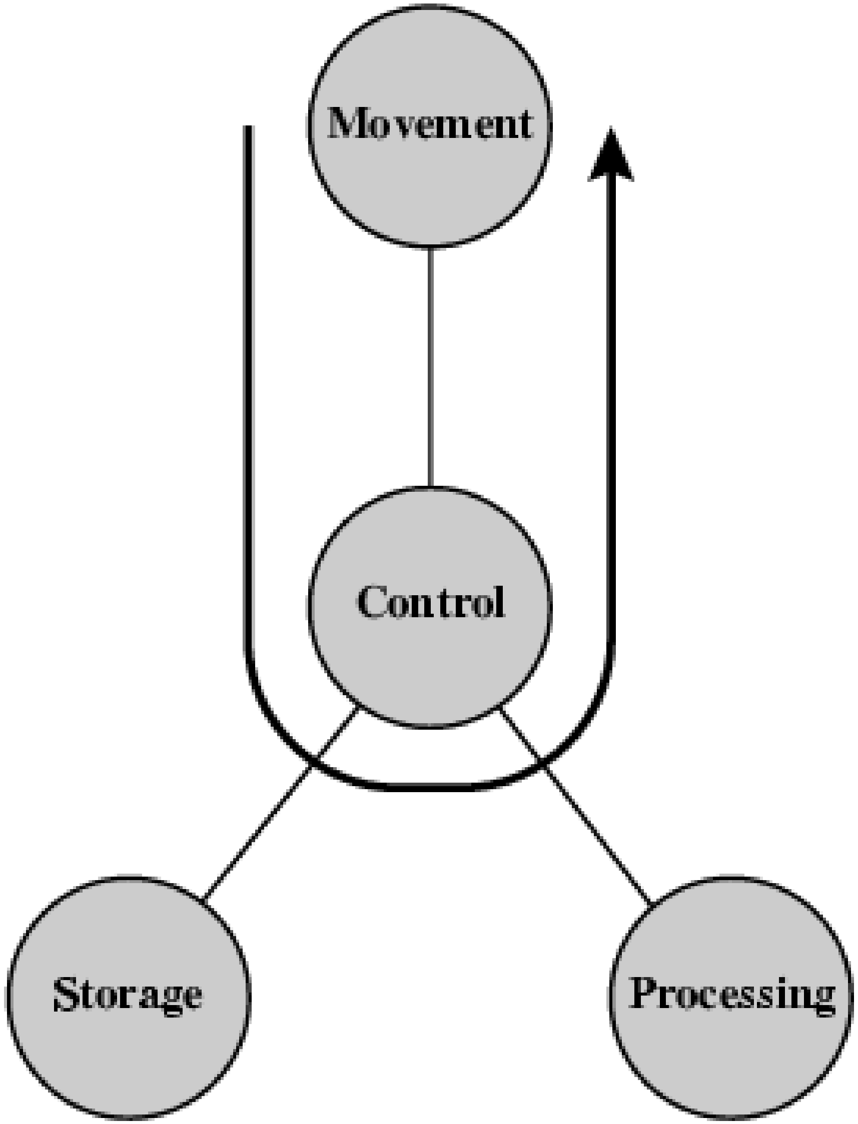
Function

- All computer functions are:
  - Data processing
  - Data storage
  - Data movement
  - Control

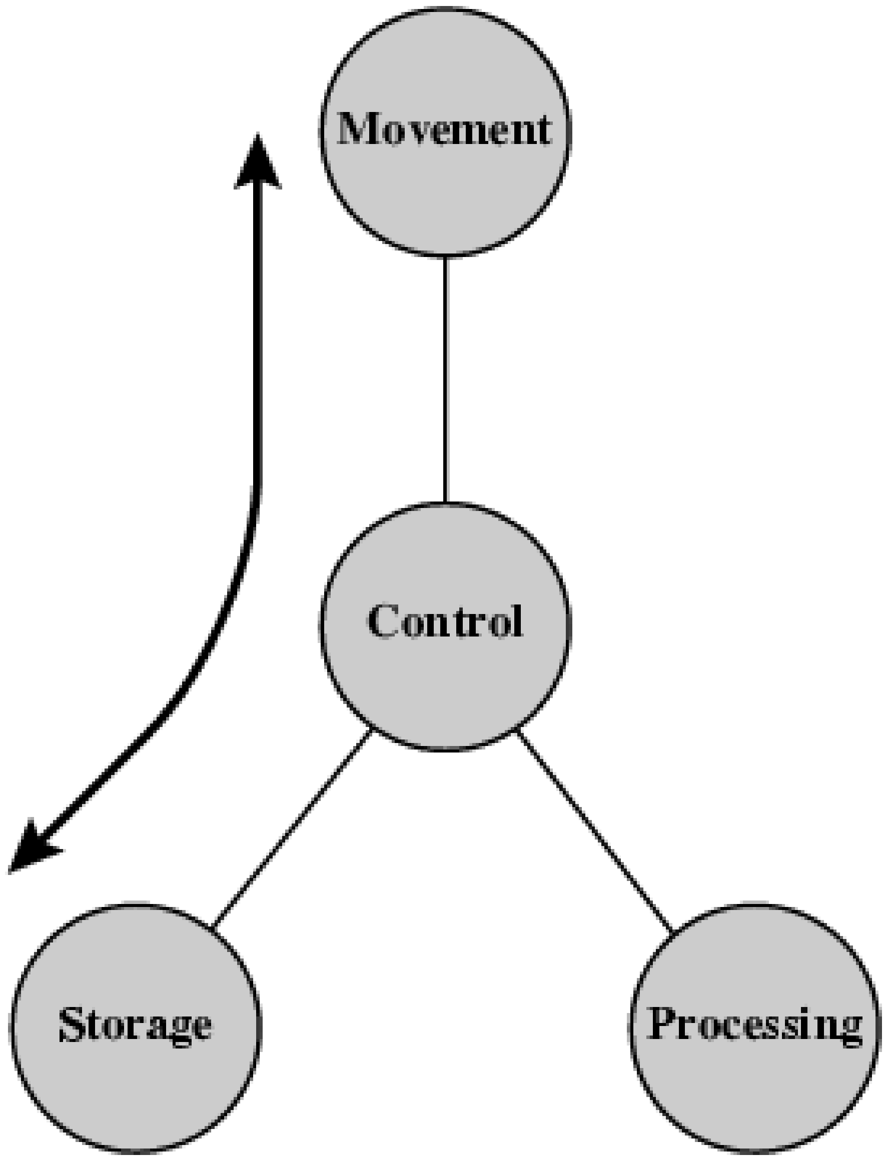
Functional View



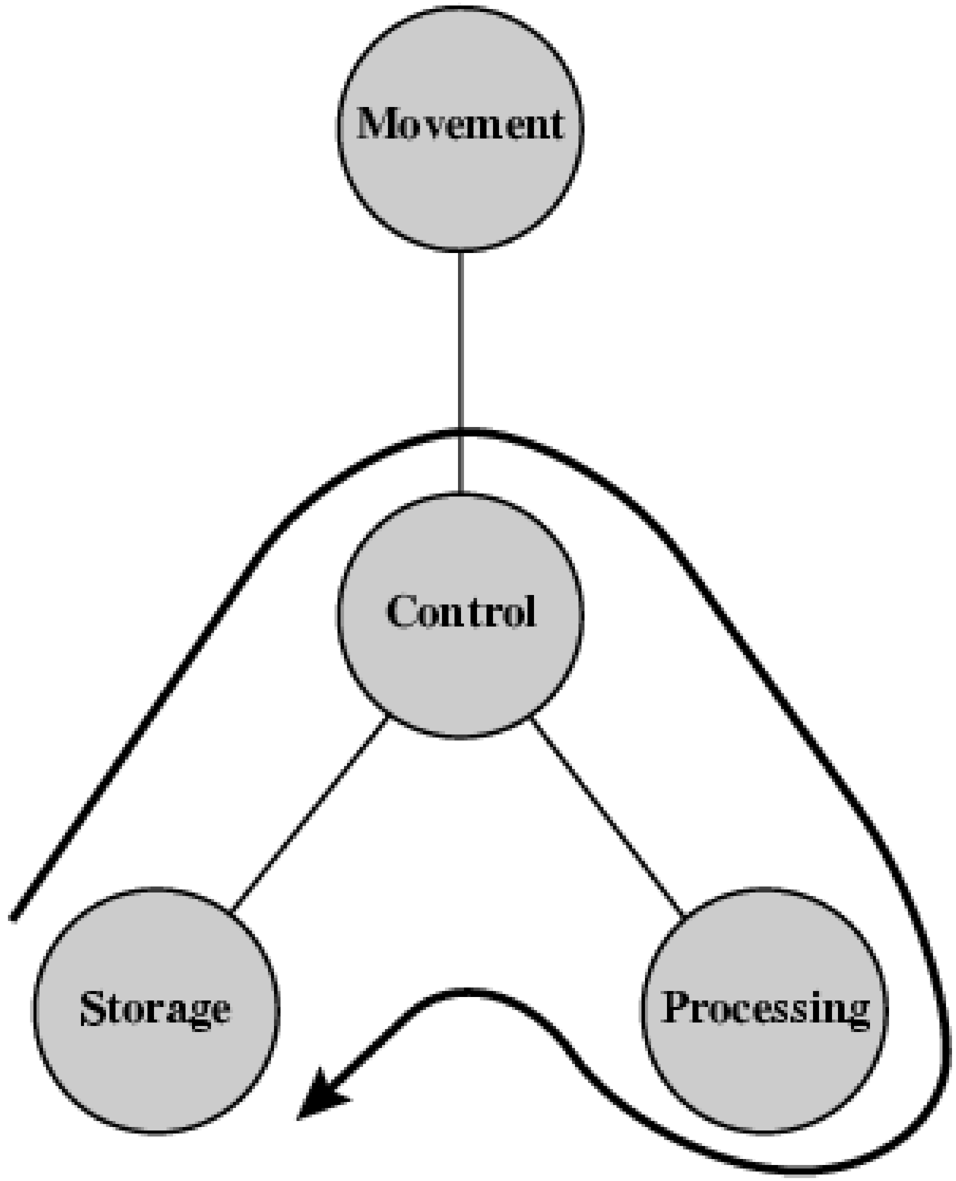
Operations (a) Data movement



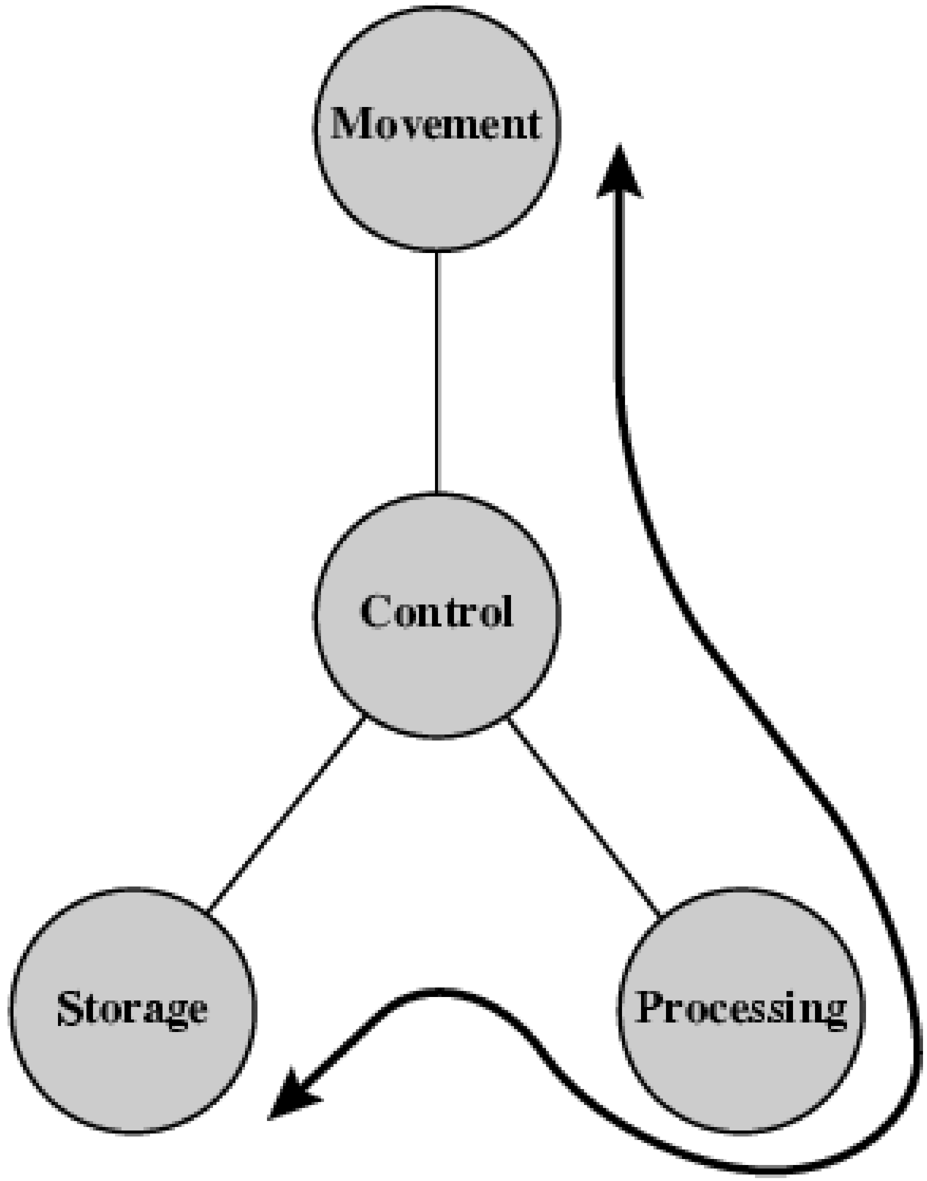
Operations (b) Storage



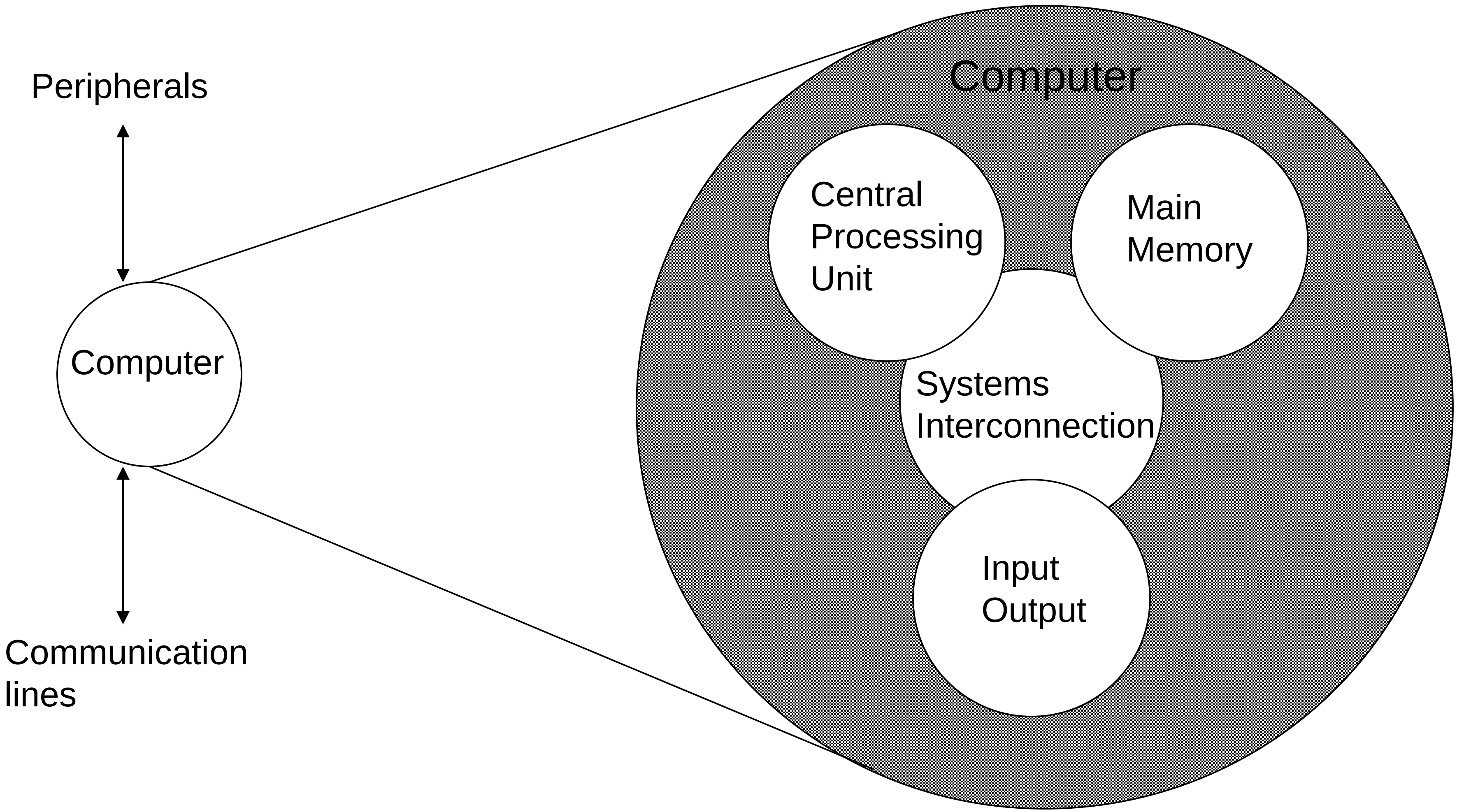
Operation (c) Processing from/to storage



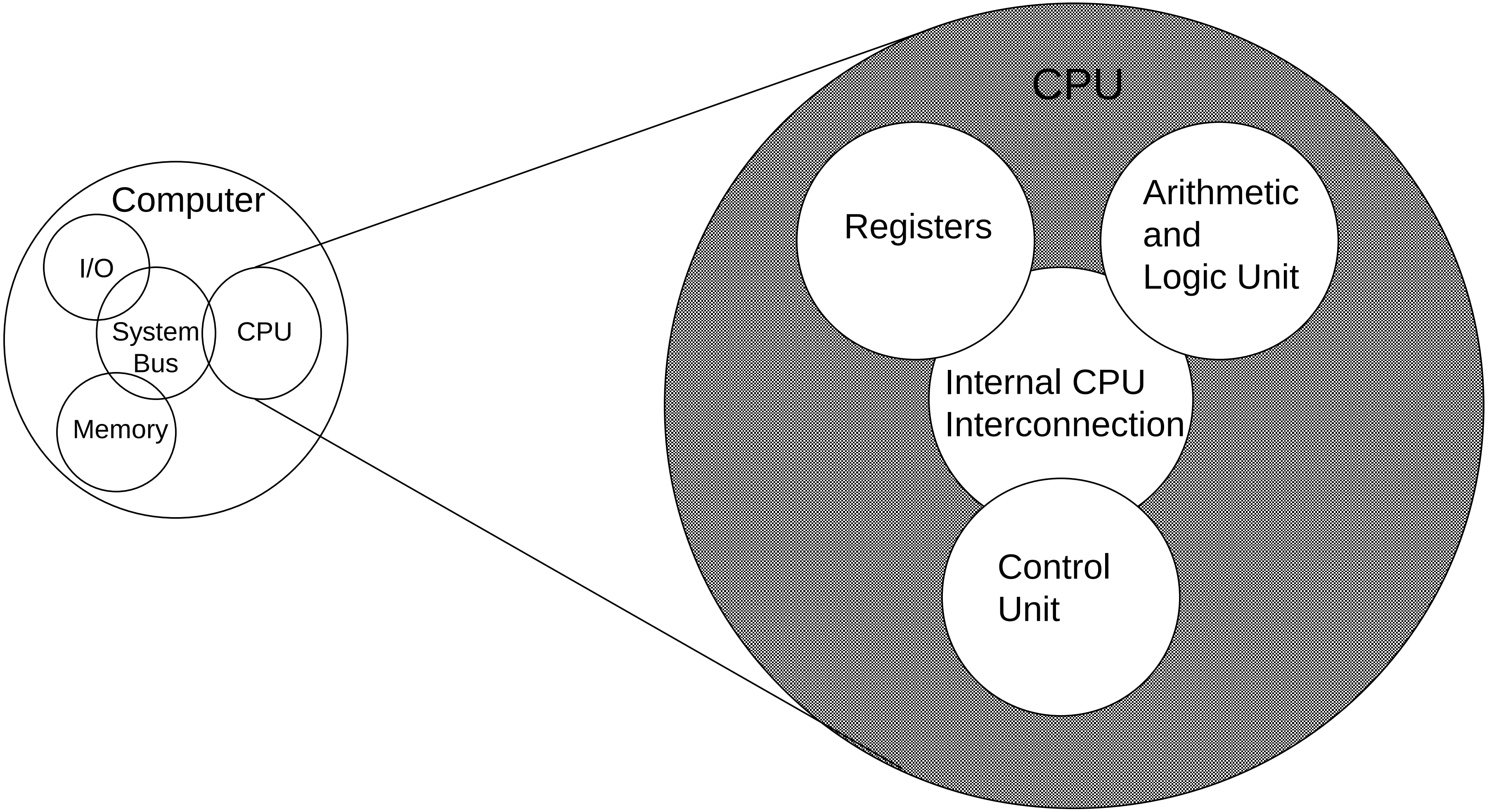
Operation (d)
Processing from storage to I/O



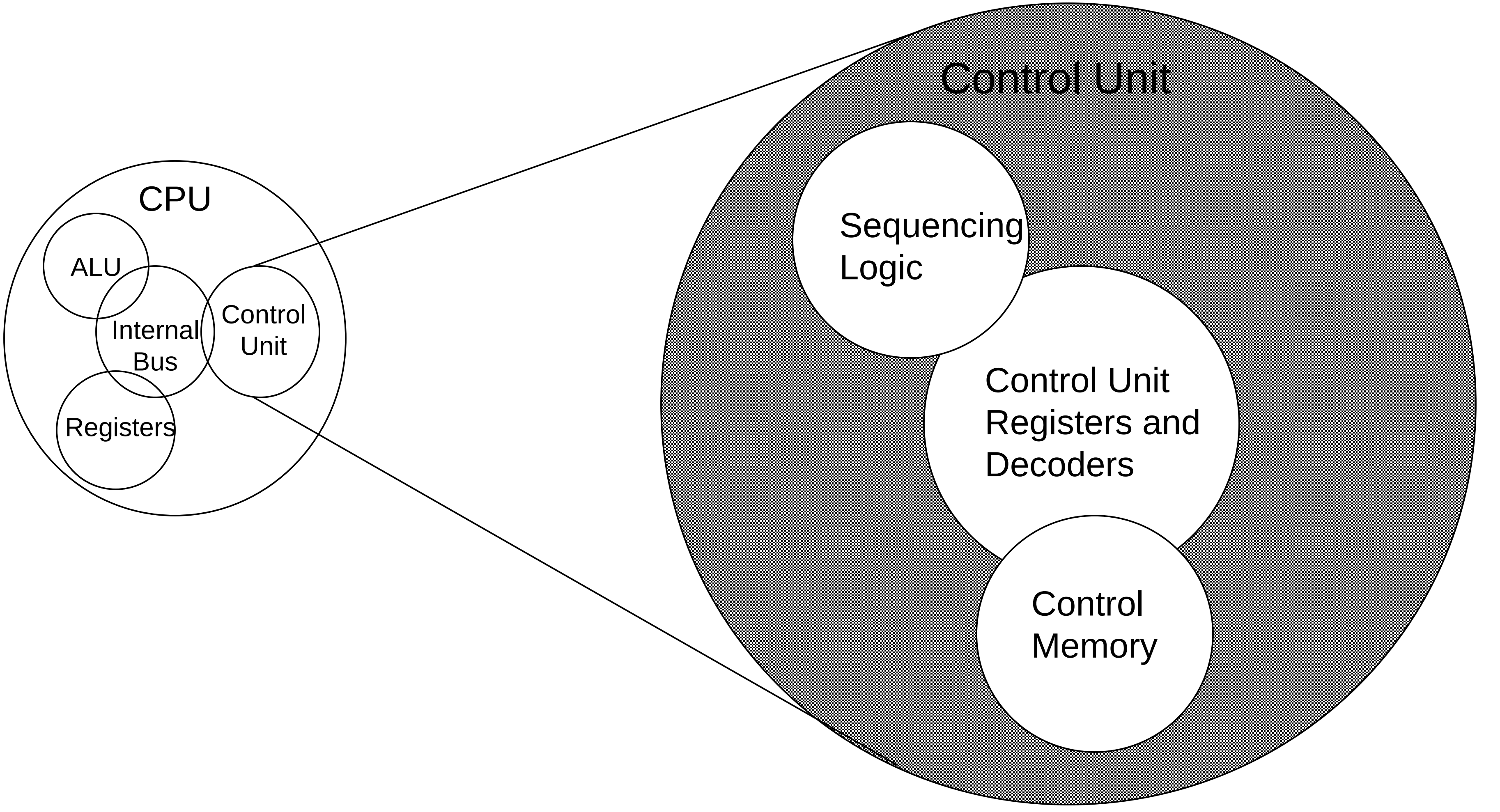
Structure - Top Level



Structure - The CPU



Structure - The Control Unit



Levels of Representation

High Level Language Program

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

Compiler

Assembly Language Program

```
lw  $15,   0($2)
lw  $16,   4($2)
sw  $16,   0($2)
sw  $15,   4($2)
```

Assembler

Machine Language Program

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

Machine Interpretation

Control Signal Specification

ALUOP[0:3] <= InstReg[9:11] & MASK

°

°

Internet Resources
- Web sites to look for

- WWW Computer Architecture Home Page
- CPU Info Center
- Processor Emporium
- ACM Special Interest Group on Computer Architecture
- IEEE Technical Committee on Computer Architecture
- Intel Technology Journal
- Manufacturer's sites
  - Intel, IBM, etc.

Internet Resources
- Usenet News Groups

- comp.arch
- comp.arch.arithmetic
- comp.arch.storage
- comp.parallel