

Computer Organization (ENCS238)

Dr. Abualsoud Hanani

Electrical and Computer Engineering

Introduction

Chapters (1 + 2)

Course Information I

| Textbook | |
|------------------|---|
| Title | Computer Organization & Architecture |
| Author | W. Stallings |
| Publisher | Prentice Hall |
| Edition | 7 th or 8 th |
| ISBN | 0-13-185644-8 |
| References | <ul style="list-style-type: none">• Fundamentals of Computer Organization and Architecture, Mostafa Abd-El-Barr & Hesham El-Rewini, 2005 by John Wiley & Sons, Inc.• Computer Systems Architecture, M. M. Mano, Prentice Hall 1992, 2nd edition• IBM PC Assembly Language & Programming, Peter Abel, Prentice Hall 5th edition• Computer Organization & Design, Patterson & Hennessy, Morgan Kaufman 1998 2nd edition |
| Course Materials | Textbook + Lecture Notes |

This course is about:

- **What computers consist of**
- **How computers work**
- **How they are organized internally**
- **What are the design tradeoffs**
- **How design affects programming and applications**

- **How to fix computers**
- **How to build myself one real cheap**
- **Which one to buy**
- **Knowing all about the Pentium IV or PowerPC**

Architecture & Organization 1

- 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 2

- 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

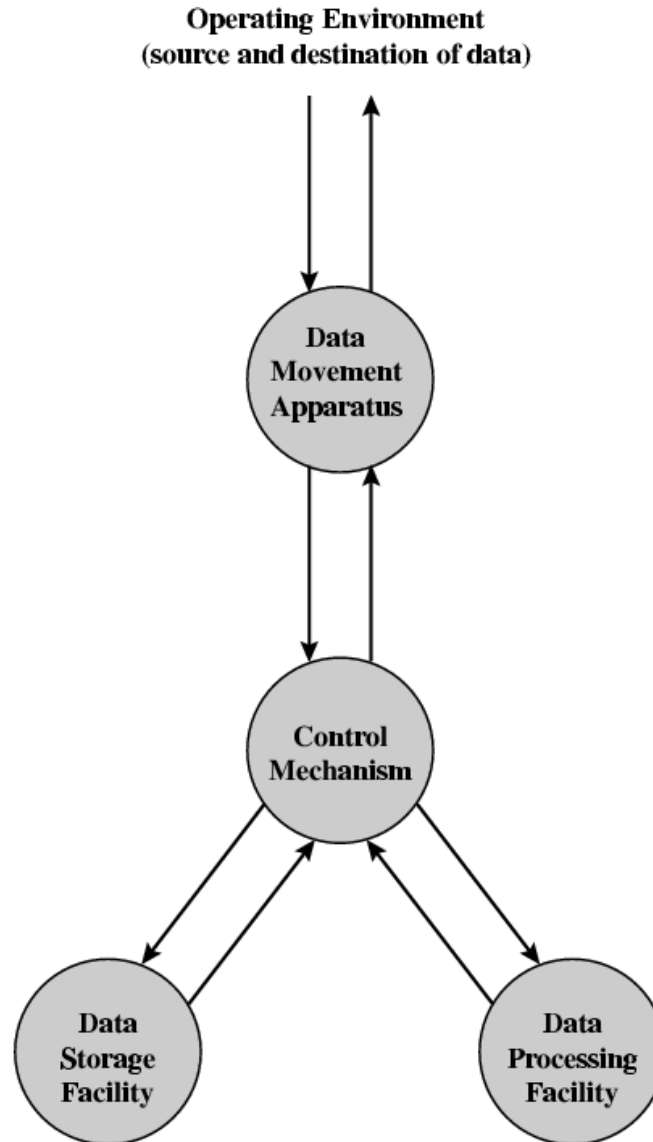
Structure & Function

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

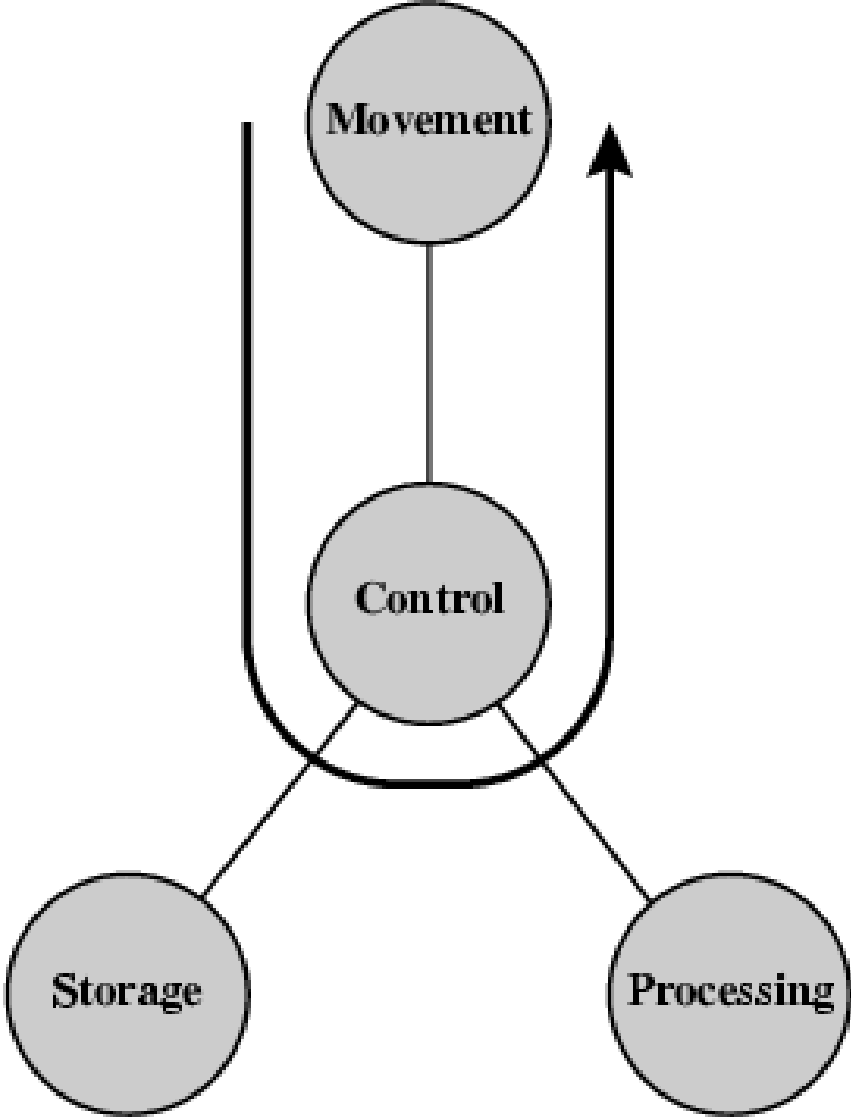
Function

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

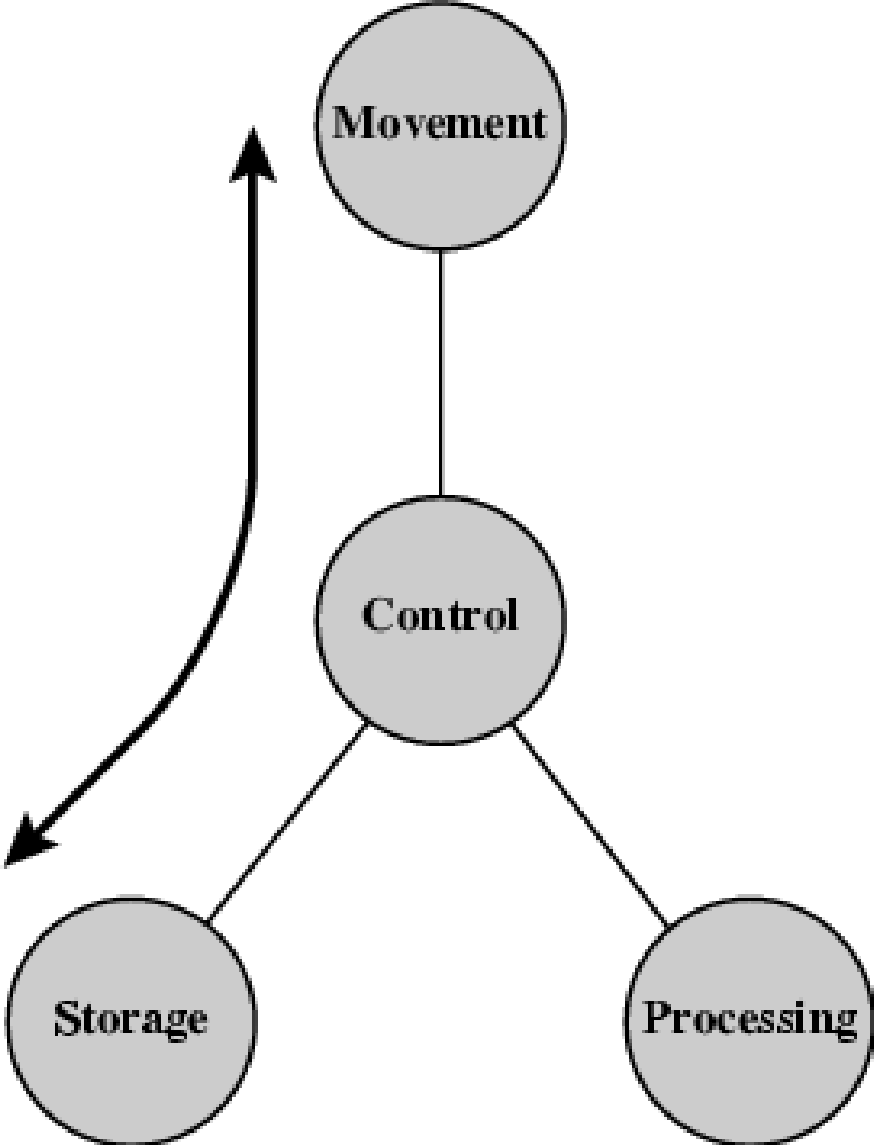
Functional View



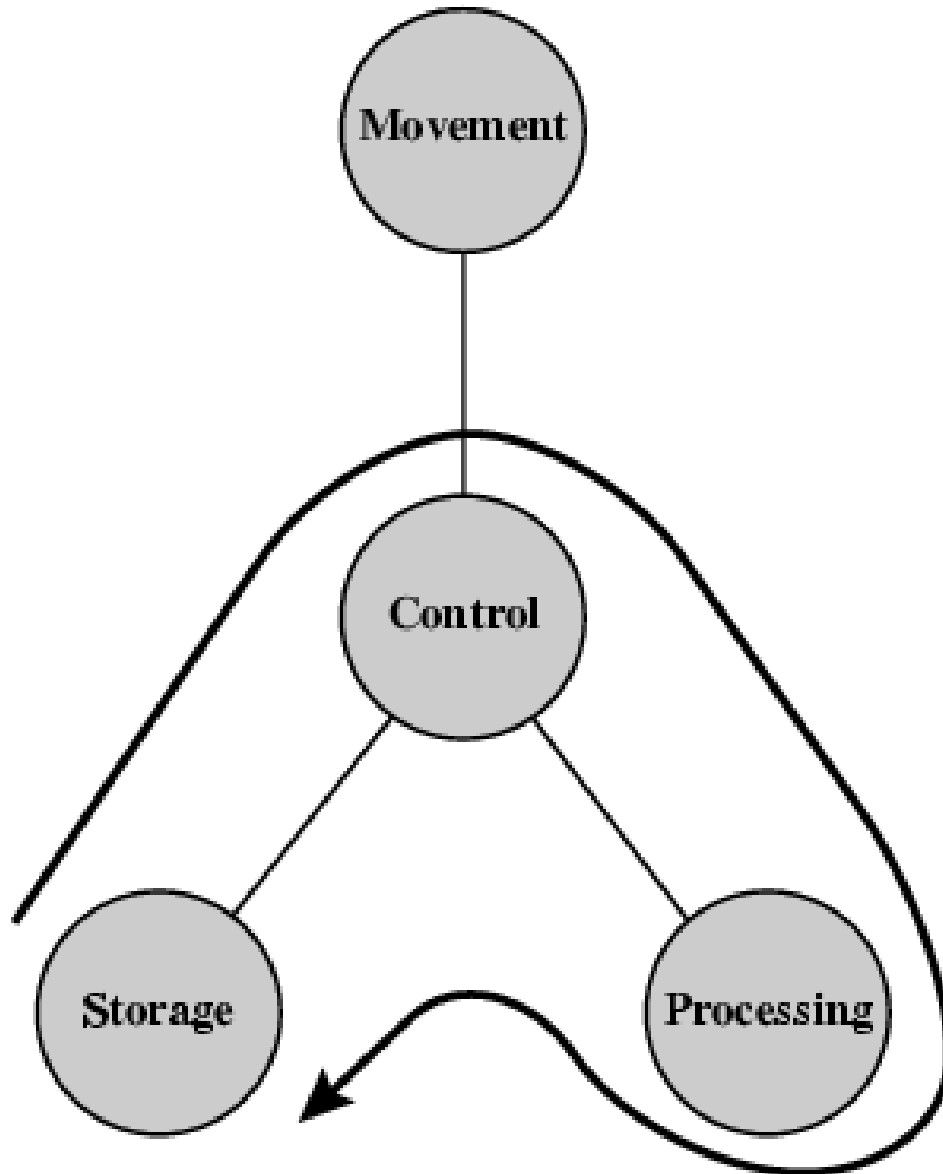
Operations (a) Data movement



Operations (b) Storage

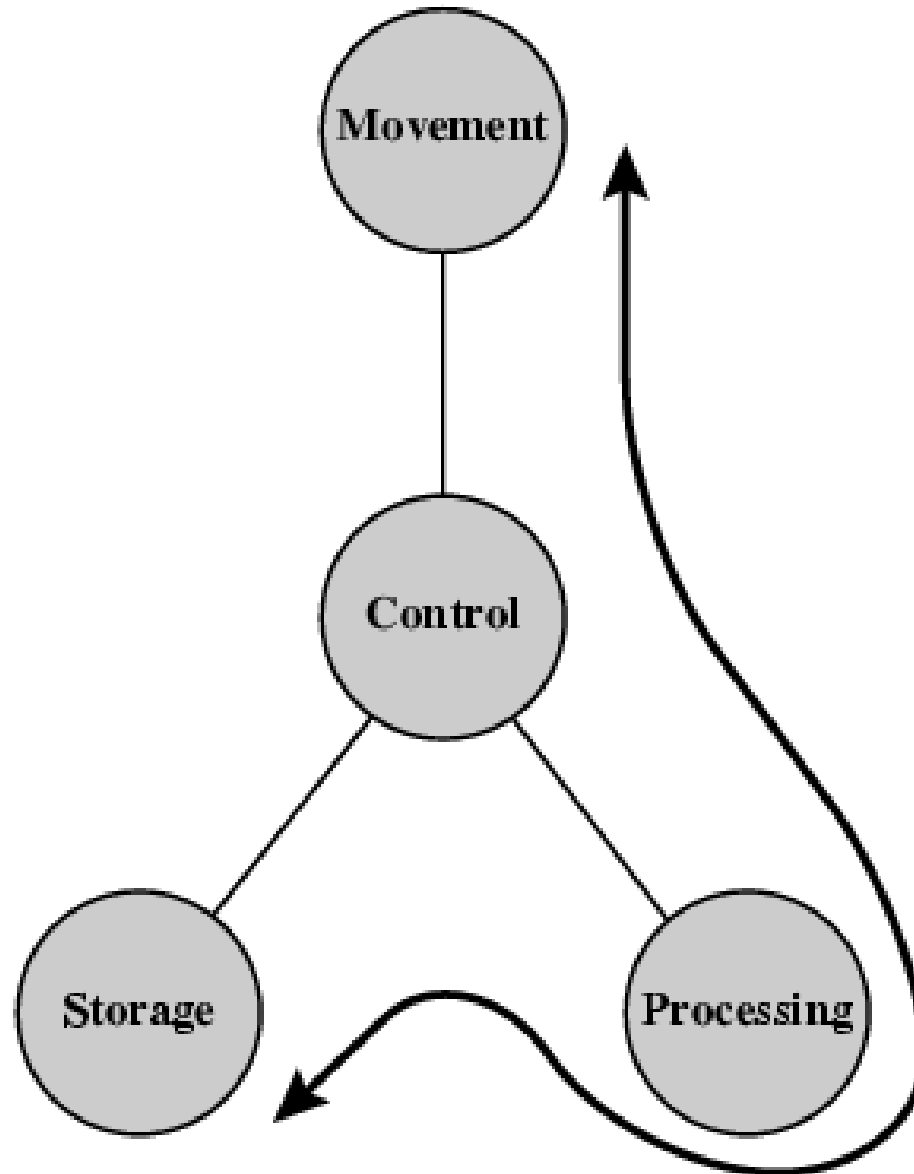


Operation (c) Processing from/to storage

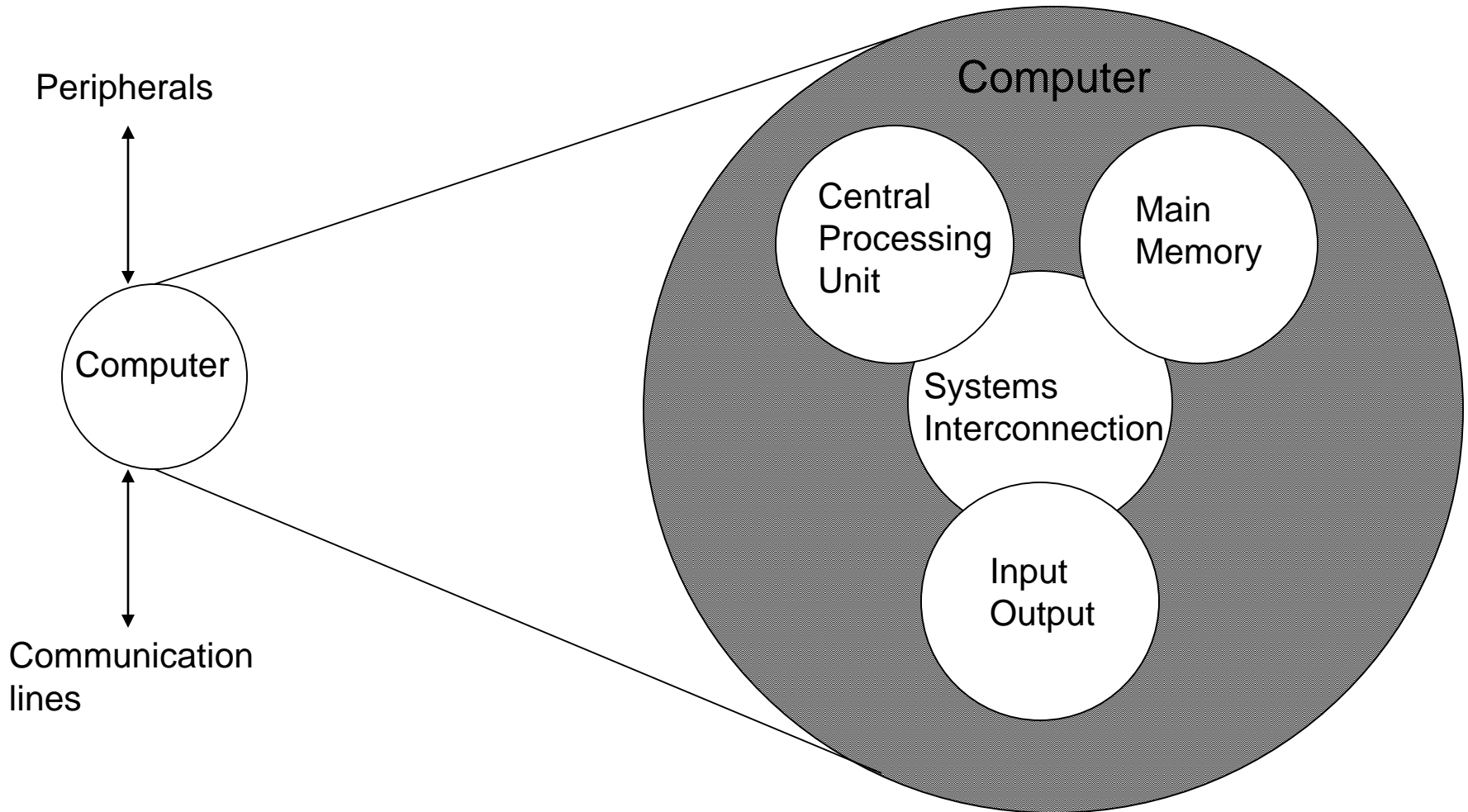


Operation (d)

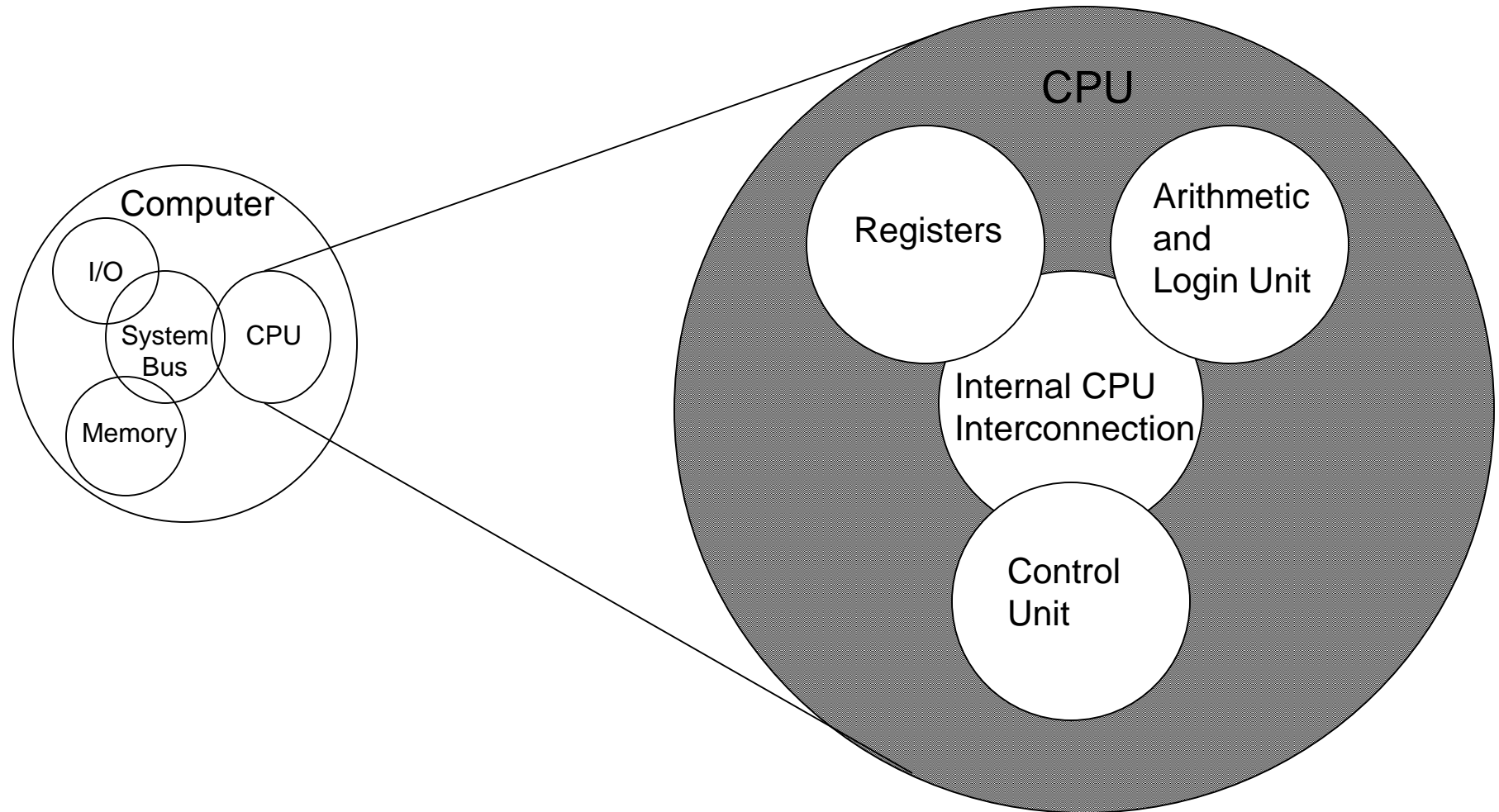
Processing from storage to I/O



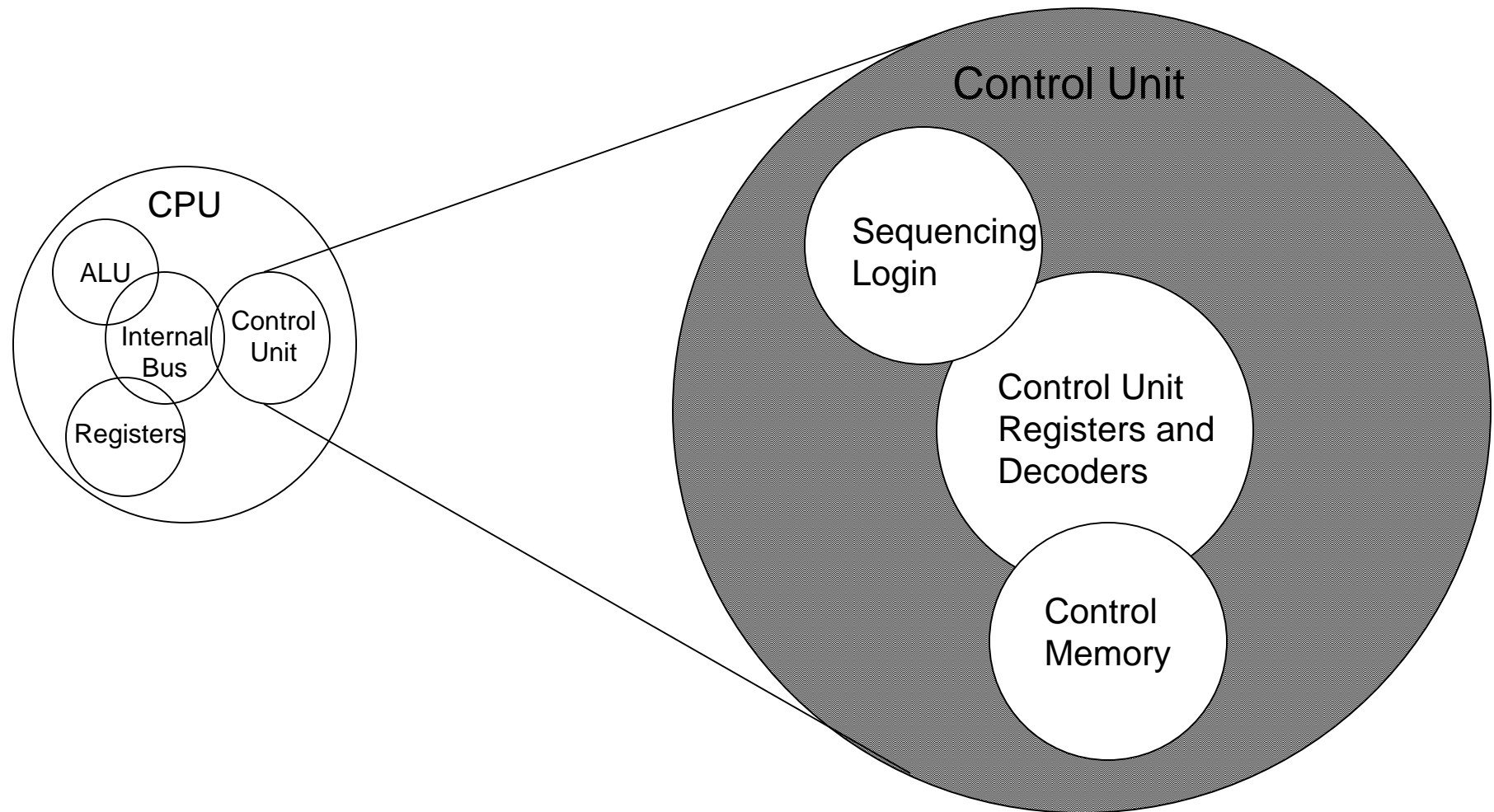
Structure - Top Level



Structure - The CPU



Structure - The Control Unit



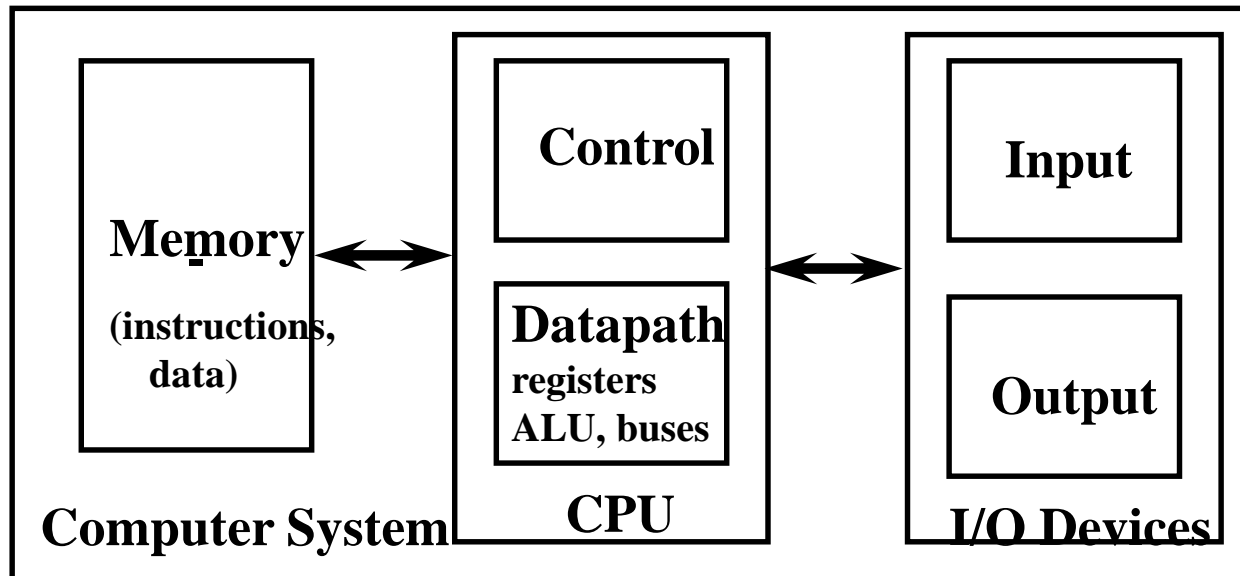
General Purpose Processor/Computer System Generations

Classified according to implementation technology:

- The First Generation, 1946-59: Vacuum Tubes, Relays, Mercury Delay Lines:
 - ENIAC (Electronic Numerical Integrator and Computer): First electronic computer, 18000 vacuum tubes, 1500 relays, 5000 additions/sec (1944).
 - First stored program computer: EDSAC (Electronic Delay Storage Automatic Calculator), 1949.
- The Second Generation, 1959-64: Discrete Transistors.
 - e.g. IBM Main frames
- The Third Generation, 1964-75: Small and Medium-Scale Integrated (MSI) Circuits.
 - e.g. Main frames (IBM 360) , mini computers (DEC PDP-8, PDP-11).
- The Fourth Generation, 1975-Present: The Microcomputer. VLSI-based Microprocessors (single-chip processor)
 - First microprocessor: Intel's 4-bit 4004 (2300 transistors), 1970.
 - Personal Computer (PCs), laptops, PDAs, servers, clusters ...
 - Reduced Instruction Set Computer (RISC) 1984

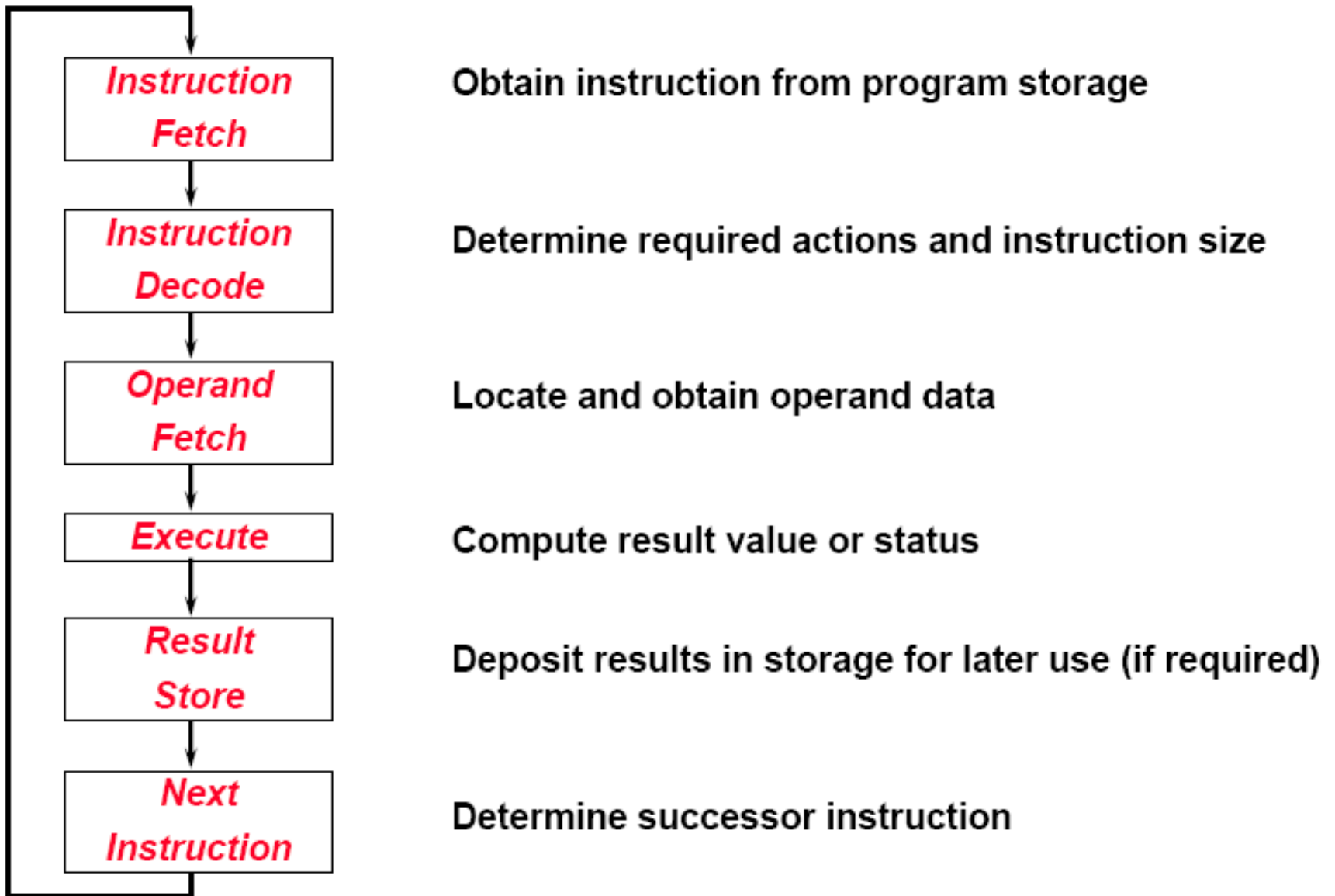
The Von Neumann Computer Model

- Partitioning of the computing engine into components:
 - **Central Processing Unit (CPU):** Control Unit (instruction decode , sequencing of operations), Datapath (registers, arithmetic and logic unit, buses).
 - **Memory:** Instruction and operand storage.
 - **Input/Output (I/O) sub-system:** I/O bus, interfaces, devices.
 - **The stored program concept:** Instructions from an instruction set are fetched from a common memory and executed one at a time

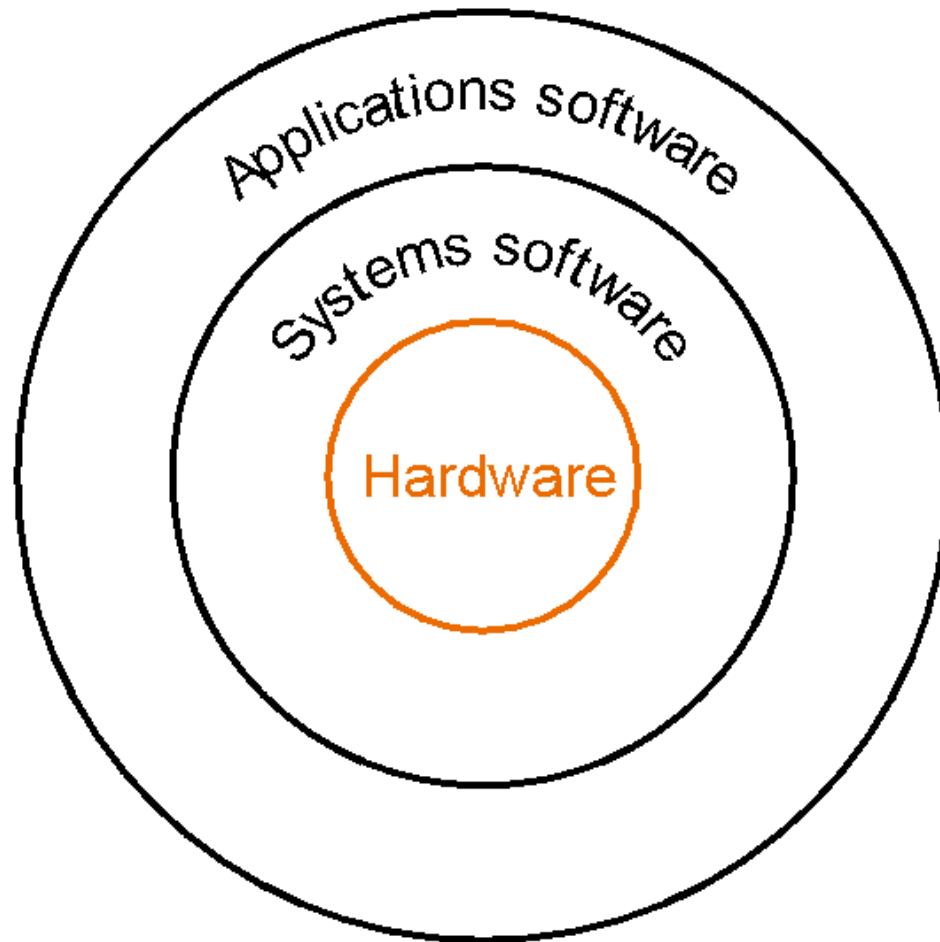


Major CPU Performance Limitation: The Von Neumann computing model implies sequential execution one instruction at a time

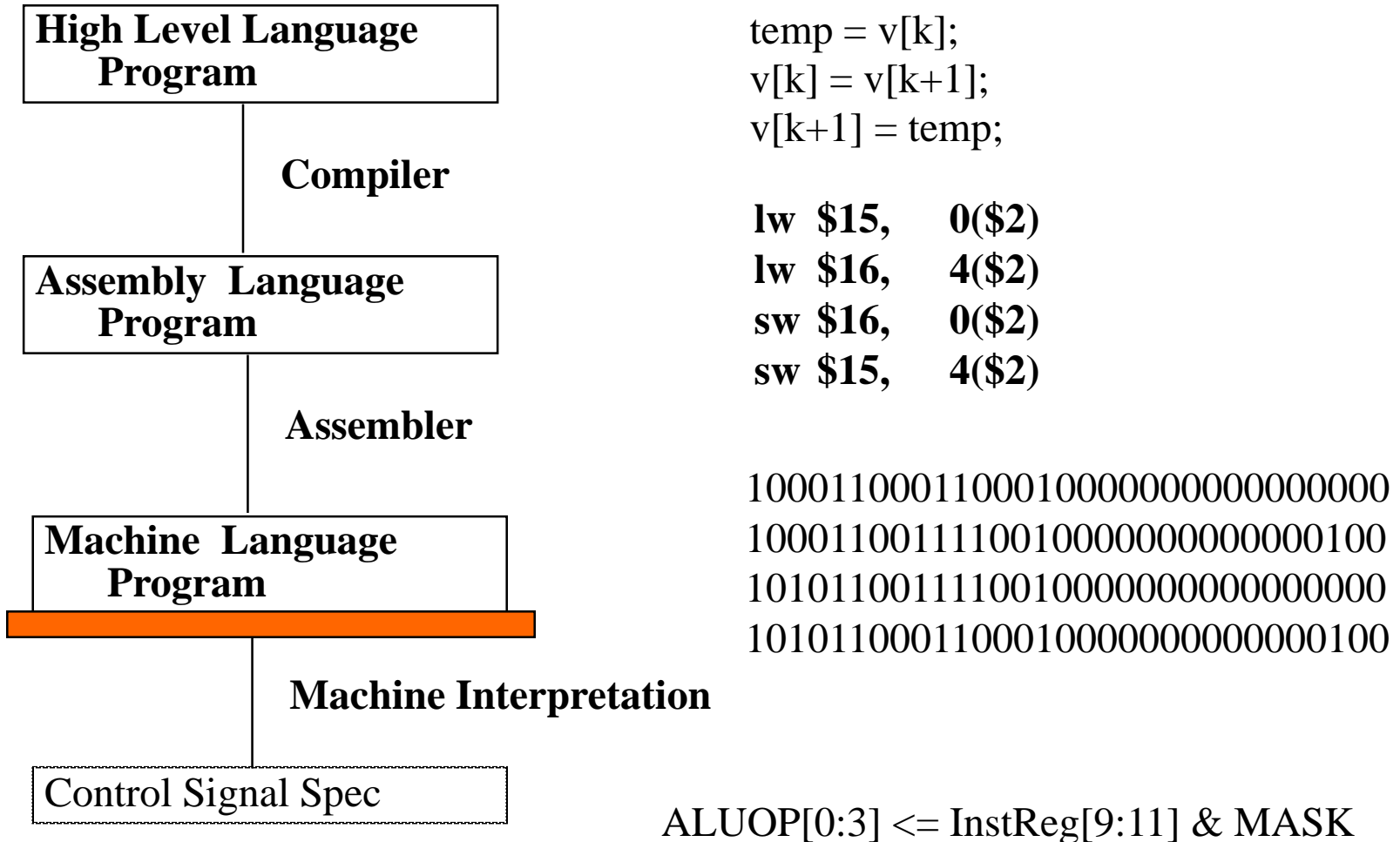
Generic CPU Machine Instruction Execution Steps



A Simplified View of The Software/Hardware Hierarchical Layers



How to Speak Computer



Need translation from application to physics

The Big (Simplified) Picture

High-level code

```
char *tmpfilename;
int num_schedulers=0;
int num_request_submitters=0;
int i,j;

if (!(f = fopen(filename,"r"))) {
  xbt_assert1(0,"Cannot open file %s",filename);
}
while(fgets(buffer,256,f)) {
  if (strncmp(buffer,"SCHEDULER",9))
    num_schedulers++;
  if (strncmp(buffer,"REQUESTSUBMITTER",16))
    num_request_submitters++;
}
fclose(f);
tmpfilename = strdup("/tmp/obsimulator_
```

COMPILER

ASSEMBLER

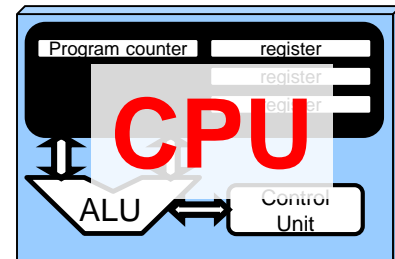
Machine code

```
010000101010110110
101010101111010101
101001010101010001
101010101010100101
111100001010101001
000101010111101011
01000000010000100
000010001000100011
101001010010101011
000101010100100101
010101010101010101
101010101111010101
101010101010100101
111100001010101001
```

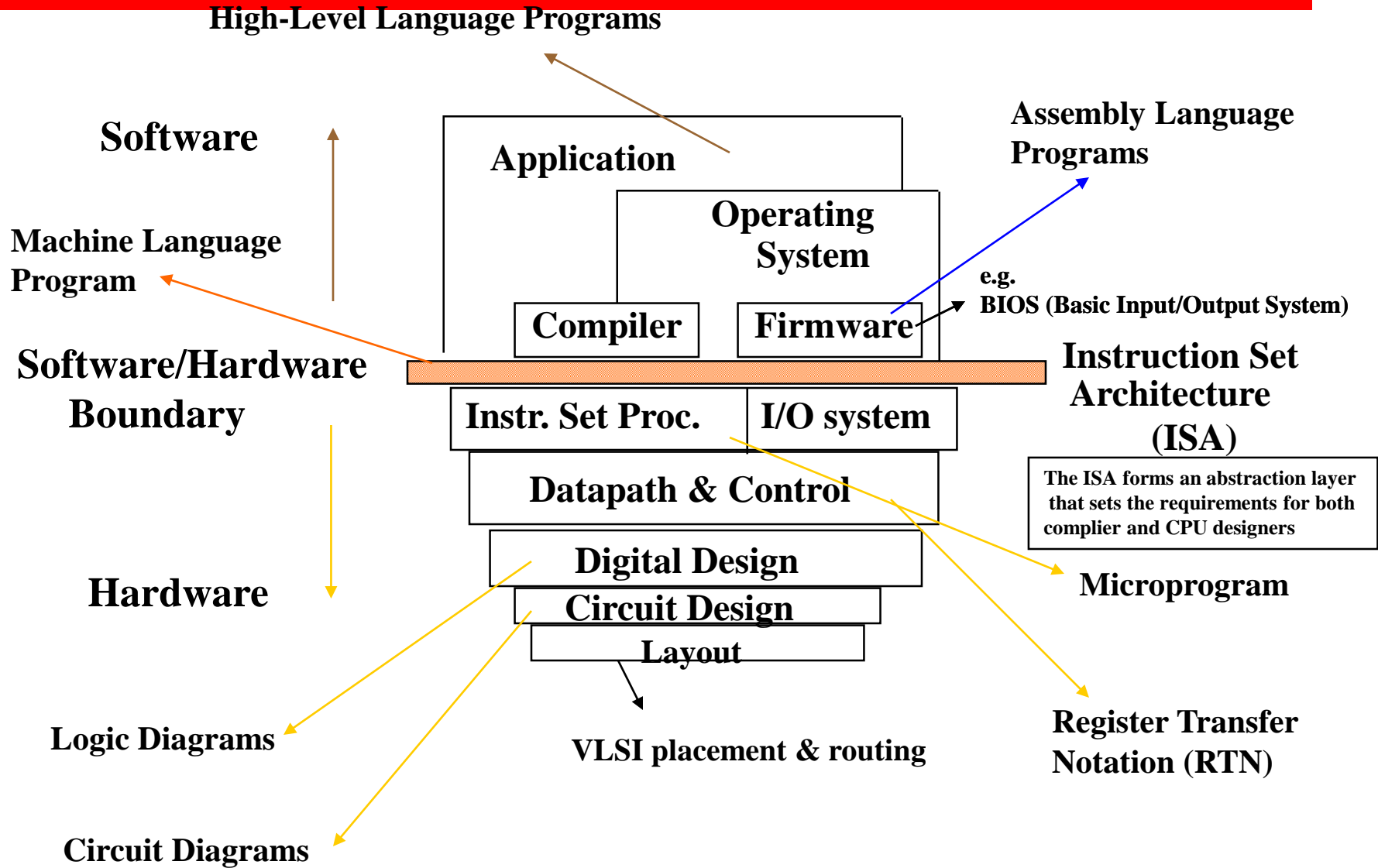
Assembly code

```
sll $t3, $t1, 2
add $t3, $s0, $t3
sll $t4, $t0, 2
add $t4, $s0, $t4
lw $t5, 0($t3)
lw $t6, 0($t4)
slt $t2, $t5, $t6
beq $t2, $zero, endif
add $t0, $t1, $zero
sll $t4, $t0, 2
add $t4, $s0, $t4
lw $t5, 0($t3)
lw $t6, 0($t4)
slt $t2, $t5, $t6
beq $t2, $zero, endif
```

CPU



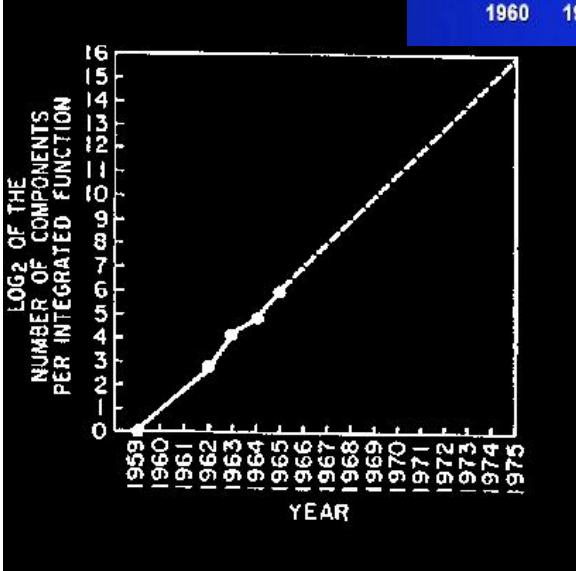
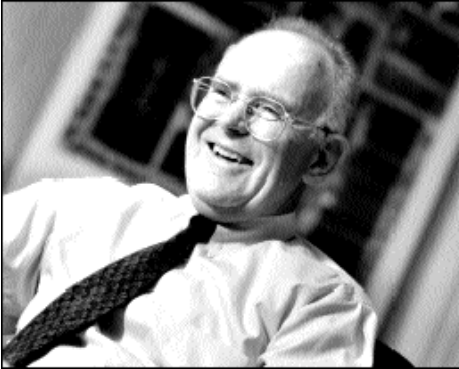
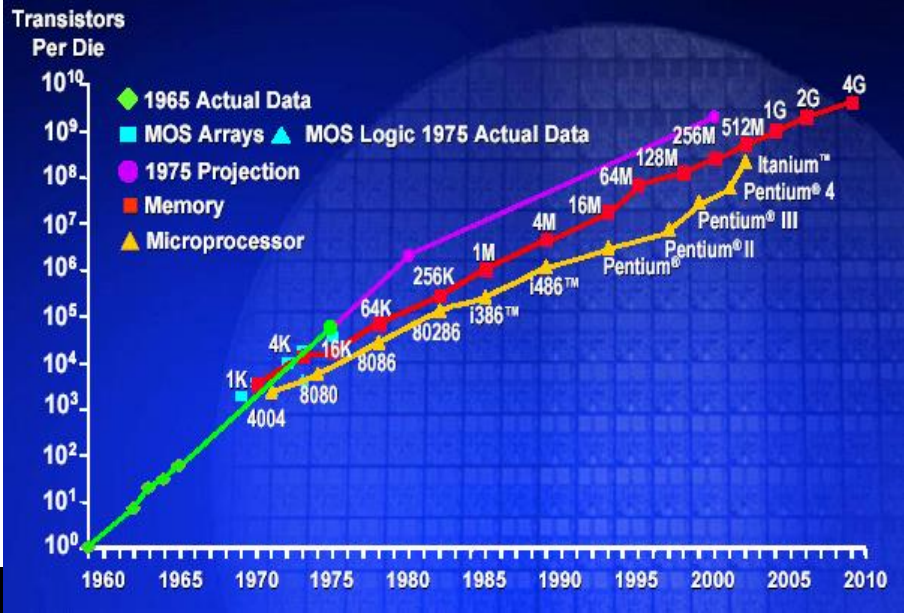
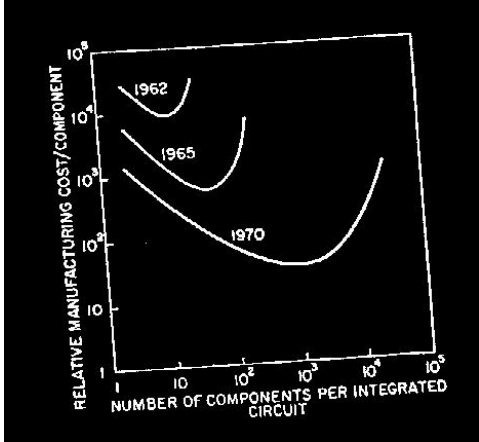
Hierarchy of Computer Architecture



Technology Change

- Technology changes rapidly
 - HW
 - Vacuum tubes: Electron emitting devices
 - Transistors: On-off switches controlled by electricity
 - Integrated Circuits(IC/ Chips): Combines thousands of transistors
 - Very Large-Scale Integration(VLSI): Combines millions of transistors
 - What next?
 - SW
 - Machine language: Zeros and ones
 - Assembly language: Mnemonics
 - High-Level Languages: English-like
 - Artificial Intelligence languages: Functions & logic predicates
 - Object-Oriented Programming: Objects & operations on objects

Moore's Law: 2X transistors / "year"



Moore's Law

- Increased density of components on chip
- Gordon Moore – co-founder of Intel
- Number of transistors on a chip will double every year
- Since 1970's development has slowed a little
 - Number of transistors doubles every 18 months
- Cost of a chip has remained almost unchanged
- Higher packing density means shorter electrical paths, giving higher performance
- Smaller size gives increased flexibility
- Reduced power and cooling requirements
- Fewer interconnections increases reliability

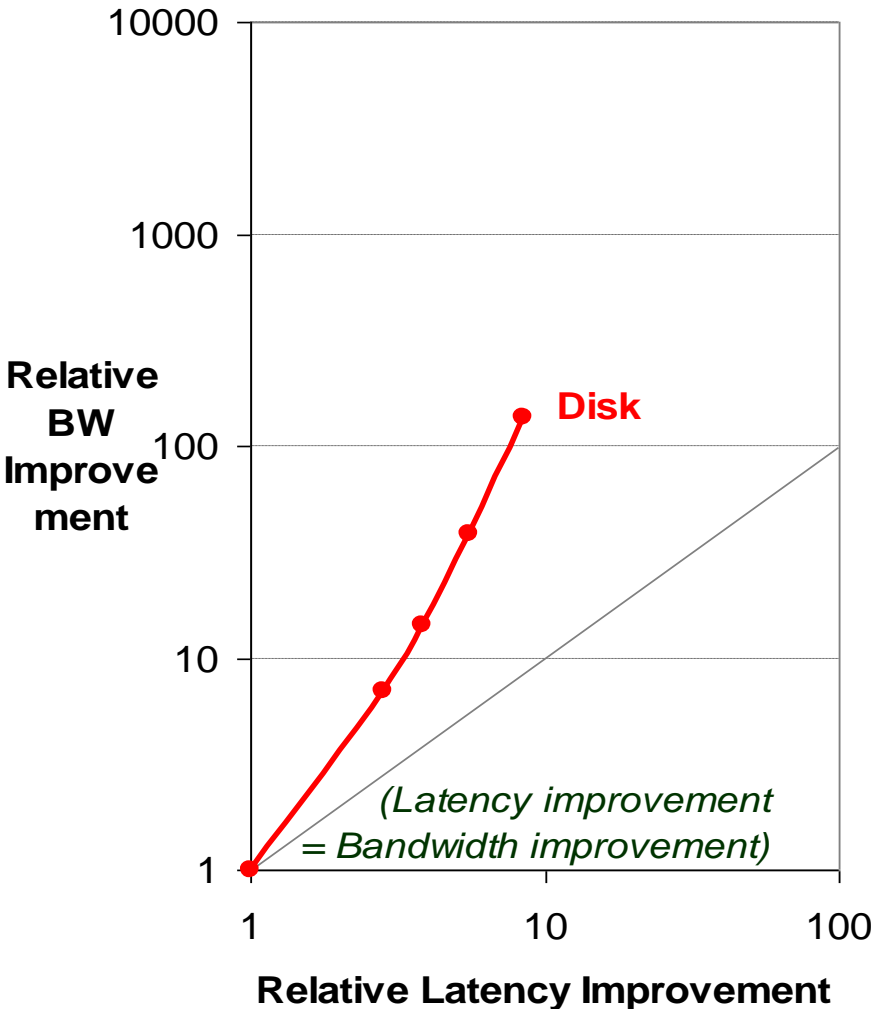
Tracking Technology Performance Trends

- Drill down into 4 technologies:
 - Disks,
 - Memory,
 - Network,
 - Processors
- Compare for Bandwidth vs. Latency improvements in performance over time
- Bandwidth: number of events per unit time
 - E.g., M bits / second over network, M bytes / second from disk
- Latency: elapsed time for a single event
 - E.g., one-way network delay in microseconds, average disk access time in milliseconds

Disks: Archaic(Nostalgic) v. Modern(Newfangled)

- CDC Wren I, 1983
- 3600 RPM
- 0.03 GBytes capacity
- Tracks/Inch: 800
- Bits/Inch: 9550
- Three 5.25" platters
- Bandwidth:
0.6 MBytes/sec
- Latency: 48.3 ms
- Cache: none
- Seagate 373453, 2003
- 15000 RPM (4X)
- 73.4 GBytes (2500X)
- Tracks/Inch: 64000 (80X)
- Bits/Inch: 533,000 (60X)
- Four 2.5" platters
(in 3.5" form factor)
- Bandwidth:
86 MBytes/sec (140X)
- Latency: 5.7 ms (8X)
- Cache: 8 MBytes

Latency Lags Bandwidth (for last ~20 years)



- Performance Milestones

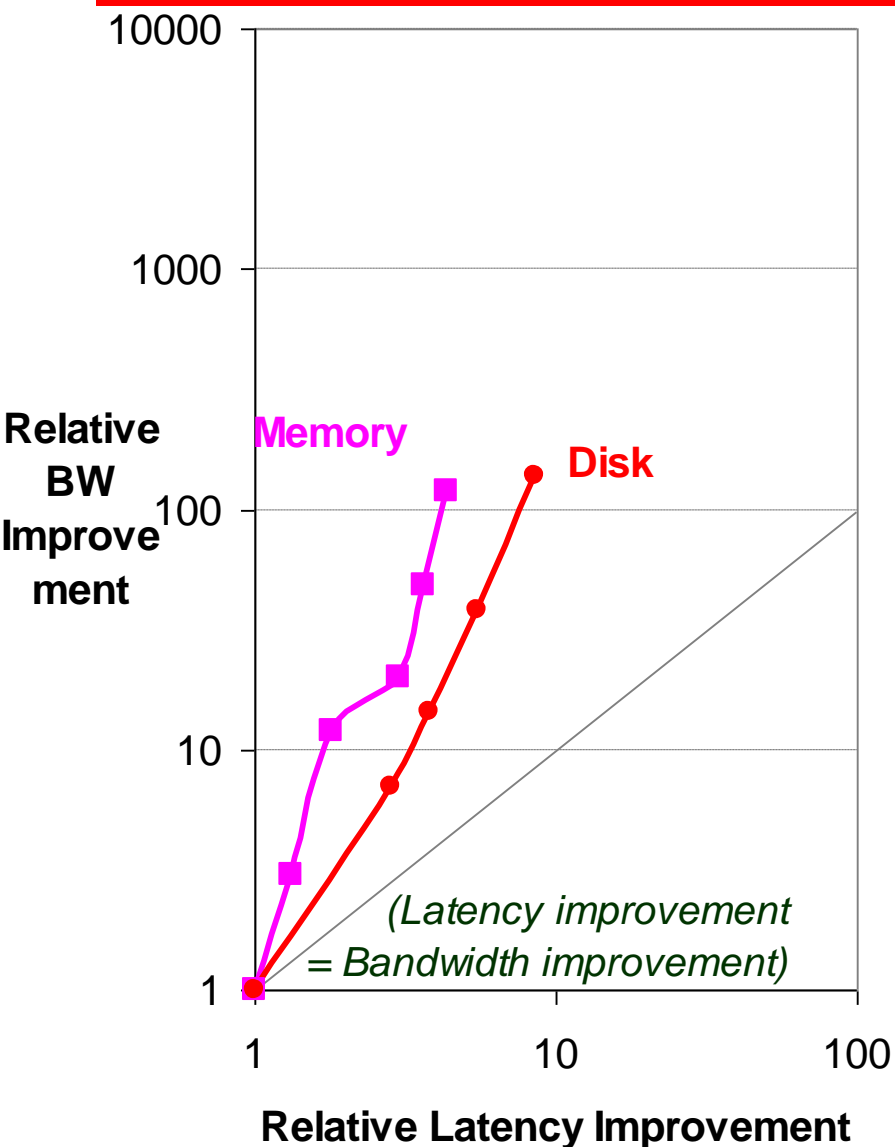
- Disk: 3600, 5400, 7200, 10000, 15000 RPM (8x, 143x)

(latency = simple operation w/o contention
BW = best-case)

Memory: Archaic (Nostalgic) v. Modern (Newfangled)

- 1980 DRAM (asynchronous)
- 0.06 Mbits/chip
- 64,000 xtors, 35 mm²
- 16-bit data bus per module, 16 pins/chip
- 13 Mbytes/sec
- Latency: 225 ns
- (no block transfer)
- 2000 Double Data Rate Synchr. (clocked) DRAM
- 256.00 Mbits/chip (4000X)
- 256,000,000 xtors, 204 mm²
- 64-bit data bus per DIMM, 66 pins/chip (4X)
- 1600 Mbytes/sec (120X)
- Latency: 52 ns (4X)
- Block transfers (page mode)

Latency Lags Bandwidth (last ~20 years)



- Performance Milestones

- Memory Module: 16bit plain DRAM, Page Mode DRAM, 32b, 64b, SDRAM, DDR SDRAM (4x,120x)
- Disk: 3600, 5400, 7200, 10000, 15000 RPM (8x, 143x)
(latency = simple operation w/o contention
BW = best-case)

LANs: Archaic (Nostalgic)v. Modern (Newfangled)

- Ethernet 802.3
- Year of Standard: 1978
- 10 Mbits/s link speed
- Latency: 3000 μ sec
- Shared media
- Coaxial cable

- Ethernet 802.3ae
- Year of Standard: 2003
- 10,000 Mbits/s link speed (1000X)
- Latency: 190 μ sec (15X)
- Switched media
- Category 5 copper wire

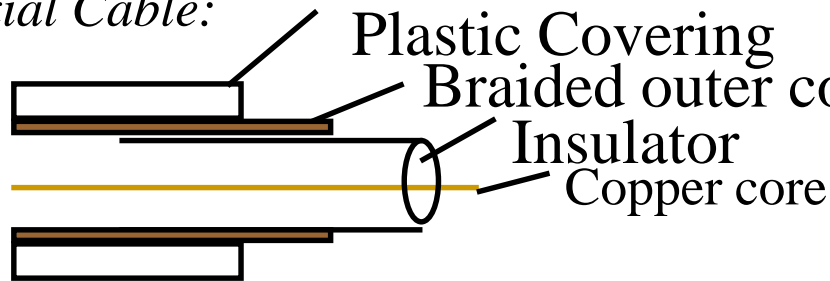
"Cat 5" is 4 twisted pairs in bundle

Twisted Pair:

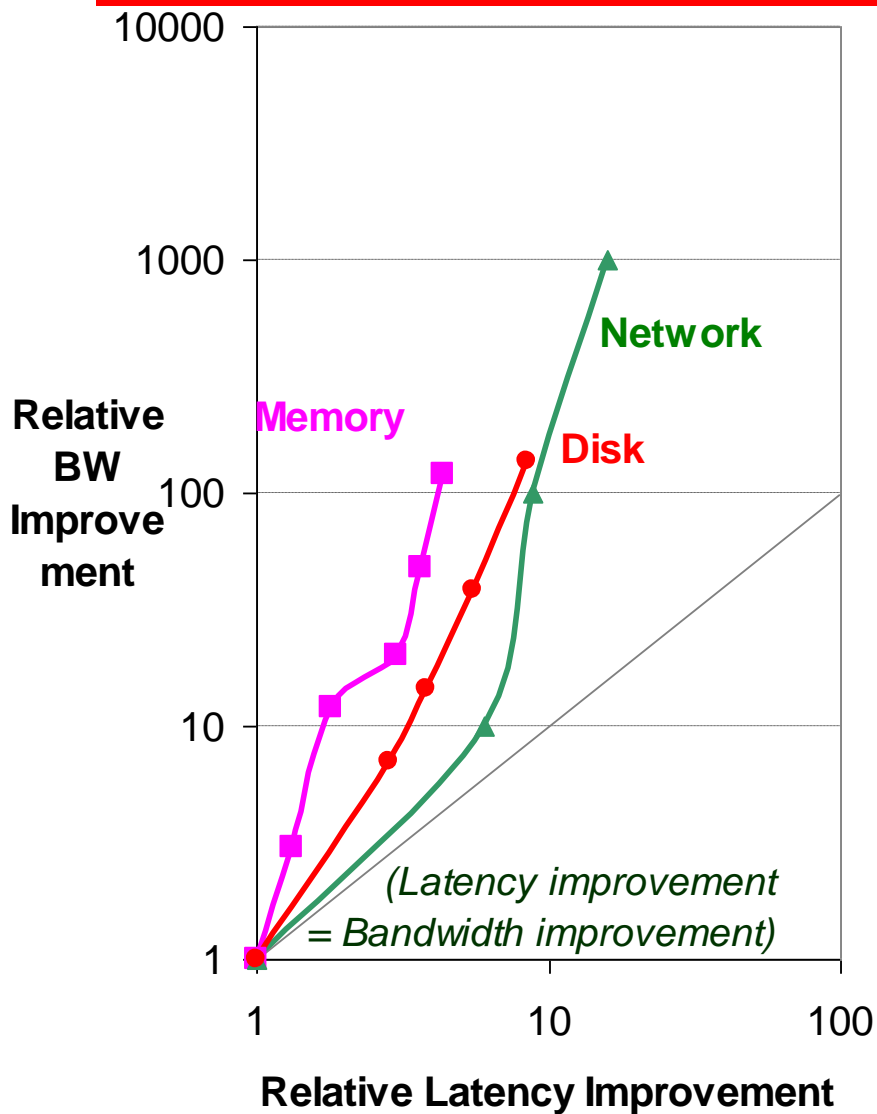


Copper, 1mm thick,
twisted to avoid antenna effect

Coaxial Cable:

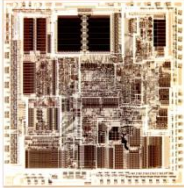



Latency Lags Bandwidth (last ~20 years)

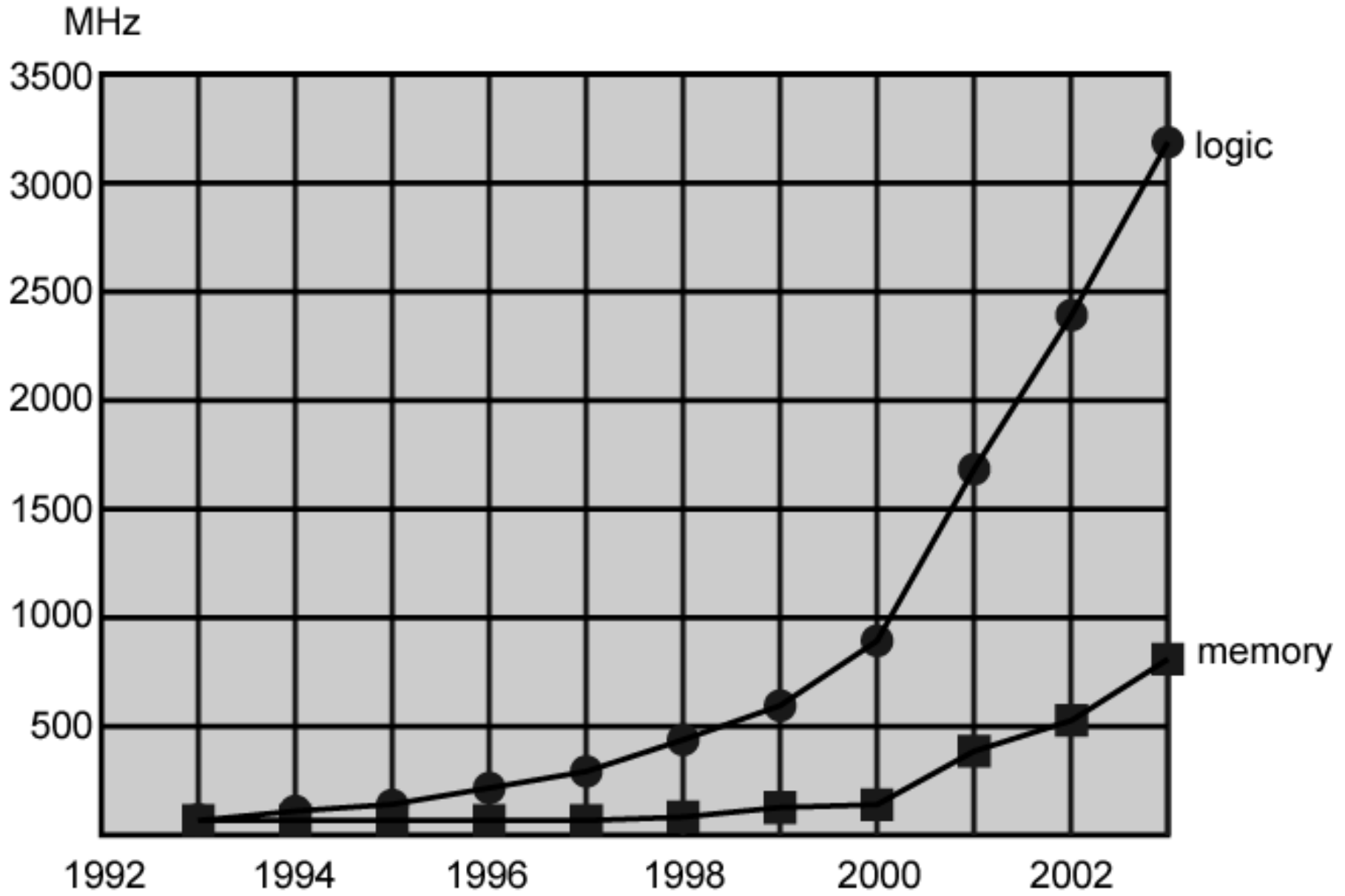


- Performance Milestones
- **Ethernet:** 10Mb, 100Mb, 1000Mb, 10000 Mb/s (16x,1000x)
- Memory Module: 16bit plain DRAM, Page Mode DRAM, 32b, 64b, SDRAM, DDR SDRAM (4x,120x)
- Disk: 3600, 5400, 7200, 10000, 15000 RPM (8x, 143x)
(latency = simple operation w/o contention
BW = best-case)

CPUs: Archaic (Nostalgic) v. Modern (Newfangled)

- 1982 Intel 80286
 - 12.5 MHz
 - 2 MIPS (peak)
 - Latency 320 ns
 - 134,000 xtors, 47 mm²
 - 16-bit data bus, 68 pins
 - Microcode interpreter, separate FPU chip
 - (no caches)
- 
- 
- 2001 Intel Pentium 4
 - 1500 MHz (120X)
 - 4500 MIPS (peak) (2250X)
 - Latency 15 ns (20X)
 - 42,000,000 xtors, 217 mm²
 - 64-bit data bus, 423 pins
 - 3-way superscalar, Dynamic translate to RISC, Superpipelined (22 stage), Out-of-Order execution
 - On-chip 8KB Data caches, 96KB Instr. Trace cache, 256KB L2 cache

Logic and Memory Performance Gap



Solutions

- Increase number of bits retrieved at one time
 - Make DRAM “wider” rather than “deeper”
- Change DRAM interface
 - Cache
- Reduce frequency of memory access
 - More complex cache and cache on chip
- Increase interconnection bandwidth
 - High speed buses
 - Hierarchy of buses

Rule of Thumb for Latency Lagging BW

- In the time that bandwidth doubles, latency improves by no more than a factor of 1.2 to 1.4
(and capacity improves faster than bandwidth)
- Stated alternatively:
Bandwidth improves by more than the square of the improvement in Latency

Improvements in Chip Organization and Architecture

- Increase hardware speed of processor
 - Fundamentally due to shrinking logic gate size
 - More gates, packed more tightly, increasing clock rate
 - Propagation time for signals reduced
- Increase size and speed of caches
 - Dedicating part of processor chip
 - Cache access times drop significantly
- Change processor organization and architecture
 - Increase effective speed of execution
 - Parallelism