

Today's topics

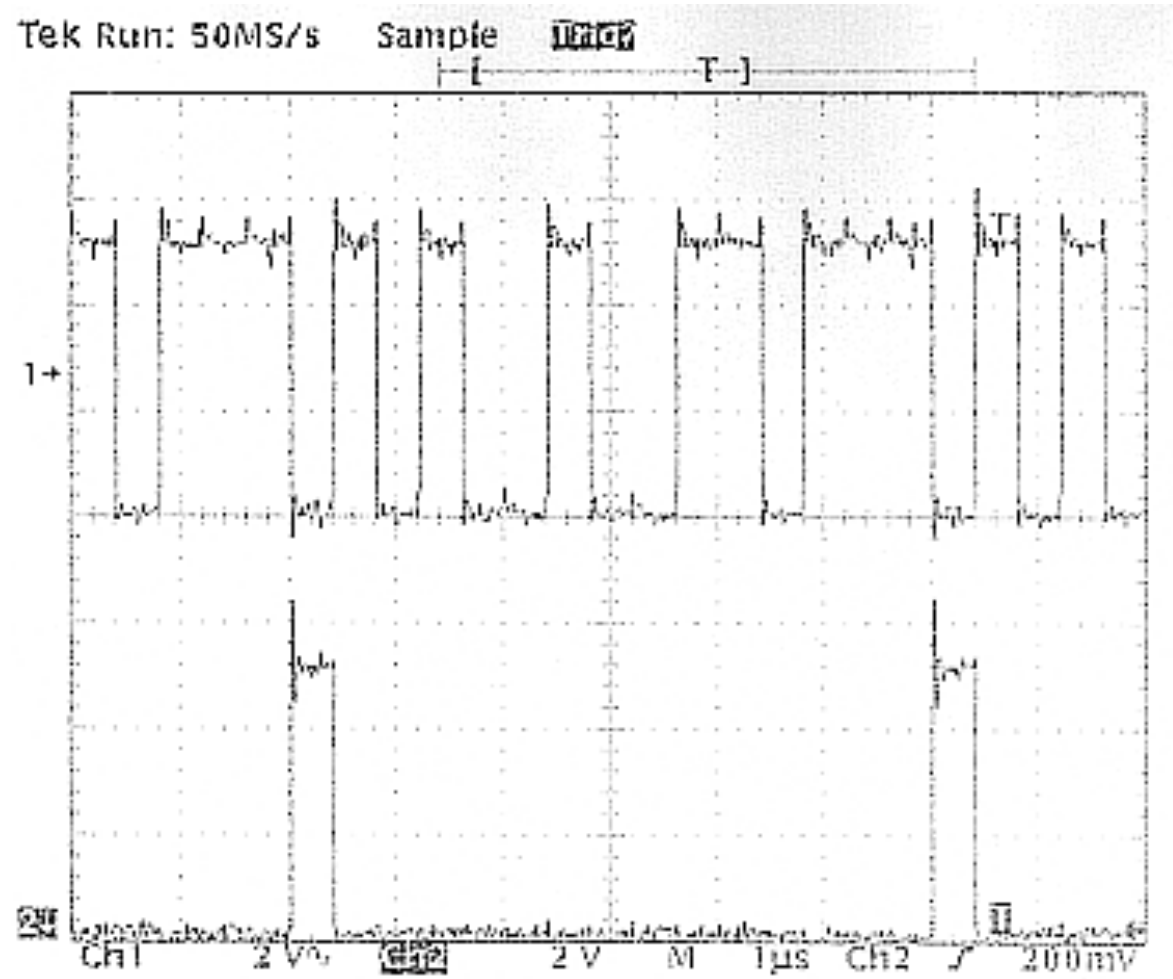
- **Binary Numbers**
 - Brookshear 1.1-1.6
- **Computer Architecture**
 - Notes from David A. Patterson and John L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, Morgan Kaufmann, 1997.
 - <http://computer.howstuffworks.com/pc.htm>
- **Slides from Prof. Marti Hearst of UC Berkeley SIMS**
- **Upcoming**
 - **Operating Systems**
 - Brookshear 3.1-3.4
 - **Security**
 - GI, 11 & Brookshear 3.7

Digital Computers

- **What are computers made up of?**
 - **Lowest level of abstraction: atoms**
 - **Higher level: transistors**
- **Transistors**
 - **Invented in 1951 at Bell Labs**
 - **An electronic switch**
 - **Building block for all modern electronics**
 - **Transistors are packaged as Integrated Circuits (ICs)**
 - **40 million transistors in 1 IC**

Binary Digits (Bits)

- Yes or No
- On or Off
- One or Zero
- 10010010



More on binary

- **Byte**
 - A sequence of bits
 - 8 bits = 1 byte
 - 2 bytes = 1 word (sometimes 4 or 8 bytes)
- **Powers of two**
- **How do binary numbers work?**

Decimal (Base 10) Numbers

- Each digit in a decimal number is chosen from **ten** symbols:
 $\{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \}$
- The position (right to left) of each digit represents a power of **ten**.
- *Example:* Consider the decimal number **2307**

		2	3	0	7
		↑	↑	↑	↑
position:	3	2	1	0	

$$2307 = 2 \times 10^3 + 3 \times 10^2 + 0 \times 10^1 + 7 \times 10^0$$

Binary (Base 2) Numbers

- Each digit in a binary number is chosen from **two** symbols:
 $\{ 0, 1 \}$
- The position (right to left) of each digit represents a power of **two**.
- *Example:* Convert binary number **1101** to decimal

	1	1	0	1
	↑	↑	↑	↑
position:	3	2	1	0

$$\begin{aligned} \mathbf{1101} &= \mathbf{1} \times 2^3 + \mathbf{1} \times 2^2 + \mathbf{0} \times 2^1 + \mathbf{1} \times 2^0 \\ &= \mathbf{1} \times 8 + \mathbf{1} \times 4 + \mathbf{0} \times 2 + \mathbf{1} \times 1 = 8 + 4 + 1 = \mathbf{13} \end{aligned}$$

Powers of Two

Decimal	Binary	Power of 2
1	1	2^0
2	10	2^1
4	100	2^2
8	1000	2^3
16	10000	2^4
32	100000	2^5
64	1000000	2^6
128	10000000	2^7

Famous Powers of Two

Kilobyte (KB)	1024 or 2^{10} bytes	1,024 bytes	Thousands of bytes
Megabyte (MB)	1024^2 or 2^{20} bytes	1,048,578 bytes	Millions of bytes
Gigabyte (GB)	1024^3 or 2^{30} bytes	1,073,741,824 bytes	Billions of bytes
Terabyte (TB)	1024^4 or 2^{40} bytes	1,099,511,627,776 bytes	Trillions of bytes

Other Number Systems

Binary	Octal	Decimal	Hexadecimal
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
<i>Base-2</i>	<i>Base-8</i>	<i>Base-10</i>	<i>Base-16</i>

Binary Addition

Rule 1	Rule 2	Rule 3	Rule 4
$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array}$	$\begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array}$	$\begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array}$	$\begin{array}{r} 1 \\ + 1 \\ \hline 10 \end{array}$

Also: $1 + 1 + 1 = 1$ with a carry of 1

Adding Binary Numbers

$$\begin{array}{r} 101 \\ + 10 \\ \hline 111 \end{array}$$

- $101 + 10 = (1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0) + (1 \times 2^1 + 0 \times 2^0)$
 $= (1 \times 4 + 0 \times 2 + 1 \times 1) + (1 \times 2 + 0 \times 1)$
- Add like terms: There is one 4, one 2, one 1
 $= 1 \times 4 + 1 \times 2 + 1 \times 1 = 111$

Adding Binary Numbers

$$\begin{array}{r} 11 \quad \leftarrow \text{carry} \\ 111 \\ + 110 \\ \hline 1101 \end{array}$$

- $111 + 110 = (1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0) + (1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0)$
 $= (1 \times 4 + 1 \times 2 + 1 \times 1) + (1 \times 4 + 1 \times 2 + 0 \times 1)$
- Add like terms: There are two 4s, two 2s, one 1
 $= 2 \times 4 + 2 \times 2 + 1 \times 1$
 $= 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 1101$
- BinaryNumber Applet

Converting Decimal to Binary

Decimal

→ → conversion → →

Binary

0

$$0 = 0 \times 2^0$$

0

1

$$1 = 1 \times 2^0$$

1

2

$$2 = 1 \times 2^1 + 0 \times 2^0$$

10

3

$$3 = 2 + 1 = 1 \times 2^1 + 0 \times 2^0$$

11

4

$$4 = 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

100

5

$$5 = 4 + 1 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

101

6

$$6 = 4 + 2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

110

7

$$7 = 4 + 2 + 1 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

111

8

$$8 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

1000

Converting Decimal to Binary

- Repeated division by **two** until the quotient is zero
- **Example:** Convert decimal number **54** to binary

$$\begin{array}{r} 0 \rightarrow 1 \\ 2 \overline{)1} \rightarrow 1 \\ 2 \overline{)3} \rightarrow 0 \\ 2 \overline{)6} \rightarrow 1 \\ 2 \overline{)13} \rightarrow 1 \\ 2 \overline{)27} \rightarrow 0 \\ 2 \overline{)54} \quad \text{remainder} \end{array}$$

Binary representation of
54 is **110110**

Converting Decimal to Binary

$2 \overline{)0}$	$\rightarrow 1$	● 1 32 = 0	plus 1 thirty-two
$2 \overline{)1}$	$\rightarrow 1$	● 6 8s = 1 32	plus 1 sixteen
$2 \overline{)3}$	$\rightarrow 0$	● 3 16s = 3 16	plus 0 eights
$2 \overline{)6}$	$\rightarrow 1$	● 13 4s = 6 8s	plus 1 four
$2 \overline{)13}$	$\rightarrow 1$	● 27 2s = 13 4s	plus 1 two
$2 \overline{)27}$	$\rightarrow 0$	● 54 = 27 2s	plus 0 ones
$2 \overline{)54}$			

- Subtracting highest power of two
- 1s in positions 5,4,2,1

$$54 - 2^5 = 22$$

$$22 - 2^4 = 6 \quad \rightarrow 110110$$

$$6 - 2^2 = 2$$

$$2 - 2^1 = 0$$

Problems

- **Convert 1011000 to decimal representation**

- **Add the binary numbers 1011001 and 10101 and express their sum in binary representation**

- **Convert 77 to binary representation**

Solutions

- Convert 1011000 to decimal representation

$$\begin{aligned}1011000 &= 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\ &= 1 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 0 \times 4 + 0 \times 2 + 0 \times 1 \\ &= 64 + 16 + 8 = 88\end{aligned}$$

- Add the binary numbers 1011001 and 10101 and express their sum in binary representation

$$\begin{array}{r}1011001 \\ + 10101 \\ \hline 1101110\end{array}$$

Solutions

- Convert 77 to binary representation

$$\begin{array}{r} 0 \rightarrow 1 \\ 2 \overline{)1} \rightarrow 0 \\ 2 \overline{)2} \rightarrow 0 \\ 2 \overline{)4} \rightarrow 1 \\ 2 \overline{)9} \rightarrow 1 \\ 2 \overline{)19} \rightarrow 0 \\ 2 \overline{)38} \rightarrow 1 \\ 2 \overline{)77} \end{array}$$

Binary representation of
77 is **1001101**

Boolean Logic

- **AND, OR, NOT, NOR, NAND, XOR**
- **Each operator has a set of rules for combining two binary inputs**
 - **These rules are defined in a Truth Table**
 - **(This term is from the field of Logic)**
- **Each implemented in an electronic device called a gate**
 - **Gates operate on inputs of 0's and 1's**
 - **These are more basic than operations like addition**
 - **Gates are used to build up circuits that**
 - **Compute addition, subtraction, etc**
 - **Store values to be used later**
 - **Translate values from one format to another**

Truth Tables

AND gate



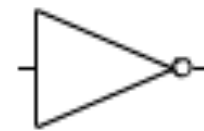
Input		Output
0	0	0
0	1	0
1	0	0
1	1	1

OR gate



Input		Output
0	0	0
0	1	1
1	0	1
1	1	1

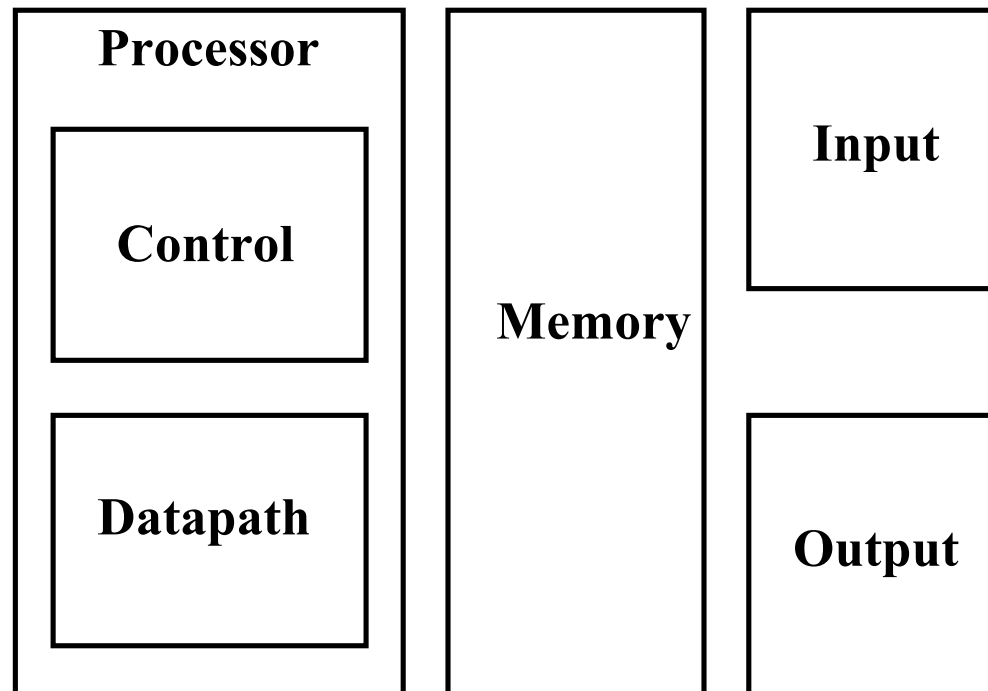
NOT gate



Input	Output
0	1
1	0

The Big Picture

- Since 1946 all computers have had 5 components
 - The Von Neumann Machine



- **What is computer architecture?**
Computer Architecture = Machine Organization +
Instruction Set Architecture + ...

Fetch, Decode, Execute Cycle

- Computer instructions are stored (as bits) in memory
- A program's execution is a loop
 - Fetch instruction from memory
 - Decode instruction
 - Execute instruction
- Cycle time
 - Measured in hertz (cycles per second)
 - 2 GHz processor can execute this cycle up to 2 billion times a second
 - Not all cycles are the same though...

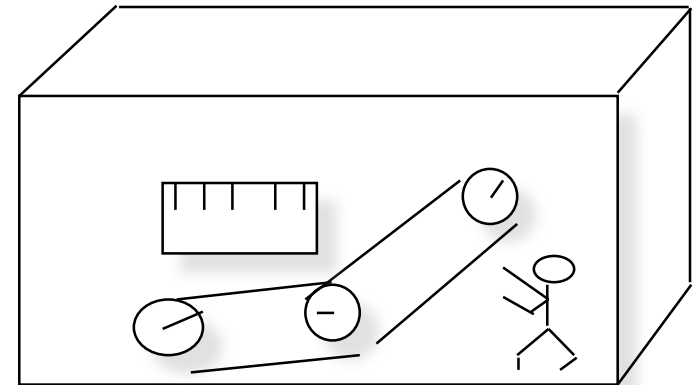
Organization

- Capabilities & Performance Characteristics of Principal Functional Units (Fus)
 - (e.g., Registers, ALU, Shifters, Logic Units, ...)
- Ways in which these components are interconnected
- Information flows between components
- Logic and means by which such information flow is controlled.
- Choreography of FUs to realize the ISA

Logic Designer's View

ISA Level

FUs & Interconnect



Memory bottleneck

- CPU can execute dozens of instructions in the time it takes to retrieve one item from memory
- **Solution: Memory Hierarchy**
 - Use fast memory
 - Registers
 - Cache memory
 - Rule: small memory is fast, large memory is slow

What is Realtime?

- **Response time**
 - **Panic**
 - How to tell “I am still computing”
 - Progress bar
- **Flicker**
 - **Fusion frequency**
- **Update rate vs. refresh rate**
 - **Movie film standards (24 fps projected at 48 fps)**
- **Interactive media**
 - **Interactive vs. non-interactive graphics**
 - computer games vs. movies
 - animation tools vs. animation
 - **Interactivity => real-time systems**
 - system must respond to user inputs without any perceptible delay
(A Primary Challenge in VR)

A great idea in computer science

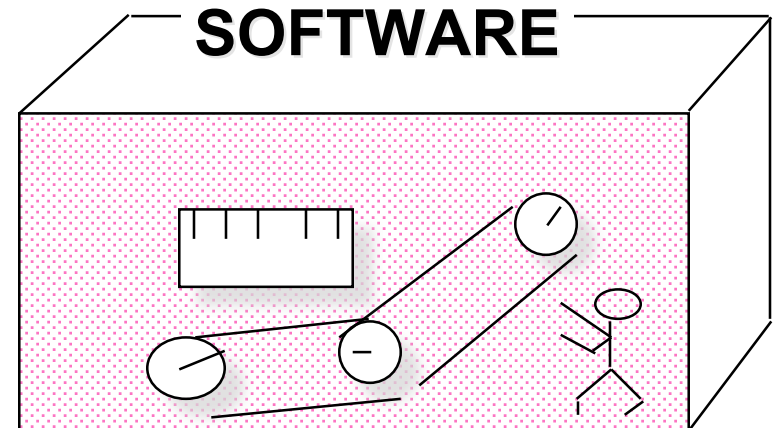
- **Temporal locality**
 - Programs tend to access data that has been accessed recently (i.e. close in *time*)
- **Spatial locality**
 - Programs tend to access data at an address near recently referenced data (i.e. close in *space*)
- **Useful in graphics and virtual reality as well**
 - Realistic images require significant computational power
 - Don't need to represent distant objects as well
- **Efficient distributed systems rely on locality**
 - Memory access time increases over a network
 - Want to access data on local machine

Instruction Set Architecture

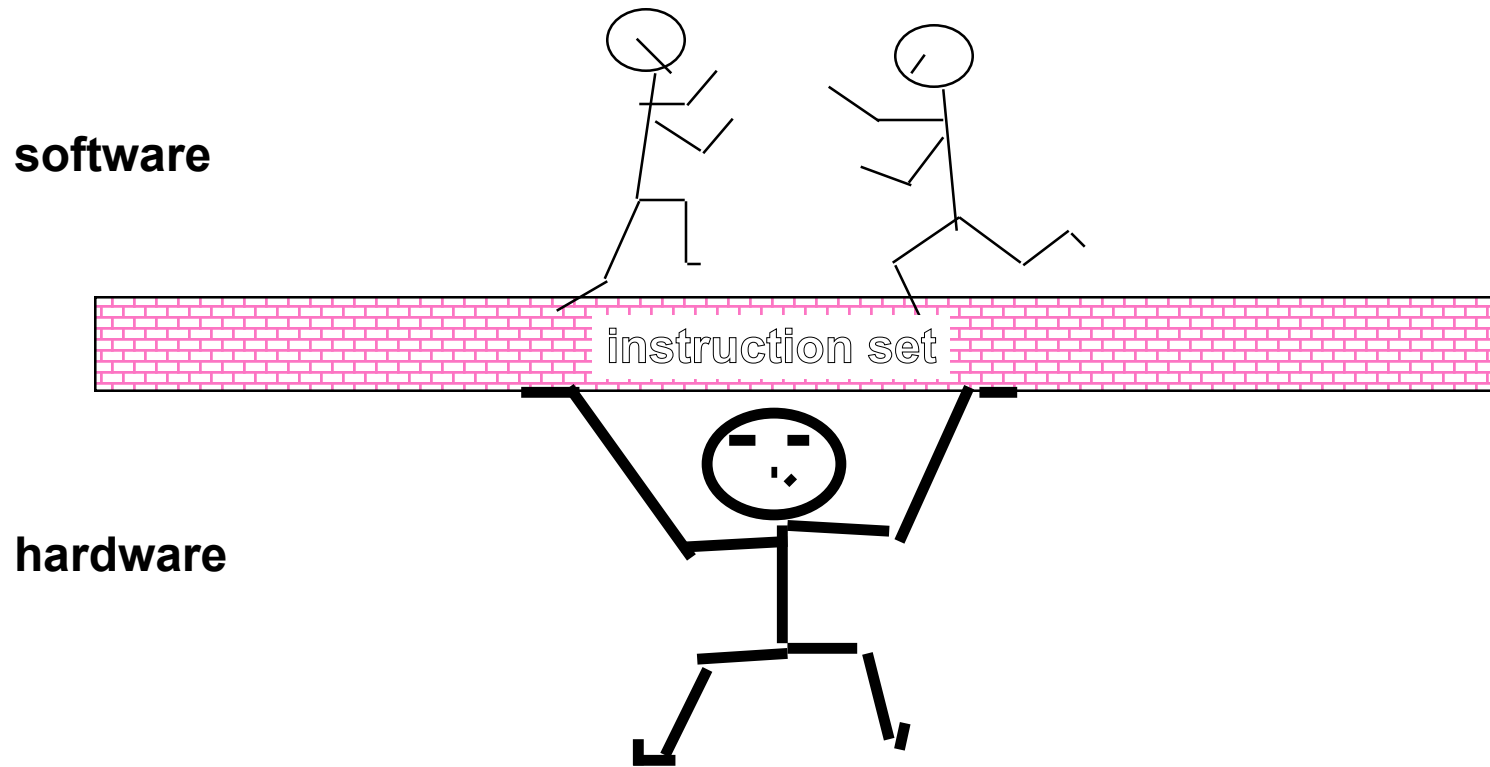
... the attributes of a [computing] system as seen by the programmer, *i.e.* the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls the logic design, and the physical implementation.

- 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

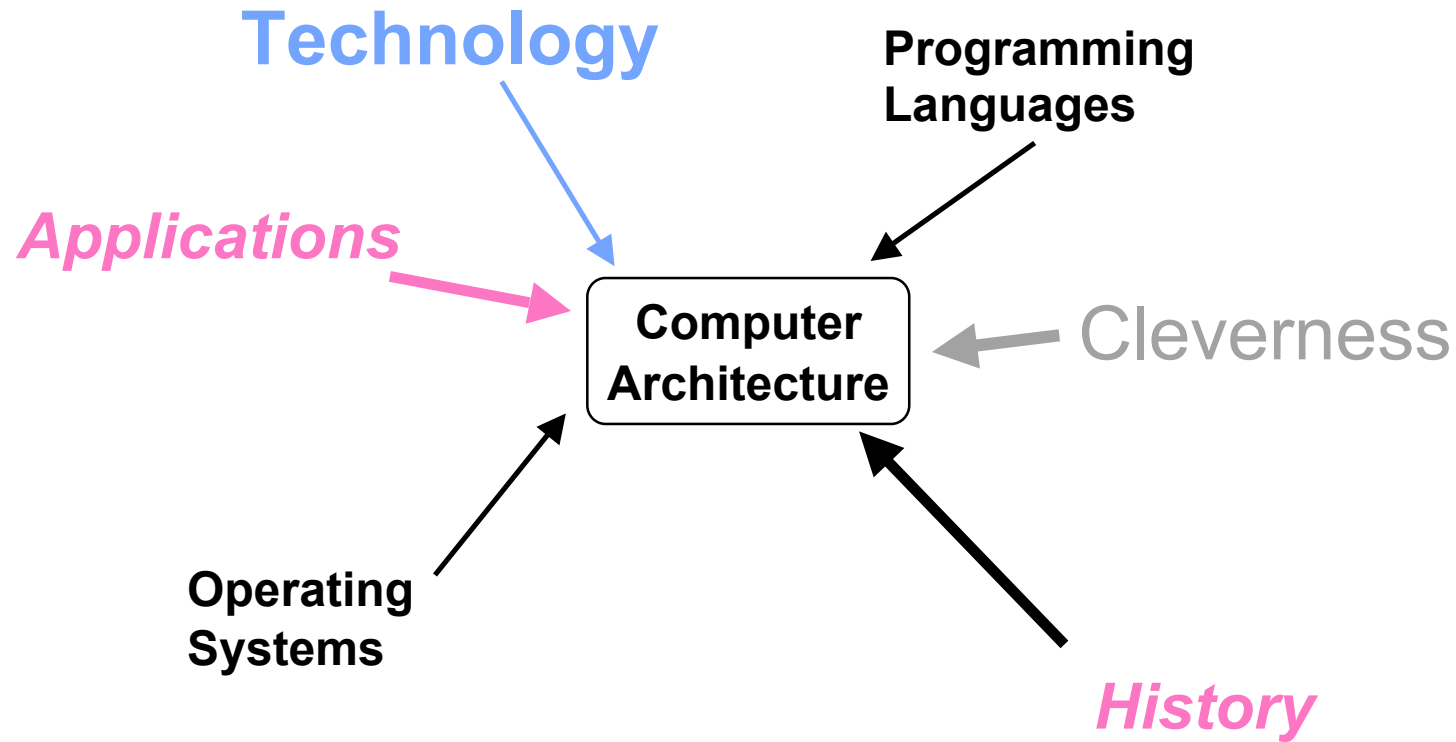


The Instruction Set: a Critical Interface



- What is an example of an Instruction Set architecture?

Forces on Computer Architecture

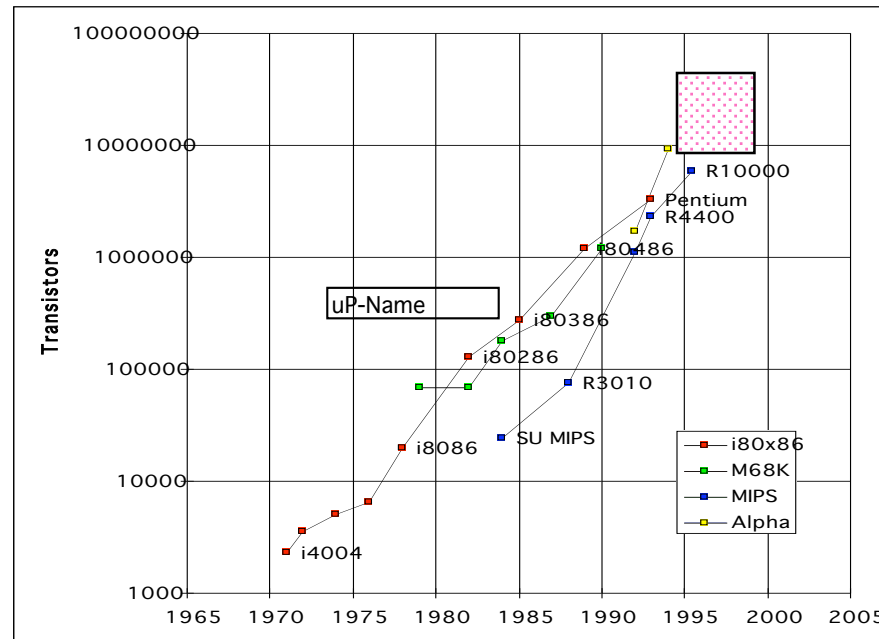


Technology

DRAM chip capacity

DRAM	
Year	Size
1980	64 Kb
1983	256 Kb
1986	1 Mb
1989	4 Mb
1992	16 Mb
1996	64 Mb
1999	256 Mb
2002	1 Gb
2004	4 Gb

Microprocessor Logic Density



- In ~1985 the single-chip processor (32-bit) and the single-board computer emerged
 - => workstations, personal computers, multiprocessors have been riding this wave since
- Now, we have multicore processors

Technology => dramatic change

- **Processor**

- logic capacity: about 30% per year
- clock rate: about 20% per year

- **Memory**

- DRAM capacity: about 60% per year (4x every 3 years)
- Memory speed: about 10% per year
- Cost per bit: improves about 25% per year

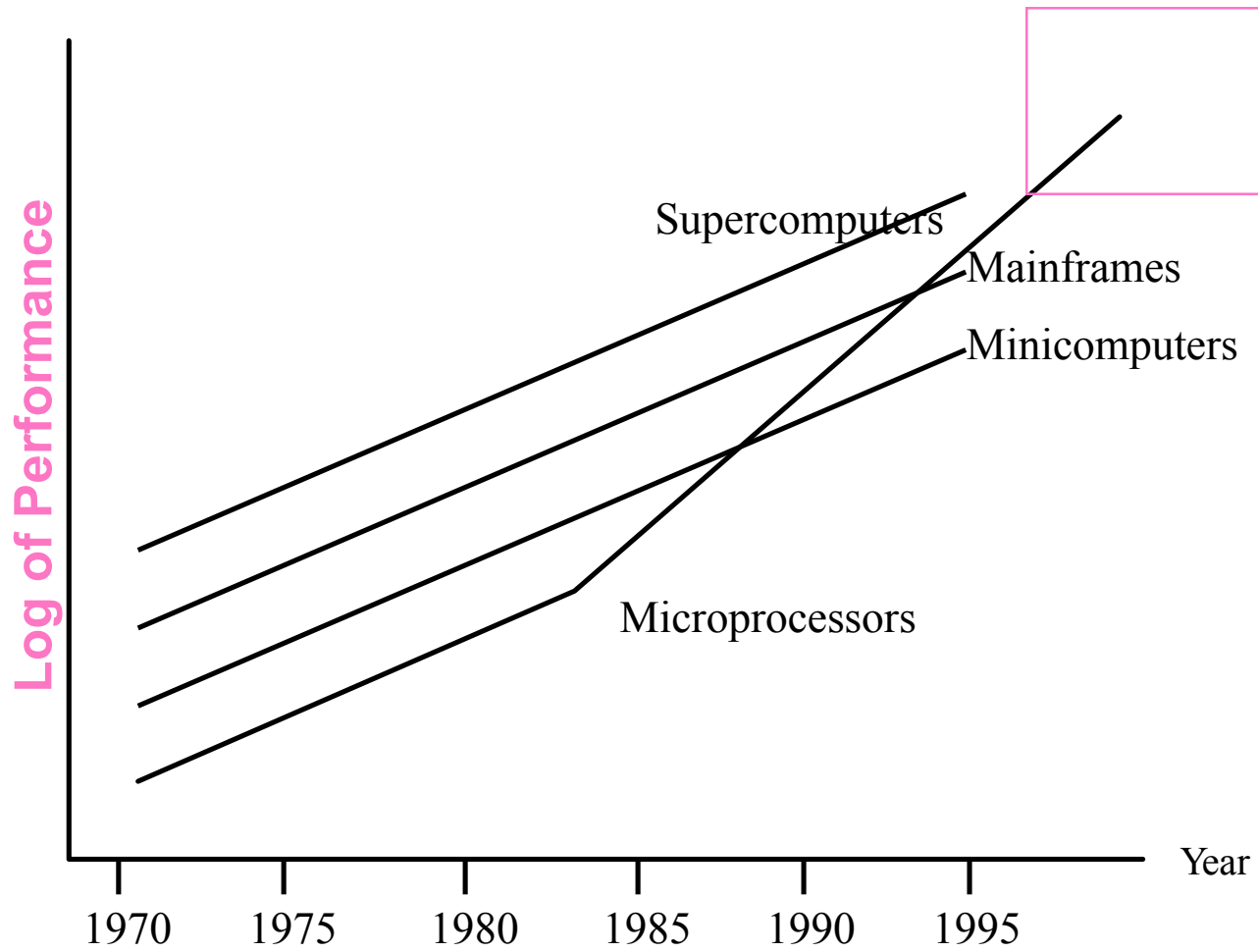
- **Disk**

- capacity: about 60% per year
- Total use of data: 100% per 9 months!

- **Network Bandwidth**

- Bandwidth increasing more than 100% per year!

Performance Trends



Laws?

- Define each of the following. What has its effect been on the advancement of computing technology?
 - Moore's Law
 - Amdahl's Law
 - Metcalfe's Law