ECE 361
Computer Architecture
Lecture 1

Prof. Alok N. Choudhary

choudhar@ece.northwestern.edu

Today's Lecture

Computer Design

- Levels of abstraction
- Instruction sets and computer architecture

Architecture design process

Interfaces

Course Structure

Technology as an architectural driver

- Evolution of semiconductor and magnetic disk technology
- New technologies replace old
- Industry disruption

Break

Cost and Price

- Semiconductor economics

Computers, Levels of Abstraction and Architecture

Computer Architecture's Changing Definition

1950s Computer Architecture

- Computer Arithmetic

1960s

- Operating system support, especially memory management

1970s to mid 1980s Computer Architecture

- Instruction Set Design, especially ISA appropriate for compilers
- Vector processing and shared memory multiprocessors

1990s Computer Architecture

- Design of CPU, memory system, I/O system, Multi-processors, Networks
- Design for VLSI

2000s Computer Architecture:

- Special purpose architectures, Functionally reconfigurable, Special considerations for low power/mobile processing, highly parallel structures

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

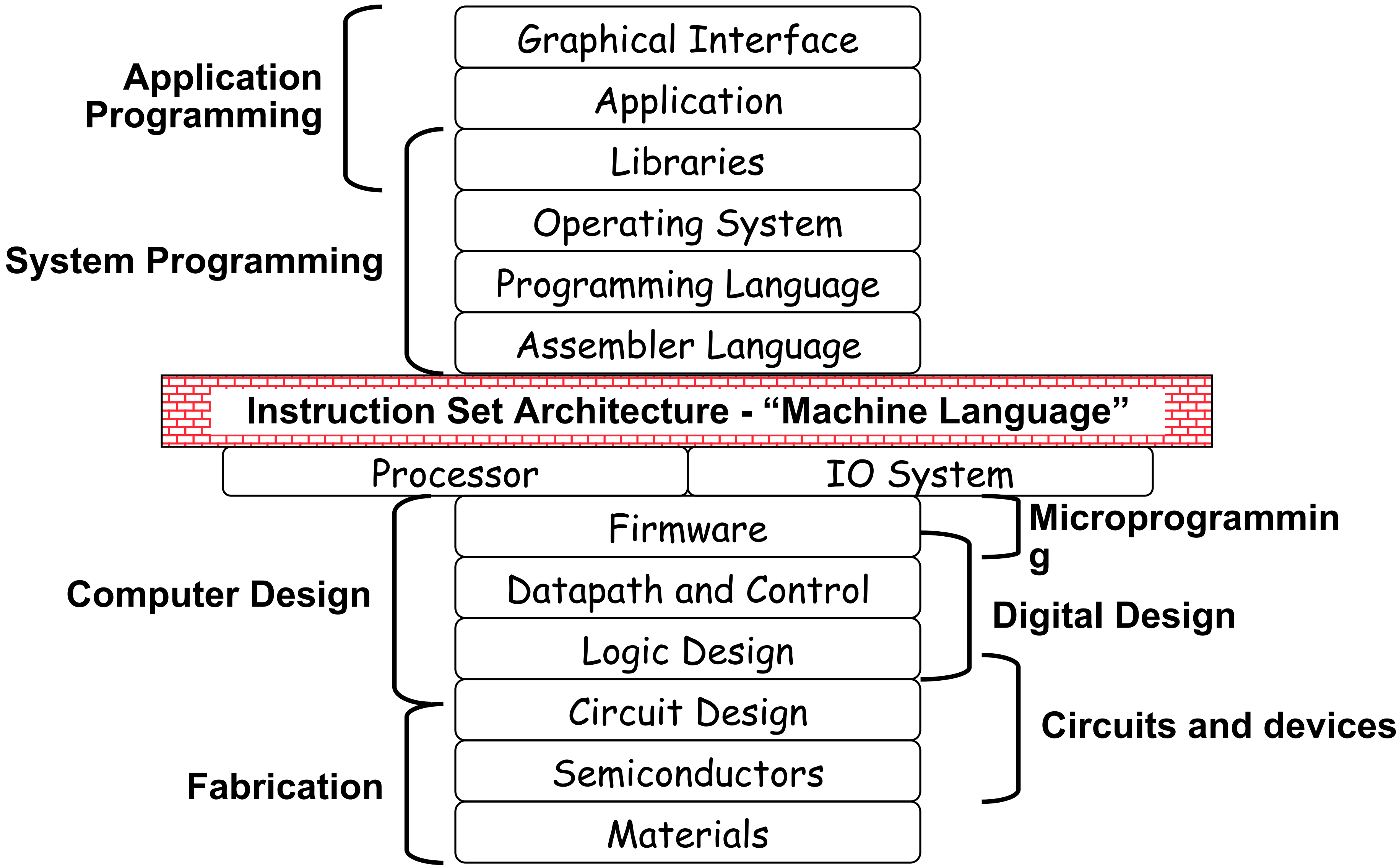
```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

Machine Interpretation

Control Signal Spec

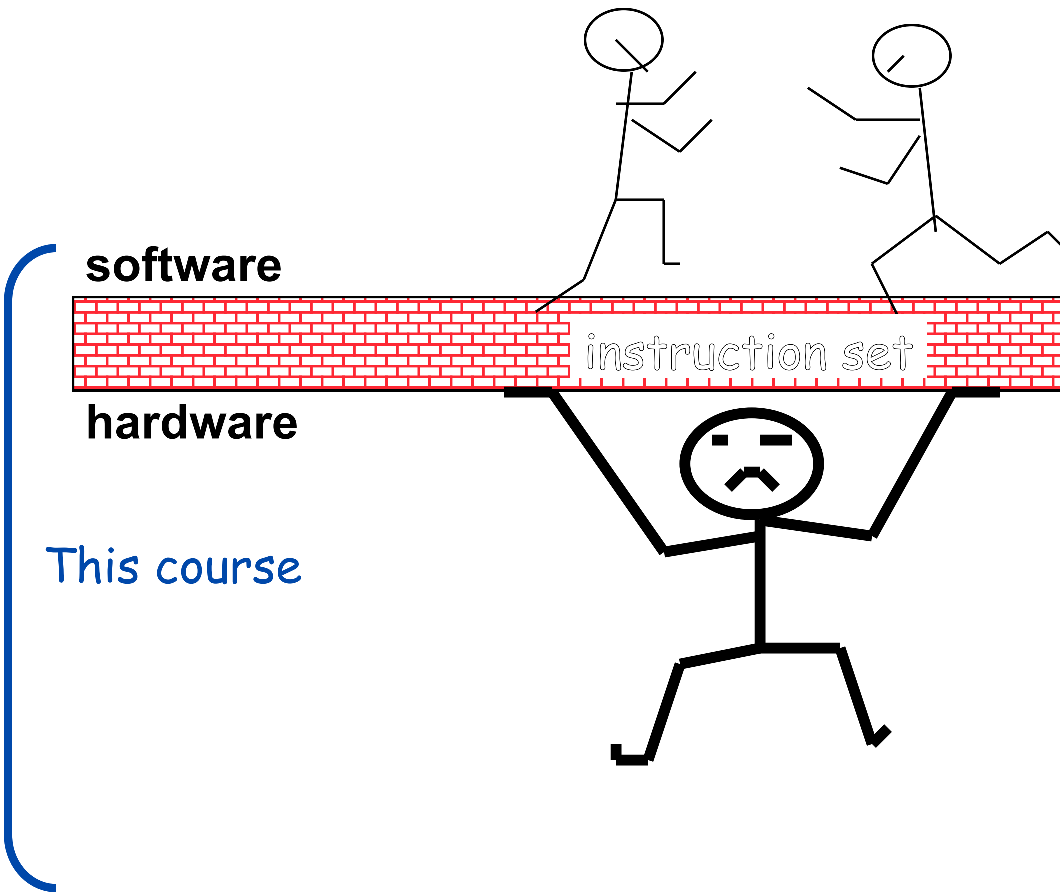
```
ALUOP[0:3] <= InstReg[9:11] & MASK
```

Levels of Abstraction



The Instruction Set: A Critical Interface

Computer Architecture =
Instruction Set Architecture +
Machine Organization



Instruction Set Design

- Machine Language
- Compiler View
- "Computer Architecture"
- "Instruction Set Architecture"

"Building Architect"

Computer Organization and Design

- Machine Implementation
- Logic Designer's View
- "Processor Architecture"
- "Computer Organization"

"Construction Engineer"

Instruction Set Architecture

Data Types
- Encoding and representation

Memory Model

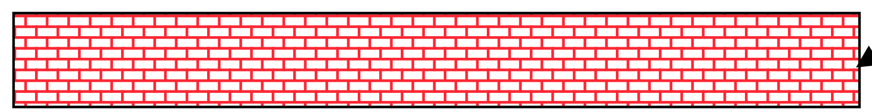
Program Visible Processor State
- General registers
- Program counter
- Processor status

Instruction Set
- Instructions and formats
- Addressing modes
- Data structures

System Model
- States
- Privilege
- Interrupts
- IO

External Interfaces
- IO
- Management

Architecture Reference Manual
Principles of Operation
Programming Guide
…



. . . 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

Computer Organization

Capabilities & Performance Characteristics of Principal Functional Units

(e.g., Registers, ALU, Shifters, Memory Management, etc.

Ways in which these components are interconnected

- Datapath - nature of information flows and connection of functional units
- Control - logic and means by which such information flow is controlled

Choreography of functional units to realize the ISA

Register Transfer Level Description / Microcode

"Hardware" designer's view includes logic and firmware

This Course Focuses on General Purpose Processors

A general-purpose computer system

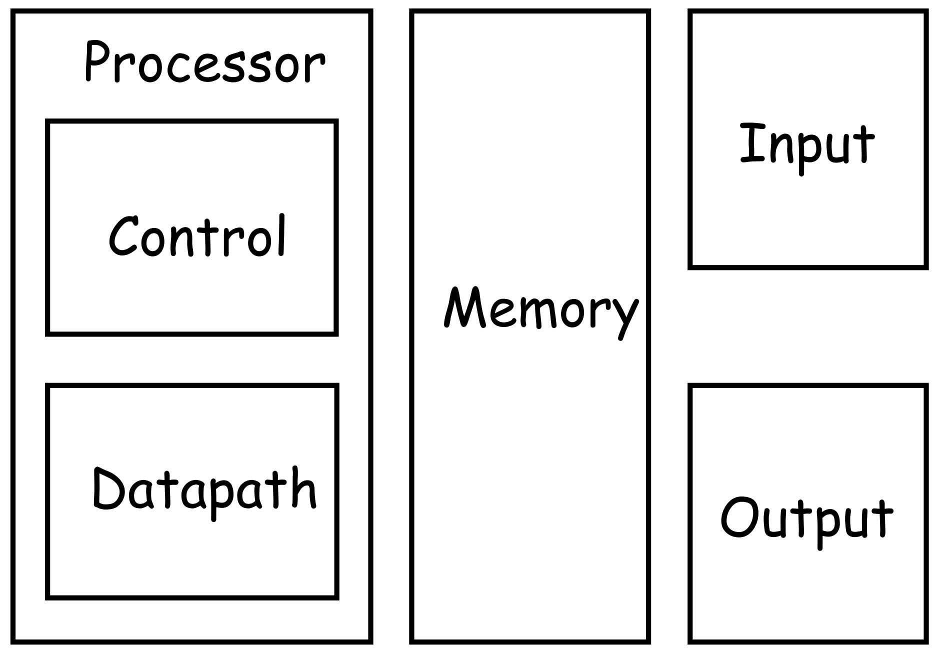
- Uses a programmable processor
- Can run “any” application
- Potentially optimized for some class of applications
- Common names: CPU, DSP, NPU, microcontroller, microprocessor

MIT Whirlwind, 1951

Unified main memory

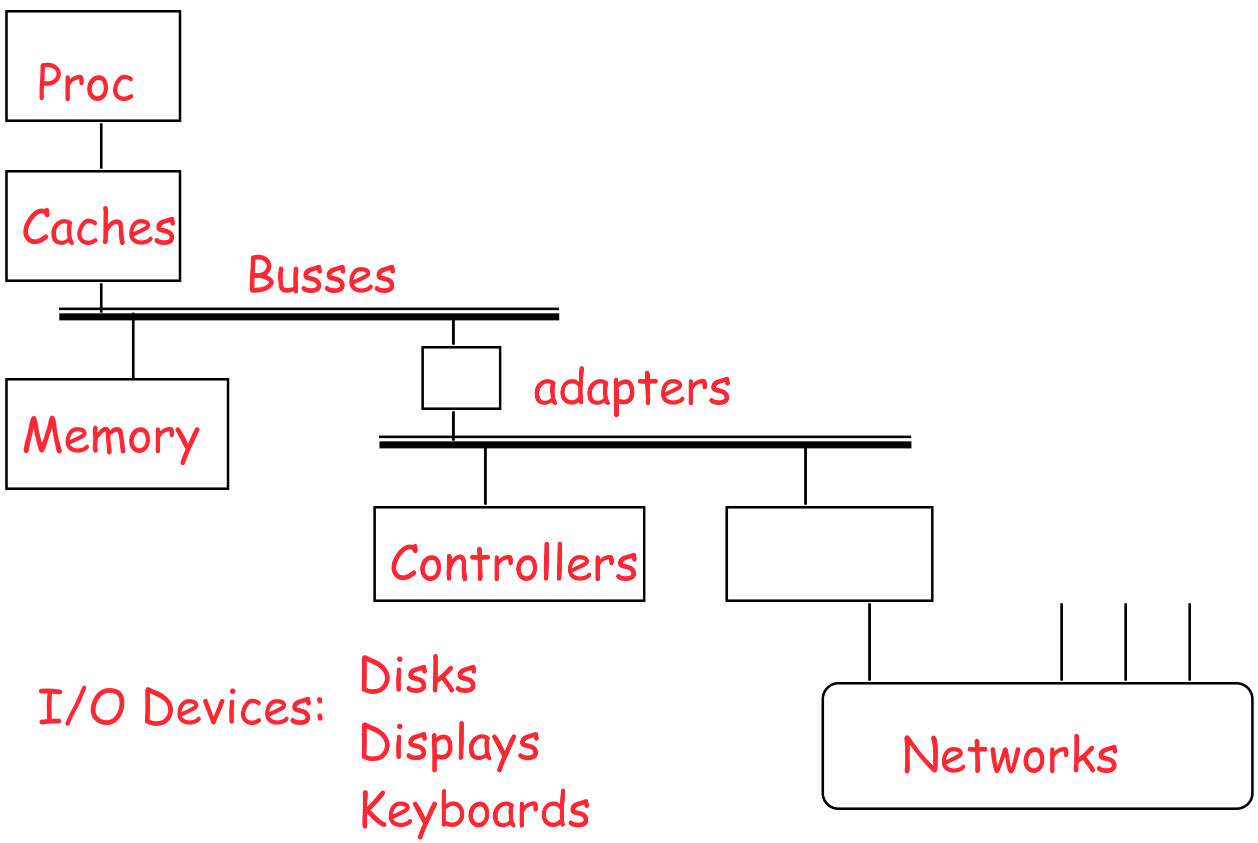
- For both programs & data
- Von Neumann computer

Busses & controllers to connect processor, memory, IO devices



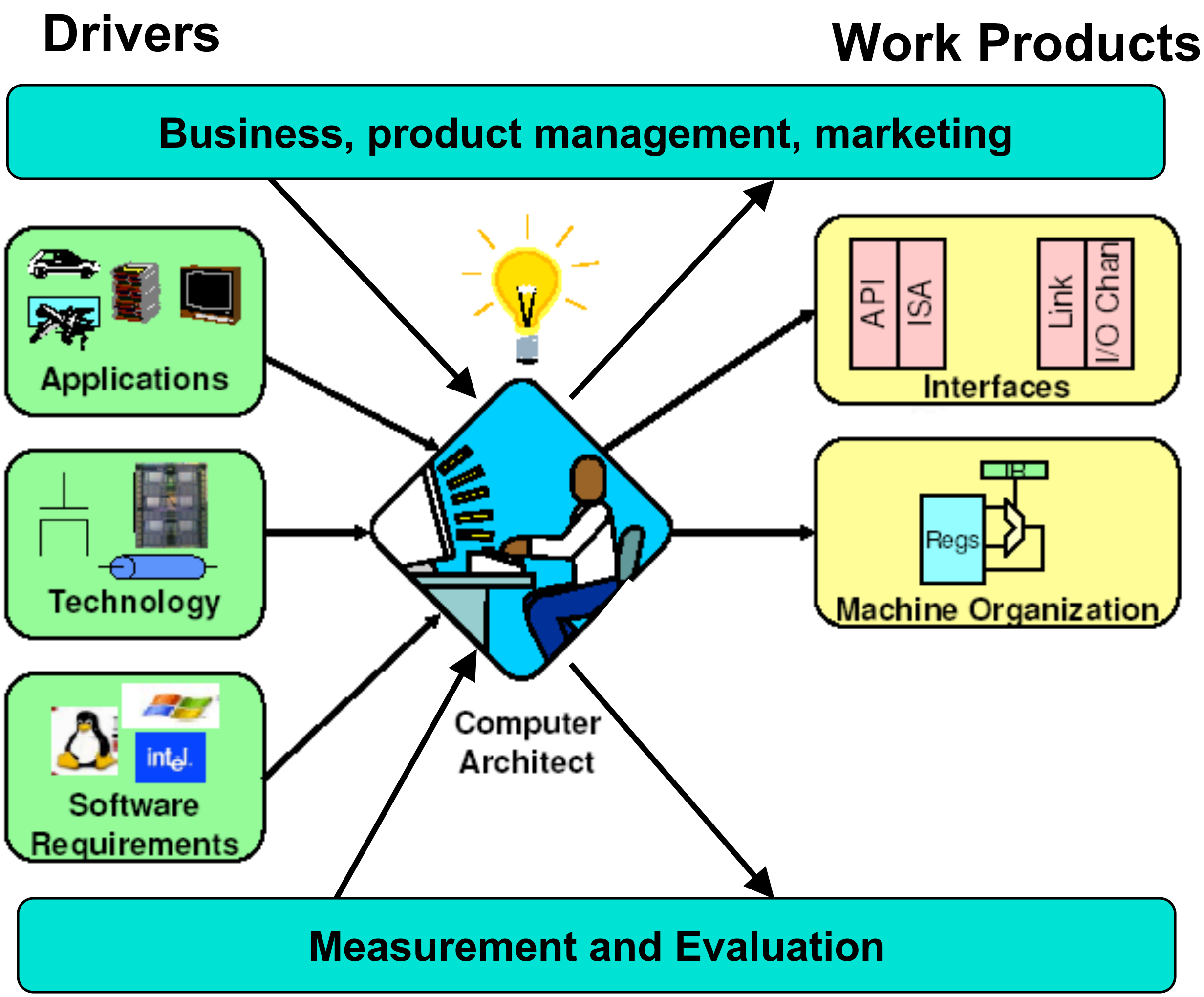
Computers are pervasive – servers, standalone PCs, network processors, embedded processors, …

Today, "Computers" are Connected Processors



All have interfaces & organizations

What does a computer architect do?



Translates business and technology drives into efficient systems for computing tasks.

Metrics of Efficiency - Examples

Desktop computing

- Examples: PCs, workstations
- Metrics: performance (latency), cost, time to market

Server computing

- Examples: web servers, transaction servers, file servers
- Metrics: performance (throughput), reliability, scalability

Embedded computing

- Examples: microwave, printer, cell phone, video console
- Metrics: performance (real-time), cost, power consumption, complexity

Applications Drive Design Points

Numerical simulations

- Floating-point performance
- Main memory bandwidth

Transaction processing

- I/Os per second and memory bandwidth
- Integer CPU performance

Media processing

- Repeated low-precision ‘pixel’ arithmetic
- Multiply-accumulate rates
- Bit manipulation

Embedded control

- I/O timing
- Real-time behavior

Architecture decisions will often exploit application behavior

Characteristics of a Good Interface Design

Well defined for users and implementers

Interoperability (Hardware) / Compatibility (Software)

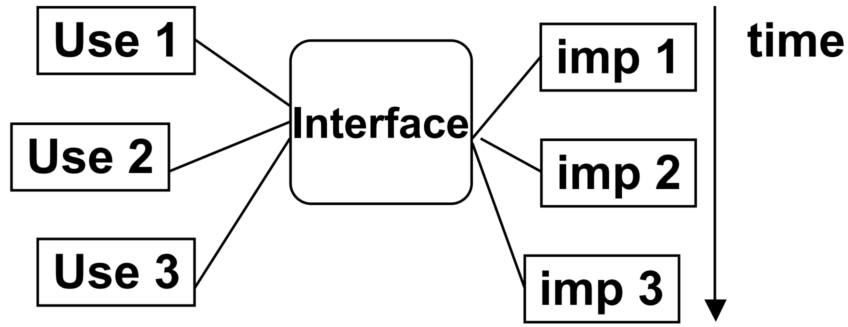
- Lasts through multiple implementations across multiple technologies (portability, compatibility)
- Efficiently supports multiple implementations
  - Competitive market
  - Compatible at multiple cost / performance design points

IP Investment Preservation

- Extensible function grows from a stable base
- Generality of application permits reuse of training, tools and implementations

Applies to many types of interfaces

- Instruction set architectures
- Busses
- Network protocols
- Library definitions
- OS service calls
- Programming languages



Interface usage can far exceed the most optimistic projections of it's designer:

- Instruction sets
  - S/360 1964 ~ present
  - X86 1972 ~ present
  - SPARC 1981 ~ present
- Network protocols
  - Ethernet 1973 ~ present
  - TCP/IP 1974 ~ present
- Programming languages
  - C 1973 ~ present

Course Structure

What You Need to Know from prerequisites

Basic machine structure

- Processor, memory, I/O

Assembly language programming

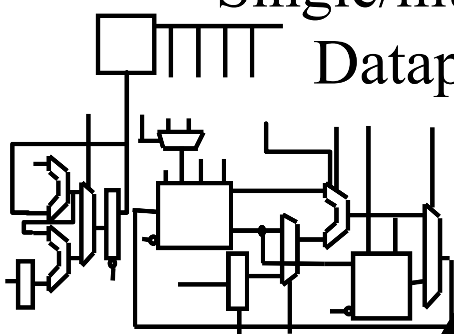
Simple operating system concepts

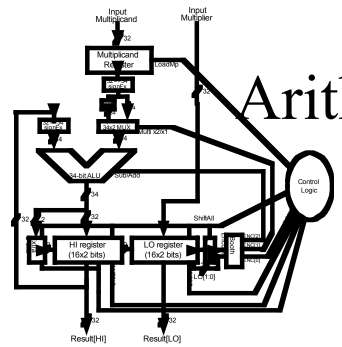
Logic design

- Logical equations, schematic diagrams, FSMs, Digital design

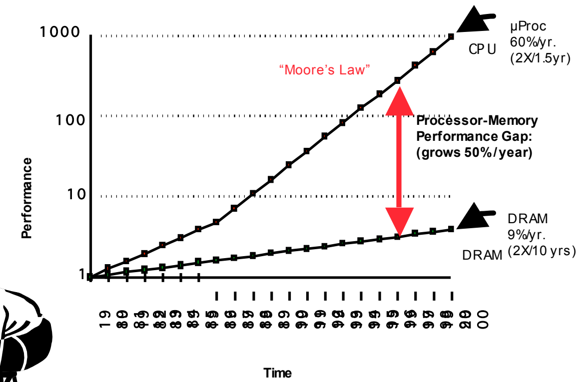
Roadmap

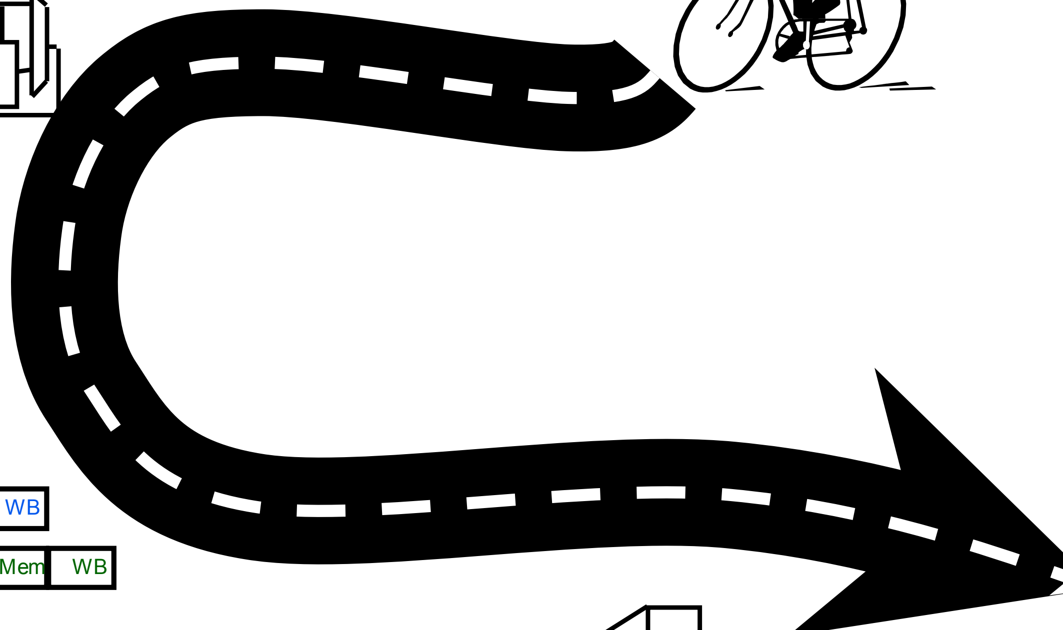
Single/multicycle Datapaths

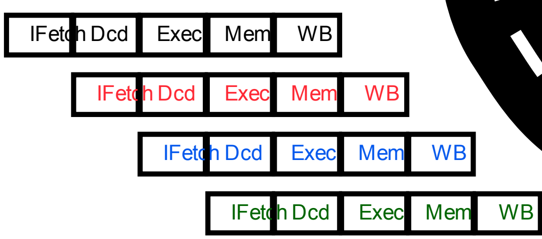


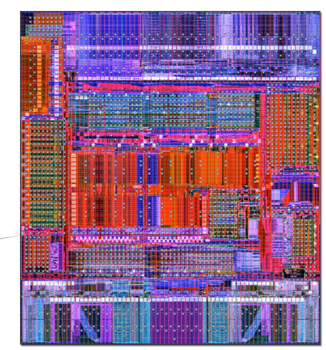


Arithmetic

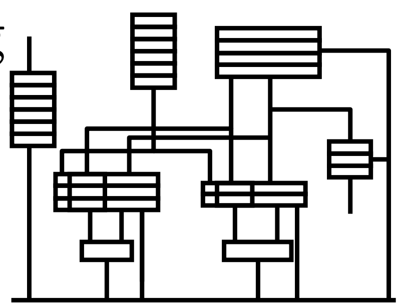


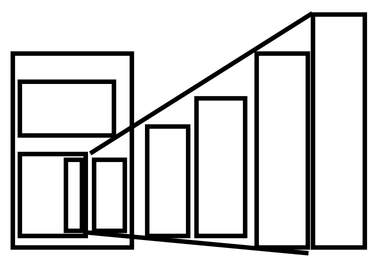




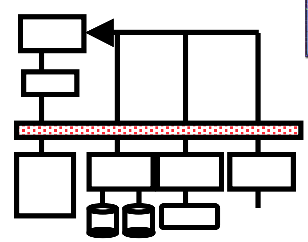


Pipelining





Memory Systems



I/O

Course Basics

Website

- www.ece.northwestern.edu/~choudhar/361/index.htm
- Check regularly for announcements
- All course materials posted -- lecture notes, homework, labs, supplemental materials
- Communicate information, questions and issues

Office Hours – Tech L469 - Tuesday 3-4pm (or by appointment)

Text supplements lectures and assigned reading should be done prior to lectures. I assume that all assigned readings are completed even if the material is not covered in class.

Homework, Labs and Exams

- Collaborative study and discussion is highly encouraged
- Work submitted must be your own
- Individual grade

Project

- Collaborative effort
- Team grade

Grade

35% Homework and Labs

- 4 homework sets
- Lab – individual grade,
  - ALU

30% Team Project

- MIPS subset
- Design and CAD intensive effort

35% Late midterm Exam (Nov 16)

- Open book, open notes

Project

Teams of 3-4 students

You will be required to

- Use advanced CAD tools – Mentor Graphics
- Design a simple processor (structural design and implementation) – MIPS subset
- Validate correctness using sample programs of your own and provided as part of the assignment

Written presentation submitted (due Dec 3, 2004)

You may also use VHDL (structural) to design your system if you know VHDL sufficiently well

Course Structure

Lectures:

- 1 week on Overview and Introduction (Chap 1 and 2)
- 2 weeks on ISA Design
- 4 weeks on Proc. Design
- 2 weeks on Memory and I/O

Reading assignments posted on the web for each week. Please read the appropriate material before the class.

Note that the above is approximate

Copy of all lecture notes available from the department for a charge (bound nicely)

Technology Drivers

Technology Drives Advances in Computer Design

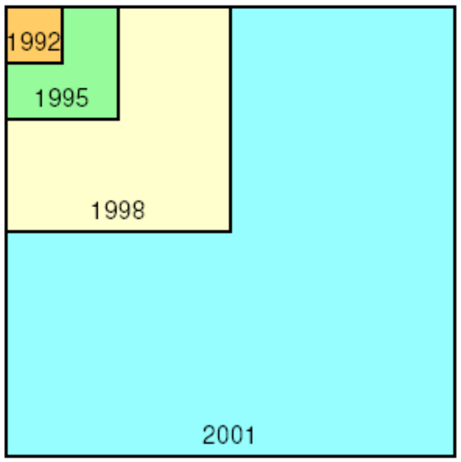
Evolution	Each level of abstraction is continually trying to improve
Disruption	Fundamental economics or capability cross a major threshold

Significant technology disruptions

Logic	**Relays → Vacuum tubes → single transistors → SSI/MSI (TTL/ECL) → VLSI (MOS)**
Registers	**Delay lines → drum → semiconductor**
Memory	**Delay lines → magnetic drum → core → SRAM → DRAM**
External Storage	**Paper tape → Paper cards → magnetic drum → magnetic disk**

Today, technology is driven by semiconductor and magnetic disk technology. What are the the next technology shifts?

Semiconductor and Magnetic Disk Technologies Have Sustained Dramatic Yearly Improvement since 1975

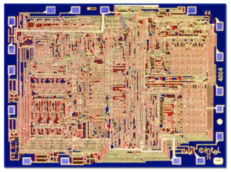


64x more devices since 1992
4x faster devices

Moore's "Law" - The observation made in 1965 by Gordon Moore, co-founder of Intel, that the number of transistors per square inch on integrated circuits had doubled every year since the integrated circuit was invented. Moore predicted that this trend would continue for the foreseeable future. In subsequent years, the pace slowed down a bit, but ***data density has doubled approximately every 18 months***, and this is the current definition of Moore's Law, which Moore himself has blessed. Most experts, including Moore himself, expect Moore's Law to hold for at least another two decades.

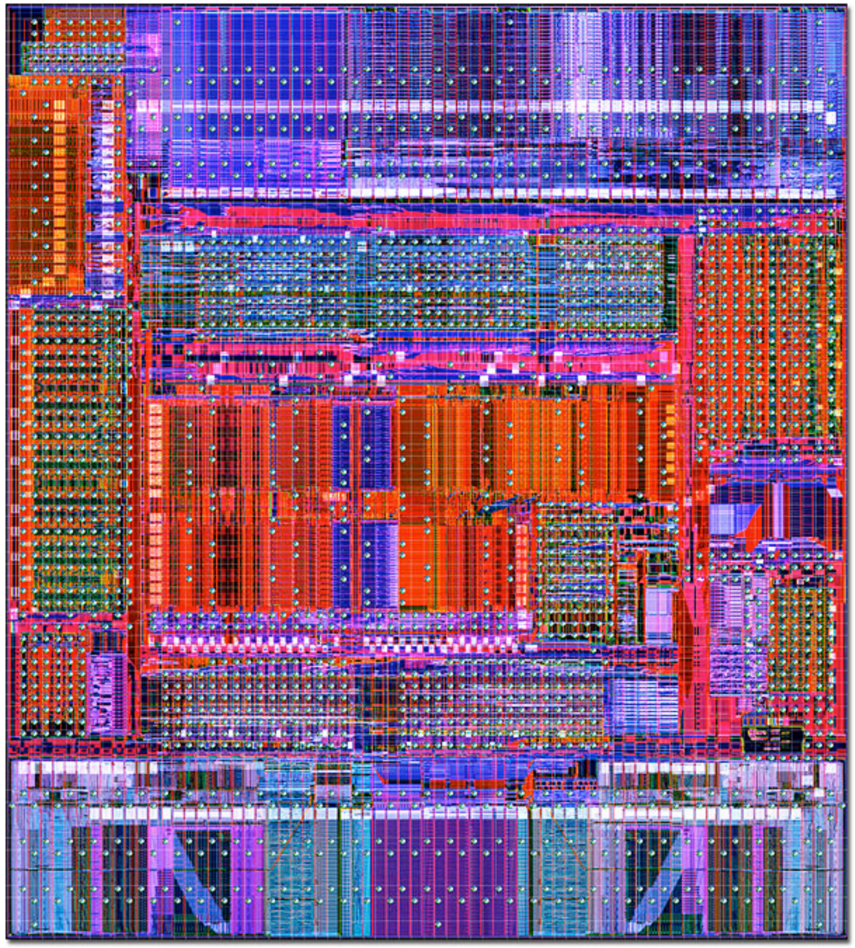
	Capacity	Speed	Cost
Logic	60%	40%	25%
Clock Rate		20%	
DRAM	60%	7%	25%
Disk	60%	3%	25%
Network		40%	25%

Device Density Increases Faster Than Die Size



1971

Intel 4004 was a 3 chip set with a 2kbit ROM chip, a 320bit RAM chip and the 4bit processor each housed in a 16 pin DIP package. The 4004 processor required roughly 2,300 transistors to implement, used a silicon gate PMOS process with 10µm linewidths, had a 108KHz clock speed and a die size of 13.5mm2. Designer – Ted Hoff.



1996

HP PA8000 – 17.68mmx19.1mm, 3.8M transistors.

	i4004	PA9000	Factor	Yearly Improvement
Area (mm)	13.5	338	1:25	14%
Transistors	2300	3,800,000	1:1652	34%

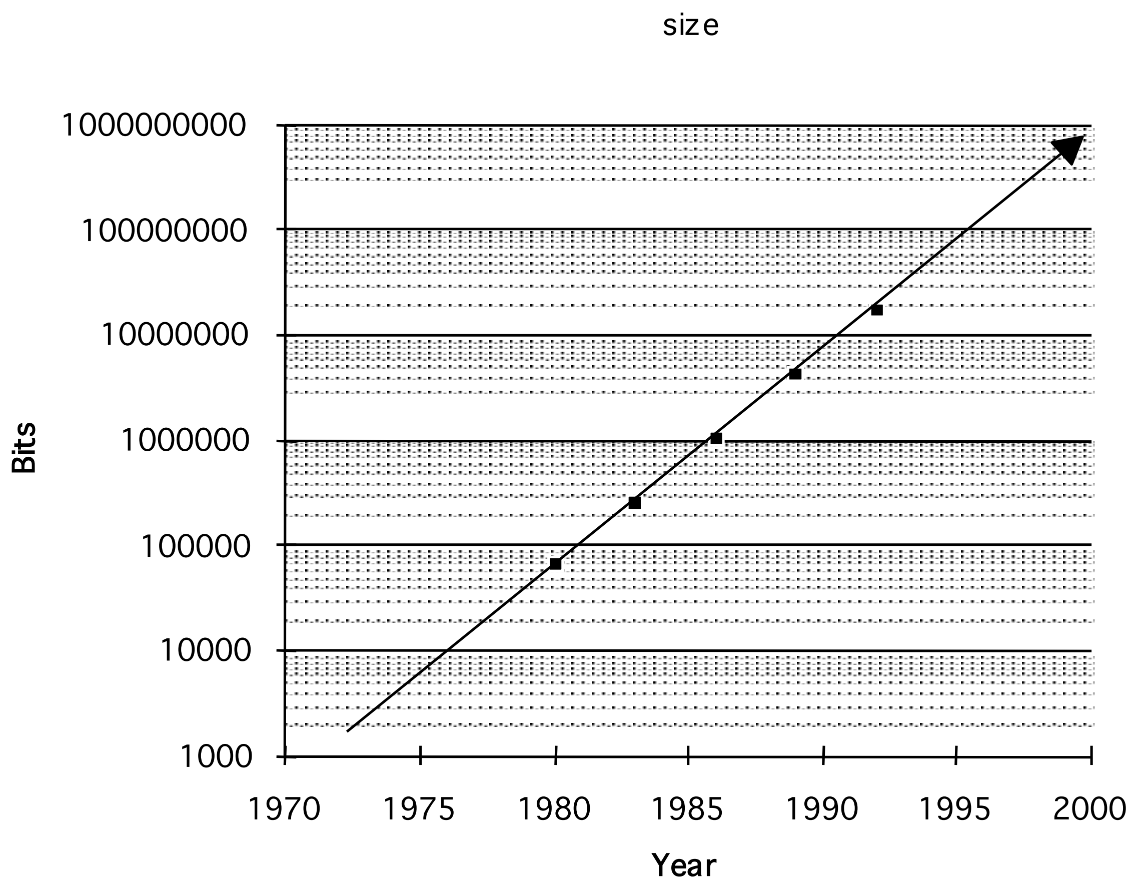
Source: http://micro.magnet.fsu.edu/chipshots/

Example: Intel Semiconductor Roadmap

Process	P856	P858	Px60	P1262	P1264	P1266
1st Production	1997	1999	2001	2003	2005	2007
Lithography	0.25um	0.18um	0.13um	90nm	65nm	45nm
Gate Length	0.20um	0.13um	<70nm	<50nm	<35nm	<25nm
Wafer Diameter (mm)	200	200	200/300	300	300	300

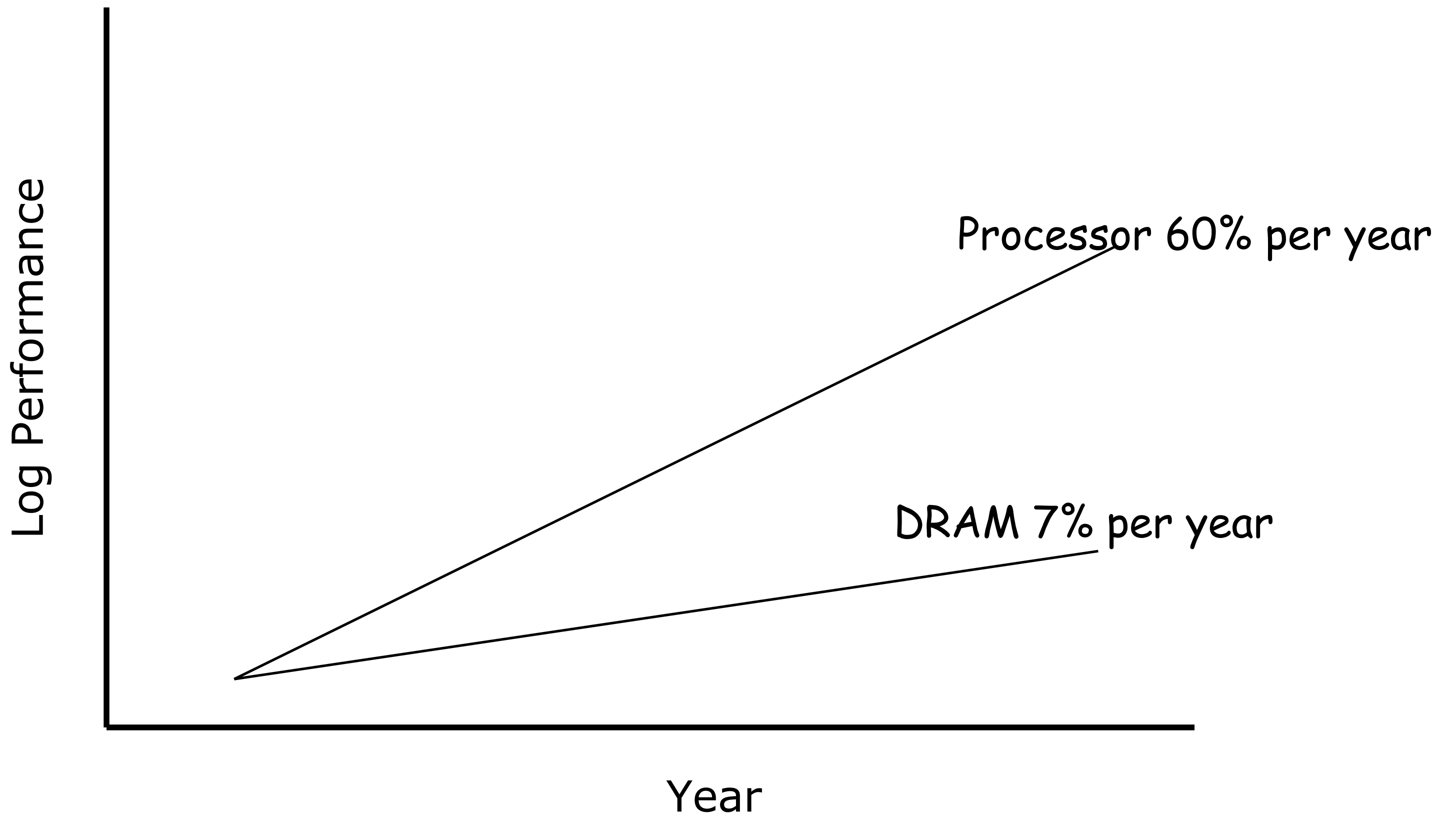
Source: Mark Bohr, Intel, 2002

DRAM Drives the Semiconductor Industry



Year	Capacity	Access
1980	64 Kb	250 ns
1983	256 Kb	220 ns
1986	1 Mb	190 ns
1989	4 Mb	165 ns
1992	16 Mb	145 ns
1996	64 Mb	120 ns
1999	256 Mb	100 ns
2002	1Gb	80 ns

Memory Wall: Speed Gap between Processor and DRAM



Source: Junji Ogawa, Stanford

The divergence between performance and cost drives the need for memory hierarchies, to be discussed in future lectures.

Semiconductor evolution drives improved designs

1970s

- Multi-chip CPUs
- Semiconductor memory very expensive
- Complex instruction sets (good code density)
- Microcoded control

1980s

- 5K – 500 K transistors
- Single-chip CPUs
- RAM is cost-effective
- Simple, hard-wired control
- Simple instruction sets
- Small on-chip caches

1990s

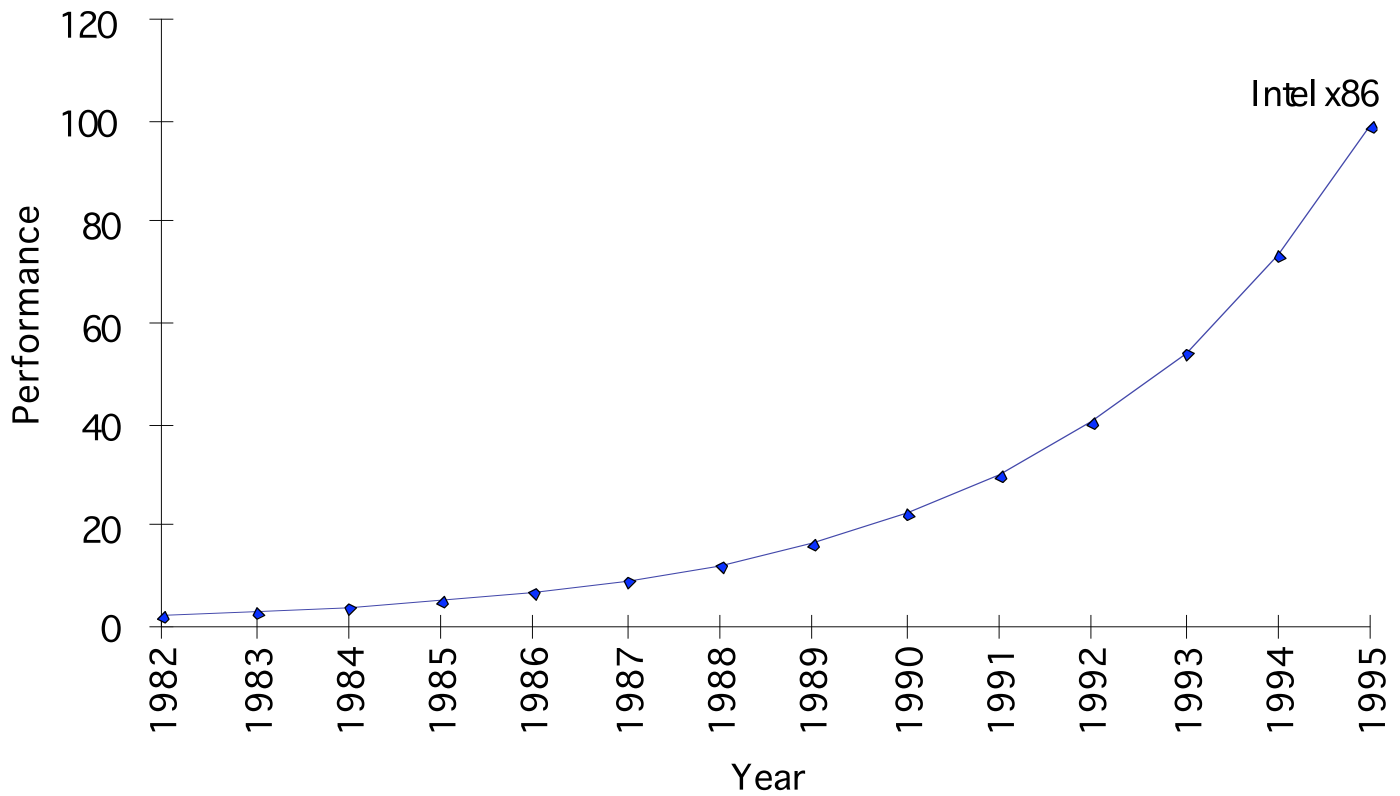
- 1 M - 64M transistors
- Complex control to exploit instruction-level parallelism
- Super deep pipelines

2000s

- 100 M - 5 B transistors
- Slow wires
- Power consumption
- Design complexity

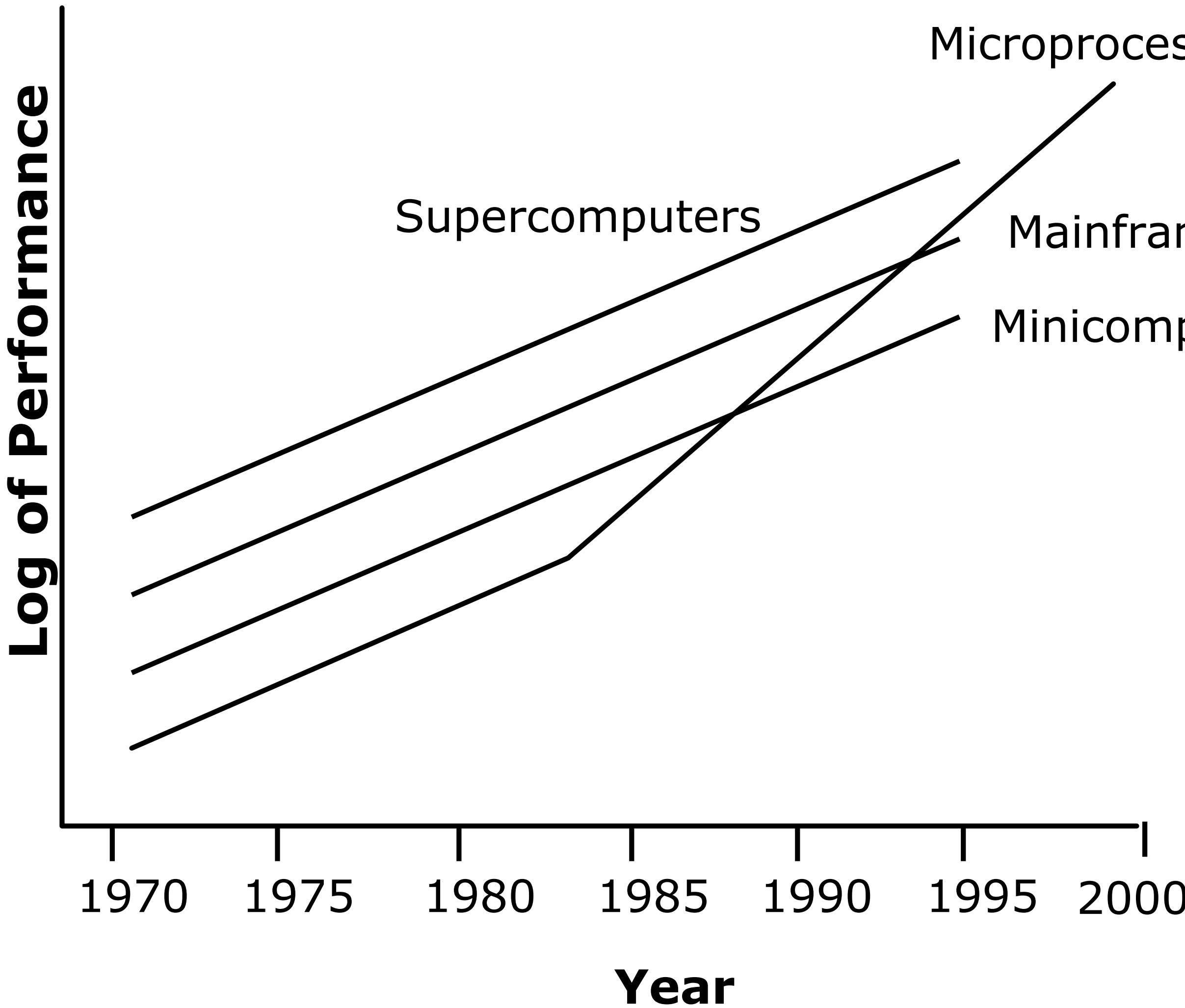
Note: Gate speeds and power/cooling also improved

Processor Performance (SPEC)



System performance improves 50% per year. More about SPEC in a future lecture.

Technology drives industry disruption and creates opportunity



Transistors
Alpha 21264: 15 million
Pentium Pro: 5.5 million
PowerPC 620: 6.9 million
Alpha 21164: 9.3 million
Sparc Ultra: 5.2 million
Pentium 4: 42 million
Itanium: 220 million

危機

Why Such Change in 10 years?

Performance

- Technology Advances
  - CMOS VLSI dominates older technologies (TTL, ECL) in cost and performance
- Computer architecture advances improves low-end
  - RISC, superscalar, RAID, …

Price: Lower costs due to …

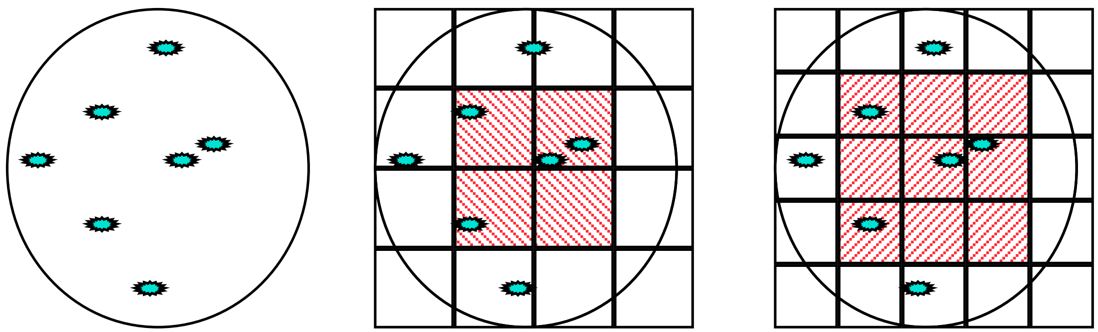
- Simpler development
  - CMOS VLSI: smaller systems, fewer components
- Higher volumes
  - CMOS VLSI : same dev. cost 1,000 vs. 100,000,000 units
- Lower margins by class of computer, due to fewer services

Function

- Rise of networking / local interconnection technology

Cost and Price

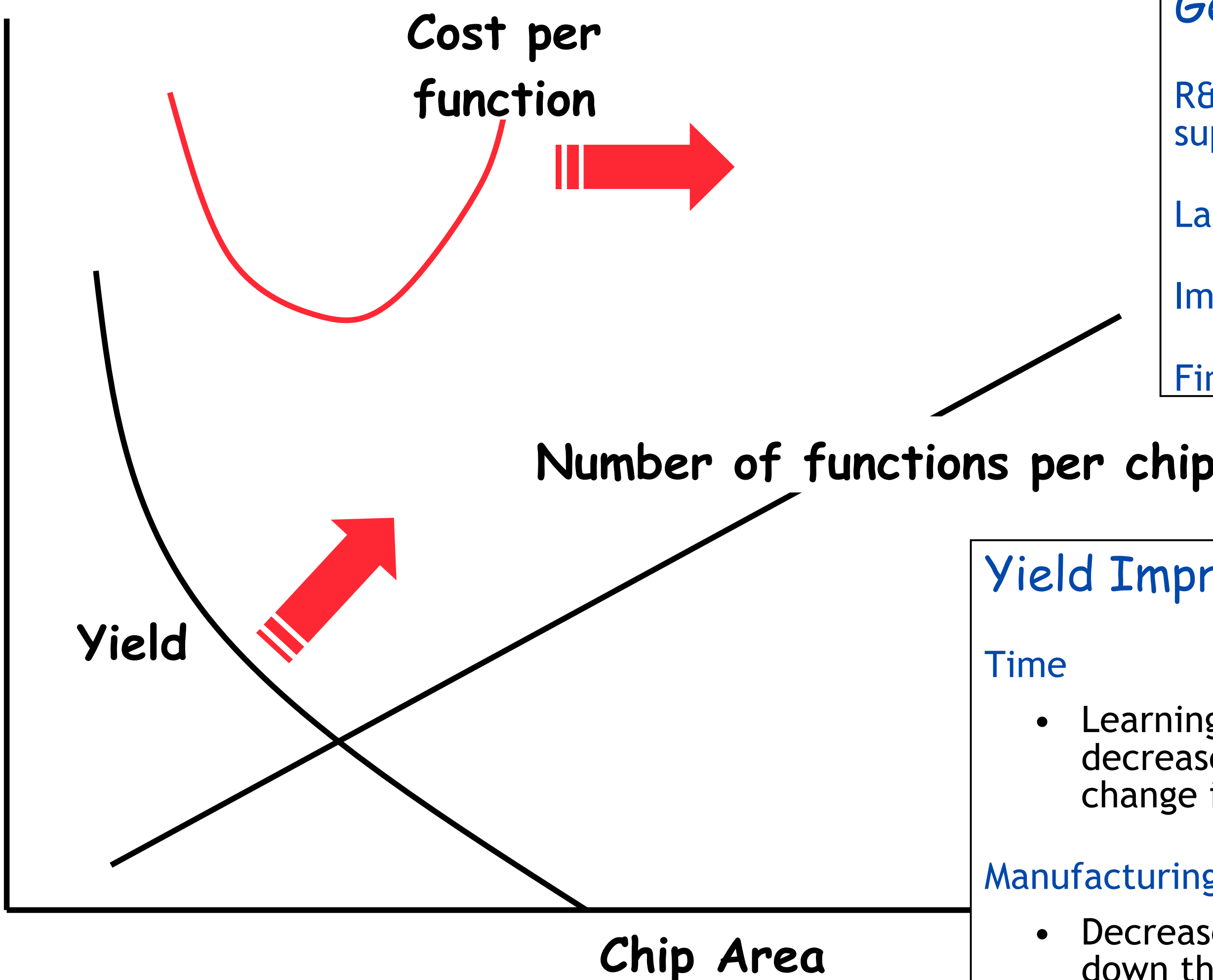
Integrated Circuit Manufacturing Costs



$$\text{Die Cost} = \frac{\text{Wafer Cost}}{\text{Dies per Wafer} \times \text{Die Yield}}$$

IC yield is a largely a function of defect density.
Yield curves improve over time with manufacturing experience.

Relationship of complexity, cost and yield



Generational Improvements

R&D and semiconductor equipment suppliers

Larger wafers

Improved materials

Finer feature sizes

Yield Improvement

Time

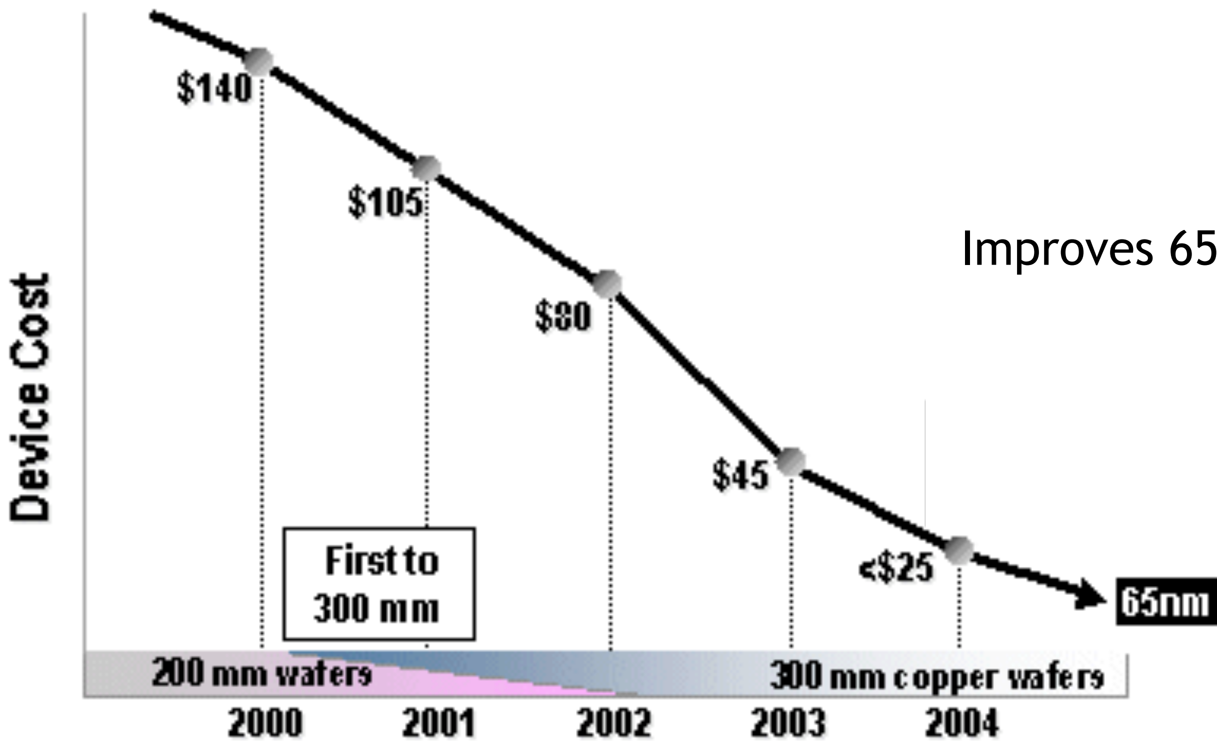
- Learning curve: manufacturing costs decrease over time measured by change in yield

Manufacturing Volume

- Decreases the time needed to get down the learning curve
- Decreases the cost due to improved manufacturing efficiency

Source: The History of the Microcomputer - Invention and Evolution, Stan Mazor

Example: FPGA Cost per 1M Gates



Improves 65% per year

Source: Xilinx

System Cost Example: Web Server

System total cost | Subsystem | % of

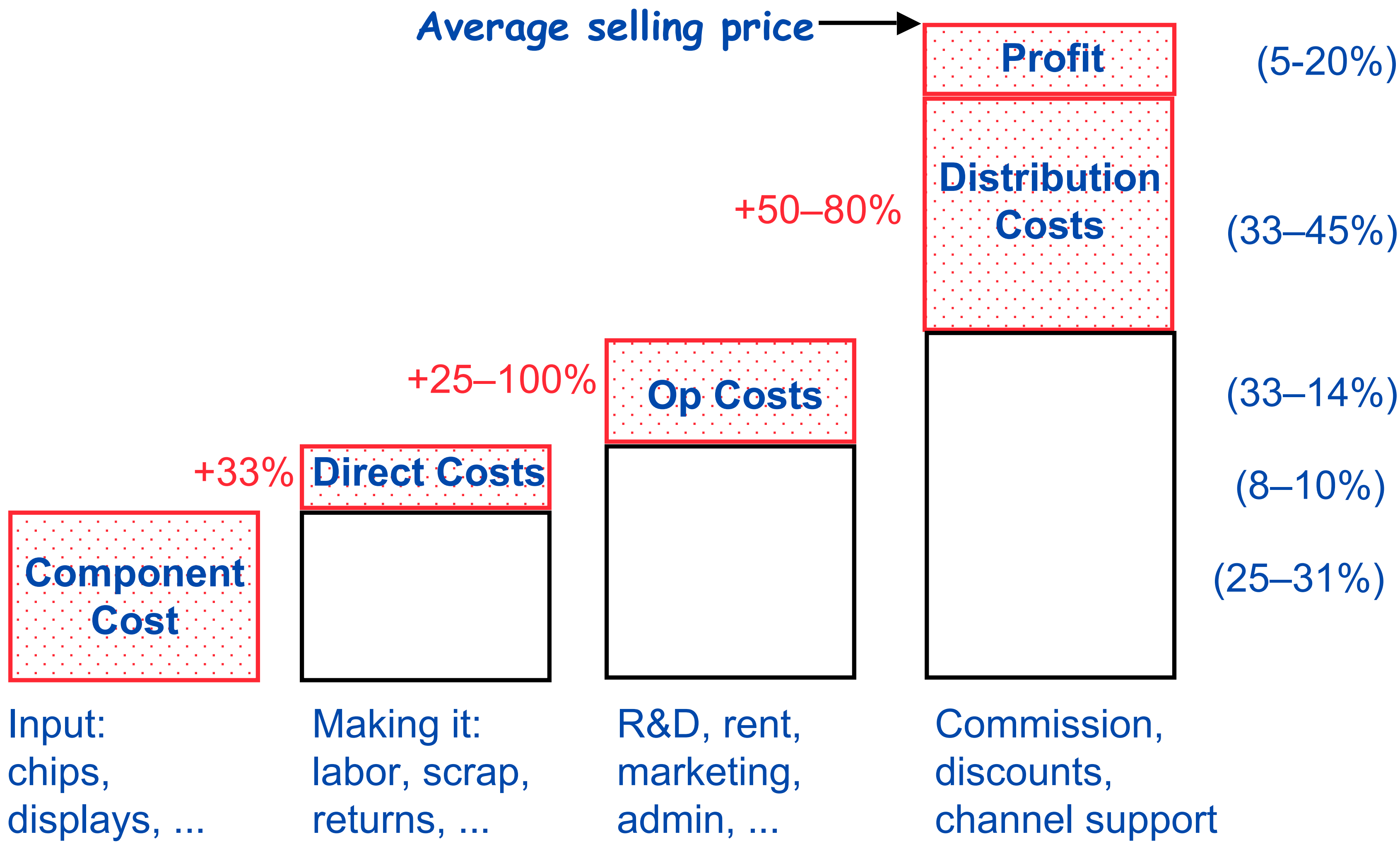
Cabinet   Sheet metal,
plastic   1%
Power supply,
fans   2%
Cables, nuts,
bolts   1%

(Subtotal)   (4%)

Motherboard
Processor   20%
DRAM
20%
I/O
system   10%
Network
interface   4%
Printed Circuit
board   1%
(Subtotal)
(60%)

Picture: http://developer.intel.com/design/servers/sr1300/

Example: Cost vs Price



Summary

Computer Design

- Levels of abstraction
- Instruction sets and computer architecture

Architecture design process

Interfaces

Course Structure

Technology as an architectural driver

- Evolution of semiconductor and magnetic disk technology
- New technologies replace old
- Industry disruption

Cost and Price

- Semiconductor economics