EECS 361
Computer Architecture
Lecture 1

Prof. Gokhan Memik

memik@eecs.northwestern.edu

Course slides developed in part by Profs. Hardavellas, Hoe, Falsafi, Martin, Roth, Lipasti, Goldstein, Mowry

What is Computer Architecture?

"Computer architecture is a set of rules and methods that describe the functionality, organization, and implementation of computer systems"

Wikipedia

"Computer Architecture is the science and art of selecting and interconnecting hardware components to create computers that meet functional, performance and cost goals."

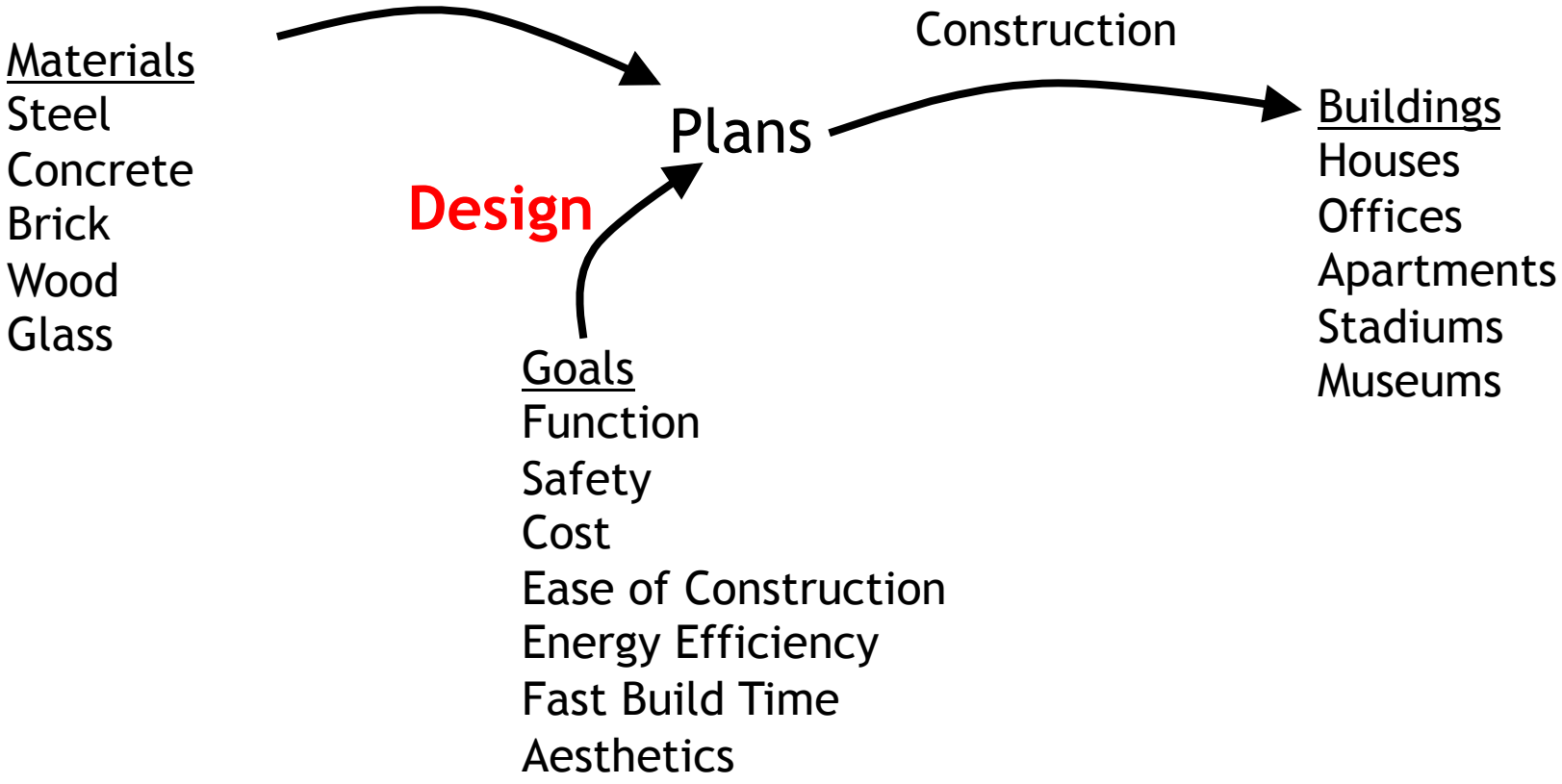
WWW Computer Architecture Page

"The term architecture is used here to describe 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, Blaauw, and Brooks, "Architecture of the IBM system/360", IBM Journal of R&D, April 1964

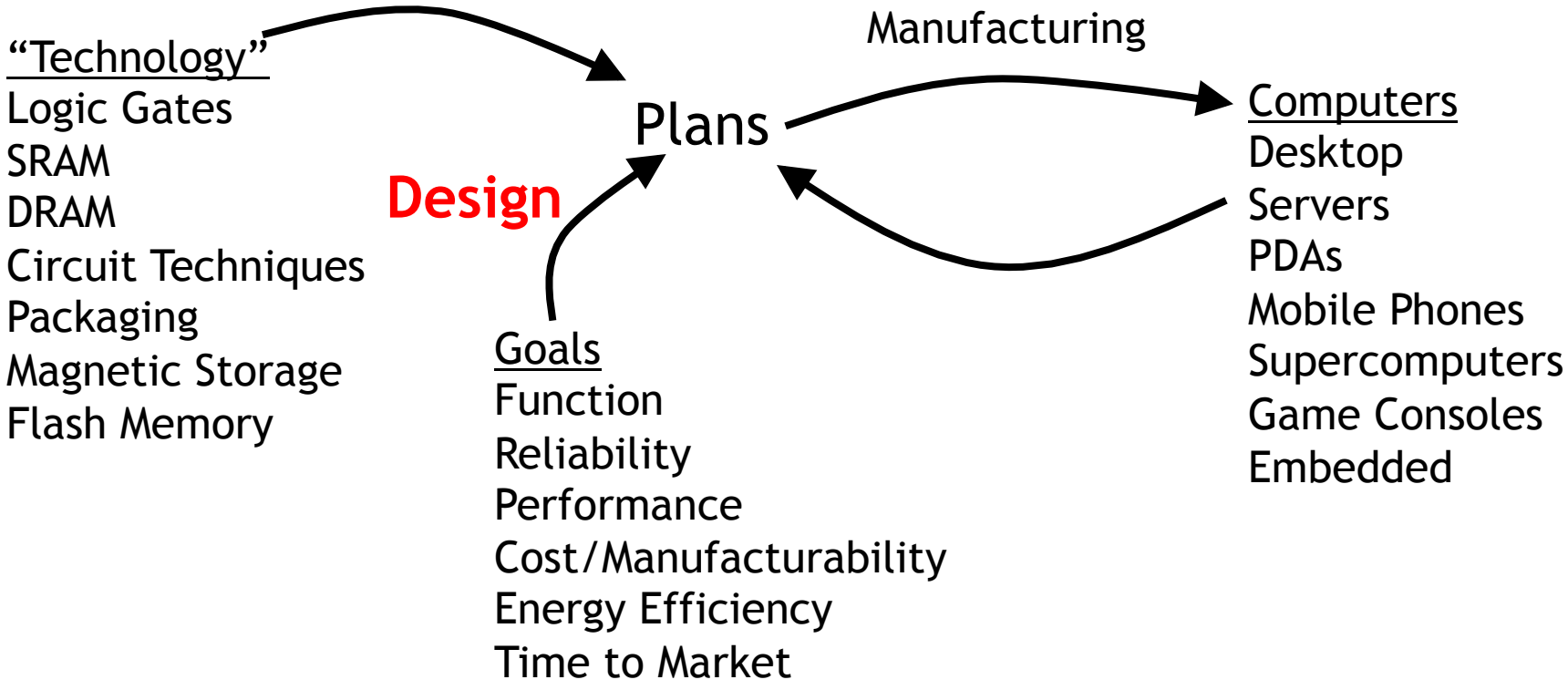
What is Computer Architecture?

The role of the *building* architect:



What is Computer Architecture?

The role of the *computer* architect:



Three important differences: age (~60 years vs. ~10000), rate of change, automated mass production (magnifies design)

Computer Architecture's Changing Definition

1940s - 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, Multicore, Highly parallel structures/architectures (e.g., GPUs), Datacenter as a computer

Today's Lecture

Computer Design

- Levels of abstraction
- Instruction sets and computer architecture

Architecture design process

Interfaces

Course Structure

Technology as an architectural driver

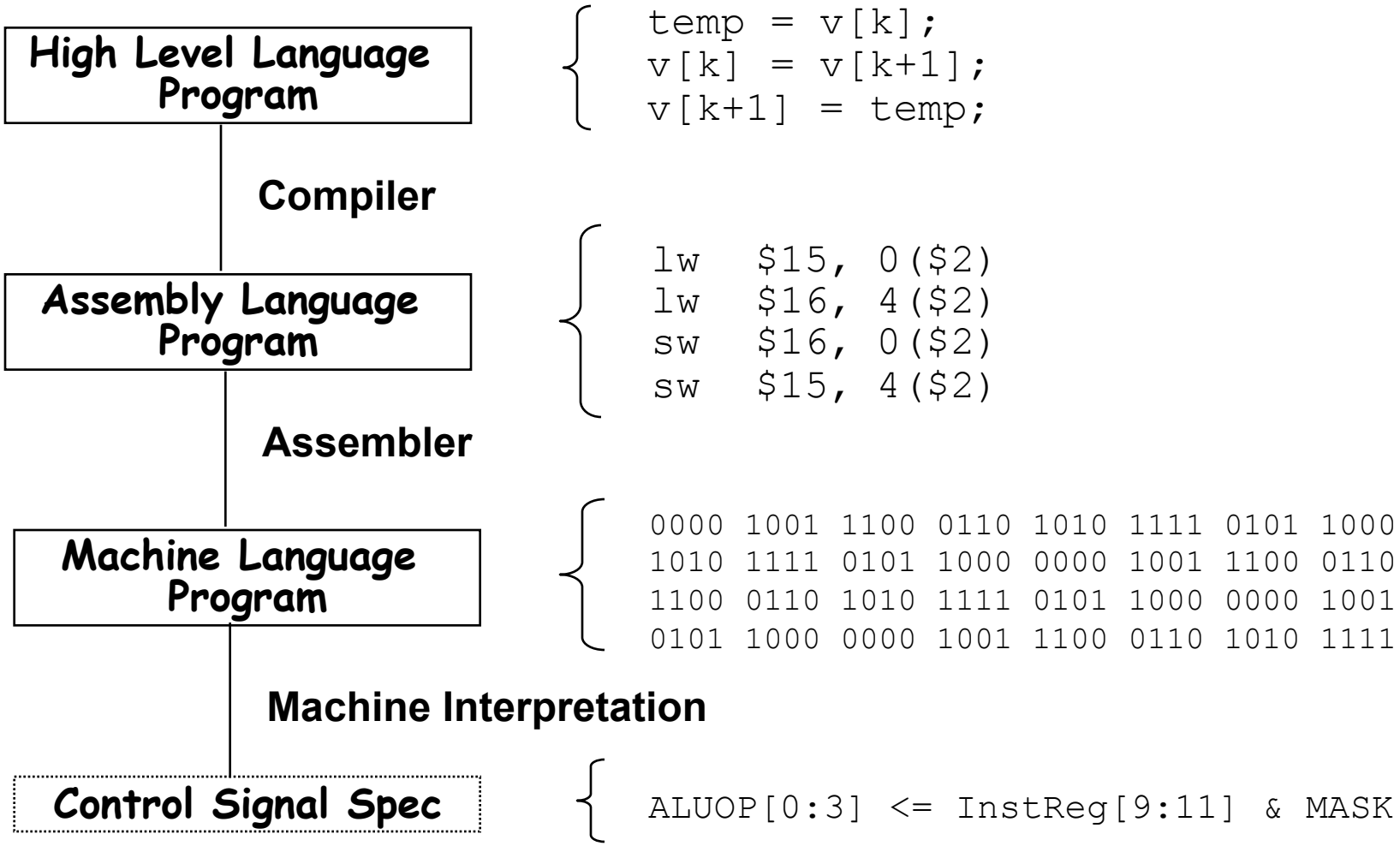
- Evolution of semiconductor and magnetic disk technology
- New technologies replace old
- Industry disruption

Cost and Price

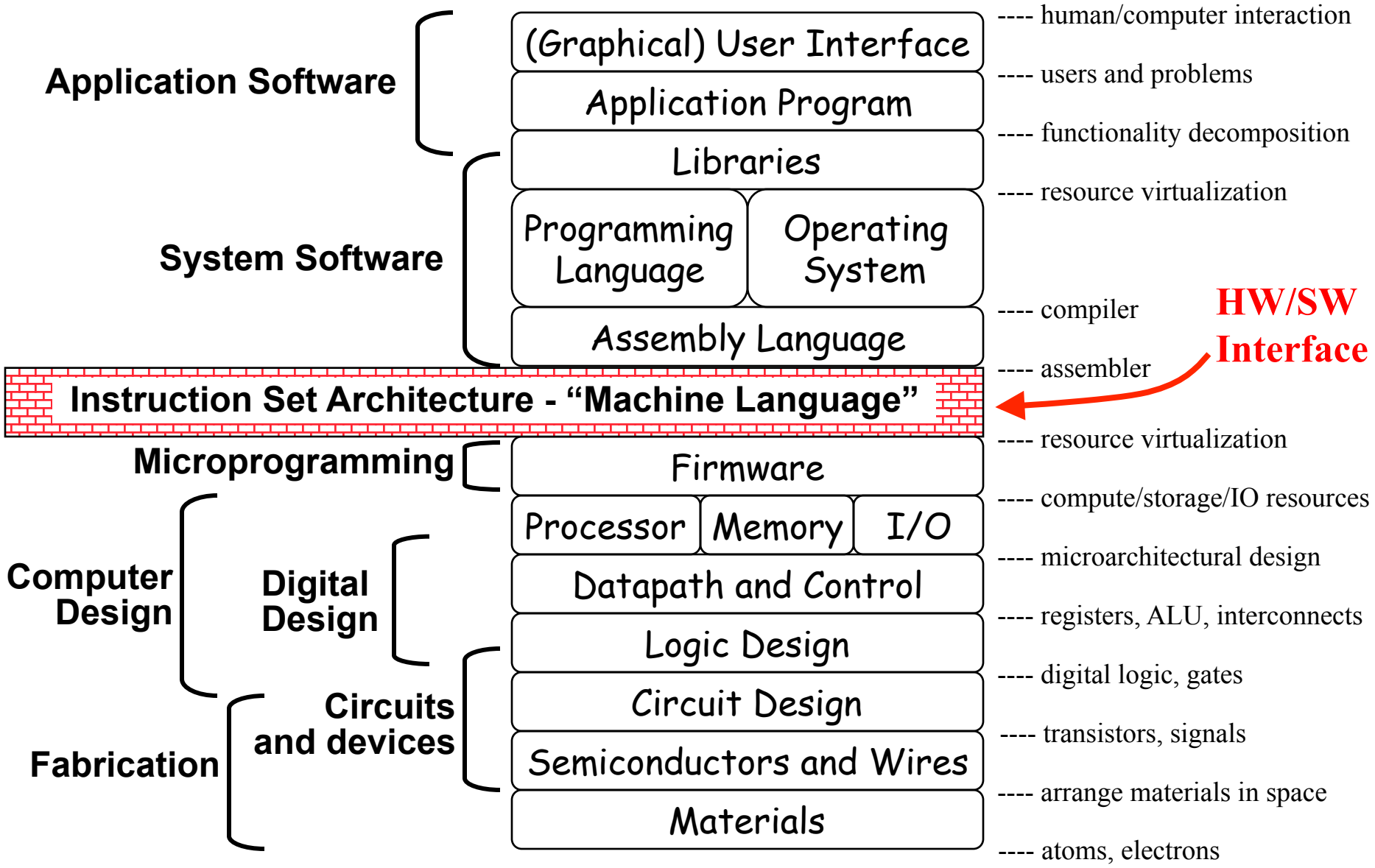
- Semiconductor economics

Computers, Levels of Abstraction and Architecture

Levels of Representation

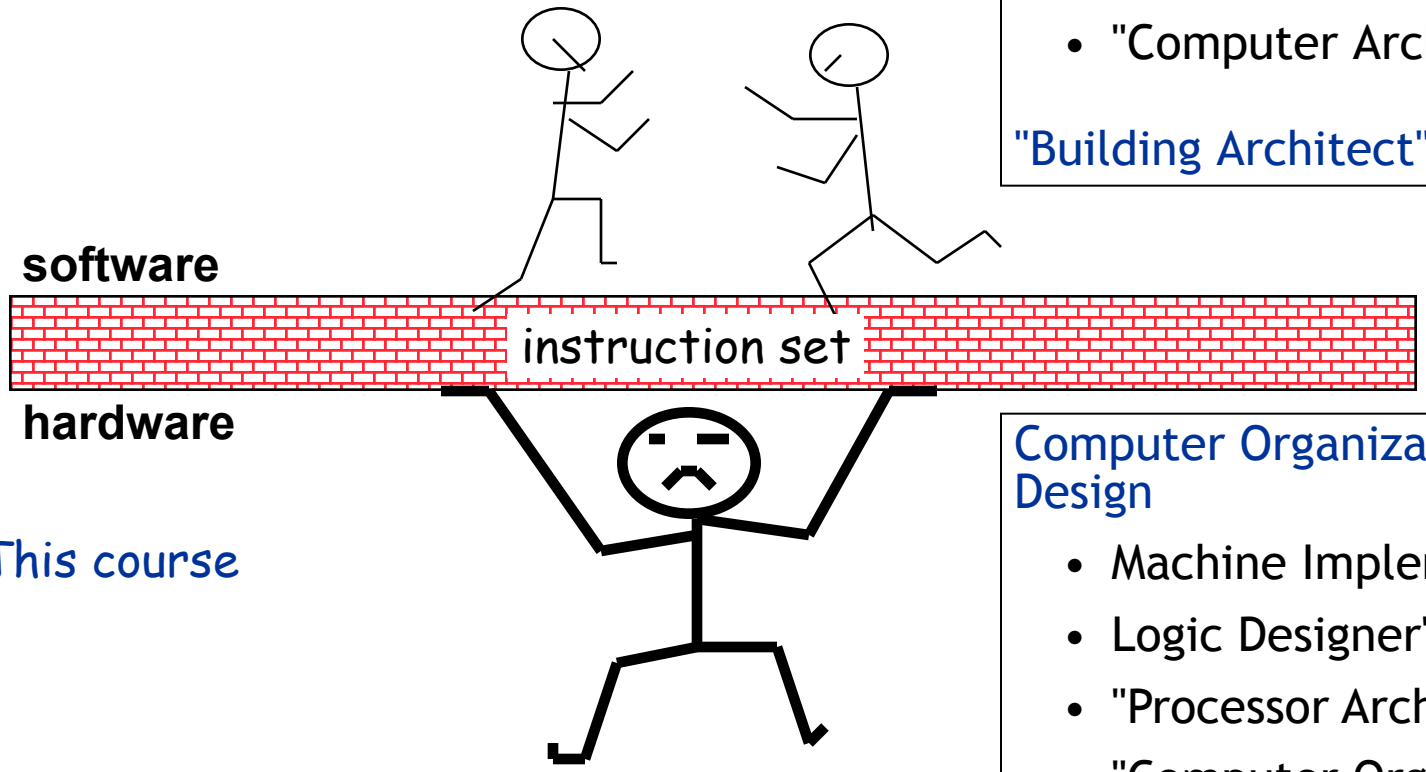


Levels of Abstraction



The Instruction Set: A Critical Interface

Computer Architecture =
Instruction Set Architecture +
Machine Organization



This course

Instruction Set Design

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

"Building Architect"

Computer Organization and Design

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

"Construction Engineer"

Characteristics of a Good Interface Design

Well defined for users and implementers

Interoperability (Hardware) / Compatibility (Software)

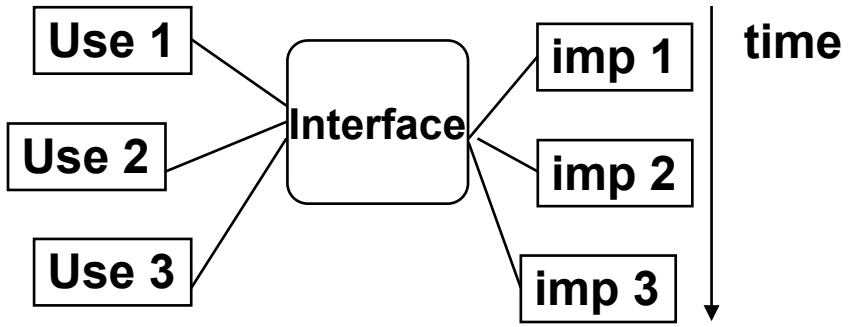
- Lasts through multiple implementations across multiple technologies (portability, compatibility)
- Efficiently supports multiple implementations
  - Competitive market
  - Compatible at multiple cost / performance design points

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
  - ARM 1985 ~ present
- Network protocols
  - Ethernet 1973 ~ present
  - TCP/IP 1974 ~ present
- Programming languages
  - C 1973 ~ present

Instruction Set Architecture

Data Types
- Encoding and representation

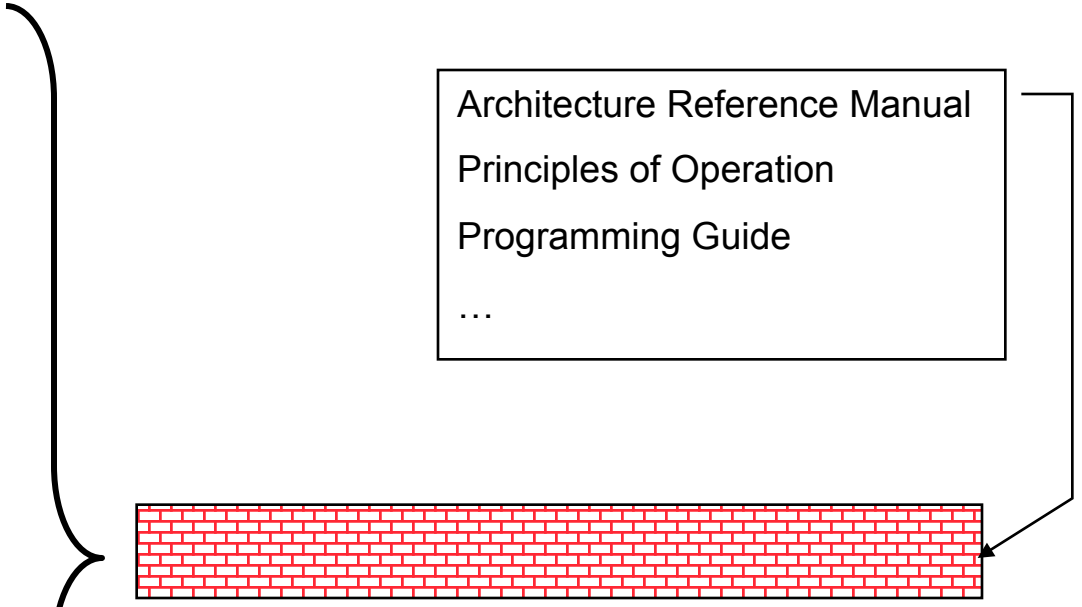
Memory Model

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

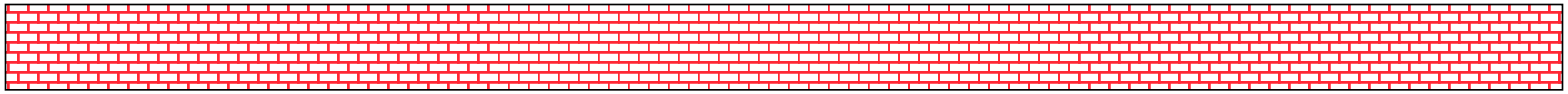
External Interfaces
- IO
- Management



. . . 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

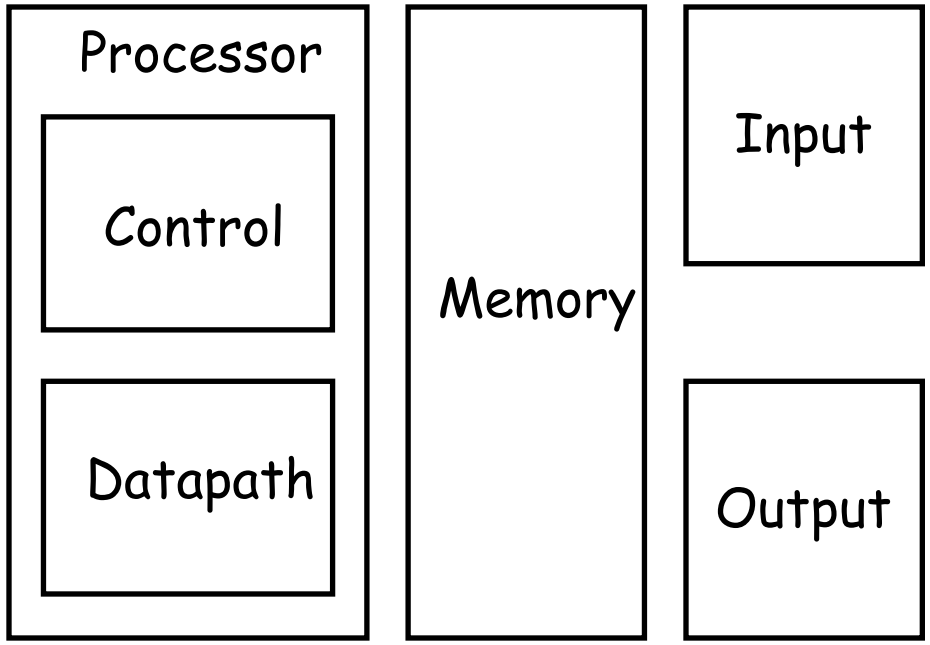
A general-purpose computer system

- Uses a programmable processor
- Can run "any" application
- Potentially optimized for some class of applications
- Common names: CPU, microprocessor

Unified main memory

- For both programs & data
- Von Neumann computer

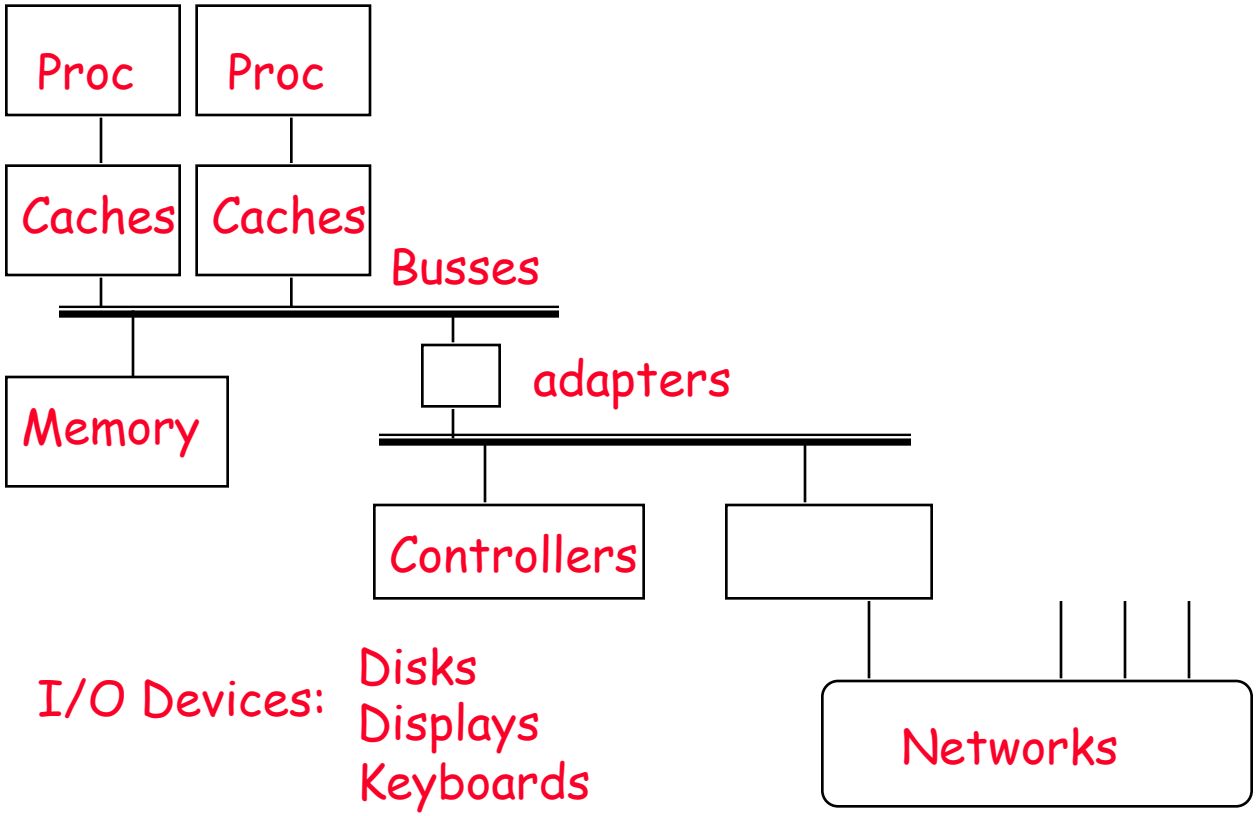
Busses & controllers to connect processor, memory, IO devices



MIT Whirlwind, 1951

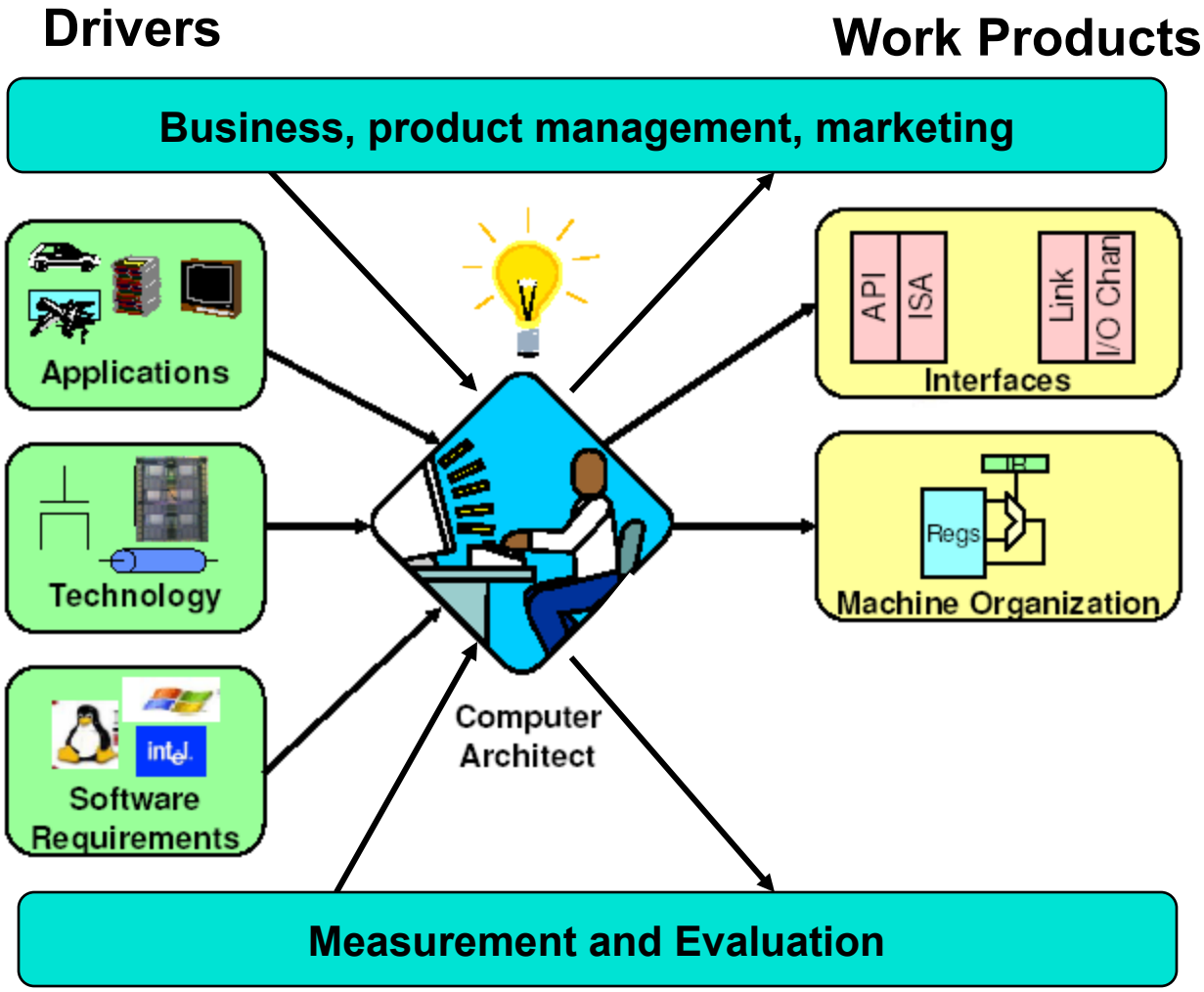
Computers are pervasive – servers, standalone PCs, network processors, embedded processors, …

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

Server computing

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

Desktop computing

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

Laptop computing

- Metrics: performance (latency), cost, power

Embedded computing

- Examples: microwave, printer, cell phone, video console, car's ABS break controller
- Metrics: performance (real-time), cost, power consumption, thermal constraints, complexity, reliability/safety

Applications Drive Design Points

Numerical simulations

- Floating-point performance
- Main memory bandwidth

Transaction processing

- I/Os per second, memory latency + 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

Course Structure

What You Need to Know from prerequisites

Basic machine structure

- Processor, memory, I/O

Assembly language programming

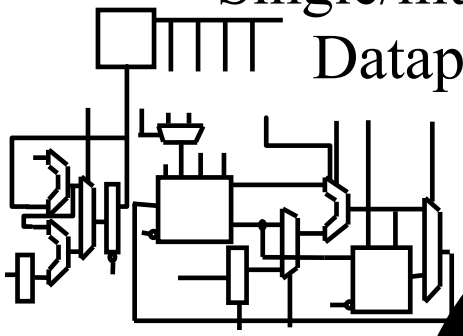
Simple operating system concepts

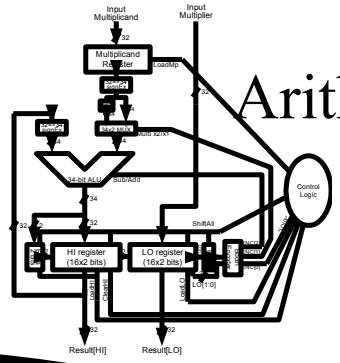
Logic design

- Logical equations, schematic diagrams, FSMs, Digital design

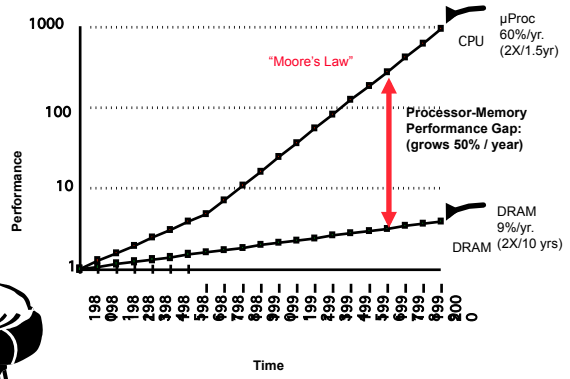
Roadmap

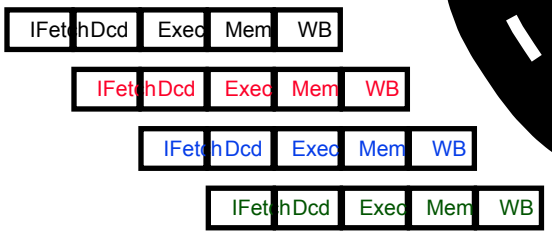
Single/multicycle Datapaths



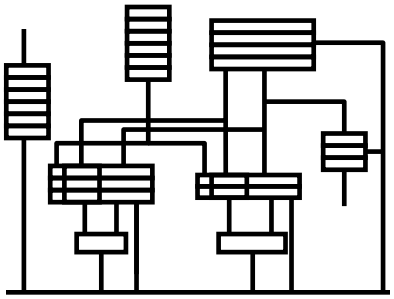


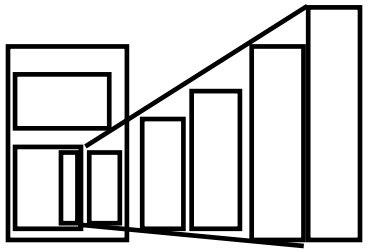
Arithmetic



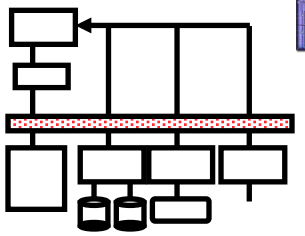


Pipelining

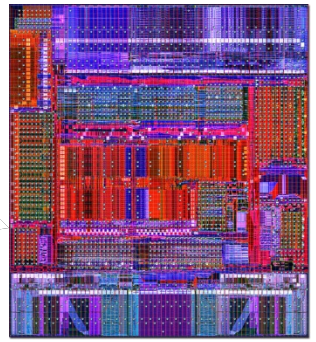




Memory Systems



I/O



Course Logistics

Website

- I will be using Canvas
- Check regularly for announcements
- All course materials posted -- lecture notes, homework, labs, supplemental materials
- Communicate information, questions and issues

Office Hours: Tech L475 – Tuesday 2-3pm (or by appointment)

Teaching Assistant:

Majed Valad Beigi, Tech L458, majed.beigi@northwestern.edu

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.

Homeworks, Lab and Exam

- Discussion is encouraged
  - Work submitted must be your own
- Individual grade

Two Projects

- Collaborative effort
- Team grade

Grade

30% Homework and Labs

- 3 or 4 homework sets
  - Each set will consist of 5 - 10 exercises from the book
- Lab – individual grade
  - Design a 32-bit ALU supporting a portion of the MIPS instructions
- Late penalty: 10% per calendar day

40% Team Projects

- MIPS subset
- Design and CAD intensive effort
- Two projects
  - A simple CPU design (tentative: 25%)
  - A cache design (tentative: 15%)
- Two demo presentations
  - Second one in finals week (tentatively December 8th)

30% Exam (One late exam – Tentatively Dec. 1st)

5% Bonus from quizzes

Project

Teams of students

Two projects

- First project
  - 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
- Second project
  - Design of a cache
  - Validate correctness using sample traces

Written documents submitted during demos

You have to use VHDL to design your system

- **You have to use STRUCTURAL VHDL**
- **Cannot use built-in constructs**

Course Structure

Lectures:

- 1 week on Overview and Introduction (Chapters 1 and 2)
- 2 weeks on ISA Design
- 4 weeks on Processor Design
- 3 weeks on Memory and I/O

Note that the above is approximate

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

READING ASSIGNMENTS FOR LECTURES 1 Through 4:

Sections 1.1 – 1.7 (inclusive)
Sections 2.1 – 2.8 (inclusive)

Technology Drivers

Technology Drives Advances in Computer Design

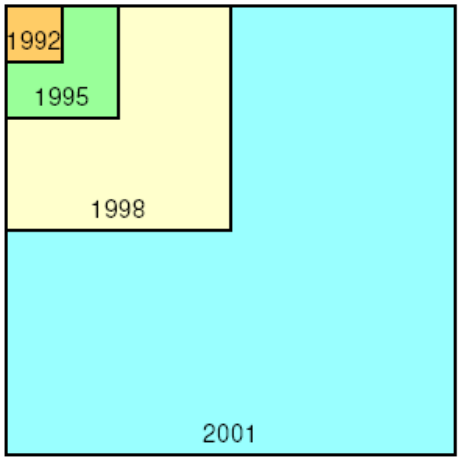
Evolution	Each level of abstraction is continually trying to improve
Disruption	Fundamental economics or capability cross a major threshold

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 → solid-state
External Storage	paper tape → paper cards → → magnetic drum → magnetic disk → magnetic disk/solid-state

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

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

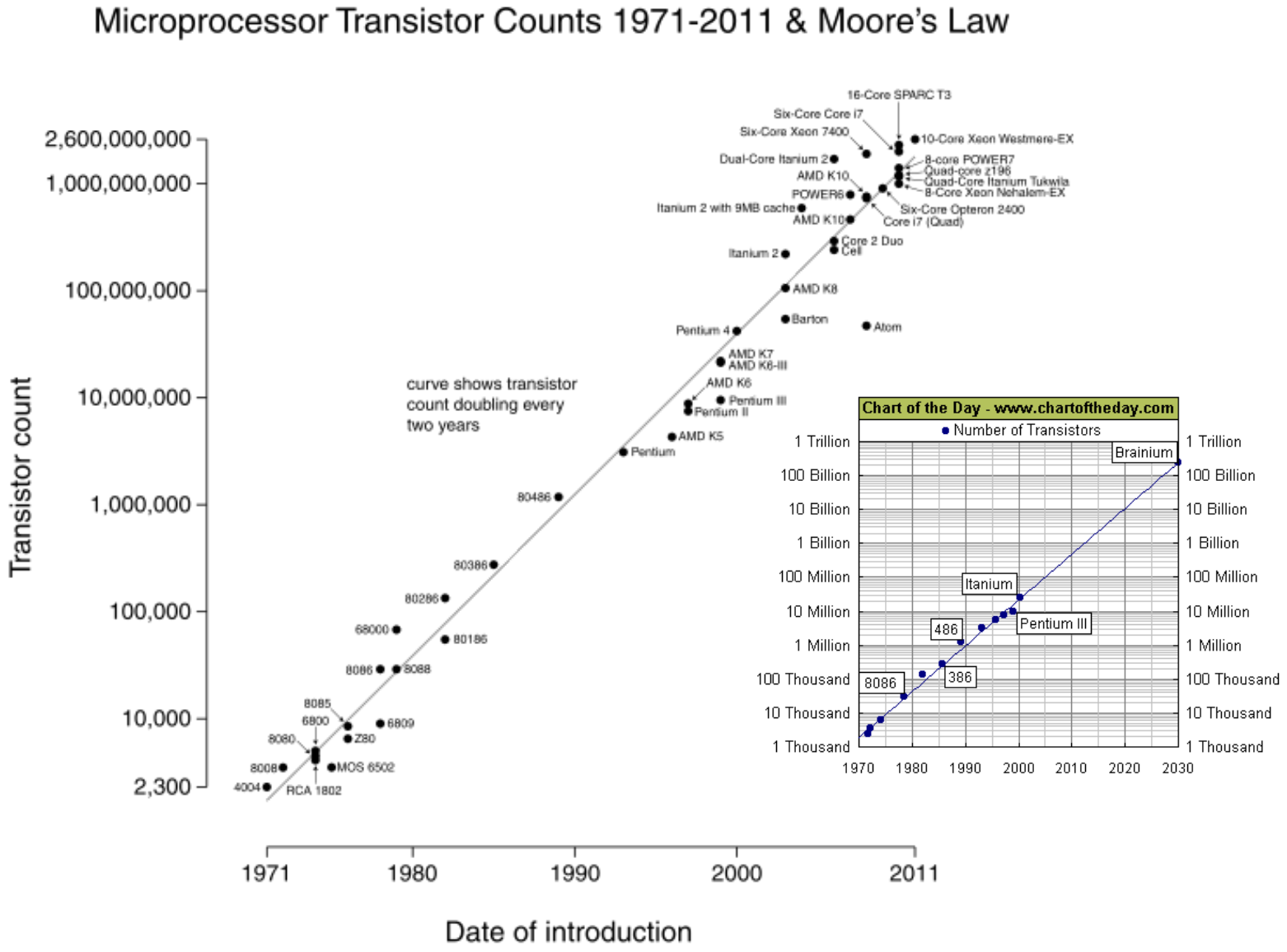


64x more devices since 1992
4x faster devices

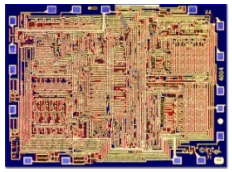
Moore's "Law" - The observation made in 1965 by Gordon Moore, co-founder of Intel, that the number of transistors in integrated circuits had doubled every year since the integrated circuit was invented. Moore predicted that this trend would continue for "at least ten years". In subsequent years, the pace slowed down a bit, but ***the number of transistors on ICs has doubled every two years***, and this is the current definition of Moore's Law, which Moore himself has blessed. This exponential growth is the driving force behind the technological and social changes in the late 20th and early 21st centuries.

	Capacity	Speed	Cost
Logic	40-50%	20%-40%	25%
Clock Rate		20%	
DRAM	40-50%	7%	25%
Disk	40-50%	3%	25%
Network		40%	25%

Moore's Law

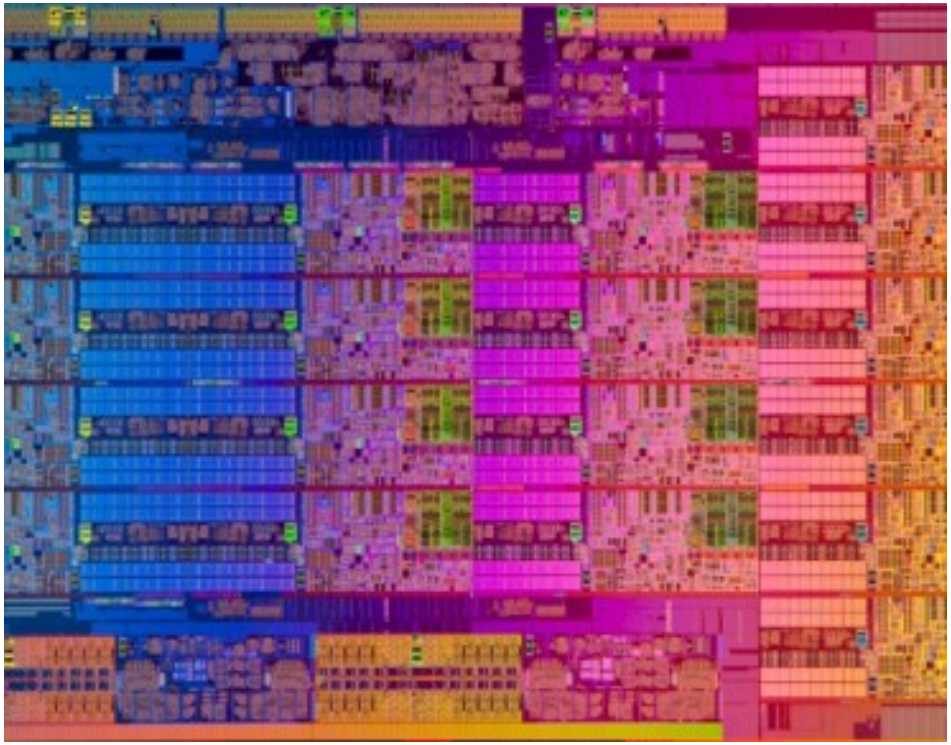

Microprocessor Transistor Counts 1971-2011 & Moore's Law
Transistor count
2,600,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
2,300
curve shows transistor count doubling every two years
16-Core SPARC T3
Six-Core Core i7
Six-Core Xeon 7400
10-Core Xeon Westmere-EX
Dual-Core Itanium 2
8-core POWER7
Quad-core z196
AMD K10
Quad-Core Itanium Tukwila
8-Core Xeon Nehalem-EX
POWER6
Six-Core Opteron 2400
Itanium 2 with 9MB cache
Core i7 (Quad)
AMD K10
Core 2 Duo
Cell
Itanium 2
AMD K8
Barton
Atom
Pentium 4
AMD K7
AMD K6-III
AMD K6
Pentium III
Pentium II
AMD K5
Pentium
80486
80386
80286
68000
80186
8086
8088
8085
6800
6809
8080
Z80
8008
MOS 6502
4004
RCA 1802
1971
1980
1990
2000
2011
Date of introduction
Chart of the Day - www.chartoftheday.com
Number of Transistors
1 Trillion
100 Billion
10 Billion
1 Billion
100 Million
10 Million
1 Million
100 Thousand
10 Thousand
1 Thousand
Brainium
Itanium
Pentium III
486
386
8086
1970
1980
1990
2000
2010
2020
2030

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 13.5mm^2. Designer – Ted Hoff.



2014

Haswell E-5
661 mm^2
5.56B transistors

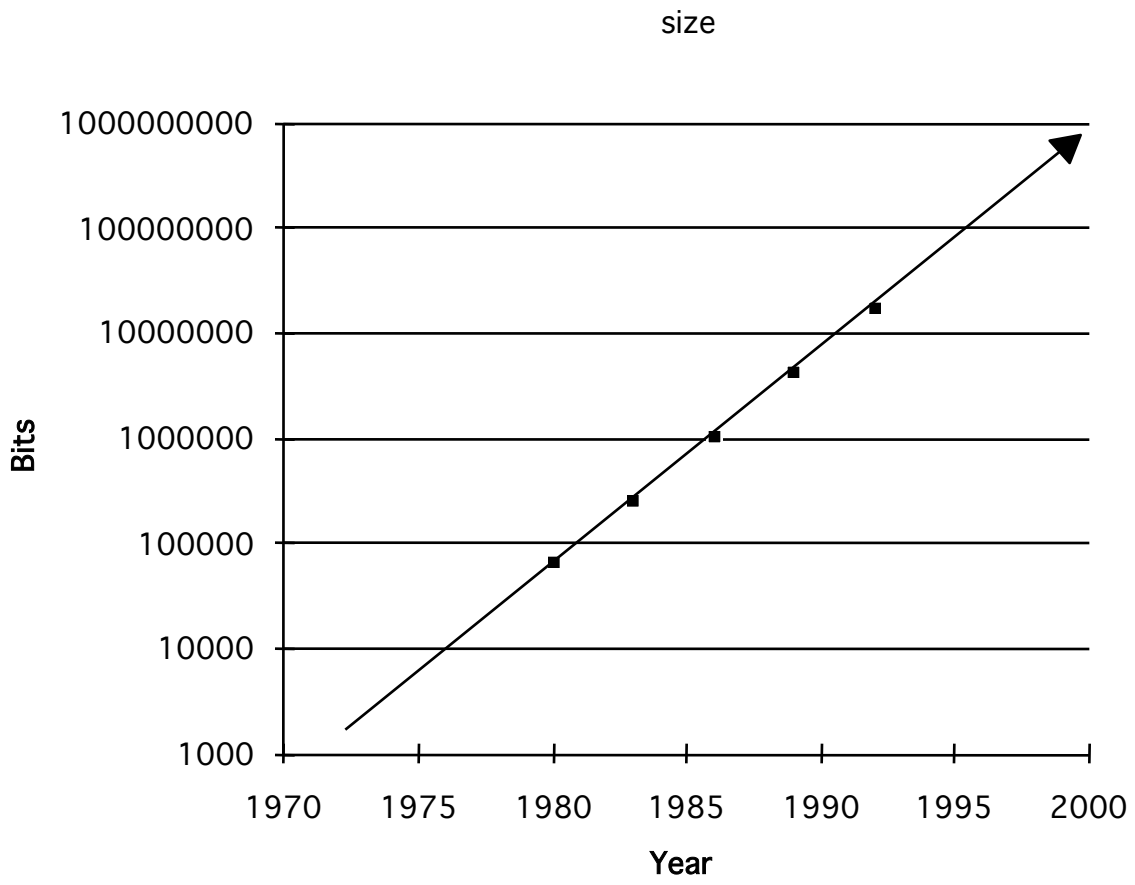
	i4004	Pentium 4	Factor	Yearly Improvement
Area(mm^2)	13.5	661	1:48	9%
Transistors	2300	5.6 B	1:2.4M	41%

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

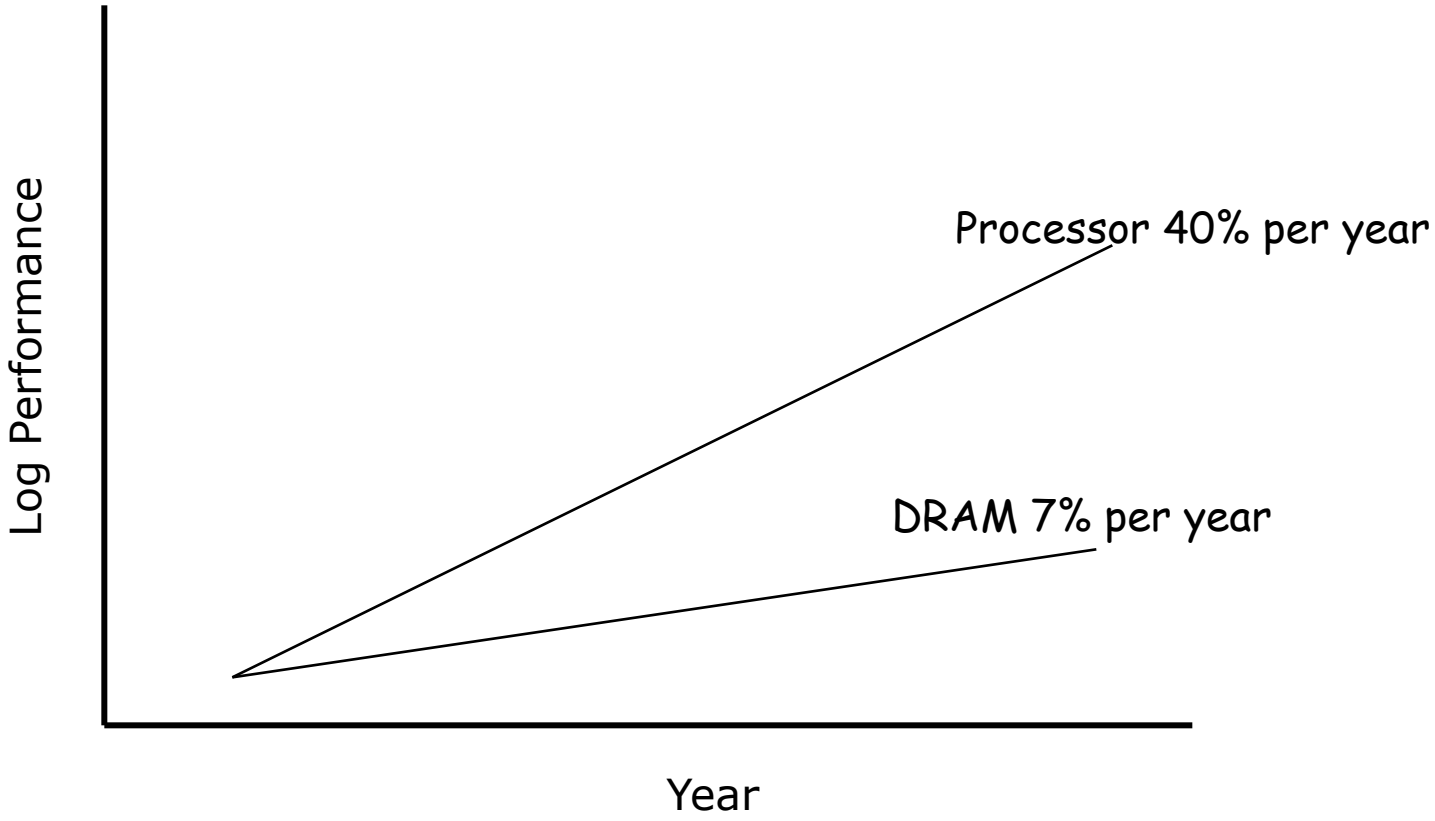
Source: Mark Bohr, Intel

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
2009	8 Gb	50 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

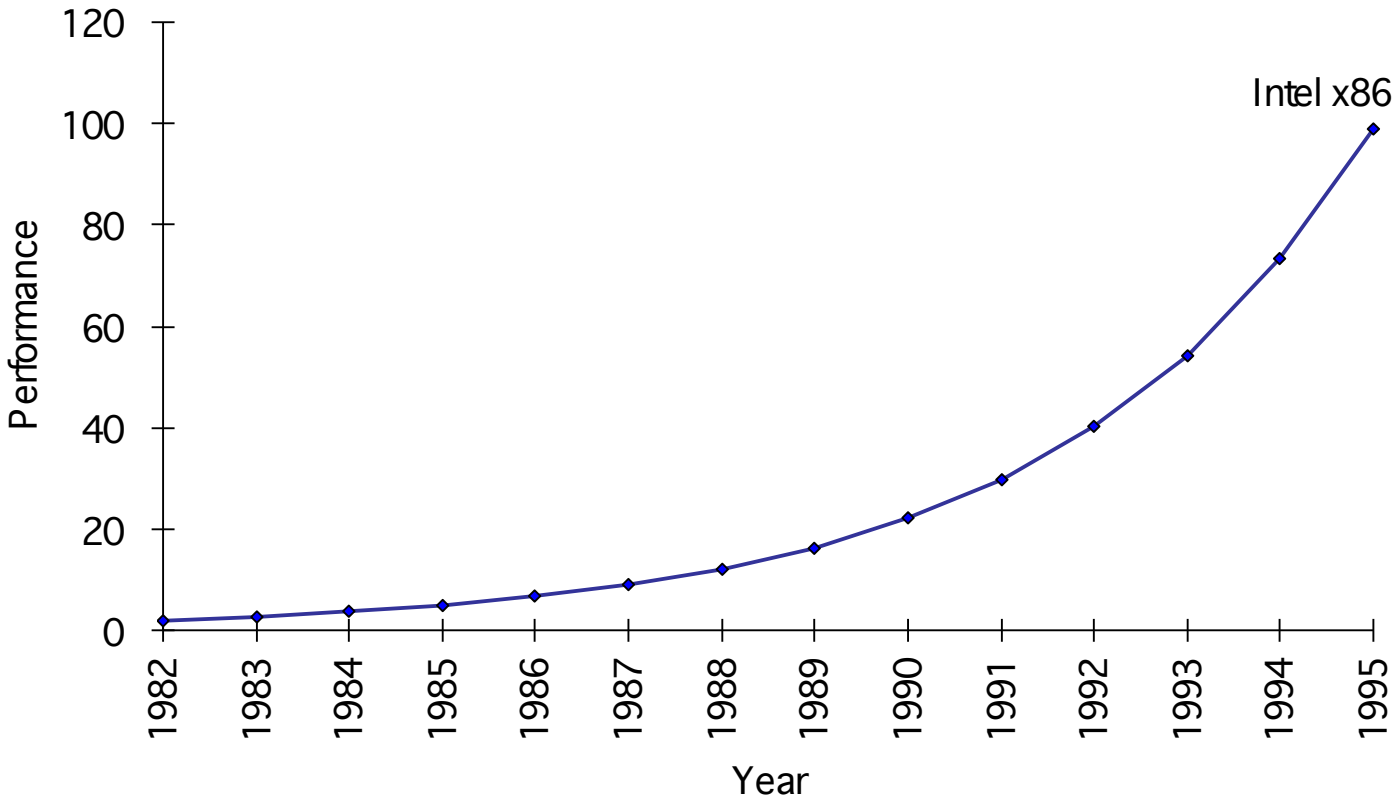
- 1 M - 64M transistors
- Complex control to exploit instruction-level parallelism
- Super deep pipelines

2000s

- 100 M - 5 B transistors
- Slow wires
- Power consumption
- Design complexity
- Multiple cores on a single chip

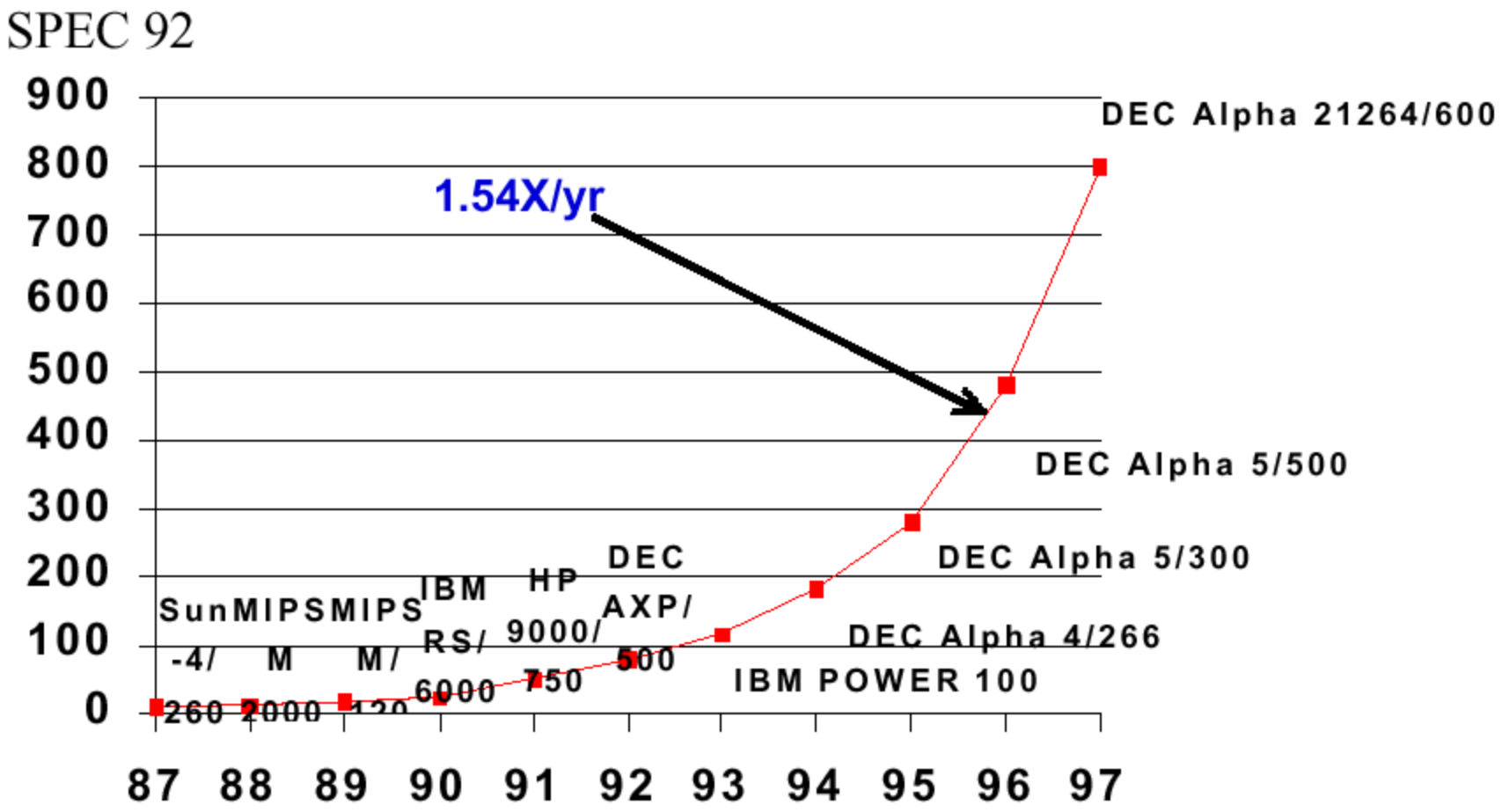
Note: Gate speeds and power/ cooling also improved

Processor Performance (SPEC)

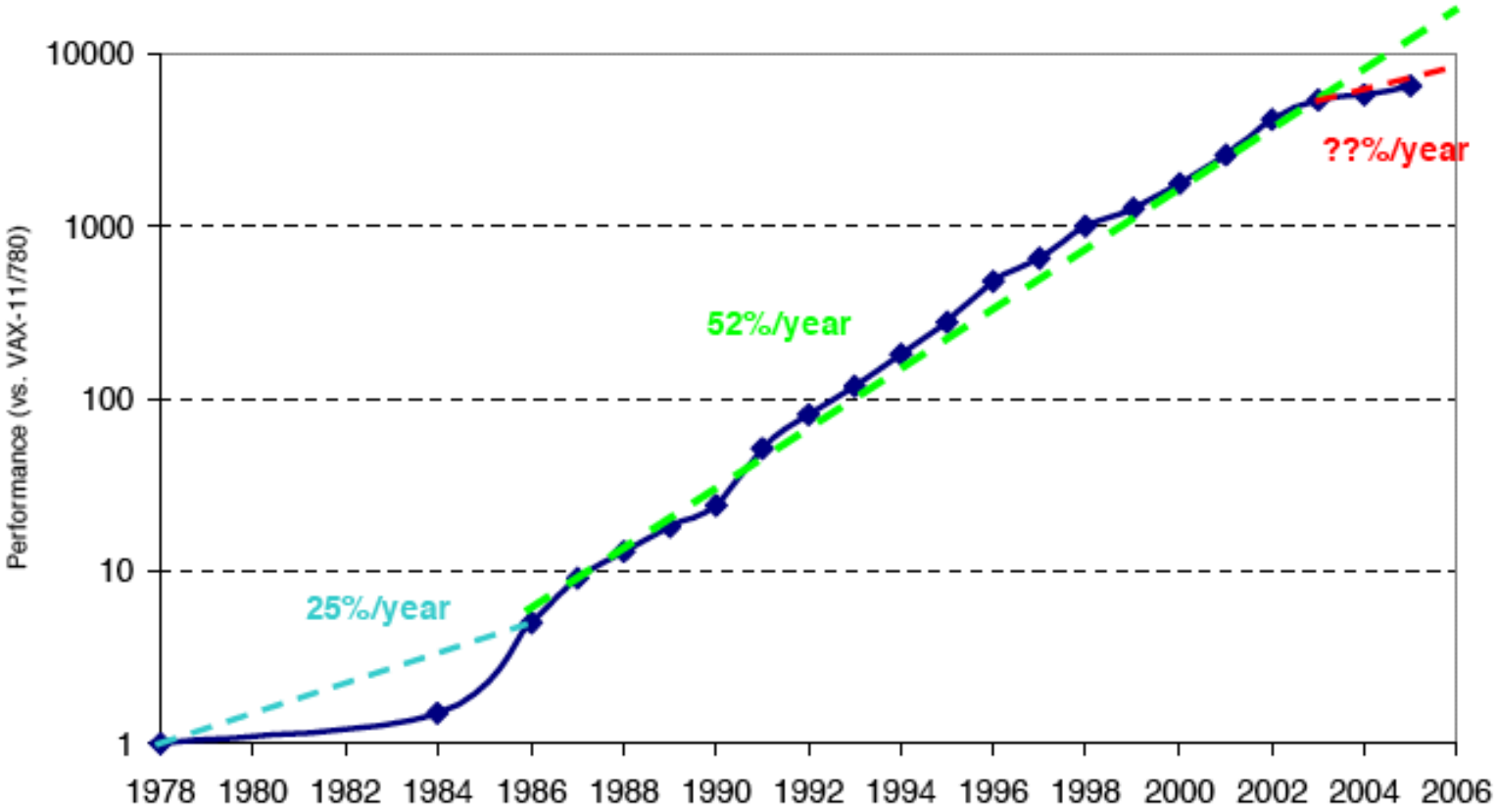


System performance improves ~50% per year.

Processor Performance (SPEC)

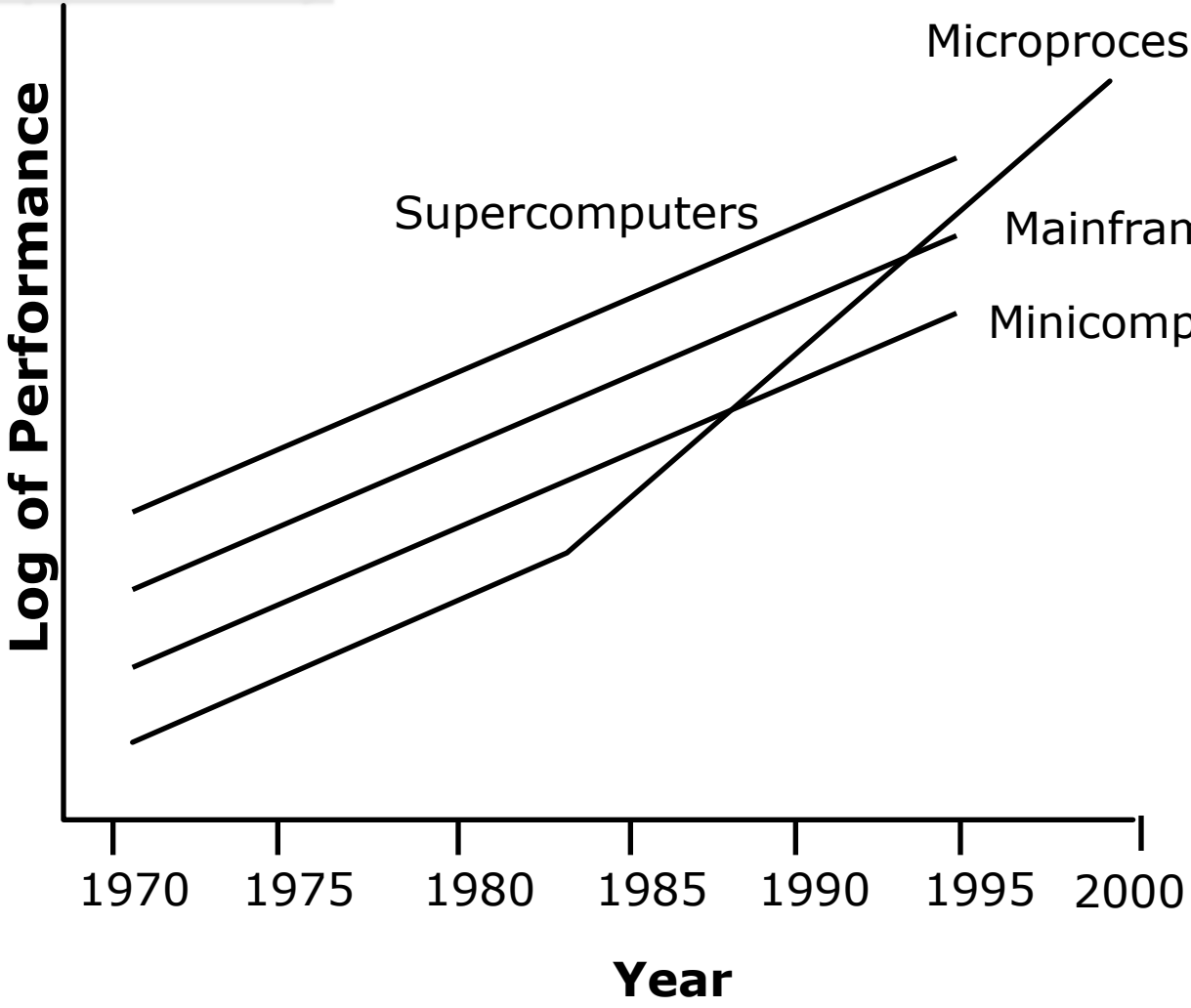


Processor Performance (SPEC)



< 20% improvement per year in the last decade.

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

Why Such Change?

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 …

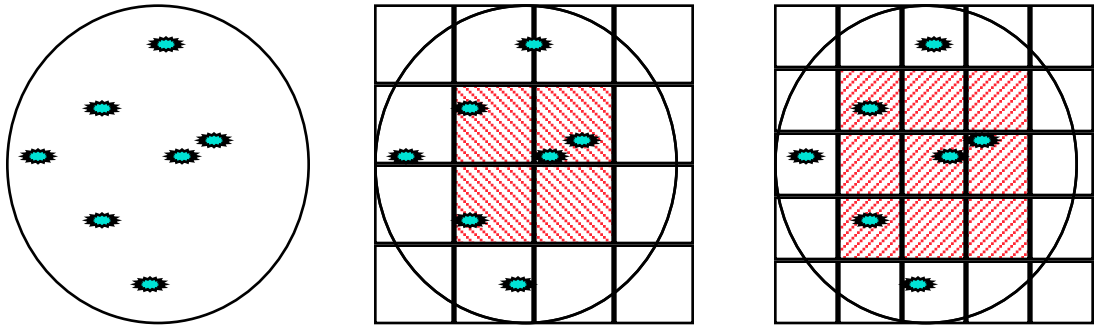
- Simpler development
  - CMOS VLSI: smaller systems, fewer components
- Higher volumes
  - CMOS VLSI : same dev. cost 1,000 vs. 100,000,000 units
- Lower margins by class of computer, due to fewer services

Function

- Rise of networking / local interconnection technology

Cost and Price

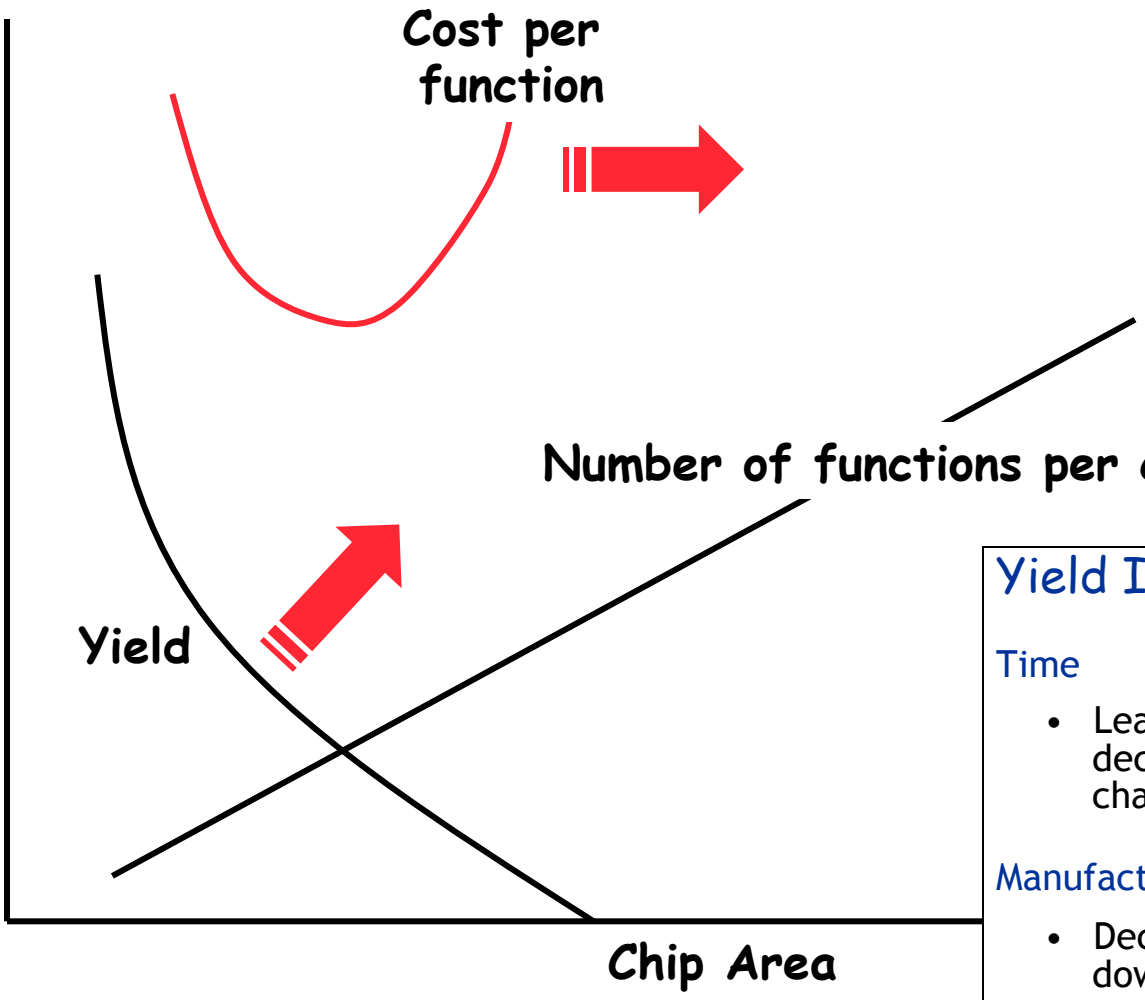
Integrated Circuit Manufacturing Costs



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

IC yield is 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

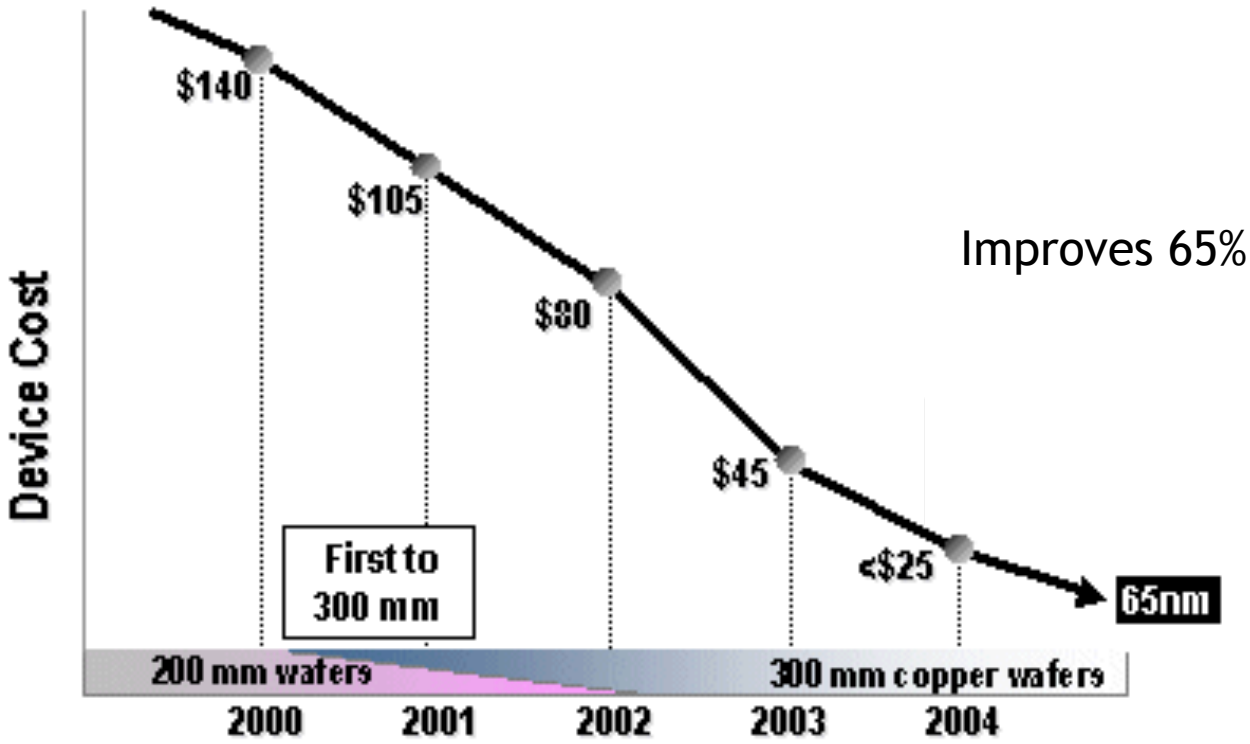
- Learning curve: manufacturing costs decrease over time measured by change in yield

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

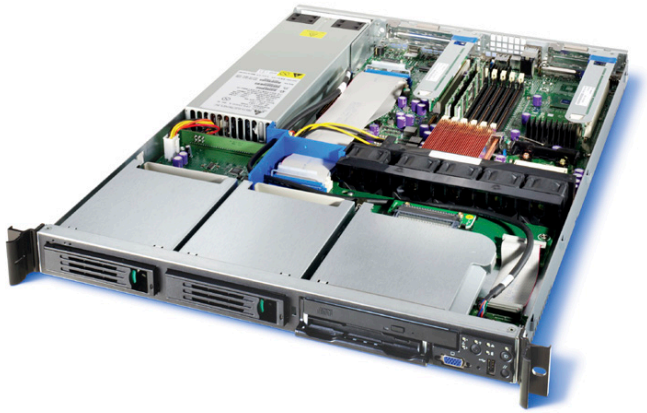


Improves 65% per year

Source: Xilinx

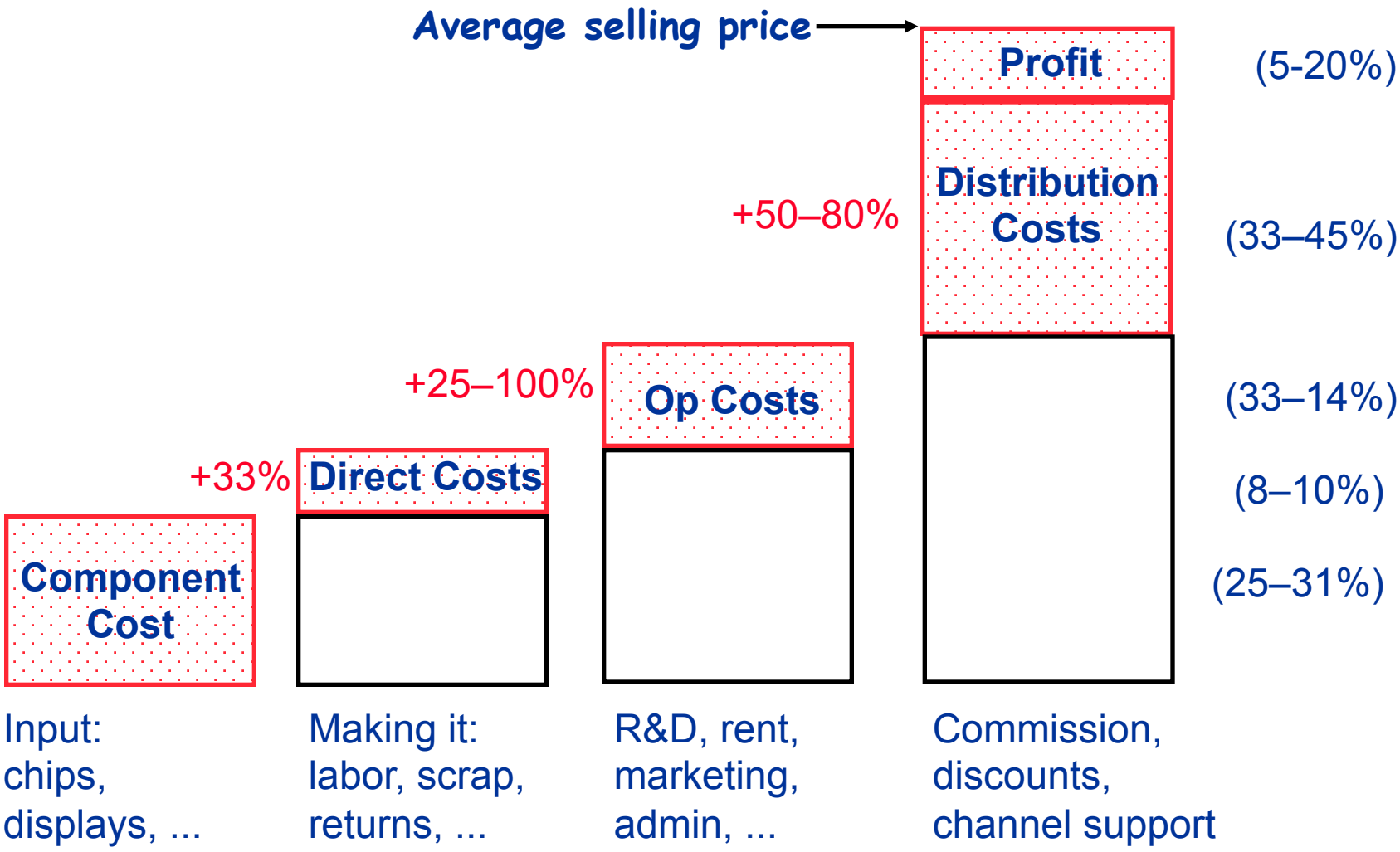
System Cost Example: Web Server

System	Subsystem	% of total cost
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%)
I/O Devices	Disks	36%
	(Subtotal)	(36%)



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