Welcome!

Introduction to Computer Architecture

(Computer Organization and Design: ARM Edition)

Instructor:

Abu Asaduzzaman (Zaman)

+1-316-978-5261

Abu.Asaduzzaman@wichita.edu

ARM – Advanced RISC Machines
RISC – Reduced Instruction Set Computing (a computer architecture)

Intro to Computer Architecture

Lecture 7 Presented by Md "Raihan" Uddin

Reading: See Reading Assignments on Blackboard

Tests: Quiz-1 (Week 4), Exam-1 (Week 5), HW-3 (Week 7), HW-4 (Week 8), …

- **Practice; Quiz-1; Handout 2a: Multilevel Computers;**
- **Intro to Structured Computer Organization (Handout 2b)**
  - **1.2: Milestones in Computer Architecture**
  - **1.3: The Computer Zoo**
  - **1.4: Example Computer Families**

CAPPLab
capplab@wichita.edu

Introduction to Computers

$$\text{CPU execution time for a program} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$$

1.6 Performance

Clock cycle time = 1 / Clock rate

- **The classic CPU performance equation**
  - $\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{Clock cycle time}$
  - $$\text{CPU time} = \frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}}$$

Instruction count: The number of instructions executed by the program.

- **Let's solve a problem.**
  - **Three types of instructions …**
  - **Which program**
    - **(a) is faster?**
    - **(b) executes the most instructions?**
    - **(c) has higher overall CPI?**

	Type A	Type B	Type C
CPI	1	2	3
Program 1	2	1	2
Program 2	4	1	1

$$\text{CPU clock cycles} = \sum_{i=1}^{n} (\text{CPI}_i \times \text{C}_i)$$

$$\text{CPU clock cycles}_1 = (2 \times 1) + (1 \times 2) + (2 \times 3) = 2 + 2 + 6 = 10 \text{ cycles}$$

$$\text{CPU clock cycles}_2 = (4 \times 1) + (1 \times 2) + (1 \times 3) = 4 + 2 + 3 = 9 \text{ cycles}$$

- **(a) 2; (b) 2; and (c) 1**

$$\text{CPI}_1 = \frac{\text{CPU clock cycles}_1}{\text{Instruction count}_1} = \frac{10}{5} = 2.0$$

$$\text{CPI}_2 = \frac{\text{CPU clock cycles}_2}{\text{Instruction count}_2} = \frac{9}{6} = 1.5$$

Practice

Which of the two computers in the table below is better? Explain you answer using the average clock cycles per instruction (CPI). Show all steps.

White Paper

Instruction Class	CPI for Class	Instruction Count	
		Program 1	Program 2
A	3	14	11
B	2	12	12
C	1	16	10

Non-White Paper

Instruction Class	CPI for Class	Instruction Count	
		Program 1	Program 2
A	2	11	14
B	3	12	15
C	1	10	16

Practice

Which of the two computers in the table below is better? Explain you answer using the average clock cycles per instruction (CPI). Show all steps.

White Paper

Instruction Class	*CPI for Class*	Instruction Count	
		Program 1	Program 2
A	*3*	14	11
B	*2*	12	12
C	*1*	16	10

Program-1: # clock cycles = 3x14 + 2x12 + 1x16 = 82; # instructions = 42;
Avg. CPI_1 = 82/42 = 1.95

Program-2: # clock cycles = 3x11 + 2x12 + 1x10 = 67; # instructions = 33;
Avg. CPI_2 = 67/33 = 2.03

Program-1 is better, because Avg. CPI_1 < Avg. CPI_2.

Practice

Which of the two computers in the table below is better? Explain you answer using the average clock cycles per instruction (CPI). Show all steps.

Non-White Paper

Instruction Class	*CPI for Class*	Instruction Count	
		Program 1	Program 2
A	*2*	11	14
B	*3*	12	15
C	*1*	10	16

Program-1: # clock cycles = 2x11 + 3x12 + 1x10 = 68; # instructions = 33;

Avg. CPI_1 = 68/33 = 2.06

Program-2: # clock cycles = 2x14 + 3x15 + 1x16 = 89; # instructions = 45;

Avg. CPI_2 = 89/45 = 1.98

Program-2 is better, because Avg. CPI_2 < Avg. CPI_1.

Introduction to Computers

Clock cycle time = 1 / Clock rate
Clock rate = 1 / Clock cycle time

1.6 Performance

- **CPU performance and its factors**
  - $\text{CPU execution time for a program} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$
  - $$\text{CPU execution time for a program} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

- **Let's solve a problem.**

Our favorite program runs in 10 seconds on computer A, which has a 2 GHz clock. We are trying to help a computer designer build a computer, B, which will run this program in 6 seconds. The designer has determined that a substantial increase in the clock rate is possible, but this increase will affect the rest of the CPU design, causing computer B to require 1.2 times as many clock cycles as computer A for this program. What clock rate should we tell the designer to target?

$$\text{CPU time}_A = \frac{\text{CPU clock cycles}_A}{\text{Clock rate}_A} \quad \Rightarrow \quad \text{CPU clock cycles}_A = 10 \text{ seconds} \times 2 \times 10^9 \frac{\text{cycles}}{\text{second}} = 20 \times 10^9 \text{ cycles}$$

$$\text{CPU time}_B = \frac{1.2 \times \text{CPU clock cycles}_A}{\text{Clock rate}_B} \quad \Rightarrow \quad \text{Clock rate}_B = \frac{1.2 \times 20 \times 10^9 \text{ cycles}}{6 \text{ seconds}} = \frac{0.2 \times 20 \times 10^9 \text{ cycles}}{\text{second}} = \frac{4 \times 10^9 \text{ cycles}}{\text{second}} = 4\text{GHz}$$

- **To run the program in 6 seconds, B must have twice the clock rate of A. ['A' 2 GHz, 'B' 4 GHz]**

Intro to Computer Architecture

Regarding Quiz-1

- **30 points, 30 minutes; class after quiz**
- **Includes Lectures 2-7**
  - ➢ Important Topics from lectures
    - ✓ Major steps to execute an instruction
  - ➢ Important Topics from Introduction to Computers (Ch. 1 from zyBooks)
    - ✓ Ch. 1 >> 1.1-1.9
- **Questions: Short questions (HW), Math-related, True/False, …**
- **Any not-allowed activities → Zero, etc. (applies to all tests)**

If you do not return your papers right after the STOP time,
I will not accept/grade your papers.

Intro to Computer Architecture

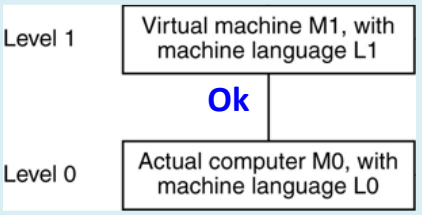
Do you have any questions?

11:26 AM

Intro to Computer Architecture

Multilevel Machines (+)

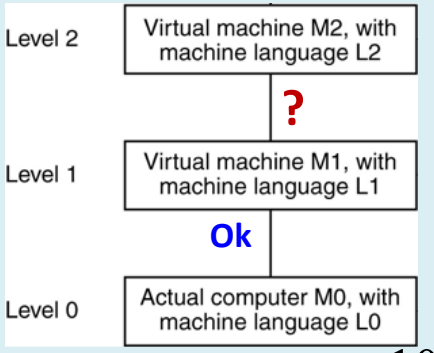
Problem: We want (1) the machine language to be very simple so as to reduce the complexity/cost ... and (2) convenient for people ...



POSSIBLE SOLUTION

Form a language (L2) of instructions that are more convenient to use by people than the built-in machine language (L1).

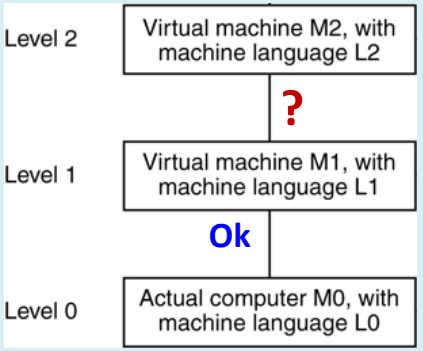
How can a program written in L2 be executed by the computer?



Intro to Computer Architecture

Multilevel Machines (+)

How can a program written in L2 be executed by the computer?



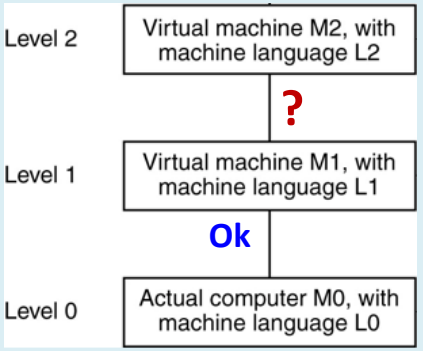
TRANSLATION

Replace each instruction in the L2 program by an equivalent sequence of instructions in L1. The new L1 program is then executed by the computer.

Intro to Computer Architecture

Multilevel Machines (+)

How can a program written in L2 be executed by the computer?

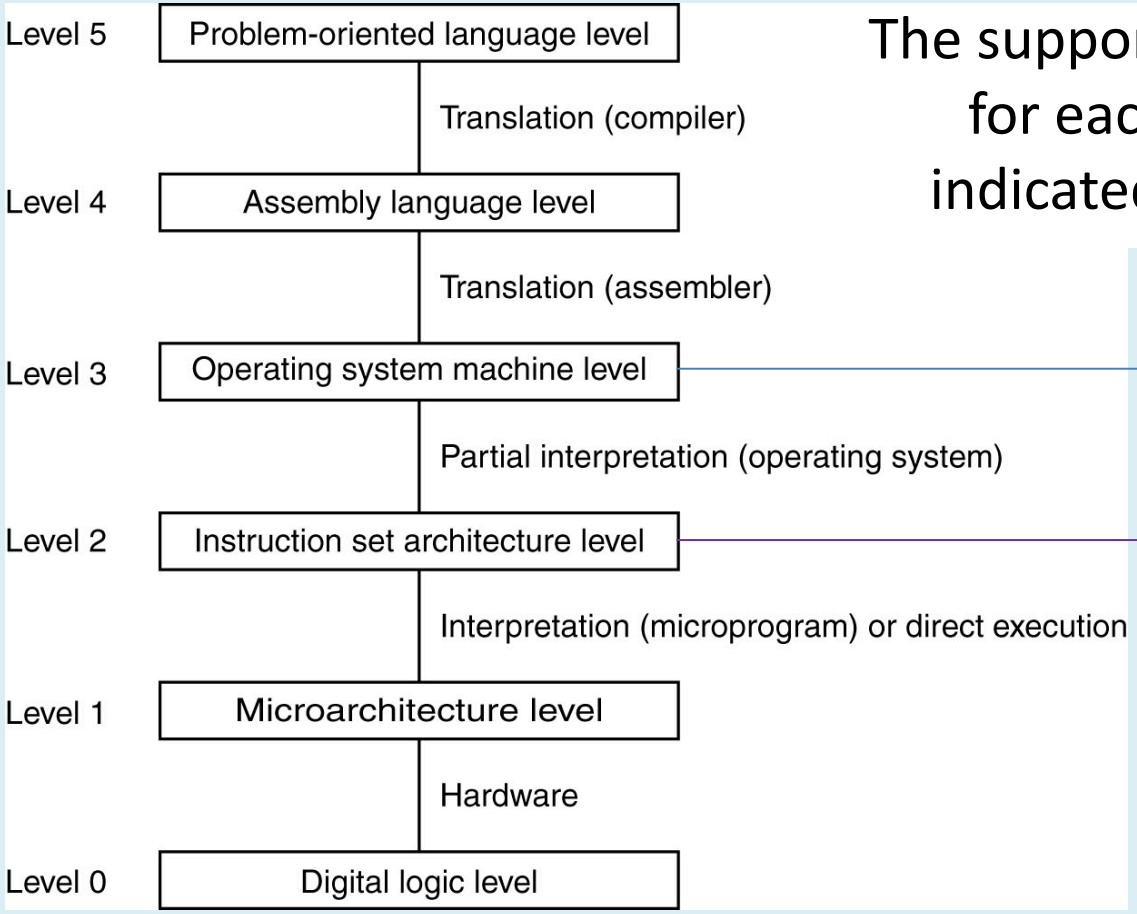


INTERPRETATION

An L1 program examines the L2 program instructions by instruction and executes the equivalent sequence of L1 instructions directly.

Intro to Computer Architecture

A Six-Level Machine GUI-Tools



The support method for each level is indicated below it.

Human Machine

Software Hardware

Solid State Physics

GUI – Graphical User Interface

Intro to Computer Architecture

Multilevel Machines

- In 1940s, two-level computer
  - ISA level (software)
  - Digital logic level (hardware)
- In 1950s, three-level computer
  - ISA level (software)
  - Microarchitecture level
    - Simplified hardware, complicated logic, design rectification, bug fix, cheap, flexible
  - Digital logic level (hardware)

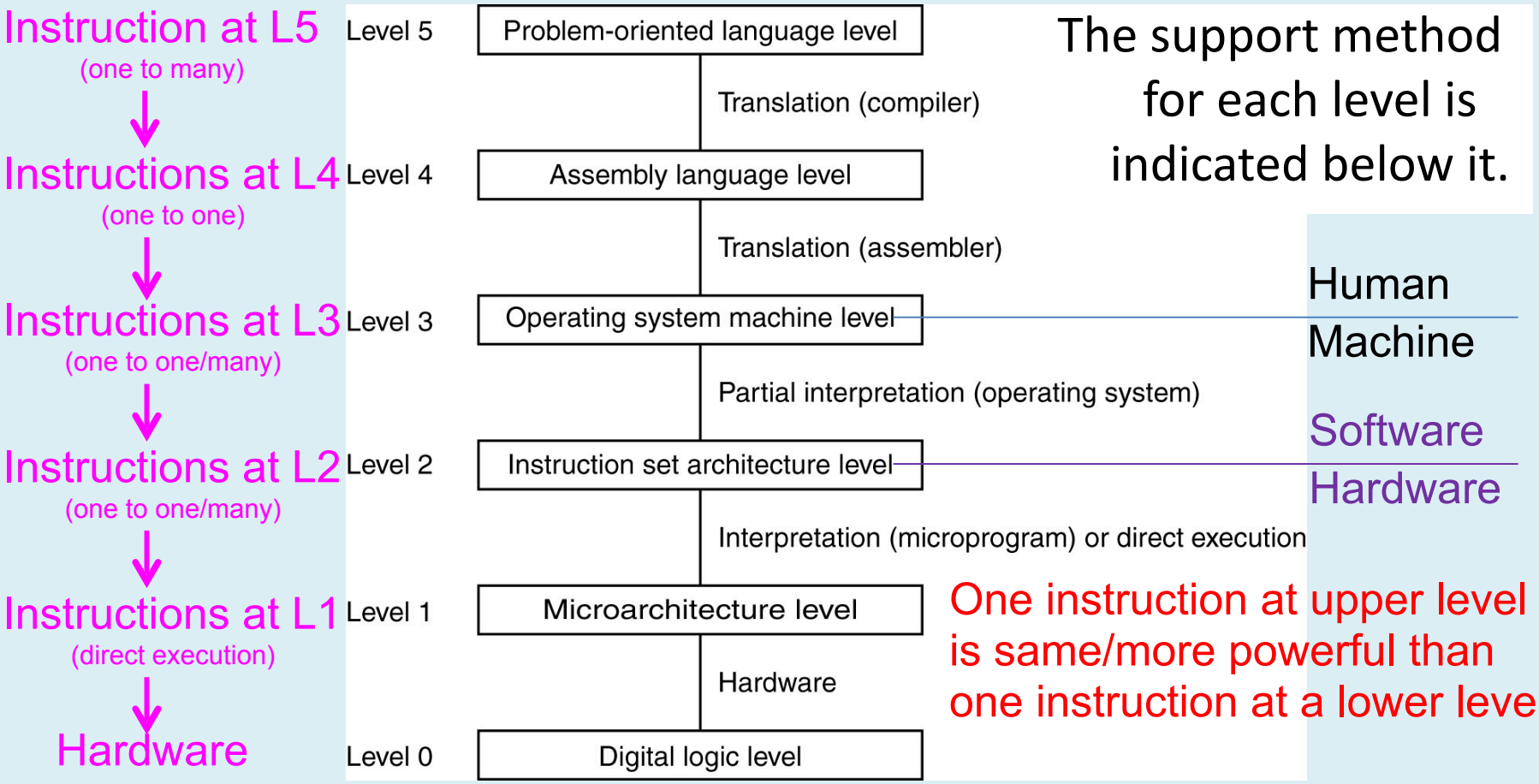
Intro to Computer Architecture

Multilevel Machines

- **In 1960s, four-level computer**
  - ➢ **Operating system machine level, 1960s, OS**
    - **Automate the operator's job (scheduling & load balancing)**
    - **More instructions/functionalities**
    - **Input / Output (I/O) operations**
    - **Multitasking (time-sharing)**
  - ➢ **ISA level (software), 1940s, machine's instruction set**
  - ➢ **Microarchitecture level, 1950s, registers, ALU, data path**
    - **Simplified hardware, complicated logic, design rectification, bug fix, cheap, flexible**
  - ➢ **Digital logic level (hardware), 1940s, gates**

Intro to Computer Architecture

A Six-Level Machine GUI-Tools

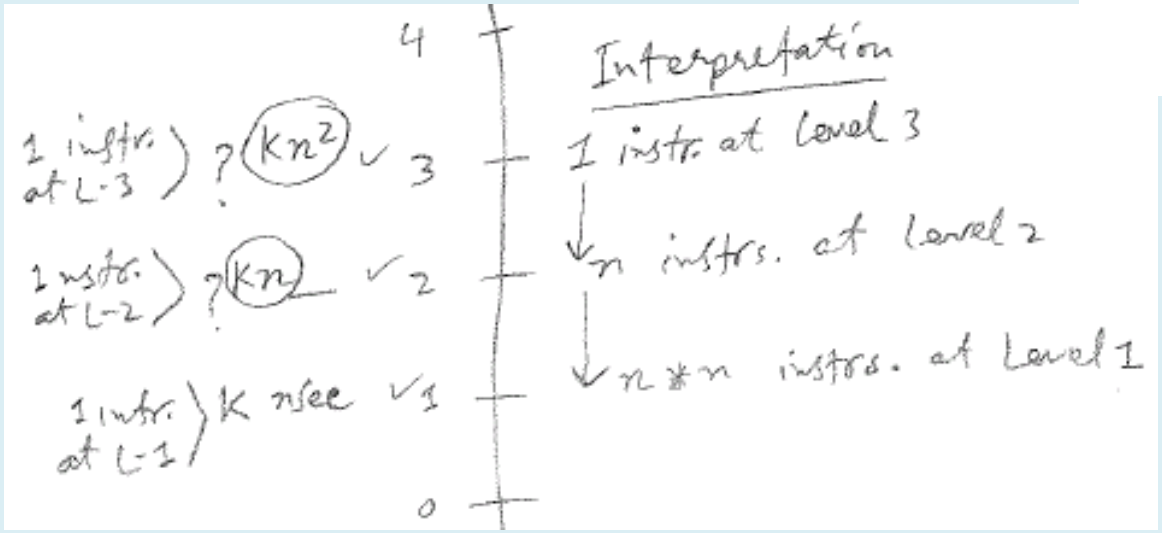


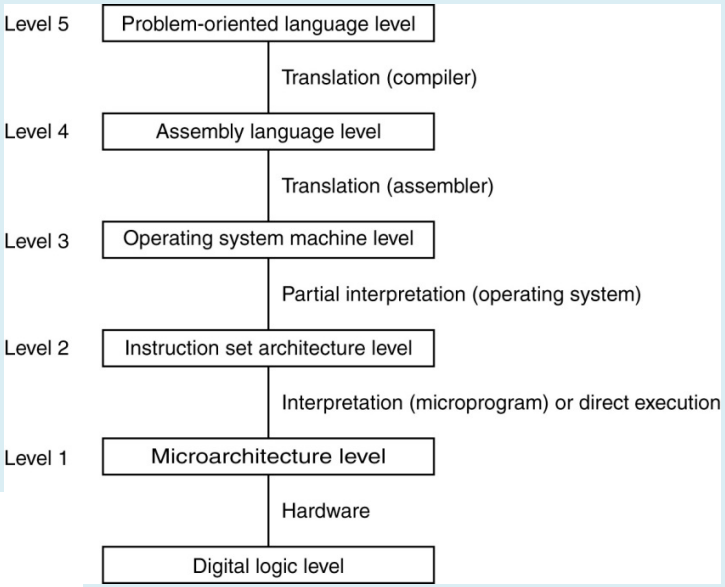
Solid State Physics

Intro to Computer Architecture

Example → Practice

- **Consider a computer with identical interpreters at levels 1, 2, and 3. It takes an interpreter n instructions to fetch, decode, and execute one instruction. A <u>level 1 instruction</u> takes k nanoseconds to execute. How long does it take for an instruction at levels 2 and 3?**





Intro to Computer Architecture

Do you have any questions?

11:26 AM

Intro to Computer Architecture

Lecture 7 Presented by Md "Raihan" Uddin

Reading: See Reading Assignments on Blackboard

Tests: Quiz-1 (Week 4), Exam-1 (Week 5), HW-3 (Week 7), HW-4 (Week 8), …

- Practice; Quiz-1; Handout 2a: Multilevel Computers;
- Intro to Structured Computer Organization (Handout 2b)
  - 1.2: Milestones in Computer Architecture
  - 1.3: The Computer Zoo
  - 1.4: Example Computer Families

CAPPLab
capplab@wichita.edu

Intro to Computer Architecture

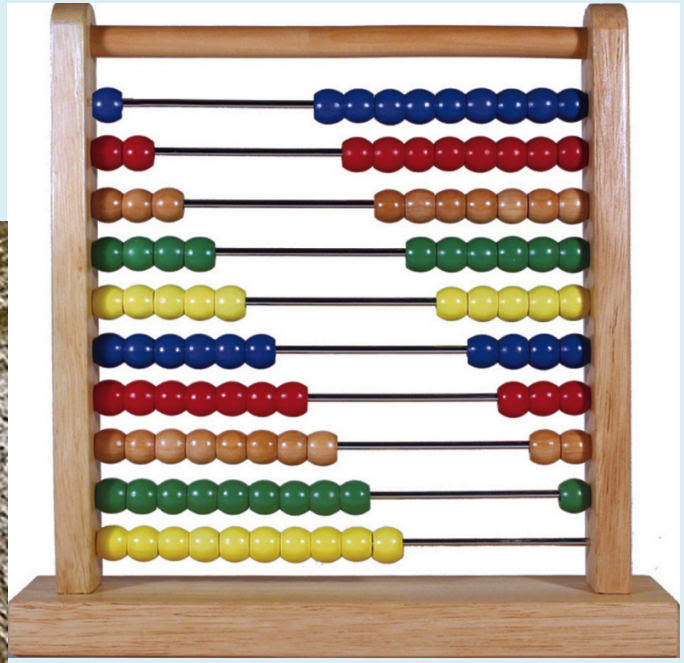
Computer Generations

- **Zeroth Generation**
  Mechanical Computers (1642 – 1945)
- **First Generation**
  Vacuum Tubes (1945 – 1955)
- **Second Generation**
  Transistors (1955 – 1965)
- **Third Generation**
  Integrated Circuits (1965 – 1980)
- **Fourth Generation**
  Very Large-Scale Integration (1980 – ?)
- **More Generations?**
  (we'll see)

Intro to Computer Architecture

Computer Generations (+)

- Before Zeroth Generation: (100 B.C. – 1642 A.D.)
  - 150-100 B.C. – Mechanical Abacus
  - 1613 A.D. – Word "Computer"



Intro to Computer Architecture

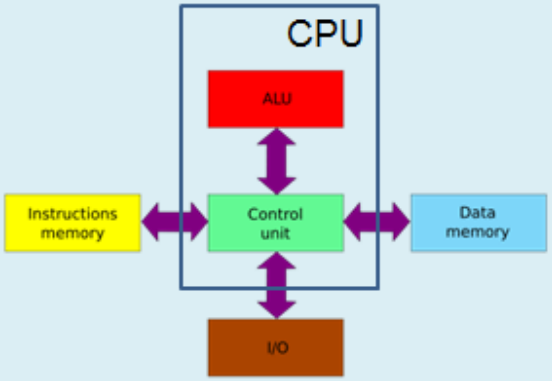
Computer Generations (+)

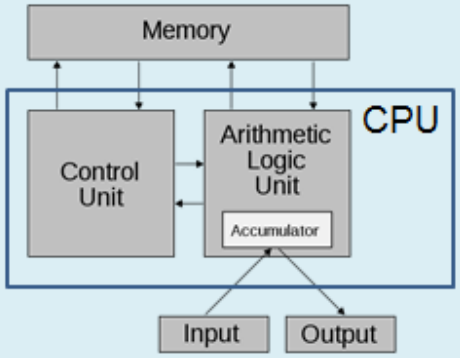
- **Zeroth Generation: Mechanical Computers (1642 – 1945)**
  - **First calculating machine by Pascal (1623-1662) in 1642**
  - **Babbage (1792-1871), "different engine", speedometer**
  - **"Analytical engine" – storage (memory) 1000 words of 50 decimal digits, mill (computation unit – add, subtract, multiply, divide), input section (punched card reader), and output section (punched and printed output)**

Intro to Computer Architecture

Computer Generations (+)

- **First Generation: Vacuum Tubes (1945 – 1955)**
  - **Enigma (Thomas Jefferson, former U.S. president)**
  - **Vacuum tubes – ENIAC (electronic numerical integrator and computer), 18,000 VTs and 1,500 relays, 30 tons, 140 KW power**
  - **The original von Neumann machine – CPU, memory (unified), I/O**
  - **The Harvard machine – CPU, memory (split into Instruction and Data), I/O**

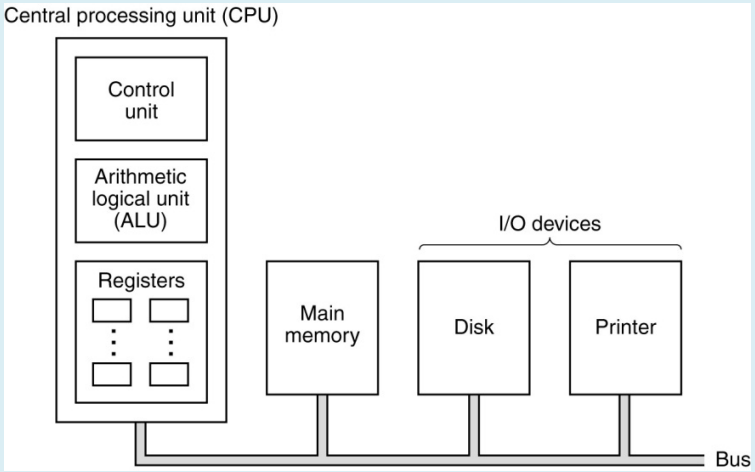




Intro to Computer Architecture

Computer Generations (+)

- **Second Generation: Transistors (1955 – 1965)**
  - **Transistor – tiny electronic switch**
  - **Size, power, cost**
  - **Programmed Data Processor (PDP)**
  - **MIT, TX-0**
  - **IBM 7094**
  - **PDP-1, MIT**
  - **Concept of supercomputers (Cray)**



Intro to Computer Architecture

Computer Generations (+)

- **Third Generation: Integrated Circuits (1965 – 1980)**
  - **Integrated circuit (IC) – dozens of transistors in a chip**
  - **IBM System/360, multiprogramming**

Intro to Computer Architecture

Computer Generations (+)

- **Fourth Generation: Very Large-Scale Integration (1980 – ?)**
  - **VLSI – millions of transistors in a chip**
  - **IBM Personal Computer (PC), Intel CPU**
  - **Graphical User Interface (GUI)**
  - **FPGA (field programmable gate array)**

Intro to Computer Architecture

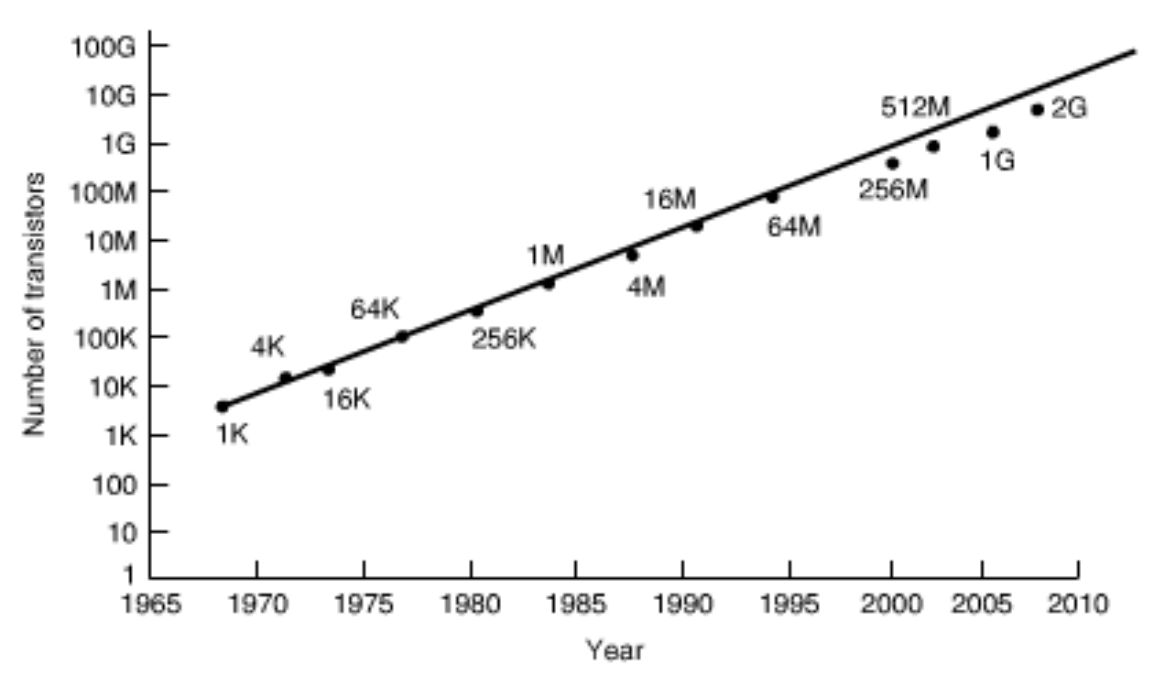
Computer Generations

- **Zeroth Generation**
  Mechanical Computers (1642 – 1945)
- **First Generation**
  Vacuum Tubes (1945 – 1955)
- **Second Generation**
  Transistors (1955 – 1965)
- **Third Generation**
  Integrated Circuits (1965 – 1980)
- **Fourth Generation**
  Very Large-Scale Integration (1980 – ?)
- **More Generations?**

Intro to Computer Architecture

Moore's law and processor design

- Gordon Earle Moore: Co-founder and Chairman Emeritus of Intel Corp.
- **Moore's Law** predicts a 60-percent annual increase in the number of transistors that can be put on a chip. *(transistors doubles every two years)*



Intro to Computer Architecture

Moore's law and processor design (+)

- What's the importance/significance of Moore's law?
- More transistors – larger memory, powerful processor
- In the past 50 years or so, semiconductor industry is moving ahead faster than any other
- In the past 50 years or so, computer/processor industry is moving ahead faster
- Design more powerful computers at the same or lower price

Intro to Computer Architecture

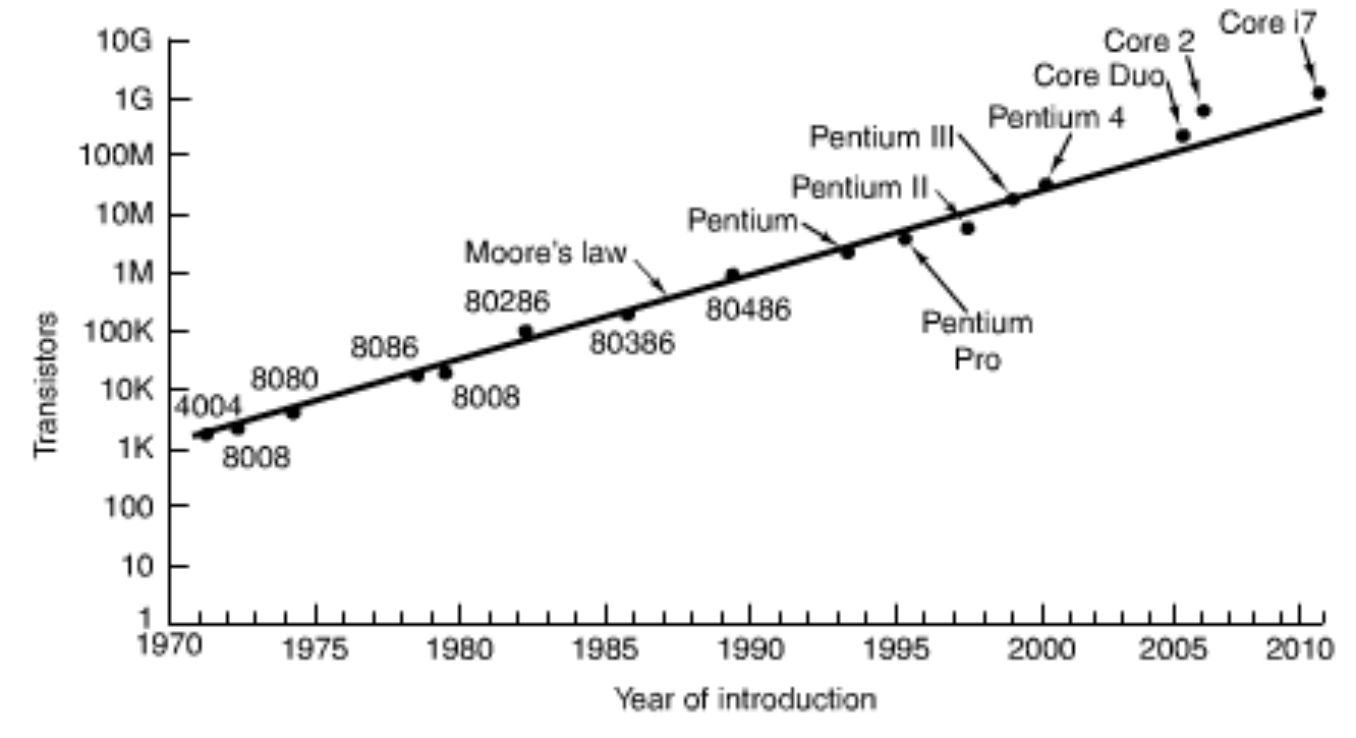
Moore's law and processor design (+)

- Let's understand the Moore's law?
- Moore's Law – computer/transistors density (not performance)
  - Reinterpreted – clock speed does not increase
  - Reinterpreted – cores/chip doubles every two years
  - ~~Reinterpreted – deal with millions of concurrent threads~~
  - ~~Reinterpreted – deal with inter-chip and intra-chip parallelism~~
  - Reinterpreted – Moore's Law is not forever...
- ~~Moore's 2nd (Rock's) Law --> costs double every 4 years~~

Intro to Computer Architecture

Moore's law and processor design (+)

- **Moore's Law** predicts a 60-percent annual increase in the number of transistors that can be put on a chip. Different chips/processors.



Intro to Computer Architecture

Computer Generations (cont'd, +)

- **Zeroth Generation**
  Mechanical Computers (1642 – 1945)
- **First Generation**
  Vacuum Tubes (1945 – 1955)
- **Second Generation**
  Transistors (1955 – 1965)
- **Third Generation**
  Integrated Circuits (1965 – 1980)
- **Fourth Generation**
  Very Large-Scale Integration (1980 – ?)
- **Fifth Generation**
  (a) Low-Power and Invisible Computers (1980 – ?)
  (b) Multicore Computers (2000 – ?)

Intro to Computer Architecture

Computer Generations (+)

- **Fifth Generation: Low-Power and Invisible Computers (1980 – ?)**
  - **Embedded systems**
  - **Internet of Things (IoT), Artificial Intelligence,**
  - **GridPad (first Tablet from Grid Systems, 1989)**
  - **PDA (personal digital assistant), Smartphone (Simon)**
  - ~~Ubiquitous/pervasive/everyware computing is a post-desktop model of human-computer interaction in which information processing has been thoroughly integrated into everyday objects and activities. Example: refrigerators "aware" of their suitably tagged contents, able to both plan a variety of menus from the food actually on hand, and warn users of stale or spoiled food.~~

Intro to Computer Architecture

Computer Generations (+)

- **Fifth Generation: Multicore Computers (2000 – ?)**
  - **VLSI – millions of transistors in a chip**
  - **2001, POWER4, first dual-core chip microprocessor, IBM**
  - **Advanced VLSI – billions of transistors in a chip (2008+)**
  - **System-on-a-chip (SoC)**
  - **Network-on-a-chip (NoC)**
  - **Multicore/many-core systems**

Intro to Computer Architecture

Do you have any questions?

11:26 AM

Intro to Computer Architecture

The Computer Zoo

The Types/Classes of Computers

- **Current spectrum (i.e., categorization) of computers**

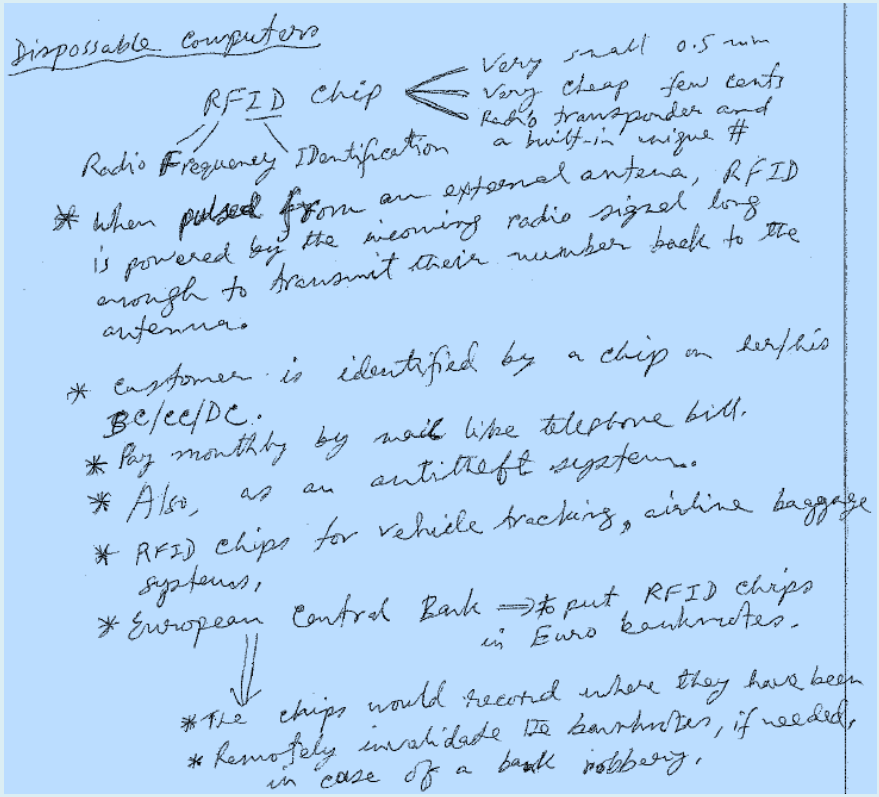
Type	Price ($)	Example application
Disposable computer	0.5	Greeting cards
Microcontroller	5	Watches, cars, appliances
Game computer	50	Home video games
Personal computer	500	Desktop or notebook computer
Server	5K	Network server
Collection of Workstations	50–500K	Departmental minisupercomputer
Mainframe	5M	Batch data processing in a bank

- **How about Supercomputers?**

CS 394 (Spring 2016): Intro to Computer Architecture

Disposable Computers

- **RFID (Radio Frequency Identification)**
  - ➢ Very small, very cheap
  - ➢ Radio transponder, a built-in unique #
  - ➢ External antenna sends pulse; RFID is powered up to transmit its unique #
  - ➢ Customer can be identified by Bank Card / Credit Card, etc.
  - ➢ Customer can pay bill monthly
  - ➢ RFID in antitheft systems



CS 394 (Spring 2016): Intro to Computer Architecture

Disposable Computers

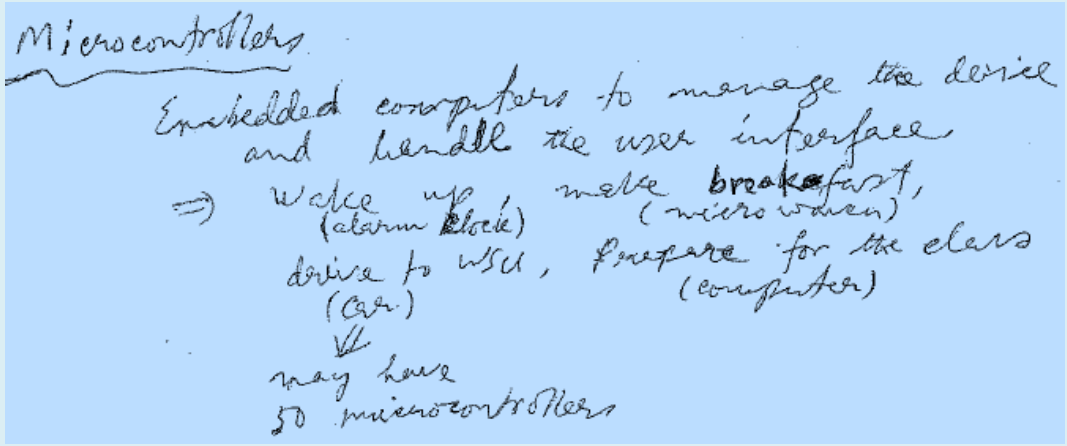
Example → Practice

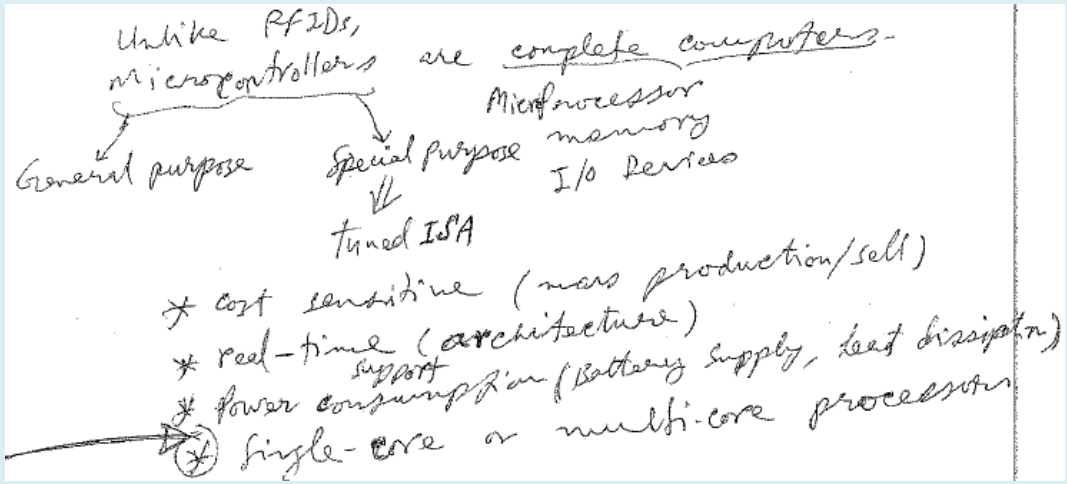
- **Suppose that, in an average, 300,000 people in Wichita consumes two packages of goods a day bearing RFID tags. How many RFID tags have to produce annually to meet the demand? At a penny a tag, what is the total cost of the tags?**
- **?**

CS 394 (Spring 2016): Intro to Computer Architecture

Microcontrollers

- Controls devices such as alarm clock, micro oven, etc.
- Hardware/Software Systems
- Embedded Systems
- Unlike RFID, microcontrollers are complete computer
- Cost-sensitive
- Real-time response

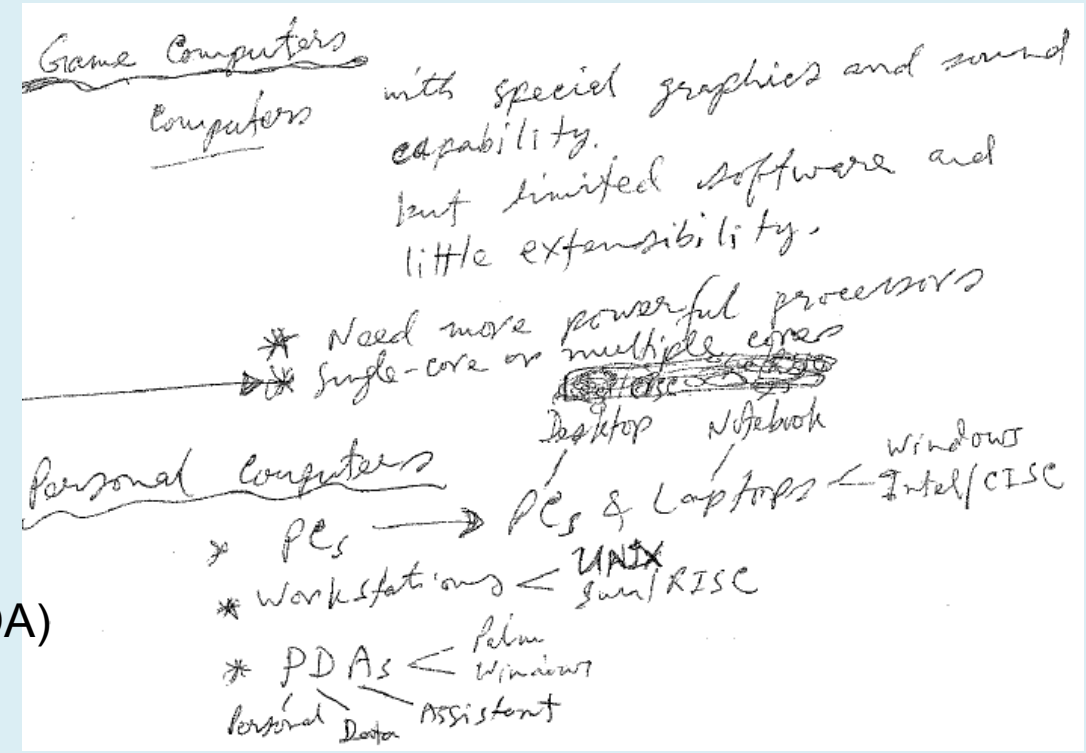




CS 394 (Spring 2016): Intro to Computer Architecture

Game Computers and Personal Computers

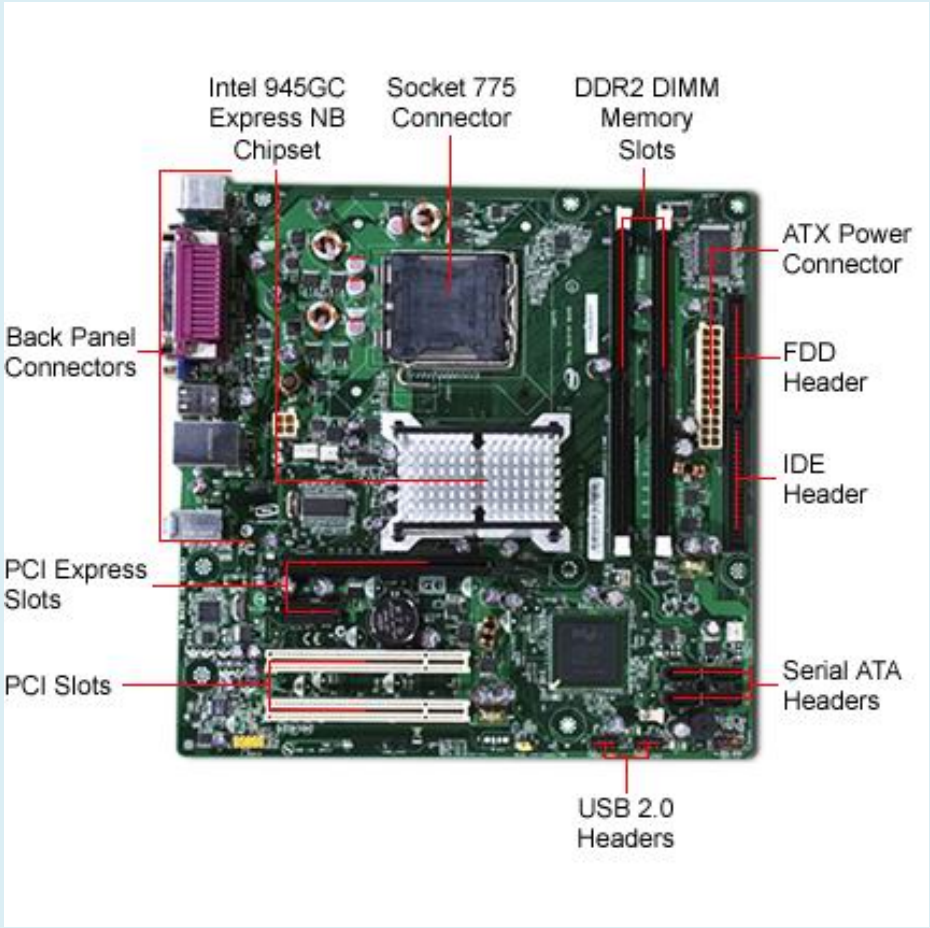
- **Game Computer**
  - ➢ Special graphics and sound cards
  - ➢ Need powerful processors
- **Personal Computer**
  - ➢ PC, Laptop, Tablet, CISC
  - ➢ Workstation, Server, RISC
  - ➢ Personal Data Assistant (PDA)



CS 394 (Spring 2016): Intro to Computer Architecture

A Printed Circuit Board (PCB) of a Desktop PC

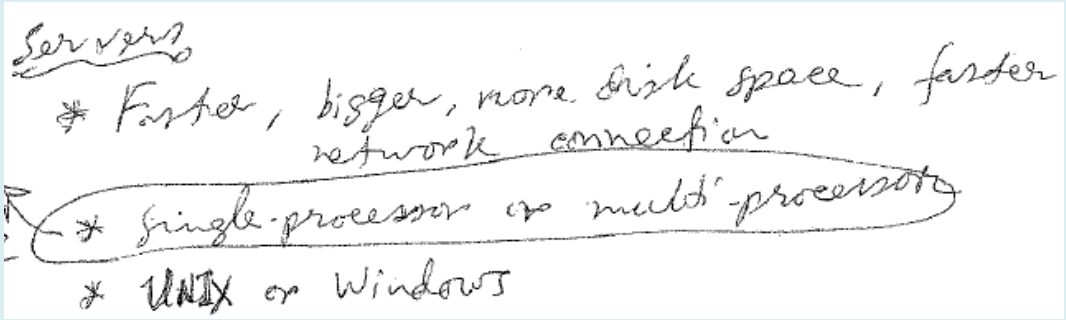
- FDD – Floppy Disk Drive ?
- IDE – Integrated Drive Electronics
- ATA – Advanced Technology Attachment
- USB – What is USB?
  - ➢ Universal Serial Bus

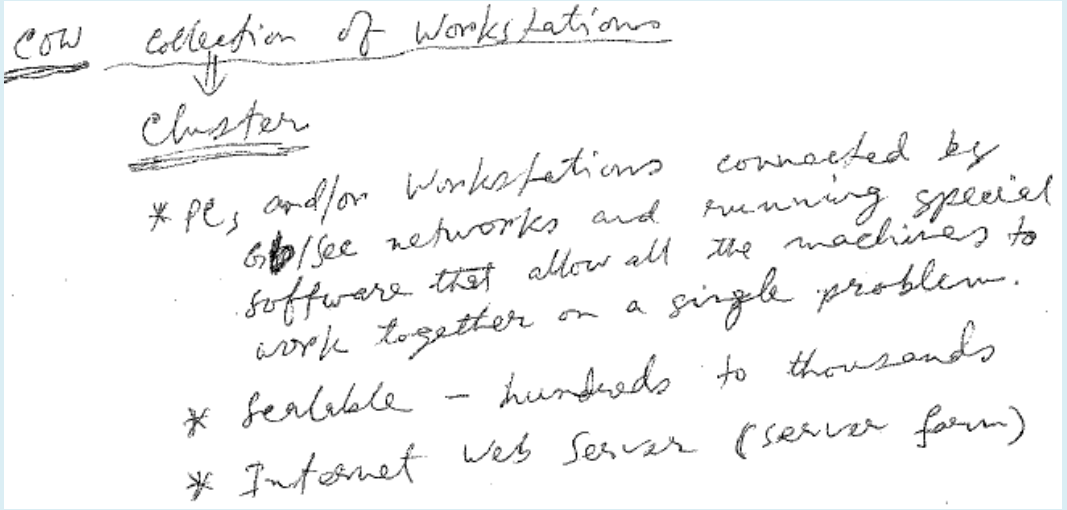


CS 394 (Spring 2016): Intro to Computer Architecture

Servers and Cluster of Workstations (COW)

- **Servers**
  - Faster, more disk space, faster network connection
  - Multi-user multi-tasking
- **Collection of Workstations**
  - Cluster
  - Scalable
  - Internet web server

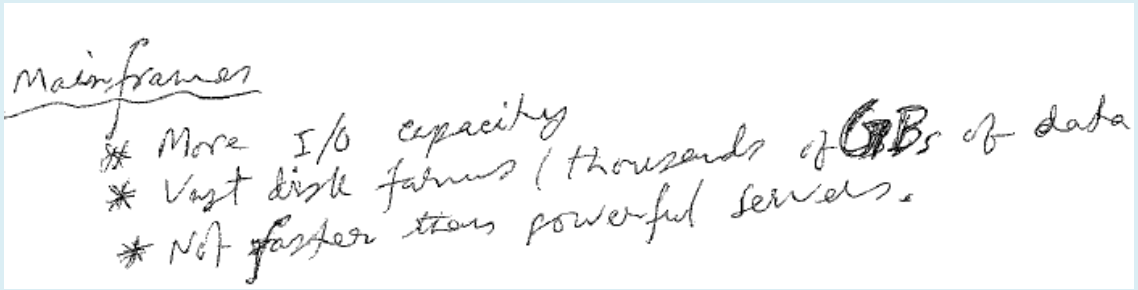




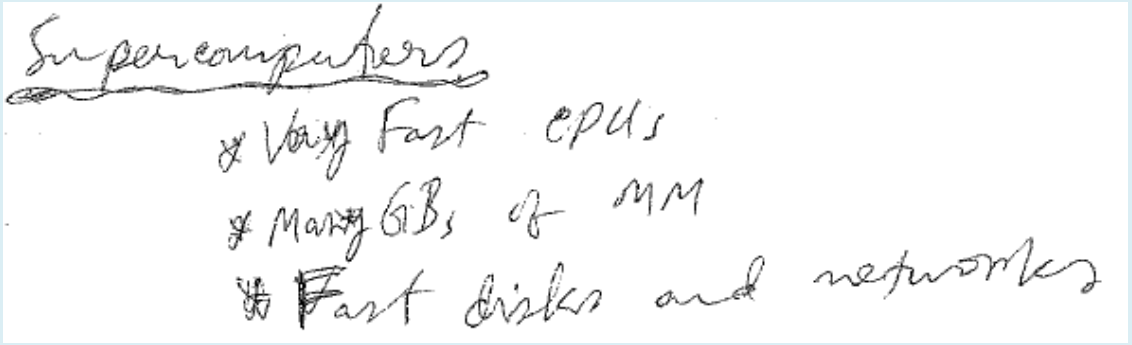
CS 394 (Spring 2016): Intro to Computer Architecture

Mainframes and Supercomputers

- **Mainframes**
  - More I/O capacity
  - Vast disk space
  - Not faster that Server



- **Supercomputers**
  - Very fast CPU
  - Many GBs of memory
  - Fast disks and network
  - Many GPU cards



CS 394 (Spring 2016): Intro to Computer Architecture

Handheld Mobile Devices

- PDA (personal digital assistant or personal data assistant) – aka palmtop
- Mobile phones
- Smartphone, iPhone, 3G, 4G, 5G, 6G?
- Tablet PC
- Many more

- Some issues: Limited resources, bandwidth, and power supply

ECE 394

Introduction to Computer Architecture

Tentative Schedule

Week Tue	Note	Important topics/readings, assignments, due dates, and reminders are listed here so that you can organize your time and academic work.
1 08/20		ECE 394: Intro to Computer Architecture, Syllabus; K-probe; zyBook 1.1 (Intro to Computers); Homework, Quiz, and Exam;
2 08/27	HW-1	HW-1 Discussion; zyBook 1.2-1.5 (eight ideas, processors); HW-1 (due on Blackboard); zyBook 1.6 (performance);
3 09/03	HW-2	9/02 (Labor Day) No Class/Lab; HW-2 (Bb); zyBook 1.7-1.9 (… uni- and multiprocessors, Core i7);
4 09/10	Quiz-1	Quiz-1 Discussion; Handout: Multilevel Computers; Quiz-1 (class test, 30-min / 30-pts, closed book);
5 09/17	Exam-1	Exam-1 Discussion; Handout: Computer Generations; Exam-1 (class test, 65-min / 65-pts, closed book);
6 09/24	Update	zyBook: 3.1 (The Processor: Introduction); zyBook: 3.2-3.3 (The Processor: Datapath, Pipelining);
7 10/01	HW-3	zyBook 3.4-3.5 (Data hazards: Forwarding versus stalling); HW-3 (Bb); zyBook 3.6 (Data hazards and Control hazards);
8 10/08	Mid-Pt HW-4	zyBook 3.7 (Parallelism via instructions); HW-4 (Bb); zyBook 3.8 (Going faster: ILP and matrix multiply);
9 10/15	Fal-Brk Quiz-2	10/12 (Sat) to 10/15 (Tue) (Fall Break) No Class; Quiz-2 (class test, 30-min / 30-pts, closed book);
10 10/22	Exam-2	Exam-2 Discussion; zyBook 4.1 (Memory Hierarchy: Introduction); Exam-2 (class test, 65-min / 65-pts, closed book);
11 10/29	Update	zyBook 4.2-4.3 (Memory Hierarchy: Caches); zyBook 4.4-4.5 (Memory Hierarchy: Virtual memory);
12 11/05	HW-5	zyBook 5.1 (Parallel Processors: Introduction); HW-5 (Bb); zyBook 5.2 (Difficulty of Parallel Processing);
13 11/12	HW-6	zyBook 5.3 (SISD, MIMD, SIMD, SPMD, and vector); HW-6 (Bb); zyBook 5.4 (Hardware multithreading);
14 11/19	Quiz-3	zyBook 5.5-5.6 (Multicore processors, graphics processing units); Quiz-3 (class test, 30-min / 30-pts, closed book);
15 11/26	Thx-Brk	Future of Computers (selected materials); 11/27 (Wed) to 12/01 (Sun) (Thanksgiving Break) No Class;
16 12/03	Exam-3	Exam-3 Discussion; Exam-3 (class test, 65-min / 65-pts, closed book);
Finals		None!
Note: A date in Column 1 indicates the Tuesday of that week. Here, 12/03 is Tueday of Week 16.		