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 5

Reading: See Reading Assignments on Blackboard

Tests: HW-2 (Week 3), Quiz-1 (Week 4), Exam-1 (Week 5), …

- **Quiz and Exam; zyBook Ch 1.1-1.6;**
- **Introduction to Computers (from zyBooks)**
  - **1.7 The power wall**
  - **1.8 The switch from uniprocessor to multiprocessor**
  - **1.9 The Intel Core i7**
- **Intro to Structured Computer Organization (Handout 2a)**
  - **1.1.1: Languages, Levels, and Virtual Machines**
  - **…**

Intro to Computer Architecture

Regarding Assignments

- **zyBooks Grading …**

ECE 394, Introduction to Computer Architecture, Fall, 2024
(Computer Organization and Design: The Hardware Software Interface)

Reading Assignments

Week	Assignment	Important Topics
1	1	zyBook Ch. 1.1: Introduction (Introduction to Computers) Ch. 1.2 – Ch. 1.4: Ch. 1.5: Technologies for building processors and memory
2	2	zyBook Ch. 1.6: Performance Ch. 1.7 – Ch. 1.8: Ch. 1.9: Intel Core i7
3	3	Handout 2a: Introduction to Structured Computer Organization 1.1.1: Languages, Levels, and Virtual Machines 1.1.2: Contemporary Multilevel Machines 1.1.3 Evolution of Multilevel Machines
4	4	Handout 2b: Introduction to Structured Computer Organization 1.2: Milestones in Computer Architecture 1.3: The Computer Zoo 1.4: Example Computer Families
5	5	zyBook Ch. 3.1: Introduction (The Processor) Ch. 3.2: Building a datapath
6	6	zyBook Ch. 3.3: An overview of pipelining Ch. 3.4: - Ch. 3.5 Ch. 3.6: Control hazards

1

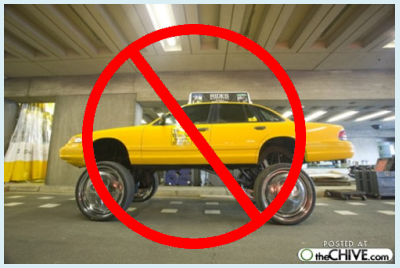
Week	Assignment	Important Topics
7	7	zyBook Ch. 3.7: Parallelism via instructions Ch. 3.8: Instruction-level parallelism
8	8	zyBook Ch. 4.1: Introduction (Memory Hierarchy)
9	9	zyBook Ch. 4.2: Memory technologies Ch. 4.3: The basics of caches Ch. 4.4: Virtual machines
10	10	zyBook Ch. 4.5: Virtual memory
11	11	zyBook Ch. 5.1: Introduction (Parallel Processors) Ch. 5.2: The difficulty of creating parallel processing programs
12	12	zyBook Ch. 5.3: SISD, MIMD, SIMD, SPMD, and vector zyBook Ch. 5.4: Hardware multithreading
13	13	zyBook Ch. 5.5: Multicore and other shared memory multiprocessors Ch. 5.6: Introduction to graphics processing units
14	None	None
15	None	None | All Readings Must Be Completed by Week 15!
16	None	None

2

General Discussion on Quiz / Exam

Quiz and Exam

- In-person in classroom, at the beginning of the class
- Closed book, by definition.
- One-page (8.5 by 11 inches) both-sided personal notes is recommended.
- **Bring your pen/pencil, calculator, etc. (no connected device)**
- Read/understand all questions.
- **Tell the exam proctor your concerns; don't ask others.**
- Make reasonable assumptions if needed.
- Answer right to the point.
- **Show all steps with your solutions for full/partial credits.**
- PRINT your name and WSU ID on test papers.
- **Submit your papers on time. Very Important!!!**



One month is:
~~28 days~~
~~29 days~~
30 days
~~31 days~~

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?

8:48 AM

Intro to Computer Architecture

Regarding Exam-1

- **65 points, 65 minutes; no class after exam**
- **Includes Lectures 2-8**
  - ➢ Introduction to Computers (Ch. 1 from zyBooks)
    - ✓ Major steps to execute an instruction
    - ✓ Ch. 1 >> 1.1-1.9
  - ➢ Intro to Structured Computer Organization (Handout 2a & 2b)
    - ✓ Handout 2a & 2b >> 1.1-1.4 (as covered)
- **Questions: Short questions (HW, Quiz), 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?

8:48 AM

Introduction to Computers

1.6 Performance

- **What is performance of computers?**
  - ➢ **(depends on several factors)**
- **Response time**
  - ➢ **Aka, execution time. The total time required for the computer to complete a task.**
- **Performance**
  - ➢ **Performance = 1 / (Execution time)**
- **If computer A runs a program in 10 sec and computer B runs the same program in 15 sec, compare A and B?**
  - ➢ **Performance of A / Performance of B**
    = Execution time of B / Execution time of A = 15 / 10 = 1.5
  - ➢ **Computer A is 1.5 times faster than Computer B.**

Introduction to Computers

1.6 Performance

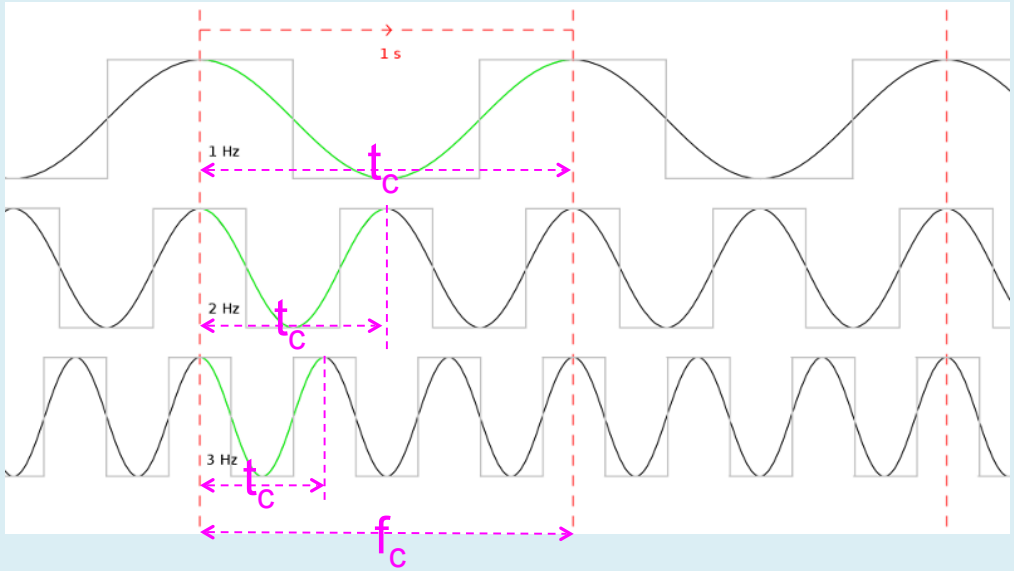
- **CPU execution time**
  - ➢ **Also called CPU time. The actual time the CPU spends computing for a specific task.**
- **User CPU time**
  - ➢ **The CPU time spent in a program itself.**
- **System CPU time**
  - ➢ **The CPU time spent in the operating system performing tasks on behalf of the program.**
- **Clock cycle**
  - ➢ **Also called cycle or tick. The time for one clock period, which runs at a constant rate.**
- **Clock period**
  - ➢ **The length of each clock cycle.**

Introduction to Computers

Clock: Cycle: Time and Rate

- **Relationship between clock/cycle time and clock rate**
  - ➢ **Clock rate (frequency) > clock frequency f_c**
  - ➢ **Clock rate f_c in hertz (Hz). [1 Hz = 1 cycle / sec]**
  - ➢ **Clock cycle time > time for one cycle t_c**
  - ➢ **So, $t_c = 1/ f_c$**
  - ➢ **And, $f_c = 1/ t_c$**

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



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]**

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

- **Instruction performance**
  - $\text{CPU clock cycles} = \text{Instructions for a program} \times \text{Average clock cycles per instruction}$
  - **Clock cycles per instruction (CPI): Average number of clock cycles per instruction for a program**
- **Let's solve a problem.**

Suppose we have two implementations of the same instruction set architecture. Computer A has a clock cycle time of 250 ps and a CPI of 2.0 for some program, and computer B has a clock cycle time of 500 ps and a CPI of 1.2 for the same program. Which computer is faster for this program and by how much?

$$\text{CPU clock cycles}_A = I \times 2.0$$

$$\text{CPU clock cycles}_B = I \times 1.2$$

$$\begin{aligned}\text{CPU time}_A &= \text{CPU clock cycles}_A \times \text{Clock cycle time} \\ &= I \times 2.0 \times 250 \text{ ps} = 500 \times I \text{ ps}\end{aligned}$$

$$\text{CPU time}_B = I \times 1.2 \times 500 \text{ ps} = 600 \times I \text{ ps}$$

$$\frac{\text{CPU performance}_A}{\text{CPU performance}_B} = \frac{\text{Execution time}_B}{\text{Execution time}_A} = \frac{600 \times I \text{ ps}}{500 \times I \text{ ps}} = 1.2$$

- **Computer A is 1.2 times as fast as computer B for this program.**

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

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.

Intro to Computer Architecture

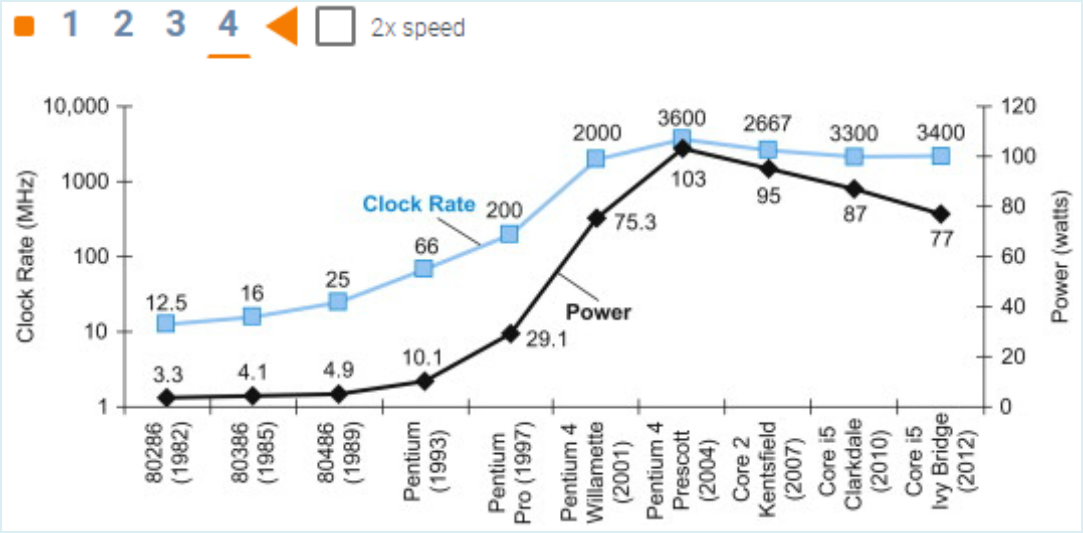
Do you have any questions?

8:48 AM

Introduction to Computers

1.7 The power wall

- The power wall represents a critical challenge in scaling processor performance due to the limits of power consumption (and heat dissipation).
- Addressing this challenge involves a combination of design innovation, cooling technology, and alternative performance strategies.



$$Energy \propto 1/2 \times Capacitive\ load \times Voltage^2$$

The power required per transistor is just the product of energy of a transition and the frequency of transitions:

$$Power \propto 1/2 \times Capacitive\ load \times Voltage^2 \times Frequency\ switched$$

Introduction to Computers

1.7 The power wall

Suppose we developed a new, simpler processor that has 85% of the capacitive load of the more complex older processor. Further, assume that it can adjust voltage so that it can reduce voltage 15% compared to processor B, which results in a 15% shrink in frequency. What is the impact on dynamic power?

- We know, $Power \propto 1/2 \times Capacitive\ load \times Voltage^2 \times Frequency\ switched$
- Find power for the old and new processors, then compare

$$\frac{\text{Power}_{new}}{\text{Power}_{old}} = \frac{\langle \text{Capacitive load} \times 0.85 \rangle \times \langle \text{Voltage} \times 0.85 \rangle^2 \times \langle \text{Frequency switched} \times 0.85 \rangle}{\text{Capacitive load} \times \text{Voltage}^2 \times \text{Frequency switched}}$$

Thus the power ratio is

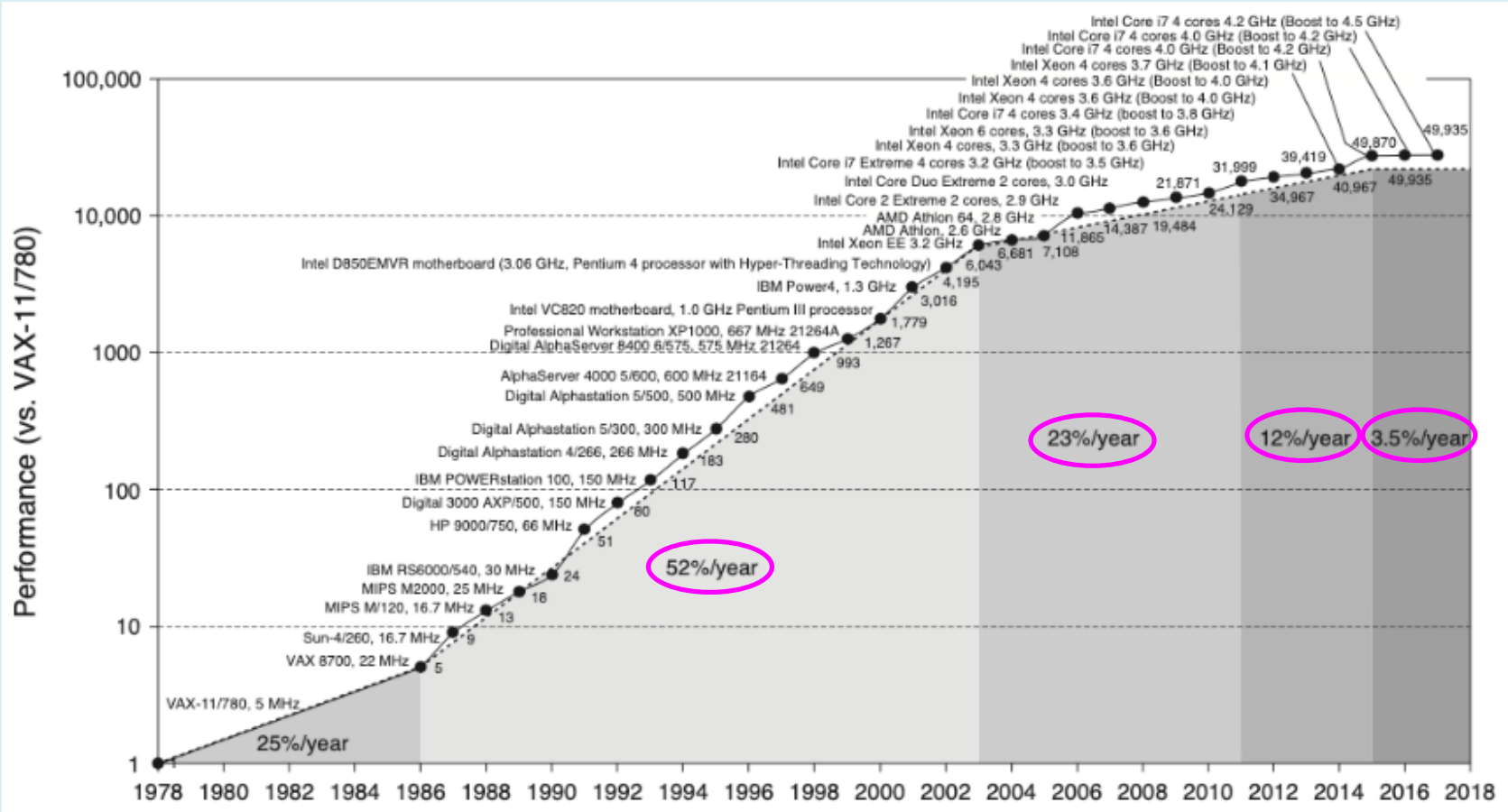
$$\frac{\text{Power}_{new}}{\text{Power}_{old}} = 0.85^4 = 0.52$$

Hence, the new processor uses about half the power of the old processor.

Introduction to Computers

1.8 From uniprocessors to multiprocessors

- Growth in processor performance in 1980-2018



Introduction to Computers

1.8 From uniprocessors to multiprocessors

- Slowing in uniprocessor performance has led to a switch to multiprocessor systems.

Introduction to Computers

1.9 The Intel Core i7

- About the Intel Core i7
  - The Intel Core i7 is a series of high-performance processors designed by Intel, typically targeted at enthusiasts, gamers, and professionals.
- Benchmark
  - A program selected for use in comparing computer performance.
- SPEC CPU benchmark
  - A widely recognized and influential suite of tests used to evaluate and compare the performance of computer processors.
- Workload
  - A set of programs run on a computer that is either the actual collection of applications run by a user or constructed from real programs to approximate such a mix.

Intro to Computer Architecture

Lecture 5

Reading: See Reading Assignments on Blackboard

Tests: HW-2 (Week 3), Quiz-1 (Week 4), Exam-1 (Week 5), …

- Quiz and Exam; zyBook Ch 1.1-1.6;
- Introduction to Computers (from zyBooks)
  - 1.7 The power wall
  - 1.8 The switch from uniprocessor to multiprocessor
  - 1.9 The Intel Core i7
- Intro to Structured Computer Organization (Handout 2a)
  - 1.1.1: Languages, Levels, and Virtual Machines
  - 1.1.2: Contemporary Multilevel Machines

CAPPLab
capplab@wichita.edu

ছাত্রদলের গান

- কাজী নজরুল ইসলাম

আমরা শক্তি আমরা বল
আমরা ছাত্রদল।
মোদের পায়ের তলায় মূর্ছে তুফান
ঊর্ধ্বে বিমান ঝড়-বাদল।
আমরা ছাত্রদল।।

We are energy, we are force
We are students.
[Google translate]

সবাই যখন বুদ্ধি যোগায়
আমরা করি ভুল!
সাবধানীরা বাঁধ বাঁধে সব,
আমরা পশি নীল অতল!
আমরা ছাত্রদল।।

আমরা ধরি মৃত্যু রাজার
যজ্ঞ-ঘোড়ার রাশ,
মোদের মৃত্যু লেখে মোদের
জীবন--ইতিহাস!
হাসির দেশে আমরা আনি
সর্বনাশী চোখের জল
আমরা ছাত্রদল।।

মোরা বিংশ শতাব্দীর!
গৌরবেরি কান্না দিয়ে
ভ'রেছি মা'র শ্যাম-আঁচল।
আমরা ছাত্রদল।

আমরা রচি ভালোবাসার
আশার ভবিষ্যৎ,
মোদের স্বর্গ-পথের অভাস দেখায়
আকাশ-ছায়াপথ!
মোদের চোখে বিশ্ববাসীর
স্বপ্ন দেখা হোক সফল।
আমরা ছাত্রদল।।

U.S.News & WORLD REPORT NEWS » News Best Countries Best States Healthiest Communities Ideas & Opinions Elections

https://www.usnews.com/news/world/articles/2024-08-14/students-who-ousted-hasina-are-helping-lead-bangladesh-from-the-streets-to-the-ministries

Home / News / World News / Students Who Ousted Hasina ...

Students Who Ousted Hasina Are Helping Lead Bangladesh, From the Streets to the Ministries

Within a week of unseating Bangladesh's longest-serving prime minister, the students who drove out former Prime Minister Sheikh Hasina were directing Dhaka's traffic

By Associated Press | Aug. 14, 2024, at 12:13 a.m.

Learning Points:

Students are the future of any nation!
Student-centered education (be proactive)

Natural Languages → Computer Languages
Translation or Interpretation? (Handout 2)

Intro to Computer Architecture

Computer: A Multilevel Machine

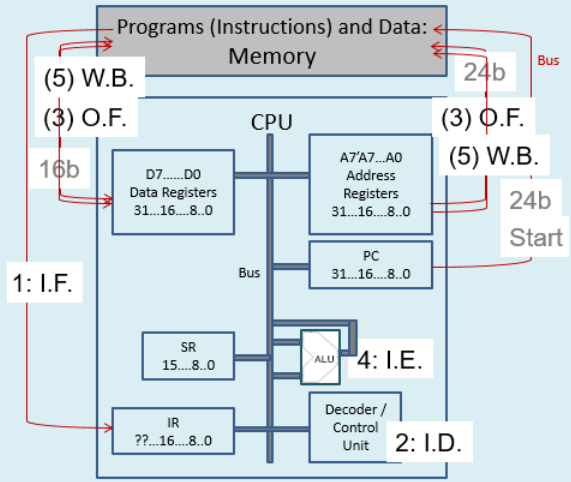
- https://www.youtube.com/watch?v=hYZUzoSxg7c

- ??
- **Different Languages**
- **Natural Languages**
- **Interpretation/Translation**
- **Computer Languages**
- **Programming Languages**
- **Interpretation/Translation**

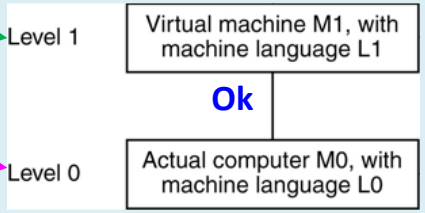
Intro to Computer Architecture

Multilevel Machines

- A digital computer solves problems by carrying out instructions.
- A program is a sequence of instructions describing how to perform/solve a certain task/problem.

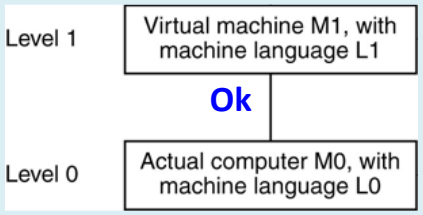


- Programs are written (by people) using a programming language.
- Machine language is a limited set of primitive instructions that are used by people to communicate with a computer.



Intro to Computer Architecture

Multilevel Machines (+)



PROBLEM

We want the machine language to be very simple so as to reduce the complexity and the cost of electronics => *becomes tedious for people*.

We want to make it convenient for people to use the computer=> *needs complex electronics*.

Intro to Computer Architecture

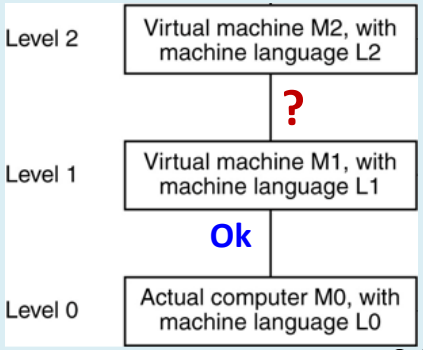
Multilevel Machines (+)

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).

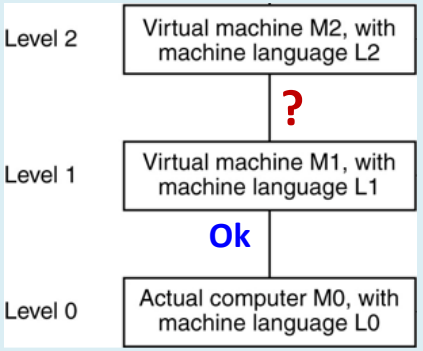
How can a program written in L2 be executed by the computer?



Intro to Computer Architecture

Multilevel Machines (+)

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



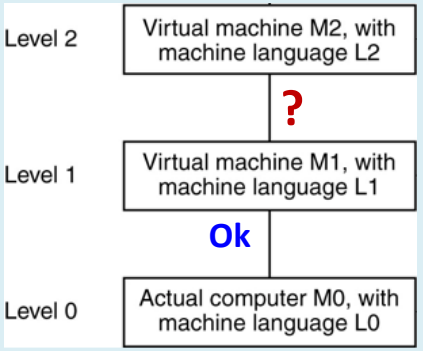
TRANSLATION

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?

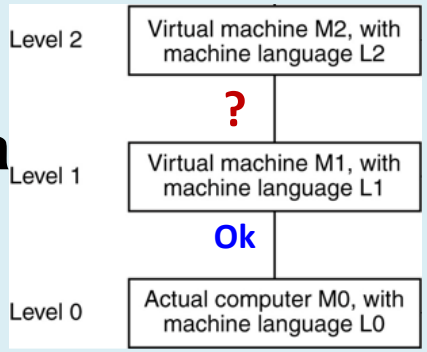


INTERPRETATION

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

Intro to Computer Architecture

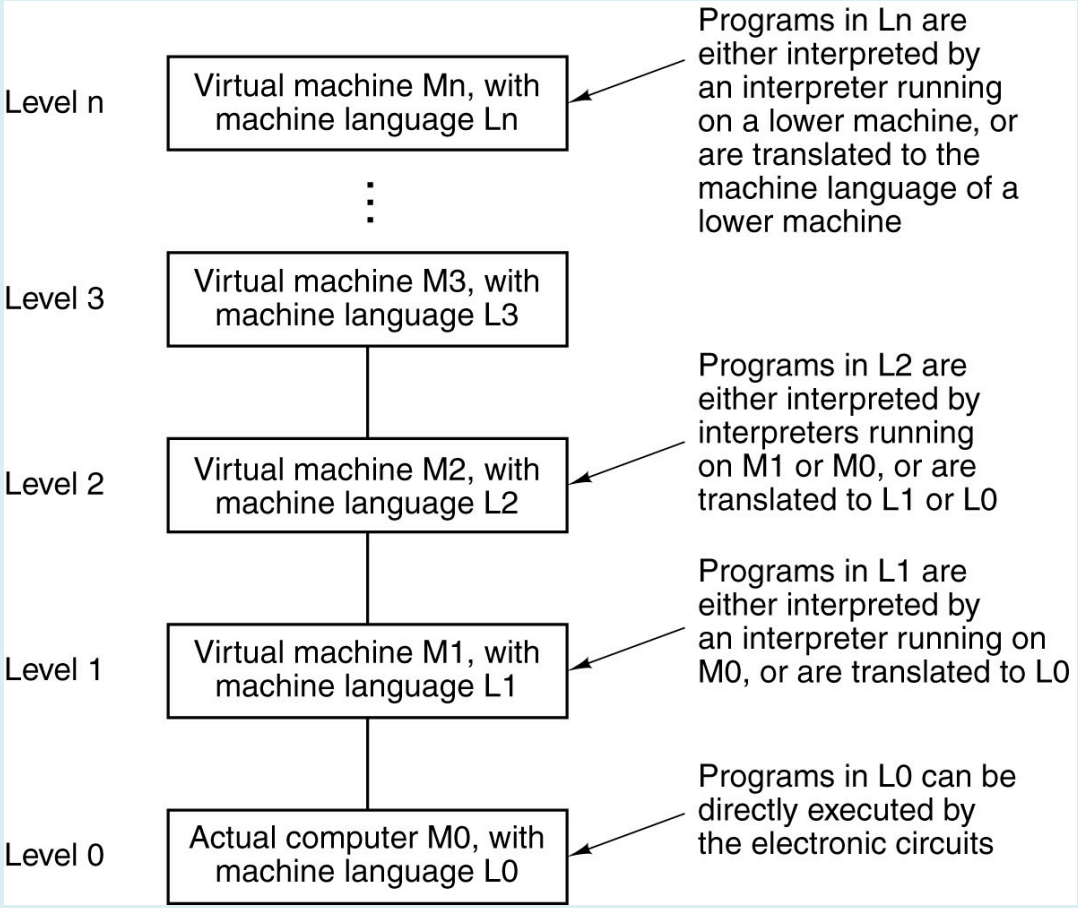
Multilevel Machines



- **It is convenient to imagine the existence of a virtual machine with machine language L2.**
- **L1 and L2 must not be too different =>L2 is far from ideal for most applications.**
- **Invent another set of instructions that is more people-oriented.**
- **Call the language formed by this set L3.**
- **This can be extended to as many levels as needed so as to make it convenient for most application programmers.**

Intro to Computer Architecture

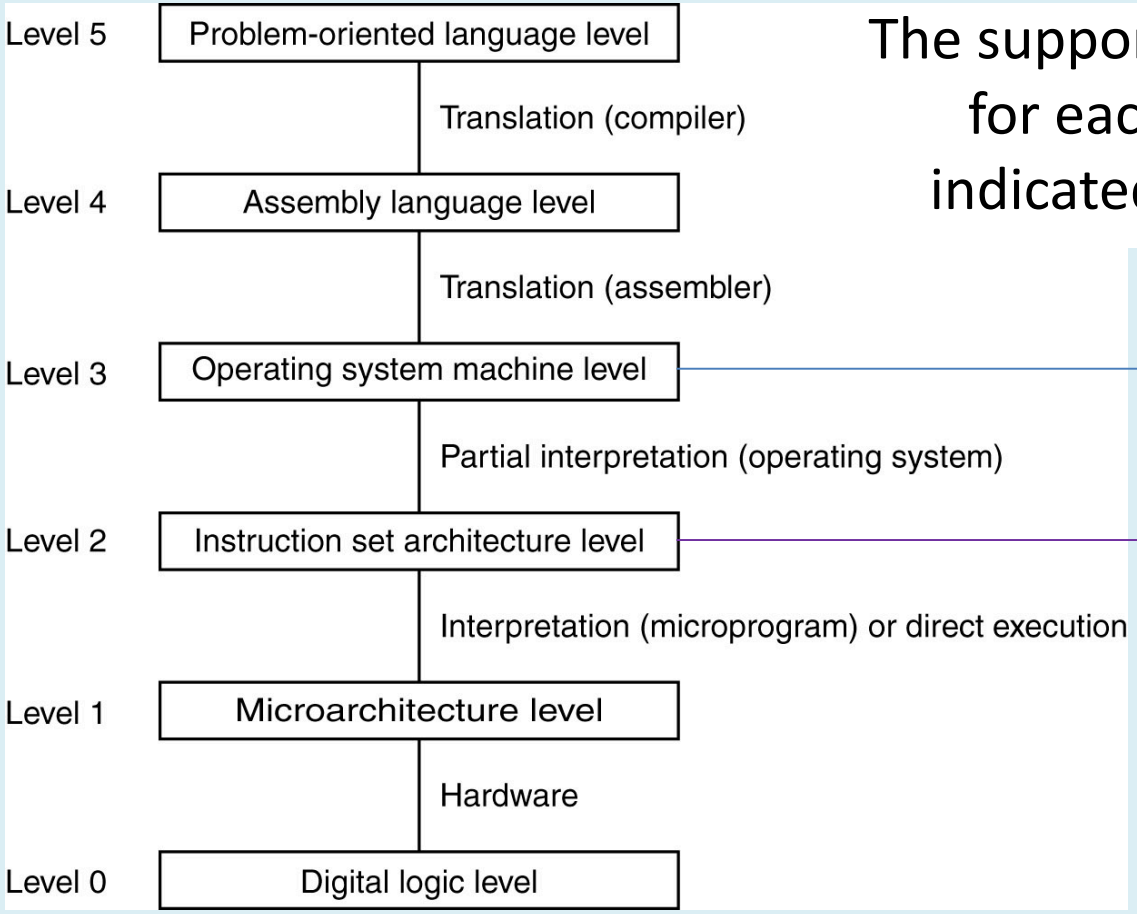
A (n+1)-level Machine



Intro to Computer Architecture

A Six-Level Machine

Level up?



The support method for each level is indicated below it.

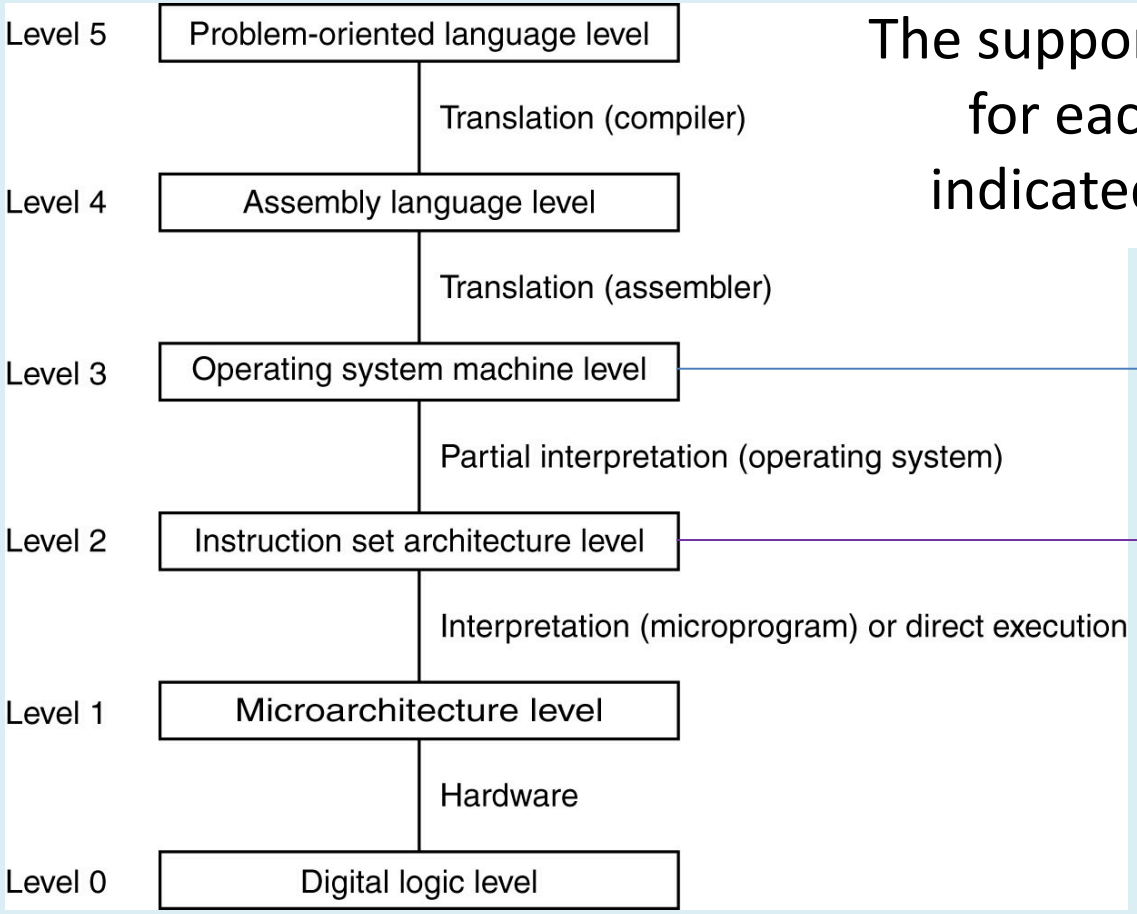
Human Machine

Software Hardware

Level below?

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

Do you have any questions?

8:48 AM

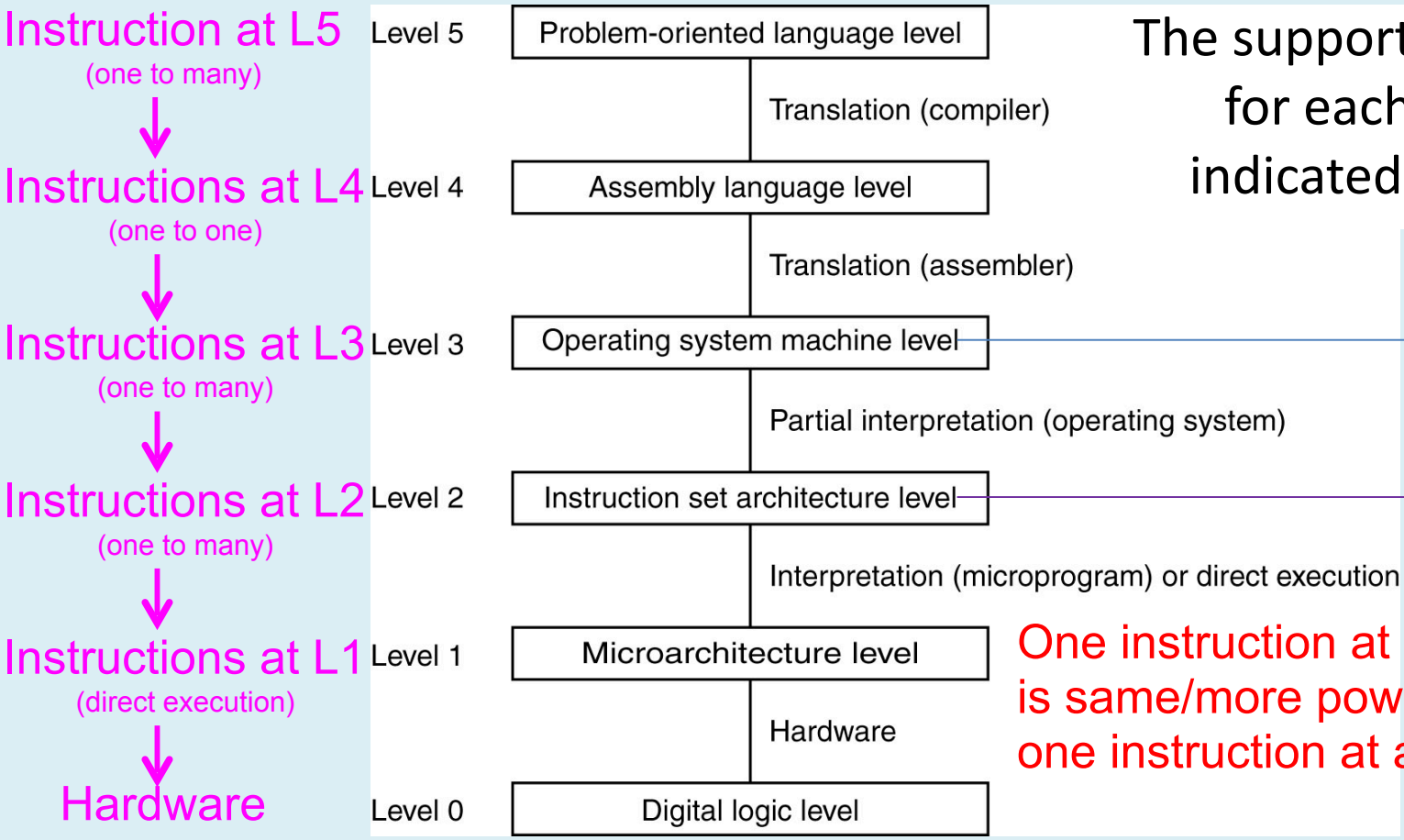
Intro to Computer Architecture

Example to Practice

- **Name the levels of a six-level machine**
  - **(L5) Problem-oriented language level**
  - **(L4) Assembly language level**
  - **(L3) Operating system machine level**
  - **(L2) Instruction set architecture level**
  - **(L1) Microarchitecture level**
  - **(L0) Digital logic level**
- **Which of L0, L1, and L2 levels came last?**
  - **(1940s) Instruction Set Architecture level**
  - **(1950s) Microarchitecture level**
  - **(1940s) Digital logic level (hardware)**

Intro to Computer Architecture

A Six-Level Machine GUI-Tools



The support method for each level is indicated below it.

Human

Machine

Software

Hardware

One instruction at upper level is same/more powerful than one instruction at a lower level.

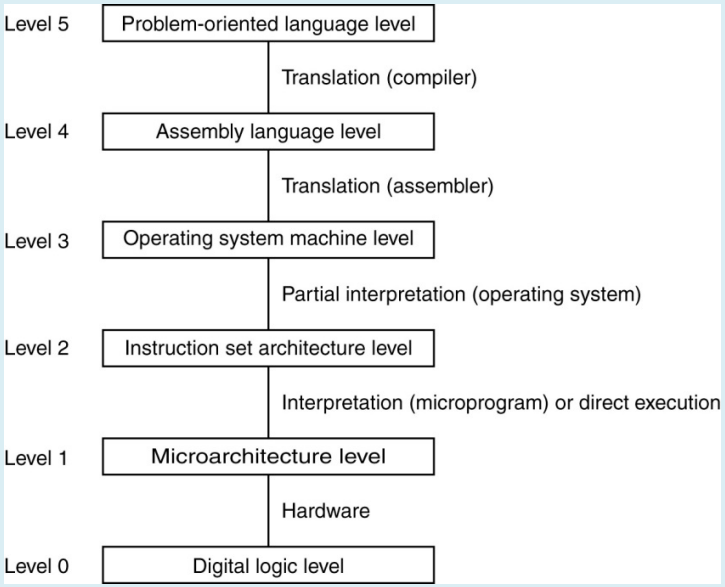
One instruction at upper level is one or more instructions at a lower level.

Solid State Physics

Intro to Computer Architecture

Example → Practice

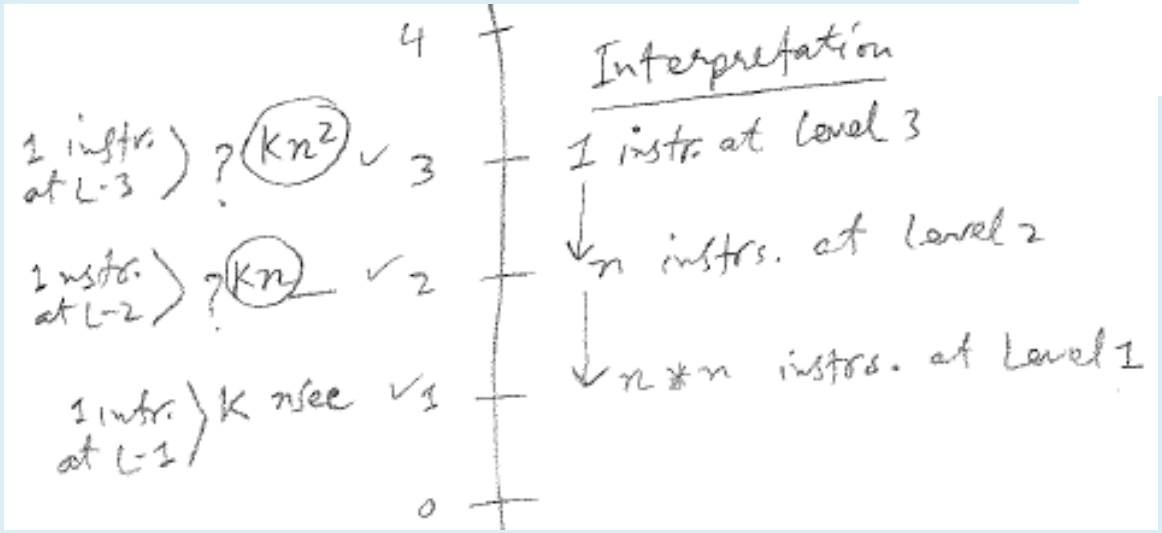
- **Consider a computer with identical interpreters at levels 1, 2, and 3. It takes an interpreter n (say, n > 1) instructions to fetch, decode, and execute one instruction. A level 1 instruction takes k (say, k > 1) nanoseconds to execute. How long does it take for an instruction at levels 2 and 3?**

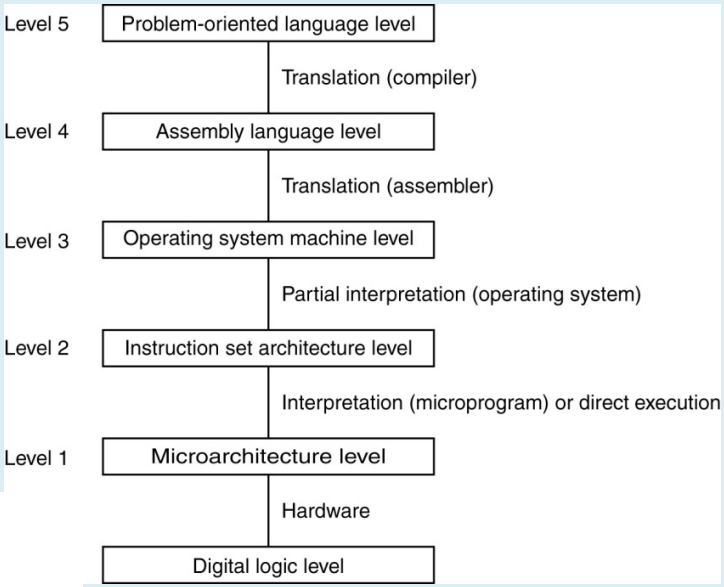


Intro to Computer Architecture

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 level 1 instruction takes k nanoseconds to execute. How long does it take for an instruction at levels 2 and 3?**





Intro to Computer Architecture

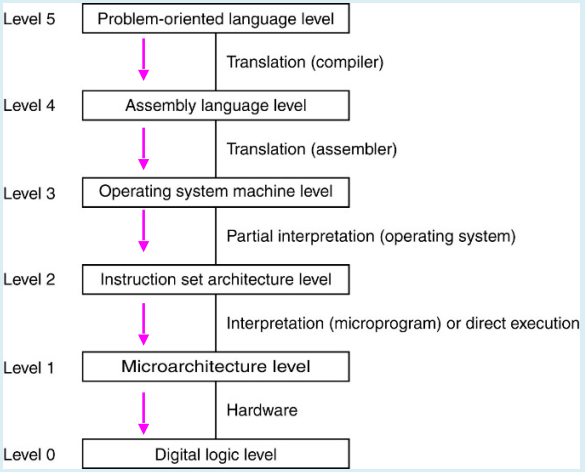
Do you have any questions?

8:48 AM

Intro to Computer Architecture

Computer Architecture: A Multilevel Approach

- ✓ The Tanenbaum and Austin book (Structured Computer Organization)
- ✓ Higher (human friendly) to lower (machine friendly)
- ✓ Multilevel Computers: https://users.cs.fiu.edu/~downeyt/cop3402/levels.html



COMPUTER ARCHITECTURE: A Quantitative Approach

- ✓ The Hennessy and Patterson book
- ✓ Quantitative principles of computer design: to make the common case fast.
- ✓ To quantify the principles → Amdahl's Law, CPU performance, Principle of Locality, Advantage of Parallelism, etc.
- ✓ Quantitative Principles of Computer Design: http://www.brainkart.com/article/Quantitative-Principles-of-Computer-Design_8830/

Intro to Computer Architecture

Computer Architecture: "Old" and "Real" views

- "Old" view of computer architecture:
  - Instruction Set Architecture (ISA) design, i.e., decisions regarding registers, memory addressing, addressing modes, instruction operands, available operations, control flow instructions, and instruction encoding
- "Real" computer architecture:
  - Specific requirements of the target machine
  - Design to maximize performance within constraints: cost, power, and availability
  - Includes ISA, microarchitecture, hardware

Intro to Computer Architecture

Do you have any questions?

8:48 AM

Intro to Computer Architecture

Milestones in Computer Architecture

- **1834, 1936, 1951, 1952, 1961**

Year	Name	Made by	Comments
1834	Analytical Engine	Babbage	First attempt to build a digital computer
1936	Z1	Zuse	First working relay calculating machine
1943	COLOSSUS	British gov't	First electronic computer
1944	Mark I	Aiken	First American general-purpose computer
1946	ENIAC I	Eckert/Mauchley	Modern computer history starts here
1949	EDSAC	Wilkes	First stored-program computer
1951	Whirlwind I	M.I.T.	First real-time computer
1952	IAS	Von Neumann	Most current machines use this design
1960	PDP-1	DEC	First minicomputer (50 sold)
1961	1401	IBM	Enormously popular small business machine
1962	7094	IBM	Dominated scientific computing in the early 1960s
1963	B5000	Burroughs	First machine designed for a high-level language
1964	360	IBM	First product line designed as a family

Intro to Computer Architecture

Milestones in Computer Architecture (+)

- **1974, 1983, 1985, 1987**

Year	Name	Made by	Comments
1965	PDP-8	DEC	First mass-market minicomputer (50,000 sold)
1970	PDP-11	DEC	Dominated minicomputers in the 1970s
1974	8080	Intel	First general-purpose 8-bit computer on a chip
1974	CRAY-1	Cray	First vector supercomputer
1978	VAX	DEC	First 32-bit superminicomputer
1981	IBM PC	IBM	Started the modern personal computer era
1981	Osborne-1	Osborne	First portable computer
1983	Lisa	Apple	First personal computer with a GUI
1985	386	Intel	First 32-bit ancestor of the Pentium line
1985	MIPS	MIPS	First commercial RISC machine
1987	SPARC	Sun	First SPARC-based RISC workstation
1990	RS6000	IBM	First superscalar machine
1992	Alpha	DEC	First 64-bit personal computer
1993	Newton	Apple	First palmtop computer

Intro to Computer Architecture

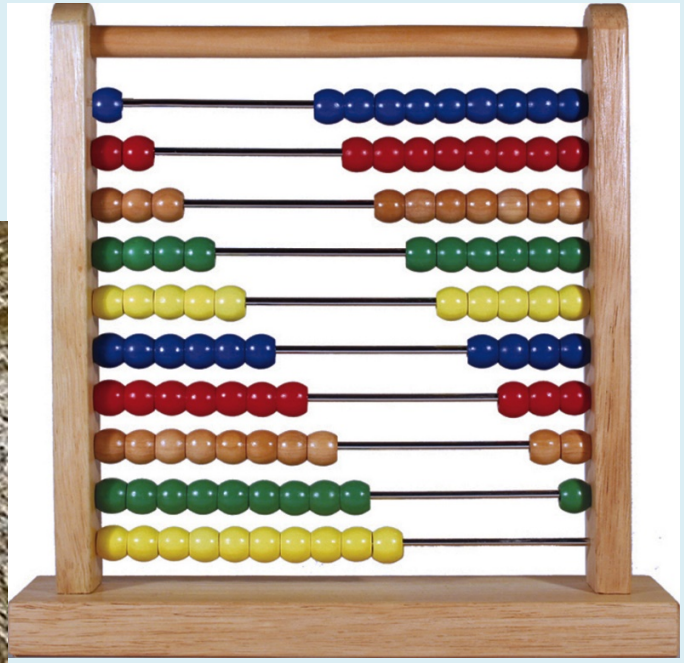
Computer Generations

- **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

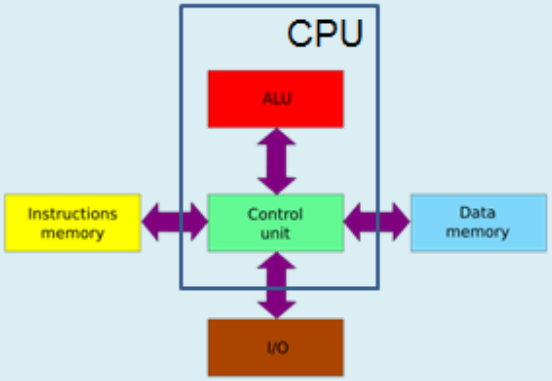
Computer Generations (+)

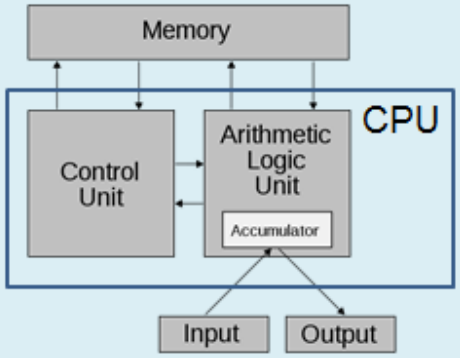
- **Zeroth Generation: Mechanical Computers (1642 – 1945)**
  - **First calculating machine by Pascal (1623-1662) in 1642**
  - **Babbage (1792-1871), "different engine", speedometer**
  - **"Analytical engine" – storage (memory) 1000 words of 50 decimal digits, mill (computation unit – add, subtract,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

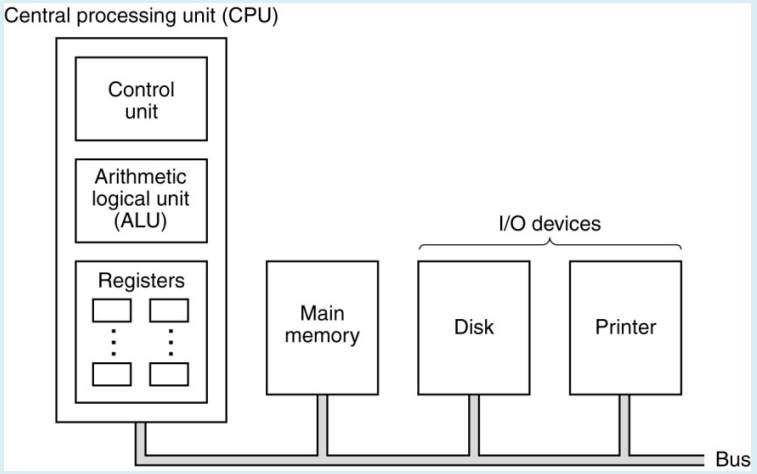




Intro to Computer Architecture

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

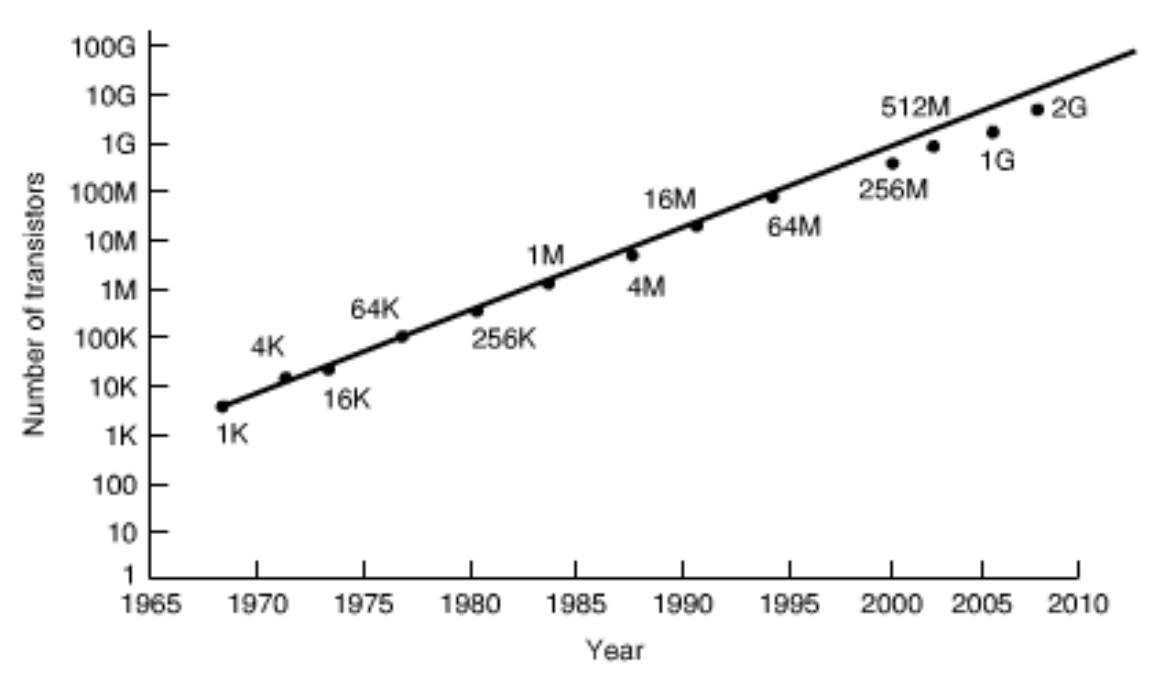
Computer Generations

- **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

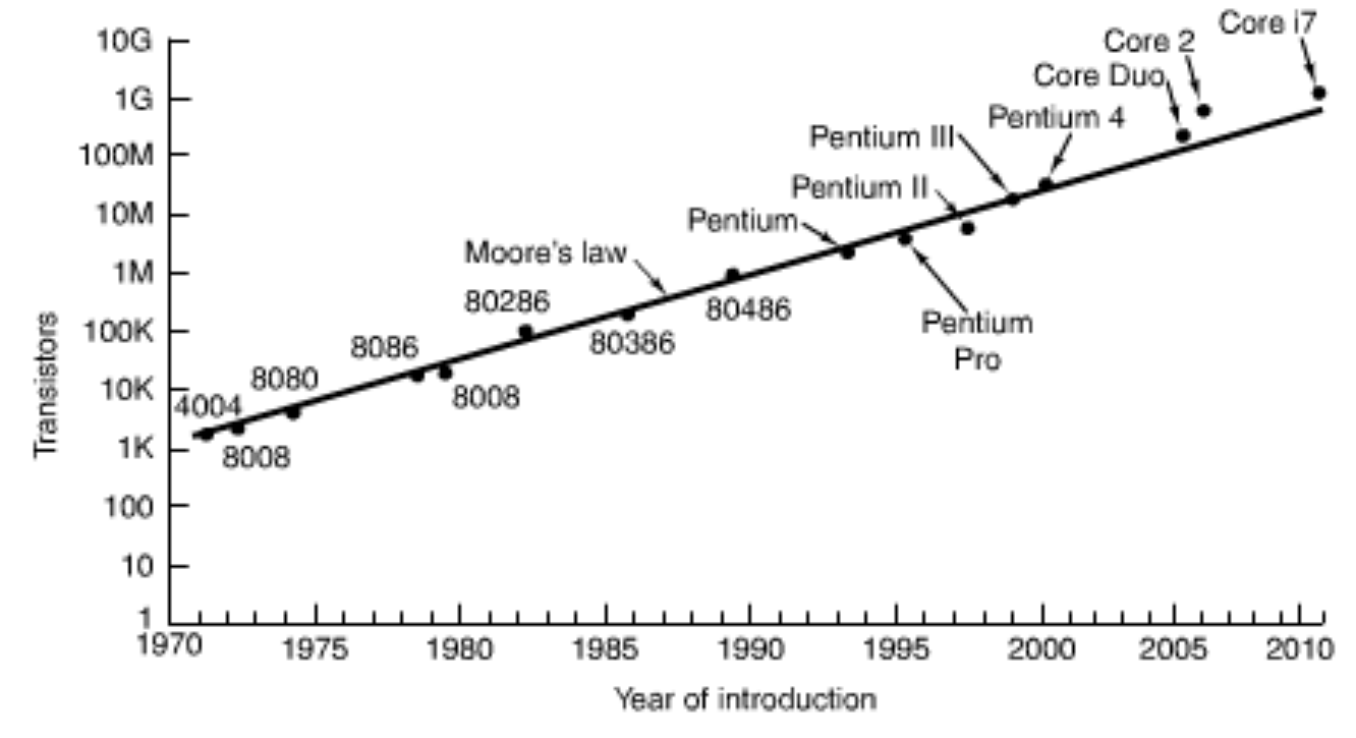
Moore's law and processor design (+)

- 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**

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.		