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 4

Reading: See Reading Assignments on Blackboard

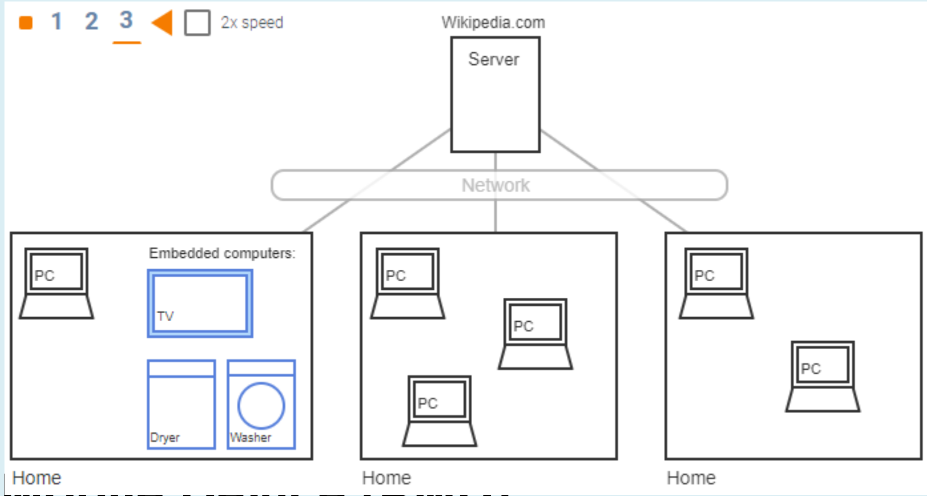
Tests: HW-1 (Week 2), HW-2 (Week 3), Quiz-1 (Week 4), …

- **zyBook Ch 1.1-1.4; HW-1;**
- **Introduction to Computers (from zyBooks)**
  - **1.5 Technologies for processors and memory**
  - **1.6 Performance**
  - **1.7 The power wall**
  - **1.8 The switch from uniprocessor to multiprocessor**
  - **1.9 The Intel Core i7**

Introduction to Computers

1.1 Introduction

- **Traditional classes of computing applications**
  - **(see zyBooks) Personal Computer (PC), Server, and Embedded Computer**

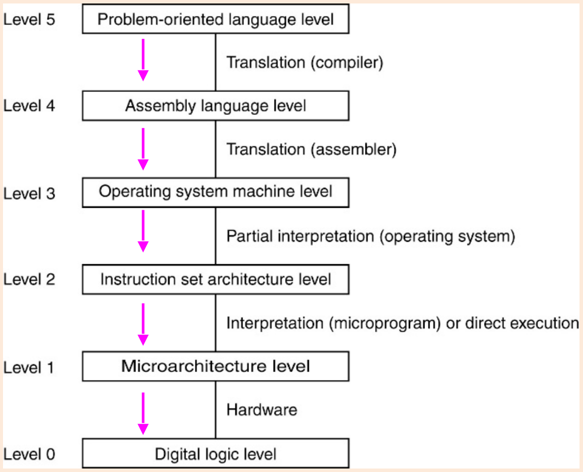


  - **Supercomputer, …**
  - **Post-PC ear → Personal Mobile Device (PMD)**
  - **Server → Cloud Computing … Warehouse Scale Computer (WSC)**
  - **Software as a Service (SaaS) – Software and Data over the Internet**

Computing Systems: Two Approaches

Computer Architecture:
A Multilevel Approach [1]

- ✓ 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 [2]

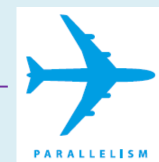
- ✓ The Hennessy and Patterson book
- ✓ Quantitative principles of computer design: to make the common case fast.
- ✓ To quantify the principles → Amdahl's Law, CPU performance, Principle of Locality, Advantage of Parallelism, etc.
- ✓ Quantitative Principles of Computer Design: http://www.brainkart.com/article/Quantitative-Principles-of-Computer-Design_8830/

[1] "Structured Computer Organization" by Tanenbaum and Austin
[2] "COMPUTER ARCHITECTURE: A Quantitative Approach" by Hennessy and Patterson

Introduction to Computers

1.2 Eight great ideas about computer architecture

- Design for Moore’s Law
- Use abstraction to simplify design
- Make the common case fast
- Performance via parallelism
- Performance via pipelining
- Performance via prediction
- Hierarchy of memories
- Dependability via redundancy



Practice Questions:
1) Assembly lines in automobile manufacturing
2) Express elevators in buildings

Answers:
1) Performance via Pipelining
2) Make the Common Case Fast

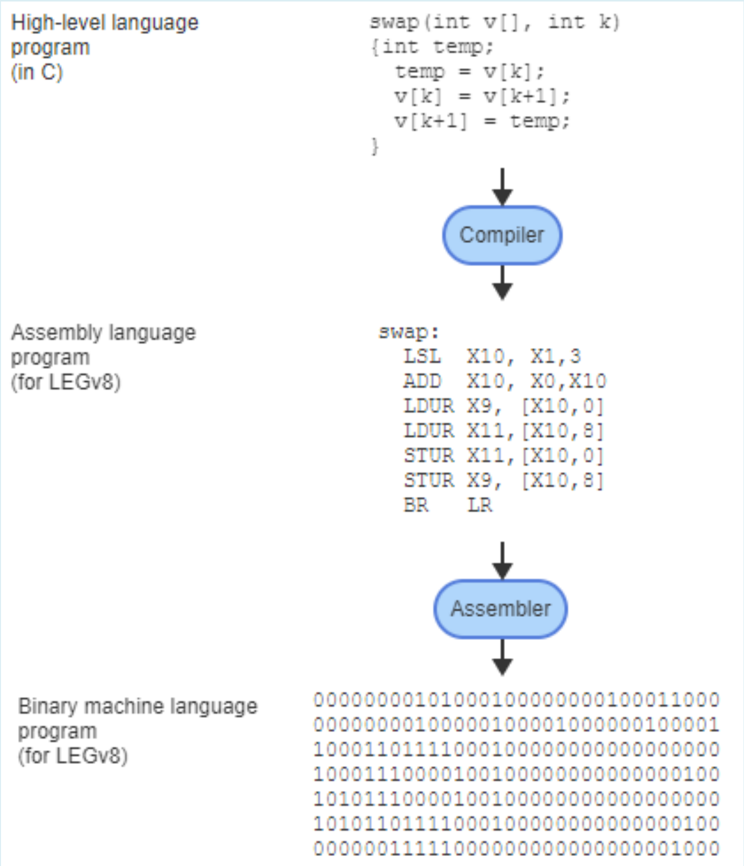
About the symbols:
up and to the right
abstract painting icon
fast small/sports car
multiple jet engines of a plane
sequence of pipes
fortune-teller's crystal ball
layered triangle icon
tractor-trailer (dual tires)

Introduction to Computers

1.3 Below your program

Underlying Software

- **Abstraction**
  - ➢ **Abstraction is a fundamental concept in computing that helps manage complexity by hiding the intricate details of a system and exposing only the essential features.**
  - ➢ **It allows one to work with higher-level concepts without needing to understand the underlying specifics.**

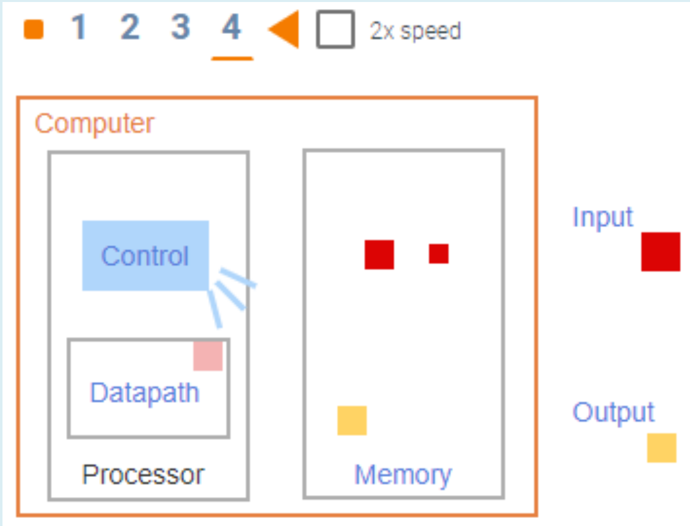


Limp Elastic General-purpose v8 (LEGv8) is a simplified, educational version of the ARMv8 architecture, which is a widely-used instruction set architecture (ISA) for ARM processors.
ARM - Advanced RISC Machines | RISC - Reduced Instruction Set Computing (a computer architecture)

Introduction to Computers

1.4 Under the covers

Underlying Hardware



- **The underlying hardware in any computer performs the same basic functions: inputting data, outputting data, processing data, and storing data.**
- **Important components**
  - **Input device: Keyboard**
  - **Output device: Display**
  - **Memory: Stores programs and data | Cache: a small fast memory**
  - **Dynamic random access memory (DRAM): Integrated Circuit (IC)**
  - **Static random access memory (SRAM): IC, faster than DRAM**
  - **Datapath: Performs operations on data**
  - **Control: Signals that determine the operation of the datapath**

Intro to Computer Architecture

Do you have any questions?

11:24 AM

Introduction to Computers

1.5 Technologies for processors and memory

- **Performance per unit cost of technology**
  - **Relative performance**

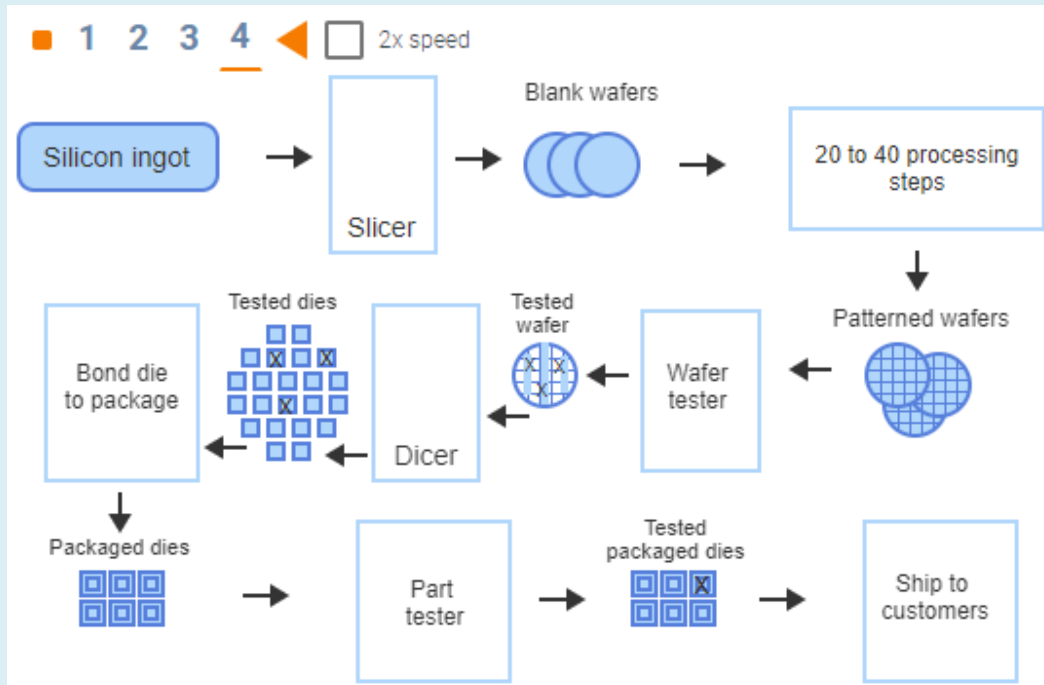
Year	Technology used in computers	Relative performance / unit cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit	900
1995	Very large-scale integrated circuit	2,400,000
2013	Ultra large-scale integrated circuit	250,000,000,000

Source: Computer Museum, Boston, with 2013 extrapolated by the authors.

- **Vacuum tube**
  - **An electronic component that was widely used in early electronic devices before the advent of transistors**
- **Transistor**
  - **An on/off switch controlled by an electric signal.**
- **Very large-scale integrated (VLSI) circuit**
  - **A device containing hundreds of thousands to millions of transistors in a chip (i.e., IC).**

1.5 Technologies for processors and memory

- **The chip manufacturing process**
- **Ingot**
  - ➢ **Silicon crystal Ingot**
- **Wafer**
  - ➢ **A slice from an ingot**
- **Defect**
  - ➢ **A microscopic flaw that can result in the failure of a die**



- **Die**
  - ➢ **The individual rectangular sections that are cut from a wafer**

Introduction to Computers

1.6 Performance

- **What is performance of computers?**
  - **(coming up)**
- **Which airplane performs better?**
  - **Speed?**
  - **Capacity?**
  - **Range?**
  - **Throughput?**

Airplane	Passenger capacity	Cruising range (miles)	Cruising speed (m.p.h.)	Passenger throughput (passengers x m.p.h.)
Boeing 777	375	4630	610	228,750
Boeing 747	470	4150	610	286,700
BAC/Sud Concorde	132	4000	1350	178,200
Douglas DC-8-50	146	8720	544	79,424

- **Throughput**
  - **Aka, bandwidth. The number of tasks completed per unit time.**
- **What is performance of computers?**
  - **(depends on several factors)**

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.**

Clock cycle time = 1 / Clock rate

Introduction to Computers

1.6 Performance

Clock cycle time = 1 / Clock rate

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

- **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?

$$\text{CPU time}_A = \frac{\text{CPU clock cycles}_A}{\text{Clock rate}_A}$$

$$\text{CPU clock cycles}_A = 10 \text{ seconds} \times 2 \times 10^9 \frac{\text{cycles}}{\text{second}} = 20 \times 10^9 \text{ cycles}$$

$$\text{CPU time}_B = \frac{1.2 \times \text{CPU clock cycles}_A}{\text{Clock rate}_B}$$

$$\text{Clock rate}_B = \frac{1.2 \times 20 \times 10^9 \text{ cycles}}{6 \text{ seconds}} = \frac{0.2 \times 20 \times 10^9 \text{ cycles}}{\text{second}} = \frac{4 \times 10^9 \text{ cycles}}{\text{second}} = 4\text{GHz}$$

  - **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**
  - $\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{Clock cycle time}$
  - $\text{CPU time} = \dfrac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}}$

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

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

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

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

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

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

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

Practice: Questions

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: Questions

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: Questions

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?

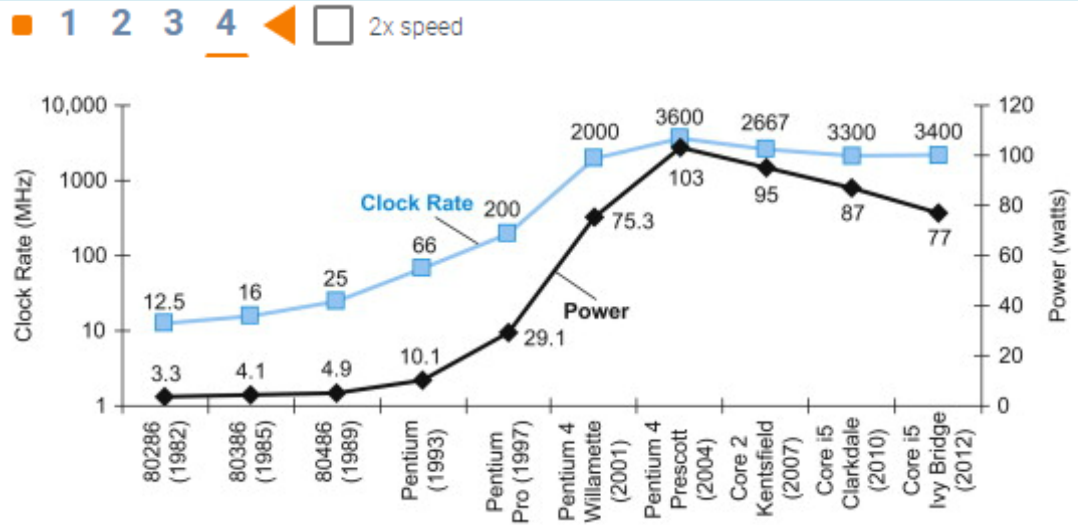
11:24 AM

CAPPLab
capplab@wichita.edu

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:

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

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.