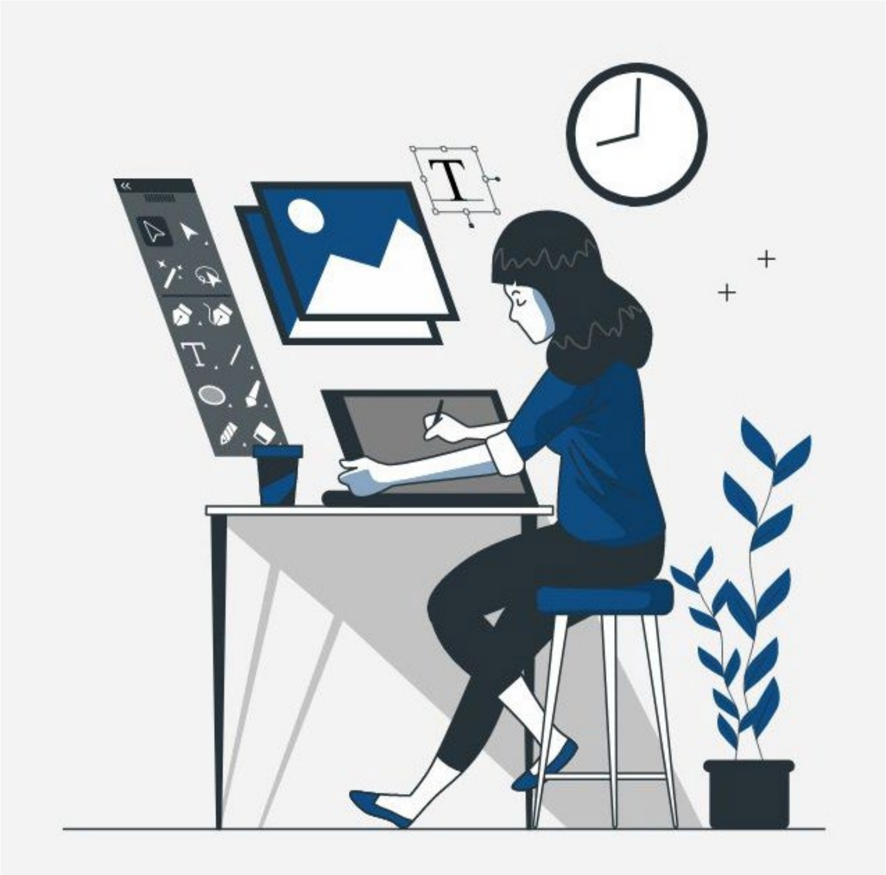


Computer Architecture

Lab. 1

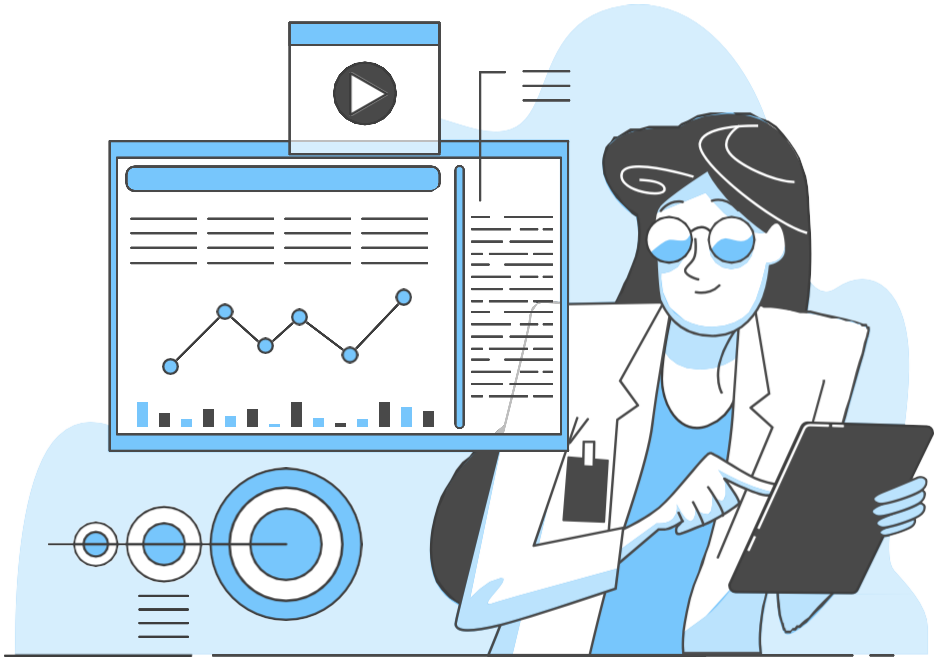
2025/05/15

Agenda

01
Introduction
Tutorials
Computer architecture

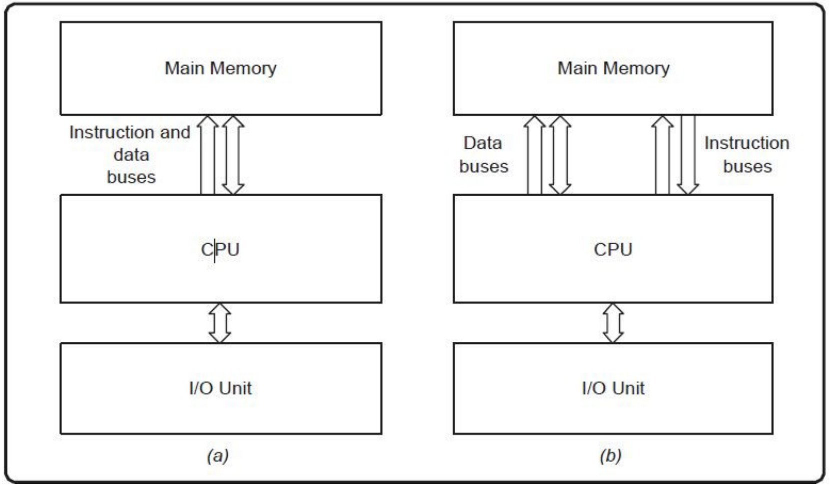
02
Computer components
Layers

03
Practice
Solutions



Computer Architecture

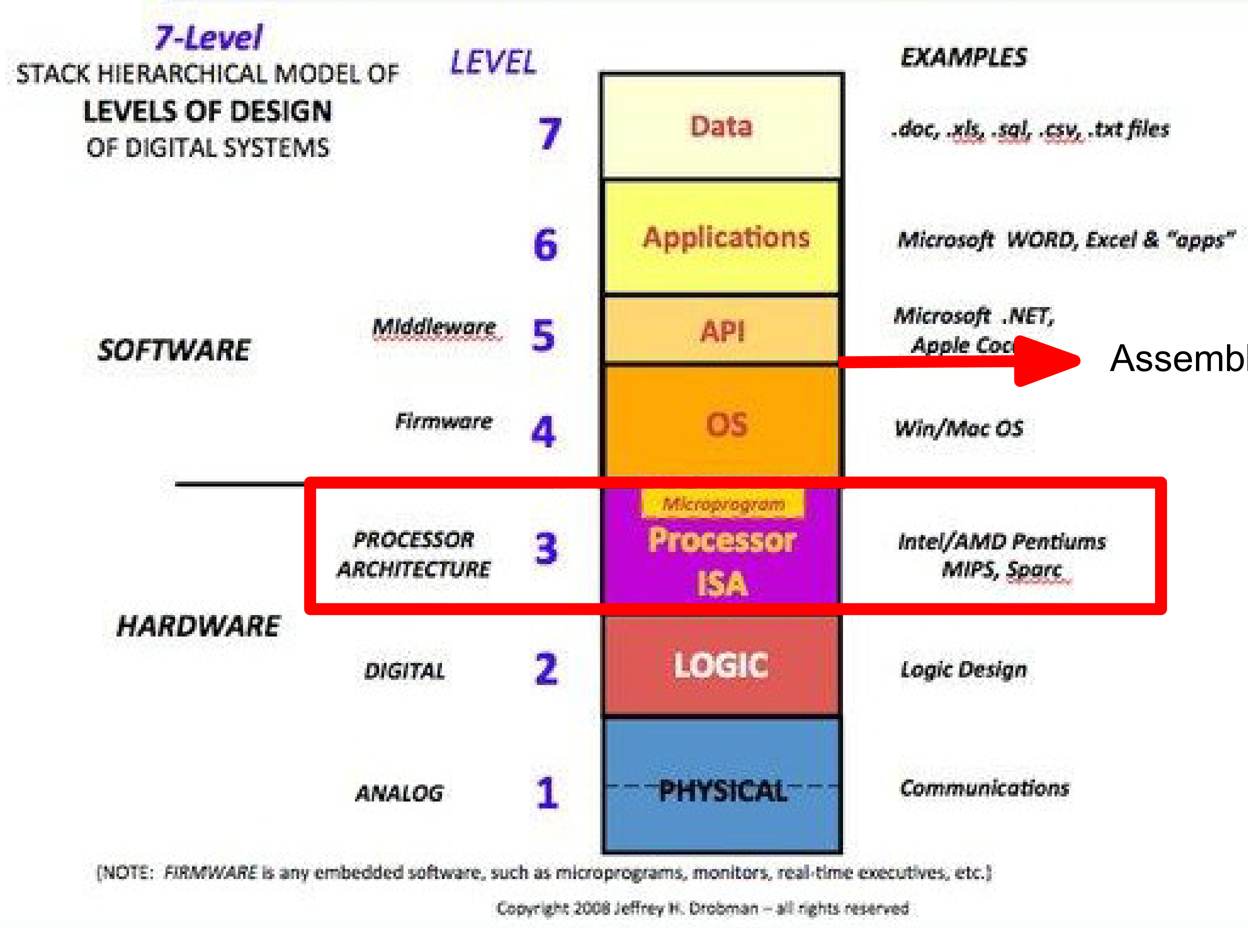
Computer architecture is a description of the structure of a computer system made from component parts.



Von Neumann Architecture (a) and Harvard Architecture (b)

How does computer architecture work?

- Purpose of computer architecture
- Data in numbers
- Manipulating data
- Booting up
- Support for temporary storage
- Support for permanent storage
- User-facing functionality



Assembly language

“Basic unit of memory.”

Bits and bytes are the basic building blocks of memory. "Bit" stands for binary digit. A bit is a one or a zero, on or off, which is how all computer information is stored.

Basic Concepts of Digital Images

Pixels

- **Definition**: A pixel, short for "picture element," is the smallest unit of a digital image. It's essentially a tiny dot that represents a single point in the image.
- **Role in Images**: Digital images are composed of a grid of pixels, each with a specific color and brightness value. When viewed together, these pixels create the complete image.
- **Example**: If you zoom in closely on a digital photo, you might start to see individual pixels, especially if the resolution is low. Each pixel in a color image typically contains three components (red, green, and blue) that combine to produce the full spectrum of color

Resolution: The resolution of a digital image refers to the number of pixels along its width and height. It's usually expressed as "width x height."

High Resolution: More pixels are packed into the same area, resulting in greater detail and clarity. For example, an image with a resolution of 1920 x 1080 has 1920 pixels horizontally and 1080 pixels vertically.
Low Resolution: Fewer pixels in the same area, which can make the image appear blurry or pixelated when enlarged. An example would be an image with a resolution of 640 x 48

RGB Color Model: Each pixel is represented by a combination of Red, Green, and Blue (RGB) values.

Bit Depth: 8 bits per color channel means each color (red, green, and blue) is represented by 8 bits (1 byte).

Frame Buffer: A storage space that holds the pixel data for an image or frame.

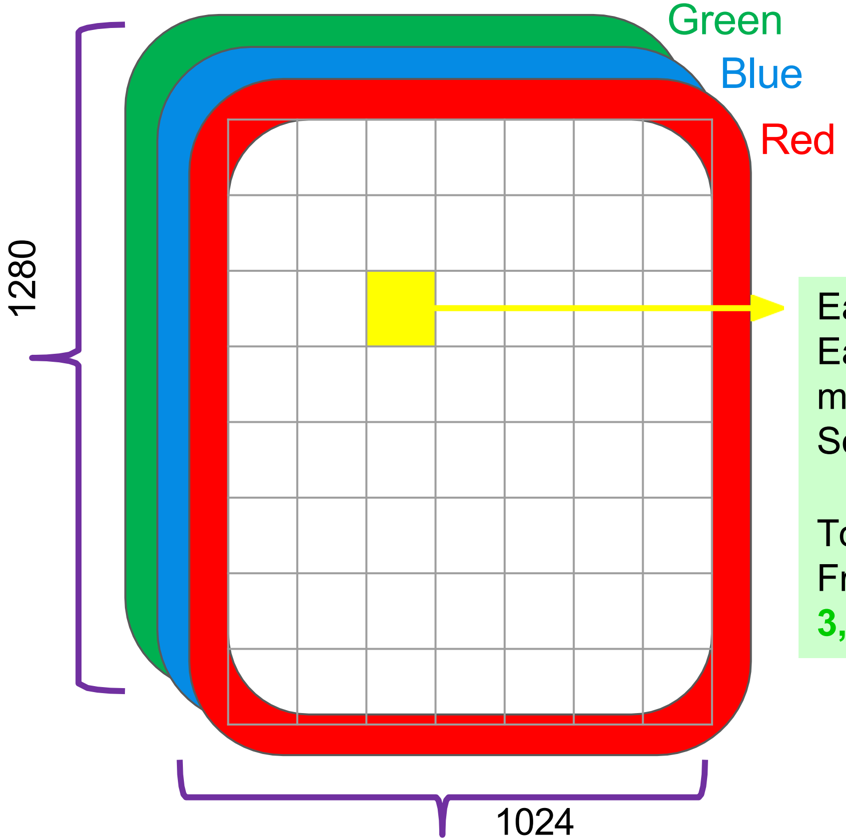
Question 1.

Assume a color display using 8 bits for each of the primary colors (red, green, blue) per pixel and a frame size of 1280 × 1024.

a. What is the minimum size in bytes of the frame buffer to store a frame?

b. How long would it take, at a minimum, for the frame to be sent over a 100 Mbit/s network?

Solution 1.



1 Byte = 8 bits

Each pixel has 3 channels
Each channel needs 8 bits which means 1 Byte -> 3*1=3
So this pixel needs 3 Bytes

Total pixels = 1280*1024 = 1,310,720
Frame buffer size= 1,310,720*3 =
3,932,160

Q1. B.

100 Mbit/s network

The network capacity to transmit data at a time unit

1 Mbit = e^6 bit (10^6)
E = 2.718

1. Choose a base -> bit
2. Unit conversion
   Byte to bit: 3,932,160 × 8 = 31,457,280 bit
   Mbit to bit: 100 * 10^6 = 10^8 bit/s
3. 31,457,280/10^8 = 0.31 seconds

1	10^8
x	31,457,280

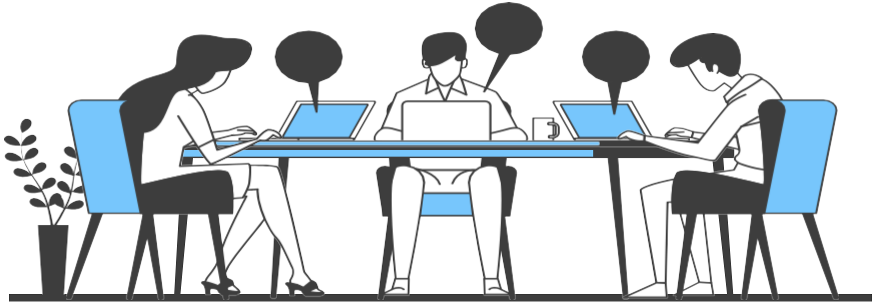
Question 2.

Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0. P3 has a 4.0 GHz clock rate and has a CPI of 2.2.

a. Which processor has the highest performance expressed in instructions per second?

b. If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.

c. We are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?



Some Terminologies

01 execution time
Response time /

02 Throughput /
bandwidth

03 Clock
rate/
Clock
cycle

04 Time

05 Wall clock
time /
response
time /
elapsed time

06 Performance

1. Clock Rate and CPI (Cycles Per Instruction)

- **Clock Rate**: The speed at which a processor executes instructions, measured in Hertz (Hz). For example, a 3 GHz clock rate means the processor performs 3 billion cycles per second.
- **CPI (Cycles Per Instruction)**: The average number of clock cycles each instruction takes to execute. It is a measure of the processor's efficiency.

2. Instructions Per Second (IPS)

- **Instructions Per Second**: The number of instructions a processor can execute in one second. This is calculated using the formula: IPS=Clock Rate/CPI

3. Execution Time and Performance

- **Execution Time**: The total time taken to complete a given task or program. It is influenced by the clock rate and CPI.
- **Performance**: Performance is often measured in terms of how many instructions can be executed per second or how quickly a task can be completed.

4. Number of Cycles

- **Number of Cycles**: The total number of clock cycles required to execute a program. This is calculated using the formula: Number of Cycles=Clock Rate×Execution Time

5. Number of Instructions

- **Number of Instructions**: The total number of instructions executed during a program's run. This is calculated by dividing the total number of cycles by the CPI: Number of Instructions=Number of Cycles/CPI

6. Impact of Changes in Execution Time and CPI

- **Effect of Execution Time Reduction**: Understanding how reducing execution time affects overall performance and how it necessitates changes in clock rate or CPI.
- **Effect of CPI Increase**: Recognizing that an increase in CPI means each instruction takes more cycles to execute, which can impact overall performance and require a higher clock rate to maintain the same execution time.

3 GHz clock rate
CPI of 1.5

2.5 GHz clock rate
CPI of 1.0

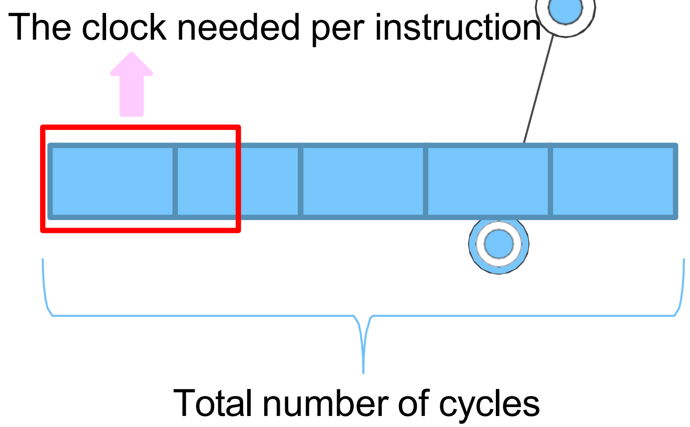
4.0 GHz clock rate
CPI of 2.2

```
Performance = (1 / Execution time)

Execution time = CPU clock cycles x clock cycle time

Execution time = CPU clock cycles / clock rate
```

performance of P1 (instructions/sec) = 3 × 10^9/1.5 = 2 × 10^9
performance of P2 (instructions/sec) = 2.5 × 10^9/1.0 = 2.5 × 10^9
performance of P3 (instructions/sec) = 4 × 10^9/2.2 = 1.8 × 10^9

cycles(P1) = 10 × 3 × 10^9 = 30 × 10^9
cycles(P2) = 10 × 2.5 × 10^9 = 25 × 10^9
cycles(P3) = 10 × 4 × 10^9 = 40 × 10^9

No. instructions(P1) = 30 × 10^9/1.5 = 20 × 10_9
No. instructions(P2) = 25 × 10_9/1 = 25 × 10_9
No. instructions(P3) = 40 × 10_9/2.2 = 18.18 × 10_9



New Execution Time=10 s×(1−0.30)=7 s

P1: New CPI1=1.5×1.20=1.8
P2: New CPI2=1.0×1.20=1.2
P3: New CPI3=2.2×1.20=2.64

```
Time = Instruction Count x CPI / clock rate
Clock rate = Instruction Count x CPI / time
```

f = No. instr. × CPI / time, then
f (P1) = 20 × 10^9 × 1.8 / 7 = 5.14 GHz
f (P2) = 25 × 10^9 × 1.2 / 7 = 4.28 GHz
f (P3) = 18.18 × 10^9 × 2.6 / 7 = 6.75 GHz

Question 3

Consider two different implementations of the same instruction set architecture.
The instructions can be divided into four classes according to their CPI (class A, B, C, and D).
P1 with a clock rate of 2.5 GHz and CPIs of 1, 2, 3, and 3, and P2 with a clock rate of 3 GHz and CPIs of 2, 2, 2, and 2.

a. Given a program with a dynamic instruction count of 1.0E6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C, and 20% class D, which implementation is faster?
b. b. What is the global CPI for each implementation?
c. Find the clock cycles required in both cases.

Q3. a.

Instructions type:
A,B,C,D

	A	B	C	D	CR(GHz)
P1 CPI	1	2	3	3	2.5
P2 CPI	2	2	2	2	3

Class A: 10^5 instr.
Class B: 2 × 10^5 instr.
Class C: 5 × 10^5 instr.
Class D: 2 × 10^5 instr.

1	1
10^5	x

Time = No. instr. × CPI / clock rate
Total time P1 = (10^5 + 2 × 10^5 × 2 + 5 × 10^5 × 3 + 2 × 10^5 × 3)/(2.5 × 10^9) = 10.4 × 10^-4 s
Total time P2 = (10^5 × 2 + 2 × 10^5 × 2 + 5 × 10^5 × 2 + 2 × 10^5 × 2)/(3 × 10^9) = 6.66 × 10^-4 s

P2 is faster

Q3. b.

`Execution time = Instruction Count x CPI / clock rate`

CPI = time * clock rate / instruction count

$CPI(P1) = 10.4 \times 10^{-4} \times 2.5 \times 10^{9}/10^{6} = 2.6$

$CPI(P2) = 6.66 \times 10^{-4} \times 3 \times 10^{9}/10^{6} = 2.0$

clock cycles(P1) = $10^5 \times 1 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 3 + 2 \times 10^5 \times 3 = 26 \times 10^5$

clock cycles(P2) = $10^5 \times 2 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 2 + 2 \times 10^5 \times 2 = 20 \times 10^5$

Thanks!

Do you have any questions?

Harshit.maheshwari@ontariotechu.net

Performance

- **Response time / execution time**: Also called execution time. The total time required for the computer to complete a task,
  -including disk accesses, memory accesses, I/O activities, operating system overhead, CPU execution time, and so on.
- **Throughput / bandwidth :** Also called bandwidth.
  -Another measure of performance, it is the number of tasks completed per unit time.
- **Time:** is the measure of computer performance:
  -the computer that performs the same amount of work in the least time is the fastest.
- **Wall clock time / response time / elapsed time:** The most straightforward definition of time is called wall clock time, response time, or elapsed time.
  -These terms mean the total time to complete a task, including disk accesses, memory accesses, input/output (I/O) activities, operating system overhead—everything.
- **CPU execution time / CPU time:** CPU execution time: Also called CPU time.
  -The actual time the CPU spends computing for a specific task.

Performance

- **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.
- **system performance / CPU performance:** We will use the term system performance to refer to elapsed time on an unloaded system and CPU performance to refer to user CPU time.
- **Clock rate** is the inverse of the clock period**.**
- **Clock cycle:** Also called tick, clock tick, clock period, clock, or cycle. The time for one clock period, usually of the processor clock, which runs at a constant rate.
- **Clock period:** The length of each clock cycle.