CS200 - Computer Organization

An Introduction

Aravind Mohan

Allegheny College

August 24, 2021

Meeting Time

- Lecture Session:
  - Tuesday, Thursday 9:35 AM - 10:50 AM, Alden 101
- Lab Session:
  - Friday 3:00 PM - 4:50 PM, Alden 101

Professor’s Office Hours

- Monday, Wednesday, and Friday:

  11:30 AM - 12:30 PM

- Tuesday, Thursday:

  11:00 AM - 12:30 PM

To schedule a meeting with me during my office hours, please visit my web site [teaching page] and click on the **Schedule** link in the top right-hand corner to schedule.

Website Details

- Professor's Website:

  https://www.cs.allegheny.edu/sites/amohan/

- Course Website:

  https://www.cs.allegheny.edu/sites/amohan/course.php?cid=MTA=

Textbooks

- Computer Organization and Design, David Patterson and John Hennessy, 5th Edition (ISBN13: 978-0124077263)
- The C Programming Language, Brian Kernighan and Dennis Ritchie, 2nd Edition (ISBN13: 978-0131103627)
- Alan Clements, Principles of Computer Hardware, 4th edition (ISBN13: 978-0199273133)

List of Tools

- GitHub - for accessing labs and lab submissions
- Docker - for completing the labs on your laptops
- Free Software: MARS

  `http://courses.missouristate.edu/kenvollmar/mars/`
- Free Software: Logisim

  `http://www.cburch.com/logisim/index.html`

To Do

- Join Slack - link accessible through course webpage. [third button on the right hand side]

Things to do before next class (1)

Please read the course syllabus: Accessible through the course website. [button on the left hand side]

Things to do before next class (2)

- Get GitHub setup completed on your laptops:
- If you have not setup GitHub on your laptop previously, **no worries** watch the YouTube videos below and follow up with the Professor if you are facing issues with the setup!
  - `https://tinyurl.com/5hkfxef3`
  - `https://tinyurl.com/m84x3vrp`

Things to do before next class (2)

- Accept the class repository link by clicking on the GitHub icon in the course repository. [second button on the right hand side]

Things to do before Friday Lab (3)

- Get Docker setup completed on your laptops:
- Docker Mac Setup:
  https://docs.docker.com/docker-for-mac/install/
- Docker Ubuntu Setup
  https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04
- Docker Windows Setup:
  https://docs.docker.com/docker-for-windows/install/

  Feel free to follow up with the Professor if you are facing issues with the setup!

Things to do before Friday Lab (3)

- If the setup goes correctly as desired, you should be able to get started and run the hello world docker container using the following command:

  ```
  docker run hello-world
  ```

- There are some more documentation for Docker get started to test your installation in the link provided below:

  https://docs.docker.com/docker-for-mac/
  https://docs.docker.com/docker-for-windows/

In a nutshell

Three vital parts to this course are:

- C Programming
- Digital Logic and Circuit Design
- Assembly Language Programming

What will I learn in this class?

In the Patterson and Hennessy textbook, several questions are listed at the top of page 8; by the end of this course, you should know how to answer them, at least in part:

1. ”How are programs written in a high-level language, such as C or Java, translated into the language of the hardware, and how does the hardware execute the resulting program?”
2. ”What is the interface between the software and the hardware, and how does software instruct the hardware to perform needed functions?”

What will I learn in this class?

3. ”What determines the performance of a program, and how can a programmer improve the performance?”
4. ”What techniques can be used by hardware designers to improve performance?”
5. ”What are the reasons for and the consequences of the recent switch from sequential processing to parallel processing?”

Important things we will do in this class

1. Write, compile, and execute programs in the C programming language
2. Write, assemble, and execute programs in the MIPS assembly language
3. Explore different types of data (integer, floating-point, character, Boolean, etc.) are internally represented and manipulated in a computer's memory
4. Assemble basic logic gates into complex logic circuits (such as a processor datapath).

My expectations from YOU

1. Attending both lecture and lab sessions regularly (see attendance policy in course syllabus)
2. Interact and engage with the materials discussed by asking questions, doing the in-class activities, and doing group discussions as appropriate.
3. Bring a notebook and start making notes

My expectations from YOU

4. Come to Office hours with questions that needs clarification
5. Complete the reading assignments provided at the end of each topic
6. Accepting the fact that we are learning some core CS concepts in this course and enjoy the process of learning computer science

Why should I learn Computer Organization?

1. Why CS-200 is listed as a CORE course?
   - The course presents a series of concepts that lets you realize the internal gimmicks of computers.
   - Compare the fundamental features of Programming in C over MIPS and further experience the events occuring at the hardware level.
   - Explore the mathematics of machine computation.
2. JOBS:

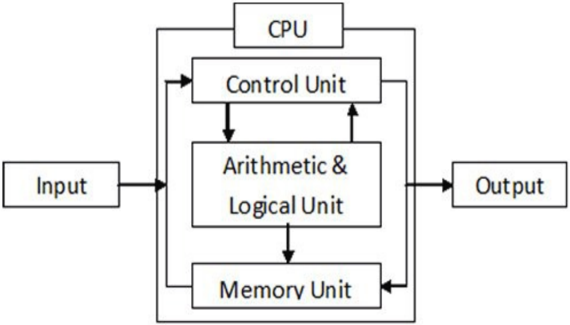
   Software engineer at Bank of America, BCBS

   Hardware engineer at Motorola, Sony

   Assembly language programmer at IBM, Intel

Inside Computers

1. Both instruction and data are stored and processed in binary form inside a computer.
2. Binary = **0's** and **1's**



Computer Specifications

1. ”I have a computer that has a 64-bit processor with 4 Gigabytes of RAM and 200 Gigabytes of hard disk space”

   Why memory space is lesser than hard disk?

2. Bit - the basic unit of information for computers. Can hold 0 or 1 values
3. Byte - equivalent to 8 bits. Each character in keyboard is stored as 1 Byte.

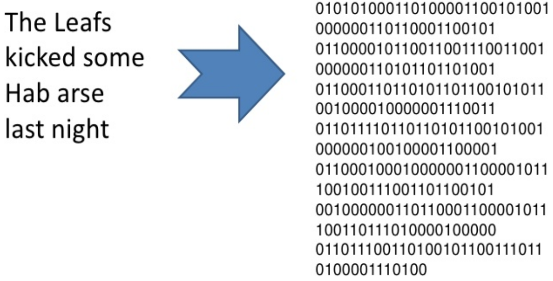
   Example: A = 01000001

4. Word - equivalent to 4 bytes. Half word is 2 bytes. This depends on processors.

Storage Capcities

1. 1 bit = 0 or 1 **(b)**
2. 8 bits = 1 byte **(B)**
3. 1000 bytes = 1 kilobyte **(KB)**
4. 1 million bytes = 1 megabyte **(MB)**
5. 1 billion bytes = 1 gigabyte **(GB)**
6. 1 trillion bytes = 1 terabyte **(TB)**

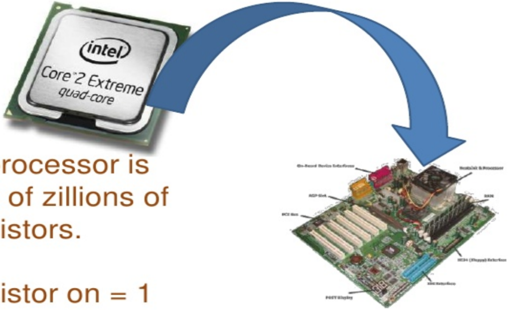
Text to binary conversion



The Leafs
kicked some
Hab arse
last night

```
010101000110100001100101001
000000110110001100101
011000010110011001110011001
000000110101101101001
011000110110101101100101011
001000010000001110011
011011110110110101100101001
000000100100001100001
011000100010000001100001011
100100111001101100101
001000000110110001100001011
100110111010000100000
011011100110100101100111011
0100001110100
```

Computer Processor and Bytes



The processor is made of zillions of Transistors.

Transistor on = 1
Transistor off = 0

How Binary nos are generated?

1. What is the maximum decimal that can be represented using 2 bits, 3 bits, and 4 bits?
2. Let us draw the 2-bit and 3-bit decimal to binary match table together!
3. Do the 4 bit table on your own.

Reminder

- **Lab 1:** on Friday.

Reading Assignment

- **PH:** 1.1 to 1.4

Questions?

Please ask if there are any Questions!