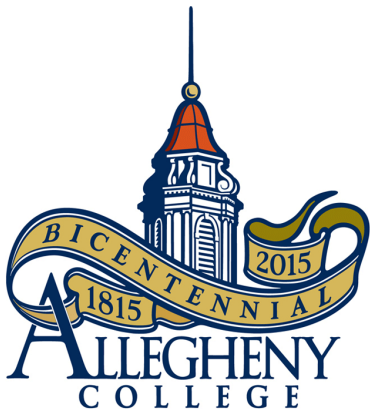
CS200 - Computer Organization

An Introduction

Aravind Mohan

Allegheny College

September 1, 2020



Meeting Time

- Lecture Session:
  - Tuesday and Thursday
    8:00 am - 9:15 am, Alden 101
- Lab Session:
  - Wednesday
    3:00 pm - 4:50 pm, Alden 101

Professor's Office Hours

- Monday, Wednesday, and Friday:
  11:15 am - 12:15 pm
- Tuesday and Thursday:
  10:00 am - 11:30 am

Send an email to schedule time outside office hours.

To schedule an office hours time slot, please visit my website [teaching page] and click on the
Schedule Meeting link located on the top right-hand corner to schedule 15 mins slots.
Let us connect and learn from each other...

Website Details

- Professor's Website:

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

- Course Website:

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

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)

Administrative Stuff!

- No Lab this week.

  First lab next week on Wednesday, 9th Sep 2020.

Administrative Stuff!

- Laboratory Assignments
- Skill Tests **(3)**
- Exams **(2)**
- Course Project
- Class Participation

Please read the **Syllabus** to get an overview of the course.

Administrative Stuff!

Laboratory Assignments	**35%**
Skill Tests	**15%**
MidTerm Exam	**10%**
Final Exam	**15%**
Course Project	**15%**
Class Participation	**10%**

Gradebook will be shared through Canvas. More details in **Syllabus**.

Tips for Success

- Attentively listen to classes and try to participate in all class discussions.
- Bring a notebook with you and start making detailed notes during every class period.
- Clarify with the Professor, if a lesson is confusing.
- Complete all the reading assignments thoroughly.
- Participate in all the in-class activities.

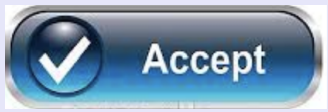
Be ready to **think**, **process**, and **implement** low-level operations

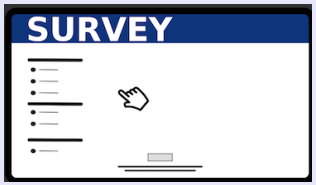
Interaction between us ...

- Any question is a valid question. No question is good or bad. So, questions are always welcome.
- Interaction is the best way to get rid of long lectures. So, let us try to interact more so that communication is a two-way stream and the class is not boring.

Ask your questions in Google Meet chat window.

Things To Do (1)





Read **Syllabus** before next class!

Things To Do (2) Offline

Install a gcc compiler on your laptop:

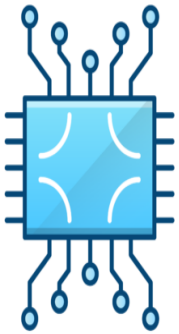
1. Ubuntu:
   - Open your terminal, and run the commands provided in the link below:
     `shorturl.at/DST45`
2. Mac:
   - Install homebrew if not already installed on your laptop:
     `shorturl.at/jnrxT`
   - Install gcc compiler:
     `shorturl.at/dmCGP`
3. Windows:
   - Install MinGW gcc compiler
     `shorturl.at/agCOV`

Reach out to Professor for any questions during installation!

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. ”What determines the performance of a program, and how can a programmer improve the performance?”
2. ”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?”

What will I learn in this class?

3. ”What is the interface between the software and the hardware, and how does software instruct the hardware to perform needed functions?”
4. ”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).

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:

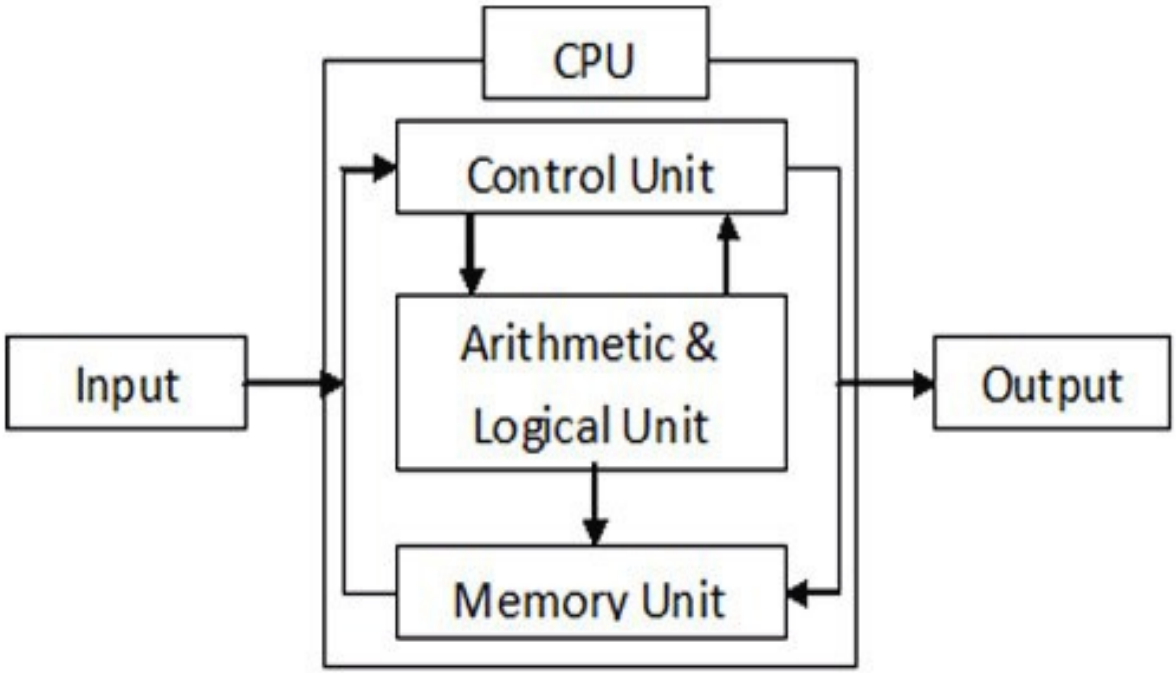
   Software engineer at Bank of America, BCBS

   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

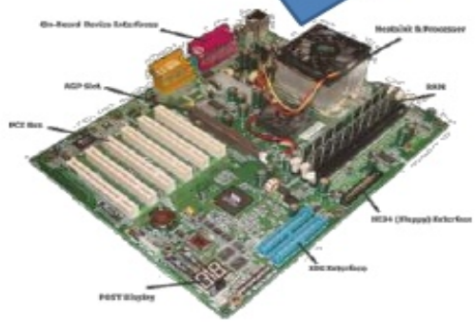
The Leafs
kicked some
Hab arse
last night

010101000110100001100101001
000000110110001100101
011000010110011001110011001
000000110101101101001
011000110110101101100101011
001000010000001110011
011011110110110101100101001
000000100100001100001
011000100010000000110000101 1
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.

To Do!

Brainstorm with your peers & come up with ideas to solve this problem?

- Add a reflection markdown file to the repository.
- Commit and Push changes.

Next Class:

Bits and Bytes:

- How does program store and manage data inside a computer?
- How can we assess the performance of a program from a hardware perspective?

Questions?

Please ask if there are any Questions!