



香港中文大學

The Chinese University of Hong Kong

CSCI2510 Computer Organization

Lecture 00: Course Information

Ming-Chang YANG

mcyang@cse.cuhk.edu.hk

Course Information



- CSCI2510 Computer Organization
- Course Time and Place
 - Lecture (*3)
 - MON 12:30~14:15 (@ ERB 404)
 - TUE 12:30~13:15 (@ ERB 404)
 - Tutorial (*1)
 - TUE 14:30~15:15 (@ LSB LT1)
- Course Website
 - <http://www.cse.cuhk.edu.hk/~mcyang/csci2510.html>

Course Instructor



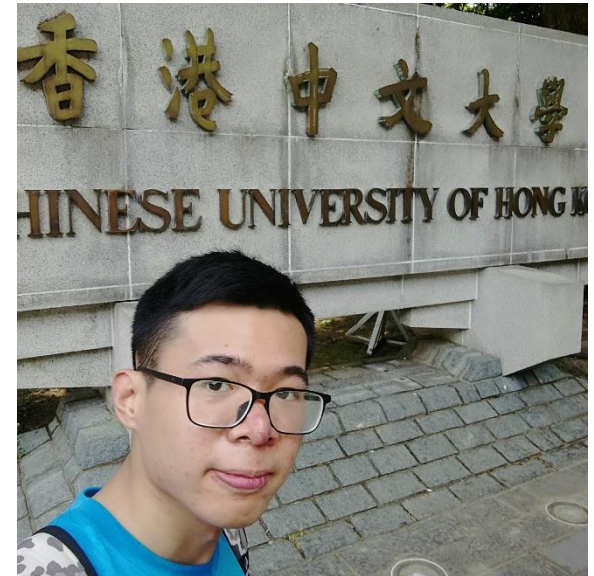
- Prof. Ming-Chang YANG (楊明昌)
 - Office: SHB 906 (3943-8405)
 - Office Hours: TUE 15:30~17:30
 - mcyang@cse.cuhk.edu.hk



Teaching Assistants



- Yuhong LIANG (梁裕宏)
 - Office: SHB 101
 - Office Hours: TUE 15:30~17:30
 - yhliang@cse.cuhk.edu.hk



- Bentian JIANG (江本田)
 - Office: SHB 913
 - Office Hours: MON 15:30~17:30
 - btjiang@cse.cuhk.edu.hk



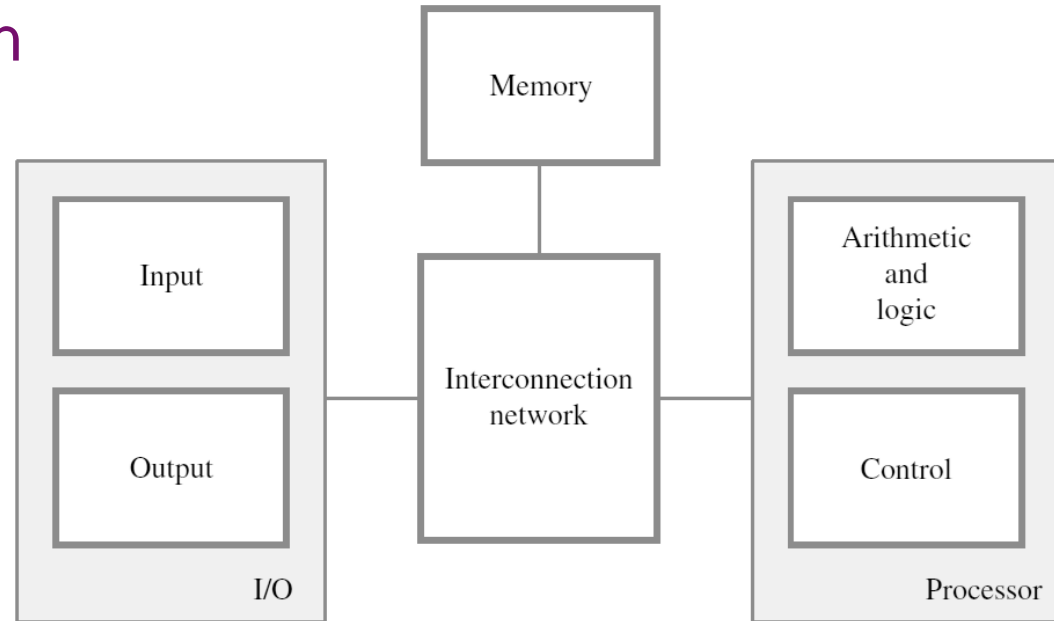
Course Description



- This course is designed to learn:

- Computer Organization

- Processor (CPU)
- Memory unit
- Input/Output units
- Interconnection buses



- Assembly Language Programming (as a tool to study)

- Internal coding of information
- Number representation
- Arithmetic operations
- Flow of information within a microcomputer

```
mov ecx, ebx
mov esp, edx
mov edx, r9d
mov rax, rdx
```

Why do we study this course?



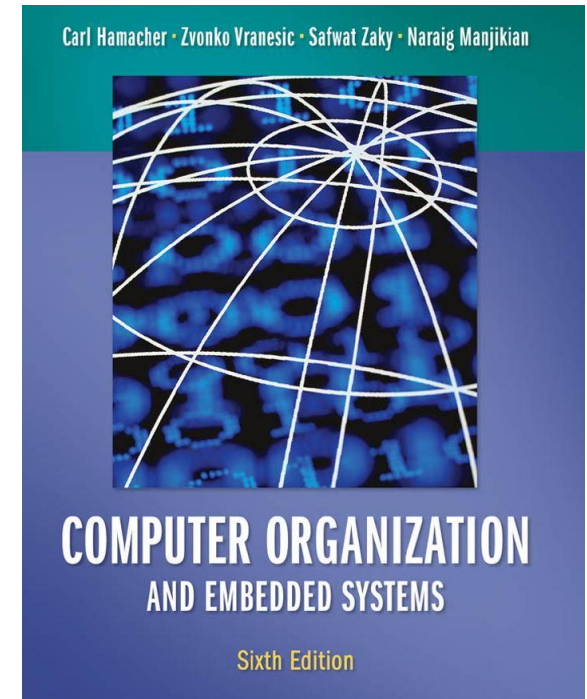
- Why do we study computer organization and assembly languages?
 - To understand how a computer works internally.
 - To maximize the efficiency of computer software.
 - To build NEW! computers.
- Why maximize efficiency?
 - To tackle programs and data of rapidly increasing sizes.
 - To save costs.
- How to maximize efficiency?
 - By understanding the limits of hardware.
 - By writing good software/program.

Course Objectives



- To understand the working and design principles of a modern computer system and its components.
- To acquire the ability of differentiating and comparing different computer systems.
- To gain hands-on experience of programming a computer using the assembly language in order to understand the inner workings of a microcomputer.

- Textbook
 - Computer Organization and Embedded Systems, 6th Ed.
 - Hamacher, Vranesic, Zaky, and Manjikian
 - McGraw Hill, 2012



Acknowledgement: Thanks to Michael Fung, Philip Leong (CUHK), Y.S. Moon (CUHK), O. Mencer (Imperial), N. Dulay (Imperial) for all slides used in this course.

Course Assessment



- Grading
 - Assignments 30%
 - Midterm Exam 25%
 - Final Exam 40%
 - Class Participation 5%
 - *Bonus* 5%
- Notes
 - Late submission per day is subject to 10% of penalty.
 - A student must attend at least 80% of lectures in order to gain all class attendance credits.

Course Schedule



**Date in red: Public holiday or class suspension*

W	Date	Lecture	Note
1	Sep 3, 4	Lec01 Basic Structure of Computers	
2	Sep 10, 11	Lec02 Number & Character Representation	
3	Sep 17, 18	Lec03 Memory Basics	HW1 & PG1
4	Sep 24, 25	Lec04 Machine Instructions	
5	Oct 1, 2	Public holiday (Oct 1), No class (Oct 2)	Tutorial (Oct 2) as usual
6	Oct 8, 9	Lec05 Program Execution	HW2 & PG2
7	Oct 15, 16	Lec06 Basic Input & Output	
8	Oct 22, 23	Lec07 Memory Hierarchy	Midterm Exam (Lec01~06)
9	Oct 29, 30	Lec08 Cache in Action	
10	Nov 5, 6	Lec09 Performance Considerations	HW3 & PG3
11	Nov 12, 13	Lec10 Virtual Memory	
12	Nov 19, 20	Lec11 Processor Internals	
13	Nov 26, 27	Lec12 Pipelining	HW4
14	Dec 3	Lec13 Control Unit	Make-up Class (Oct 2)

Programming Tools



- Microsoft Macro Assembler v14 (under Microsoft Visual Studio 2015)
 - Community Edition:
 - Free for Genuine Windows users
 - Full-featured industrial-grade software
 - Usage Guideline:
 - Install **Visual Studio Community 2015**
 - <https://www.visualstudio.com/>
 - Create C/C++ Project and accept default MASM/ML build rule (details to be discussed in tutorials)



Important Notes



- Visit our course website regularly
- Plagiarism will **NOT** be tolerated
 - Don't copy!
 - Don't let other(s) copy!
 - Can discuss but write up the solutions by yourself!
- Honesty in Academic Work:
 - <http://www.cuhk.edu.hk/policy/academichonesty/>

The best way to learn is through practice!