Introduction
(AY2016/7 Semester 2)
Why Learn Computer Organisation? (From user to builder)

- You want to call yourself a computer scientist/specialist.
- You want to build software people use.
- You need to make purchasing decisions.
- You need to offer “expert” advice.
- Hardware and software affect performance
  - Algorithm determines number of source-level statements (eg: CS1010, CS1020, CS3230)
  - Language, compiler, and architecture determine machine instructions (COD chapters 2 and 3)
  - Processor and memory determine how fast instructions are executed (COD chapters 5, 6 and 7)
- Understanding performance (COD chapter 4)
So, what’s a computer? (1/6)

Example: An automobile augments our power of locomotion.

A computer is a device capable of solving problems according to designed programs. It simply augments our power of storage and speed of calculation.
So, what’s a computer? (2/6)

- From computer organisation perspective, we study the components and how they work together
  - Processor, memory, input/output devices, networks, …

So, what’s a computer? (3/6)

1. Power supply
2. Motherboard
3. Central Processing Unit (CPU)
4. Random Access Memory (RAM)
5. Hard drive
6. Cooling fan
7. I/O devices

Once you are acquainted with about the dissimilar parts that make up a total computer it gets a lot easier. Today’s computer consists of around eight main devices; some of the advanced computers might have a few additional mechanisms. What are these eight main components and what are they used for? We will start with beginner level facts to get you in progress.

First is the Power Supply. The authority provides is used to provide electrical...
So, what’s a computer? (4/6)

- PC motherboard
- Pentium processor

Credit: http://www.computer-hardware-explained.com/what-is-a-motherboard.html
So, what’s a computer? (5/6)

- Computer Organisation

- Computer
  - CPU
    - Control
    - Datapath
  - Memory
  - Devices
    - Input
    - Output
So, what’s a computer? (6/6)

- Next generation…


Intel announces Edison, a computer the size of an SD card

By Sean Hollister on January 6, 2014 10:01 pm

Delving into depth reveals more information

Abstraction omits unnecessary details
ABSTRACTION (2/2)

- Need to learn abstractions such as:
  - Application and system software
  - Assembly language and machine language
  - Architectural issues such as pipelining, caches, virtual memory
  - Combinational logic, arithmetic circuits
  - Sequential logic, finite state machines
  - Boolean logic (1s and 0s)
  - Transistors used to build logic gates (CMOS)
  - Semi-conductors/silicon used to build transistors
  - Properties of atoms, electrons and quantum dynamics

- So much to learn!
Coordination of many *levels of abstraction*

Under a rapidly *changing set of forces*

Design, measurement, *and* evaluation
LEVELS OF REPRESENTATION

High Level Language Program (e.g., C)

Compiler

Assembly Language Program (e.g., MIPS)

Assembler

Machine Language Program (MIPS)

Machine Interpretation

Hardware Architecture Description (Logic, Logisim, etc.)

Architecture Implementation

Logic Circuit Description (Logisim, etc.)

temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;

lw $t0, 0($s2)
lw $t1, 4($s2)
sw $t1, 0($s2)
sw $t0, 4($s2)

0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
SYLLABUS OUTLINE (REVISIT)

- Number systems
- Boolean algebra
- Logic gates and circuits
- Simplification
- Combinational circuits
- Sequential circuits
- Assembly language
- Performance
- The processor: Datapath and control
- Pipelining
- Memory hierarchy: Cache
- Input/output

Preparation: 2 weeks
Logic Design: 2 weeks
Computer Organisation
PRACTICAL ASPECTS

- Practical experience
  - Logic design experiments
  - Logisim software
  - QTSpim software
QUOTES

- “What I hear, I forget.
  What I see, I remember.
  What I do, I understand” – Chinese Proverb
  不闻不若闻之，闻之不若见之，见之不若知之，知之不若行之；学至于行之而止矣。
  – 《荀子·儒效》

- “The important thing is to understand what you are doing, rather than to get the right answer” – Tom Lehrer