Chapter0: Intro

1. Homework 20%

- 9次Homework按点给分，最后折算到19%
- 问卷占比1%
- 截止时间：两天后17：30

2. Lab 40%

- 每个lab三次实验课时间验收
- 报告ddl：验收截止后两天
- 验收流程

> 要求代码传学在浙大，助教从学在浙大下载下来。
>
> 先用autograder跑，全通过就让同学描述其算法，再问一到两个问题考察其理解。

- **Penalty**

> Wrong Answer: -10% of Check part each time.
> Delay: -20% of the corresponding part per day.
> **Cheating: -100% of this lab. Additionly, -10% of the final score of this course.**
> **that upload your answer to the Internet is also CHEATING!!!**

Lab1	Lab2	Lab3	Lab4	Lab5	Lab6
4%	5%	6%	7%	8%	10%
machine code	basic assembly	data structure	interruption	recursion	assembler or executor (in C)

3. Final 40%

- Final 秋学期第一周周五下午，两个半小时，闭卷
- 斩杀线：往年为50分

4. Tip

- 面对面建群
- 介绍LC3
- 这门课的学习方法
- **不能用第二版书**
- 每周的安排

Chapter1: Welcome Aboard

1. Two major themes

- Abstraction

  > A technique for establishing a simpler way for a person to interact with a system, removing the details that are unnecessary.
  >
  > e.g. taxi; turn on the light in a dark room

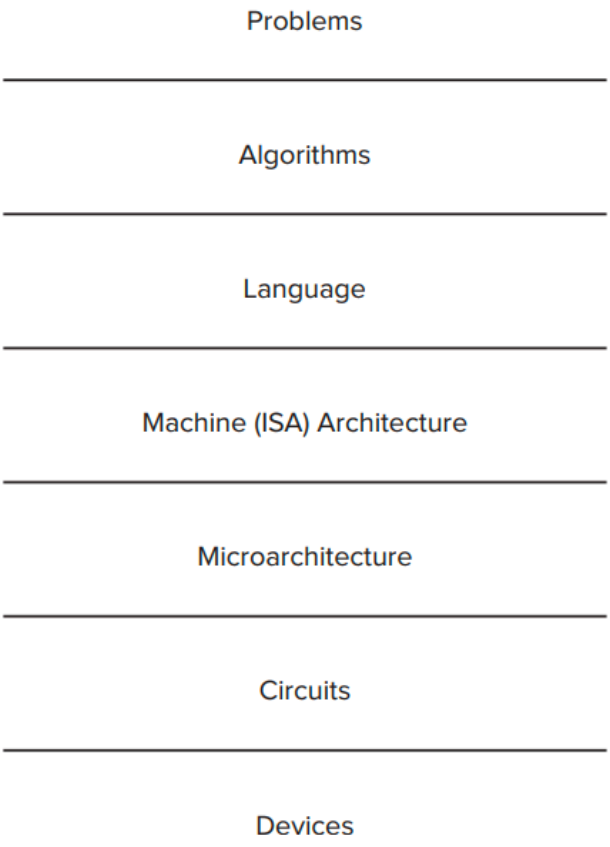
- Hardware VS Software

2. Computer system

- CPU (central processing unit/processor/microprocessor)
- Memory
- I/O (peripheral)

> All computers (the biggest and the smallest, the fastest and the slowest, the most expensive and the cheapest) are capable of computing exactly the same things if they are given enough time and enough memory.

3. Levels of transformation



- Problems

  > ambiguity

- Algorithm

  > step-by-step procedure

    - definiteness
    - effective computability
        > each step can be carried out by a computer
    - finiteness

- Program

  > high level (c) -- compiler;
  > low level (assembly language) -- assembler

- **ISA (Instruction Set Architecture)**

  > opcodes, data types, and **addressing modes**

- Microarchitecture

  > the implementation of the ISA
  > **an ISA correspondes to many microarchitecture;**
  > **an microarchitecture correspondes to only one ISA;**

- Logical Circuit
- Electronic Circuit
- Electrons

Chapter 2: Bits, Data Types, and Operations

1. Bit code

Same pattern of bits correspondes to different meaning.

2. Data Type

Representation	Value Represented			
	Unsigned	**Signed Magnitude**	**1's Complement**	**2's Complement**
00000	0	0	0	0
00001	1	1	1	1
00010	2	2	2	2
00011	3	3	3	3
00100	4	4	4	4
00101	5	5	5	5
00110	6	6	6	6
00111	7	7	7	7
01000	8	8	8	8
01001	9	9	9	9
01010	10	10	10	10
01011	11	11	11	11
01100	12	12	12	12
01101	13	13	13	13
01110	14	14	14	14
01111	15	15	15	15
10000	16	−0	−15	−16
10001	17	−1	−14	−15
10010	18	−2	−13	−14
10011	19	−3	−12	−13
10100	20	−4	−11	−12
10101	21	−5	−10	−11
10110	22	−6	−9	−10
10111	23	−7	−8	−9
11000	24	−8	−7	−8
11001	25	−9	−6	−7
11010	26	−10	−5	−6
11011	27	−11	−4	−5
11100	28	−12	−3	−4
11101	29	−13	−2	−3
11110	30	−14	−1	−2
11111	31	−15	−0	−1

- Unsigned Integer

> $5 = 00101$
> range: $[0, 2^k - 1]$
>
> overflow

- Signed Integer

- Signed Magnitude
  > range: $[-(2^{k-1}-1), 2^{k-1}-1]$
- 1's Complement (反码)
  > range: $[-(2^{k-1}-1), 2^{k-1}-1]$
- 2's Complement (补码)
  > range: $[-(2^{k-1}), 2^{k-1}-1]$
  > -16的相反数无法表示
- Conversion between
- Addition and Subtraction
  > Signed Magnitude:
  >
  > > $00000001 + 10000001 = 10000010$
  > > 需要对符号位单独处理，增加了硬件开销
  >
  > 1's Complement
  >
  > > $00000001 + 11111110 = 11111111$
  > > 两个0，浪费空间
  >
  > 2's Complement
  >
  > > 补码加减本质：加y正向移动(+1)y次，减y反向移动(-1)y次（参照patt画的那个循环）
  > >
  > > Sign Extension
  > >
  > > > leading 0s do not affect the value of a positive number, leading 1s do not affect the value of a negative number
  > > >
  > > > 不同位数的整数加减时需要对齐
  > >
  > > Overflow
  > >
  > > > positive + positive = negative;
  > > >
  > > > negative + negative = positive;
  > > >
  > > > **Q:** Why does the sum of a negative 2’s complement number and a positive 2’s complement number never generate an overflow?

- Logical Variable
  > Logical variable: 0 or 1
  >
  > - AND
  > - OR
  > - NOT (complement operation)
  > - XOR (Exclusive-OR)

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Bit Vector

> An m-bit pattern where each bit has a logical value (0 or 1) independent of the other bits is called a bit vector, some of the bits identify the presence of the property and other bits identify the absence of the property.
>
> bit mask作用：用OR和1置位，用AND和0复位，用AND和1查询

- ASCII (American Standard Code for Information Interchange)

  > eight-bit code

- Hexadecimal Notation

  > more convenient

- Floating Point (To be continued)