Homework 1

> 2023.07.12 沈韵泓 3200104392

Chap 1

1.2 [1] Can a higher-level programming language instruct a computer to compute more than a lower-level programming language?

No.

Since both higher-level programming language & lower-level programming language would be ultimately translated into the machine language (the lowest), they have the same ability to instruct a computer to computer.

1.4 [1] Name one characteristic of natural languages that prevents them from being used as programming languages.

The *ambiguity* .

According to the textbook, the sentence "Time flies like an arrow" has at least 3 possible interpretations.

=> Those ambiguity-caused multiple interpretations would cause the computer to not know which interpretation to follow, preventing natural languages from being used as programming languages.

1.10 [1] Name three characteristics of algorithms. Briefly explain each of these three characteristics.

1. `Detiniteness`： Each step should be precisely stated.
2. `Effective Computability`： Each step can be carried out by a computer。
3. `Finiteness`： The procedure could ultimately terminate.

1.16 [1] Name at least three things specified by an ISA.

1. The interface between the computer program directing the computer hardware and the hardware carrying out those directions.
2. The set of instructions that the computer can carry out.
3. `data type`： The acceptable representations for operands.

4. `addressing mode`：The mechanisms that the computer can use to figure out where the operands are located.
5. The number of unique locations that comprise the computer's memory
6. The number of individual 0s and 1s that are contained in each location.

1.18 [4] How many ISAs are normally implemented by a single microarchitecture? Conversely, how many microarchitectures could exist for a single ISA?

- A single microarchitecture could normally implemente **ONLY ONE** ISA.
- A single ISA could be implemented by **MANY** microarchitectures (which means INFINITE).

Chap 2

2.4 Given *n* bits, how many unsigned integers can be represented with the *n* bits? What is the range of these integers?

- `n bits` can represent 2^n unsigned integers.
- The range of these integers is $[0\ ,\ 2^n - 1]$

2.8
a. What is the largest positive number one can represent in an eight-bit 2's complement code? Write your result in binary and decimal.
b. What is the greatest magnitude negative number one can represent in an eight-bit 2's complement code? Write your result in binary and decimal.
c. What is the largest positive number one can represent in *n*-bit 2's complement code?
d. What is the greatest magnitude negative number one can represent in *n*-bit 2's complement code?

a. The largest positive number can be represented by 8-bit 2's complement code is:

$$2^{8-1} - 1 = (127)_{Decimal}\ \&\ (0111\ 1111)_{Binary}$$

b. The greatest magnitude negative number can be represented by 8-bit 2's complement code is:

$$-2^{8-1} = (-128)_{Decimal}\ \&\ (1000\ 0000)_{Bianry}$$

c. The largest positive number can be represented in n-bit 2's complement code is:

$$(2^{n-1} - 1)_{Decimal}$$

d. The greatest magnitude negative number can be represented in n-bit 2's complement code is:

$$(-2^{n-1})_{Decimal}$$

2.17 Add the following 2's complement binary numbers. Also express the answer in decimal.

a. 01 + 1011
b. 11 + 01010101
c. 0101 + 110
d. 01 + 10

a. 01 + 1011 =

$$= 0001 + 1011$$
$$= (1100)_{Binary} = (-4)_{Decimal}$$

b. 11 + 01010101 =

$$= 1111\ 1111 + 0101\ 0101$$
$$= (0101\ 0100)_{Binary} = (84)_{Decimal}$$

c. 0101 + 110 =

$$= 0101 + 1110$$
$$= (0011)_{Binary} = (3)_{Decimal}$$

d. 01 + 10 =

$$= (11)_{Binary} = (-1)_{Decimal}$$

2.20 The following binary numbers are four-bit 2's complement binary numbers. Which of the following operations generate overflow? Justify your answer by translating the operands and results into decimal.

a. 1100 + 0011
b. 1100 + 0100
c. 0111 + 0001
d. 1000 − 0001
e. 0111 + 1001

index	overflow	operands	result
a	No	(-4) + 3	-1
b	No	(-4) + 4	0
c	Yes	7 + 1	-8
d	Yes	(-8) - 1	7
e	No	7 + (-7)	0

2.34 Compute the following:

a. NOT(1011) OR NOT(1100)
b. NOT(1000 AND (1100 OR 0101))
c. NOT(NOT(1101))
d. (0110 OR 0000) AND 1111

a. NOT(1011) OR NOT(1100)

$$\begin{aligned} &= 0100 \; or \; 0011 \\ &= 0111 \end{aligned}$$

b. NOT(1000 AND (1100 OR 0101))

$$\begin{aligned} &= not(1000 \; and \; 1101) \\ &= not(1000) \\ &= 0111 \end{aligned}$$

c. NOT(NOT(1101))

$$\begin{aligned} &= not(0010) \\ &= 1101 \end{aligned}$$

d. (0110 OR 0000) AND 1111

$$\begin{aligned} &= 0110 \; and \; 1111 \\ &= 0110 \end{aligned}$$