**Chapter 1**

**1.3 What difficulty with analog computers encourages computer designers to use digital designs?**

1) Analog machines produced an answer by measuring some physical quantity. Therefore, the difficulty with analog machines is that it is very hard to increase their accuracy.

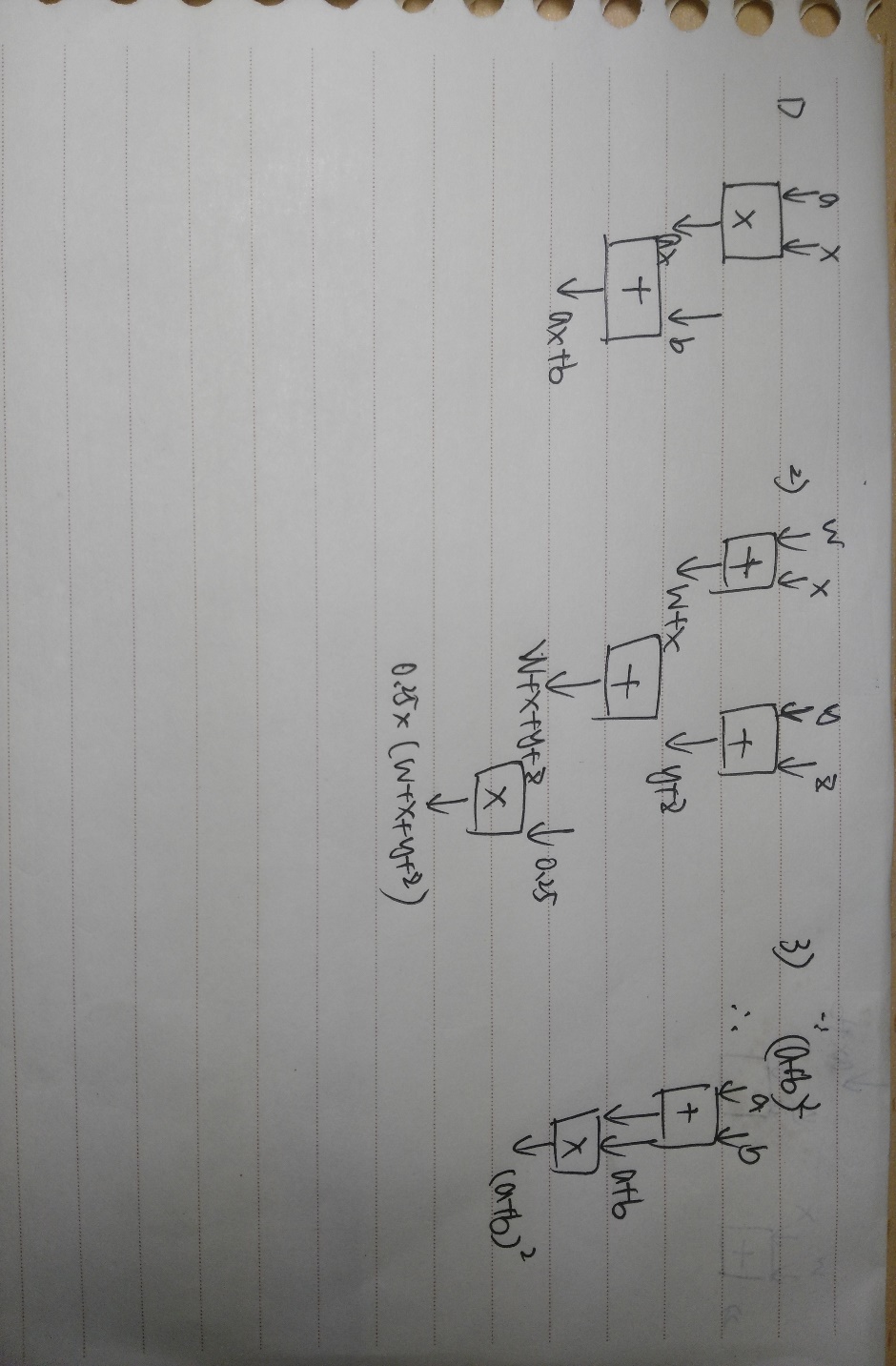
2) Digital machines perform computations by manipulating a fixed finite set of digits or letters – came to dominate computing. In addition, compared with digital computers before modern computer such as digital adding machines, when you think of a new kind of computation, you do not have to buy or design a new modern computer.

**1.5 Say we had a “black box”, which takes two numbers as input and outputs their sum. Say we had another box capable of multiplying two numbers together. We can connect these boxes together to calculate p \* (m + n). Assume we have an unlimited number of these boxes. Show how to connect them together to calculate.**

**1)ax + b**

**2)the average of the four input numbers w, x, y and z**

**3)a^2+2\*ab+b^2(can you do it with one add box and one multiply box?)**

****

**1.6 Write a statement in a natural language and offer two different interpretations of that statement.**

G20 was not held in a day. (just a funny complaint since I think it’s kind of ridiculous even to close universities and stop experiments just for a some kind of important meeting)

1. G20, the important meeting, can’t be well organized in a short time. The government are making effort to well prepare.
2. Sometimes, the government are just dragging too much people in and making overmuch effort to beautify the city and disrupt people’s normal life.

**1.7 The discussion of abstraction in Section 1.3.1 noted that one does not need to understand the makeup of the components as long as “everything about the detail is just fine.” The case was made that when everything is not fine, one must be able to deconstruct the components, or be at the mercy of the abstractions. In the taxi example, suppose you did not understand the component, that is, you had no clue how to get to the airport. Using the notion of abstraction, you simply tell the driver “Take me to the airport.” Explain when this is a productivity and when it could result in very negative consequences.**

1) Productivity: If the taxi driver is a trustful man, the driver will get you to the destination as cheaply and quickly as possible. Therefore, you don’t need to know and tell the driver the exact way to the airport. In addition, you will feel more comfortable.

2) Negative consequences: If the taxi driver is not as honorable as you imagine, he may get you to the destination late enough or even lead you to miss your flight as Professor Patt mentioned in the class. Or the driver may fare too much than what it should have been.

**1.8 John said, “I saw the man in the park with a telescope.” What did he mean? How many reasonable interpretations can you provide for this statement? List them. What property does this sentence demonstrate that makes it unacceptable as a statement in a program?**

1) I saw the man who was in the park and took a telescope with him.

2) I used a telescope to see the man who was in the park.

3) I was in a park and saw the man who took a telescope with him.

4) I was in a park and used a telescope to see the man.

5) The property this sentence demonstrate that makes it unacceptable as a statement in a program is that its ambiguity. To infer the meaning of a sentence, a listener is often helped by the tone of voice of the speaker, or at the very least, the context of the sentence.

**1.13 Two computers, A and B, are identical except for the fact that A has a subtract instruction and B does not. Both have add instructions. Both have instructions that can take a value and produce the negative of that value. Which computer is able to solve more problems, A or B? Prove your result.**

A and B can solve same problems. Even though B can not perform a subtract instruction, B can change the second number to the corresponding negative value and add it to the first number. Therefore, both them have the same ability to solve same problems.

**1.15 Identify one advantage of programming in a higher-level language compared to a lower-level language. Identify one disadvantage.**

Advantage: Higher-level language is usually easier to write and understand. Therefore, it can save more time when coding.

Disadvantage: Compared with lower-level language, higher-level language usually executes less efficiently.

**1.16 Name at least three things specified by an ISA.**

1)Operand: the set of instructions the computer can carry out, that is, what operations the computer can perform and what data is needed by each operation

2)Data Type: a legitimate representation for an operand

3)Addressing Modes: the mechanisms that the computer can use to figure out where the operands are located

**1.17 Briefly describe the difference between an ISA and a microarchitecture.**

ISA is the interface to the computer with 0s and 1s. It contains things mentioned above, such as operands, data types and addressing modes. However, compared with a microarchitecture, it does not include those true physical things which microarchitecture owns.

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

One ISA is implemented by a single microarchitecture, but many microarchitectures could exist for a single ISA.

**Chapter 2 Bits, Data Types, and Operations**

**2.8 a. What is the largest positive number one can represent in an 8-bit 2's**

**complement code? Write your result in binary and decimal.**

D: 127 B: 01111111

**b. What is the greatest magnitude negative number one can represent in**

**an 8-bit 2's complement code? Write your result in binary and**

**decimal.**

D: -128 B: 10000000

**c. What is the largest positive number one can represent in n-bit 2's**

**complement code?**

D: 2^(n-1)-1

**d. What is the greatest magnitude negative number one can represent in**

**n-bit 2's complement code?**

D: -2^(n-1)

**2.10 Convert the following 2's complement binary numbers to decimal.**

**a. 1010** -6

**b. 01011010** 90

**c. 11111110** -2

**d. 0011100111010011** 14803

**2.11 Convert these decimal numbers to 8-bit 2's complement binary numbers.**

**a. 102** 01100110

**b. 64** 01000000

**c. 33** 00100001

**d. -128** 10000000

**e. 127** 01111111

**2.29 Fill in the following truth table for a one-bit AND operation.**

**X Y XANDY**

**0 0** 0

**0 1** 0

**1 0** 0

**1 1** 1

**2.30 Compute the following. Write your results in binary.**

**a. 0 1 0 1 0 1 1 1 AND 1 1 0 1 0 1 11** 01010111

**b. 1 0 1 AND 110** 100

**C. 1 1 1 0 0 0 0 0 AND 10110100** 110100000

**d. 0 0 0 1 1 1 1 1 AND 10110100** 00010100

**e. ( 0 0 1 1 AND 0110) AND 1101** 0000

**f . 0011 AND (0110 AND 1101)** 0000

**2.33Compute the following:**

**a. 0 1 0 1 0 1 1 1 OR 1 1 0 1 0 1 11** 11010111

**b. 1 0 1 OR 110** 111

**c. 1 1 1 0 0 0 0 0 OR 1 0 1 1 0 1 00** 11110100

**d. 0 0 0 1 1 1 1 1 OR 1 0 1 1 0 1 00** 10111111

**e. ( 0 1 0 1 OR 1 1 0 0 ) OR 1 1 01** 1101

**f . 0 1 0 1 OR (1100 OR 1101)** 1101

**2.34 Compute the following:**

**a. NOT ( 1 0 1 1 ) OR N O T ( l l O O)** 0111

**b. NOT ( 1 0 0 0 AND (1100 OR 0 1 0 1 ))** 0111

**c. NOT (NOT ( 1 1 0 1 ) )** 1101

**d. (0110 OR 0000) AND 1111** 0110

**2.53 Fill in the truth table for the equations given. The first line is done as an**

**example.**

***Q1*= NOT(A AND B)**

***Q2 =* NOT(NOT(A) AND NOT(B))**

**A B Q1 Q2**

**0 0 1 0**

1 0 1 1

0 1 1 1

1 1 0 1

**2.54 Fill in the truth table for the equations given. The first line is done as an**

**example.**

***Q* 1= NOT(NOT(X) OR (X AND Y AND Z))**

***Q2 =* NOT((Y OR Z) AND (X AND Y AND Z))**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | Z | Q1 | Q2 |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |