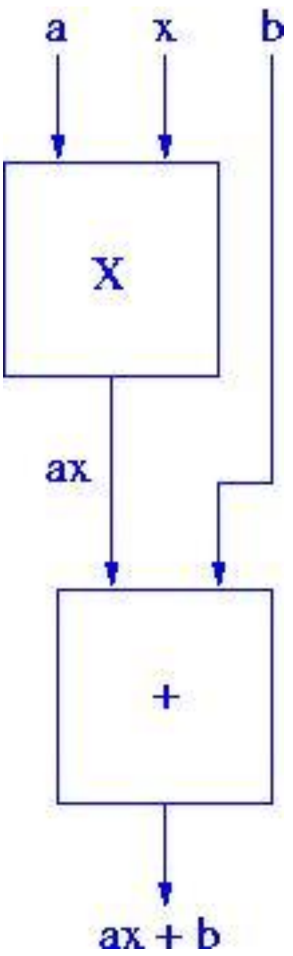
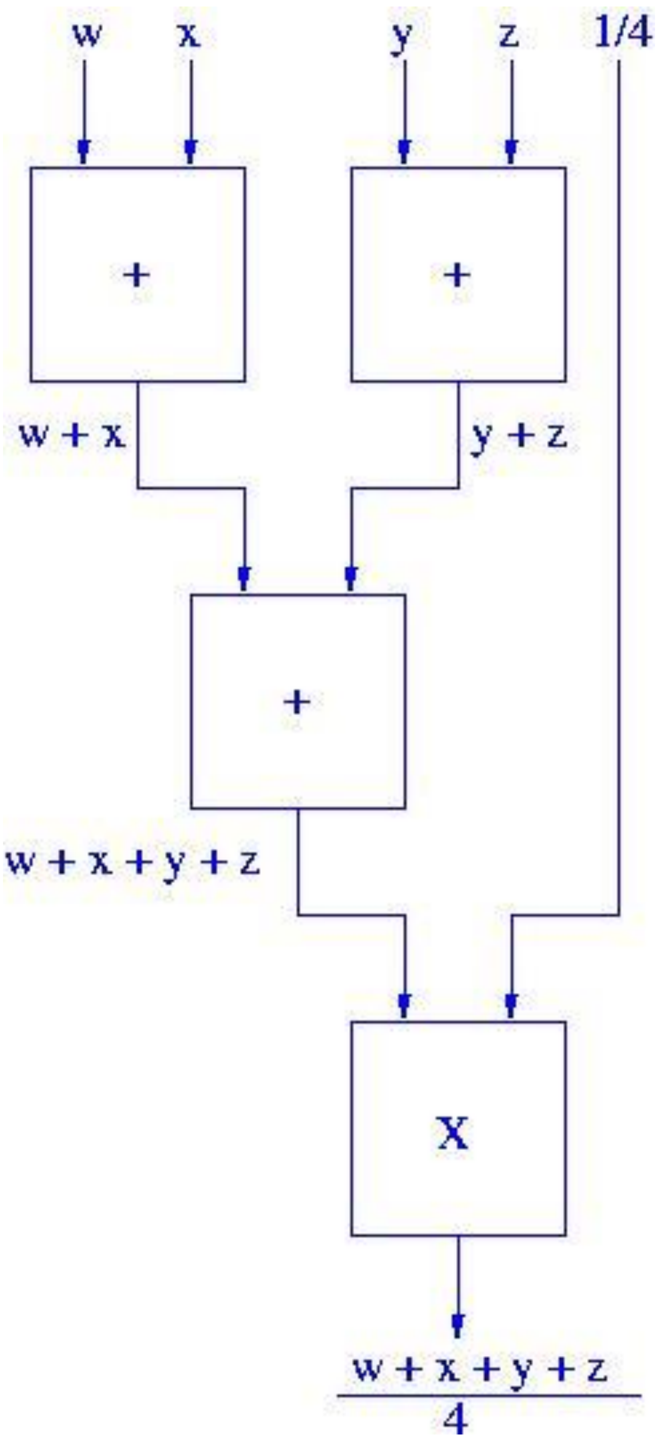
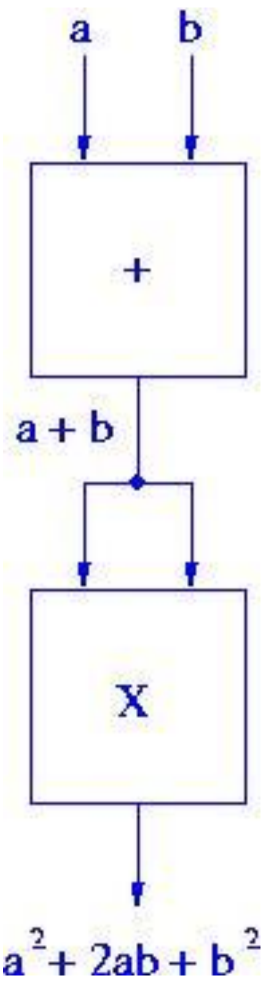
1. (Adapted from problem 1.5 in the textbook)
   Say we had a "black box," which takes two numbers as input and outputs their sum. See Figure 1.10a in the Textbook or the following figure. Say we had another box capable of multiplying two numbers together. See figure 1.10b. We can connect these boxes together to calculate p * (m + n). See Figure 1.10c. Assume we have an unlimited number of these boxes. Show how to connect them together to calculate:
   a. ax+b



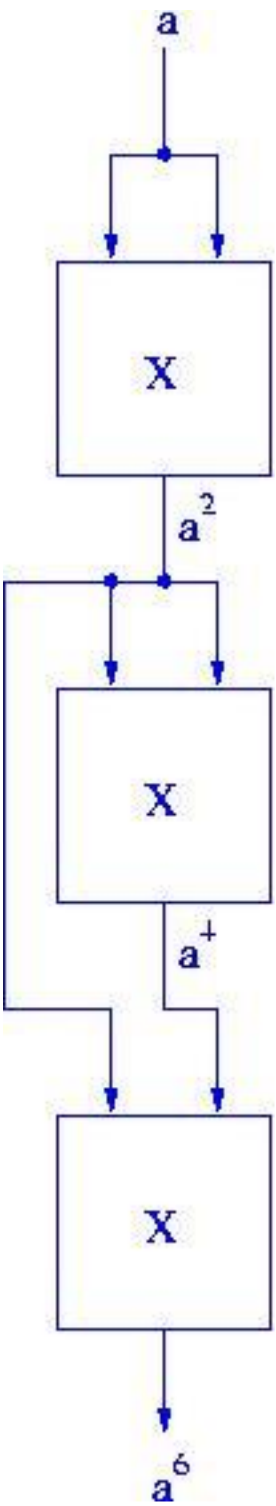
   b. The average of the four input numbers w, x, y, and z



c. $a^2 + 2ab + b^2$ (can you do it with one add box and one multiply box?)



d. a^6 (can you do it using only 3 multiply boxes?)



2. (1.14)
   Suppose we wish to put a set of names in alphabetical order. We call the act of doing so sorting. One algorithm that can accomplish that is called the bubble sort. We could then program our bubble sort algorithm in C, and compile the C program to execute on an x86 ISA. The x86 ISA can be implemented with an Intel Core microarchitecture. Let us call the sequence "Bubble Sort, C program, x86 ISA, Core

microarchitecture" one transformation process.
Assume we have available four sorting algorithms and can program in C, C++, Pascal, Fortran, and COBOL. We have available compilers that can translate from each of theses to either x86 or SPARC, and we have available three different microarchitectures for x86 and three different microarchitectures for SPARC.

a. How many transformation processes are possible?

   120

b. Write three examples of transformation processes.

   1. Bubble sort, c++, x86, Core microarchitecture
   2. Bubble sort, COBOL, x86, Core microarchitecture
   3. Bubble sort, Pascal, x86, Core microarchitecture

c. How many transformation processes are possible if instead of three different microarchitectures for x86 and three different microarchitectures for SPARC, there were two for x86 and four for SPARC.

   120

3. (2.3)
   a. Assume that there are about 400 students in your class. If every student is to be assigned a unique bit pattern, what is the minimum number of bits required to do this?

      9

   b. How many more students can be admitted to the class without requiring additional bits for each student's unique bit pattern?

      112

4. (Adapted from 2.13)
   Without changing their values, convert the following 2's complement binary numbers into 8-bit 2's complement numbers.
   a. 010110

      0001 0110

   b. 1101

      1111 1101

c. 1111111000

   11111000

d. 01

   00000001

5. (Adapted from 2.17)
   Compute the following. Assume each operand is a 2's complement binary number.
   a. 01 + 1011

      1100

   b. 11 + 01010101

      01010100

   c. 0101 + 110

      0011

   d. 01 + 10

      11

6. Without changing their values, convert the following 8-bit 2's complement binary numbers into decimal numbers.
   a. 01010101

      85

   b. 10001101

      -115

   c. 10000000

      -128

   d. 11111111

      -1

7. Express the value 0.3 in the 32-bit floating point format that we discussed in class today. Feel free to only show fraction bits [22:15], rather than all the fraction bits, [22:0]. Notation: The symbol [22:15] signifies all 8 bits from bit 22 to bit 15.

   0 01111101 00110011

8. Convert the following floating point representation to its decimal equivalent:

   1 10000010 10101001100000000000000

   -13 19/64

9. (Adapted from 2.50)

   Perform the following logical operations. Express your answers in hexadecimal notation.
   a. xABCD OR x9876

      xBBFF

   b. x1234 XOR x1234

      x0000

   c. xFEED AND (NOT(xBEEF))

      x4000

10. (2.54)

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

    Q1 = 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
0	1	1	0	1

1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0