Introduction to Computing Systems
Homework 1

PB18111697 王章瀚

November 3, 2019

1 (Adaptedfromproblem1.5inthetextbook)

The answers are as following.

a.

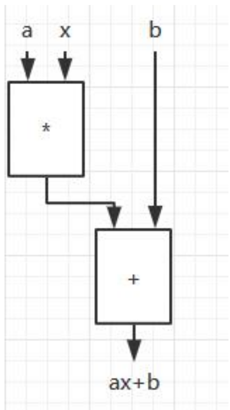

Figure 1: 1.a $ax + b$

b.

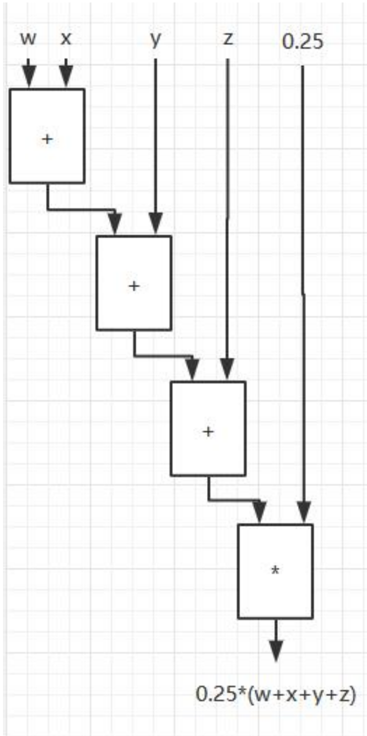

Figure 2: 1.b average of x, y, z, w

c.

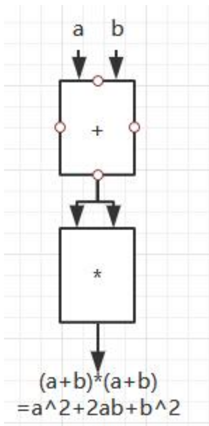

Figure 3: 1.c $a^2 + 2ab + b^2$

d.

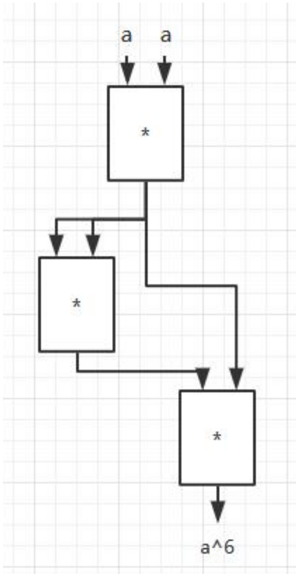

Figure 4: 1.d a^6

e.

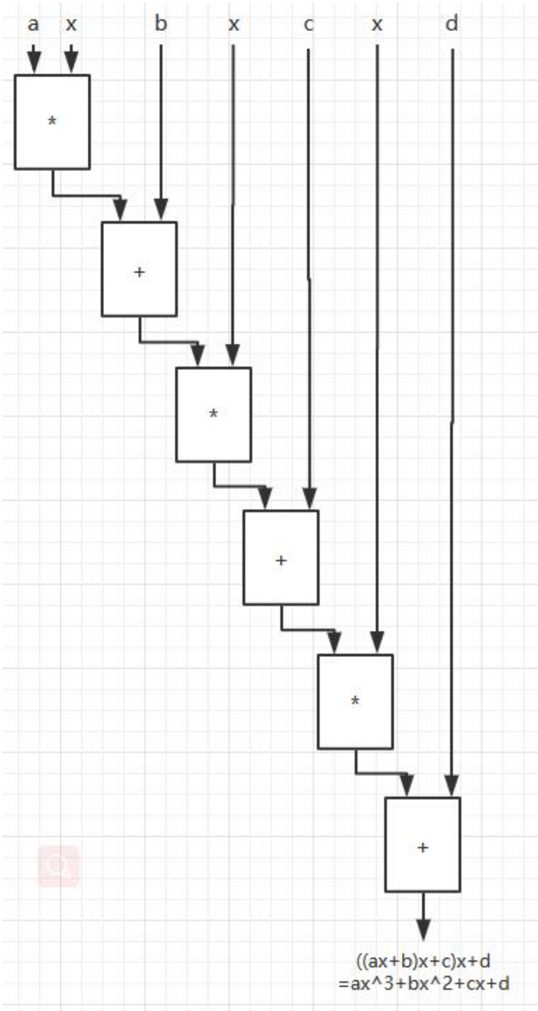

Figure 5: 1.e $ax^3 + bx^2 + cx + d$

2 (Adaptedfromproblem1.12inthetextbook)

a.

No. It lacks definiteness. This description ignored the sequence of all steps, which means we can add row1 to row2 then row1 to row4, as well as the other sequence. Even though all kinds of sequences can result the same answer, it still lacks definiteness.

b.

No. It lacks finiteness. It is obviouly that there are an infinite number of prime numbers and natural numbers, which means the algorithm can't be finite.

c.

Yes. It doesn't lack any of definiteness, effective computability or finiteness.

d.no. finiteness

No. It seems to lack the definiteness and the effective computability. First, on each flip, the state of the coin is unpredictable, which renders the description missing definiteness. What's more, the computers nowadays cannot really generate a random number, making the item uncomputable.

e.no. x-1. finiteness

The item is an algorithm when and only when the number given is a positive integer. Suppose the number firstly given as x. After the first 6 steps, the number becomes x-1. This ensures all positive integer can finally become 0, while other numbers can't; so if the number given isn't a positive integer, then the item turns out to be infinite.

3 (2.3)

a.

$log_2 400 \approx 8.6$, so the minimum number of bits required is 9.

b.

$2^9 - 400 = 112$, so 112 students can be admitted.

4 (2.8)

a.

$0111\ 1111_2, 127_{10}$

b.

$1000\ 0000_2, -128_{10}$

c.

$(2^{n-1} - 1)_{10}$

d.

$(2^{n-1})_{10}$

5 (Adaptedfrom2.13)

a.

$010110_2 \rightarrow 0001\ 0110_2$

b.

$1101_2 \rightarrow 1111\ 1101_2$

c.

$1111111000_2 \rightarrow 1111\ 1000_2$

d.

$01_2 \rightarrow 0000\ 0001_2$

6 (Adaptedfrom2.17)

a.

$01 + 1011 = 0001 + 1011 = 1100$

b.

$11 + 01010101 = 1111\ 1111 + 0101\ 0101 = 0101\ 0100$

c.

$0101 + 110 = 0101 + 1110 = 0011$

d.

$01 + 10 = 11$

7

a.(2.21)

If and only if one of highest and 2nd highest bit's carry occurs, then a overflow occurs.

b.(2.22)

In the equation$\underline{1000\ 0000\ 0000\ 0000 + 1000\ 0000\ 0000\ 0000 = 0000\ 0000\ 0000\ 0000}$, overflow occurs.

c.(2.25)

Because the negative 2's complement number(n bits) ranges $from\ -2^{n-1}\ to\ -1$, while the positive one ranges $from$ 1 to $2^{n-1}-1$, which indicates that the sum is always in the range of $[-2^{n-1}, 2^{n-1}-1]$; so the sum can always expressed as a n bits 2's complement number.

d.

Just as described in a.(2.21), if the situation of carry's occuring is the different at highest and 2nd highest bit, then overflow occurs; otherwise, the adding is normal.

8 (2.34)

a.

$NOT(1011)\ OR\ NOT(1100) = 0100\ OR\ 0011 = 0111$

b.

$NOT(1000\ AND\ (1100\ OR\ 0101)) = NOT(1000\ AND\ 1101) = NOT(1000) = 0111$

c.

$NOT(NOT(1101)) = 1101$

d.

$(0110\ OR\ 0000)\ AND\ 1111 = 0110\ AND\ 1111 = 0110$

9 (2.50)

a.

$$\begin{aligned}\text{x5478}\ AND\ \text{xFDEA} &= 0101\ 0100\ 0111\ 1000\ AND\ 1111\ 1101\ 1110\ 1010 \\ &= 0101\ 0100\ 0110\ 1000 = \frac{\text{xA468}}{x5468}\end{aligned}$$

b.

$$\begin{aligned}\text{xABCD}\ OR\ \text{x1234} &= 1010\ 1011\ 1100\ 1101\ OR\ 0001\ 0010\ 0011\ 0100 \\ &= 1011\ 1011\ 1111\ 1101 = \text{xBBFD}\end{aligned}$$

c.

$$NOT(NOT(\text{xDEFA}))\ AND\ \text{NOT(xFFFF)} = NOT\ \text{x2106}\ AND\ \text{x0000} = \text{x1111}$$

$$NOT(NOT(\text{xDEFA}))\ AND\ \text{NOT(xFFFF)} = \text{xDEFA}\ AND\ \text{x0000} = \text{x0000}$$

d.

$$\begin{aligned}\text{x00FF}\ XOR\ \text{x325C} &= 0000\ 0000\ 1111\ 1111\ XOR\ 0011\ 0010\ 0101\ 1100 \\ &= 0011\ 0010\ 1010\ 0011 = \frac{\text{x3293}}{x32A3}\end{aligned}$$

10 (2.55)

a.

The maximum unsigned decimal value is $4^3 - 1 = 64 - 1 = \underline{63}$

b.

The maximum unsigned decimal value is $\underline{4^n - 1}$

c.

$023 + 221 = \underline{310}$

d.

Since $42 = 2 \times 4^2 + 2 \times 4^1 + 2 \times 4^0$, the answer is $\underline{222}$

e.

$123.3_4 = 01\ 10\ 11.11_2 = \underline{11011.11_2}$

f.

$123.3_4 = 01\ 10\ 11.11_2 = 011011.11_2$. And in IEEE floating point format, it will be $\underline{0\ 10000011\ 10111100000000000000000}$

g.4^{4^m}

There is 4^m kinds of inputs and each input may result 4^1 kinds of outputs, which means $4^m \times 4^1 = \underline{4^{m+1}}$kinds of unique functions this black box can implement.