ECE 270 (Spring 2022)

Homework 2 Solutions

Due on 01/28/2022 (Friday) by 11:59 pm sharp on BrightSpace.

1. (a) Binary: 011 101 010 001 Octal: 3 5 2 1

   (b) Binary: 0111 0101 0001 Hex: 7 5 1

2. Base-3: $(2\ 11\ 10\ 12\ 22\ 21\ 11\ 22)_3$

   Base-9: 2 4 3 5 8 7 4 8

3. $(78)_9 = 63 + 8 = (71)_{10}$

   $(135)_{12} = 144 + 36 + 5 = (185)_{10}$

   $(185)_{10} - (71)_{10} = (114)_{10} = (162)_8$

   So, X = 162

4. Firstly, we are looking for a base that is larger than 5 based on the digits in the equation. The equation found by the explorer's is $5x^2 – 50x + 125 = 0$. Term this as Equation – 1.

Given x = 5, x = 8 are solutions to it. The equation formed by these values of x is (x-5) (x-8) = 0 which is equal to x^2 -13x + 40 = 0. Term this as Equation – 2.

Logically, both equations should be the same, but in different bases, one in base 10, another is in martian's base. So, to equate, try to multiply Equation 2 with '5' as 5 is same in base 10 and martian's base which is higher than 5. So, doing this on both sides, we get $5x^2 – 65x + 200 = 0$. Term this as Equation – 3.

Since Equation 1 is in martian language which has let's say base 'k' which is a non-decimal base, we have to convert it into base 'k' to get Equation 3. As per law, we see that in both equations 1,3 -50 in base 'k' of equation 1 is equal to 65 in base 10 of equation 3. Similarly, 125 in base 'k' is equal to 200 in base 10.

We need to find the value of 'k' here. So, we write $5k^1 + 0k^0 = 65$ (or) $k^2 + 2k + 5 = 200$. Solving any of them, we get k = 13.

5. 233 in binary is 11101001. MSB = 1 is for the negative sign of the number.

1011101001 – signed -233

127 in binary is 1111111. Prepend with 0 and MSB = 0 is for the positive sign of the number.

0001111111 – signed 127

6. a) addition: 1010010 , subtraction: 11100

   b) addition: 1010101 , subtraction: 111

a)

$$\begin{array}{r} 1\,1\,1\,1\,1 \\ 110111 \\ +\quad 11011 \\ \hline 1110010 \\ \hline \end{array} \qquad \begin{array}{r} 110111 \\ -\quad 11011 \\ \hline 11100 \\ \hline \end{array}$$

b)

$$\begin{array}{r} 1\,1\,1 \\ 101110 \\ +\quad 100111 \\ \hline 1010101 \\ \hline \end{array} \qquad \begin{array}{r} 101110 \\ -\quad 100111 \\ \hline 000111 \\ \hline \end{array}$$

7. decimal – 103165

   binary – 11001001011111101

   hexadecimal - 192FD

   octal – 311375

$(606526)_7$

$= 6\times7^5 + 6\times7^3 + 5\times7^2 + 2\times7^1 + 6\times7^0$

$= (103165)_{10} \longrightarrow$ decimal

For binary:

2	103165	
2	51582	1
2	25791	0
2	12895	1
2	6447	1
2	3223	1
2	1611	1
2	805	1
2	402	1
2	201	0
2	100	1
2	50	0
2	25	0
2	12	1
2	6	0
2	3	0
	1 -	1

$= (11001001011111101)_2 \rightarrow$ binary

Group bits 4 by 4 in binary representation for hexadecimal:

$= (1\ \underline{1001}\ \underline{0010}\ \underline{1111}\ \underline{1101})_2 \rightarrow$ binary

$=\ 1\quad 9\quad 2\quad F\quad D \rightarrow$ hexadecimal

Group bits 3 by 3 in binary representation for octal:

$= (\underline{11}\ \underline{001}\ \underline{001}\ \underline{011}\ \underline{111}\ \underline{101})_2 \rightarrow$ binary

$=\ 3\quad 1\quad 1\quad 3\quad 7\quad 5 \rightarrow$ octal

8. Two's complement: -32 to 31

   Signed no system: -31 to 31

9. $(752)_8 \rightarrow 111101010$

   $(424)_8 \rightarrow 100010100$

   Perform binary subtraction to get 011010110 $\rightarrow$ **$(326)_8$**,

   Similarly, **$(F32B)_{16} + (2AE6)_{16} = (11E11)_{16}$**

10. 10. Consider an n bit 2's complement no:

    $A = B_{n-1}B_{n-2}\ldots B_1B_0 \rightarrow$ (n-1)th bit represents the sign.

    If A is positive $(A)_{10} = B_{n-2}.2^{n-2} + B_{n-3}.2^{n-1} + \ldots.B_0.2^0$

    On left-shiting the bits, ignoring the carryout and having 0 at the LSB

    $B = B_{n-2}\ldots B_1B_00$ (Note A and B have same no. of bits)

    $(B)_{10} = B_{n-2}.2^{n-1} + B_{n-3}.2^{n-2} + \ldots.B_0.2^1 + 0 = 2.(B_{n-2}.2^{n-2} + B_{n-3}.2^{n-1} + \ldots.B_0.2^0) = 2(A)_{10}$

    If A is negative and $B_{n-2} = 1$, $(A)_{10} = -B_{n-1}.2^{n-1} + B_{n-2}.2^{n-2} + \ldots.B_0.2^0$

    $= -1.2^{n-1} + 1.2^{n-2} + \ldots.B_0.2^0$

    $= 2^{n-2}(-2+1) + B_{n-3}.2^{n-3} + \ldots.B_0.2^0$

$$= -2^{n-2} + B_{n-3}.2^{n-3} + \ \ldots.B_0.2^0$$

On left-shiting the bits, ignoring the carryout and having 0 at the LSB

$B = B_{n-2}\ldots B_1B_00$ $(B_{n-2} = 1)$

$(B)_{10} = -1.2^{n-1} + B_{n-3}.2^{n-2} + \ldots.B_0.2^1 + 0 = 2.(-2^{n-2} + B_{n-3}.2^{n-3} + \ldots.B_0.2^0) = 2(A)_{10}$

In the other case where B_{n-2} is 0, overflow occurs.

Overflow rule: If two numbers with the same sign (both positive or both negative) are added, then overflow occurs if and only if the result has the opposite sign. In other words, if left shifting changes the value of MSB, we know that overflow has occurred and the resultant no is outside of the range of nos that can be represented by the -bit 2's complementary number.

Ex: If $(6)_{10}$ = 0110, on left shiting it becomes 1100, which is $(-4)_{10}$. Hence overflow has occurred.

If $(6)_{10}$ = 00110, on left shifting it becomes 01100, which is $(12)_{10}$ ie twice of 6.