Course Name: _________________

Course Number and Section: **14:332:xxx:xx**

Experiment: [Experiment 1– Introduction, GitHub tutorial, Number representation]
Lab Instructor: Jalal Abdulbaqi

Date Performed: October 3rd

Date Submitted: October 3rd

Submitted by: [Roshni Shah, 172005723]

Course Name: ________________

Course Number and Section: **14:332:xxx:xx**

! Important: Please include this page in your report if the submission is a paper submission. For electronic submission (email or Sakai) please omit this page.

-------------------------For Lab Instructor Use ONLY-------------------------

GRADE: ______________

COMMENTS:

Electrical and Computer Engineering Department
School of Engineering

```
Lab 1:
        Number Representation:
         1.1)
            a)
        A = 10 D = 13
        B = 11  E = 14
        C = 12 F = 15

               0b10001110
               2^7 +0+0+0+2^3+2^2+2^0+0= 128+8+4+2= 142 (Binary to Decimal)

               0b10001110
               8+14 = 8E (Binary to Hexadecimal)


               0xC3BA
               1100 0011 1011 1010 (Hexadecimal to Binary)

               0xC3BA
                 Placeholders: C= 16^3 3=16^2 B=16^1 A=16^0
                 Values: C=12 3=3 B=11 A=10
                 (10*16^0) + (11*16^1) + (3*16^2) + (12*16^3) = 50,106 (Hexadecimal to
        Decimal)


                81
                 111 001 (Decimal to Binary)

                81
                 81/16=5 remainder 1
                 5/16=0 remainder 5
                  =51 (Decimal to Hexadecimal)


                0b100100100
                   2^8+0+0+2^5+0+0+2^2+0+0= 256+32+4 = 292 (Binary to Decimal)

                0b100100100
                   1 0010 0100 (Binary to Hexadecimal)  = 124 ?????????????


                 0xBCA1
```

```
   1011 1100 1010 0001 (Hexadecimal to Binary)

OxBCA1
   Placeholders: B=16^3  C=16^2 A=16^1  1=16^0
   Values: B=11 C=12 A=10 1=1
   (1*16^0)+(10*16^1)+(12*16^2)+(11*16^3)= 1+160+3072+45056= 48289
(Hexadecimal to Decimal)

0
 0 (Decimal to Binary)
0
 0000 (Decimal to Hexadecimal)

42
  101010 (Decimal to Binary)

42
   42/16 =2 remainder 10
   2/16 =0 remainder 2
   Read from bottom to top
   2 --> 10
   = 2A  (Decimal to Hexadecimal)

0xBAC4
     1011 1010 1100 0100 (Hexadecimal to Binary)

0xBAC4
    Placeholders: B=16^3 A=16^2 C=16^1 4=16^0
    Values: B=11 A=10 C=12 4=4
    (4*16^0)+(12*16^1)+(10*16^2)+(11*16^3)= 47809
```

b)

```
2^14= 2^10 * 2^4= Ki 2^4

2^43= 2^40 * 2^3 = Ti 2^3

2^23= 2^20 * 2^3 = Mi 2^3

2^58= 2^50 * 2^8 = Pi 2^8

2^64= 2^60 * 2^4 = Ei 2^4
2^42= 2^40 * 2^2 = Ti 2^2
```

c)

2Ki= 2^11

512Pi= 2^59

256Ki= 2^18

32Gi= 2^35

64Mi=2^26

8Ei= 2^63

2.2)

Unsigned

Largest = 11111111 = 255 ---> Largest + 1 = 11111111 + 1 = 00000000 = 0

Two’s complement

The largest two’s complement is 10000000 in which the 8th bit=-128 but to get the positive value of this we flip the bits (one's complement) 01111111=127 which is the same as doing the largest+1

Largest: 127 Largest+1: 128

2)

Unsigned

0 = 0000 0000, 3 = 0000 0011 -3 can’t be represented because as an unsigned integer because that ranges from 0 to 255, and there are no negatives

Two’s Complement

0 = 0000 0000, 3 = 0000 0011, -3 = 0000 0011 → flipped + 1 → 1111 1101

3)

Unsigned

42 = 00101010, -42 can’t be represented as an unsigned integer because the range of an unsigned integer is from 0 to 255, there are no negatives

Two’s Complement

42 = 00101010, -42 = 00101010 → flipped + 1 → 11010110

4)

There is no largest integer because an encoding scheme that uses 8 bits could have different ranges of offsets. Such as going to 1 to 256 instead of 0 to 255

5) If there is an x and its complement, x’ and you take the sum of both the result will always be 1111… and this works for any n digit bit for example, if x=1010 and x'=0101then x+x’=1111

6)

Decimal is used mainly for human calculation. Decimals can be used easier for mental math or calculators and since we have 10 fingers we started off with decimals in the beginning.

Binary numbers are used a lot of the time for computers. It is easier to design a circuit with binary signals and its easier for machines to read with than bigger radix signals.

Hexadecimals are good to display binary digits. As one hex corresponds to 4 binary digit then it makes it easier to do computing with bigger numbers.

3.1)

For 0,,e we only need 2 bits as there are 3 stages

1) 1 TiB= 2^40 bits
2) 2 TiB is doubled 1 TiB = 2.199*10^12

To address every byte of memory it has to be 41 bits long to hold that 2TiB of memory

3) 0 bits are needed to represent e as there is only one variable and as it is a single stage it doesn't need any bits

.