Gasser Ahmed
gasser18@vt.edu
ECE 5484, Homework 1

1. Answer the following questions:
    a. Convert 3.7 seconds into milliseconds
       1 second = 10^3 millisecond
       Therefore 3.7 seconds = $3.7x10^3$ milliseconds=3700 milliseconds
    b. How many nanoseconds are in 3 microseconds?
       1 second = 10^9 nanosecond, 1 second = 10^6 microsecond
       Therefore 10^9 nanosecond = 10^6 microseconds
       Therefore 10^3 nanosecond = 1 microseconds
       Therefore 3 microseconds = $3x10^3$ nanoseconds = 3000 nanoseconds
    c. How many megabytes (MB) are in 1.3 gigabytes (GB)?
       1 MB = 10^6 B, 1 GB = 10^9 B
       Therefore 1 GB = 10^3 MB
       Therefore 1.3 GB = 1.3 x 10^3 MB = 1300 MB

---

2. In the von Neumann model, explain the purpose of the:
    a. Processing unit: performs all the arithmetic and logic functions.
    b. Program counter: holds the address of the next instruction to be executed in a program.

---

3. Briefly explain the three main types of Cloud computing platforms.
    a. Infrastructure-as-a-service (IaaS): used for Internet-based access to storage and computing power. It lets you rent IT infrastructure servers and virtual machines, storage, networks, and operating systems from a cloud provider on a pay-as-you-go basis.
    b. Platform-as-a-service (PaaS): gives developers the tools to build and host web applications. PaaS is designed to give users access to the components they require to quickly develop and operate web or mobile applications over the Internet, without worrying about setting up or managing the underlying infrastructure of servers, storage, networks, and databases.
    c. Software-as-a-service (SaaS): a method for delivering software applications over the Internet where cloud providers host and manage the software applications making it easier to have the same application on all of your devices at once by accessing it in the cloud.

---

4. Convert the following decimal values to unsigned binary representation.
    a. 28.3125:

Integer part:

28÷2=14	remainder 0
14÷2=7	remainder 0
7÷2=3	remainder 1
3÷2=1	remainder 1
1÷2=0	remainder 1 stop

Fraction part:

0.3125x2=0.625	integer 0
0.625x2=1.25	integer 1
0.25x2=0.5	integer 0
0.5x2=1.0	integer 1
0.0x2=0.0	integer 0 stop

Therefore $(28.3125)_{10}$= $(11100.0101)_2$

---

b. 127.0

Integer part:

127÷2=63	remainder 1
63÷2=31	remainder 1
31÷2=15	remainder 1
15÷2=7	remainder 1
7÷2=3	remainder 1
3÷2=1	remainder 1
1÷2=0	remainder 1

Therefore (127.0)10= $(1111111)_2$

---

c. 128.0

Integer part:

128÷2=64	remainder 0
64÷2=32	remainder 0
32÷2=16	remainder 0
16÷2=8	remainder 0
8÷2=4	remainder 0
4÷2=2	remainder 0
2÷2=1	remainder 0
1÷2=0	remainder 1

Therefore (128)10= $(10000000)_2$

---

d. 132.5625

Integer part:

132÷2=66	remainder 0
66÷2=33	remainder 0
33÷2=16	remainder 1
16÷2=8	remainder 0
8÷2=4	remainder 0
4÷2=2	remainder 0
2÷2=1	remainder 0
1÷2=0	remainder 1 stop

Fraction part:

0.5625x2=1.125	integer 1
0.125x2=0.25	integer 0
0.25x2=0.5	integer 0
0.5x2=1.0	integer 1
0.0x2=0.0	integer 0 stop

Therefore (132.5625)10= $(10000100.1001)_2$

---

e. 13.2

Integer part:

13÷2=6	remainder 1
6÷2=3	remainder 0
3÷2=1	remainder 1
1÷2=0	remainder 1 stop

Fraction part:

0.2x2=0.4	integer 0
0.4x2=0.8	integer 0
0.8x2=1.6	integer 1
0.6x2=1.2	integer 1
0.2x2=0.4	integer 0
……	

Therefore (13.2)10= $(1101.00110011....)_2$

---

5. Convert the following unsigned binary numbers to decimal representation.
   a. 110.110 $=1x2^2 +1x2^1 +0x2^0 +1x2^{-1} +1x2^{-2}$=4+2+0+0.5+0.25=6.75
   b. 1.101=$1x2^0+1x2^{-1} +0x2^{-2} +1x2^{-3}$=1+0.5+0+0.125=1.625
   c. 10111.0111=$1x2^4+0x2^3+1x2^2+1x2^1+1x2^0+0x2^{-1}+1x2^{-2}+1x2^{-3}+1x2^{-4}$
      =16+0+4+2+1+0+0.25+0.125+0.0625=23.4375
   d. 11111111=$2^7+2^6+2^5+2^4+2^3 +2^2+2^1+2^0$=128+64+32+16+8+4+2+1=255
   e. 1110.01=$1x2^3+1x2^2+1x2^1+0x2^0+0x2^{-1} +1x2^{-2}$=8+4+2+0+0+0.25=14.25

---

6. Suppose a computer uses radix-3 (ternary) logic instead of radix-2 (binary) logic to represent unsigned integers.
    a. a. What is the radix-3 representation of the decimal value 37?
       37÷3=12 remainder 1
       12÷3=4 remainder 0
       4÷3=1 remainder 1
       1÷3=0 remainder 1 stop
       Therefore $(37)_{10}$ = $(1101)_3$

    b. What is the largest value that can be represented by 6 digits?
       Since ternary takes values 0, 1, and 2
       Therefore, the largest 6-digit number will be $(222222)_3$
       $(222222)_3=2x3^5 +2x3^4 +2x3^3 +2x3^2 +2x3^1 +2x3^0=486+162+54+18+6+2 = (728)_{10}$

    c. Why do you think that binary logic is much more commonly used than ternary logic? Be brief
       I think that’s because all modern computers operate using "switches" that are either in an "on" state or an "off" state depending on whether electricity is running through them and are represented by 1 and 0 respectively in the binary system. That is, electric signals can have only two states: on (1) when the current flows and off (0) when it doesn't, which doesn’t make sense to have ternary logic used for no reason which also adds more expenses to build those ternary logic systems and increases error rates.

---

7. Convert the hexadecimal number FEED A BEE to binary
   F: 1111, E: 1110, E: 1110, D: 1101, A: 1010, B: 1011, E: 1110, E: 1110
   Therefore $(FEEDABEE)_{16}$ = $(11111110111011011010101111101110)_2$

---

8. Represent each of the following decimal numbers in binary using 8-bit signed magnitude, one's complement, two's complement, and excess-127 representations:
   a. 35
      i. 8-bit signed magnitude:
         35÷2=17 remainder 1
         17÷2=8 remainder 1
         8÷2=4 remainder 0
         4÷2=2 remainder 0
         2÷2=1 remainder 0
         1÷2=0 remainder 1 stop
         Therefore $(35)_{10}$ = $(00100011)_2$

      ii. one's complement:
         Since it is a positive number, the result is same as the signed magnitude representation = $(00100011)_2$

      iii. two's complement:
         Since it is a positive number, the result is same as the signed magnitude representation = $(00100011)_2$

      iv. excess-127:
         35+127=162
         162÷2=81 remainder 0
         81÷2=40 remainder 1
         40÷2=20 remainder 0
         20÷2=10 remainder 0
         10÷2=5 remainder 0
         5÷2=2 remainder 1
         2÷2=1 remainder 0
         1÷2=0 remainder 1 stop
         Therefore = $(10100010)_2$

b. -35

i. 8-bit signed magnitude:
Same as previous question (8.a) but replace the first bit to be 1 to represent the -ve sign instead of 0 (+ve), so the answer will be $(10100011)_2$

ii. one's complement: after bit inversion = $(11011100)_2$

iii. two's complement:

bit inversion of 00100011 = 11011100

  11011100
+00000001
=11011101 (no carry)

Therefore = $(11011101)_2$

iv. excess-127:
-35+127=92

92÷2=46	remainder 0
46÷2=23	remainder 0
23÷2=11	remainder 1
11÷2=5	remainder 1
5÷2=2	remainder 1
2÷2=1	remainder 0
1÷2=0	remainder 1 stop

Therefore = $(01011100)_2$

c. 97

i. 8-bit signed magnitude:

97÷2=48	remainder 1
48÷2=24	remainder 0
24÷2=12	remainder 0
12÷2=6	remainder 0
6÷2=3	remainder 0
3÷2=1	remainder 1
1÷2=0	remainder 1 stop

Therefore $(97)_{10}$ = $(01100001)_2$

ii. one's complement:

Since it is a positive number, the result is same as the signed magnitude representation = $(01100001)_2$

iii. two's complement:

Since it is a positive number, the result is same as the signed magnitude representation = $(01100001)_2$

iv. excess-127:

97+127=224

224÷2=112	remainder 0
112÷2=56	remainder 0
56÷2=28	remainder 0
28÷2=14	remainder 0
14÷2=7	remainder 0
7÷2=3	remainder 1
3÷2=1	remainder 1
1÷2=0	remainder 1 stop

Therefore = $(11100000)_2$

d. -97

   i. 8-bit signed magnitude:
      Same as previous question (8.c) but replace the first bit to be 1 to represent the -ve sign instead of 0 (+ve), so the answer will be $(11100001)_2$

   ii. one's complement: after bit inversion = $(10011110)_2$

   iii. two's complement:
      bit inversion of 01100001 = 10011110
        10011110
      +00000001
      =10011111 (no carry)
      Therefore = $(10011111)_2$

   iv. excess-127:
      -97+127=30
      30÷2=15 remainder 0
      15÷2=7 remainder 1
      7÷2=3 remainder 1
      3÷2=1 remainder 1
      1÷2=0 remainder 1 stop
      Therefore = $(00011110)_2$

---

9. If the floating-point number representation on a certain system has a sign bit, a 4-bit exponent and a 5-bit significand:
    a. a. What is the largest positive and the smallest positive number that can be stored on this system if the storage is normalized? (Assume no bits are implied, there is no biasing, exponents use two's complement notation, and exponents of all zeros and all ones are allowed.)
        - Two's complement 4-bit Exponent, no bias: This means the exponent can represent values in the range -8 (corresponding to 1000) to 7 (corresponding to 0111).
        - Normalized without implied bit: This means significand always starts with a 1.

    Exponent:

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1

    Therefore, largest positive will be the one having the largest significand (11111) and exponent values (0111 = 7): $0.11111\text{x}2^{0111}$=$0.11111\text{x}2^{7}$ = $(1111100)_2$

    $=1\text{x}2^6+ 1\text{x}2^5 +1\text{x}2^4 +1\text{x}2^3 +1\text{x}2^2 +0\text{x}2^1 + 0\text{x}2^0 = (124)_{10}$

    And smallest positive will have the smallest possible exponent (1000 = -8) and keep the first bit of the significand to satisfy the normalized requirement (10000): $0.10000\text{x}2^{1000}$ = $0.10000\text{x}2^{-8}$=$(0.000000001)_2$=$0\text{x}2^{-1}+ 0\text{x}2^{-2} +0\text{x}2^{-3} +0\text{x}2^{-4} +0\text{x}2^{-5}+0\text{x}2^{-6} + 0\text{x}2^{-7}+0\text{x}2^{-8} + 1\text{x}2^{-9}$ = $(0.001953125)_{10}$

    b. What bias should be used in the exponent if we prefer all exponents to be non-negative? Why would you choose this bias?
    bias= $2^{(\text{exponent-1})}$-1=$2^{(4-1)}$-1=2^3-1=8-1=7. I would choose that bias because it is more efficient to use a biased exponent, because we can use simpler integer circuits designed specifically for unsigned numbers when comparing the values of two floating-point numbers. So, this would help us in representing negative exponents.

10. Answer the following questions about character encoding.

a. The ASCII code for the letter E is 1000101, and the ASCII code for the letter e is 1100101. Given that the ASCII code for the letter M is 1001101, without looking at Table 2.7, what is the ASCII code for the letter m?

$E=1000101=1x2^6 + 1x2^2 +1x2^0=64+4+1=69$
$e=1100101=1x2^6 + 1x2^5+ 1x2^2+ 1x2^0 =64+32+4+1=101$
$M=1001101=1x2^6+ 1x2^3 + 1x2^2+ 1x2^0=64+8+4+1=77$
Since e-E=101-69=32
Therefore m-M=32
and m=32+77=109

109÷2=54	remainder 1
54÷2=27	remainder 0
27÷2=13	remainder 1
13÷2=6	remainder 1
6÷2=3	remainder 0
3÷2=1	remainder 1
1÷2=0	remainder 1 stop

Therefore m = $(1101101)_2$

b. The EBCDIC code for the letter E is 11000101, and the EBCDIC code for the letter e is 1000 0101. Given that the EBCDIC code for the letter M is 1101 0100, without looking at Table 2.7, what is the EBCDIC code for the letter m?

$E=11000101=1x2^7 + 1x2^6+ 1x2^2+ 1x2^0=197$
$e=10000101=1x2^7 + 1x2^2+ 1x2^0=133$
$M=11010100=1x2^7 + 1x2^6+ 1x2^4+ 1x2^2=212$
Since E-e=64
Therefore M-m=64
and m=212-64=148

148÷2=74	remainder 0
74÷2=37	remainder 0
37÷2=18	remainder 1
18÷2=9	remainder 0
9÷2=4	remainder 1
4÷2=2	remainder 0
2÷2=1	remainder 0
1÷2=0	remainder 1 stop

Therefore m = $(10010100)_2$