**CMPT 295**

**Assignment 1: BITS**

**Submitted:** (1/29/2016)

1.Matin, Mehrshad, 301117102, mmatin@sfu.ca

**TABLE OF CONTENTS**

1. Documentation File Index: listing all supporting documentation files, including:
   1. Work and answers to any questions asked in the assignment;
   2. Any supporting word or other text files.

1. Schematic File Index: listing any digital circuit design screen shots if required
2. Source File Index: listing any C and x86-64 assembly source files submitted
3. For each of the following encoding schemes, what value is represented by the codeword 1011001001?
4. hexadecimal

(1\*2^0) + (1\*2^3)+ (1\*2^6)+ (1\*2^7)+(1\*2^8)

1+8+64+128+256 = 457

713/16 = 44(9)

44/16 = 2(12)

2/16 = 0(2)

1,12,9 🡪 2C9

1. natural binary

(2^0)+(2^3)+(2^6)+(2^7)+(2^9) = 713

1. sign-magnitude

1011001001

2^0 + 2^3 + 2^6 + 2^7

-201

1. 2's complement

-311

Comp of codeword = 0100110110

Add 1 = 0100110110 + 0000000001 = 0100110111

2^0 + 2^1 + 2^2 + 2^4 + 2^5 + 2^8 = -311

1. Bias.

2^9 -1 = 512 -1 = 511

1011001001 = 713

713 – 511 = 202

1. floating point if the fraction field is 3 bits.

1 **|** 011001 **|** 001

Exp=25

Bias = 2^(5) -1 = 31

25 = 31+p

P= -6

1.001 \* 2^(-6)

0.00001001

2^(-6) + 2^(-9)

0.015625 + 0.00390625 = -0.017578125

1. floating point if the fraction field is 4 bits

1 **|** 01100 **|** 1001

Exp=12

Bias = 2^(5-1) -1 = 15

12 = 15+p

P= -3

1.1001 \* 2^(-3)

0.011001

0.125+0.0625+0.0078125 = -0.1953125

2) For a floating point encoding using 6-bit codewords and a fraction field of 2 bits, list all the codewords that represent positive integers and their corresponding integer values.

1)001100 (exp =3, 3=3+p, p=0, 1.00\*(2^0) = 1, 2^0 = 1)

2)010100 (exp=5, 5=3+p, p=2, 1.00\*(2^2) = 100, 2^2=4)

3)011000 (exp=6, 6=3+p, p=3, 1.00\*(2^3) = 1000, 2^3 = 8)

4)010000 (exp= 4, 4=3+p, p=1, 1.00\*(2^1) = 10, 2^1=2)

5)010101 (exp =5, 5=3+p, p=2, 1.01\*(2^2)= 101, (2^0)+(2^2) = 5)

6)011001 (exp=6, 6=3+p, p=3, 1.01\*(2^3) = 1010, (2^1)+(2^3)= 10)

7)010110 (exp = 5, 5=3+p, p=2, 1.10\*(2^2)=110, 2^1 + 2^2 = 6)

8)011010 (exp = 6, 6=3+p, p=3, 1.10\*(2^3)=1100, 2^2 + 2^3=12)

9)010010 (exp = 4, 4=3+p, p=1, 1.10\*(2^1)=11.0, 2^0 + 2^1= 3)

10)010111 (exp = 5, 5=3+p, p=2, 1.11\*(2^2) = 111, 2^0 + 2^1+ 2^2 = 7)

11)011011 (exp = 6, 6=3+p, p=3, 1.11\*(2^3) = 1110, 2^1+2^2+2^3 = 14)

3)Decode the following codewords assuming a oating point encoding was used with a fraction field of 4-bits:

|  |  |
| --- | --- |
| 010110011 | 19 |
| 111110000 | -infinity |
| 100111010 | -0.1015625 |
| 000000111 | 6.835937 \* 10 ^-3 |
| 111100000 | -128 |
| 010111100 | 28 |
|  |  |

a)

0 **|** 1011 **|** 0011

Exp=11

Bias= 2^3 -1 = 7

11=7+p

P=4

1.0011 \* 2^4 = 10011 🡪 19

b)

1 **|** 1111 **|** 0000

-infinity

c)

1 **|** 0011 **|** 1010

Exp=3

Bias= 2^3 -1 = 7

3=7+P

P= -4

1.1010\*2^-4 = 0.00011010 🡪 2^-4 + 2^-5 + 2^ -7 = - 0.1015625

d)

0 | 0000 | 0111

Exp=0

Bias= 2^2 -1= 3

1-bias

1-7 = -6

0.0111\* 2^-6 = 6.835937 \* 10 ^-3

e)

1| 1110 | 0000

Exp=14

Bias= 2^3 -1 = 7

14= 7+p

P=7

1.0000\*2^7 = 10000000 🡪 2^7 = -128

f)

0 **|** 1011 **|** 1100

Exp= 11

Bias= 2^3 -1 = 7

11 = 7+p

P= 4

1.1100 \* 2^4 = 11100 🡪 2^2 + 2^3 + 2^4 = 4+8+16 = 28

4)

Operand 1 Operand2 -Product-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1.25 | 0x3f40 | 3.75 | 0x40e0 | 0x0000412c |
| 0 | 0x0000 | 3.75 | 0x40e0 | 0x00000000 |
| 0 | 0x0000 | 0 | 0x0000 | 0x00000000 |
| 2^7 | 0x4600 | 1 | 0x3f00 | 0x00004600 |
| 2^7 | 0x4600 | 2^-7 | 0x3800 | 0x00007f00 |
| 2^63 | 0x7e00 | 2 | 0x4000 | 0x00007f00 |
| 2^32 | 0x6000 | 2^31 | 0x5f00 | 0x00007f00 |
| 2^-62 | 0x0100 | 2^-62 | 0x0100 | 0x00007f00 |

To run the c files, please follow:

gcc –c main.c

gcc –c fplib.c

gcc main.o fplib.o –o runmain

./runmain