## **Department of Electrical and Computer Engineering**

### **The University of Texas at Austin**

Name: Xinyuan Pan (xp572)

EE 460N, Spring 2017

Problem Set 5

Due: April 17, before class

Yale N. Patt, Instructor

Chirag Sakhuja, Sarbartha Banerjee, Jon Dahm, Arjun Teh, TAs

## 

## **Instructions**

You are encouraged to work on the problem set in groups and turn in one problem set for the entire group. The problem sets are to be submitted on Canvas. Only one student should submit the problem set on behalf of the group. The only acceptable file format is PDF. Include the name of all students in the group in the file.

*You will need to refer to the assembly language handouts and the LC-3b ISA on the course website.*

**Problem 1**

Determine the decimal value of the following IEEE floating point numbers.

* 1. 1 10000000 10100000000000000000000
  2. 0 00000000 01010000000000000000000
  3. 1 11111111 00000000000000000000000

1. -3.25
2. 1.25 x 2^-128
3. negative infinity

**Problem 2**

Using a residue number system with two moduli, represent all of the decimal values between 0 and 11 inclusive when the moduli are

1. 4 and 3
2. 6 and 2

|  |  |  |
| --- | --- | --- |
|  | 4 | 3 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 0 |
| 4 | 0 | 1 |
| 5 | 1 | 2 |
| 6 | 2 | 0 |
| 7 | 3 | 1 |
| 8 | 0 | 2 |
| 9 | 1 | 0 |
| 10 | 2 | 1 |
| 11 | 3 | 2 |

|  |  |  |
| --- | --- | --- |
|  | 6 | 2 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 2 | 2 | 0 |
| 3 | 3 | 1 |
| 4 | 4 | 0 |
| 5 | 5 | 1 |
| 6 | 0 | 0 |
| 7 | 1 | 1 |
| 8 | 2 | 0 |
| 9 | 3 | 1 |
| 10 | 4 | 0 |
| 11 | 5 | 1 |

**Problem 3**

Using the Booth Multiplication Algorithm, multiply the two unsigned 10-bit numbers 0011011110 and 0001110010. Show the intermediate results after each step.

Multiplicand: 0011011110

Multiplier: 0001110010 negative: 1110001110

Step Product

1. 0000000000 0011011110 0
2. 0000000000 0001101111 0
3. 1111000111 0000110111 1
4. 1111100011 1000011011 1
5. 1111110001 1100001101 1
6. 1111111000 1110000110 1
7. 0000110101 0111000011 0
8. 1111100001 1011100001 1
9. 1111110000 1101110000 1
10. 0000110001 0110111000 0
11. 0000011000 1011011100 0

product is 00000110001011011100

**Problem 4**

From Tanenbaum, 4th edition, Appendix B, 4.

The following binary floating-point number consists of a sign bit, an excess 63, radix 2 exponent, and a 16-bit fraction. Express the value of this number as a decimal number.

**0 0111111 0000001111111111**

b1.0000001111111111 = 1.0156

**Problem 5**

From Tanenbaum, 4th edition, Appendix B, 5.

To add two floating point numbers, you must adjust the exponents (by shifting the fraction) to make them the same. Then you can add the fractions and normalize the result, if need be. Add the single precision IEEE floating-point numbers 3EE00000H and 3D800000H and express the normalized result in hexadecimal. ['H' is a notation indicating these numbers are in hexadecimal]

x3EE00000 = b0 01111101 11… = 1.11 x 2^-2

x3D800000 = b0 01111011 00… = 1.00 x 2^-4 = 0.01 x 2^-2

sum = 10.00 x 2^-2 = 1.0 x 2^-1 = b0 01111110 00… = x3F000000

**Problem 6**

From Tanenbaum, 4th edition, Appendix B, 6.

The Tightwad Computer Company has decided to come out with a machine having 16-bit floating-point numbers. The model 0.001 has a floating-point format with a sign bit, 7-bit, excess 63 exponent and 8-bit fraction. Model 0.002 has a sign bit, 5-bit, excess 15 exponent and a 10-bit fraction. Both use radix 2 exponentiation. What are the smallest and largest positive normalized numbers on both models? About how many decimal digits of precision does each have? Would you buy either one?

0.001: largest: 1.11111111 x 2^63 smallest: 1.0 x 2^-62

precision: (8+1)/10 \* 3 = 2.7 decimal digits of precision

0.002: largest: 1.1111111111 x 2^15 smallest: 1.0 x 2^-14

precision: (10+1)/10 \* 3 = 3.3 decimal digits of precision

model 0.001 for wider range, model 0.002 for higher precision.

**Problem 7**

The following numbers are represented exactly with a 9-bit floating point representation, in the format of the IEEE Floating Point standard:

-infinity, -1, 0, 5/16, 19.5, 48.

1. How many bits are needed for the fraction?

5

1. What is the bias?

1

1. Write each number in in the 9-bit floating point representation, below:

|  |  |
| --- | --- |
| Value | Representation |
| 48 | 0 110 00000 |
| 19.5 | 0 101 00111 |
| 5/16 | 0 000 01010 |
| 0 | 0 000 00000 |
| -1 | 1 001 00000 |
| -infinity | 1 111 00000 |