CMPT 295

Name: Junchen Li

Student ID: 301385486

Date: 2020/1/29

Assignment 2

Objectives:

In this assignment, you will gain familiarity with:

- IEEE floating point representation

Submission:

- Submit your document called **Assignment_2.pdf**, which must include your answers to all of the questions in Assignment 2.
    - Add your full name and student number at the top of the first page of your document **Assignment_2.pdf**.
- When creating your assignment, first include the question itself and its number then include your answer, keeping the questions in its original numerical order.
- **If you hand-write your answers (as opposed to using a computer application to write them):** When putting your assignment together, do not take photos (no .jpg) of your assignment sheets! Scan them instead! Better quality -> easier to read -> easier to mark!
- Submit your assignment electronically on CourSys

Due:

- Thursday, Jan. 30 at 3pm
- Late assignments will receive a grade of 0, but they will be marked in order to provide the student with feedback.

Requirements:

- **Show your work** (as illustrated in lectures).

Marking scheme:

This assignment will be marked as follows:

- Questions 1 and 2 will be marked for correctness.

The amount of marks for each question is indicated as part of the question.

A solution will be posted after the due date.

---

1. [8 marks] Floating point conversion and Rounding.
   a. Represent the following numbers in IEEE floating point representation (single precision), clearly showing the effect of rounding on the "frac" (mantissa), if rounding occurs, and express your final answer in binary and in hexadecimal:
      I. 0.001111111_2 **1.111111_2 E=-3 exp=2^{8-1}-1-3=127-3=124_{10}&it is a positive number. 124_{10}->01111100_2 The final answer is: 0 01111100 111111.**
      II. 3.1416015625_{10} **11.0010010001_2 E=1 1.10010010001 exp:1+127=128_{10}->10000000_2 And it is a positive number. The final answer is 0 10000000 10010010001**
      III. **-**0.9_{10} **-0.1110011001100$^{23}{}_2$… E=-1 rounding occurs (1 01111110 110011001100 …) (0.111001*1001*$_2$(001<1/2-down) -0.11100…1101$^{23}{}_2$ The final answer is ≈-0.11100110011001100110$_2$ and it round to -0.8999999762 the hexadecimal is 0XBF666666.**
      IV. $1/3_{10}$ (a third) **0.0101010101… E=-2 exp:127-2=125_{10}->01111101_2 (0 01111101 01010101010101010101011) rounding occurs 0.010(101>1/2=up) 0.011_2 The final answer is ≈0. 01010101010101010101011$_2$ and it rounding to 0.3333333433 the hexadecimal is 0X3EAAAAAB.**
   b. Convert 0x4AEA4C1A from IEEE floating point representation (single precision) to a real number.

0100 1010 1110 1010 0100 1100 0001 1010_2-> 0 10010101 11010100100110000011010 149-127=22 move to right 22 bits' location 1.11010100100110000011010 11101010010011000001101.0 The final answer is 7677453.

c. Round the following binary numbers (rounding position is bolded - 2^{-4} position) following the rounding rules of the IEEE floating point representation.

I. $1.0011\mathbf{1}111_2$ **111>1/2—Rounding up 1.0100 1.2421875-1.25=-0.0078125**

II. $1.1001\mathbf{0}01_2$ **001<1/2—Rounding down 1.1001 1.5703125-1.5626=0.0077125**

III. $1.0111\mathbf{1}00_2$ **100=1/2— half way—1.0111+0.1=1.1000 even (up to even) 1.1000 1.46875-1.5=-0.03125**

IV. $1.0110\mathbf{0}100_2$ **100=1/2— half way—1.0110+0.1=1.0111 odd (down to even) 1.0110 1.40625-1.375=0.03125**

For each of the above rounded binary numbers, indicate what type of rounding you performed and compute the value that is either added to or subtracted from the original number (listed above) as a result of the rounding process. In other words, compute the error introduced by the rounding process.

2. [12 marks] Creating hypothetical smaller floating-point representations based on the IEEE floating point format allows us to investigate this encoding scheme more easily, since the numbers are easier to manipulate and compute.

Below is a table listing several real numbers represented as 6-bit floating-point numbers (w = 6). The format of these 6-bit floating-point numbers is as follows: 1 bit is used to express for the sign, 3 bits are used to express "exp" (k = 3) and 2 bits are used to represent "frac" (n = 2), in the following order: sign exp frac.

Complete the table (the same way as in Figure 2.35 in our textbook) then answer the questions below the table.

Tip: Have a look at Figure 2.35 in our textbook, which illustrates a similar table for a hypothetical 8-bit floating-point format. This will give you an idea of how to complete the table. Also, Figure 2.34 displays the complete range of these 6-bit floating point numbers as well as their values between -1.0 and 1.0. This diagram may be helpful when you are checking your work.

		Exponent			**Fraction**		**Value**		
Description	**Bit representation**	**exp**	**E**	**2^E**	**frac**	**M**	**M 2^E**	**V（分数）**	**Decimal**
zero	0 000 00	**0**	**1-3=-2**	**1/4**	**0/4**	**0/4=0**	**0**	**0**	**0.0**
Smallest positive denormalized	0 000 01	**0**	**-2**	**1/4**	**2^{-2}=1/4**	**1/4**	**1/16**	**$(-1)^0$*1/16 =1/16**	**0.0625**
	0 000 10	**0**	**-2**	**1/4**	**2^{-1}=1/2**	**1/2**	**1/8**	**$(-1)^0$*1/8 =1/8**	**0.125**
Largest positive denormalized	0 000 11	**0**	**-2**	**1/4**	**$2^{-1}+2^{-2}$=3/4**	**3/4**	**3/16**	**$(-1)^0$*3/16 =3/16**	**0.1875**
Smallest positive normalized	0 001 00	**1**	**-2**	**1/4**	**0/4**	**1+0/4=4/4=1**	**1/4**	**$(-1)^0$*1/4 =1/4**	**0.25**
	0 001 01	**1**	**-2**	**1/4**	**1/4**	**5/4**	**5/16**	**$(-1)^0$*5/16 =5/16**	**0.3125**
	0 001 10	**1**	**-2**	**1/4**	**1/2**	**3/2**	**3/8**	**$(-1)^0$*3/8 =3/8**	**0.375**
	0 001 11	**1**	**-2**	**1/4**	**3/4**	**7/4**	**7/16**	**$(-1)^0$*7/16 =7/16**	**0.4375**

	0 010 00	**2**	**-1**	**1/2**	**0**	**1+0=1**	**1/2**	**$(-1)^0$*1/2 =1/2**	**0.5**
	0 010 01	**2**	**-1**	**1/2**	**1/4**	**5/4**	**5/8**	**$(-1)^0$*5/8 =5/8**	**0.625**
	0 010 10	**2**	**-1**	**1/2**	**1/2**	**3/2**	**3/4**	**$(-1)^0$*3/4 =3/4**	**0.75**
	0 010 11	**2**	**-1**	**1/2**	**3/4**	**7/4**	**7/8**	**$(-1)^0$*7/8 =7/8**	**0.875**
One	0 011 00	**3**	**0**	**1**	**0**	**1**	**1*1=1**	**$(-1)^0$*1 =1**	**1.0**
	0 011 01	**3**	**0**	**1**	**1/4**	**5/4**	**5/4**	**$(-1)^0$*5/4 =5/4**	**1.25**
	0 011 10	**3**	**0**	**1**	**1/2**	**3/2**	**3/2**	**$(-1)^0$*3/2 =3/2**	**1.5**
	0 011 11	**3**	**0**	**1**	**3/4**	**7/4**	**7/4**	**$(-1)^0$*7/4 =7/4**	**1.75**
	0 100 00	**4**	**1**	**2**	**0**	**1**	**2**	**$(-1)^0$*2 =2**	**2**
	0 100 01	**4**	**1**	**2**	**1/4**	**5/4**	**5/2**	**$(-1)^0$*5/2 =5/2**	**2.5**
	0 100 10	**4**	**1**	**2**	**1/2**	**3/2**	**3**	**$(-1)^0$*3 =3**	**3**
	0 100 11	**4**	**1**	**2**	**3/4**	**7/4**	**7/2**	**$(-1)^0$*7/2**	**3.5**

								=3	
	0 101 00	**5**	**2**	**4**	**0**	**1**	**4**	**$(-1)^0$*4=4**	**4**
	0 101 01	**5**	**2**	**4**	**1/4**	**5/4**	**5**	**$(-1)^0$*5**	**5**
	0 101 10	**5**	**2**	**4**	**1/2**	**3/2**	**6**	**$(-1)^0$*6**	**6**
	0 101 11	**5**	**2**	**4**	**3/4**	**7/4**	**7**	**$(-1)^0$*7**	**7**
	0 110 00	**6**	**3**	**8**	**0**	**1**	**8**	**$(-1)^0$*8=8**	**8**
	0 110 01	**6**	**3**	**8**	**1/4**	**5/4**	**10**	**$(-1)^0$*10**	**10**
	0 110 10	**6**	**3**	**8**	**1/2**	**3/2**	**12**	**$(-1)^0$*12**	**12**
Largest positive normalized	0 110 11	**6**	**3**	**8**	**3/4**	**7/4**	**14**	**$(-1)^0$*14**	**14**
+ Infinity	**0 111 00**	–	–	–	–	–	–	∞	–
NaN	**0 111 10** **(frac part cannot be all zero)**	–	–	–	–	–	–	NaN	–

a. What is the value of the bias? **Answer: #exp=3, #frac=2 Bias= 2^{3-1}-1=3**

b. Consider two adjacent denormalized numbers. How far apart are they? Expressed this difference ("delta") as a decimal number.

Answer:

0 001 10 0.375_{10}

0 001 11 0.4375_{10}

difference=0.4375-0.375= 0.0625

c. Consider two adjacent normalized numbers …

   a. with the **exp** field set to 001. How far apart are they?
   Answer: 0 001 10->0.375_{10} 0 001 11-> 0.4375_{10} difference=0.4375-0.375=0.0625. 2^{-4}

   b. with the **exp** field set to 010. How far apart are they?
   Answer: 0 010 10->0.75_{10} 0 010 11->0.875_{10} difference=0.875-0.75 =0.125. 2^{-3}

   c. with the **exp** field set to 011. How far apart are they?
   Answer: 0 011 10->1.5_{10} 0 011 11->1.75_{10} difference=1.75-1.5=0.25. 2^{-2}

   Expressed these differences ("delta") as decimal numbers.

d. Without doing any calculations, can you guess how far apart are two adjacent normalized numbers …

   a. with the **exp** field set to 100? **2^{-1}=1/2=0.5**

   b. with the **exp** field set to 101? **2^{0}=1**

   c. with the **exp** field set to 110? **2^{1}=2**

e. What is the "range" (not contiguous) of real numbers that can be represented using this 6-bit floating-point representation?

   Answer: the range is between -14 and 14. Range: [-14~14].

f. What is the range of the normalized exponent **E** (**E** found in the equation $v = (-1)^s M 2^E$) which can be represented by this 6-bit floating-point representation?

Answer: Exp: 0~6 E: [-2~3]

g. Give an example of a real number that cannot be represented using this 6-bit floating-point representation, but is within the "range" of representable values.

Answer: for example, 0 101 00->4_{10} and 0 101 01->5_{10}. The number between 4 and 5 cannot be representation by 6-bit floating-point. Like 4.5.

h. Give an example of a real number that would overflow if we were trying to represent it using this 6-bit floating-point representation. The best way to answer this question is to convert this real number into a 6-bit IEEE floating-point representation and clearly indicate why it would overflow.

Answer: When over the biggest decimal value 14, going beyond his overflows to +∞. For example the decimal number $15_{10.}$ The binary of 15 is 01111_2 which cannot put 111 into exp part because it will become special value +∞/-∞. Frac part=000…0 which means operation that overflows. So 15 would have overflow occurred.

i. How close is the value of the "frac" of the largest normalized number to 1? In other words, what is ε (epsilon)? Expressed it as a decimal number.

Answer: 1-3/4=1/4. The epsilon is 1/4.