CSM 61C Spring 2020

Floating Point

Exam Question Compilation

This document is a PDF version of old exam questions by topic, ordered from least difficulty to greatest difficulty.

Questions:

- Fall 2019 Midterm Q1
- Spring 2018 Final Q9
- Summer 2019 Midterm 2 Q1
- Summer 2018 Final Q4
- Summer 2019 Final Q1.7-Q1.11
- Fall 2018 Final M2 Part 1
- Fall 2017 Final Q9
- Summer 2018 Midterm 1 Q1

<u>Q1) *Float, float on...*</u> (7 pts = 2 + 3 + 2)

You notice that floats can generally represent much larger numbers than integers, and decide to make a modified RISC instruction format in which all immediates for jump instructions are treated as 12-bit floating point numbers with a mantissa of 7 bits and with a standard exponent bias of 7. *Hint: Refer to reference sheet for the floating point formula if you've forgotten it...the same ideas hold even though this is only a 12-bit float…*

a) To jump the farthest, you set the float to be the most positive (not ∞) integer representable. *What are those 12 bits* (in hex)?	b) What is the *value* of that float (in decimal)?	c) Between 0 and (b)'s answer (inclusive), *how many integers are not representable*?
0x		

Problem 9 ***[F-1] Floating Point*** **(13 points)**

IEEE 754-2008 introduces half precision, which is a binary floating-point representation that uses 16 bits: 1 sign bit, 5 exponent bits (with a bias of 15) and 11 significand bits. This format uses the same rules for special numbers that IEEE754 uses. Considering this half-precision floating point format, answer the following questions:

(a) For 16-bit half-precision floating point, how many different valid representations are there for NaN?

____

(b) What is the smallest non-infinite number it can represent? You can leave your answer as an expression.

____

(c) What is the largest non-infinite number it can represent? You can leave your answer as an expression.

____

(d) How many floating point numbers are in the interval [1, 2) (including 1 but excluding 2)?

____

Question 1: Floating *Points to your Cheat Sheets - 10 pts

For all of the following questions we are using the IEEE 754 single precision floating point from lecture. If you do not remember the details, some can be found on the back side of the green sheet.

1. Represent **14.75** in its floating point representation. Put your answer in hexadecimal.

0x____

2. Represent **-2^{-147}** in its floating point representation. Put your answer in hexadecimal.

0x____

3. What value is represented by 0xFF800001?

____

For the remaining questions we are going to consider 2 possible changes:

- **Option 1**: Adding a bit to signficand and removing a bit from the exponent
- **Option 2**: Adding a bit to the exponent and removing a bit from the significand

For each of the following questions select whether **option 1, option 2, neither,** or **both** will accomplish the presented task. Assume that the bias also shifts to be $\mathbf{2^{exp\_bits - 1} - 1}$.

4. Represent pi more accurately than our IEEE 754 single precision floating point.

Ⓐ Option 1 Ⓑ Option 2 Ⓒ Neither Ⓓ Both

5. Represent smaller positive numbers than IEEE 754 single precision floating point.

Ⓐ Option 1 Ⓑ Option 2 Ⓒ Neither Ⓓ Both

6. Represent more numbers in the range [1, 2) than IEEE 754 single precision floating point.

Ⓐ Option 1 Ⓑ Option 2 Ⓒ Neither Ⓓ Both

7. Represent more numbers than IEEE 754 single precision floating point.

Ⓐ Option 1 Ⓑ Option 2 Ⓒ Neither Ⓓ Both

SID: ______________

Question 4: loating Point (9 pts)

Being the feisty floating point fanatic you are, you devise a new scheme to interpret 32-bit floats. You keep the breakdown of the Sign/Exponent/Significand fields the same. However, there is no implicit 1. or 0. before the significand bits; also, you evaluate the 23 significand bits as an **unsigned number**, which you then multiply with $2^{Exp - Bias}$ and $(-1)^{Sign}$ as you normally would.

$$(-1)^{Sign} \times Significand_{unsigned} \times 2^{Exp - Bias}$$

where $Bias$ = 127.
There are no denormalized numbers.

Exponent	Significand	Value
Largest	Zero	±Infinity
Largest	Nonzero	NaN

We walk through one way of representing 3 in this scheme, using a Significand of 0b00...011 = 3

$(-1)^0 \times 3 \times 2^{127 - Bias} = 1 \times 3 \times 2^0 = 3$ **Sign**: 0, **Exponent**: 127 = 0x7F, **Significand**: 3 = 0x000003

Answer the following questions comparing the standard IEEE Floating Point and your new scheme (let's call it Bloating Point).

1) Bloating Point represents a larger amount of unique numeric values than Floating Point. Ⓣ Ⓕ

2) 0x3D80001E is a valid representation of 1.875 in Bloating Point. (Hint: 1.875 = 15/8 Ⓣ Ⓕ

3) There exists a 32-bit number whose Floating Point value and Bloating Point value are the same. Ⓣ Ⓕ

4) How many Bloating Point numbers evaluate to $+2^{-124}$?

____

5) What is the smallest positive number divisible by 5 that Bloating Point cannot represent? You may leave all or part of your answer as a power of 2.

____

You propose a new 16 bit floating point number. It has:

- 1 sign bit
- 11 exponent bits
- 4 significand bits
- A bias of 1023
- All other rules consistent with IEEE 754 floating point.

7. Represent 4.75 in our new floating point scheme

Sign:0b____

Exponent: 0b____

Significand:0b ____

8. How many numbers does our floating point scheme represent in the range [0, 1) (the range 0 to 1, where 0 is included and 1 is not)? For this question assume -0 is not in this interval. You may leave your answer unsimplified.

____ numbers

Now let's compare to a 16 bit two's complement number.

9. Which can represent a larger number (ignore infinities)?

Ⓐ Our Floating Point Scheme Ⓑ Two's Complement

10. Which scheme represents more numbers in the range [1, 64)?

Ⓐ Our Floating Point Scheme Ⓑ Two's Complement

11. Which scheme represents more numbers in the range [64, 128)?

Ⓐ Our Floating Point Scheme Ⓑ Two's Complement

M2) *Floating down the C...* [this is a 2-page question] (8 points = 1,1,2,1,1,1,1, 20 minutes) SID_______

Consider an 8-bit "minifloat" SEEEMMMM (1 sign bit, 3 exponent bits, 4 mantissa bits). All other properties of IEEE754 apply (bias, denormalized numbers, ∞, NaNs, etc). The bias is -3.

a) How many minifloats are there in the range [1, 4)? (i.e., 1 ≤ **f** < 4) __________

b) What is the number line distance between 1 and the smallest minifloat bigger than 1? __________

c) Write **times2** in one line using integer operations. Assume the leftmost "E" bit of **f** (bolded above) is 0.

```
minifloat times2(minifloat f) { return f * 2.0; }
```

```
times2: ______ a0, a0, ______ ## Assume f is in the lowest byte of the register

        ret
```

Q9: Floating Point

Some of the 61C TAs get tired of having to convert floating-point values into 32 bits. As a result they propose the following smaller floating-point representation.
It consists of a total of 12 bits as shown below:

Sign (1)	Exponent (4)	significand (7 bits)
0	1 — 4	5 — 11

- The largest exp remains reserved as in traditional floating point
- The smallest exp follows the same denormalized formula as traditional floating point
- **Numbers are rounded to the closest representable value. Any numbers that have 2 equidistant representations are rounded down towards zero.**

All of the parts of this question refer to this floating-point scheme.

1. What is the smallest nonzero positive value that can be represented? Write your answer as a numerical expression in the answer packet.

2. Consider some **positive normalized** floating-point number p where p is described as p = 2^y * 1.significand. What is the distance (i.e. the difference) between p and the next-largest number after p that can be represented? Write your answer as a numerical expression.

3. Now instead let p be a positive denormalized number described as p = 2^y * 0.significand. What is the distance between p and the next-largest number after p that can be represented?

SID: ________________

Question 1: Number Representation and Floating Point (12 pts)

Given the following bit string 0b1111 1100, answer the following questions:

1) What is this bitstring's value if it was interpreted as an **unsigned number**?

____

2) What is this bitstring's value if it was interpreted in **two's complement**?

____

3) Suppose the bit string was represented as **fixed point** where the bits following the dot (.) represent the base (2) to the power of a negative exponent. What number does the bitstring 0b1111.1100 represent?

____

4) Now let's devise a scheme for interpreting fixed point numbers as positive or negative values. Complete the following sentence:
   Given an 8-bit fixed point bitstring 0bXXXX.XXXX with a value of Y, in order to compute -Y, we must flip all the bits of Y and add:

____

5) What is the value of 0b1111.1100 given the **two's complement fixed point** representation described above?

____

6) You are given the following field breakdown and specifications of an 8-bit floating point, which **follows the same rules** as standard 32-bit IEEE floats, except with different field lengths:

Sign: 1 bit
Exponent: 3 bits
Significand: 4 bits

Exponent Value	Significand Value	Floating Point Value
Smallest	Zero, Non-Zero	±0, Denormalized
Largest	Zero, Non-zero	±Infinity, NaN

What is the floating point value of 0b1111 1100:

____

7) We now modify the floating point description in part 6 so that the exponent field is now in **two's complement** instead of in bias notation. Compute the floating point value of 0b1111 1100.

____