AMATH 483 / 583 - HW1

Contents

1 Problems

This assignment is due Friday April 7 2023 by midnight PDT.

1. Write a C or C++ program that finds a practical measure of your machine's SP (32 bit) and DP (64 bit) precision by taking the difference of 2 numbers and comparing the result to zero in each data type. You will submit the code, and in the written homework will state the values you obtained running your code for each data type. You may find https://www.w3schools.com/cpp/cpp_while_loop.asp useful if you are beginning in C++ programming.Use `while` and iterate on $j = 0, 1, \ldots$. Hint: `while` (($1-(1+\frac{1}{2^j}) \neq 0$) {... `j++;`}

2. What are the largest and smallest SP (32 bit) and DP (64 bit) numbers that can be represented in IEEE arithmetic? Show your work as this is analytic.

3. Write a C or C++ program to multiply the integers 200*300*400*500 on your computer? What is the result? You will submit the code, and in the written homework will name the effect you observed AND provide a definition and math formula for *underflow* and *overflow* for IEEE SP and DP representations.

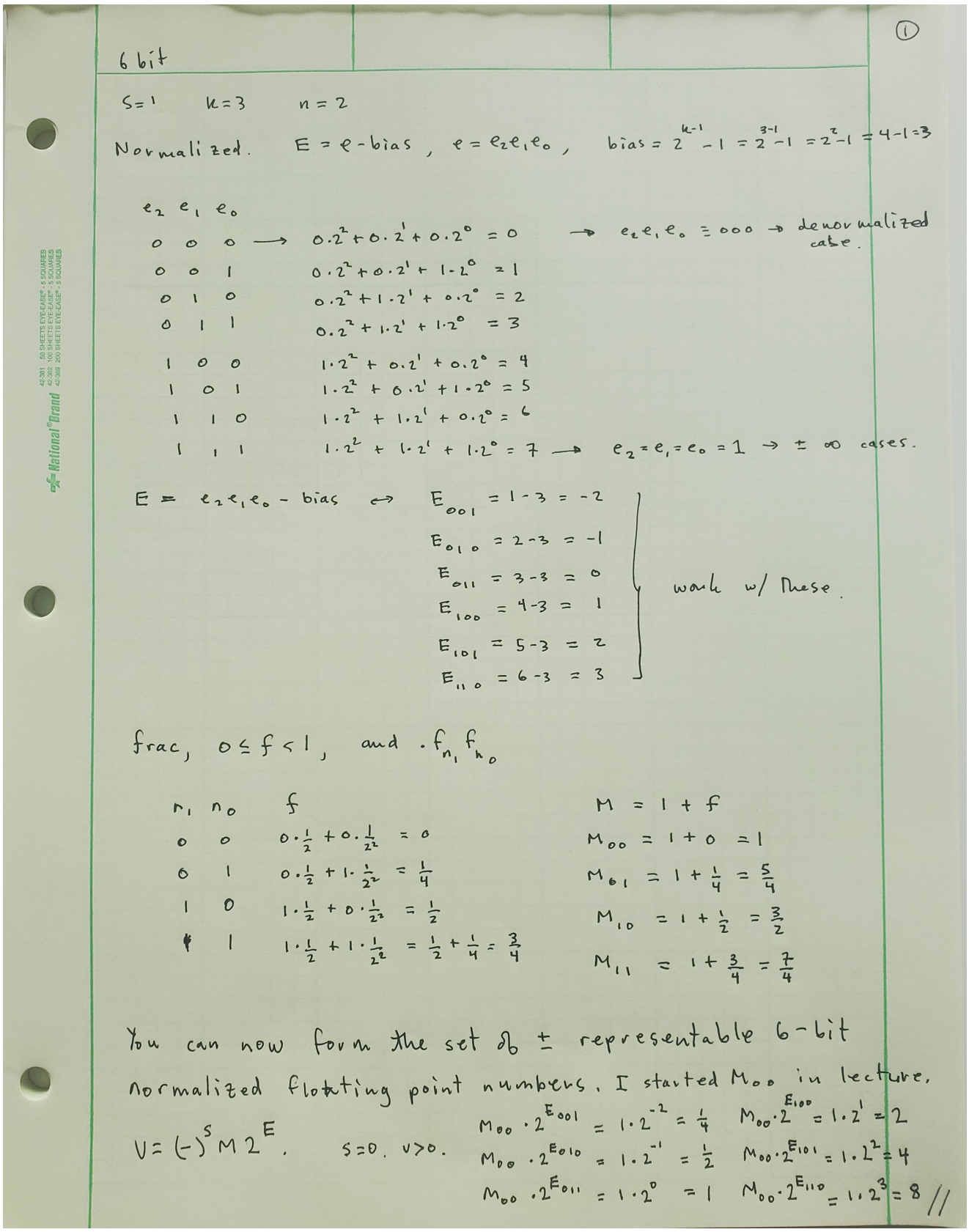
4. How many SP (32 bit) normalized floating point numbers are there? Same question for DP (64 bit). Provide a math formula and show your work.

5. Consider a 6 bit floating point system with one sign bit ($s = 1$), a 3-bit exponent ($k = 3$), and a 2-bit mantissa ($n = 2$). Enumerate by hand all the representable normalized and denormalized numbers. Plot the distribution of the representable numbers on a line. I gifted you a head start in the support materials section.

Other Read Lumsdaine's problem set 1: https://amath583.github.io/sp21/assignments/ps1.html Create a directory (folder) named `hw1_<uwnetid>`. Put a copy of your written solutions in a single `.pdf` file, and your two C or C++ computer programs for problems 1 and 3 in the directory. From a terminal (SHELL) please issue the command: `tar -cvf hw1_<uwnetid>.tar hw1_<uwnetid>`. You will submit the file `hw1_<uwnetid>.tar` to Canvas for grading.

```
[WE42365:~/Desktop] d3y402% mkdir hw1_k8r
[WE42365:~/Desktop] d3y402% mv k8r_hw1.pdf k8r_hw1_p1.cpp k8r_hw1_p3.cpp hw1_k8r/
[WE42365:~/Desktop] d3y402% tar -cvf hw1_k8r.tar hw1_k8r/
hw1_k8r/
hw1_k8r/k8r_hw1.pdf
hw1_k8r/k8r_hw1_p1.cpp
hw1_k8r/k8r_hw1_p3.cpp
[WE42365:~/Desktop] d3y402% ls -lstr hw1_k8r.tar
24 -rw-r--r--  1 d3y402  PNL\Domain Users  10240 Mar 31 17:27 hw1_k8r.tar
[WE42365:~/Desktop] d3y402%
```

2 Support Materials



6 bit ①

$s=1 \quad k=3 \quad n=2$

Normalized. $E = e - \text{bias}$, $e = e_2e_1e_0$, $\text{bias} = 2^{k-1}-1 = 2^{3-1}-1 = 2^2-1 = 4-1=3$

e_2 e_1 e_0	
0 0 0	$0\cdot2^2+0\cdot2^1+0\cdot2^0 = 0$ → $e_2e_1e_0 = 000$ → denormalized case.
0 0 1	$0\cdot2^2+0\cdot2^1+1\cdot2^0 = 1$
0 1 0	$0\cdot2^2+1\cdot2^1+0\cdot2^0 = 2$
0 1 1	$0\cdot2^2+1\cdot2^1+1\cdot2^0 = 3$
1 0 0	$1\cdot2^2+0\cdot2^1+0\cdot2^0 = 4$
1 0 1	$1\cdot2^2+0\cdot2^1+1\cdot2^0 = 5$
1 1 0	$1\cdot2^2+1\cdot2^1+0\cdot2^0 = 6$
1 1 1	$1\cdot2^2+1\cdot2^1+1\cdot2^0 = 7$ → $e_2=e_1=e_0=1$ → $\pm\infty$ cases.

$E = e_2e_1e_0 - \text{bias}$ ↔

$E_{001} = 1-3 = -2$
$E_{010} = 2-3 = -1$
$E_{011} = 3-3 = 0$
$E_{100} = 4-3 = 1$
$E_{101} = 5-3 = 2$
$E_{110} = 6-3 = 3$

} work w/ these.

frac, $0 \le f < 1$, and $.f_{n_1}f_{n_0}$

n_1 n_0	f
0 0	$0\cdot\frac{1}{2}+0\cdot\frac{1}{2^2} = 0$
0 1	$0\cdot\frac{1}{2}+1\cdot\frac{1}{2^2} = \frac{1}{4}$
1 0	$1\cdot\frac{1}{2}+0\cdot\frac{1}{2^2} = \frac{1}{2}$
1 1	$1\cdot\frac{1}{2}+1\cdot\frac{1}{2^2} = \frac{1}{2}+\frac{1}{4} = \frac{3}{4}$

$M = 1 + f$

$M_{00} = 1+0 = 1$
$M_{01} = 1+\frac{1}{4} = \frac{5}{4}$
$M_{10} = 1+\frac{1}{2} = \frac{3}{2}$
$M_{11} = 1+\frac{3}{4} = \frac{7}{4}$

You can now form the set of ± representable 6-bit normalized floating point numbers. I started M_{00} in lecture.

$V = (-)^S M 2^E$. $S=0$. $V>0$.

$M_{00}\cdot2^{E_{001}} = 1\cdot2^{-2} = \frac{1}{4}$ $M_{00}\cdot2^{E_{100}} = 1\cdot2^1 = 2$
$M_{00}\cdot2^{E_{010}} = 1\cdot2^{-1} = \frac{1}{2}$ $M_{00}\cdot2^{E_{101}} = 1\cdot2^2 = 4$
$M_{00}\cdot2^{E_{011}} = 1\cdot2^0 = 1$ $M_{00}\cdot2^{E_{110}} = 1\cdot2^3 = 8$ //

for $s=1$, $v<0$ and M_{00} case we get $\{-\frac{1}{4}, -\frac{1}{2}, -1, -2, -4, -8\}$

$s=0$, $v>0$ and M_{00}, $\{\frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8\}$.

In the HW, please find the normalized and denormalized representable 6-bit numbers. +2 points for plotting the distribution. Let me help you a little more. Make sure you get these in your analysis.

Largest denormalized.

Def. $e_2e_1e_0 = 000$. $E = 2 - 2^{k-1} = 2 - 4 = -2$.

frac, all 1s

$M = f = 1 - 2^{-n}$. Here $n=2$. $M = 1 - 2^{-2} = 1 - \frac{1}{4} = \frac{3}{4}$

$V_{\text{largest denormalized}} = (-1)^0 \cdot M \cdot 2^E = 1 \cdot \frac{3}{4} \cdot 2^{-2} = \frac{3}{16}$ //

Smallest denormalized.

1 in least significant bit, 0s otherwise

$M = f = 2^{-n} = 2^{-2} = \frac{1}{4}$

$E = 2 - 2^{k-1} = 2 - 2^2 = 2 - 4 = -2$

$V = (-)^s M 2^E$. $V_{\text{smallest denormalized}} = (-)^1 \frac{1}{4} \cdot 2^{-2} = -\frac{1}{16}$ //

Largest Normalized. $V = (2 - 2^{-n}) \cdot 2^{2^{k-1} - 1}$

$V_{\text{largest normalized}} = (2 - 2^{-2}) \cdot 2^{2^{3-1} - 1} = \left(2 - \frac{1}{4}\right) \cdot 2^3 = \frac{7}{4} \cdot 8$

$= 14$ //

Smallest Normalized. ($s=0$ case) $V = 2^{-2^{k-1} + 2} = 2^{-2^2 + 2} = 2^{-4+2} = 2^{-2} = \frac{1}{4}$ //

$V_{\text{smallest normalized}}$