CSARCH2 Mock Long Exam 1

by: Clive Jarel Ang

Important Reminders:

1. Read ALL instructions carefully and thoroughly before answering this mock exam.

2. The use of calculators and other computing devices are NOT allowed in the exam. However, you will be answering this in the comfort of your own home, so I literally have 0 control over the enforcement of that rule. ¯_(ツ)_/¯

3. Cheating in ANY form during the actual exam will be considered a major offense, merit you a 0.0 in the course, and would result in both Sir Rog and I becoming very sad.

4. This exam is GOOD FOR 3 HOURS. To be sufficiently prepared for the long exam proper, try to finish this mock exam in a shorter amount of time (while keeping a high score obv).

5. Yes, I'm sadistic and this mock exam reflects that.

I. Concepts of Data Representation in Memory

1) Which signed integer format/s has a different representation for both -0 and +0?	
2) What is the bias or excess for E' in IEEE-754 single-precision?	
3) What is the bias or excess for E' in IEEE-754 double-precision?	
4) What does sNaN stand for?	
5) What special case in binary floating point numbers has the quiet bit set to 1?	
6) What is the default rule used when performing Round to Nearest? (tie away from zero, tie to even)	

II. Understanding Integer Representation

Bits	Format	Lower bound	Upper Bound
8	Unsigned		
	S&M		
	1's C		
	2's C		

Bits	Format	Lower bound	Upper Bound
16	Unsigned		
	S&M		
	1's C		
	2's C		

III. Integer Representation

Represent the following decimal integers to the specified integer representation. **Answer in hexadecimal**. If the number cannot be represented, write "N/A".

Decimal	8-bit Unsigned Integer	8-bit Signed Integer (S&M)	8-bit Signed Integer (1's C)	8-bit Signed Integer (2's C)
-0				
+42				
-82				
+127				
-127				
+128				
-130				
+255				

Decimal	16-Bit Unsigned Integer	16-Bit Signed Integer (S&M)	16-Bit Signed Integer (1's C)	16-Bit Signed Integer (2's C)
-32768				
+32767				
+40000				
+65535				

IV. Operations on Signed and Unsigned Integers

Equation	Output	Will it overflow if seen as...	
		unsigned?	signed?
0010 1010 - 0101 1010			
1100 1101 + 1001 1111			
1100 0001 - 1001 1100			
0110 1000 + 0101 1010			

V. Floating Point Representation

- For E', put a space every 4 bits.
- For the fractional part, use ellipse.
- If the answer is specified to be in hex, put a space every 4 hex digits.
- Write "N/A" if it can't be represented.
- If applicable, specify special cases after mantissa (+/- Infinity, sNaN, qNaN, Denormalized).

Express the following using IEEE 754 Single Precision (Binary-32) format:

1. $-1.011_2 \times 2^{-2}$
2. $+0.0000001_2 \times 2^{-120}$
3. $-101.25_{10} \times 2^2$
4. $+110.1001101011_2 \times 2^{127}$

#	Sign Bit	Exponent	Mantissa
1.			
2.			
3.			
4.			

Express the following using IEEE 754 Double Precision (Binary-64) format:

1. $+1.1_2 \times 2^{1023}$
2. $+19.0375_{10} \times 10^3$

#	IEEE 754 Double Precision Format (**IN HEX**)
1.	
2.	

VI. Internal Memory Representation

- Write "N/A" if it can't be represented.
- If applicable, specify special cases (+/- Infinity, sNaN, qNaN, Denormalized).

Internal Memory (hexadecimal)	View as ...	Decimal or Special Case Equivalent
	%hhu	105
	%hhd	-2
0xFFFFC	%hd	
0x7FC00000	%f	
	%f	- Infinity
0x8000000000000008	%lf	

VII. Floating Point Rounding 1

Round to 7 bits	Truncate	Floor	Ceiling	Round to nearest (tie to even)
$+12345.6789_{10}$				
-0.00098650_{10}				
$+1.10110110_2$				
-0.00011011_2				
$+9.99999949_{10}$				
-101.111111_2				

VIII. Floating Point Rounding 2

$+69.490550_{10}$	7 decimal digits	6 decimal digits	5 decimal digits	4 decimal digits
Truncate				
Floor				
Ceiling				
Round to nearest (tie to even)				

-111.1101010_2	8 binary digits	7 binary digits	6 binary digits	5 binary digits
Truncate				
Floor				
Ceiling				
Round to nearest (tie to even)				

IX. Floating Point Operations

Perform the computation $1.0110010110 11_2 \times 2^6 \ + \ 1.1001101101_2 \times 2^3$ to 8 bits (use round to nearest, ties to even if needed). All answers should be in normalized form.

a) Perform without guard, round, and sticky bits.

		$Base^{Exp}$
Operand 1		
Operand 2		
Final Sum		

b) Perform with guard(G), round(R), and sticky(S) bits.

		G	R	S	$Base^{Exp}$
Operand 1					
Operand 2					
Final Sum		-	-	-	

X. Memorizing Base-2 [Bonus]

x	2^x
1	
2	
3	
4	
5	
6	
7	
8	
9	

x	2^x
10	
11	
12	
13	
14	
15	
16	
17	
18	

x	2^x
19	
20	
21	
22	
23	
24	
25	