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?	1's Complement and S&M
2) What is the bias or excess for E' in IEEE-754 single-precision?	+127
3) What is the bias or excess for E' in IEEE-754 double-precision?	+1023
4) What does sNaN stand for?	Signaling Not a Number
5) What special case in binary floating point numbers has the quiet bit set to 1?	qNaN
6) What is the default rule used when performing Round to Nearest? (tie away from zero, tie to even)	Tie to Even

II. Understanding Integer Representation

Bits	Format	Lower bound	Upper Bound
8	Unsigned	0	255
	S&M	-127	+127
	1's C	-127	+127
	2's C	-128	+127

Bits	Format	Lower bound	Upper Bound
16	Unsigned	0	65 535
	S&M	-32 767	+32 767
	1's C	-32 767	+32 767
	2's C	-32 768	+32 767

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	N/A	0x80	0xFF	0x00
+42	0x2A	0x2A	0x2A	0x2A
-82	N/A	0xD2	0xAD	0xAE
+127	0x7F	0x7F	0x7F	0x7F
-127	N/A	0xFF	0x80	0x81
+128	0x80	N/A	N/A	N/A
-130	N/A	N/A	N/A	N/A
+255	0xFF	N/A	N/A	N/A

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	N/A	N/A	N/A	0x8000
+32767	0x7FFF	0x7FFF	0x7FFF	0x7FFF
+40000	0x9C40	N/A	N/A	N/A
+65535	0xFFFF	N/A	N/A	N/A

IV. Operations on Signed and Unsigned Integers

Equation	Output	Will it overflow if seen as...	
		unsigned?	signed?
0010 1010 - 0101 1010	1101 0000	Yes	No
1100 1101 + 1001 1111	0110 1100	Yes	Yes
1100 0001 - 1001 1100	0010 0101	No	No
0110 1000 + 0101 1010	1100 0010	No	Yes

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.	1	0111 1101	0110..0
2.	0	0000 0000	10..0 (Denormalized)
3.	1	1000 0111	100 1010 10..0
4.	0	1111 1111	0..0 (+ Infinity)

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.	0x7FE8 0000 0000 0000
2.	0x40D2 9760 0000 0000

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
0x69	%hhu	105
0xFE	%hhd	-2
0xFFFFC	%hd	-4
0x7FC00000	%f	qNaN
0xFF800000	%f	- Infinity
0x8000000000000008	%lf	Denormalized

VII. Floating Point Rounding 1

Round to 7 bits	Truncate	Floor	Ceiling	Round to nearest (tie to even)
$+12345.6789_{10}$	+12345.67	+12345.67	+12345.68	+12345.68
-0.00098650_{10}	−0.000986	−0.000987	−0.000986	−0.000986
$+1.10110110_2$	+1.101101	+1.101101	+1.101110	+1.101110
-0.00011011_2	−0.000110	−0.000111	−0.000110	−0.000111
$+9.99999949_{10}$	+9.999999	+9.999999	+10.00000	+9.999999
-101.1111111_2	−101.1111	−110.0000	−101.1111	−110.0000

VIII. Floating Point Rounding 2

$+69.490550_{10}$	7 decimal digits	6 decimal digits	5 decimal digits	4 decimal digits
Truncate	+69.49055	+69.4905	+69.490	+69.49
Floor	+69.49055	+69.4905	+69.490	+69.49
Ceiling	+69.49055	+69.4906	+69.491	+69.50
Round to nearest (tie to even)	+69.49055	+69.4906	+69.491	+69.49

-111.1101010_2	8 binary digits	7 binary digits	6 binary digits	5 binary digits
Truncate	−111.11010	−111.1101	−111.110	−111.11
Floor	−111.11011	−111.1110	−111.111	−1000.0
Ceiling	−111.11010	−111.1101	−111.110	−111.11
Round to nearest (tie to even)	−111.11010	−111.1101	−111.111	−111.11

IX. Floating Point Operations

Perform the computation $1.0110010110112_2 \times 2^6 \; + \; 1.1001101101_2 \times 2^3$ 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	1.0110011	2^6
Operand 2	0.0011010	2^6
Final Sum	1.1001101	2^6

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

		G	R	S	$Base^{Exp}$
Operand 1	1.0110010	1	1	1	2^6
Operand 2	0.0011001	1	0	1	2^6
Final Sum	1.1001100	-	-	-	2^6

X. Memorizing Base-2 [Bonus]

x	2^x
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512

x	2^x
10	1 024
11	2 048
12	4 096
13	8 192
14	16 384
15	32 768
16	65 536
17	131 072
18	262 144

x	2^x
19	524 288
20	1 048 576
21	2 097 152
22	4 194 304
23	8 388 608
24	16 777 216
25	33 554 432