CS61c Midterm Review (fa06) answers

Number representation and Floating points

Somewhat-Trivial Questions:

1) What is 0xffff ffff in decimal form (you can use $*2^y$ in your answer)
   - Unsigned int: $2^{32}-1$
   - Sign and magnitude: $-2^{31}-1$
   - One's complement: -0
   - Two's complement: -1
2) J-Lo whispers her age to you: "Thirty six". Was that big-endian or little-endian (mt fa05)
   Big-Endian
3) Let x = 0xffff ffff. What is the decimal value (if any) using Single Precision floats?
   NaN .
4) Let x = 0x0040 0000. What is the decimal value (if any) using Single Precision floats?
   2^{-127} .
5) What is a mebi? 2^{20} .
6) Let x = 0x4000 0000, y = 0x3f800000. What is x+y? (truncate)
   3 .

Conceptual/Midterm type Questions:

7) Let x = 0x4000 0000, y = 0x0040 0000. What is x+y? (truncate)
   2 .
8) What is minimum positive *float x* such that x + 1 = x (You've done this before in lab, and this *has* been on the Midterm before!)
   $X = 2^{24}$ = 0b4b800000
9) True/**False**: Float arithmetic is associative. i.e. (x+y)+z = x+(y+z) . If not, show a counterexample. $X+1+1 = 2^{24}+1+1$
10) Let x = Mebi, find y such that x+y have 0xb11 in the rounding bits on the alignment step (assume there are 2 such bits)
    i.e. x = 1.********** ********** *** | 00 * 2^n
    y = *.********** ********** *** | 00 * 2^n
    x + y = *.********** ********** *** | 11 * 2^n
    $x = 2^{20}$
    $x = 1*2^{20+127}$ = 0x4980 0000
    easiest to find x+y = 1.000….0 | 11 * 2^{20}
    thus y = 0.00….0 | 11 * 2^{20} = $2^{20}*(2^{-24}+2^{-25})$
    $= 2^{-4}*2^{-5}$
    $= 2^{-4}*(1 + ½)$
    y = 0x3dc0 0000
    - Compute x+y, using truncation.
      x+y = 1.000….0 | 11 * 2^{20}
      = 1.000….0 * 2^{20}
      = 2^{20} = 0x4980 0000
    - Compute x+y using "Round to zero"
      x+y = 1.000….0 | 11 * 2^{20}

$= 1.000\ldots.0 * 2^{20}$

$= 2^{20} = 0x4980\ 0000$

- Compute x+y using "round to Plus Infinity"

$x+y = 1.000\ldots.0 \mid 11 * 2^{20} = 2^{20} = 0x4980\ 0000$

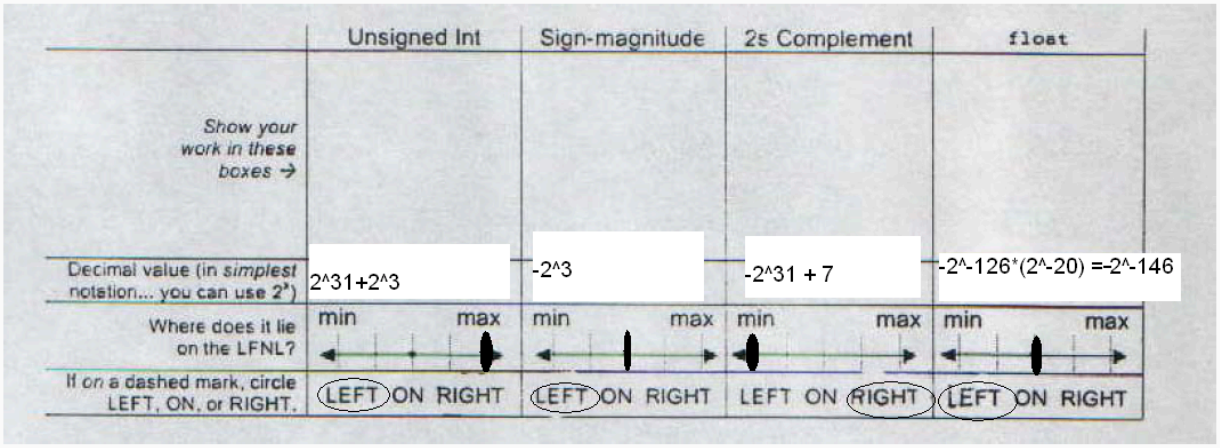
$= 1.000\ldots.1 * 2^{20}$

$= (1+2^{-23}) * 2^{20}$

$= 2^{20} + 1/8 = 0x4980\ 0001$

From Dan's previous Midterm (fa05)

a) Your favorite 32-bit hex quantity is `0x80000008`. For each of the encodings, tell us what the decimal value of this number is, and where on a linear, finite number line (LFNL) that number would lie given *all* the possible *finite* numbers that can be encoded. There are five dashed marks on our LFNL: the leftmost dash marks "min", the smallest encodable *finite* value (closest to -∞) and the rightmost dash marks "max", the largest. The middle dash marks the halfway point between min and max (not necessarily zero!), and the other two dashes mark ¼ and ¾ of the distance from min to max. If the mark you make falls pretty much on one of the existing dashed marks, in the space below the LFNL circle whether (if we were to zoom in) your mark would be to the "LEFT", "ON", or to the "RIGHT" of our dashed mark. E.g., if the 32-bit hex quantity were `0x0`, the Sign-magnitude we'd put our mark directly on the *middle* dashed mark and circle "ON". If it were `0x1`, it's be the same mark but we'd circle "RIGHT". Remember, this number line is linear & finite! Show your work.



b) What is min, max, and positive min (greater than 0) for each of the types specified above?

Type:	Min	max	positive (normalized) min (>0)
Unsigned int	0	$2^{32}-1$	1
Sign-Magnitude	$-(2^{31}-1)$	$2^{31}-1$	1
2s complement	-2^{31}	$2^{31}-1$	1
Float			2^{-126}

c) What is the largest (unsigned) integer that a float/double can store?

float: $2^8 + 2^9 + \ldots + 2^{31} = 2^{31} * (\sum_{i=0}^{23} 2^{-i}) = 2^{31} * (1 + \sum_{i=1}^{23} 2^{-i})$ =0x4f7f ffff

double: $2^{32}-1=\sum_{i=0}^{31}2^{i}$ = 0x41e ffff fffc 0000 0000 0000 0000 0000

d) What is the largest (unsigned) integer that float/double can't store?

float: $2^{32}-1$ double: none.