Yingjie Lian
HW- 5
CS-3810
02.20.2018

Answer 1:

First, since 26 decimal in binary = 00011010
so -26 decimal in binary format is 2's complement of above

-26= 11100110

0.3125 decimal in binary = 01010000000000000000
so -0.3125 decimal in binary format is 2's complement of above
- .3125 = 10110000000000000000

So -26.3125 to IEEE 754 single precision floating point format=

1 11100110 10110000000000000000

Answer 2:

So first bit shows number is negative.
So calculating the original number we have to find 2's complement for exponent and fraction part
Equals to 2's complement of 10000111 = 01111001 = 121
So decimal equivalent of exponent = -121
Similarly decimal equivalent of fractional part 2's complement of 11010000000000000000 = 00110000000000000000 = 0.1875
So overall IEEE 754 single-precision floating-point register 1 10000111
11010000000000000000 = decimal **-120.1875**

Answer 3:

So first bit shows number is negative.
So calculating the original number we have to find 2's complement for exponent and fraction part
Equals to 2's complement of 10000000011 = 011111111101 = 1021

So decimal equivalent of exponent = -1021
Similarly decimal equivalent of fractional part 2's complement of
10100000000000000000000000000000000000000000000000 =
01100000000000000000000000000000000000000000000000 = 0.375
So overall the decimal number is **-1021.375**

Answer 4:

We assume A = 0 10000010 00100000000000000000000
B = 0 10000101 11000000000000000000000
In this form, it would be easier for me to describe.
First, we take a look at the first digit. Since they are both 0, we know that A+B must be positive.
Then, we will convert the exponent of A and B to decimal.
So the exponent of A = 130, the exponent of B = 133
Next, we normalize to the higher exponent. In this case, we will normalize the exponent of A.
Difference = 133 -130 = 3
So A 23 bits = 1.00100000000000000000000 * 2^130
So we will rewrite A 23 bits as = 0.00100100000000000000000 * 2^133
The last two digits of original A 23 bits get lost because the computer has no space to store it.

Then we have B 23 bits here = 1.11000000000000000000000 * 2^133
So A+B 23 bits will be = 1.11100100000000000000000

Finally, the answer of A+B (32 bits) is **0 10000101 11100100000000000000000**

Answer 5:

Compute the truth table for a logic block that takes in a 3-bit input and produces a 2-bit output that represents the number of 1's in the 3-bit input.
Let “ABC” be the input, then “XY” be the output.

A	B	C	X	Y
0	0	0	0	0
1	0	0	0	1
1	1	0	1	0

1	1	1	1	1
1	0	1	1	0
0	1	1	1	0
0	0	1	0	1
0	1	0	0	1

SOP Boolean equations:

X = ABC’ + ABC + AB’C + A’BC

Y = AB’C’ + ABC + A’B’C + A’BC’