Nate Flasher, Tessa D'Souza, Joey Gorski – Floating Point Representation Homework

2.85)

a) Formula for exponent E = Exp – Bias => 3 - (2^1-1) = 2

M = 1.11 = 7/4

F = 0.11 = ¾

v = ((-1)*s)*M*2^E => (1)*(7/4)*(2^2) = 7

Bit representation (not twos complement) will look like: Sign bit is 0. Exp bits are 00000010. Frac bits are 1100…..0

b) z = length of number in bits

exp = k-1

bias = 2^(k-1)-1

E = Exp – Bias => k-1 - 2^(k-1)-1

M = 1.1111111111111….1

F = 0.1111111111111…..1

v = ((-1)*s)*M*2^E => (1)*(a number super close to 2)*((k-1) – 2^(k-1)-1) = largest possible odd number

Bit representation (not twos complement) will look like: sign bit is 0, exp bits are (n+2^(k-1), fraction bits will be 1111111111……..1.

c) exp = 2^(k-1)

bias = 1

E = Exp – Bias => 2^(k-1)-1-1

M = 1

F = 0

v = ((-1)*s)*M*2^E => (1)*(a number super close to 2)*(2^(n-1) - 2^((n-1)-1)-1) = largest possible odd number

Bit representation (not twos complement) will look like: sign bit is 0, exp bits are 11….101, frac bits are 00…..00.

2.86)

Bias = (2^k-1)-1 => (2^14)-1 = 16383

E (for the smallest possible denormalized and normalized value) = (exp – bias) => (1 – 16383) = -16382

E (for the largest possible normalized value) = (exp – bias) => (65534 – 16383) = 49152

65534 came from the binary value 1111111111111110 because this is the max possible value with an exp of size 15. Then we subtract that from the bias to get 49152.

	Value	**Decimal**
Smallest (+) denormalized	(2^(-63))*(2^(-16382))	An extremely small number
Smallest (+) normalized	1(sign)*1(fractional)*(2^(-16382))	Extremely small number large than a denormalized number
Largest (+) normalized	1*(2^64)*(2^49152)	An extremely large number

2.87)

Description	**Hex**	**M**	**E**	**V**	**D**
-0:	8000	0.0	0	-0	-0.0
smallest value > 2:	4001	1025/1024	1	1025*2^-9	2.001953125
512:	6000	1.0	9	512*2^0	512.0
largest denormalized:	03FF	1+(1/0.9990234375)	-15	1.9990234375*2^0	1.9990234375
-infinity:	FC00	--	--	-inf	-inf
3BB0:	3BB0	1+(1/0.921875)	-1	1.0921875*2^-1	0.54609375

2.89)

a) Yields 1 because both values of x will be rounded in the same way.

b) Does not yield 1 if you set y = to two's complement minimum and x = 0.

c) Yields 1 because all values will be between two's complement minimum **and** two's complement maximum.

d) Does not yield 1 if dx = max val, dy = max val -1, and dz = max val -2.

e) Does not yield 1 if x = 1 and z = 0.

2.90)

Blank 1: -149

Blank 2: 0

Blank 3: 0

Blank 4: -126

Blank 5: 0

Blank 6: 1 << (x+149)

Blank 7: 128

Blank 8: x + 127

Blank 9: 0

Blank 10: 255

Blank 11: 0