Willis Allstead
2/20/18
CS 219

HW4

9.5

a) $001100 = 0\cdot2^5+0\cdot2^4+1\cdot2^3+1\cdot2^2+0\cdot2^1+0\cdot2^0 = 0+0+8+4+0+0 = 12$
b) $000011 = 1\cdot2^1+1\cdot2^0 = 3$
c) $011100 = 1\cdot2^4+1\cdot2^3+1\cdot2^2+0\cdot2^1+0\cdot2^0 = 28$
d) $111100 = 1\cdot2^5+1\cdot2^4+1\cdot2^3+1\cdot2^2+0\cdot2^1+0\cdot2^0 = 60$
e) $101010 = 1\cdot2^5+0\cdot2^4+1\cdot2^3+0\cdot2^2+1\cdot2^1+0\cdot2^0 = 42$

9.11

a) $C = 12\cdot16^0 = 12$
b) $9F = 9\cdot16^1+15\cdot16^0 = 144+15 = 159$
c) $D52 = 13\cdot16^2+5\cdot16^1+2\cdot16^0 = 3328+80+2 = 3410$
d) $67E = 6\cdot16^2+7\cdot16^1+14\cdot16^0 = 1536+112+14 = 1662$
e) $ABCD = 10\cdot16^3+11\cdot16^2+12\cdot16^1+13\cdot16^0 = 40960+2816+192+13 = 43981$

10.1

In sign-magnitude representation, if the MSB is 0 then the number is positive, else it is negative. Two's complement representation is found by finding the sign-magnitude representation and adding a binary 1 to the binary number if the integer is negative. In Biased representation, a fixed number is added to the integer.

10.6

Two's complement of a number is a process/operation performed on a binary number which allows the user to know if a number is negative or not. Two's complement *representation* is how the result of that process is represented, similar to binary representation, hex representation, etc.

10.8

Significand, base, sign, exponent

10.9

Non-negative floating-point numbers can be treated as integers for comparing. Two's complement, which would normally be used for signed values, would make comparison more difficult than biased representation

10.10

Positive overflow happens when a value goes above the boundary limits. Exponent overflow happens with floating point numbers when the exponent part of the floating point number exceeds the maximum. Significand overflow also happens with floating point numbers and happens when a carry is started from the MSB when two significands are added.

10.23

a) -5 = 1-10000001-01000000000000000000000
b) -6 = 1-10000001-10000000000000000000000
c) -1.5 = 1-01111111-10000000000000000000000
d) 384 = 0-10000111-10000000000000000000000

e) 1/16 = 0-01111011-00000000000000000000000
f) -1/32 = 1-01111010-00000000000000000000000