Nick Hall

Assembly Language HW1

1.

A. 0011 = 3, 1101 = D, 1101 = D, 1010 = A Answer = 3DDA

B. 1100 = C, 1010 = A, 1010 = A, 0011 = 3 Answer = CAA3

C. 1101 = D, 1110 = E, 0111 = 7, 1001 = 9 Answer = DE79

D. 1100 = C, 0010 = 2, 1101 = D, 0011 = 3 Answer = C2D3

2. To convert signed hexadecimal numbers to decimal check to see if the number is negative or positive by looking at the first digit. If it is >= 8 the number is negative and if it is <= 7 the number is positive. If the number negative create a two's compliment by subtracting 15 from each digit and then add one to that number. Then convert to decimal.

A. 5CE9 = (5X16^3) + (12X16^2) + (14X16^1) +(9X16^0) = 23,785

B. B143 = 4EBC + 1 = 4EBD = (4X16^3) + (14X16^2) + (11X16^1) + (13X16^0) = -20,157

C. 8CF2 = 730D + 1 = 730E = (7X16^3) + (3X16^2) + (0X16^1) + (14X16^0) = -29,454

D. 52D0 = (5X16^3) + (2X16^2) + (13X16^1) + (0X16^0) = 21,200

3. To convert signed binary numbers that start with 1 to decimal create a two's compliment of the binary number and convert to decimal. Since the number started with 1 it is a negative number. If the original binary number started with a 0 it can be converted to decimal as if it is an unsigned binary number

A. 10110011 = 01001100 + 1 = 01001101 = (1X2^6) + (1X2^3) + (1X2^2) + (1X2^0) = -77

B. 00101010 = (1X2^5) + (1X2^3) + (1X2^1) = 42

C. 11110001 = 00001110 + 1 = 00001111 = (1X2^3) + (1X2^2) + (1X2^1) + (1X2^0) = -15

D. 11011100 = 00100011 + 1 = 00100100 = (1X2^5) + (1X2^2) = -36

4.

412/2 = 206 Remainder = 0

206/2 = 103 Remainder = 0

103/2 = 51 Remainder = 1

51/2 = 25 Remainder = 1

25/2 = 12 Remainder = 1

12/2 = 6 Remainder = 0

6/2 = 3 Remainder = 0

3/2 = 1 Remainder = 1

1/2 = 0 Remainder = 1

.75X2 = 1.50 Carry = 1

.50X2 = 1.00 Carry = 1

Answer = 0000000110011100.11

5.

Two's compliment gave the correct answer when subtracting binary numbers. One's compliment produces a result that is one less than the correct answer. Also, One's compliment did not produce 0 when the positive and negative version of the number were added together. Two's compliment solves both of these problems. First, two's compliment adds one to the one's compliment. Second, when the two's compliment version of the number is added to the inverse it produces 0. Finally, the two's compliment form of the number can do subtractions much easier. This makes it more efficient because the microprocessor only has to add two numbers together, which uses the same circuitry that is used for addition.

http://www.learnabout-electronics.org/Digital/dig15.php