

Assignment 2

1. (4 points) Write the following both as binary numbers and as binary-encoded decimals
 - (a) 456 decimal equals 0100 0101 0110 as binary encoded decimal and $456_{(10)} = 1\ 1100\ 1000_{(2)}$ as binary
 - (b) 623 decimal equals 0110 0010 0011 as binary encoded decimal and $623_{(10)} = 1\ 0110\ 1111_{(2)}$ as binary
2. (4 points) For the following 8 bit binary numbers, write the 1's compliment and the 2's compliment
 - (a) $b = 0110\ 1111$
One's compliment: 1001 0000
Two's compliment: 1001 0001
 - (b) $a = 0011\ 1000$
One's compliment: 1100 0111
Two's compliment: 1100 1000.
3. Using the values for a and b of the previous problem
 - (a) (2 points) Find the decimal values for a and b .
 $b = 64 + 32 + 15 = 111$
 $a = 32 + 16 + 8 = 56$
 - (b) (4 points) Using 1's compliment method, find the value of $a - b$ and $b - a$.

Recall from the lecture that:

$$b - a = b + (1\text{s compliment of } a) + 1 - 10000\ 0000 \quad (1)$$

$$= b + (1\text{s compliment of } a) - 1111\ 1111 \quad (2)$$

$$= -\left[1\text{s of compliment of } (b + (1\text{s compliment of } a))\right] \quad (3)$$

Now to calculate $b - a$, using line 1 above

$b + 1$'s compliment of a is 1 0011 0110

and adding one gives 1 0011 0111

and removing the far left 1 gives $0011\ 0111_{(2)} = 55_{(10)}$.

In the case of calculating $a - b$ in which the result of the subtraction turns out to be negative, it is more convenient to use line 3 above.

(c) (2 points) Using 2's complement method, find the value of $a - b$ and $b - a$

The 2's complement method differs from the 1's complement only in respect to when one adds 1 - since forming the 2's complement is best done by forming the 1's complement and then adding one. The calculations would be essentially similar to what occurs in (b)

4. (6 points) For 16 bit binary numbers a where the high order bit is zero, the one's complement is formed by subtracting from: 1111 1111 1111 1111 and the two's complement is formed by subtracting from: 1 0000 0000 0000 0000.

Given the decimal numbers $b = 837$ and $a = 661$

(a) find the 16 bit binary equivalents

$$b = 837_{(10)} = 0000\ 0011\ 0100\ 0101_{(2)}$$
$$a = 661_{(10)} = 0000\ 0010\ 1001\ 0101_{(2)}$$

(b) find the 1's complement of a

$$1111\ 1101\ 0110\ 1010$$

(c) using the 1's complement method, find the difference $b - a$.

b plus 1's complement of a is:

1 0000 0000 1010 1111 and adding 1 gives:

1 0000 0000 1011 0000 and removing the far left 1 gives:

$$0000\ 0000\ 1011\ 0000_{(2)} = 176_{(10)}$$

5. (6 points) Consider the following further example of the 32-bit floating point data type

0 | 1000 0011 | 001 0100 0000 0000 0000 0000 The sign field is 0 so the number is positive, and the exponent field is decimal 131. Since $131 - 127 = 4$, the exponent has value 4. Combining a 1 to the left of the *binary point* (the equivalent of a decimal point) with the fraction field to the right of the binary point gives 1.0010. The exponent 4 indicates that this must now be multiplied by 2^4 . As an answer we then have

$$1.0010100_{(2)} \cdot 2^4 = 1\ 0010.1_{(2)} = 18.5_{(10)}$$

Now - given the 32 bit floating point data type

$$1\ | 1000\ 0010\ | 001\ 0100\ 0000\ 0000\ 0000\ 0000$$

as above find the equivalent decimal number.

Solution: The far left 1 in the *sign* position indicates that the number is negative. The next 8 bits to the right of the sign position is the exponent field and we see that $1000\ 0010 = 130_{10}$. Since $130 - 127 = 3$, the exponent of 2 is 3. Using the fraction part we see that the number is

$$-1.0010\ 0100_{(2)} \times 2^3 = -1001.01_{(2)}$$

The decimal equivalent of this binary number is $-(9 + \frac{1}{2^2}) = -9\frac{1}{4}$.

6. (6 points) Given the base 10 number $9\frac{11}{32}$ find the floating point representation.

As a base 2 number $9 = 1001_2$, and $\frac{11}{32} = \frac{1}{4} + \frac{1}{16} + \frac{1}{32}$ - which in base 2 becomes 0.01011. Thus in base 2 $9\frac{11}{32} = 1001.01011 = 1.00101011 \times 2^3$. As before we have mixed base 2 and base 10 in this expression. Now the fraction portion of this expression is everything to the left of the "decimal point" - i.e. .00101011 and the exponent is +3. This means that the exponent part of the floating point expression must have value 130 - namely in binary 1000 0010. The floating point data type is then

$$0\ |1000\ 0010\ |001\ 0101\ 1000\ 0000\ 0000\ 0000$$