CPSC 121 - NUMBER REPRESENTATION SOLUTIONS

Problem 1. [Number Representation]

(1) Represent 1609_{10} in unsigned binary notation with the minimum number of bits necessary to represent the number.
(2) Represent 11000111_2 in decimal notation Assume this is an unsigned binary number.
(3) Find the 16-bit two's complement of 410_{16}.
(4) Find the hexadecimal number whose 16-bit two's complement is $1011\,0000\,1001\,1111$.
(5) What is the binary representation of the decimal value -15 in 8 bits?
(6) What decimal value is represented by the 8-bit unsigned binary number 11110010?
(7) What decimal value is represented by the 8-bit signed binary number 11110010?

Solution.

(1) $1609_{10} = 1024 + 512 + 64 + 8 + 1 = 2^{10} + 2^9 + 2^6 + 2^3 + 2^0 = 11001001001_2$.

Remark. If it is not obvious how to get the sum of powers of 2, do the following.

...	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]

1609 is odd, so [0] is 1. $\lfloor 1609/2 \rfloor = 804$ is even, so [1] is 0. $\lfloor 804/2 \rfloor = 402$ is even, so [2] is 0. $\lfloor 402/2 \rfloor = 201$ is odd, so [3] is 1. $\lfloor 201/2 \rfloor = 100$ is even, so [4] is 0. $\lfloor 100/2 \rfloor = 50$ is even, so [5] is 0. $\lfloor 50/2 \rfloor = 25$ is odd, so [6] is 1. $\lfloor 25/2 \rfloor = 12$ is even, so [7] is 0. $\lfloor 12/2 \rfloor = 6$ is even, so [8] is 0. $\lfloor 6/2 \rfloor = 3$ is odd, so [9] is 1. $\lfloor 3/2 \rfloor = 1$ is odd, so [10] is 1. $\lfloor 1/2 \rfloor = 0$, so [10] is the most significant bit. We get $1609_{10} = 11001001001_2$.

(2) $11000111_2 = 2^7 + 2^6 + 2^2 + 2^1 + 2^0 = 128 + 64 + 4 + 2 + 1 = 199_{10}$.
(3) $410_{16} = 0100\,0001\,0000_{\,2}$. $0000\,0100\,0001\,0000 \xrightarrow{\text{flip}} 1111\,1011\,1110\,1111 \xrightarrow{\text{add 1}} 1111\,1011\,1111\,0000$.
(4) $1011\,0000\,1001\,1111 \xrightarrow{\text{flip}} 0100\,1111\,0110\,0000 \xrightarrow{\text{add 1}} 0100\,1111\,0110\,0001$. $0100\,1111\,0110\,0001_{\,2} = 4\text{F}61_{16}$.
(5) $|-15| = 15$. $15_{10} = 8 + 4 + 2 + 1 = 1111_2$. $0000\,1111 \xrightarrow{\text{flip}} 1111\,0000 \xrightarrow{\text{add 1}} 1111\,0001$.
(6) $1111\,0010_{\,2} = 2^7 + 2^6 + 2^5 + 2^4 + 2^1 = 128 + 64 + 32 + 16 + 2 = 242_{10}$.
(7) $1111\,0010 \xrightarrow{\text{flip}} 0000\,1101 \xrightarrow{\text{add 1}} 0000\,1110$. $00001110_2 = 2^3 + 2^2 + 2^1 = 8 + 4 + 2 = 14$. $14 \xrightarrow{\text{opposite}} -14$.

Problem 2. Extra Propositional Logic and Circuit Design Problem:

Design a circuit that takes in two 4-bit unsigned numbers and returns true if and only if the two numbers are equal, or, if the two numbers differ by 8 and are both odd. The two input numbers can be represented by x_3, x_2, x_1, x_0 and by y_3, y_2, y_1, y_0, the x_i's representing one number and the y_i's representing the other number.

(1) First design an expression, and ultimately, a small piece of circuitry that takes in two inputs and returns true if and only if both inputs are the same, that is they are both true or they are both false. Hint, begin with the expression $p \leftrightarrow q$ and try applying the logical equivalence rules or utilizing a truth table in order to represent this expression in a concise circuit.

Solution:

$p \leftrightarrow q$	
$\equiv (p \to q) \wedge (q \to p)$	by Defiintion of the biconditional
$\equiv (\sim p \vee q) \wedge (\sim q \vee p)$	by Defiintion of the implication
$\equiv \sim (\sim (\sim p \vee q) \vee \sim (\sim q \vee p))$	by De Morgan's Law
$\equiv \sim ((p \wedge \sim q) \vee (q \wedge \sim p))$	by De Morgan's Law
$\equiv \sim (p \oplus q)$	Definition of XOR

Therefore, $(p \leftrightarrow q) \equiv \sim (p \oplus q)$. Therefore, we can construct the circuitry for the conditions that require two bits to have the same value (for both bits to be true or for both bits to be false) by using an XNOR gate between the two bits that are being compared.

(2) Now construct the circuit that solves the problem. You may find it helpful to use the result from part one when you need to compare whether the two bits are the same. You may find it helpful to represent the circuit as $c_1 \vee c_2$, where c_1 corresponds to the first condition, both numbers are equal, and c_2 corresponds to the second condition where both numbers are odd and differ by 8.

Solution:

First, we can split the problem into two cases, the first is where the input numbers are equal, and the second is where the numbers differ by 8 and are both odd.

Considering the first case where the input numbers are equal, we can return a true or false answer to this problem by comparing each x_i and y_i pair and determining whether they are both false or both true. Applying the result to part 1 we can represent c_1 as:

$$c_1 \equiv \sim (x_3 \oplus y_3) \wedge \sim (x_2 \oplus y_2) \wedge \sim (x_1 \oplus y_1) \wedge \sim (x_0 \oplus y_0)$$

Now we can consider the second case, where the input numbers differ by 8 and are both odd. Firstly, the two numbers will be odd if and only if both x_0 and y_0 are true (or 1). Therefore, c_2 will include the necessary condition $(x_0 \wedge y_0)$. If the two input numbers were to be equal, and odd, we could simply apply part 1 to the x_i and y_i pairs for $i = 1, 2$ and 3. To account for the condition where the numbers differ by 8, we can notice that when the 3^{rd} (counting from zero) bit of an unsigned number is changed from 0 to 1, then the value of the number increases by 8. We can thus use the expression $x_3 \oplus y_3$ to determine if the 3^{rd} bits of the two numbers differ. Since we want the numbers to strictly differ by 8 we will still require that x_2 and y_2, and, x_1 and y_1, are respectively equal. We can thus represent c_2 as:

$$c_2 \equiv \ (x_3 \oplus y_3) \wedge \sim (x_2 \oplus y_2) \wedge \sim (x_1 \oplus y_1) \wedge (x_0 \wedge y_0)$$

To complete the circuit, we simply combine conditions c_1 and c_2 with an OR gate: $c_1 \vee c_2$.