Jovan Shandro

Homework 4

Problem 4.1

Solution:

- Since the output should be X=1, in the AND gate we need all the three inputs 1. Therefore, C has to be 1.

- Then the XNOR gate needs to be 1, so there is $\overline{B \oplus C} = \overline{B \oplus 1} = 1$. Directly, we find B=1.

- In the same way we find for the XOR gate: $A \oplus B = A \oplus 1 = 1$. Therefore A=0 and the final result for the input condition is $C = 1, B = 1, A = 0$.

Problem 4.2

Solution:
(a) For the given gate, the truth table would be:

A	B	C	Y
0	0	0	1
0	0	1	1
1	0	0	1
1	0	1	0
0	1	0	0
0	1	1	1
1	1	0	0
1	1	1	1

(b) Considering the above table and simplifying the expression, the final result is:

$\overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C + A \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot B \cdot C + A \cdot B \cdot C$
$= \overline{A} \cdot \overline{B} \cdot (\overline{C} + C) + A \cdot \overline{B} \cdot \overline{C} + BC \cdot (\overline{A} + A)$
$\boldsymbol{= \overline{A} \cdot \overline{B} + A \cdot \overline{B} \cdot \overline{C} + B \cdot C}$

Problem 4.3

Solution:

(a) Unsigned number: 27. We find $27_{10} \rightarrow$ binary:

$$27/2 = 13(1) \rightarrow 13/2 = 6(1) \rightarrow 6/2 = 3(0) \rightarrow 3/2 = 1(1) \rightarrow 1/2 = 0(1)$$

Binary representation: 11011_2

Since we need to have an 8-bit representation, we add leading zeros until we complete 7 bits and then add then add a leading bit to show the sign bit, which in this case is 0 for positive: $\mathbf{00011011_2}$

(b) Unsigned number: 66. We find $66_{10} \rightarrow$ binary:

66/2 =33 (0)$\rightarrow$ $33/2 = 16(1) \rightarrow 16/2 = 8(0) \rightarrow 8/2 = 4(0) \rightarrow 4/2 = 2(0) \rightarrow 2/2 = 1(0) \rightarrow 1/2 = 0(1)$

Binary representation: 1000010_2

The number lacks only one bit, which will be the sign bit (0 since it's positive):

$$\mathbf{01000010_2}$$

(c) Unsigned number: 18. We find $18_{10} \rightarrow$ binary:

$18/2 = 9(0) \rightarrow\ 9/2 = 4(1) \rightarrow\ 4/2 = 2(0) \rightarrow\ 2/2 = 1(0) \rightarrow\ 1/2 = 0(1)$

Binary representation: 10010_2
In the case of negative numbers, we do the following: complete the remaining bits with leading zeros, invert the number and the add 1:
$\rightarrow 00010010_2$
$\rightarrow 11101101_2$
$\rightarrow 11101101_2 + 1 = \mathbf{11101110_2}$

(d) Unsigned number: 127. We find: $127_{10} \rightarrow$ binary:

$$127/2 = 63(1) \rightarrow\ 63/2 = 31(1) \rightarrow\ 31/2 = 15(1) \rightarrow\ 15/2 = 7(1) \rightarrow$$
$$7/2 = 3(1) \rightarrow\ 3/2 = 1(1) \rightarrow\ 1/2 = 0(1)$$

Binary representation: 1111111_2
After adding the sign bit (0 since it's positive): $\mathbf{01111111_2}$

(e) We already found the binary representation of positive 127 in d: 01111111_2

After inverting the bits $\rightarrow 10000000_2$
We add 1 $\rightarrow 10000000_2 + 1 = \mathbf{10000001_2}$

(f) Unsigned number $128_{10} \rightarrow$ binary:

$$128/2 = 64(0) \rightarrow\ 64/2 = 32(0) \rightarrow\ 32/2 = 16(0) \rightarrow\ 16/2 = 8(0) \rightarrow$$
$$8/2 = 4(0) \rightarrow\ 4/2 = 2(0) \rightarrow\ 2/2 = 1(0) \rightarrow\ 1/2 = 0(1)$$

Binary representation: 10000000_2
$\rightarrow$ Invert the number:01111111_2
$\rightarrow 01111111_2 + 1 = \mathbf{10000000_2}$
We see that we get the same result for -128 as for 128. This happens because 128 cannot be expressed in 8 bits as the largest number that can be expressed using 8 bits is 011111111 (127). The smallest one is 10000000, which is the one we want, -128.

(g)

$$131_{10} \rightarrow \text{ binary}$$
$$131/2 = 65(1) \rightarrow\ 65/2 = 32(1) \rightarrow\ 32/2 = 16(0) \rightarrow\ 16/2 = 8(0) \rightarrow$$
$$8/2 = 4(0) \rightarrow\ 4/2 = 2(0) \rightarrow\ 2/2 = 1(0) \rightarrow\ 1/2 = 0(1)$$

Binary representation: 10000011_2

The binary number we got has 8 bits, so there is no place where we can add a leading sign bit. Therefore 131 cannot be represented in the 8 bit format as we're asked to.

(h)We first find the binary representation of 7, then invert the digits and at last add 1:

$$7/2 = 3(1) \rightarrow\ 3/2 = 1(1) \rightarrow\ 1/2 = 0(1)$$

Binary representation: 111_2
$\rightarrow 00000111_2$
$\rightarrow 11111000_2$
$\rightarrow 11111000 + 1 = \mathbf{11111001_2}$

Problem 4.4
Solution:

If the number is positive we will perform normal conversion method. If the number is negative, we will subtract 1 from the binary number we have, invert the bits, and then perform normal conversion again. We check the sign bit to see whether the number is negative or positive:

(a) $00011000_2 \longrightarrow 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 =$
$= 16 + 8 = +24$

(b) $\rightarrow 11110101 - 1 = 11110100$
$\rightarrow 00001011$
$\rightarrow 00001011_2 \longrightarrow 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 =$
$= 1 + 2 + 8 = 11$
The number is -11_{10}

(c) $01011011_2 \longrightarrow 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 =$
$= 64 + 16 + 8 + 2 + 1 = +91$

(d) $\rightarrow 10110110 - 1 = 10110101$
$\rightarrow 01001010$
$\rightarrow 01001010_2 \longrightarrow 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 =$
$= 64 + 8 + 2 = 74$
The number is -74_{10}

(e) $\rightarrow 11111111 - 1 = 11111110$
$\rightarrow 00000001$
$\rightarrow 00000001_2 \longrightarrow 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1$
The number is -1_{10}

(f) $01101111_2 \longrightarrow 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 =$
$= 64 + 32 + 8 + 4 + 2 + 1 = +111$

(g) $\rightarrow 10000001 - 1 = 10000000$
$\rightarrow 01111111$
$\rightarrow 01111111_2 \longrightarrow 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 =$
$= 64 + 32 + 16 + 8 + 4 + 2 + 1 = 127$
The number is -127_{10}

(h) $\rightarrow 10000000 - 1 = 01111111$
$\rightarrow 10000000$
$\rightarrow 10000000_2 \rightarrow 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 128$
The number is -128_{10}

Problem 4.5
Solution:
(a) We first convert 27 and 36 to BCD and then add them

$27_{10} = 00100111_{\text{BCD}}$; $\quad 36_{10} = 00110110_{\text{BCD}}$

	00100111	
+	00110110	
	01011101	←Invalid code groups, add 6
+	****0110	
	01100011	

$01100011_{\text{BCD}} = 63_{10}$

(b) We first convert 73 and 29 to BCD and then add them

$73_{10} = 01110011_{\text{BCD}}$; $\quad 29_{10} = 00101001_{\text{BCD}}$

$$\begin{array}{rrl}
 & 01110011 & \\
+ & 00101001 & \\
\hline
 & 10011100 & \leftarrow\text{Invalid code groups, add 6 twice} \\
+ & 01100110 & \\
\hline
 & 000100000010 &
\end{array}$$

$000100000010_{\text{BCD}} = 102_{10}$

Problem 4.6
Solution:

Unsigned numbers written in n-bit representation are in the range from 0 up to $2^n - 1$. Considering this, we have:

(a) For 8-bit representation the range is: $0 \rightarrow 2^8 - 1 = 255_{10}$.

(b) For positive numbers, considering the fact that the first bit is the sign bit, considering the 7 remaining bits, the range is $0 \rightarrow 2^7 - 1 = 127_{10}$. For the negative numbers, we're supposed to have the same amount of numbers as in the positive side, so instead of zero we will add another number in the end, so the range is $-1 \rightarrow -128$. Overall range: $-128 \rightarrow 127$.

(c) Same as in a, the range is: $0 \rightarrow 2^{11} - 1 = 2048 - 1 = 2047$.

(d) Same as in point b, the range is: $-2^{10} \rightarrow 2^{10} - 1$ or $-1024 \rightarrow 1023$.

(e) The range is: $-2^{15} \rightarrow 2^{15} - 1$ or $-32768 \rightarrow 32767$.