Homework 4

Problem 4.1

Solution:

The boolean expression of the circuit is: X = (A XOR B) $\wedge$ (B XNOR C) $\wedge$ C

A	B	C	A XOR B	B XNOR C	X
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	1	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	1	0	1	0

As it is evident from the truth table, X = 1 if we have the following input condition: $\overline{\mathbf{A}} \wedge \mathbf{B} \wedge \mathbf{C}$

Problem 4.2

Solution:

Boolean Expression from the circuit: $\overline{AB + C}$ XOR $(\overline{A} * (B + C))$

a) Truth Table:

A	B	C	$\overline{AB + C}$	$(\overline{A} * (B + C))$	X
0	0	0	1	0	1
0	0	1	0	1	1
0	1	0	1	1	0
0	1	1	0	1	1
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	0	0

b) Sum of products: $\overline{\mathbf{ABC}} + \overline{\mathbf{AB}}\mathbf{C} + \overline{\mathbf{A}}\mathbf{BC} + \mathbf{A}\overline{\mathbf{BC}}$

Problem 4.3

Solution:

a) $+27_{10} = \mathbf{00011011}$

27	/ 2	result	13	remainder	**1**
13	/ 2	result	6	remainder	**1**
6	/ 2	result	3	remainder	**0**
3	/ 2	result	1	remainder	**1**
1	/ 2	result	0	remainder	**1**

b) $+66_{10}$ = **01000010**

66	/ 2	result	33	remainder	**0**
33	/ 2	result	16	remainder	**1**
16	/ 2	result	8	remainder	**0**
8	/ 2	result	4	remainder	**0**
4	/ 2	result	2	remainder	**0**
2	/ 2	result	1	remainder	**0**
1	/ 2	result	0	remainder	**1**

c) -18_{10} = **11101110**

Convert the positive version of the number to a binary representation: $18_{10} = 00010010_2$

18	/ 2	result	9	remainder	**0**
9	/ 2	result	4	remainder	**1**
4	/ 2	result	2	remainder	**0**
2	/ 2	result	1	remainder	**0**
1	/ 2	result	0	remainder	**1**

Flip the bits: $!(00010010) = 11101101$

Add 1: $-18_{10} = 11101101 + 00000001 =$ **11101110**

d) 127_{10} = **01111111**

127	/ 2	result	63	remainder	**1**
63	/ 2	result	31	remainder	**1**
31	/ 2	result	15	remainder	**1**
15	/ 2	result	7	remainder	**1**
7	/ 2	result	3	remainder	**1**
3	/ 2	result	1	remainder	**1**
1	/ 2	result	0	remainder	**1**

e) $-127_{10} = 10000001$

Convert the positive version of the number to a binary representation: $127_{10} = 01111111_2$

127	/ 2	result	63	remainder	**1**
63	/ 2	result	31	remainder	**1**
31	/ 2	result	15	remainder	**1**
15	/ 2	result	7	remainder	**1**
7	/ 2	result	3	remainder	**1**
3	/ 2	result	1	remainder	**1**
1	/ 2	result	0	remainder	**1**

Flip the bits: $!(01111111) = 10000000$

Add 1: $-127_{10} = 10000000 + 00000001 =$ **10000001**

f) $-128_{10} = 10000000$

Convert the positive version of the number to a binary representation: $128_{10} = 10000000_2$

128	/ 2	result	64	remainder	**0**
64	/ 2	result	32	remainder	**0**
32	/ 2	result	16	remainder	**0**
16	/ 2	result	8	remainder	**0**
8	/ 2	result	4	remainder	**0**
4	/ 2	result	2	remainder	**0**
2	/ 2	result	1	remainder	**0**
1	/ 2	result	0	remainder	**1**

Flip the bits: $!(10000000) = 01111111$

Add 1: $-128_{10} = 01111111 + 00000001 = \mathbf{10000000}$

g) $+131_{10} = \mathbf{10000011}$ As it is evident, the number is positive and after converting by using only 8 bits we get a binary number where the most significant bit is 1, meaning that it is negative. Thus, in this case we have an overflow.

131	/ 2	result	65	remainder	**1**
65	/ 2	result	32	remainder	**1**
32	/ 2	result	16	remainder	**0**
16	/ 2	result	8	remainder	**0**
8	/ 2	result	4	remainder	**0**
4	/ 2	result	2	remainder	**0**
2	/ 2	result	1	remainder	**0**
1	/ 2	result	0	remainder	**1**

h) $-7_{10} = 11111001$

Convert the positive version of the number to a binary representation: $7_{10} = 00000111_2$

7	/ 2	result	3	remainder	**1**
3	/ 2	result	1	remainder	**1**
1	/ 2	result	0	remainder	**1**

Flip the bits: $!(00000111) = 11111000$

Add 1: $-7_{10} = 11111000 + 00000001 = \mathbf{11111001}$

Problem 4.4

Solution:

a) $00011000 = 24_{10}$

The number is positive, only the conversion to decimal is needed:

$$00011000 = (0*2^7) + (0*2^6) + (0*2^5) + (1*2^4) + (1*2^3) + (0*2^2) + (0*2^1) + (0*2^0) = \mathbf{24_{10}}$$

b) $11110101 = -11_{10}$

Value is negative because it starts with 1.

Flip the bits: $!(11110101) = 00001010$

Add 1: $00001010 + 1 = 00001011$

Convert to decimal and add the minus: $00001011 = (1*2^3)+(0*2^2)+(1*2^1)+(1*2^0) = \mathbf{-11_{10}}$

c) $01011011 = 91_{10}$

The number is positive, only the conversion to decimal is needed:

$01011011 = (0*2^7)+(1*2^6)+(0*2^5)+(1*2^4)+(1*2^3)+(0*2^2)+(1*2^1)+(1*2^0) = \mathbf{91_{10}}$

d) $10110110 = -74_{10}$

Value is negative because it starts with 1.

Flip the bits: $!(10110110) = 01001001$

Add 1: $01001001 + 1 = 01001010$

Convert to decimal and add the minus: $01001010 = (0*2^7)+(1*2^6)+(0*2^5)+(0*2^4)+(1*2^3)+(0*2^2)+(1*2^1)+(0*2^0) = \mathbf{-74_{10}}$

e) $11111111 = -1_{10}$

Value is negative because it starts with 1.

Flip the bits: $!(11111111) = 00000000$

Add 1: $00000000 + 1 = 00000001$

Convert to decimal and add the minus: $00000001 = (1*2^0) = \mathbf{-1_{10}}$

f) $01101111 = 111_{10}$

The number is positive, only the conversion to decimal is needed:

$01101111 = (0*2^7)+(1*2^6)+(1*2^5)+(0*2^4)+(1*2^3)+(1*2^2)+(1*2^1)+(1*2^0) = \mathbf{111_{10}}$

g) $10000001 = -127_{10}$

Value is negative because it starts with 1.

Flip the bits: $!(10000001) = 01111110$

Add 1: $01111110 + 1 = 01111111$

Convert to decimal and add the minus: $01111111 = (0*2^7)+(1*2^6)+(1*2^5)+(1*2^4)+(1*2^3)+(1*2^2)+(1*2^1)+(1*2^0) = \mathbf{-127_{10}}$

h) $10000000 = -128_{10}$

Value is negative because it starts with 1.

Flip the bits: $!(10000000) = 01111111$

Add 1: $01111111 + 1 = 10000000$

Convert to decimal and add the minus: $10000000 = (1 * 2^7) = \mathbf{-128_{10}}$

Problem 4.5

Solution:

For any addition greater than 9_{10}, add binary equivalent of 6 to the Binary sum obtained and get its BCD equivalent (the comma operator distinguishes the digits):

$27 + 36 = 0010, 0111_{BCD} + 0011, 0110_{BCD} = 0101, 1101 + 0000, 0110 = \mathbf{0110, 0011_{BCD}} = 63_{10}$

$73 + 29 = 0111, 0011_{BCD} + 0010, 1001_{BCD} = 1001, 1100 + 0000, 0110 = 1010, 0010 + 0110, 0000 = \mathbf{0001, 0000, 0010_{BCD}} = 102_{10}$

Problem 4.6

Solution:

a) What is the range of unsigned decimal numbers that can be represented by using 8 bits?

The range is 0 to 255 because the biggest number can be expressed as $11111111_2 = 255_{10}$ and the smallest one is $00000000_2 = 0_{10}$. In general, an m-bit unsigned number represents all numbers in the range 0 to $2^m - 1$.

b) What is the range of signed decimal numbers that can be represented by using 8 bits (including the sign bit)?

The simplest case is to use 1 sign bit and 7 value bits. The range is -127 to 127 because the biggest number can be expressed as $01111111_2 = 127_{10}$ and the smallest one is $11111111_2 = -127_{10}$ considering that the most significant bit is the sign bit.

c) What is the range of unsigned decimal numbers that can be represented by using 11 bits?

The range is 0 to 2047 because the biggest number can be expressed as $11111111111_2 = 2047_{10}$ and the smallest one is $00000000_2 = 0_{10}$. In general, an m-bit unsigned number represents all numbers in the range 0 to $2^m - 1$.

d) What is the range of signed decimal numbers that can be represented by using 11 bits (including the sign bit)?

The simplest case is to use 1 sign bit and 10 value bits. The range is -1023 to 1023 because the biggest number can be expressed as $01111111111_2 = 1023_{10}$ and the smallest one is $11111111111_2 = -1023_{10}$ considering that the most significant bit is the sign bit.

e) What is the range of signed decimal numbers that can be represented by using 16 bits (including the sign bit)?

The simplest case is to use 1 sign bit and 15 value bits. The range is -32767 to 32767 because the biggest number can be expressed as $0111111111111111_2 = 32767_{10}$ and the smallest one is $1111111111111111_2 = -32767_{10}$ considering that the most significant bit is the sign bit.