COE 202, Term 162
Digital Logic Design

HW# 4 Solution

Q.1. Obtain the 1`s and 2`s complement of the following binary numbers: 01100, 00001,00000.

Number	1's complement	2's Complement
01100	10011	10100
00001	11110	11111
00000	11111	00000

Q.2. Find the 10`s complement of $(935)_{11}$.

10's complement of $(935)_{11}$ = 175

$$\begin{array}{rccc} & 10 & 10 & 10 \\ - & 9 & 3 & 5 \\ \hline & 1 & 7 & 5 \end{array}$$

Q.3. Show how the decimal integer -120 would be represented in 2`s complement notation using 8 bits and 16 bits, respectively.

−120

we represent +120 usi g 8-bits

7	6	5	4	3	2	1	0
0	1	1	1	1	0	0	0

2's complement is 1000 1000

−120 represented in 16 bits will be just a sign extension of 8-bit representation

1111 1111 1000 1000

Q.4. Perform subtraction with the following binary numbers using 2`s complement and 1`s complement, assuming that numbers are represented in 6 bits. Check the answer by straight subtraction:

(i) 11010 - 1101

(ii) 11010 - 10000

(iii) 10010 – 10011

(i) 11010 − 1101

1's complement

$$\begin{array}{rr} & 011010 \\ + & 110010 \\ \hline 1 \leftarrow & 001100 \\ + & 1 \end{array} \leftarrow \text{we add the carry out}$$

$$\begin{array}{rr} & 001101 \end{array}$$

straight subtraction

$$\begin{array}{rr} & 011010 \\ - & 1101 \\ \hline & 001101 \end{array}$$

2's complement

$$\begin{array}{rr} & 011010 \\ + & 110011 \\ \hline 1 \leftarrow & 001101 \end{array}$$

(ii) 11010 − 10000

1's complement

$$\begin{array}{rr} & 011010 \\ + & 101111 \\ \hline 1 \leftarrow & 001001 \\ + & 1 \end{array} \leftarrow \text{we add the carry out}$$

$$\begin{array}{rr} & 001010 \end{array}$$

straight subtraction

$$\begin{array}{rr} & 011010 \\ - & 010000 \\ \hline & 001010 \end{array}$$

2's complement

$$\begin{array}{rr} & 011010 \\ + & 110000 \\ \hline 1 \leftarrow & 001010 \end{array}$$

(iii) 10010 − 10011

1's complement

$$\begin{array}{r} 010010 \\ +\ 101100 \\ \hline 111110 \end{array}$$

This represents −1

straight subtraction

$$\begin{array}{r} 010010 \\ -\ 010011 \\ \hline 111111 \end{array}$$

Note here that there is a borrow

2's complement

$$\begin{array}{r} 010010 \\ +\ 101101 \\ \hline 111111 \end{array}$$

This represents −1

Q.5. A microcontroller uses 8-bit registers. Give the following in both binary and decimal:

(i) The maximum unsigned number that can be stored.

(i) maximum unsigned number

$2^8 - 1 = 255$ 1111 1111

(ii) The smallest (negative) number and the largest (positive) number that can be stored using the sign-magnitude notation.

(ii) sign-magnitude

smallest negative number

$-(2^7 - 1) = -127$ 1111 1111

largest positive number

$2^7 - 1 = +127$ 0111 1111

(iii) The smallest (negative) number and the largest (positive) number that can be stored using the 2`s complement notation.

(iii) 2's complement

smallest negative number

$-2^{8-1} = -2^{7} = -128$ 1000 0000

largest positive number

$+2^{7}-1 = 127$ 0111 1111

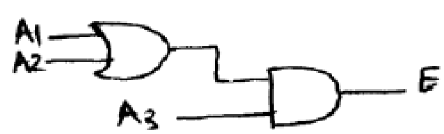
Q.6. Design a combinational circuit that detects an error in the representation of a decimal digit in BCD. In other words, obtain a logic diagram whose output is equal to 1 when the inputs contain any one of the six unused bit combinations in the BCD code.

The output is 1 if the BCD code is in the range 1010 – 1111.

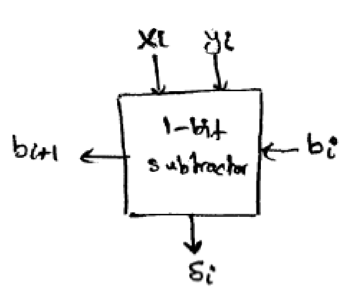
$A_3A_2 \backslash A_1A_0$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

$$E = A_3A_2 + A_3A_1$$

$$= A_3(A_2 + A_1)$$



Q.7. It is required to design a 4-bit ripple-borrow subtractor to find the subtraction X-Y for the two unsigned numbers, $X=X_3\text{-}X_0$, and $Y=Y_3\text{-}Y_0$. Design a 1-bit full subtractor and show how it can be used to construct the 4-bit subtractor.



b_i	x_i	y_i	b_{i+1}	s_i
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

b_i \ $x_i y_i$	00	01	11	10
0	0	1	0	0
1	1	1	1	0

$$b_{i+1} = \overline{x_i}\, y_i + b_i \overline{x_i} + b_i y_i$$

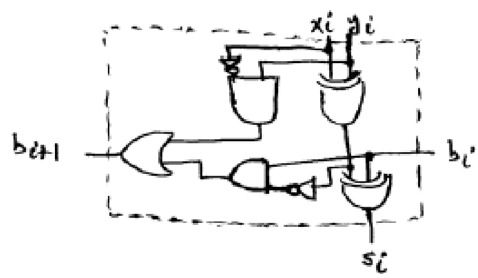
$$= \overline{x_i}\, y_i + b_i (y_i + \overline{x_i})$$

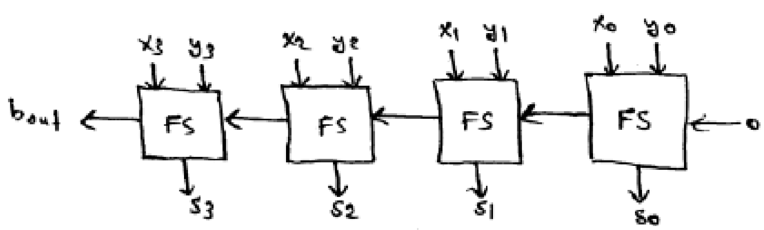
$$= \overline{x_i}\, y_i + b_i (y_i \oplus \overline{x_i})$$

$$= \overline{x_i}\, y_i + b_i (y_i \oplus x_i)'$$

b_i \ $x_i y_i$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$s_i = x_i \oplus y_i \oplus b_i$$





Q.8. Design two simplified combinational circuits that generate the 9's complement of (a) a BCD digit and (b) an excess-3 digit. Then compare the gate and literal count of the two circuits. Assume in both cases that input combinations not corresponding to decimal digits give don't care outputs.

9's complement

(a) BCD digit

Digit	x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	1	0	0	1
1	0	0	0	1	1	0	0	0
2	0	0	1	0	0	1	1	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	0	0
6	0	1	1	0	0	0	1	1
7	0	1	1	1	0	0	1	0
8	1	0	0	0	0	0	0	1
9	1	0	0	0	0	0	0	0

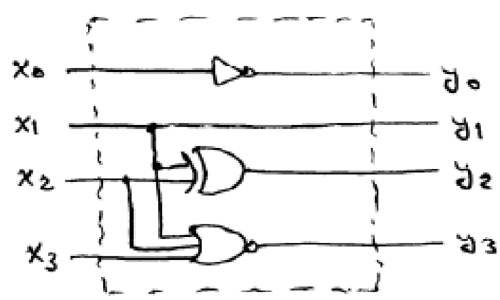
$y_0 = \overline{x_0}$, $y_1 = x_1$

$x_3x_2 \backslash x_1x_0$	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	x	x	x	x
10	0	0	x	x

$x_3x_2 \backslash x_1x_0$	00	01	11	10
00	1	1	0	0
01	0	0	0	0
11	x	x	x	x
10	0	0	x	x

$$y_2 = x_2\overline{x_1} + \overline{x_2}x_1$$
$$= x_2 \oplus x_1$$

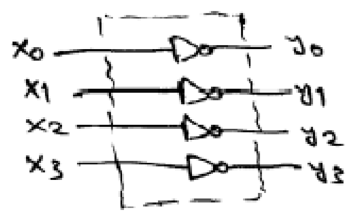
$$y_3 = \overline{x_3}\,\overline{x_2}\,\overline{x_1}$$



(b) Excess-3 digit

Digit	x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	1	1	1	1	0	0
1	0	1	0	0	1	0	1	1
2	0	1	0	1	1	0	1	0
3	0	1	1	0	1	0	0	1
4	0	1	1	1	1	0	0	0
5	1	0	0	0	0	1	1	1
6	1	0	0	1	0	1	1	0
7	1	0	1	0	0	1	0	1
8	1	0	1	1	0	1	0	0
9	1	1	0	0	0	0	1	1

$y_0 = \overline{x_0}$, $y_1 = \overline{x_1}$, $y_2 = \overline{x_2}$, $y_3 = \overline{x_3}$



(c) we can see that for the excess-3 code, the 9's complement circuit has 4 literals and 4 inverter gates. However, for the BCD code, the 9's complement circuit has 9 literal and one inverter, one NOR, and one XOR gate. This is the advantage of using the excess-3 code as the 9's complement is obtained by finding the 1's complement of the code.

Q.9. Construct a BDC adder-subtractor using a BCD adder and the 9’s complement designed in Q3, as well as other logic or functional blocks as necessary. Use block diagrams for the components, showing only inputs and outputs where possible.

BCD Adder - Subtractor