

27th October, 2016**Home Assignment – 12**

1. In a certain chemical process control application, a particular relay must operate for certain input conditions. There are three input variables A, B and C, where A is the MSB and C is the LSB. The requirements are as follows:
 Relay must be ON for 010, 011 and 100. Relay must be OFF for 000, 001 and 110. The remaining input condition will never occur.
 (a) Obtain a minimum expression for the above relay function using K-map.
 (b) Realize the above expression using NAND gates only.
2. A 4-input NAND gate is defined by the function $f(w, x, y, z) = \overline{w.x.y.z}$. Design a 4-input NAND gate using 2-input NAND gates only.

3. A combinational circuit has 3 outputs $F1, F2$ and $F3$ as follows:

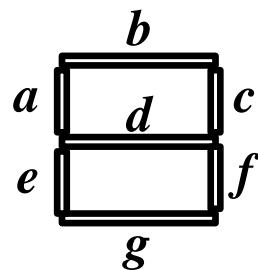
$$F1 = \overline{x}.\overline{y}.\overline{z} + x.z$$

$$F2 = \overline{x}.\overline{y}.\overline{z} + \overline{x}.y$$

$$F3 = \overline{x}.\overline{y}.z + x.y$$

Design the circuit with decoder and external gates.

4. BCD (binary coded decimal) is a representation of decimal digits using four bits. The decimal digit can have values within 0 to 9. The four bit binary number represents a decimal number N , where $N = 2^3b_3 + 2^2b_2 + 2^1b_1 + 2^0b_0$, and $b_3b_2b_1b_0$ is binary number having four bits. We want to use a seven-segment display as shown below to display the decimal number corresponding to 4-bit BCD input. Each segment of display is an LED which glows when logic is 1. Determine the Boolean expression representing logic for each element of display. Minimize the expression using K-map.



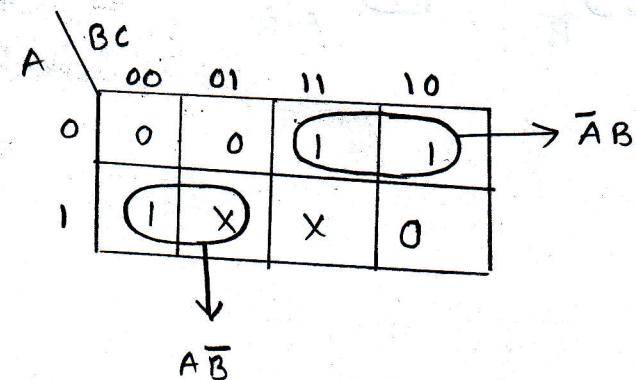
5. The four bit binary number represents a decimal number A , where $A = 2^3a_3 + 2^2a_2 + 2^1a_1 + 2^0a_0$, and $a_3a_2a_1a_0$ is binary number having four bits. Design a circuit using 16 to 1 MUX to check whether A is prime number or not. Recall that a number A is prime number if it is a natural number greater than 1 that has no other divisors except 1 and the number itself.

6. Construct an 8-to-1 MUX using minimum number of 4-to-1 and 2-to-1 MUXs as building blocks.
7. Design the circuit using an 8×1 Multiplexer to implement the logic function $f(A, B, C, D) = \sum m(2, 5, 6, 8, 9, 10, 11, 13, 14)$.
8. Implement half-adder using 2-input NOR gates only.
9. Implement full adder using two 4×1 Multiplexers.

By assuming bit '1' when the relay is ON and bit '0' when the relay is OFF; we got the following truth table for the given conditions.

Input			Output (Y)
A	B	C	
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	X
1	1	0	0
1	1	1	X

a) Using K-map the minimized expression is,



$$\therefore Y = A\bar{B} + \bar{A}B$$

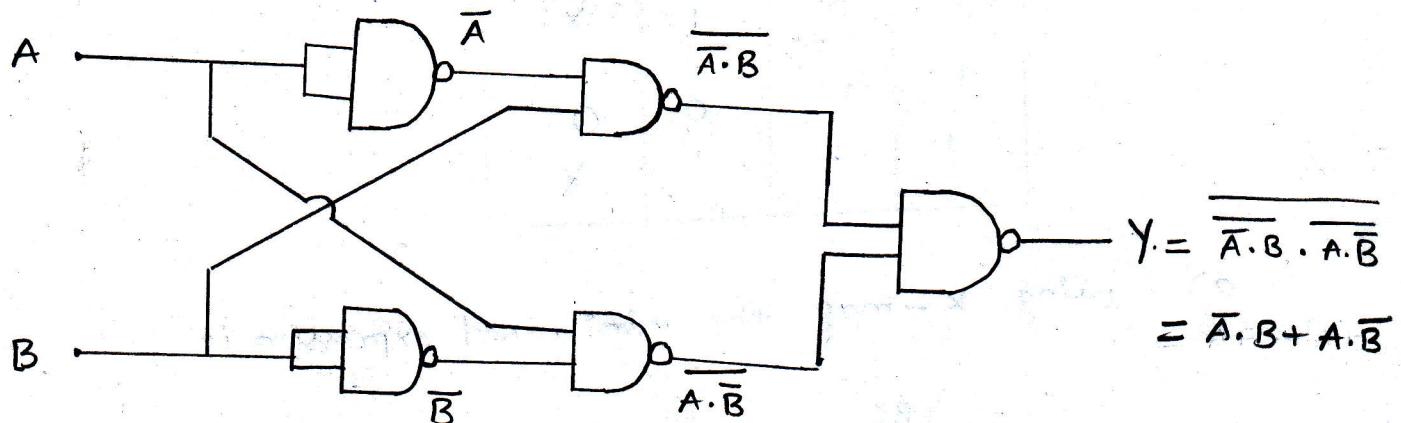
- b) The expression, using NAND gates, can be realized as follows.

$$\begin{aligned} Y &= \overline{A \cdot B} + A \cdot \overline{B} = \overline{\overline{A \cdot B}} + \overline{A \cdot \overline{B}} \\ &= \overline{\overline{A \cdot B} \cdot A \cdot \overline{B}} \end{aligned}$$

this implies, we need to generate:

- i) $\overline{A}, \overline{B}$
- ii) $\overline{A \cdot B}, A \cdot \overline{B}$
- iii) and then $\overline{\overline{A \cdot B} \cdot A \cdot \overline{B}}$

Now the realization is,



given,

$$f(w, x, y, z) = \overline{w \cdot x \cdot y \cdot z}$$

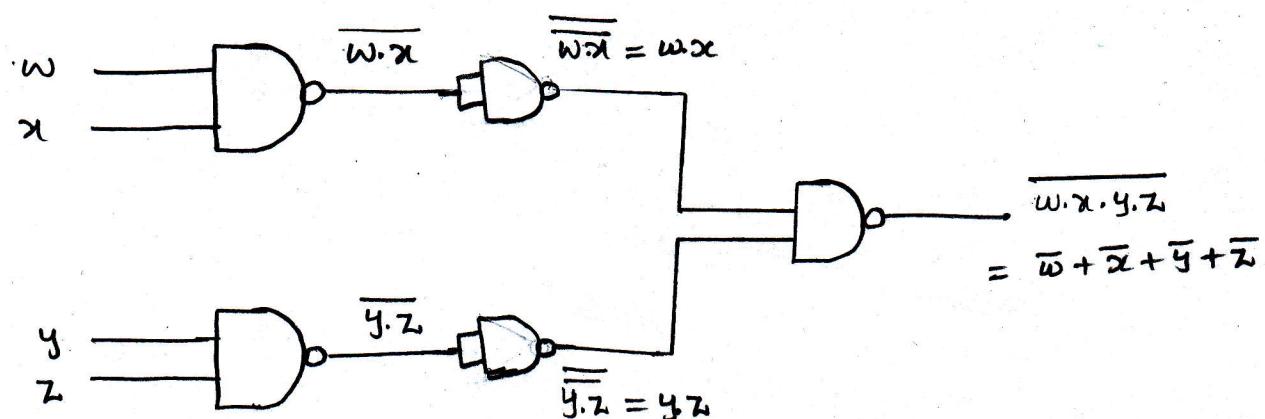
To design 4-input NAND gate using 2-input NAND gates and NOT gates,

$$\begin{aligned} f(w, x, y, z) &= \overline{\overline{w \cdot x} + \overline{y \cdot z}} \\ &= \overline{\overline{w} \cdot \overline{x} \cdot \overline{y} \cdot \overline{z}} \\ &= \overline{\overline{w} \cdot \overline{x}} \cdot \overline{\overline{y} \cdot \overline{z}} \end{aligned}$$

we have to generate,

- i) $\overline{w \cdot x}$ and $\overline{y \cdot z}$ then
- ii) $\overline{\overline{w} \cdot \overline{x}}$ and $\overline{\overline{y} \cdot \overline{z}}$
- iii) and finally $\overline{\overline{w} \cdot \overline{x}} \cdot \overline{\overline{y} \cdot \overline{z}}$

The final realization is,



given

$$F_1 = \bar{x} \cdot \bar{y} \cdot \bar{z} + x \cdot z$$

$$F_2 = x \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y$$

$$F_3 = \bar{x} \cdot \bar{y} \cdot z + x \cdot y$$

Let us represent each function in canonical sum of products form.

$$F_1 = \bar{x} \cdot \bar{y} \cdot \bar{z} + x \cdot z \cdot (y + \bar{y})$$

$$= \bar{x} \cdot \bar{y} \cdot \bar{z} + x \cdot \bar{y} \cdot z + x \cdot y \cdot z = \sum m(0, 5, 7)$$

$$F_2 = x \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y \cdot (z + \bar{z})$$

$$= x \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y \cdot z + \bar{x} \cdot y \cdot \bar{z} = \sum m(4, 3, 2)$$

$$F_3 = \bar{x} \cdot \bar{y} \cdot z + x \cdot y \cdot (z + \bar{z})$$

$$= \bar{x} \cdot \bar{y} \cdot z + x \cdot y \cdot z + x \cdot y \cdot \bar{z} = \sum m(1, 7, 6)$$

x	y	z	F_1	F_2	F_3
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	0	1	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	1	0	1

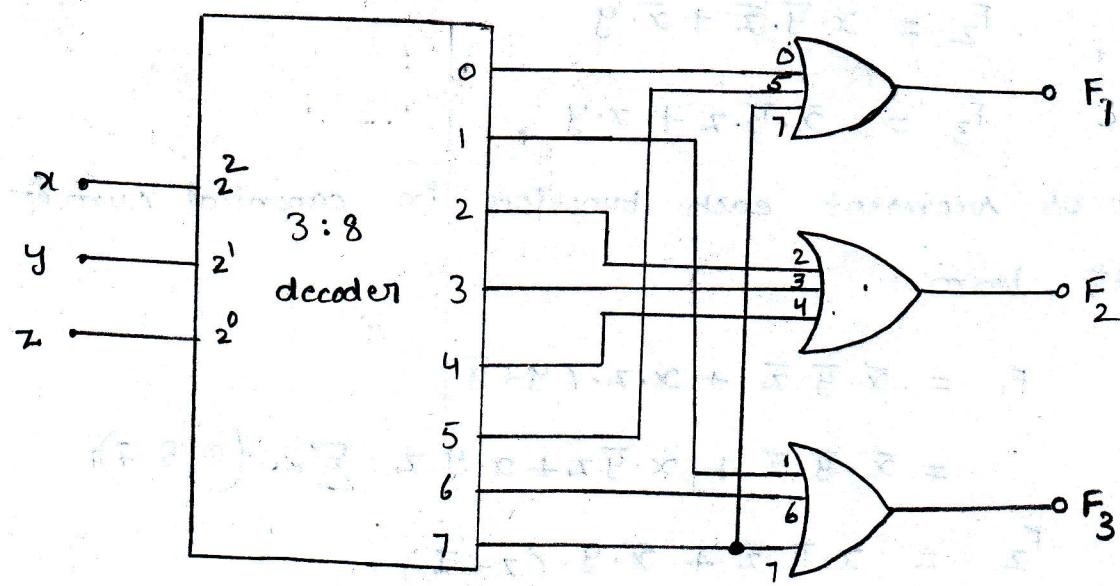
$$\Rightarrow F_1 = \sum m(0, 5, 7)$$

$$F_2 = \sum m(2, 3, 4)$$

$$F_3 = \sum m(1, 6, 7)$$

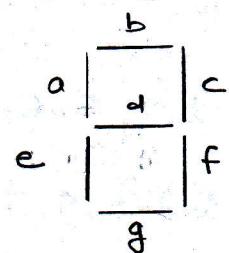
Now the binal design becomes,

P-5



SOLN. ④

P-6



b_3	b_2	b_1	b_0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	0	1	1
0	0	0	1	1	0	0	1	0	0	1
0	0	1	0	1	0	1	1	1	1	0
0	0	1	1	1	0	1	1	1	0	1
0	1	0	0	1	1	0	1	1	0	1
0	1	0	1	1	1	0	1	0	1	1
0	1	1	0	1	1	0	1	1	1	1
0	1	1	1	1	0	1	1	0	0	1
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	1	0	1
1	0	1	0	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x

x = don't care

$$a = \sum m(0, 4, 5, 6, 8, 9) + d(10, 11, 12, 13, 14, 15)$$

$$b = \sum m(0, 2, 3, 5, 6, 7, 8, 9) + d(10, 11, 12, 13, 14, 15)$$

$$c = \sum m(0, 1, 2, 3, 4, 7, 8, 9) + d(10, 11, 12, 13, 14, 15)$$

$$d = \sum m(2, 3, 4, 5, 6, 8, 9) + d(10, 11, 12, 13, 14, 15)$$

$$e = \sum m(0, 2, 6, 8) + d(10, 11, 12, 13, 14, 15)$$

$$f = \sum m(0, 1, 3, 4, 5, 6, 7, 8, 9) + d(10, 11, 12, 13, 14, 15)$$

$$g = \sum m(0, 2, 3, 5, 6, 8, 9) + d(10, 11, 12, 13, 14, 15)$$

a :-

		00	01	11	10	
		00	1	0	0	0
		01	1	1	0	1
		11	d	d	d	d
		10	1	1	d	d

$$\Rightarrow a = b_3 + \bar{b}_1 \bar{b}_0 + b_2 \bar{b}_1 + b_2 \bar{b}_0$$

b :-

		00	01	11	10	
		00	1	0	1	1
		01	0	1	1	1
		11	d	d	d	d
		10	1	1	d	d

$$b = b_3 + b_1 + b_2 b_0 + \overline{b}_2 \overline{b}_0$$

C :-

		b ₁ b ₀	00	01	11	10
		b ₃ b ₂	00	01	11	10
			(1)	1	(1)	1
			1	0	1	0
			d	d	d	d
			(1)	1	d	d

$$\Rightarrow C = \overline{b}_1 \overline{b}_0 + b_1 b_0 + \overline{b}_2$$

d :-

		b ₁ b ₀	00	01	11	10
		b ₃ b ₂	00	01	11	10
			0	0	(1)	(1)
			1	1	0	1
			d	d	d	d
			1	1	d	d

$$d = b_3 + \overline{b}_2 b_1 + b_2 \overline{b}_1 + b_1 \overline{b}_0$$

e:-

$$e = b_2 \bar{b}_0 + \bar{b}_2 \bar{b}_1$$

$b_3 b_2$	$b_1 b_0$	00	01	11	10
00	1)	0	0	1)	1)
01		0	0	0	1
11		d	d	d	d
10	1)	0	d	d	d

f:-

$$f = b_2 + \bar{b}_1 + b_0$$

$b_3 b_2$	$b_1 b_0$	00	01	11	10
00	1	1	1	0	
01	1	1	1	1	
11	d	d	d	d	
10	1	1	d	d	

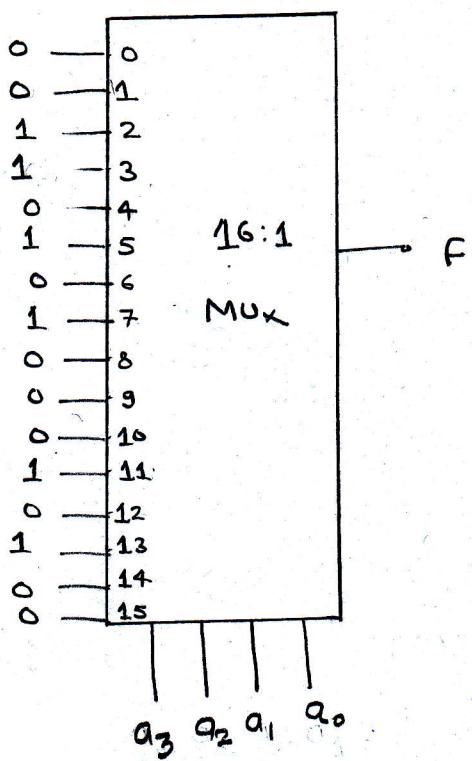
g:-

$b_3 b_2$	$b_1 b_0$	00	01	11	10
00	1	0	1	1	1
01	0	1	0	1	
11	d	d	d	d	
10	1	1	d	d	d

$$g = b_2 \bar{b}_1 b_0 + b_1 \bar{b}_0 + b_3 + \bar{b}_2 b_1 + \bar{b}_2 \bar{b}_0$$

$\therefore A$ is 4 bit number so it can have value from 0 - 15. The prime numbers in this range are 2, 3, 5, 7, 11, 13

$$\Rightarrow \text{so } F = \sum_m (2, 3, 5, 7, 11, 13)$$



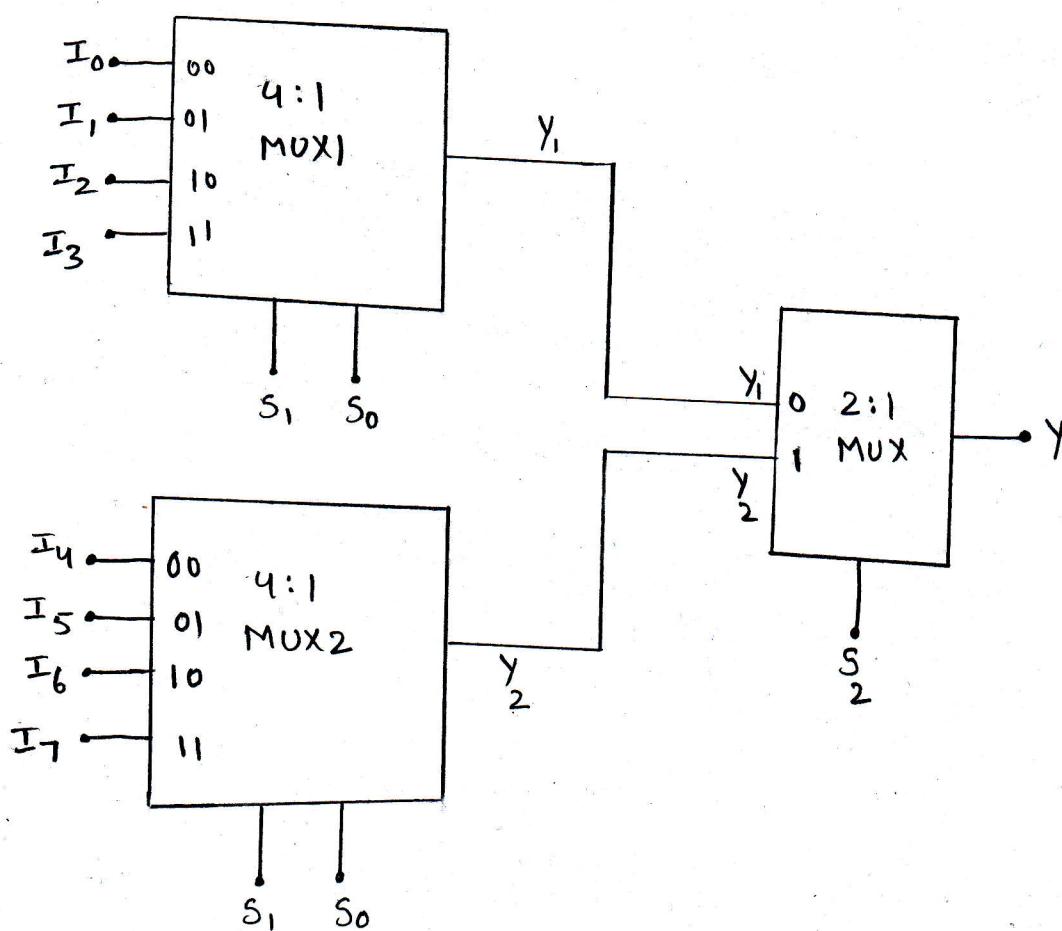
\Rightarrow so $F = 1$, when A is prime,

else $F = 0$;

Assuming $I_0, I_1, I_2 \dots I_7$ are the inputs and S_2, S_1 , and S_0 are the selection lines then an 8:1 MUX will have the following truth table.

S_2	S_1	S_0	y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

we can implement 8:1 MUX using two 4:1 MUX and one 2:1 MUX as shown below.



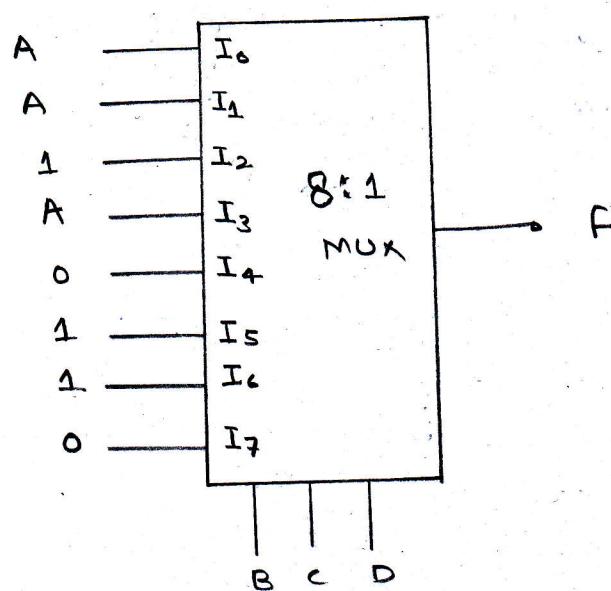
SOLN. (7)

P-12

$$\therefore f = \sum m(2, 5, 6, 8, 9, 10, 11, 13, 14)$$

Let ABCD is 4-Bit number. So truth table is,

	2^3	2^2	2^1	2^0	F
0	0	0	0	0	$F = A$
8	1	0	0	0	1
9	0	0	0	1	$F = A$
1	1	0	0	1	1
2	0	0	1	0	1
10	1	0	1	0	1
3	0	0	1	1	$F = A$
11	1	0	1	1	1
4	0	1	0	0	$F = 0$
12	1	1	0	0	0
5	0	1	0	1	$F = 1$
13	1	1	0	1	1
6	0	1	1	0	$F = 1$
14	1	1	1	0	1
7	0	1	1	1	$F = 0$
15	1	1	1	1	0



The truth table for half-adder is,

Input		Output	
A	B	Sum (S)	Carry (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\Rightarrow S = \overline{A} \cdot B + A \cdot \overline{B} \quad \text{and} \quad C = A \cdot B$$

To implement this using NOR gates,

$$\begin{aligned} S &= \overline{A} \cdot B + A \cdot \overline{B} = \overline{\overline{A} \cdot B + A \cdot \overline{B}} = (\overline{\overline{A} \cdot B}) \cdot (\overline{A \cdot \overline{B}}) \\ &= (\overline{A + \overline{B}}) \cdot (\overline{\overline{A} + B}) = \overline{AB + \overline{A}\overline{B}} \\ &= \overline{\overline{A} \cdot \overline{B} + \overline{A} \cdot B} = \overline{\overline{A} + \overline{B} + \overline{A + B}} \end{aligned}$$

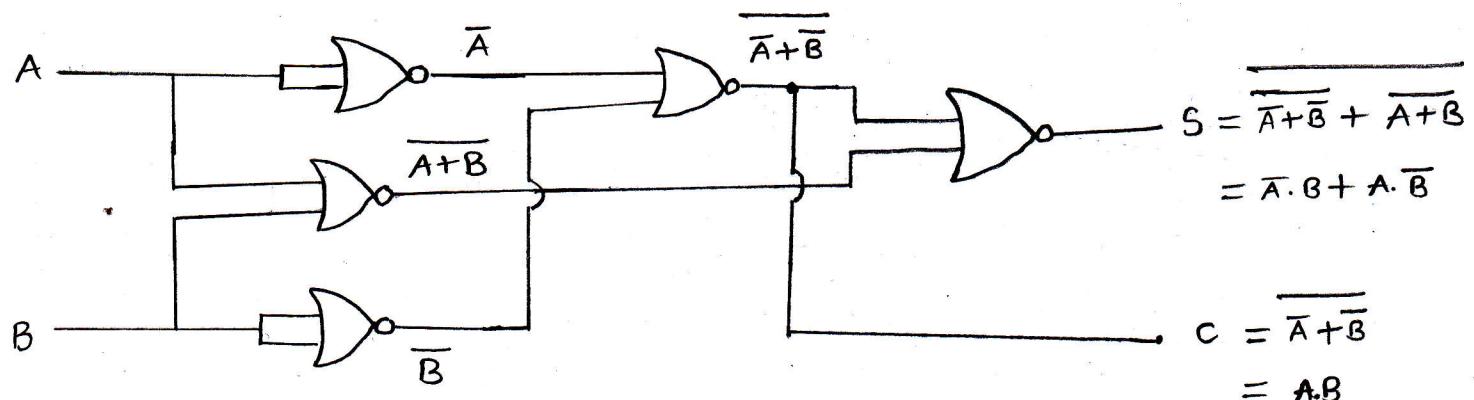
$$C = A \cdot B = \overline{\overline{A} \cdot B} = \overline{\overline{A} + \overline{B}}$$

This implies we have to generate,

$$i) \overline{A} \text{ and } \overline{B}$$

$$ii) \overline{A + \overline{B}} \text{ and } \overline{\overline{A} + B}$$

to get above 'S' and 'C'.



A full adder has the following truth table

input			output	
x	y	Cin	Sum (S)	carry-out (Cout)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

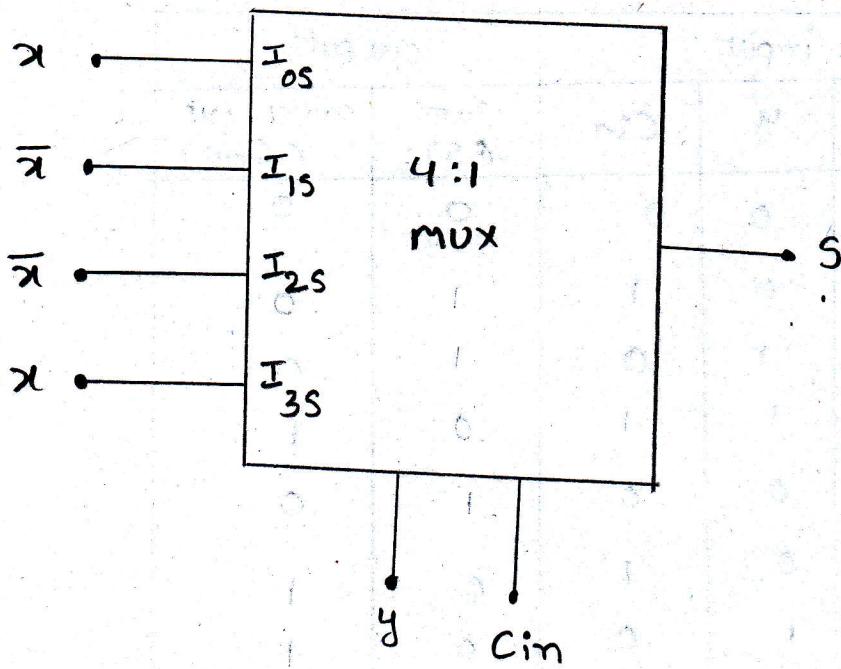
we use two 4:1 MUX such that one implements the sum (S) and other implements the carry-out (Cout).

Also let's use y, Cin as selection lines and 'x' as the input line to the multiplexer.

x	y	Cin	S	Cout	
0	0	0	0	0	$\left. \begin{array}{l} S = x \\ \text{Cout} = 0 \end{array} \right\}$
1	0	0	1	0	
0	0	1	1	0	$\left. \begin{array}{l} S = \bar{x} \\ \text{Cout} = x \end{array} \right\}$
1	0	1	0	1	
0	1	0	1	0	$\left. \begin{array}{l} S = \bar{x} \\ \text{Cout} = x \end{array} \right\}$
1	1	0	0	1	
0	1	1	0	1	$\left. \begin{array}{l} S = x \\ \text{Cout} = 1 \end{array} \right\}$
1	1	1	1	1	

Now the sum-MUX look like,

P-15



And the carry-out-mux is

