

Pokhara University
Faculty of science and technology

Subject . Digital Logic Capsule
Computer Engineering I Semester
Software Engineering I Semester
Ele and Eletrical Engineering III Semester

Name..

Subject..

Semester..

By. Er. Manoj Kumar Singh

Availeval. Bijayapur library

Design a combinational CKT which takes three input numbers and produces an output equal to square of the input. [2013 Fall]

Q1 Analysis.

$$\text{Max No.} = 111 = \text{square of } 7 = 7^2 = 49$$

$$49 \text{ in binary} = 110001$$

Total no. of bits in O/P seq = 6.

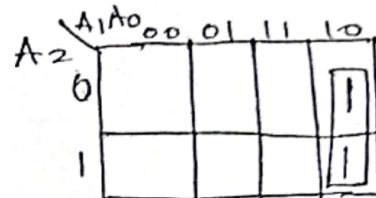
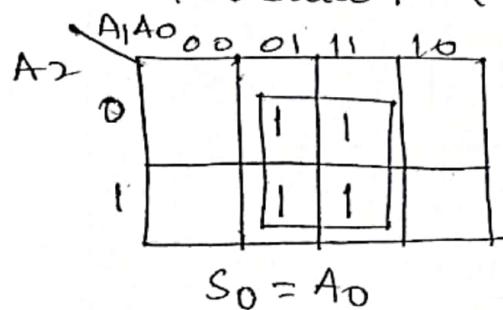
	I/P			O/P					
	A ₂	A ₁	A ₀	S ₅	S ₄	S ₃	S ₂	S ₁	S ₀
0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	1
2	0	1	0	0	0	0	1	0	0
3	0	1	1	0	0	1	0	0	1
4	1	0	0	0	1	0	0	0	0
5	1	0	1	0	1	1	0	0	1
6	1	1	0	1	0	0	1	0	0
7	1	1	1	1	1	0	0	0	1

- square of 0=0
square of 1=1
square of 2=4
square of 3=9
square of 4=16
square of 5=25
square of 6=36
square of 7=49

Truth Table

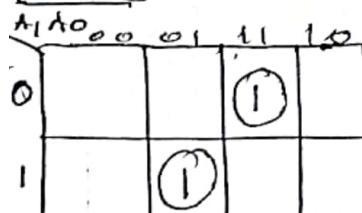
Now let us draw the K-Map for S₀ for S₂

[S₁ = 0 forced to make K-Map]

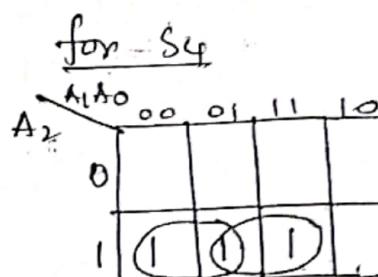


$$S_2 = A_1 \bar{A}_0$$

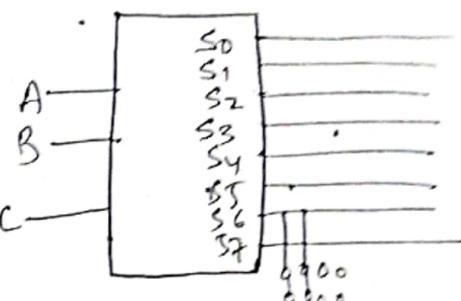
for S₃



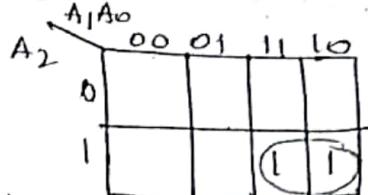
$$S_3 = A_0 A_1 \bar{A}_2 + A_0 \bar{A}_1 A_2$$



$$S_4 = \bar{A}_1 A_2 + A_0 A_2$$



for S₅



$$S_5 = A_1 A_2$$

Now, let us draw the logic diagram.

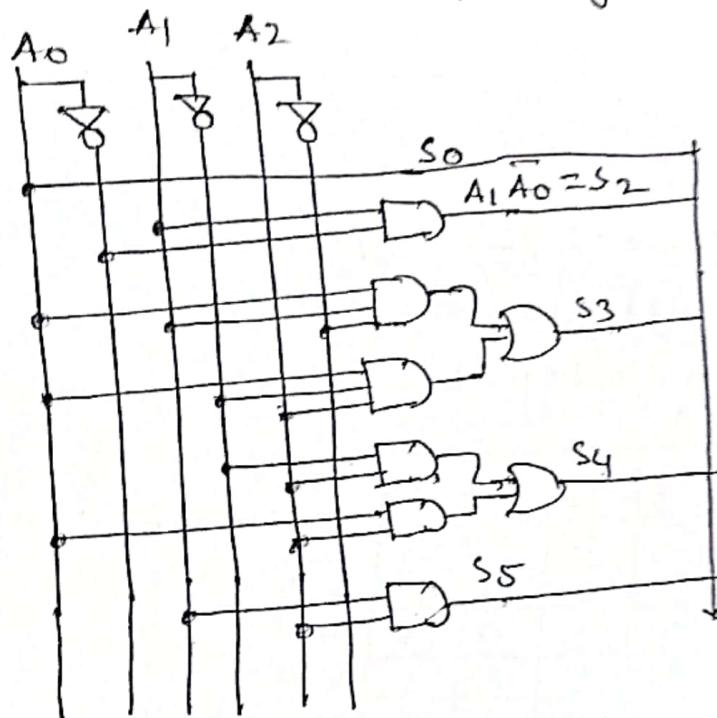


Fig:- logic diagram of the ~~given~~ 3 bit square off.

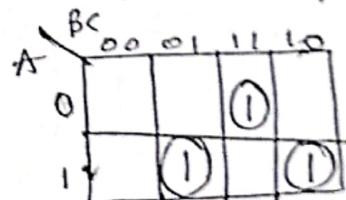
2. Design a Combinational logic Ckt with 3 inputs that prov 1 o/p when exactly two Variables are 1.

⇒ Let us draw the TruthTable.

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

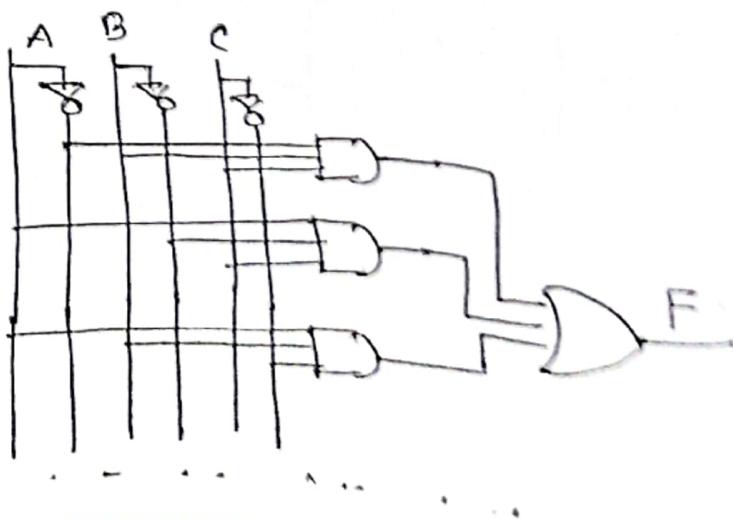
TruthTable.

Now K-Map for F



$$F = \overline{A}B\overline{C} + A\overline{B}\overline{C} + AB\overline{C}$$

Now draw the logic diagram.



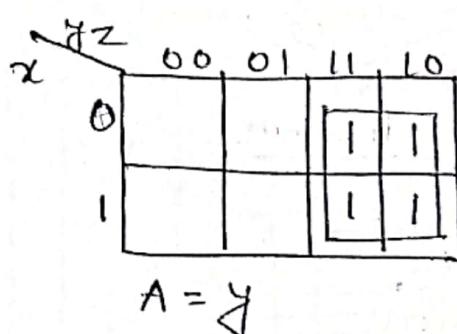
Q. Design a combinational CKT with three inputs x, y, z and three outputs A, B, C when the binary input is 0, 1, 2 or 3, the binary output is two greater than the input. When the binary input is 4, 5, 6 or 7, the binary output is two less than the input.

Sol:

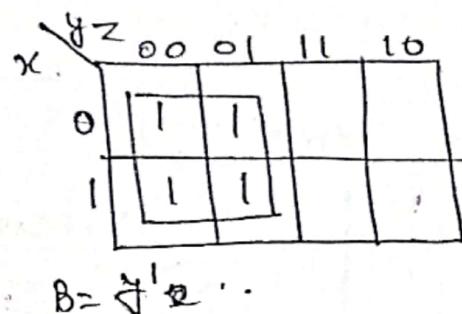
	x	y	z	A	B	C
0	0	0	0	0	1	0
1	0	0	1	0	1	1
2	0	1	0	1	0	0
3	0	1	1	1	0	1
4	1	0	0	0	1	0
5	1	0	1	0	1	1
6	1	1	0	1	0	0
7	1	1	1	1	0	1

Truth Table

No. 1 K Map for Output A

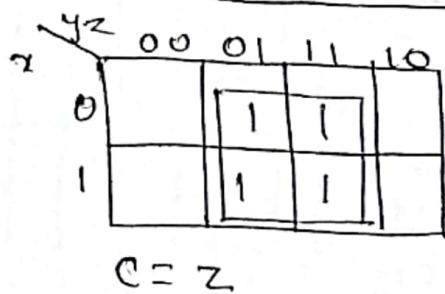


K-Map for Output B



Again

K-Map for Output C



Now draw the logic diagram

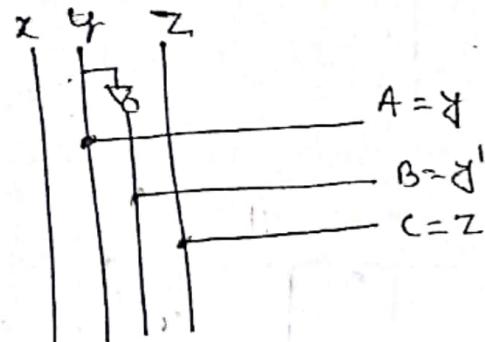


Fig. Logic diagram of given circuit

Design a combinational circuit with three inputs and one output.

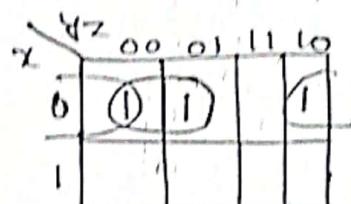
- (a) The output is 1 when binary value of the inputs is less than or equal to 3. The output is 0 otherwise.
- (b) The output is 1 when the binary value of the input is an odd number.
- (c) The output is 1 when the binary value of the input is an even number.

Solution → (a)

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Truth table.

Now write the K-map for F



$$F = \bar{x}\bar{y} + \bar{x}\bar{z}$$

Now draw the logic diagram of given Boolean function.

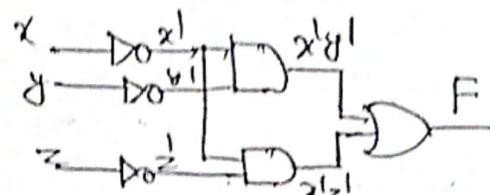


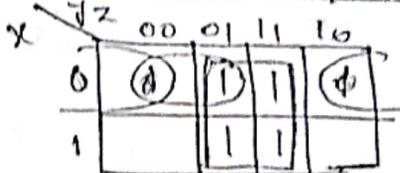
Fig: logic diagram

(b)

for odd No. O/P = 1

x	y	z	F
0	0	0	0
1	0	1	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	0	0
7	1	1	1

K-map for O/P F



$$F = z$$

Now logic circuit

$$z \rightarrow F$$

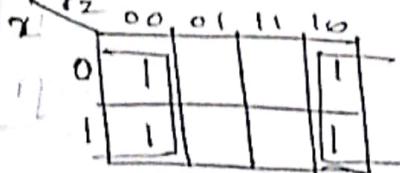
Fig: logic circuit

(c)

for even No. O/P = 1

x	y	z	F
0	0	0	1
1	0	0	0
2	0	1	0
3	0	1	0
4	1	0	1
5	1	0	0
6	1	0	1
7	1	1	0

K-map for O/P F



$$F = \bar{z}$$

$$z \rightarrow \bar{z} \rightarrow F$$

$F(w,x,y,z) = \sum(1,6,7,8,11,13)$ with don't care condition
 $d(w,x,y,z) = \sum(0,2,3,4,10,12)$. Then implement the function using NOR gates only. [2012 Fall]

Sol. Here given:

$$F(w,x,y,z) = \sum(1,6,7,8,11,13)$$

$$d(w,x,y,z) = \sum(0,2,3,4,10,12)$$

		wz	$(y+z)$	$(\bar{y}+z)$	$(\bar{y}+\bar{z})$	$(\bar{\bar{y}}+z)$	
		00	0	X	X		(V group)
		01	X		O	O	(III group)
		11	X	O			(IV group)
		10	O		O	X	

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

Now Double complementing on both sides we get

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

Now, Implementing the function using NOR gate only.

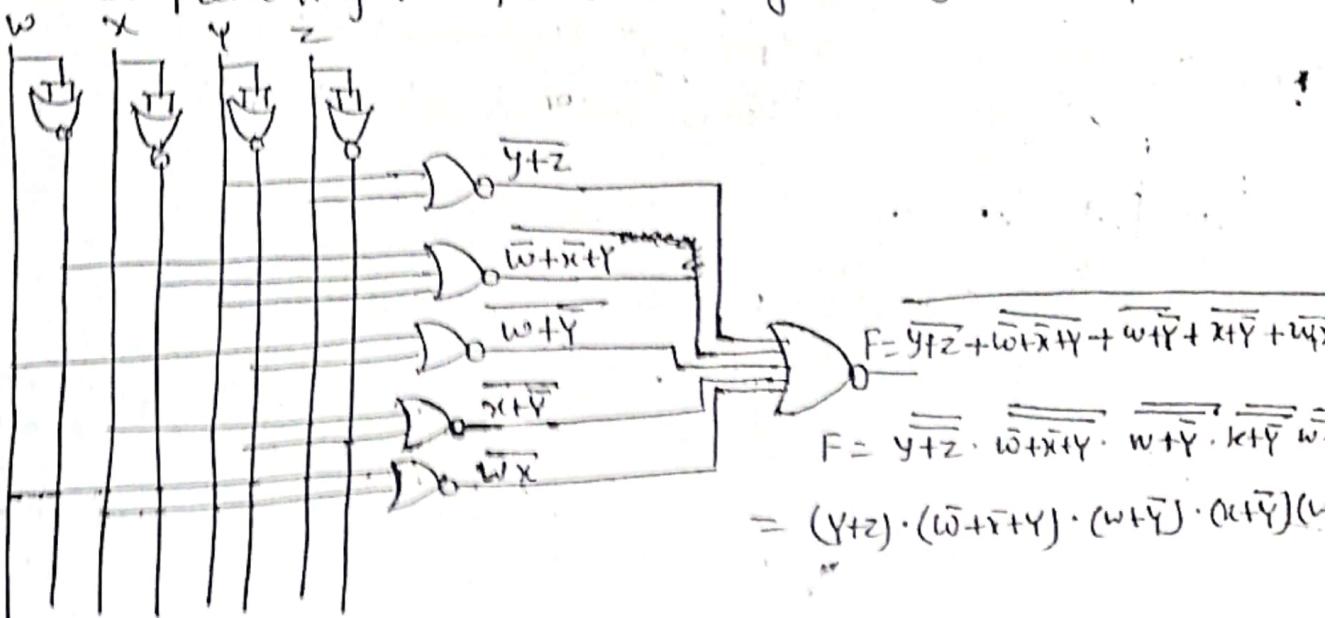


Fig. Logic diagram using NOR gate only.

Prepared By: O. Manoj (O. Singh).

* Use K-Map to simplify the given function & implement the simplified function using NOR gate only.

$$F(A, B, C, D) = \sum(1, 3, 8, 9, 12, 13, 14, 15) \quad \phi. \text{ don't care}$$

$$d(A, B, C, D) = \sum(2, 7, 10) \quad (\text{2012 Spring})$$

Q. 9: Here, Given

$$F(A, B, C, D) = \sum(1, 3, 8, 9, 12, 13, 14, 15)$$

$$d(A, B, C, D) = \sum(2, 7, 10)$$

AB\CD	00	01	11	10
00		0	0	X
01			X	
11	0		0	0
10	0	0		X

$$F = (B + C + \bar{D}) \cdot (A + \bar{C} + \bar{D}) \cdot (\bar{A} + \bar{B} + \bar{C}) \cdot (\bar{A} + D)$$

Now, Implementing this function by using NOR gate only

Now, Double complementing on both sides we get

$$\begin{aligned} F &= \overline{(B + C + \bar{D}) \cdot (A + \bar{C} + \bar{D}) \cdot (\bar{A} + \bar{B} + \bar{C}) \cdot (\bar{A} + D)} \\ &= \overline{(B + C + \bar{D})} + \overline{(A + \bar{C} + \bar{D})} + \overline{(\bar{A} + \bar{B} + \bar{C})} + \overline{(\bar{A} + D)} \end{aligned}$$

Now, Implementing this function by using NOR gate only

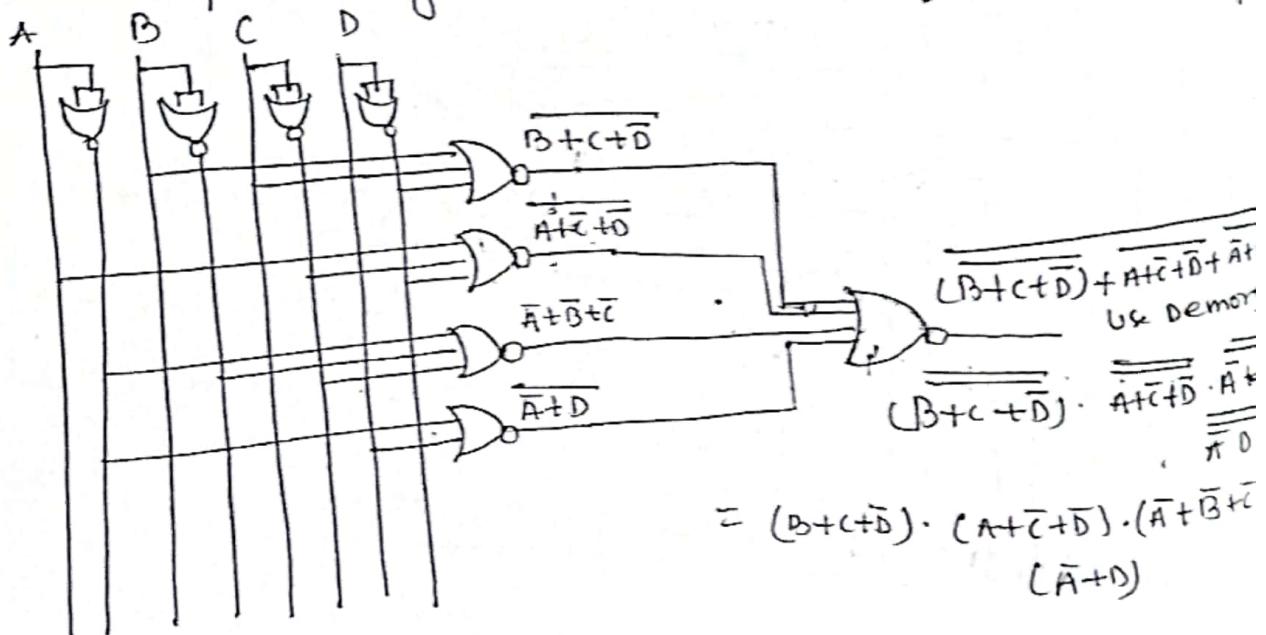


Fig: Logic diagram using NOR gates only.

Prepared By:- Er. Manoj Kr. Singh.

(1) Design a logic function chart to implement the Boolean function.

2

$$F(A, B, C, D) = \sum(1, 3, 7, 11, 15)$$

$$D(A, B, C, D) = \sum(0, 2, 5) \text{ in the term of}$$

(i) SOP

(ii) implement with NAND-NAND gate
[2013 Fall] only.

Sol:

$$F(A, B, C, D) = \sum(1, 3, 7, 11, 15)$$

$$D(A, B, C, D) = \sum(0, 2, 5)$$

		CD		00		01		11		10	
		A	B	00	X	11	1	11	1	10	X
				00							
		00	01								
		11	10								

$$F = \overline{AD} + CD$$

Now, Double complementing on Both side we get

$$\overline{F} = \overline{\overline{AD} + CD}$$

$$= \overline{\overline{AD} \cdot \overline{CD}} \quad (\text{using De morgan's theorem})$$

Now, Implement by using NAND gate only.

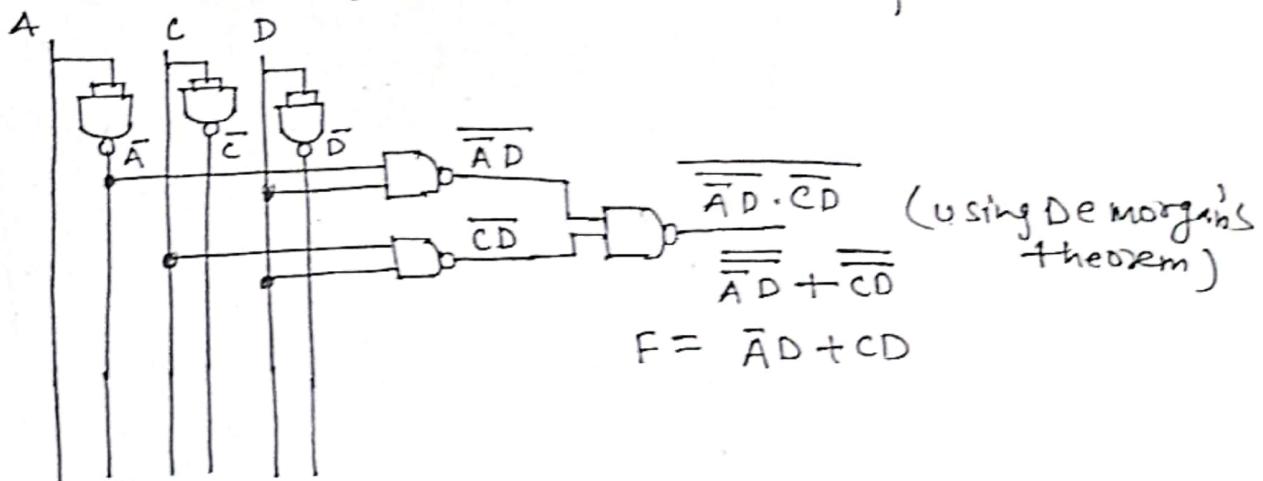


Fig:- logic diagram using NAND gate only

Prepared By: ER. Manoj Kumar Singh

2

(2) Obtain a reduced expression for the following
 $F(A, B, C, D) = \sum m(0, 1, 2, 5, 7, 8, 9, 10, 13, 15)$ and
realize the minimized function using NOR gate only.

Soln → Here,

$$F(A, B, C, D) = \sum m(0, 1, 2, 5, 7, 8, 9, 10, 13, 15)$$

		CD	$(C+D)$	$(C+\bar{D})$	$(\bar{C}+D)$	$(\bar{C}+\bar{D})$	
	AB	00	00	01	11	10	
	00	00	0	0	0	0	
	01	01	0	0	0	0	
	11	11	0	0	0	0	
	10	10	0	0	0	0	

$$F = (B + \bar{C} + \bar{D}) \cdot (\bar{B} + D)$$

Now, Double complementing on both sides we get -

$$\begin{aligned} F &= \overline{(B + \bar{C} + \bar{D}) \cdot (\bar{B} + D)} \\ &= \overline{(B + \bar{C} + \bar{D})} + \overline{(\bar{B} + D)} \end{aligned}$$

Now, Implement by using NOR gate only -

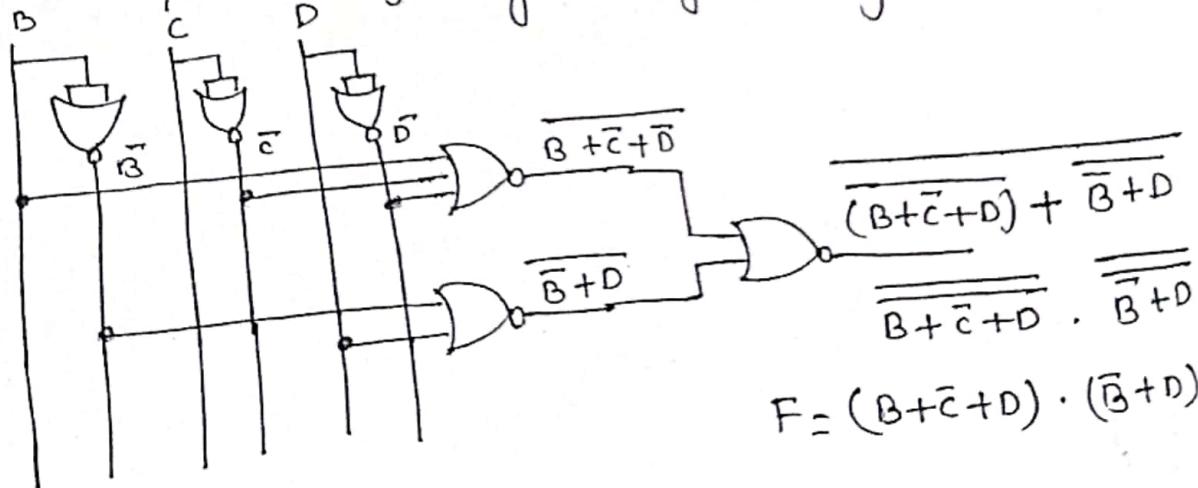


Fig: logic diagram

(Q) Obtain a K-map for the following
 $F(A, B, C, D) = \sum m(0, 1, 2, 8, 10, 11, 14, 15)$ and realize
the minimized function using NOR gate only. 3

⇒ Here,

$$F(A, B, C, D) = \sum m(0, 1, 2, 8, 10, 11, 14, 15) \text{ for Maxterm}$$

		CD				(C+D)				(C+D)				(C+D)				
		00	01	11	10	00	01	11	10	00	01	11	10	00	01	11	10	
AB	00																	
01																		
11																		
10																		

$$F(A, B, C, D) = \prod M(3, 4, 5, 6, 7, 9, 13)$$

$$F = (A + \bar{C} + \bar{D}) \cdot (\bar{A} + C + \bar{D}) \cdot (A + \bar{B}) \cdot (\bar{B} + C)$$

Now, Double complementing on Both Sides we get.

$$\begin{aligned} \overline{\overline{F}} &= \overline{(A + \bar{C} + \bar{D}) \cdot (\bar{A} + C + \bar{D}) \cdot (A + \bar{B}) \cdot (\bar{B} + C)} \\ &= \overline{(A + \bar{C} + \bar{D})} + \overline{(\bar{A} + C + \bar{D})} + \overline{(A + \bar{B})} + \overline{(\bar{B} + C)} \end{aligned}$$

Now, Implement by using NOR gate only.

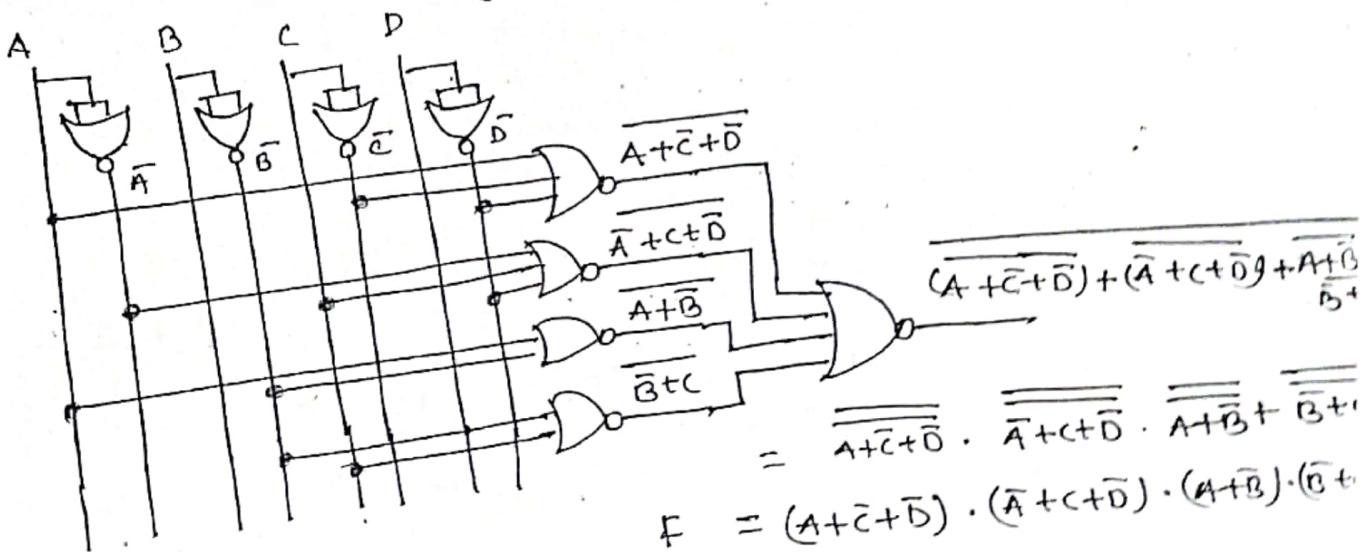


Fig. logic diagram.

- (4) Design a logic circuit to implement one Boolean function.
- $$F(A, B, C, D) = \sum(1, 3, 4, 5, 7, 9, 13, 14, 15)$$
- $$D(A, B, C, D) = \sum(0, 2, 8)$$
- (i) Sum of product
(ii) Implement with NAND-NAND gate only.

Soln: Here, Given

$$F(A, B, C, D) = \sum(1, 3, 4, 5, 7, 9, 13, 14, 15)$$

$$D(A, B, C, D) = \sum(0, 2, 8)$$

		CD	00	01	11	10
		AB	00	01	11	10
	X		1	1	1	X
	1		1	1	1	..
			1	1	1	1
	X		1	1	X	

$$F = ABC + \bar{A}\bar{C} + \bar{A}D + \bar{C}D$$

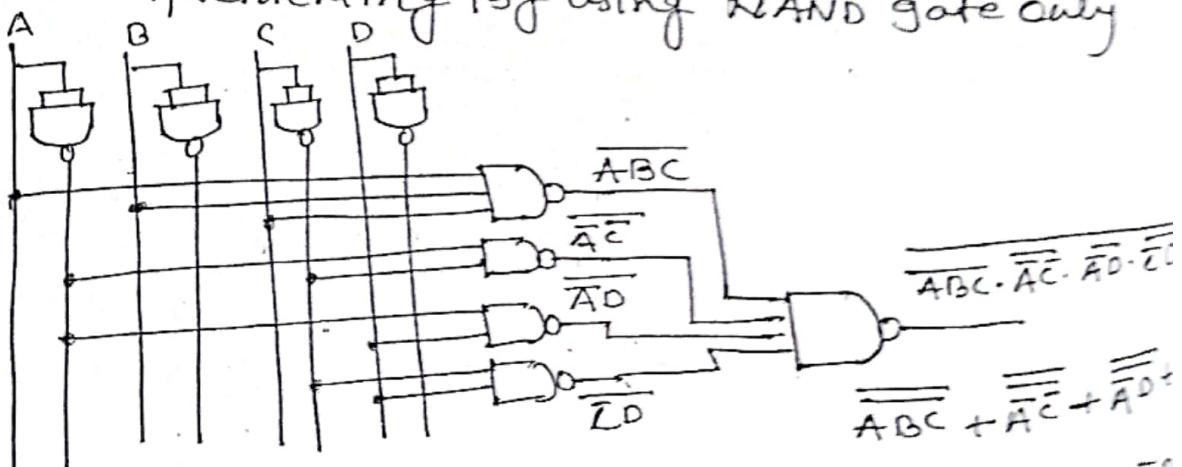
$$(i) SOP F = ABC + \bar{A}\bar{C} + \bar{A}D + \bar{C}D$$

Now, Double complementing on Both side we get

$$\overline{\overline{F}} = \overline{\overline{ABC + \bar{A}\bar{C} + \bar{A}D + \bar{C}D}}$$

$$= \overline{\overline{ABC} \cdot \overline{\overline{AC}} + \overline{\overline{AD}} + \overline{\overline{CD}}}$$

Now, Implementing By using NAND gate only



$$F = ABC + \bar{A}\bar{C} + \bar{A}D + \bar{C}D$$

Fig: logic diagram.

Q. A Boolean Function is given by $F(A, B, C, D) = \sum(3, 4, 5, 7, 9, 13, 14, 15)$ and don't care condition $d(A, B, C, D) = \sum(0, 2, 8)$. Simplify it using K-map and implement using NAND gate only.
(2014 Fall)

\Rightarrow Here, (A, B, C, D)
Given $f = \sum(3, 4, 5, 7, 9, 13, 14, 15)$
 $d(A, B, C, D) = \sum(0, 2, 8)$

		CD	00	01	11	10
		A\B	00			
		00			(1)	
		01	(1)	(1)	(1)	
		11		(1)	(1)	(1)
		10	(1)			

$$F = \overline{ABC} + \overline{ACD} + \overline{ACD} + ABC$$

Now, Double complementing on both side we get

$$\begin{aligned}\overline{\overline{F}} &= \overline{\overline{\overline{ABC} + \overline{ACD} + \overline{ACD} + ABC}} \\ &= \overline{\overline{ABC} \cdot \overline{ACD} + \overline{ACD} \cdot \overline{ABC}}\end{aligned}$$

Now, Implement by using NAND gate only.

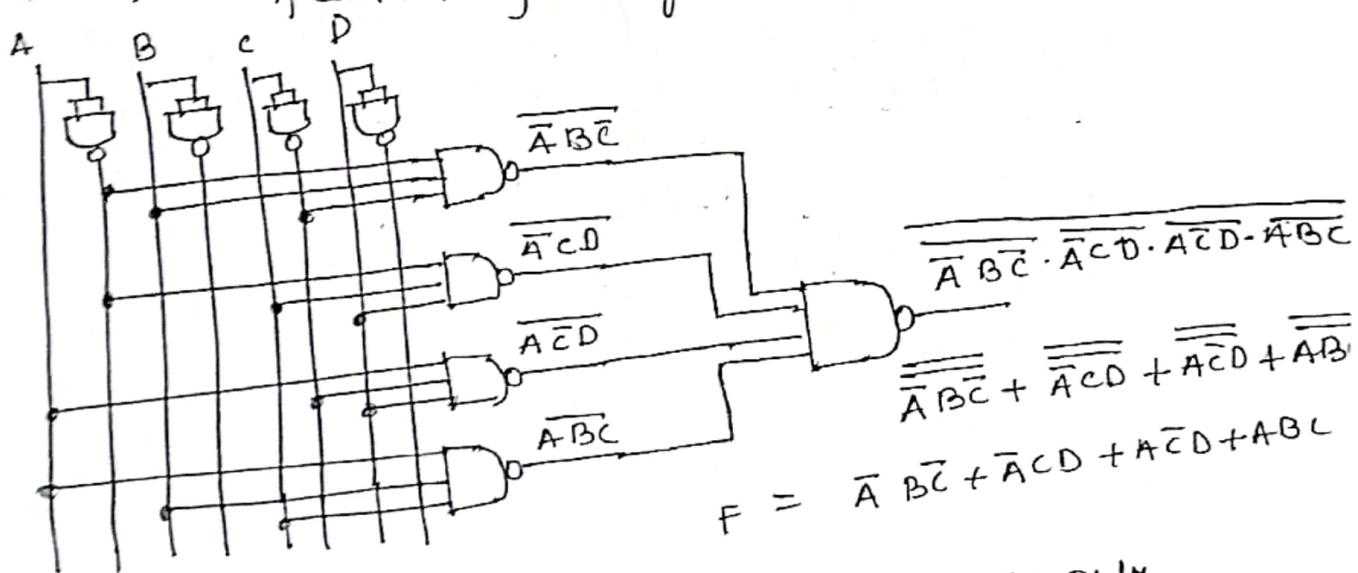


Fig: logic diagram using NAND gate only.

* A Boolean function is given by $F(A, B, C, D) = \sum(3, 4, 6, 8, 10, 12, 14)$ and don't care condition $d(A, B, C, D) = \sum(0, 2, 7)$. Simplify it using K-Map & implement using NAND gate only.

Sol: [2015 Fall] ^{Spring}
Here, Given

$$F(A, B, C, D) = \sum(3, 4, 6, 8, 10, 12, 14)$$

$$d(A, B, C, D) = \sum(0, 2, 7)$$

		CD	00	01	11	10
		AB	00	X	1	X
		01	1		X	1
		11	1			1
		10	1			1

$$F = \bar{A}C + \bar{D}$$

Now, Double complementing on Both side we get

$$\overline{\overline{F}} = \overline{\overline{\bar{A}C + \bar{D}}} \\ = \overline{\overline{\bar{A}C} \cdot \overline{\bar{D}}}$$

Now Implement by using NAND gate only.

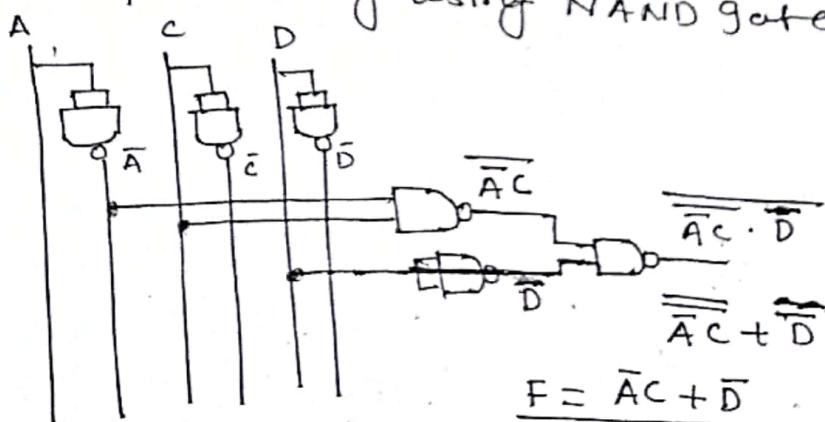


Fig: Logic diagram

Prepared By:- AR. Manoj Kumar...

Q. Simplify given function using K-Map & implement it in a circuit.

$$F(w,x,y,z) = \sum(1,4,5,6,12,14,15) \text{ & don't care}$$

Condition D(w,x,y,z) = $\sum(11,13)$

[2016 spring]

Soln:

$$F(w,x,y,z) = \sum(1,4,5,6,12,14,15)$$

$$D(w,x,y,z) = \sum(11,13)$$

w\z	00	01	11	10
00		(1)		
01	1	(1)		1
11	1	X	1	1
10			X	

$$F = \bar{w}\bar{y}z + x\bar{z} + w\bar{x}$$

Now, Double complementing on both sides we get

$$\overline{\overline{F}} = \overline{\overline{\bar{w}\bar{y}z + x\bar{z} + w\bar{x}}}$$
$$= \overline{\overline{\bar{w}\bar{y}z}} \cdot \overline{\overline{x\bar{z}}} \cdot \overline{\overline{w\bar{x}}}$$

Now, Implement by using NAND gate

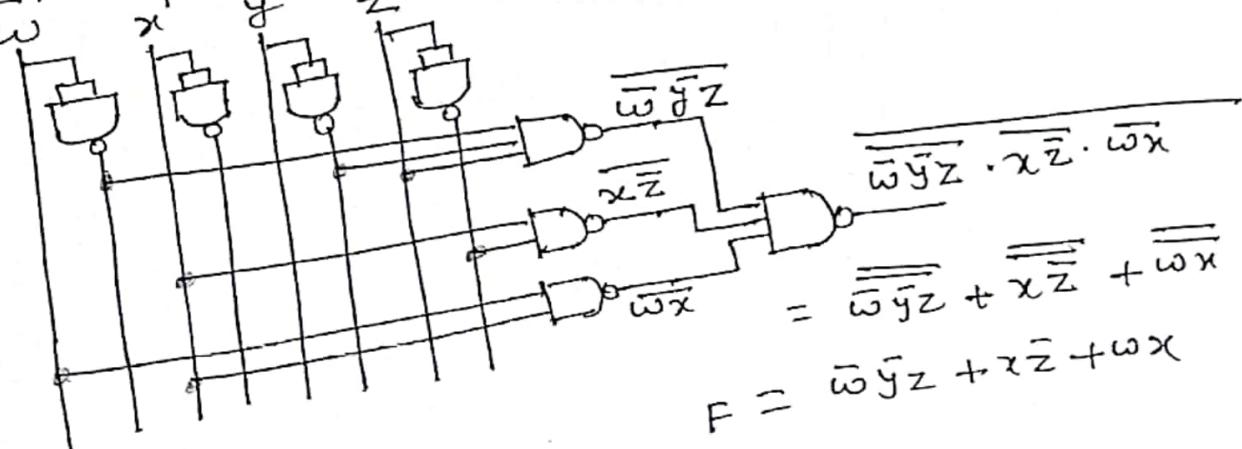


Fig. logic diagram.

Q. Simplify the following Boolean expression using K-Map & implement using NAND gates only.

$$F(A, B, C, D) = \sum(0, 2, 4, 6, 12, 15)$$

$$D(A, B, C, D) = \sum(8, 10, 14) \quad [2019 Fall]$$

Soln

		CD	00	01	11	10
		AB	00	1		
		01	1			1
		11	1		1	X
		10	X			X

$$F = ABC + \bar{D}$$

Now, Double complementing on Both sides we get

$$\begin{aligned} \overline{\overline{F}} &= \overline{\overline{ABC} + \bar{D}} \\ &= \overline{\overline{ABC} \cdot \bar{D}} \end{aligned}$$

Now, Implementing by using NAND gates only

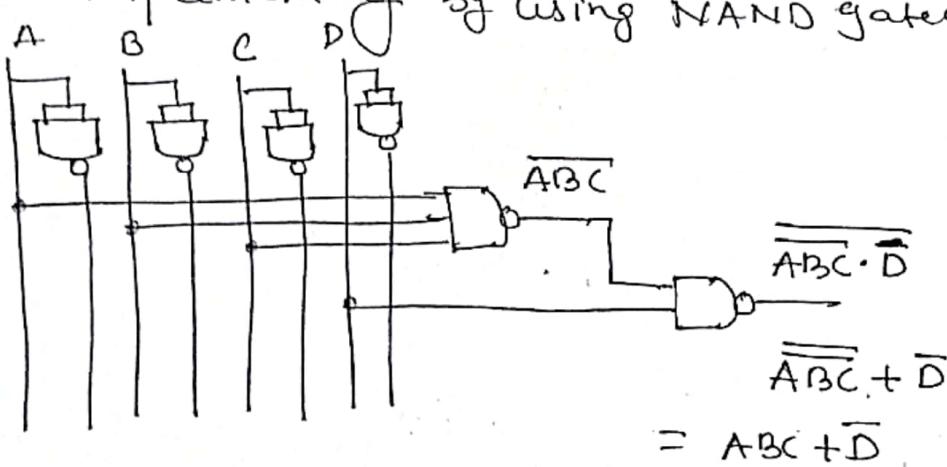


Fig. logic diagram.

- Q. Use K-Map to simplify the given function & once by considering the don't care condition & once by ignoring the don't care condition & realize it using the basic gates
- $$F(A, B, C, D) = \sum(1, 4, 8, 12, 13, 15) \oplus \text{don't care } d(A, B, C, D)$$
- $$= \sum(3, 14) \quad [2017 \text{ Spring}]$$

Sol. (i) First case

Considering the don't care Condition

$$F(A, B, C, D) = \overline{\sum}(1, 4, 8, 12, 13, 15)$$

$$d(A, B, C, D) = \sum(3, 14)$$

		CD	00	01	11	10
AB	00		1	X		
		1				
AB	01	1				
			1	1	1	X
AB	11	1				
AB	10	1				

$$F = \overline{A} \overline{B} \overline{D} + B \overline{C} \overline{D} + A \overline{C} \overline{D} + AB$$

Logic diagram

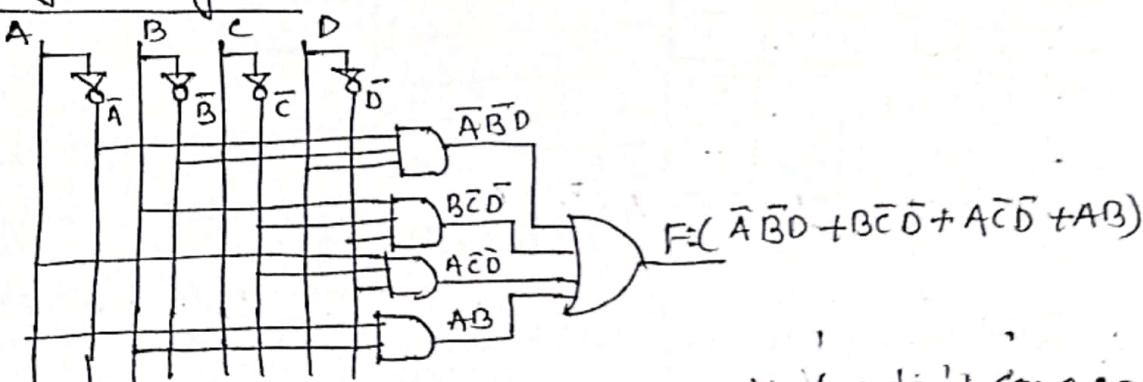


Fig. logic diagram with considering don't care condition

(ii) Second case

Ignoring the don't care condition

$$F(A, B, C, D) = \sum(1, 4, 8, 12, 13, 15)$$

		CD	00	01	11	10
AB	00		1			
		1				
AB	01	1				
			1	1	1	
AB	11	1				
AB	10	1				

$$F = \overline{A} \overline{B} \overline{C} \overline{D} + B \overline{C} \overline{D} + A \overline{C} \overline{D} + ABD$$

) logic diagram in next page

Prepared By:- Er. Manoj Kr. Singh

$$F = \overline{A}\overline{B}\overline{C}D + B\overline{C}\overline{D} + A\overline{C}\overline{D} + ABD$$

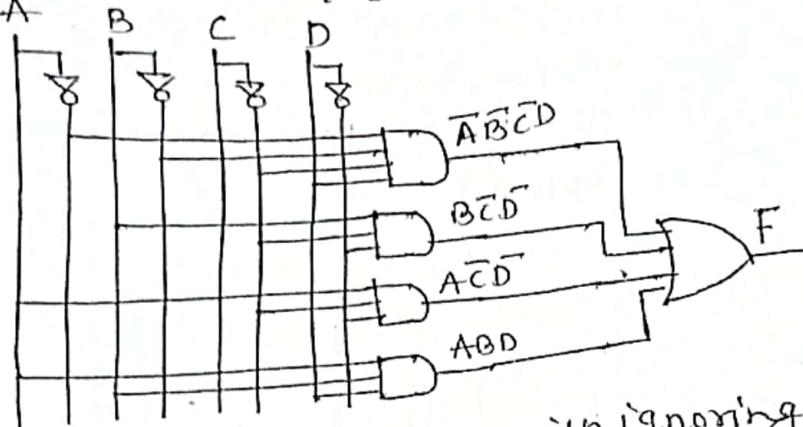


Fig:- logic diagram with ignoring the don't care cond.

- Q. Use K-Map to simplify the given Boolean function w/ don't care condition & realize it using basic gates only
 $F = \sum(1, 4, 8, 12, 13, 15) \quad \Phi d = \sum(3, 7, 11, 14)$ (2018 Fall)
Soln. Here Given.

$$F = \sum(1, 4, 8, 12, 13, 15) \quad \Phi d = \sum(3, 7, 11, 14)$$

AB	CD	00	01	11	10
00		1	X		
01		1	X		
11		1	1	1	X
10		1	X		

$$F = \overline{A}\overline{B}D + B\overline{C}\overline{D} + A\overline{C}\overline{D} + AB$$

No. 2: Realizing this expression using basic gates only:

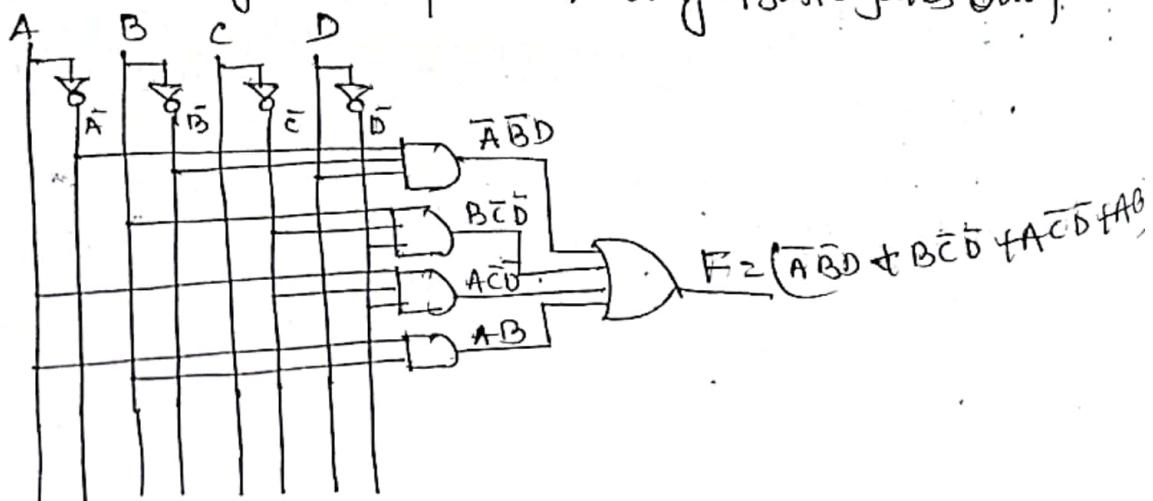


Fig:- logic diagram.

Prepared by: En. Manoj Kr. Singh,

* Design a combinational circuit whose input is a four bit number and at the output, we obtain the two's complement of the input.

⇒ Let A_3, A_2, A_1, A_0 be the four bit input & B_3, B_2, B_1, B_0 represents the two's complement of input.

(i) First let us write the Truth Table.

Decimal	Input				Output			
	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1	1
2	0	0	1	0	1	1	1	0
3	0	0	1	1	1	1	0	1
4	0	1	0	0	1	1	0	0
5	0	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	0
7	0	1	1	1	1	0	0	1
8	1	0	0	0	1	0	0	0
9	1	0	0	1	0	1	1	1
10	1	0	1	0	0	1	0	1
11	1	0	1	1	0	1	0	0
12	1	1	0	0	0	0	1	1
13	1	1	0	1	0	0	0	1
14	1	1	1	0	0	0	0	1
15	1	1	1	1	0	0	0	1

Truth Table.

K-Map Simplification.

For Output B_3

	A_3A_2	00	01	11	10
A_1A_0	00	0	14	12	① ₈
01	1	1	15	13	9
11	1	1	17	15	11
10	1	1	6	4	10

$$B_3 = \overline{A_3}A_2 + \overline{A}_1\overline{A}_0A_3\overline{A}_2 + A_0\overline{A}_3$$

$$= \overline{A}_3A_2 + \overline{A}_1\overline{A}_0A_3\overline{A}_2 + A_1\overline{A}_3$$

$$= \overline{A}_3A_2 + \overline{A}_1\overline{A}_0A_3\overline{A}_2 + \overline{A}_3(A_0+A_1)$$

		A_3A_2	00	01	11	10
A_1A_0	00	0	14	12	8	Group 2
01	1	1	15	13	9	Group 1
11	1	1	17	15	11	
10	1	1	6	4	10	

$$B_2 = A_2\overline{A}_1\overline{A}_0 + A_0\overline{A}_2 + A_1\overline{A}_2$$

$$= \overline{A}_0\overline{A}_1A_2 + \overline{A}_2(A_0+A_1)$$

$$\text{But } \overline{A}_0 \cdot \overline{A}_1 = \overline{A}_1 + \overline{A}_0 \text{ (De Morgan's)}$$

$$B_2 = A_2(\overline{A}_0 + \overline{A}_1) + \overline{A}_2(A_0 + A_1)$$

$$\text{Let } A_0 + A_1 = X$$

$$B_2 = A_2\overline{X} + \overline{A}_2X = A_2 \oplus X$$

$$B_2 = A_2 \oplus (A_0 + A_1)$$

For output B_1

		00	01	11	10
		00			
		01	1	1	1
		11	3	7	5
		10	2	1	1
$A_3 A_2$	00				
$A_1 A_0$	00				
$A_3 A_2$	01				
$A_1 A_0$	01	1	1	1	1
$A_3 A_2$	11				
$A_1 A_0$	11	1	1	1	1
$A_3 A_2$	10				
$A_1 A_0$	10				

For output B_0

		00	01	11	10
		00			
$A_3 A_2$	00				
$A_1 A_0$	01	1	1	1	1
$A_3 A_2$	11	1	1	1	1
$A_1 A_0$	10				

$$B_1 = A_0 \bar{A}_1 + \bar{A}_0 A_1 \\ = A_0 \oplus A_1$$

$$B_0 = A_0$$

$$\therefore B_3 = \bar{A}_3 A_2 + \bar{A}_1 \bar{A}_0 A_3 \bar{A}_2 + \bar{A}_3 (A_0 + A_1) \\ B_2 = A_2 \oplus (A_2 + A_1)$$

Finally let us realize the combinational CKT using b_{ai} gates.

$A_3 \quad A_2 \quad A_1 \quad A_0$ input

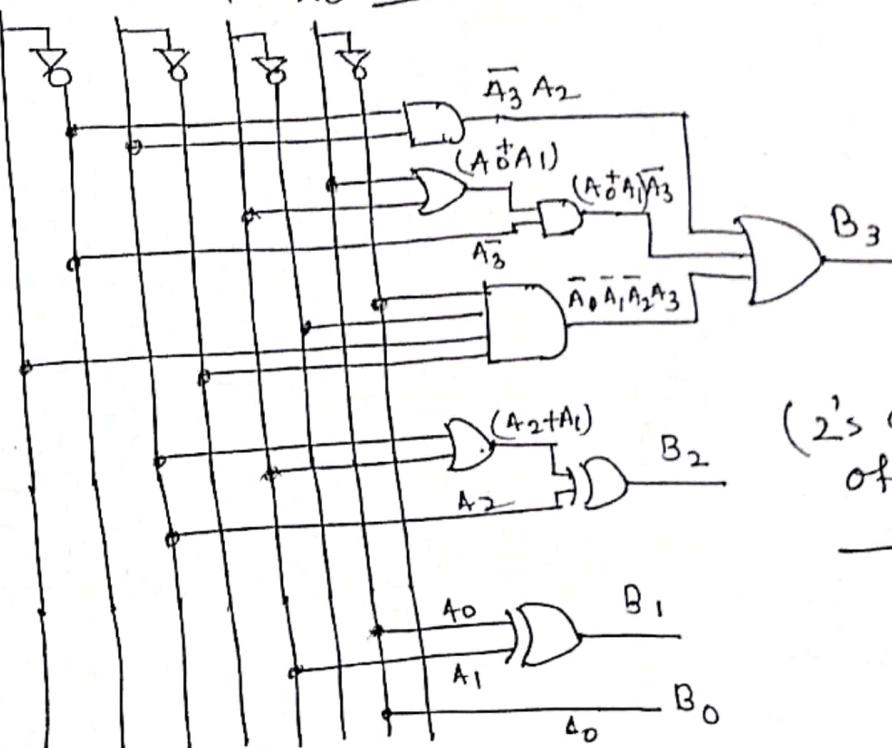


Fig: The Required combinational CKT

- & Design a code conversion ckts which converts BCD to excess-3 code (2012 fall) (Done in class)
- Q. Design a combinational ckts that has four inputs A_3, A_2, A_1, A_0 . One of the O/Ps is high when majority of i/p's are high. Second O/P is high only when all inputs are of same type (2017 spring) (Done in class)

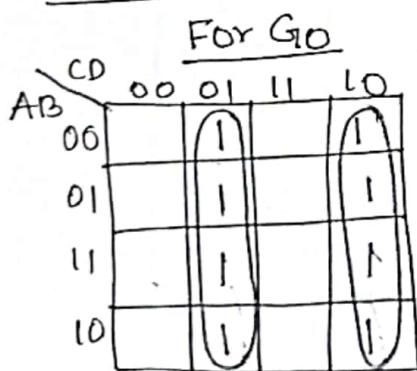
* Design a code conversion circuit which converts Binary code to Gray code. (2019 Fall)

8019 → First let us write the truth table

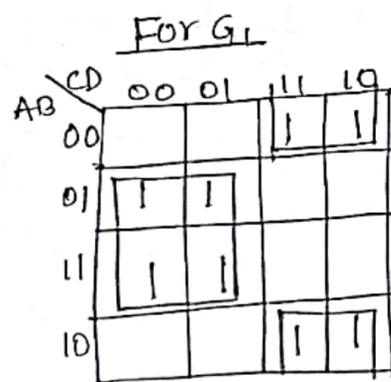
Decimal	Binary Code D C B A	Gray Code G3 G2 G1 G0
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 0 0 0
9	1 0 0 1	1 1 0 1
10	1 0 1 0	1 1 1 1
11	1 0 1 1	1 1 1 0
12	1 1 0 0	1 0 1 0
13	1 1 0 1	1 0 1 1
14	1 1 1 0	1 0 0 1
15	1 1 1 1	1 0 0 0

Truth table of Binary code to Gray code.

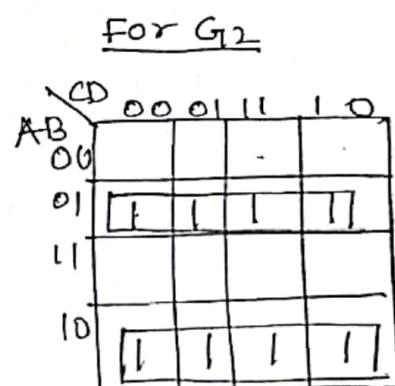
b- Map Simplification:



$$G_0 = A\bar{B} + \bar{A}B \\ = A \oplus B$$



$$G_1 = B\bar{C} + \bar{B}C \\ = B \oplus C$$



$$G_2 = C\bar{D} + \bar{C}D \\ = C \oplus D$$

For Q3

AB	CD	00	01	11	10
00					
01					
11		1	1	1	1
10		1	1	1	1

$$G_3 = A$$

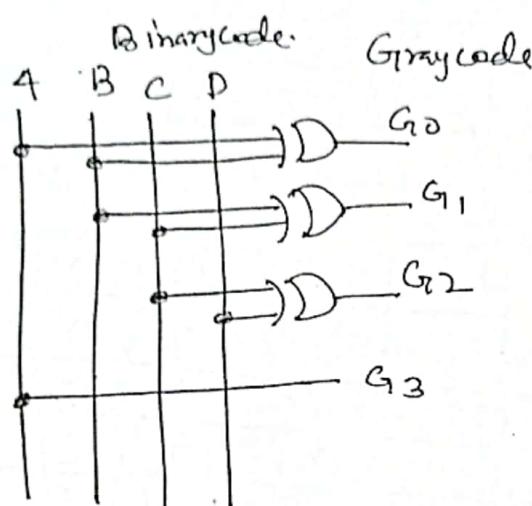


Fig. Logic diagram of Binary to gray code converter.

- * Similarly we can design Gray code to Binary code converter
- Design a code converter circuit to convert BCD to Gray code converter.

BCD code				Gray code			
B ₃	B ₂	B ₁	B ₀	G ₃	G ₂	G ₁	G ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1

Table! Truth table for BCD to gray code converter

K-Map Simplification

For G_0

		00	01	11	10
		B ₃ B ₂	00	01	11
		00	1		1
		01	1	1	
		11	X	X	X
		10	1	X	X

$$G_0 = \overline{B}_1 B_0 + B_1 \overline{B}_0 \\ = B_1 \oplus B_0$$

For G_1

		00	01	11	10
		B ₃ B ₂	00	01	11
		00			1 1
		01	1	1	
		11	X	X	X X
		10			X X

$$G_1 = B_2 \overline{B}_1 + \overline{B}_2 B_1 \\ = B_2 \oplus B_1$$

For G_2

		00	01	11	10
		B ₃ B ₂	00	01	11
		00			
		01	1	1	1
		11	X	X	X X
		10	1	1	X X

$$G_2 = B_2 + B_3$$

For G_3

		00	01	11	10
		B ₃ B ₂	00	01	11
		00			
		01			
		11	X	X	X X
		10	1	1	X X

$$G_3 = B_3$$

Logic diagram

BCD code.

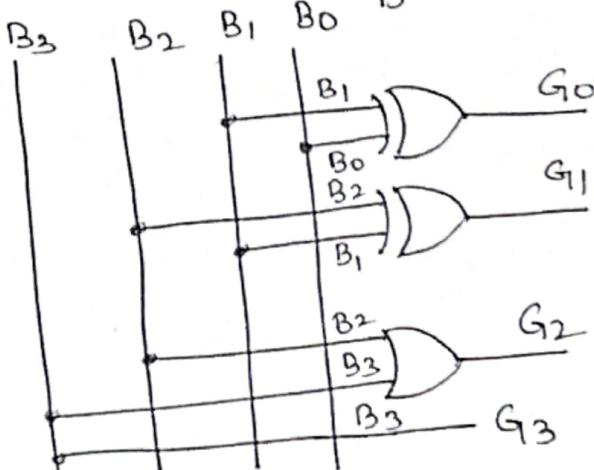


Fig. BCD to gray code converter

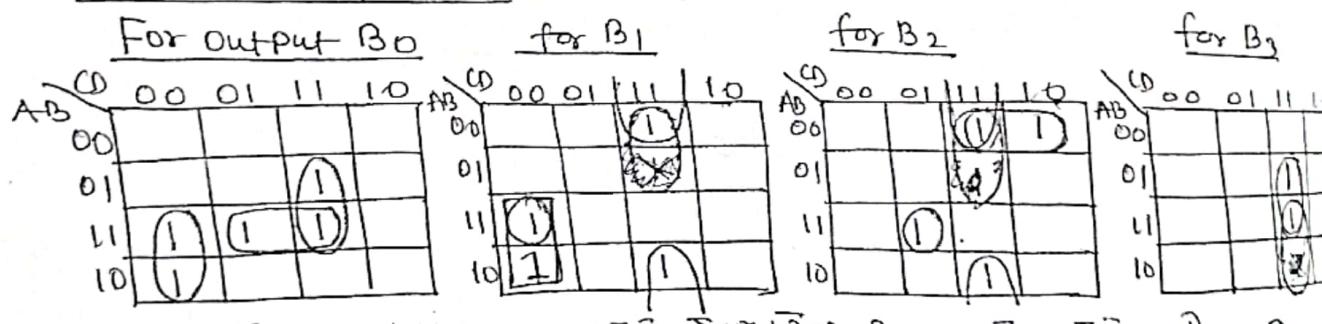
Q) Design a combinational CKT that converts a decimal digit from the 2421 code to BCD (2016 Fall)

=) Let us write the truth table

Decimal Digit	$\begin{smallmatrix} 2 \\ D \end{smallmatrix}$ $\begin{smallmatrix} 4 \\ C \end{smallmatrix}$ $\begin{smallmatrix} 2 \\ B \end{smallmatrix}$ $\begin{smallmatrix} 1 \\ A \end{smallmatrix}$	$\begin{smallmatrix} 8 \\ B_3 \end{smallmatrix}$ $\begin{smallmatrix} 4 \\ B_2 \end{smallmatrix}$ $\begin{smallmatrix} 2 \\ B_1 \end{smallmatrix}$ $\begin{smallmatrix} 1 \\ B_0 \end{smallmatrix}$
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0
3	0 0 1 1	0 0 1 1
4	0 1 0 0	0 1 0 0
5	1 0 1 1	0 1 0 1
6	1 1 0 0	0 1 1 0
7	1 1 0 1	0 1 1 1
8	1 1 1 0	1 0 0 0
9	1 1 1 1	1 0 0 1

Truth Table of 2421 code to BCD code.

K-Map Simplification:



$$B_0 = A\bar{C}\bar{D} + BCD + ABD \quad B_1 = A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{D} + \bar{B}CD \quad B_2 = A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C} + \bar{B}CD \quad B_3 = BCD$$

Finally let us realize the combinational CKT using basic gates.

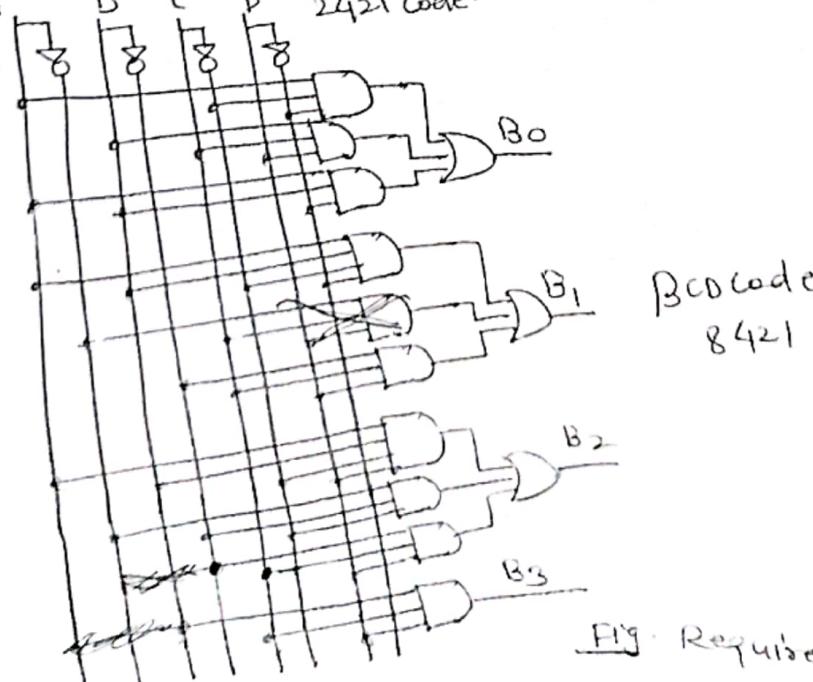


Fig: Required Combinational CKT from 2421 code to BCD

Q. Design a code converter Ckt which converts 84-2-1 C
to binary. [2012 spring]

Sol: First let us write the truth table.

Decimal	8 4 -2 -1	Binary
	D C B A	B ₃ B ₂ B ₁ B ₀
0	0 0 0 0	0 0 0 0
1	0 1 1 1	0 0 0 1
2	0 1 1 0	0 0 1 0
3	0 1 0 1	0 0 1 1
4	0 1 0 0	0 1 0 0
5	1 0 1 1	0 1 0 1
6	1 0 1 0	0 1 1 0
7	1 0 0 1	0 1 1 1
8	1 0 0 0	1 0 0 0
9	1 1 1 1	1 0 0 1

Lk-Map Simplification.

for B₀

AB	CD	00	01	11	10
00	00	1	1		
01	01	1	1		
11	11	1	1	1	1
10	10	1	1	1	1

$$B_0 = AB\bar{0} + A\bar{C}\bar{D} + A\bar{C}D$$

for B₁

AB	CD	00	01	11	10
00	00	1	1		
01	01	1	1		
11	11	1	1	1	1
10	10	1	1	1	1

$$B_1 = \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D}$$

for B₂

AB	CD	00	01	11	10
00	00	1	1		
01	01	1	1		
11	11	1	1	1	1
10	10	1	1	1	1

$$B_2 = \bar{A}\bar{B}C\bar{D}$$

for B₃

AB	CD	00	01	11	10
00	00	1	1		
01	01	1	1		
11	11	1	1	1	1
10	10	1	1	1	1

$$B_3 = \bar{A}\bar{B}\bar{C}D + A\bar{B}D$$

Finally let us realize the combinational Ckt using basic gate we set.

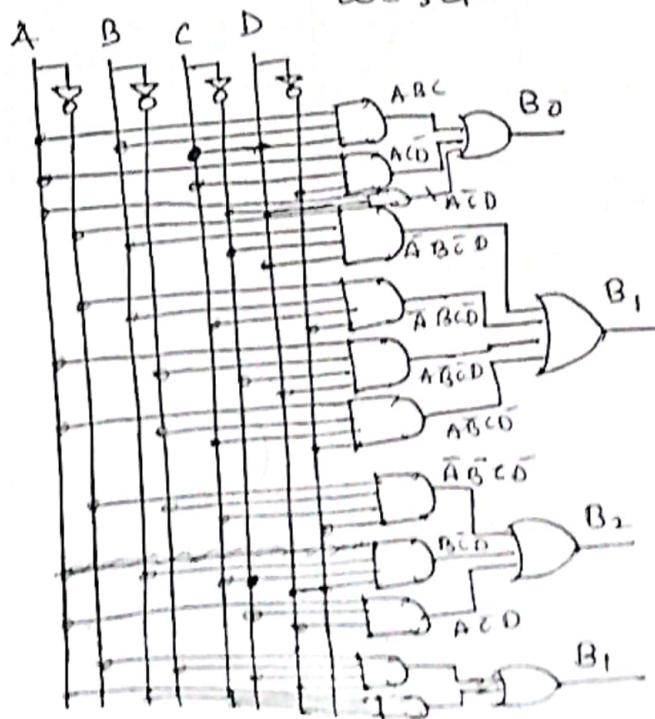


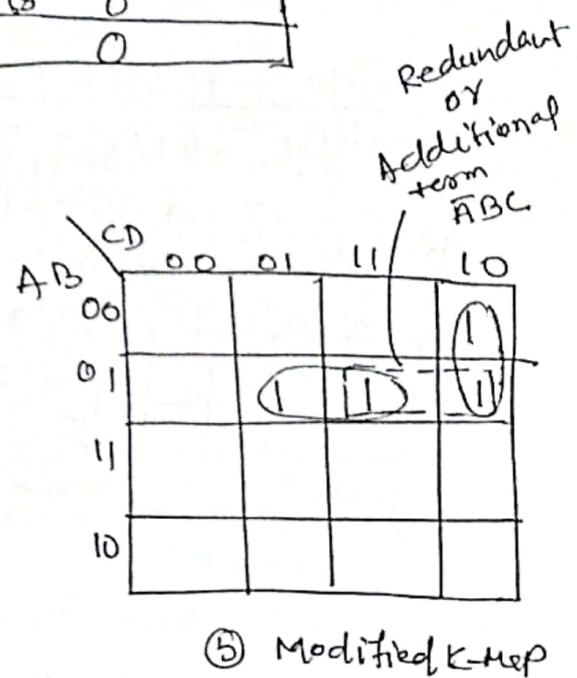
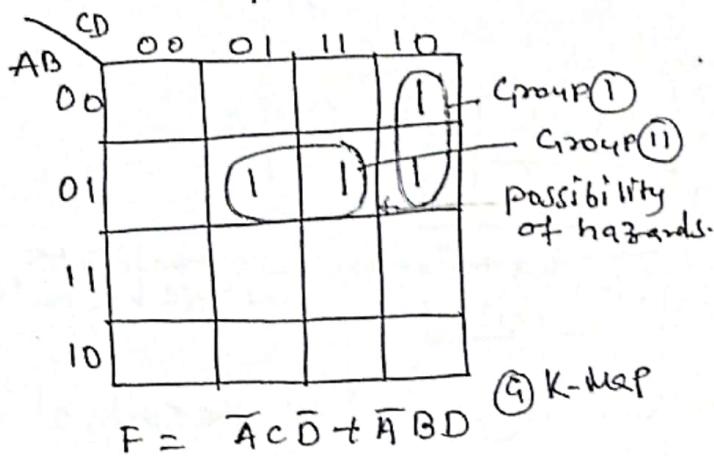
Fig Required Ckt for 84-2-1 code to Binary.

* Design a combinational logic CKT that has 4 inputs numbered 0 through 3, and one output. The output is required to go HIGH whenever any one or more of the input numbers 2, 5, 6 or 7. The CKT should be free of static hazards.

Sol → First let us write the Truth table.

Inputs				Output Y
D	C	B	A	
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0

K-Map simplification
for output Y



In fig. the possible point where hazards can occur are shown above. With the normal grouping the expression of f will be given by

$$F = \overline{A} \overline{C} \overline{D} + \overline{A} \overline{B} D$$

from fig(b) the additional term $\overline{A} \overline{B} C$ introducing in the output expression. Therefore, expression of the O/P of a hazard free CKT is given by.

$$F = \overline{A} \overline{C} \overline{D} + \overline{A} \overline{B} D + \underline{\overline{A} \overline{B} C}$$

New term

The Hazard free CKT is given below

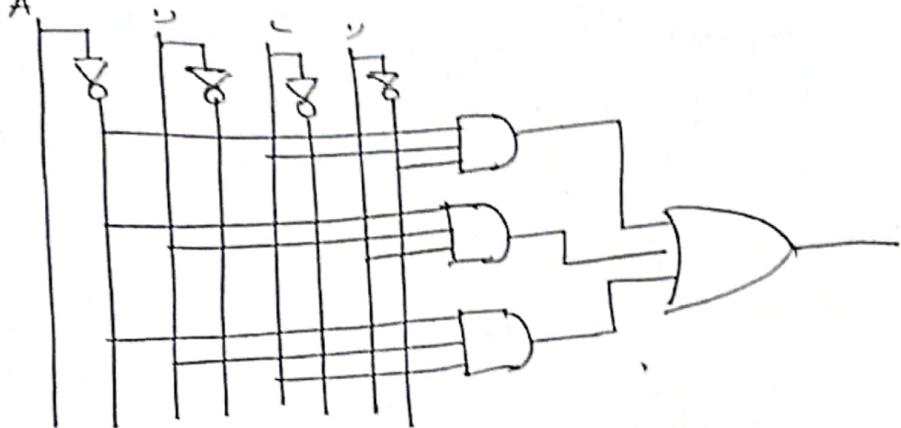
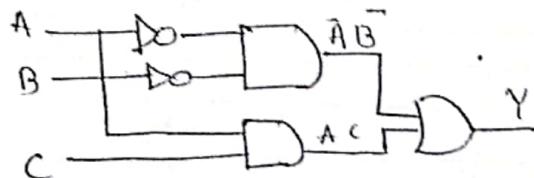


Fig. Realization or Required combinational ckt of Hazard free ckt

* Modify the ckt of fig. to make its hazard free.

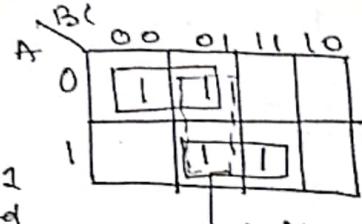
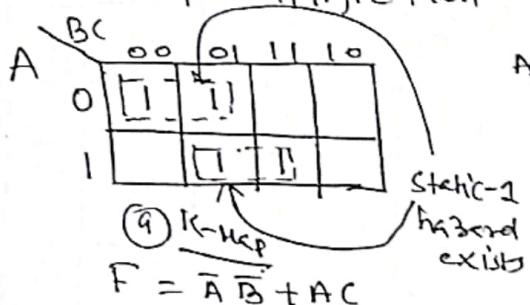


Soln

O/p \neq Y

$$Y = \bar{A}\bar{B} + AC = \bar{A}\bar{B}(C\bar{C}) + AC(B+\bar{B}) \\ = \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + ABC + A\bar{B}C$$

K-map simplification.



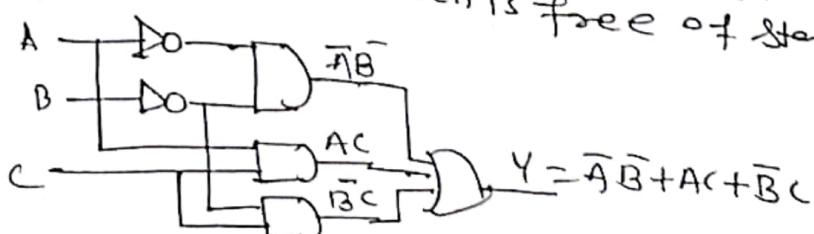
(b) New group $\bar{B}C$ to eliminate static 1-hazard

The K-map is Modified as shown in (b) to eliminate static 1-hazard.

New expression for o/p will be,

$$F = \bar{A}\bar{B} + AC + \bar{B}C$$

The modified ckt which is free of static 1-hazards.



additional gate to avoid static 1-hazard

Fig. Hazard free ckt

Q. Implement the following Boolean function $F = \Sigma(1, 3, 5)$ using 4×1 MUX. (2014 Fall, 2014 Spring, 2015 Fall, Spring, 2016 Spring,

Soln → Here, Given that

$$F = \Sigma(1, 3, 5, 6)$$

Let us assume three Variable they are A, B, C

$$F(A, B, C) = \Sigma(1, 3, 5, 6)$$

Now, Let us write the Truth table of a given function

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Now Next we write the design table or Implementation table.

	D0	D1	D2	D3	
\bar{A}	0	1	2	3	Row1
A	4	5	6	7	Row2
Input to MUX	0	1	A	\bar{A}	

Fig: Design table

Lastly we draw the logic diagram.

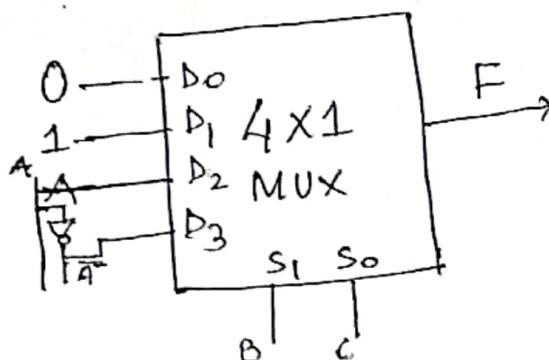


Fig. Logic diagram

Prepared By - Er. Manoj Kr. Singh.

* Construct 8×1 MUX using 4×1 MUX & explain with truth table. [2019 Fall]

Q1 → The cascading of two 4×1 MUX results in 8×1 MUX is shown below:

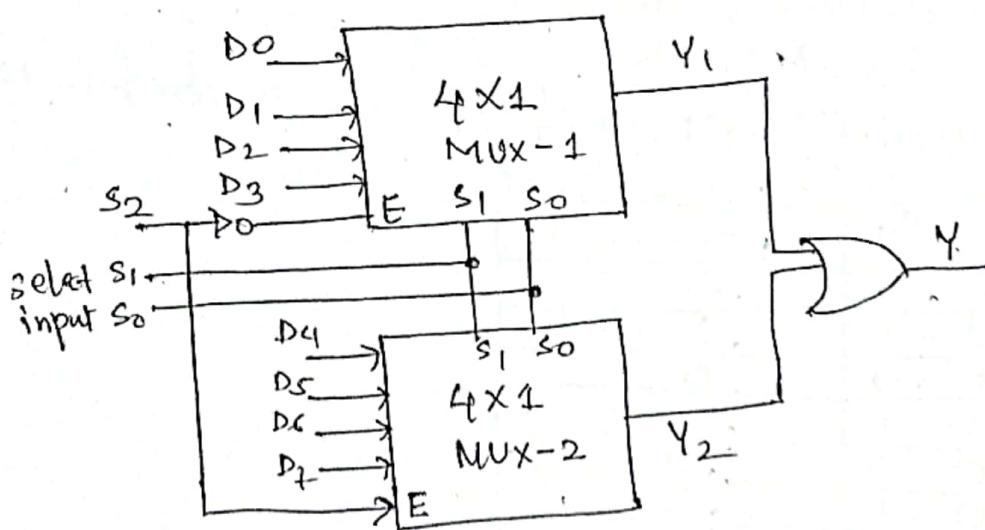


Fig: 8×1 MUX by cascading two 4×1 MUX

Truth Table

Select i/p			O/P
S ₂	S ₁	S ₀	
0	0	0	D ₀
0	0	1	D ₁
0	1	0	D ₂
0	1	1	D ₃
1	0	0	D ₄
	0	1	D ₅
	1	0	D ₆
	1	1	D ₇

MUX1 is enable

MUX2 is enable.

There are in all eight data inputs (D_0 to D_7). The select lines s_1 & s_0 of both 4×1 MUX are connected in parallel whereas a third select i/p s_2 is used for enabling one MUX at a time. The output of the two Multiplexers are ORed to obtain the final o/p Y .

Prepared By:- Er. Manoj Kr. Singh.

Q1. Implement the following logic function using 4X1 MUX
 $F(A, B, C) = \prod M(0, 1, 3, 5, 7)$

Soln Here Given Function is

$$F(A, B, C) = \prod M(0, 1, 3, 5, 7)$$

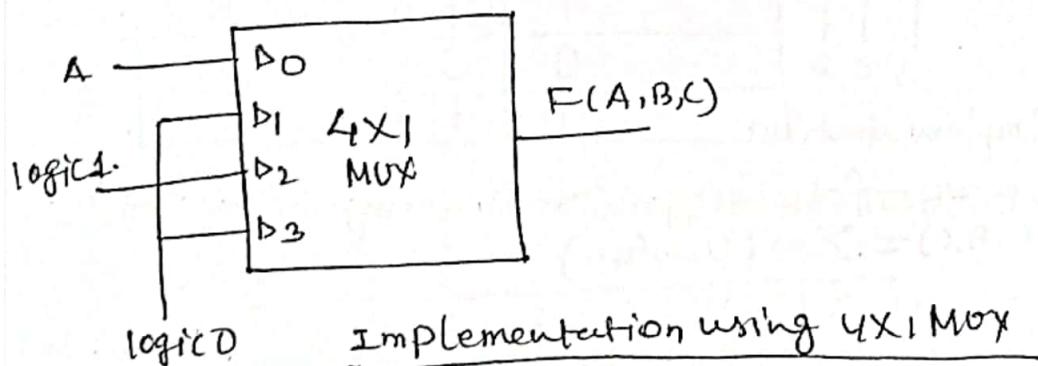
The given expression is in the POS form so, we must enclose those maxterms which are not included in the given boolean expression.

Implementation table

Inputs	D ₀	D ₁	D ₂	D ₃
\bar{A}	0	1	(2)	3
A	(4)	5	(6)	7
Inputs to MUX	A	0	1	0

encircle the maxterms which are not included in the given Boolean expression

Logic diagram,



Implementation using 4X1 MUX

Q2. Implement the following Boolean expression using 8X1 MUX
 $F(A, B, C, D) = \overline{ABD} + ABC + \overline{BCD} + \overline{ACD}$

Soln (i) Let us convert the given expression in standard SOP form
 $\overline{ABC}(C+\bar{C}) + ABC(D+\bar{D}) + \overline{BCD}(A+\bar{A}) + \overline{ACD}(B+\bar{B})$

$$= \overline{ABC}\bar{B} + \overline{ABC}\bar{D} + ABC\bar{D} + ABC\bar{D} + \overline{ABC}\bar{D} + \overline{ABC}\bar{D} + \overline{ABC}\bar{D} + \overline{ABC}\bar{D}$$

$$\text{Now, } F(A, B, C, D) = \sum m(2, 0, 15, 14, 11, 3, 7, 3)$$

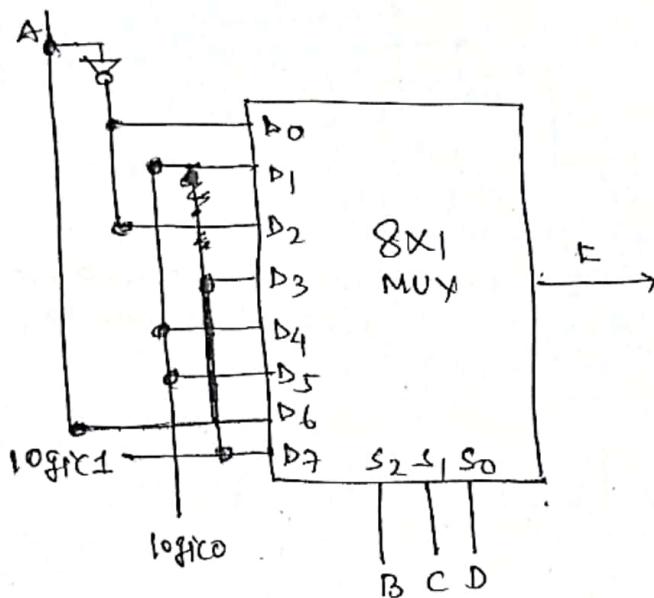
writing in arranging form

$$F(A, B, C, D) = \sum m(0, 2, 3, 7, 11, 14, 15)$$

Implementation table.

Inputs	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	D ₈	D ₉
\bar{A}	①	!	②	③	4	5	6	⑦		
A	8	9	10	⑪	12	13	⑭	⑮		
Input to MUX	A	0	\bar{A}	1	00	A	1			

(iii) logic diagram.



Implementation:

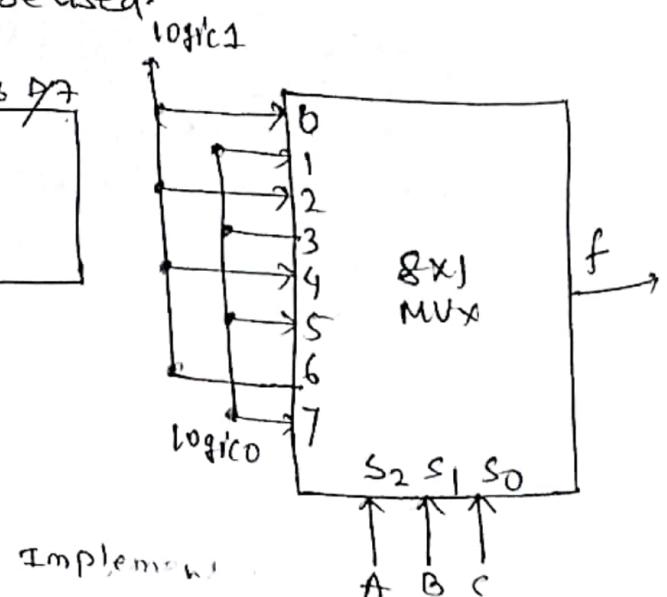
Q Implement the following expression using suitable Multiple
 $F(A, B, C) = \sum m(0, 2, 4, 6)$

Soln $F(A, B, C) = \sum m(0, 2, 4, 6)$

Here, is three Variable E, MUX having → three select i/p must be
therefore 8x1 MUX must be used.

Implementation table

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



Implementation

Q. Implement the following function
 $Y(A, B, C, D) = \sum(0, 1, 2, 5, 9, 11, 13, 15)$. (2012 Fall,

Soln!
Here, Given, $Y(A, B, C, D) = \sum(0, 1, 2, 5, 9, 11, 13, 15)$

First, Let us write the truth table of given function

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

ii) Next we write the implementation table.

	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅ -D ₆	D ₇
A'	0	1	2	3	4	5	6 7
A	8	9	10	11	12	13	14 15
Input to MUX	A'	1	A'	A	0	1	0 A

Table: Implementation Table

Lastly we draw the logic diagram

Next Page. PTO

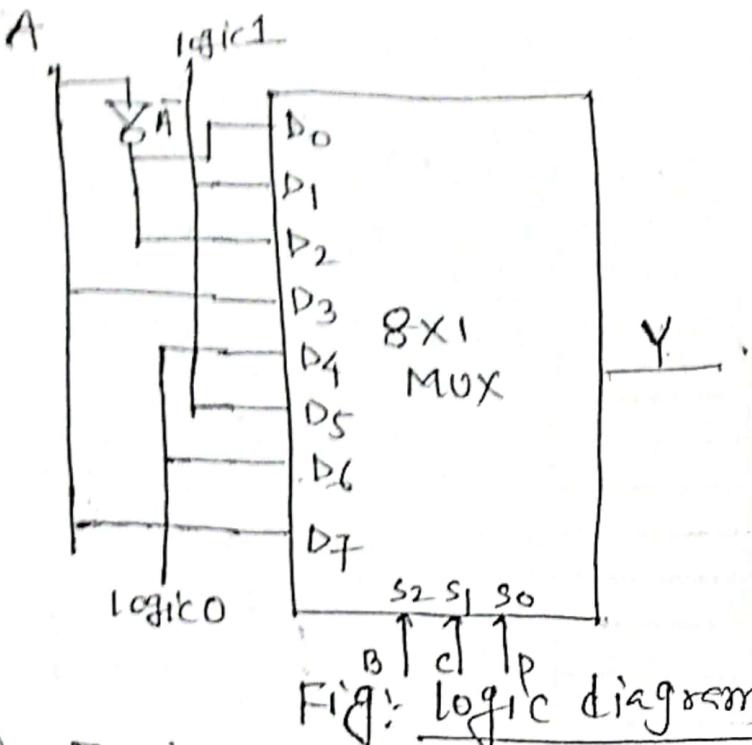


Fig: logic diagram

Q. Implement the following with appropriate MUX.
(iii) $E_1 = A_1 \oplus B_1$

$$(iii) F(A, B, C, D) = \sum_{(0,1,3,4,8,9,15)} (2014 Fall, 2015 Spring)$$

Ques 91 Here Given. $(A, B, C, D) = \{0, 1, 3, 4, 8, 9, 15\}$

Here is Four Variable MUX having four select line so use 16x1 MUX.

Logic 1

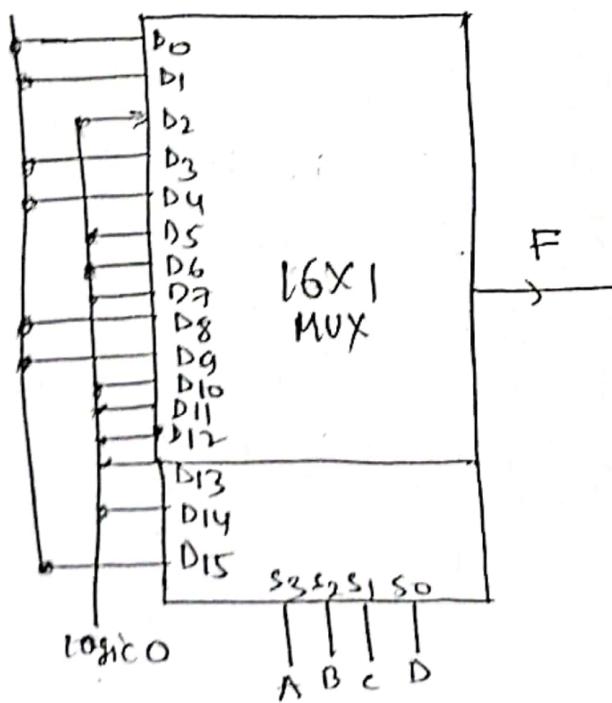


Fig: logic diagram of given function with appropriate MUX (16×1):

Q1) Implement the Full Subtractor using a 1x8 DEMUX

Sol: → The Full subtractor has three inputs A, B and Bin & two O/Ps namely Difference D & borrow out (Bout). The Truth Table of Full subtractor is .

Inputs			Outputs	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Table: Truth Table of a F.S

We express the difference & borrow out outputs in Standard SOP form as

$$D = \sum m(1, 2, 4, 7)$$

$$B_{out} = \sum m(1, 2, 3, 7)$$

We connect D_{in} to logic 1 permanently & connect A, B & Bin to select S_2, S_1, S_0 respectively.

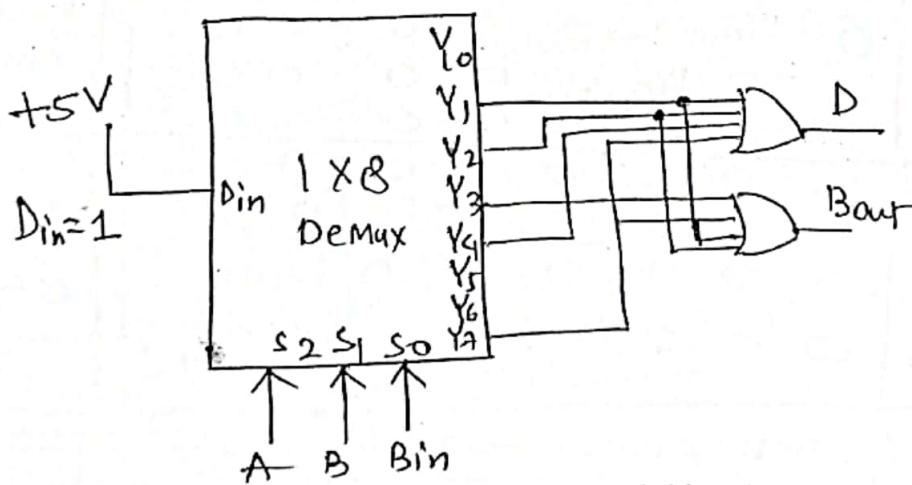


Fig: F.S using 1X8 DEMUX

Q2) Implement a Full adder using Demultiplexers

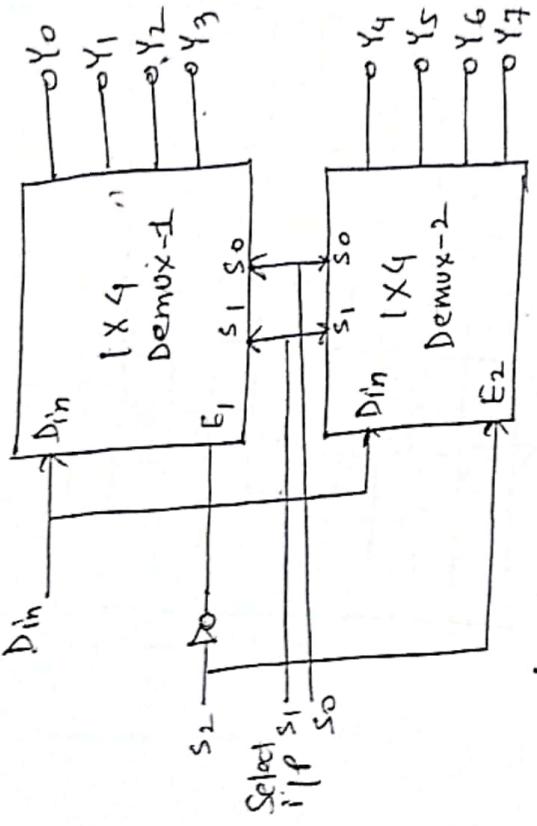
Sol: → Similarly, we can do as Full subtractor.

(D.I.Y.)

Prepared By:- Er. Manoj Kr. Singh.

At Observe a 1x8 line DEMUX using two 1x4 line DEMUX.

Soln:-



1x8 DEMUX using two 1x4 DEMUX

The select line S_1, S_0 of the two 1x4 DEMUX are connected parallel with each other & S_2 is used for selecting of two 1x4 DEMUX. S_2 is connected directly to enable (E) i/p of DEMUX-2 whereas inverted S_2 is connected to the enable i/p of DEMUX-1. The T.T of this circuit is shown below.

S_2	S_1	S_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8
0	0	0	D _{in}	0	0	0	0	0	0	0	0
0	0	1	0	D _{in}	0	0	0	0	0	0	0
0	1	0	0	0	D _{in}	0	0	0	0	0	0
0	1	1	0	0	0	D _{in}	0	0	0	0	0
-	0	0	0	0	0	0	D _{in}	0	0	0	0
-	0	1	0	0	0	0	0	D _{in}	0	0	0
-	1	0	0	0	0	0	0	0	D _{in}	0	0
1	1	1	0	0	0	0	0	0	0	D _{in}	0

Table: T.T of 1x8 DEMUX using two 1x4 DEMUX

Q. Implement a Full adder using 4×1 MUX.

Sol: (i) First let us write the truth table of Full adder

inputs			outputs	
A	B	Cin	Sum(S)	Carry(C)
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

the sum & carry outputs can be expressed in the standard SOP form

$$S = \sum m(1, 2, 4, 7) \quad C = \sum m(3, 5, 6, 7)$$

(ii) Now we write the implementation table for sum & carry output.

For sum

	D ₀	D ₁	D ₂	D ₃
\bar{A}	0	1	2	3
A	4	5	6	7
Input to MUX	A	\bar{A}	\bar{A}	A

For carry

	D ₀	D ₁	D ₂	D ₃
\bar{A}	0	1	2	3
A	4	5	6	7
Input to MUX	0	A	A	1

Table: Implementation table for sum & carry.

(iii) Logic diagram

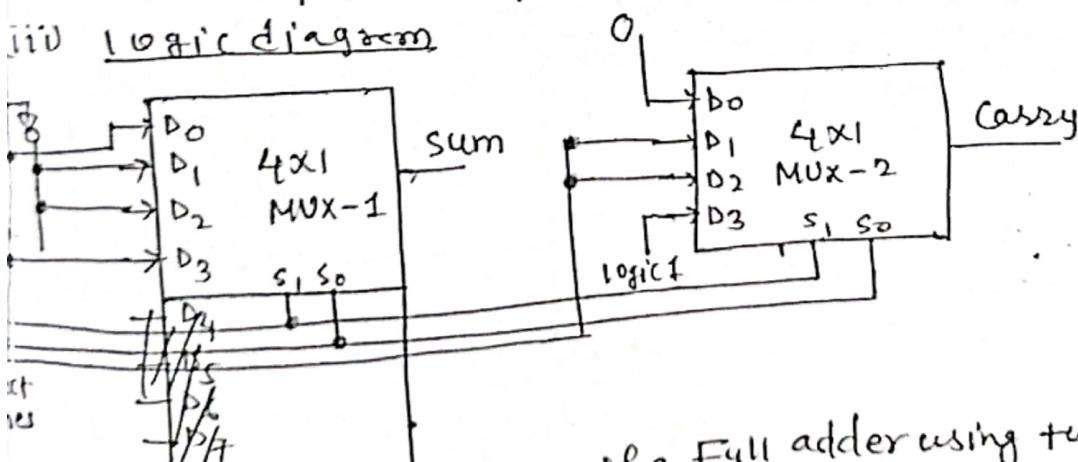


Fig: Implementation of a full adder using two 4×1 MUX.

Q. Implement a Full adder using 8X1 MUX
6019 (1) First let us write the truth table of a F.A

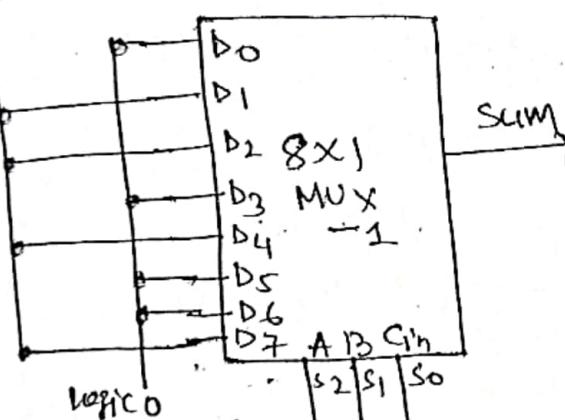
Inputs			Outputs	
A	B	cin	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(ii) The sum and carry can be expressed in the SOP form

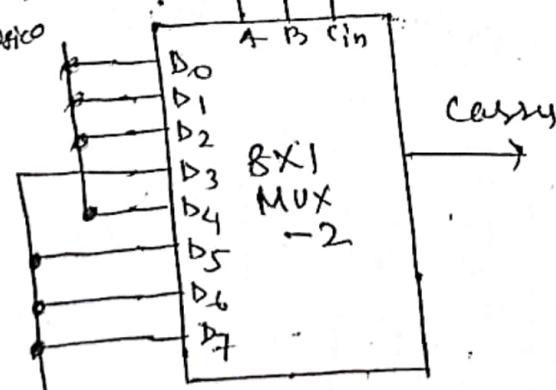
$$S = \sum m(1, 2, 4, 7) \oplus C = \sum m(3, 5, 6, 7)$$

(iii) we realize the ckt using ^{two} 8X1 MUX.

logict1



logict0



logic1

Full adder using 8X1 MUX

Note: (we can ^{also} realize ckt separately for sum & carry)

Remarks

8 inputs +
why
 $2^8 = 2^3$
 $n=3$ sel

Implementation of 16x1 MUX using 4x1 MUX

Implementation is shown below.

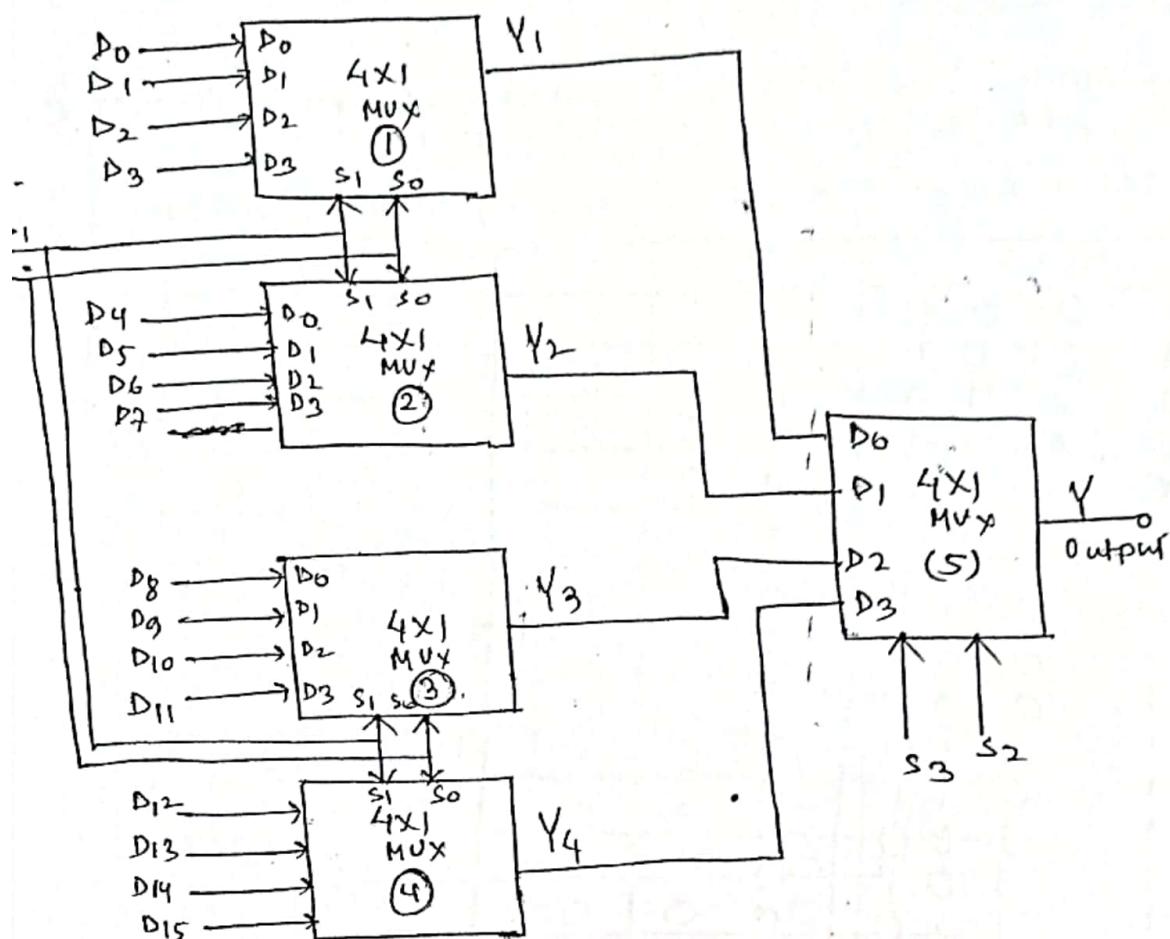


Fig: 16x1 MUX using 4x1 MUX

Function table of above circuit

MUX O/P				Final O/P, Y
S ₃	S ₂	S ₁	S ₀	Y ₁ Y ₂ Y ₃ Y ₄
0	0	0	0	D ₀ D ₄ D ₈ D ₁₂
	0	0	1	D ₁ D ₅ D ₉ D ₁₃
	0	1	0	D ₂ D ₆ D ₁₀ D ₁₄
	0	1	1	D ₃ D ₇ D ₁₁ D ₁₅
1	0	0	0	D ₄ D ₈ D ₁₂ D ₁₆
	0	0	1	D ₅ D ₉ D ₁₃ D ₁₇
	0	1	0	D ₆ D ₁₀ D ₁₄ D ₁₈
	0	1	1	D ₇ D ₁₁ D ₁₅ D ₁₉
-	0	0	0	D ₈ D ₁₂ D ₁₆ D ₂₀
	0	0	1	D ₉ D ₁₃ D ₁₇ D ₂₁
	0	1	0	D ₁₀ D ₁₄ D ₁₈ D ₂₂
	0	1	1	D ₁₁ D ₁₅ D ₁₉ D ₂₃
-	1	0	0	D ₁₂ D ₁₆ D ₂₀ D ₂₄
	1	0	1	D ₁₃ D ₁₇ D ₂₁ D ₂₅
	1	1	0	D ₁₄ D ₁₈ D ₂₂ D ₂₆
	1	1	1	D ₁₅ D ₁₉ D ₂₃ D ₂₇

S₃S₂=0
Therefore
MUX-S
selects Y

S₃S₂=0
MUX-S
selects Y

S₃S₂=1
Therefore
MUX-S
selects Y

S₃S₂=1
Therefore
MUX-S
selects Y

The select input S_1 & S_0 of MUX 1, 2, 3 & 4 are connected together. The select i/p S_3 & S_2 are applied to MUX 5. The o/p Y_1, Y_2, Y_3, Y_4 are applied to data i/p D_0, D_1, D_2, D_3 of MUX-5 as shown in fig.

Q. Implement the following function with multiplexers
 Sol. $F(A, B, C, D) = \sum(0, 3, 4, 8, 9, 15)$
 First let us write the Truth table of given function.

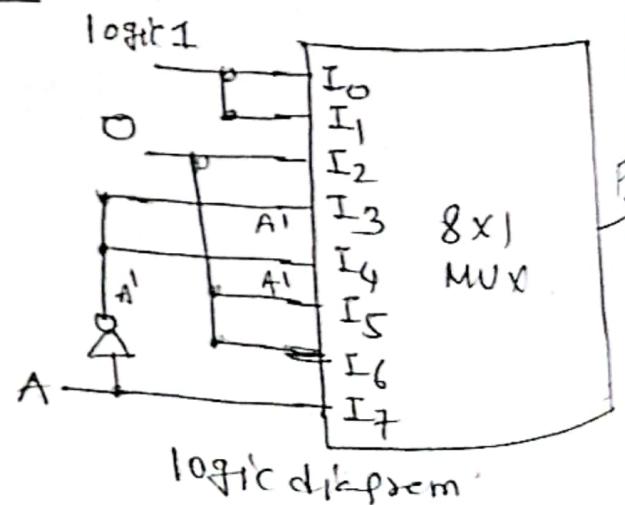
A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Table → T.T

Implementation Table.

i/p	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
A'	6	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
i/p to MUX	1	1	0	A'	A'	0	0	A

Finally draw the logic diagram

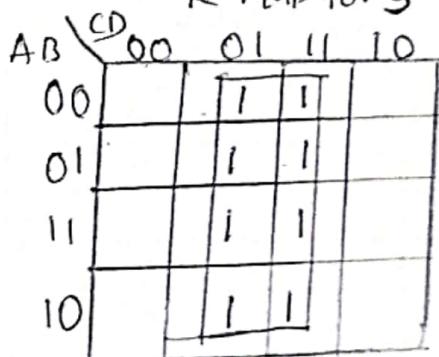


Q. Design a combinational circuit to produce +
2's complement of a 4-bit binary number.

Soln:

Decin NO.	A	B	C	D	P	Q	R	S
0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1	1
2	0	0	1	0	1	1	1	0
3	0	0	1	1	1	1	0	1
4	0	1	0	0	1	1	0	0
5	0	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	0
7	0	1	1	1	1	0	0	1
8	1	0	0	0	1	0	0	0
9	1	0	0	1	0	1	1	1
10	1	0	1	0	0	1	1	0
11	1	0	1	1	0	1	0	1
12	1	1	0	0	0	1	0	0
13	1	1	0	1	0	0	1	1
14	1	1	1	0	0	0	1	0
15	1	1	1	1	0	0	0	1

K Map for S



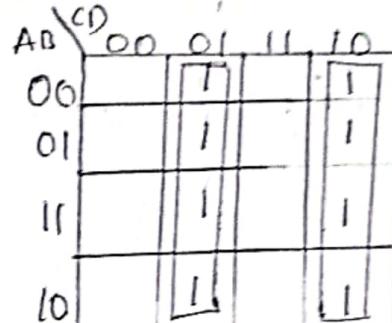
$$S = D$$

K-Map for Q



$$Q = \overline{B}C + \overline{B}D + B\overline{C}\overline{D}$$

K Map for R

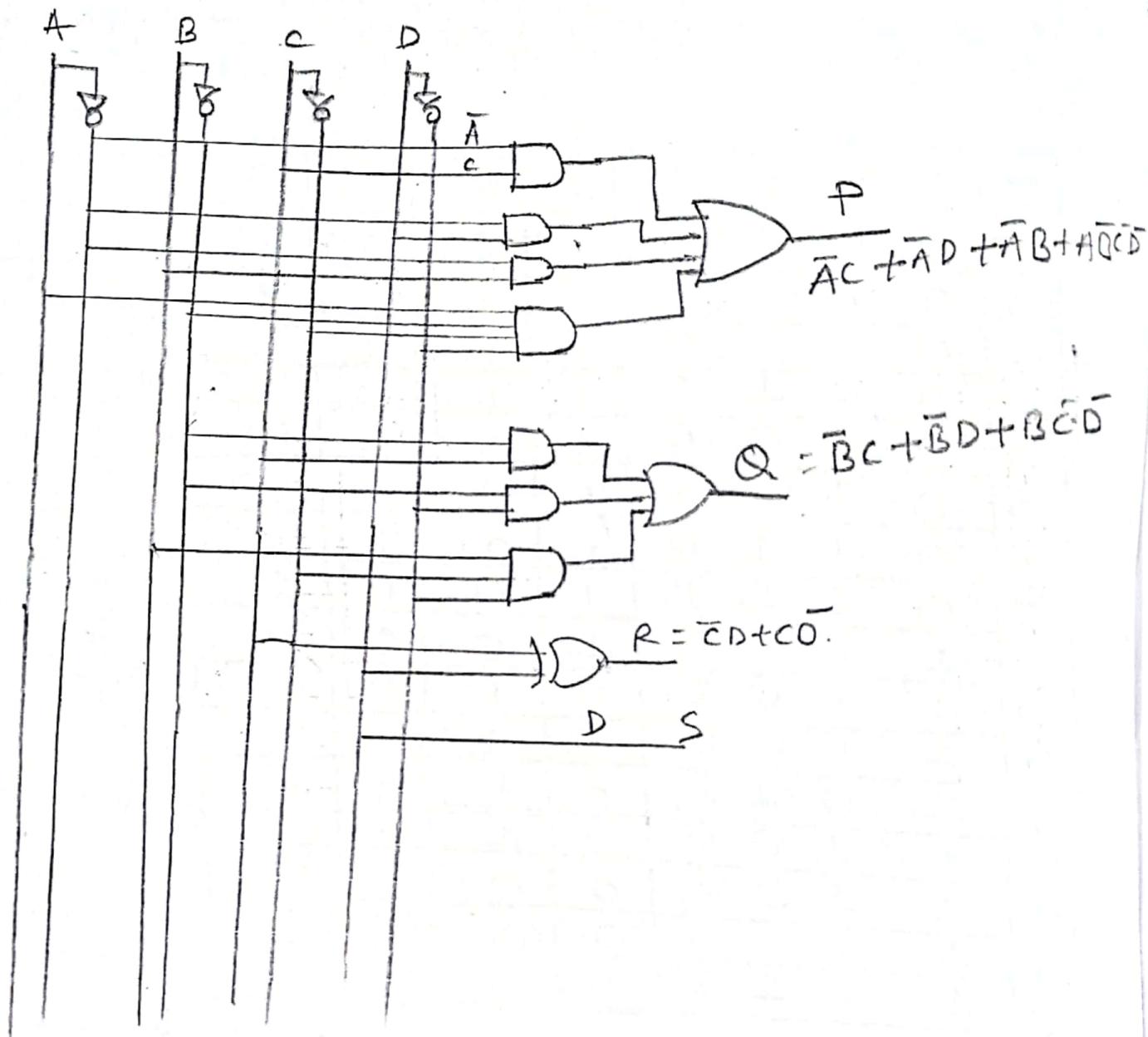


$$R = \overline{CD} + C\overline{D} = C \oplus D$$

K-Map for P



$$P = \overline{A}C + \overline{A}D + \overline{A}\overline{B} + A\overline{B}\overline{C}\overline{D}$$



Q. Design 3-bit Synchronous Upcounter by using JK P/F

Sol:

$$3 \cdot 8^{\frac{n}{2}} = 3 \cdot 8^{\frac{3}{2}} = 3 \cdot 8^{\frac{3}{2}} = 3 \cdot 8^{\frac{3}{2}} = 3 \cdot 8^{\frac{3}{2}}$$

$$\text{No of state } N = 2^n$$

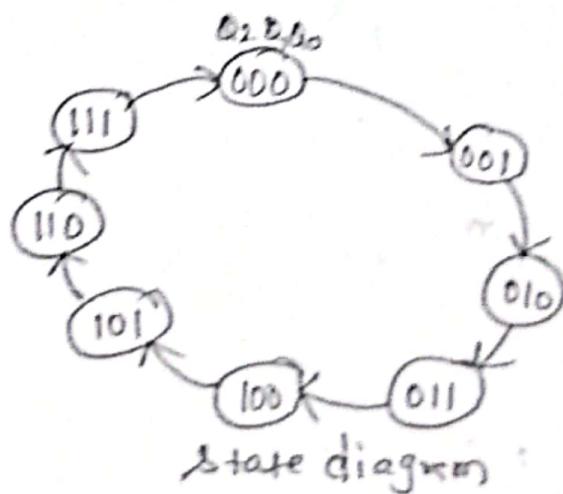
$$n = 3 \quad N = 2^3 = 8 \text{ state}$$

$$\text{Max count } 2^n - 1 = 2^3 - 1 = 8 - 1 = 7$$

Count 0 to 7

Excitation Table of JK P/F

Q(t)	Q(t+1)	JK
0	0	0 X
0	1	1 X
1	0	X 1
1	1	X 0



Ckt Excitation Table

Present State Q ₂ Q ₁ Q ₀	Next State Q ₂ Q ₁ Q ₀	J ₀ K ₀	J ₁ K ₁	J ₂ K ₂
0 0 0	0 0 1	1 X	0 X	0 X
0 0 1	0 1 0	X 1	1 X	0 X
0 1 0	0 1 1	1 X	X 0	0 X
0 1 1	1 0 0	X 1	X 1	0 1 X
1 0 0	1 0 1	1 X	0 X	X 0
1 0 1	1 1 0	X 1	1 X	X 0
1 1 0	1 1 1	1 X	X 0	X 0
1 1 1	0 0 0	X 1	X 1	X 1

		for J_0			
$Q_2 Q_1$		00	01	11	10
Q_0	0	0	1	1	0
1	1	(X)	1	0	0
		3	X	X	X

$J_0 = 1$

		for J_1			
$Q_2 Q_1$		00	01	11	10
Q_0	0	0	0	X	0
1	1	(X)	1	0	0

$J_1 = Q_0$

		for J_0			
$Q_2 Q_1$		00	01	11	10
Q_0	0	X	X	X	X
1	1	1	1	1	1

$K_0 = 1$

		for J_2			
$Q_2 Q_1$		100	01	11	10
Q_0	0	0	X	X	
1	1	(X)	X	X	

$J_2 = Q_1 Q_0$

		for J_1			
$Q_2 Q_1$		100	01	11	10
Q_0	0	X	0	0	X
1	(X)	1	1	X	

$K_1 = Q_0$

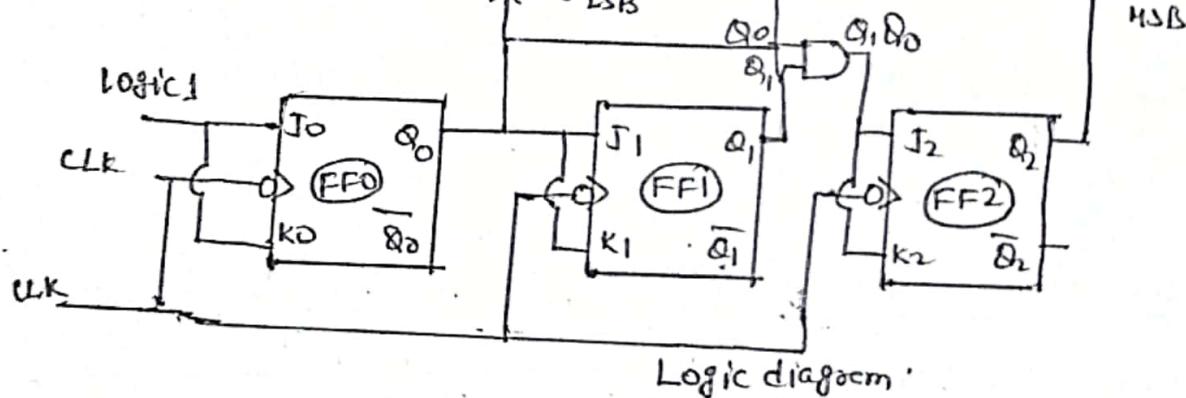
		for K_2			
$Q_2 Q_1$		00	01	11	10
Q_0	0	X	X	0	0
1	(X)	1	0	0	0

$K_2 = Q_1 Q_0$

$$J_0 = K_0 = 1$$

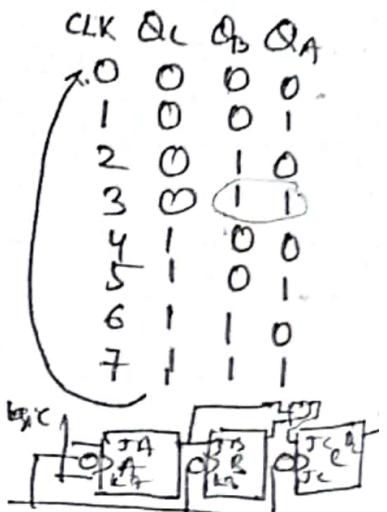
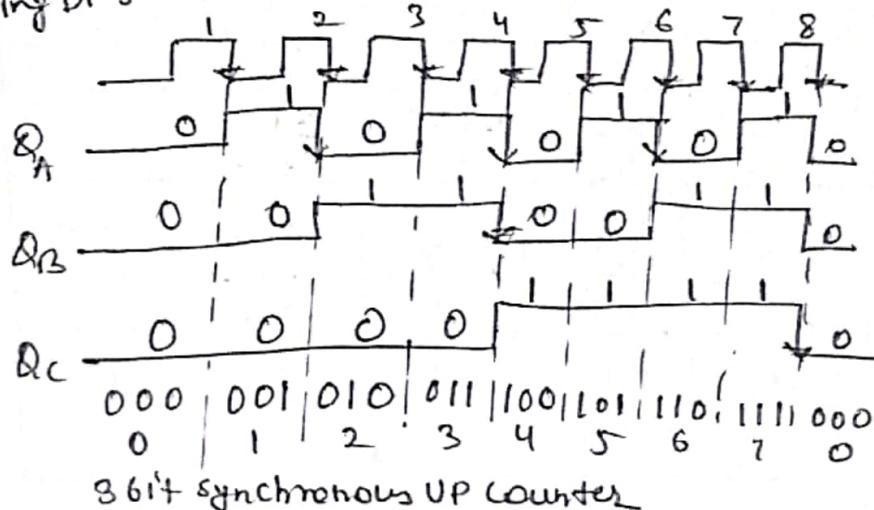
$$J_1 = K_1 = Q_0$$

$$J_2 = K_2 = Q_1 Q_0$$



Note:

Timing Diagrams:



Here, Q_A changes on each CP as we progress from its original state to its final state & then back to its original state. To produce this operation $F/F A$ is held in the toggle mode by connecting $J + K$ i/p to high, for $F/F B$ toggles when Q_A is 1. When Q_A is 0 $F/F B$ is in no-change mode. Similarly its $F/F C$ has to change only when $Q_B + Q_A$ are at 1.

a. Design 2-bit synchronous UP Counter using JK FF.

b1 →

2-bit = 2 Flip Flop

No. of states $N = 2^n$

$n = 2 \quad N = 2^2 = 4$ state

Max. Count $2^n - 1 = 2^2 - 1 = 4 - 1 = 3$

Count = 0 to 3

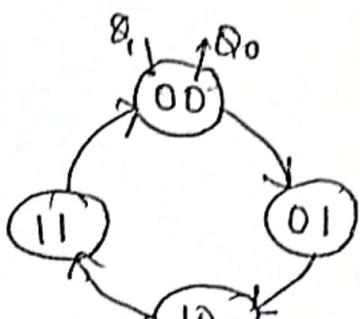


Fig: state diagram

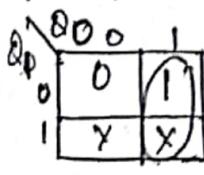
Excitation Table of JK FF

$J(U)$	$K(U)$	$J(K)$
0	0	$\bar{Q}X$
0	1	$1\bar{X}$
1	0	$X1$
1	1	$X0$

JK Excitation Table.

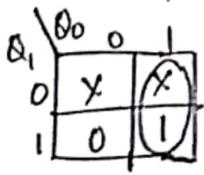
PS		NS		I/P	
Q_1	Q_0	Q_1	Q_0	J_1	K_1
0	0	0	1	0	\bar{X}
0	1	1	0	1	\bar{X}
1	0	1	1	\bar{X}	0
1	1	0	0	\bar{X}	1

for J_1



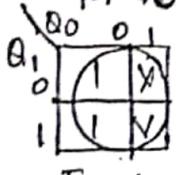
$$J_1 = Q_0$$

for K_1



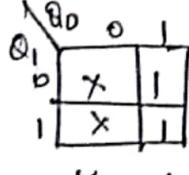
$$K_1 = Q_0$$

for J_0



$$J_0 = 1$$

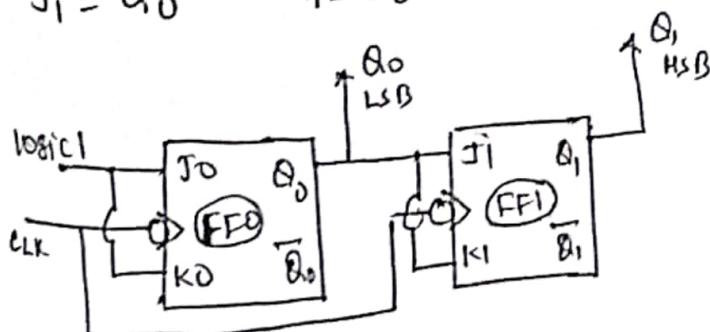
for K_0



$$K_0 = 1$$

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0$$



Logic diagram.