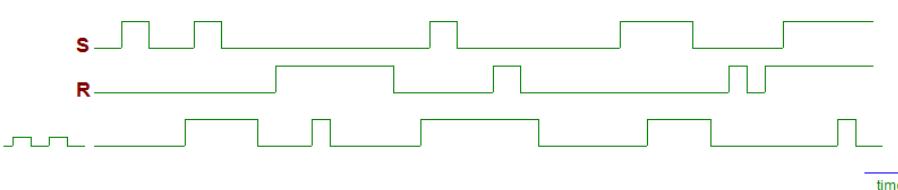


QUESTION BANK

UNIT-II

1	Differentiate between i) Combinational Logic circuits and sequential logic circuits ii) Latch and Flip flop
2	Write short note on basic bistable element
3	With the neat logic diagram and function table briefly explain the operation of i) JK Flip flop ii) T Flip flop iii) SR Flip Flop iv) D Flip Flop
4	With neat logic diagram and function table explain D Flip flop and also write its timing diagram.
5	Draw the output for the S, R and clock input as shown in below figure. (consider initial state as 0). 
6	Explain the operation of Master Slave JK flip flop with truth table and timing diagram. Write the symbol for the same.
7	Implement $f(a, b, c) = \sum m(1,4,5,7)$ using i) 8:1 MUX ii) 4:1 MUX using a and b as select lines.
8	Realize the following Boolean function using the appropriate multiplexer and external gates. i) $f(a, b, c, d) = \sum m(0,1,5,6,7,9,10,15)$ with a, b, c as select lines. ii) $f(w, x, y, z) = \sum m(1,3,5,7,8,9,11,13,14)$ with w, x, y as select lines iii) $f(a, b, c) = \sum m(1,3,6,7)$ with a, b as select lines and a,b,c as select lines. iv) $f(w, x, y) = \sum m(0,2,3,6)$ with w,x as select lines and w,x,y as select lines.
9	Implement $u = ad + \overline{bc} + bd$ using 8:1 MUX.
10	Implement the given function using decoder with minimum number of gate inputs. i) $F1 = \sum m(0,1,3)$ ii) $F2 = \sum m(0,2,3,5,7)$ iii) $F3 = \sum m(1,3,5,6,7)$ iv) $F4 = \pi M(1,2,3,5,6,7)$

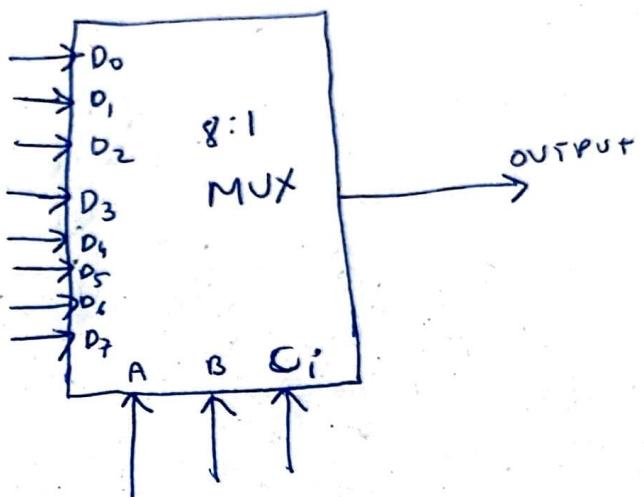
	v) $F5 = ab + bc + ac$ vi) $F6 = \pi M(0,3)$
11	Design 2:4 and 3: 8 decoder.
12	Explain the working of a 4-bit binary parallel adder/subtractor using XOR gates with a relevant block diagram and with an example.
13	What are the characteristic equations of Flip flop? Obtain the characteristic equation for i) JK Flip flop ii) T Flip flop iii) SR Flip Flop iv) D Flip Flop
14	Design a combinational logic circuit to compare two 2 bit numbers.
15	Explain octal to binary (8-to-3-line encoder) conversion using an encoder
16	Implement half adder using 2to4 line decoder.
17	Explain the working of the 3-to-8-line decoder.
18	Design a 4-to-1-line multiplexer using basic gates and implement 1 bit comparator using 4:1 MUX.
19	Implement (a) $u = a + b\bar{c} + b$ using 8:1 and 4:1 MUX. (b) $F(A, B, C) = \Sigma m(0,1,3,4)$ using 4:1 MUX and 8:1 MUX
20	Implement the given function using a decoder with a minimum number of gate inputs. i) $F1 = \sum m(0,1,3)$ ii) $F2 = \sum m(0,2,3,5,7)$ iii) $F3 = \sum m(1,3,5,6,7)$ iv) $F4 = \pi M(1,2,3,5,6,7)$ v) $F5 = ab + bc + ac$ vi) $F6 = \pi M(0,3)$
23	With neat logic diagram and function table explain the working of SR latch and D latch and also write its timing diagram.
21	Design half subtractor and full subtractor circuits using logic gates.
22	Design 1-bit comparator using basic gates
23	Implement the following expressions using 4:1 multiplexer i) $f(x, y) = \Sigma m(1,2)$ ii) $f(a, b, c) = a+b+c$
24	Implement the following expressions with 3-to-8-line decoder with minimum number of gate inputs i) $f_1(x, y, z) = \bar{x}\bar{y} + z$ ii) $f_2(x, y, z) = \Sigma m(0,2,3,4,7)$ iii) $f_3(x, y, z) = \Pi M(0,2,3,4,7)$ iv) $f_4(x, y, z) = xy^+yz$

25	Explain the working of basic bistable element and write the symbol, logic diagram and function table of Set- Reset latch made with NOR gates.
26	Convert JK flip flop to T flip flop.
27	Convert JK flip flop to D flip flop.
28	Implement the given function using an active high decoder: i. $F1(a,b,c)=\sum m(1,2,6)$ ii. $F2(a,b,c)= \sum m(0)$ iii. $F3(a,b,c)= \sum m(3,4,5)$
29	Explain the working of 4 to 2 Encoder
30	Explain with an example the working of 4-bit binary parallel adder/subtractor with relevant block diagram.
31	Explain the working of D flip flop with relevant logic diagram and function table.
32	Explain the working of 2x1 Multiplexer.
33	Implement a Full Adder using a 4x1 multiplexer.
34	Implement $u=a+bc'$ using 4:1 Mux.
35	Design a circuit using 3:8 decoder and OR gates that realizes following functions- $f1(A,B,C)=\sum m(0,4,6)$, $f2=A'+AC$
36	Write a note on i. Latch ii. Flip-flop iii. Characteristic equation of D flip-flop
37	With a neat diagram illustrate the design of full adder using two half adders.
38	Implement the function $F(A,B,C,D)=\sum m(6,7,9,10,13)$ using 8:1 multiplexer

$$\Rightarrow \beta(a, b, c) = \sum m(1, 4, 5, 7)$$

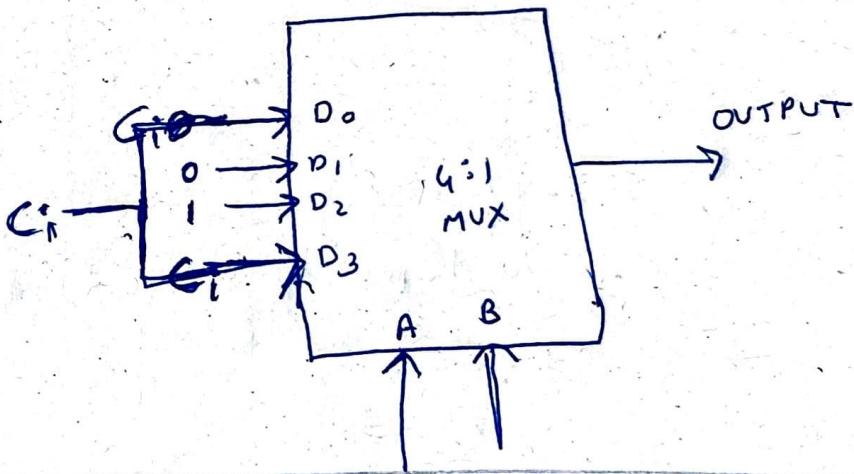
8:1 MUX

I/P			O/P
A	B	C _i	D
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1



4:1 MUX :-
Reduced truth table :-

I/P		O/P
A	B	D
0	0	C _i
0	1	0
1	0	1
1	1	C _i



8]

$$F_6 = \pi M(0, 3)$$

a, b, c

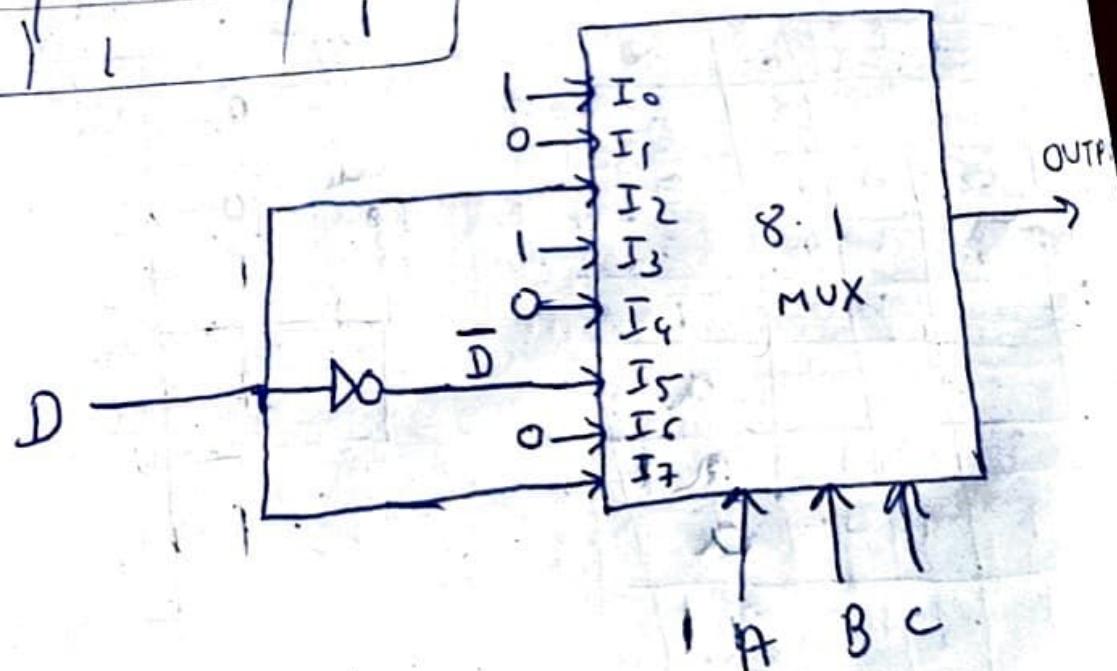
$$1) F(a, b, c, d) = \sum m(0, 1, 5, 6, 7, 10, 15)$$

reduced tr. table

A	B	C	D	I
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

A	B	C	I
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

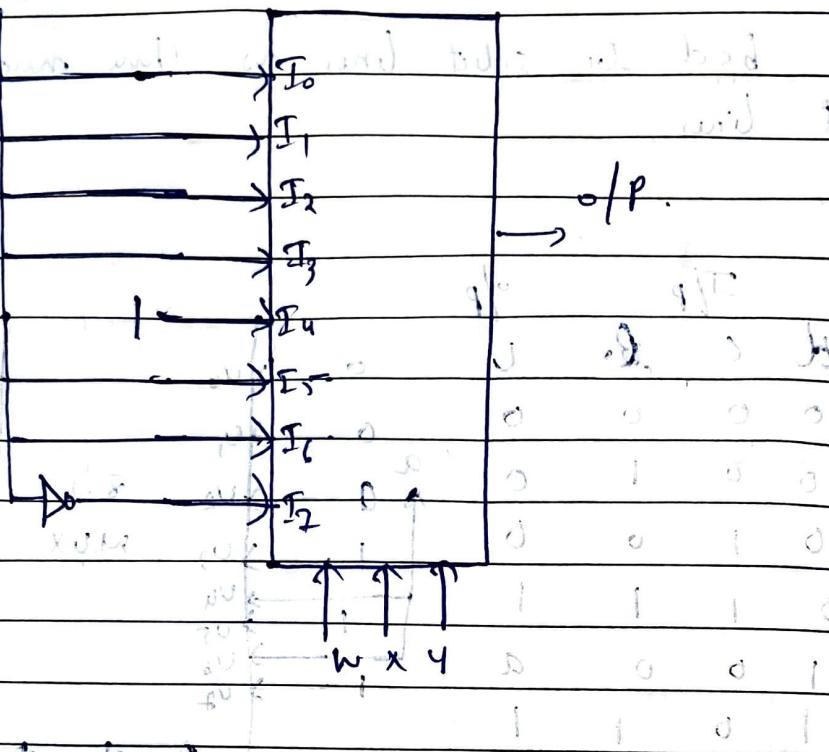
8:1 MUX



$$8) \text{ i) } F(a, b, c, d) = \sum m(1, 3, 5, 7, 8, 9, 11, 13, 14)$$

w	x	y	z	I					
0	0	0	0	0	{	1	1	0	0
0	0	1	0	1	{	1	1	0	0
0	0	0	1	0	0	1	1	0	0
0	0	1	1	1	{	1	1	0	0
0	1	0	0	0	0	1	1	0	0
0	1	0	1	1	{	1	0	1	0
0	1	1	0	0	0	1	1	1	2
0	1	1	1	1	{	1	0	0	1
1	0	0	0	1	1		1	0	1
1	0	0	1	1	1		1	1	0
1	0	1	0	0	0		1	1	1
1	0	1	1	1	1				
1	1	0	0	0	0				
1	1	0	1	1	1				
1	1	1	0	1	1				
1	1	1	1	0	1				

Reduced OT-T.



$$8 \quad (\text{ii}) \quad f(a, b; c) = \text{Em}(1, 3, 6, 7) \quad a, b \text{ as Schit lines} \\ \text{and } ab \perp c$$

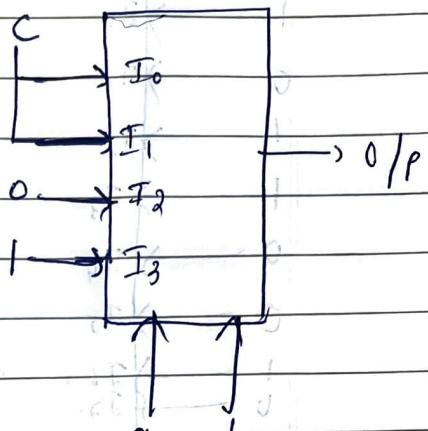
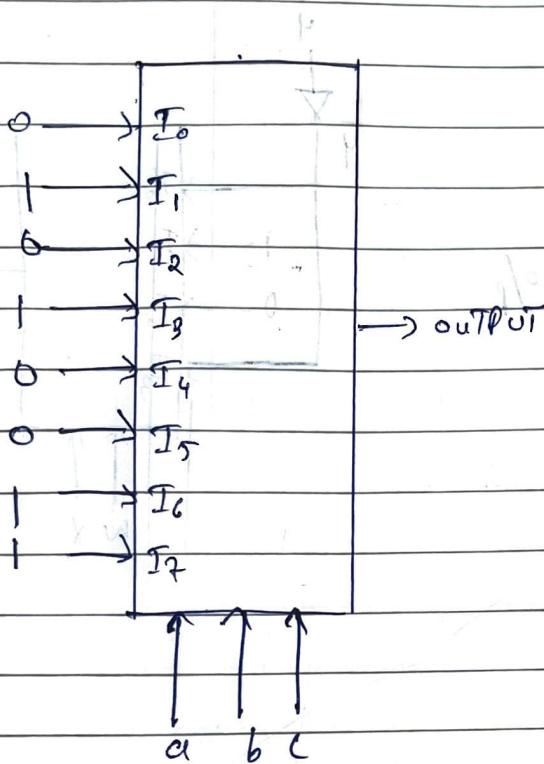
a b c I Am Redundant, T-F (in H) nos.

0	0	1	1	{	1	1	a	b	x	I
0	1	0	0	{	0	0	0	0	0	C
0	1	1	1	{	1	1	0	1	1	C
1	0	0	0	{	0	0	1	0	0	
0	1	0	0	{	1	1	1	1	1	
1	1	0	1	{	0	0	0	0	0	
0	1	0	1	{	1	1	0	1	1	

8. ii) A

Mux.

Ans: 1:6 MUX



8. iv) $f(w, x, y) = \sum m(0, 2, 3, 6)$ with w, x as select lines
and w, x, y " "

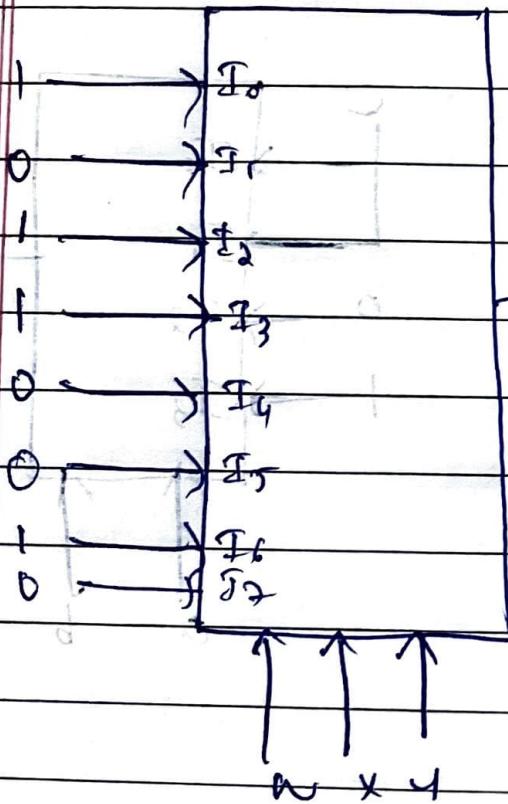
w	x	y	I
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

P.T.T for 4:1 mux.

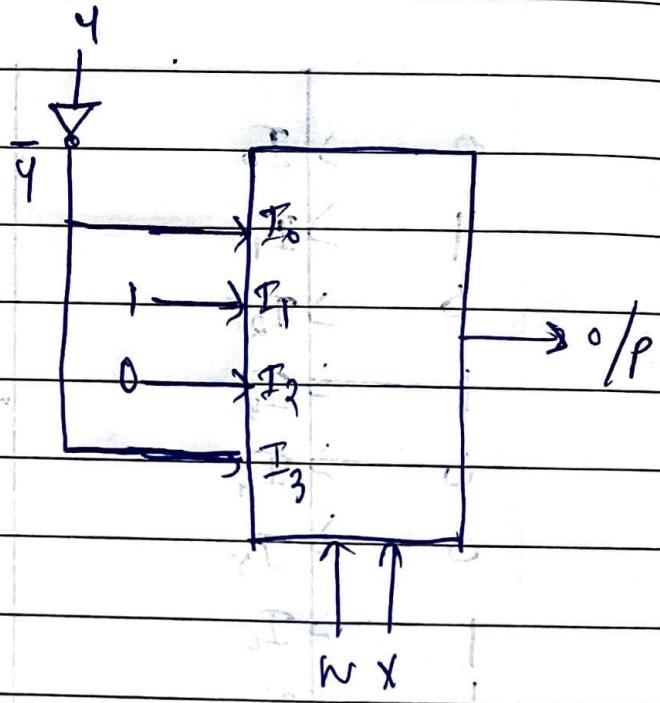
w	x	I
0	0	4
0	1	1
1	0	0
1	1	4

8:1 max.

4:1 max

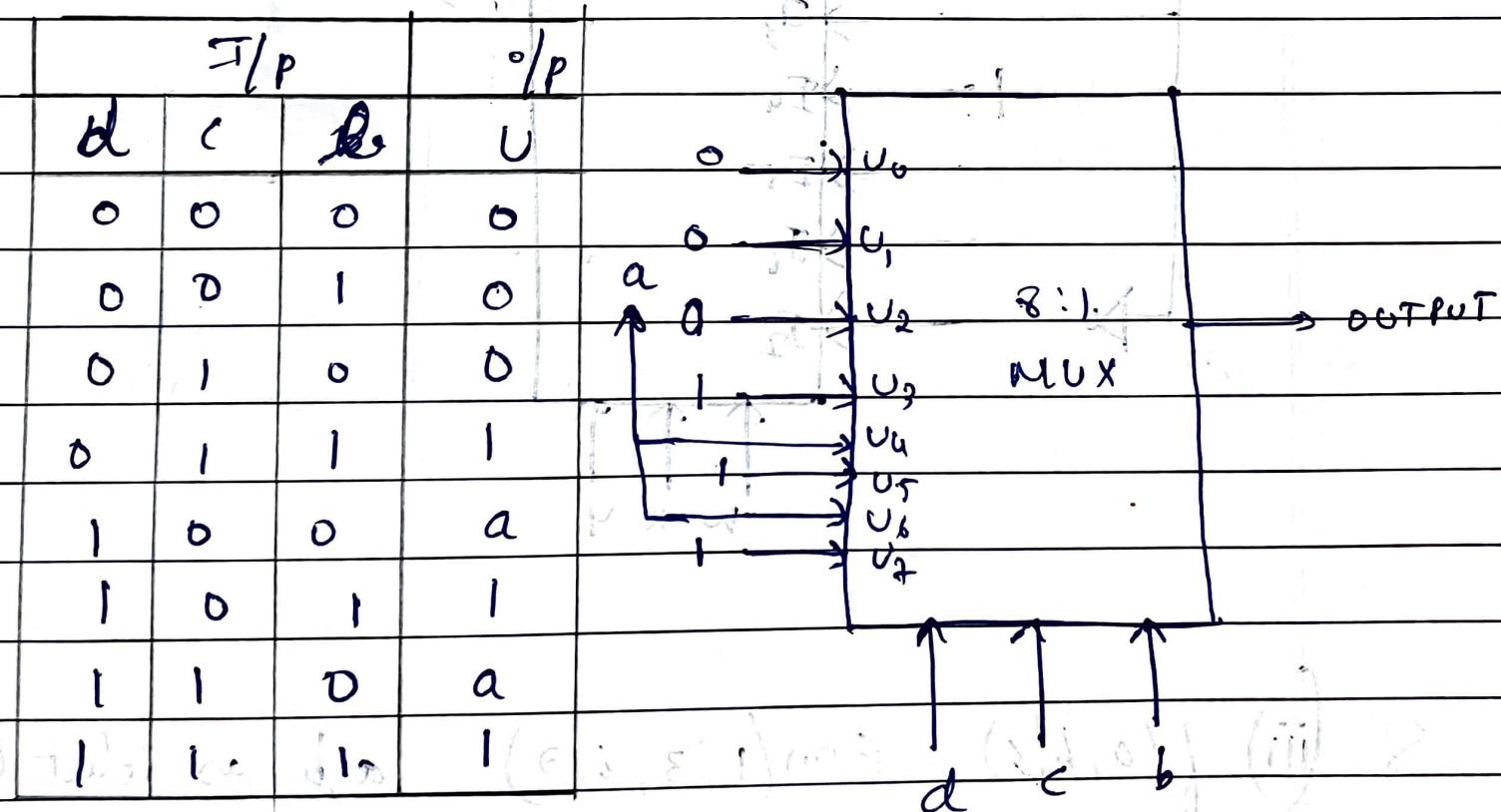


$\rightarrow \frac{\partial}{\partial}$
Dyadic C

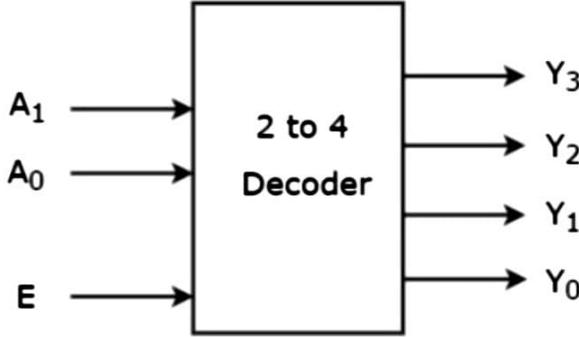


9) Implement $U = ad + bc + bd$ using 8:1 MUX

Ans Let bcd be select lines as there needs to be 3 select lines

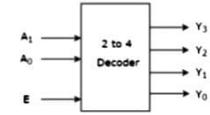


2 to 4 Line Decoder (Active High Outputs)



43

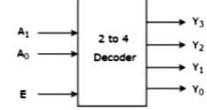
2 to 4 Line Decoder



Enable		Inputs		Outputs			
E	A ₁	A ₀		Y ₃	Y ₂	Y ₁	Y ₀
0	x	x		0	0	0	0
1	0	0		0	0	0	1
1	0	1		0	0	1	0
1	1	0		0	1	0	0
1	1	1		1	0	0	0

44

2 to 4 Line Decoder

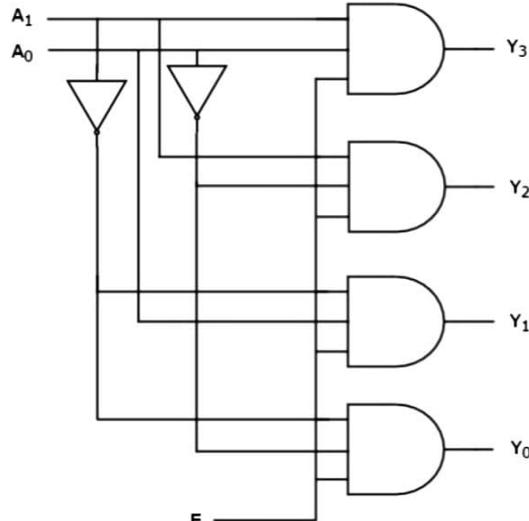


$$Y_3 = E \cdot A_1 \cdot A_0$$

$$Y_2 = E \cdot A_1 \cdot A_0'$$

$$Y_1 = E \cdot A_1' \cdot A_0$$

$$Y_0 = E \cdot A_1' \cdot A_0'$$

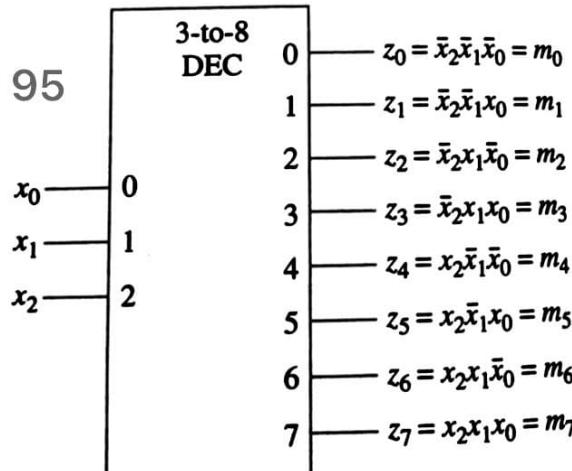


45

3 to 8 decoder

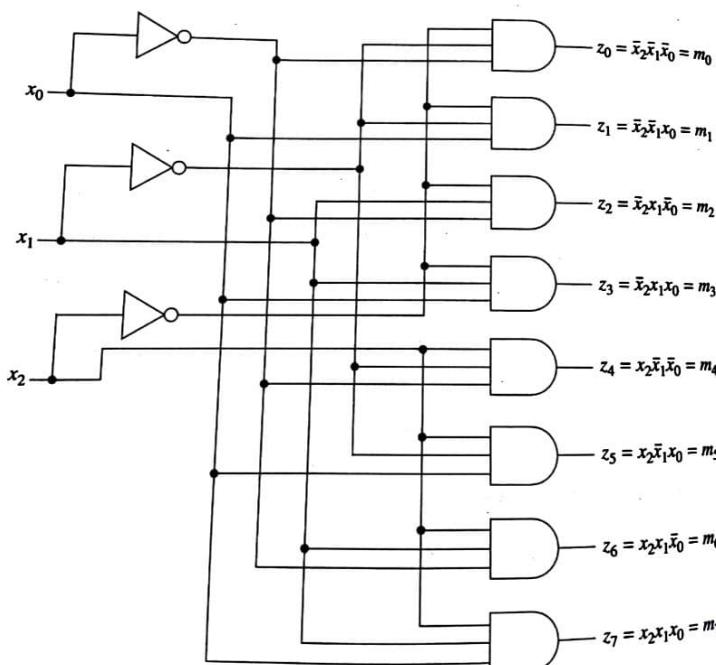
11)

60 of 95



Truth Table of 3:8 decoder

Inputs $x_2 \ x_1 \ x_0$	Outputs							
	z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7
0 0 0	1	0	0	0	0	0	0	0
0 0 1	0	1	0	0	0	0	0	0
0 1 0	0	0	1	0	0	0	0	0
0 1 1	0	0	0	1	0	0	0	0
1 0 0	0	0	0	0	1	0	0	0
1 0 1	0	0	0	0	0	1	0	0
1 1 0	0	0	0	0	0	0	1	0
1 1 1	0	0	0	0	0	0	0	1



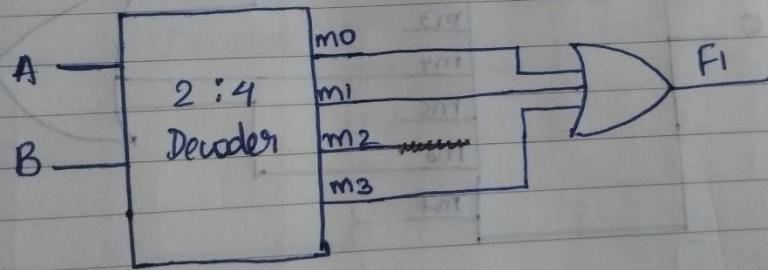
10) i) $F_1 = \sum m(0, 1, 3)$

12

\rightarrow	A	B	F_1
1	0	0	1
	0	1	1
2	1	0	0
	1	1	1

$F_1 = \sum m(0, 1, 3)$

$\bar{F}_1 = \sum m(2)$

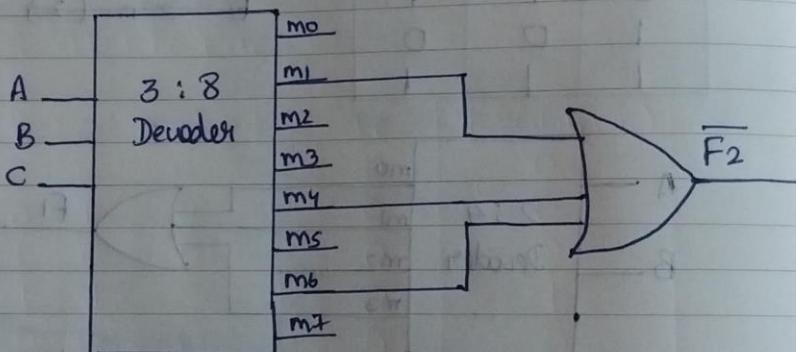


ii) $F_2 = \sum m(0, 2, 3, 5, 7)$

\rightarrow	A	B	C	F_2
	m0	0	0	1
10	m1	0	0	0
	m2	0	1	1
11	m3	0	1	1
	m4	1	0	0
12	m5	1	0	1
	m6	1	1	0
1	m7	1	1	1

$F_2 = \sum m(0, 2, 3, 5, 7)$

$\bar{F}_2 = \sum m(1, 4, 6)$



24 25 26
2023 APRIL
999)

$$F_3 = \sum m(1, 3, 5, 6, 7)$$

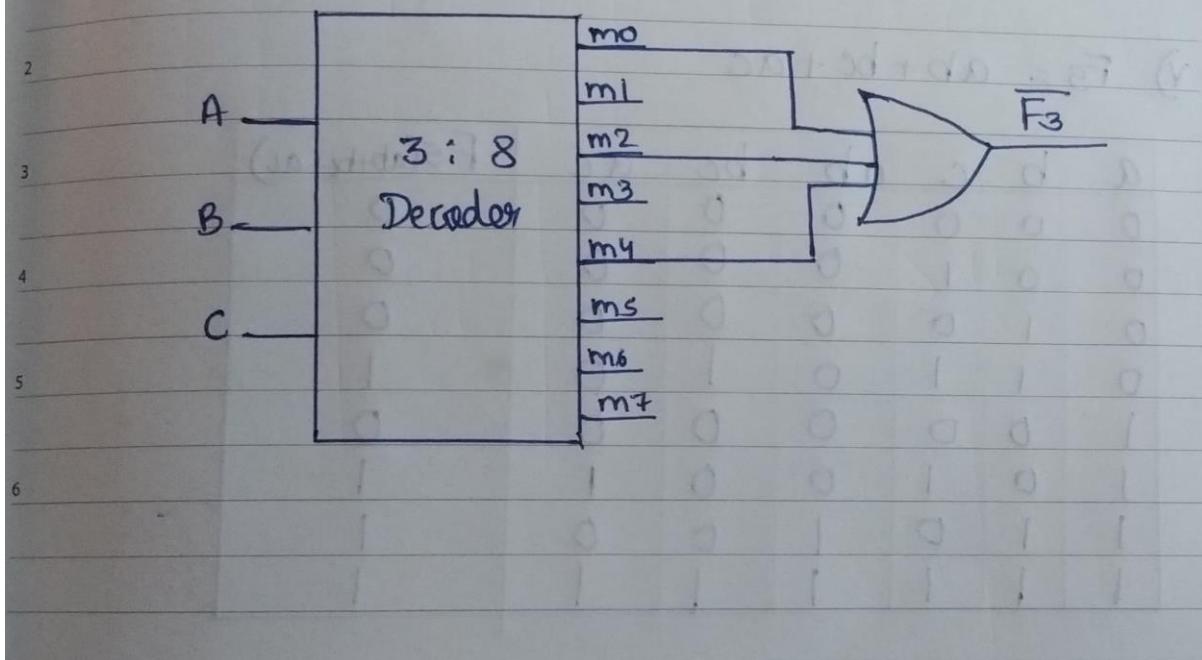
SUNDAY
(071-294) Wk 10

12

→	A	B	C	F_3
9	0	0	0	0
10	0	0	1	1
11	0	1	0	0
12	0	1	1	1
1	1	0	0	0
	1	0	1	1
	1	1	0	1
	1	1	1	1

$$(F_3 = \sum m(1, 3, 5, 6, 7))$$

$$\overline{F_3} = \sum m(0, 2, 4)$$

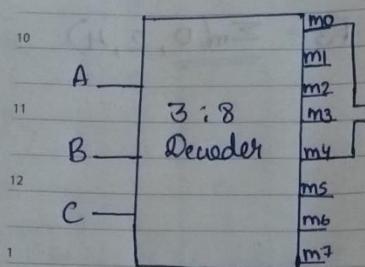


13

MONDAY
Wk 11 (072-293)[maxterm $\bar{F}_4 = 0$]

iv) $F_4 = \bar{\Sigma}M(1, 2, 3, 5, 6, 7)$

$\rightarrow F_4 = \Sigma m(0, 4)$

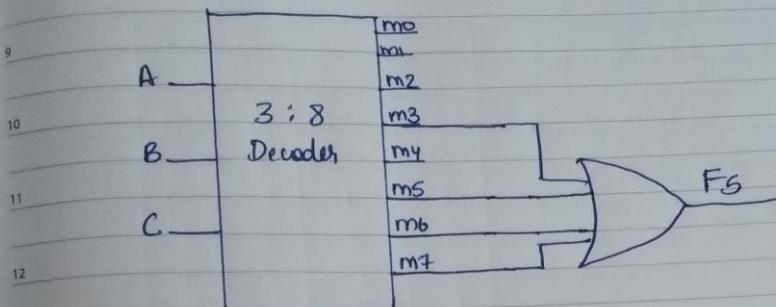


A	B	C	F_4
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

v) $F_5 = ab + bc + ac$

a	b	c	ab	bc	ac	$F_5(ab+bc+ac)$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	1
1	0	0	0	0	0	0
1	0	1	0	0	1	1
1	1	0	1	0	0	1
1	1	1	1	1	1	1

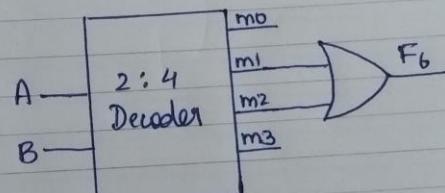
$F_5 = \Sigma m(3, 5, 6, 7)$



vi) $F_6 = \bar{\Sigma}M(0, 3)$

$\rightarrow F_6 = \Sigma m(1, 2)$

A	B	F_6
0	0	0
0	1	1
1	0	0
1	1	0



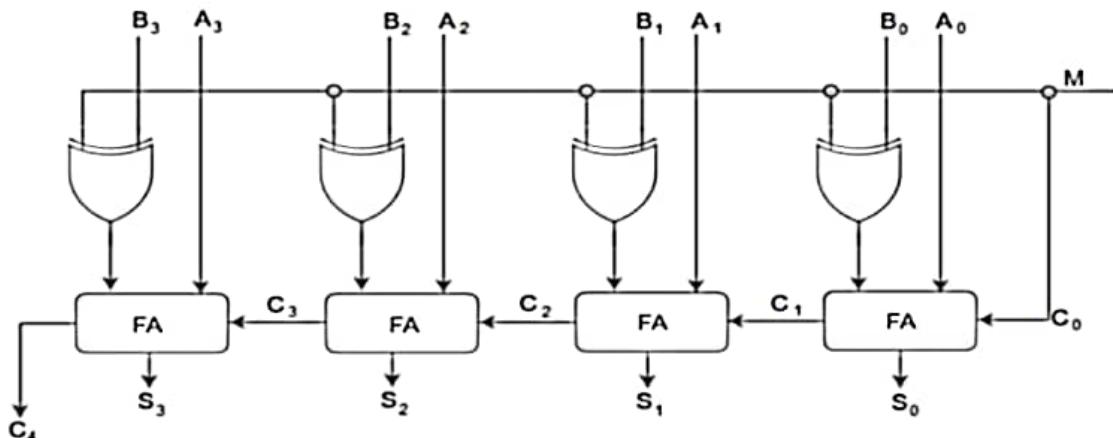
12]. 4-bit Parallel Adder/Subtractor using XOR

- In Digital Circuits, a Binary adder/subtractor is capable of both the addition and subtraction of binary numbers in one circuit itself.
- The operation is performed depending on the binary value the control signal holds. It is one of the components of the ALU (Arithmetic Logic Unit).
- Let's consider two 4-bit binary numbers A and B as inputs to the Digital Circuit for the operation with digits.

A_0	A_1	A_2	A_3	for A
B_0	B_1	B_2	B_3	for B

The circuit consists of 4 full adders since we are performing operations on 4-bit numbers. There is a control line M that holds a binary value of either 0 or 1 which determines that the operation is carried out is addition or subtraction.

4 bit adder-subtractor:



- When the mode input (M) is at a low logic, i.e. '0', the circuit acts as an adder and when the mode input is at a high logic, i.e. '1', the circuit acts as a subtractor.
- The exclusive-OR gate connected in series receives input M and one of the inputs B.
- When M is at a low logic, we have $B \oplus 0 = B$. The full adders receive the value of B, the input carry is 0, and the circuit performs A plus B.
- When M is at a high logic, we have $B \oplus 1 = B'$ and $C_0 = 1$. The B inputs are complemented, and a 1 is added through the input carry. The circuit performs the operation A plus the 2's complement of B.

Examples:

$$\begin{aligned} M=0: A &= 0010, B = 0100 \\ S &= 0110, C_4 = 0 \end{aligned}$$

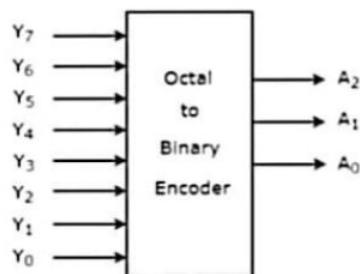
$$\begin{aligned} M=0: A &= 1011, B = 0110 \\ S &= 0001, C_4 = 1 \end{aligned}$$

$$\begin{aligned} M=1: A &= 0010, B = 0100 \\ S &= 1110, C_4 = 0 \end{aligned}$$

$$\begin{aligned} M=1: A &= 1011, B = 0110 \\ S &= 0101, C_4 = 1 \end{aligned}$$

15].

Octal to Binary(8 to 3 line) Encoder

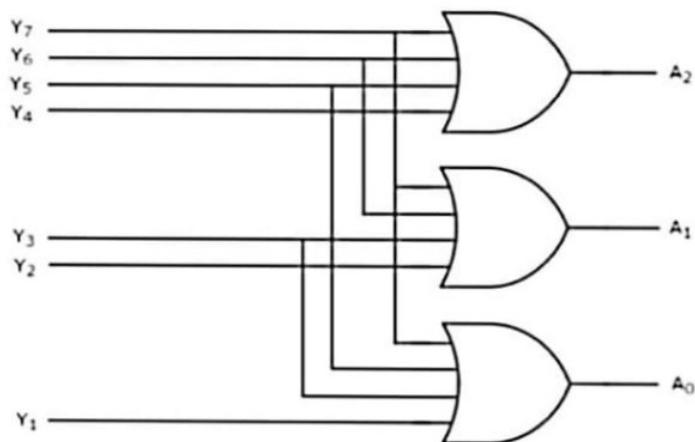


Inputs								Outputs		
Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	1	1	1

$$A_2 = Y_7 + Y_6 + Y_5 + Y_4$$

$$A_1 = Y_7 + Y_6 + Y_3 + Y_2$$

$$A_0 = Y_7 + Y_5 + Y_3 + Y_1$$



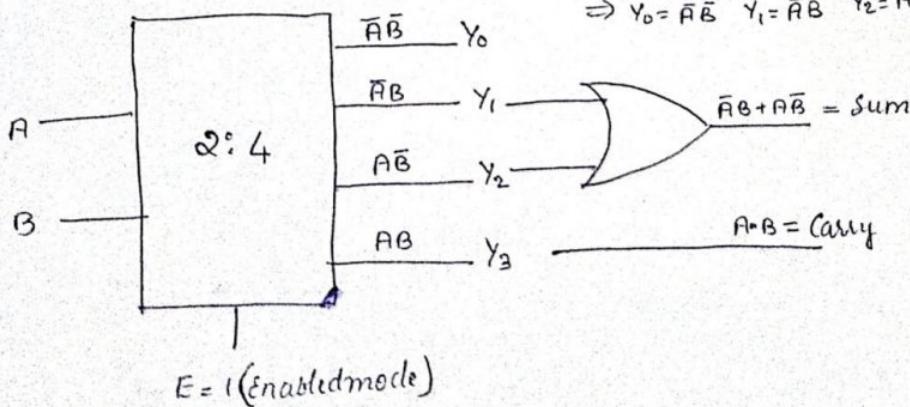
16]. Half adder using 2 to 4 line decoder

$$\Rightarrow \text{Sum} = \bar{A}B + A\bar{B} = A \oplus B = \Sigma m(1, 2)$$

$$\text{Carry} = A \cdot B = \Sigma m(3)$$

A	B	y_0	y_1	y_2	y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$$\Rightarrow y_0 = \bar{A}\bar{B} \quad y_1 = \bar{A}B \quad y_2 = A\bar{B} \quad y_3 = AB$$



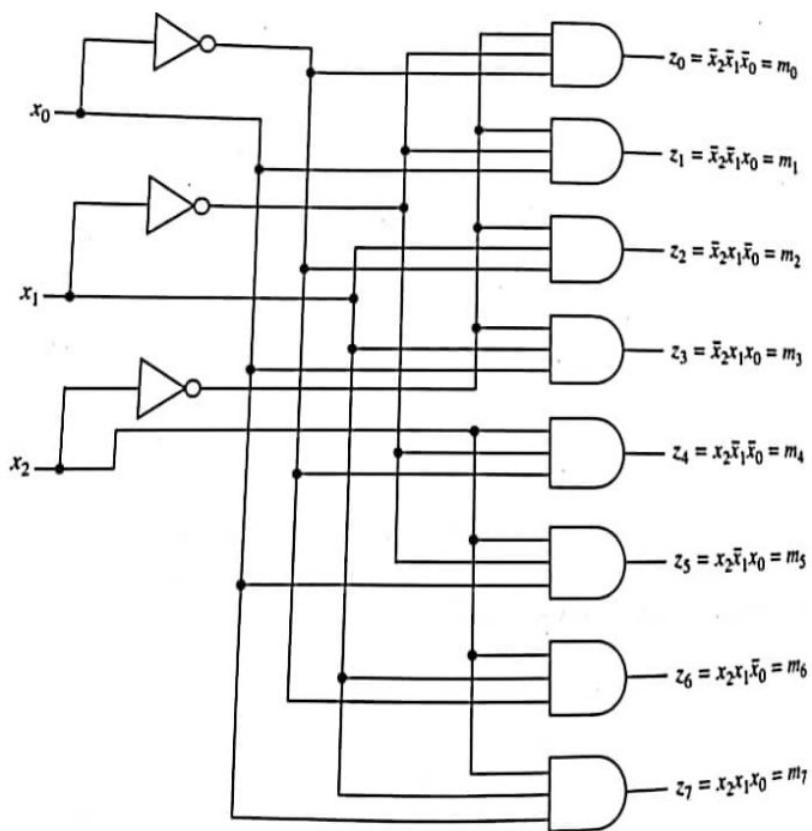
17]. 3 to 8 decoder

3-to-8 DEC	
x_0	0
x_1	1
x_2	2
	3
	4
	5
	6
	7

$z_0 = \bar{x}_2 \bar{x}_1 \bar{x}_0 = m_0$
$z_1 = \bar{x}_2 \bar{x}_1 x_0 = m_1$
$z_2 = \bar{x}_2 x_1 \bar{x}_0 = m_2$
$z_3 = \bar{x}_2 x_1 x_0 = m_3$
$z_4 = x_2 \bar{x}_1 \bar{x}_0 = m_4$
$z_5 = x_2 \bar{x}_1 x_0 = m_5$
$z_6 = x_2 x_1 \bar{x}_0 = m_6$
$z_7 = x_2 x_1 x_0 = m_7$

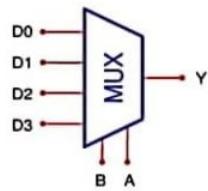
Truth Table of 3:8 decoder

Inputs	Outputs							
$x_2 \ x_1 \ x_0$	z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7
0 0 0	1	0	0	0	0	0	0	0
0 0 1	0	1	0	0	0	0	0	0
0 1 0	0	0	1	0	0	0	0	0
0 1 1	0	0	0	1	0	0	0	0
1 0 0	0	0	0	0	1	0	0	0
1 0 1	0	0	0	0	0	1	0	0
1 1 0	0	0	0	0	0	0	1	0
1 1 1	0	0	0	0	0	0	0	1

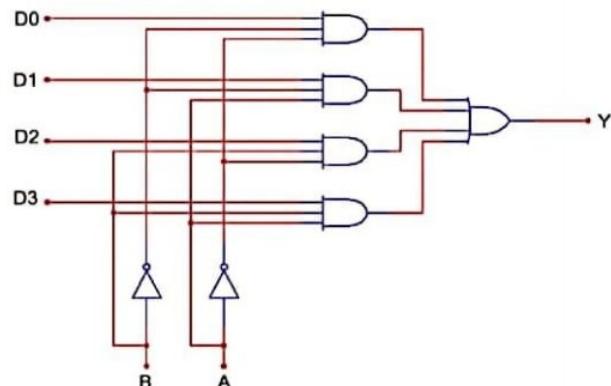


18].

4:1 Multiplexer



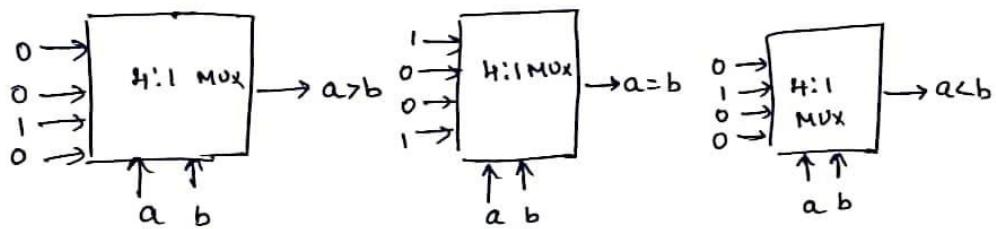
B	A	Y
0	0	D0
0	1	D1
1	0	D2
1	1	D3



$$Y = \bar{B}\bar{A}D_0 + \bar{B}AD_1 + B\bar{A}D_2 + BAD_3$$

18) Implement 1 bit comparator using 4:1 Mux.

a	b	$a > b$	$a = b$	$a < b$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0



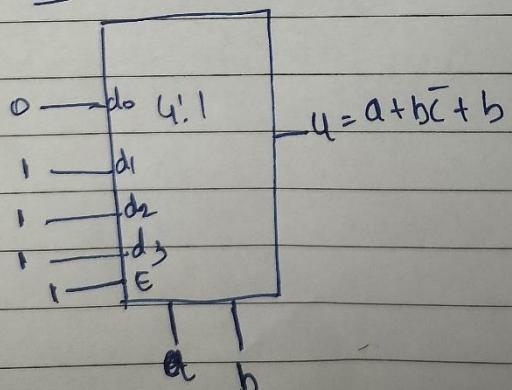
19) Implement 8:1 and 4:1

a) $u = a + b\bar{c} + b$

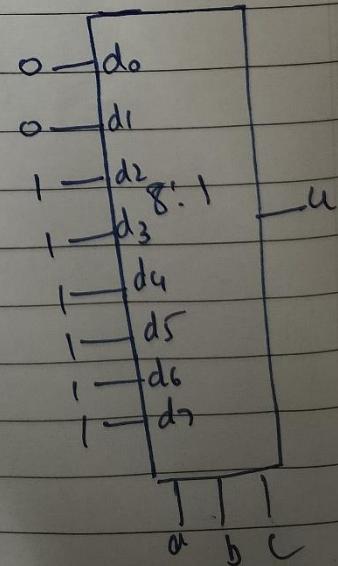
b) $F(A, B, C) = \sum m(0, 1, 3, 4)$

\check{a}	\check{b}	c	\bar{c}	$b\bar{c}$	u
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	1	1
1	1	1	0	0	1

4:1



8:1



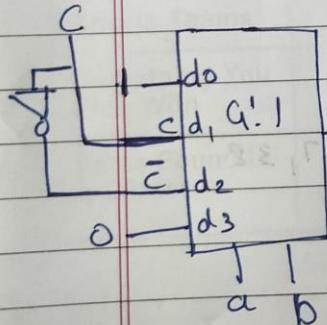
b) $a \ b \ c \quad f$

0 0 0	1	J_1	ut्तमेन्द्रियाः प्रभवतः
0 0 1	1	J_2	उत्तमेन्द्रियाः भवति उत्तमं
0 1 0	0	J_3	अनुप्राप्ताः उत्तमाः प्रभवतः उत्तमं
0 1 1	1	J_4	प्राप्ताः उत्तमाः प्रभवतः उत्तमं
1 0 0	1	J_5	प्राप्ताः प्रभवतः उत्तमं
1 0 1	0		उत्तमः प्रभवतः उत्तमं
1 1 0	0	J_6	
, , , 1 0			

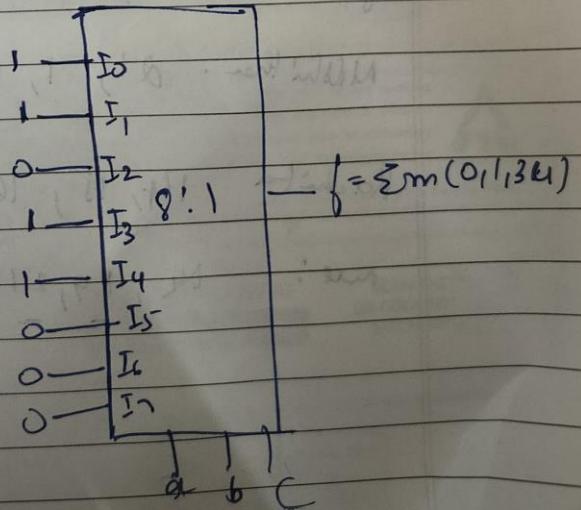
4: 1

अवृत्ति मृत्ति वर्णन
अवृत्ति मृत्ति वर्णन

संकेतिक वर्णन



$$f = \Sigma m(0, 1, 3, 4)$$



$$f = \Sigma m(0, 1, 3, 4)$$

20.

$$F_1 = \Sigma m(0, 1; 3)$$

$$F_2 = \Sigma m(0, 2, 3, 5, 7)$$

$$\bar{F}_2 = \Sigma m(1, 4, 6)$$

$$F_3 = \Sigma m(1, 3, 5, 6, 7)$$

$$\bar{F}_3 = \Sigma m(0, 2, 4)$$

$$F_4 = \Pi M(1, 2, 3, 5, 6, 7)$$

$$F_4 = \Sigma m(0, 4)$$

$$F_5 = ab + bc + ac$$

$$= ab(c+\bar{c}) + bc(a+\bar{a}) + ac(b+\bar{b})$$

$$= abc + ab\bar{c} + \underline{abc} + \bar{a}bc + \underline{abc} + a\bar{b}c$$

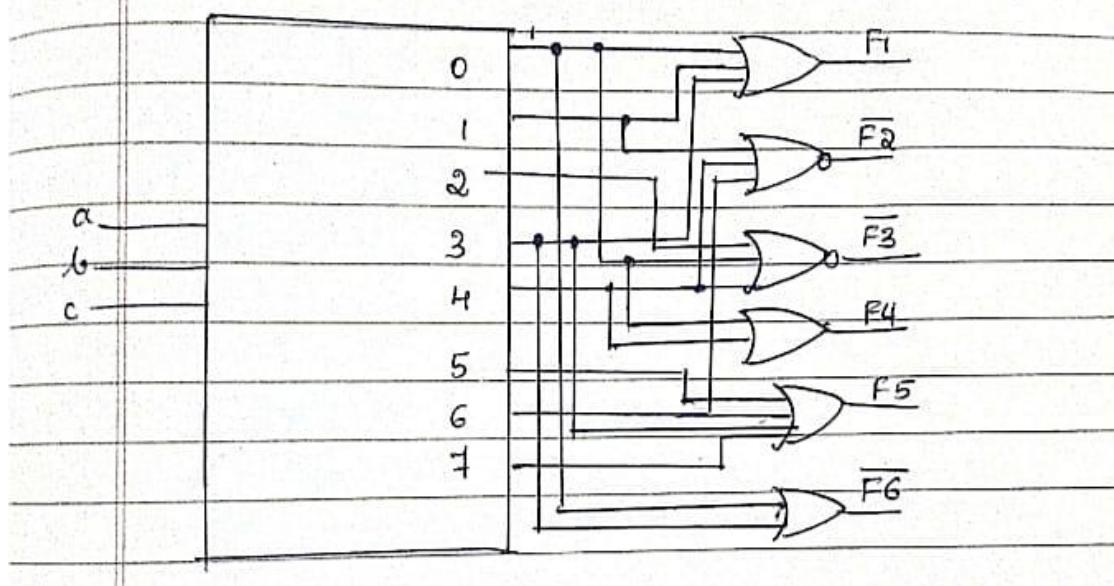
$$= abc + ab\bar{c} + \bar{a}bc + a\bar{b}c$$

$$F_5 = \underline{\Sigma m(3, 5, 6, 7)}$$

$$F_6 = \Pi M(0, 3)$$

$$F_6 = \Sigma m(1, 2, 4, 5, 6, 7)$$

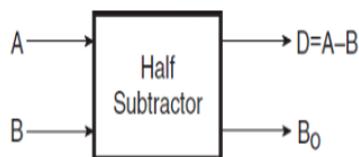
$$\bar{F}_6 = \Sigma m(0, 3)$$



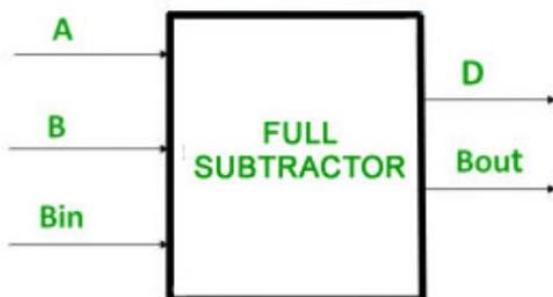
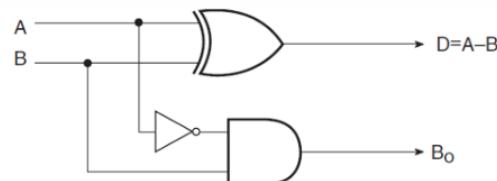
Half Subtractor

It produces the difference between the two binary bits at the input and produces two outputs difference and borrow (to indicate if a 1 has been borrowed)

In the subtraction (A), A is called a **Minuend bit** and B is called a **Subtrahend bit**.



A	B	D	B_0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



INPUT			OUTPUT	
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

D

		00	01	11	10
		0	1	0	1
A	B	0	1	0	1
0	0	0	1	0	1
1	0	1	0	1	0

Bout

		00	01	11	10
		0	1	1	1
A	B	0	0	1	0
0	0	0	1	1	1
1	0	0	0	1	0

$$\begin{aligned}
 D &= \bar{A}\bar{B}B_{in} + \bar{A}B\bar{B}_{in} + A\bar{B}\bar{B}_{in} + AB\bar{B}_{in} \\
 &= B_{in}(\bar{A}\bar{B} + AB) + \bar{B}_{in}(\bar{A}B + A\bar{B}) \\
 &= B_{in}(A \oplus B) + \bar{B}_{in}(A \oplus B) \\
 &= (A \oplus B) \oplus B_{in}
 \end{aligned}$$

$$\text{Bout} = \bar{A}B_{in} + \bar{A}B + BB_{in}$$

1-bit comparator

Truth table for a one-bit comparator

A	B	A EQ B	A > B	A < B
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

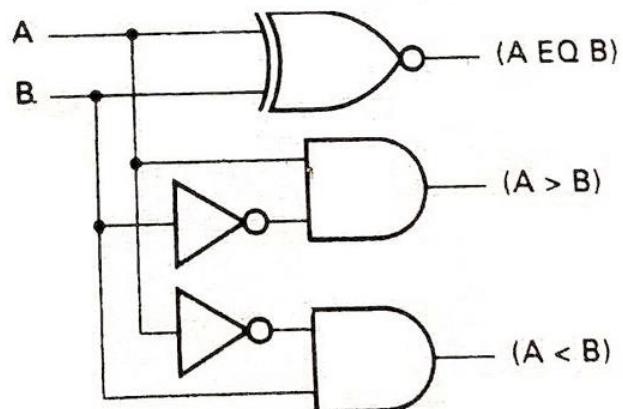
The equations for each output are

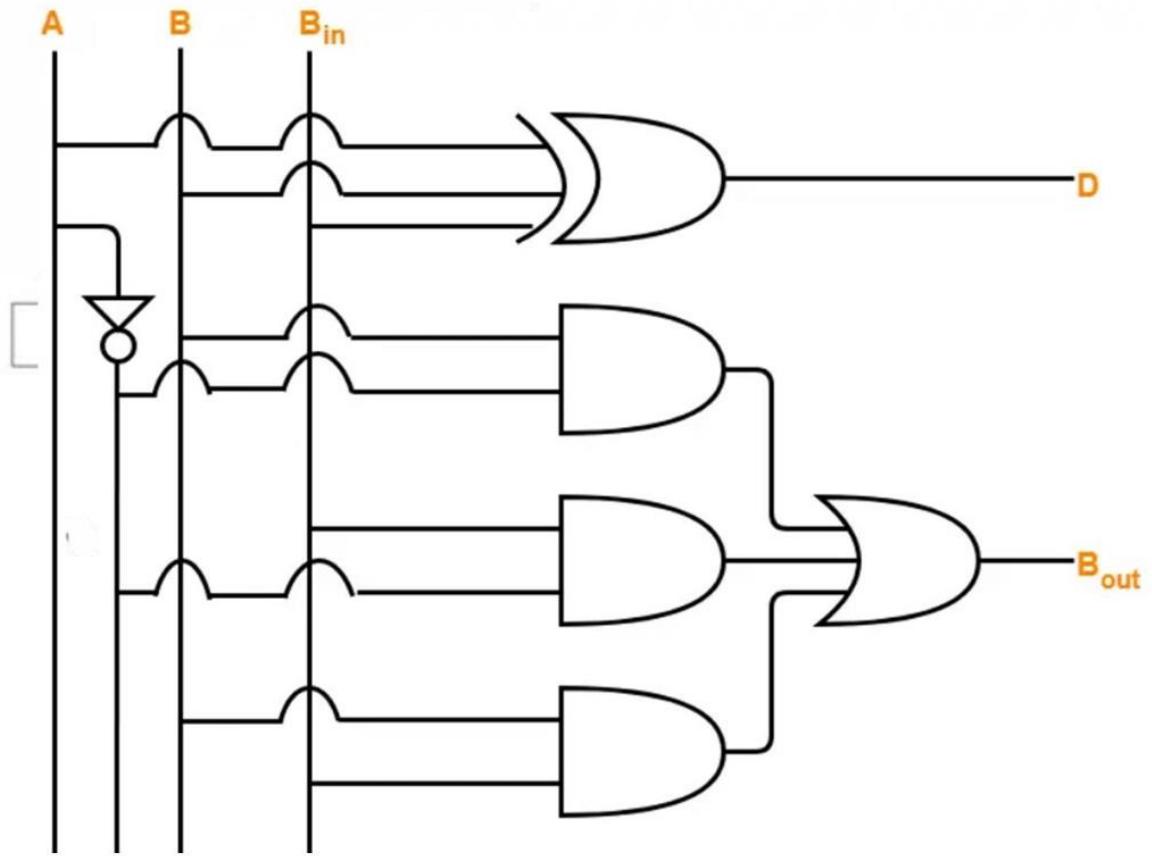
$$A \text{ EQ } B = A'B' + AB = (A \oplus B)'$$

$$A > B = AB'$$

$$A < B = A'B$$

Logic for a single-bit
comparator





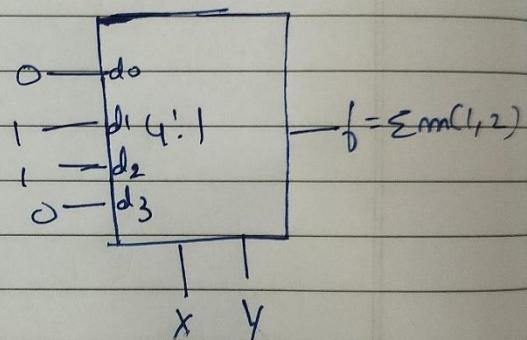
23

Implement using 4:1 mux

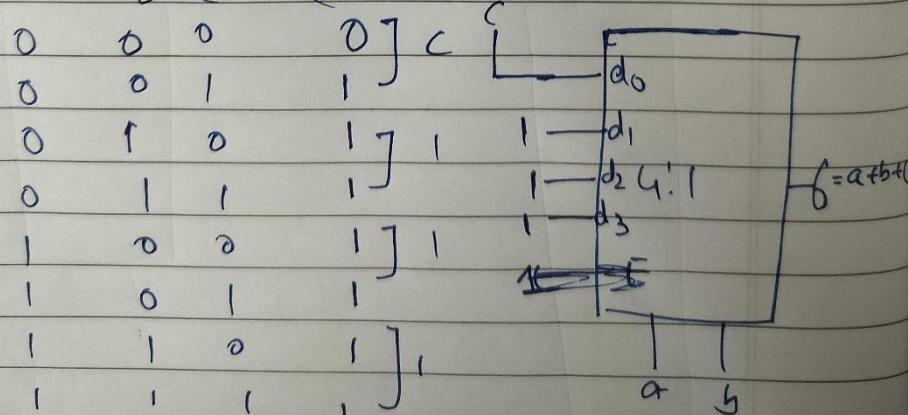
$$(i) f(x, y) = \Sigma m(1, 2)$$

$$(ii) f(a, b, c) = a + b + c$$

	x	y	b
0	0	0	0
0	0	1	1
1	0	0	1
1	1	0	0



$$(iii) a \quad b \quad c \quad (a+b+c)$$

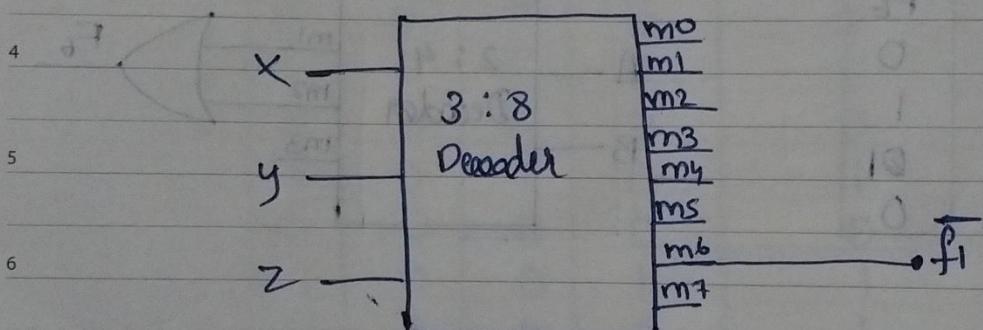


$$24) \text{ i) } f_1(x, y, z) = \overline{xy} + z$$

	x	y	z	xy	\overline{xy}	$\overline{xy} + z$
9	0	0	0	0	1	1
10	0	0	1	0	1	1
11	0	1	0	0	1	1
12	0	1	1	0	1	1
13	1	0	0	0	1	1
14	1	0	1	0	1	1
15	1	1	0	1	0	0
16	1	1	1	1	0	1

$$2 \Rightarrow f_1(x, y, z) = \sum m(0, 1, 2, 3, 4, 5, 7)$$

$$3 \Rightarrow f_1(\overline{x}, \overline{y}, z) = \sum m(6)$$

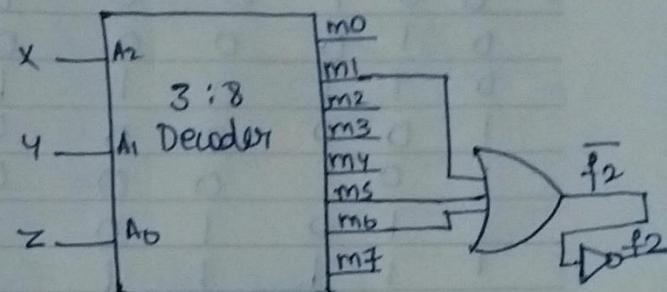


$$\text{ii) } f_2(x, y, z) = \sum m(0, 2, 3, 4, 7)$$

	x	y	z	f_2
9	0	0	0	1
10	0	0	1	0
11	0	1	0	1
12	0	1	1	1
13	1	0	0	1
14	1	0	1	0
15	1	1	0	0
16	1	1	1	1

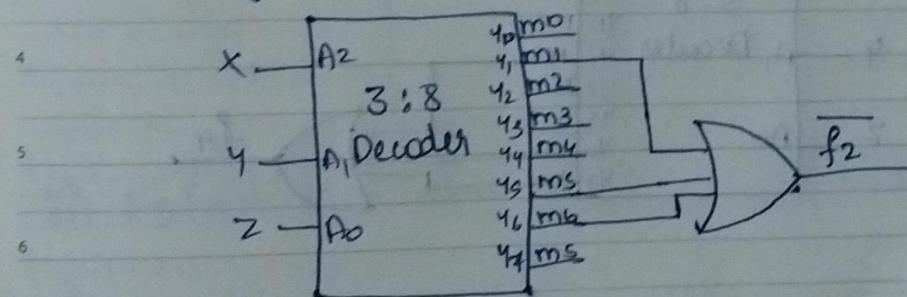
$$f_2 = \sum m(0, 2, 3, 4, 7)$$

$$\bar{f}_2 = \sum (1, 5, 6)$$



$$\text{iii) } f_3(x, y, z) = \sum m(1, 5, 6)$$

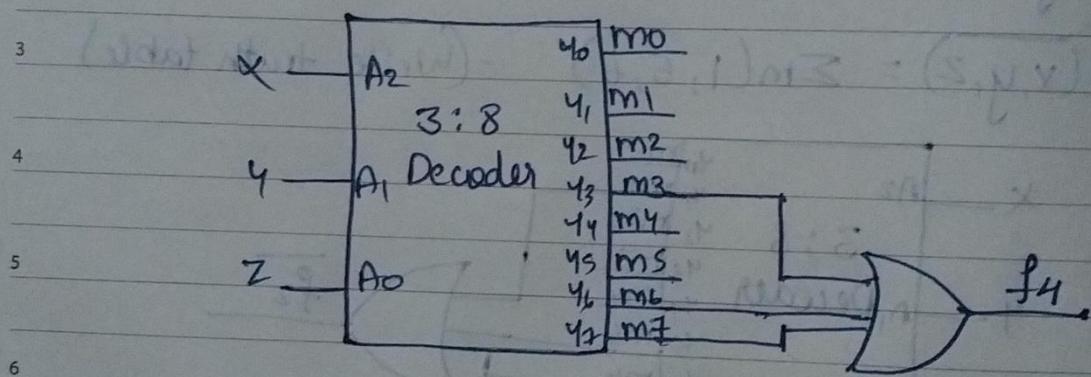
→ $f_3(\bar{x}, \bar{y}, z) = \sum m(1, 5, 6)$ - (Write truth table)



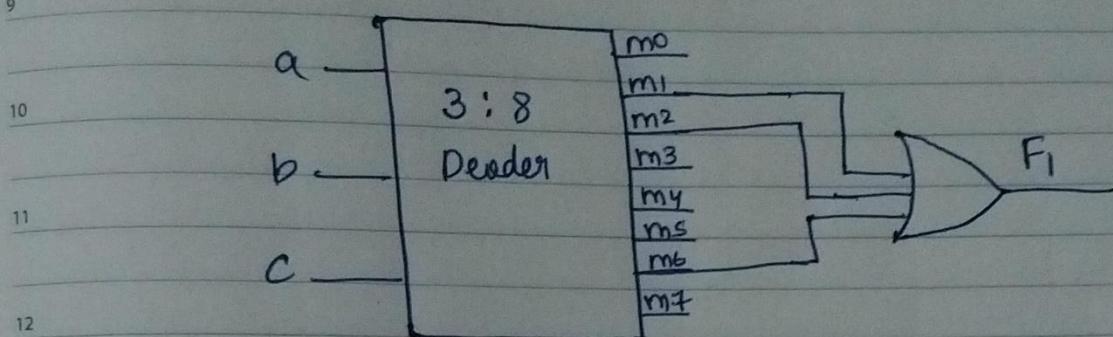
iv) $f_4(x, y, z) = xy + yz$

	x	y	z	xy	yz	$xy + yz$
9	0	0	0	0	0	0
10	0	0	1	0	0	0
11	0	1	0	0	0	0
12	0	1	1	0	1	1
13	1	0	0	0	0	0
14	1	0	1	0	0	0
15	1	1	0	1	0	1
16	1	1	1	1	1	1

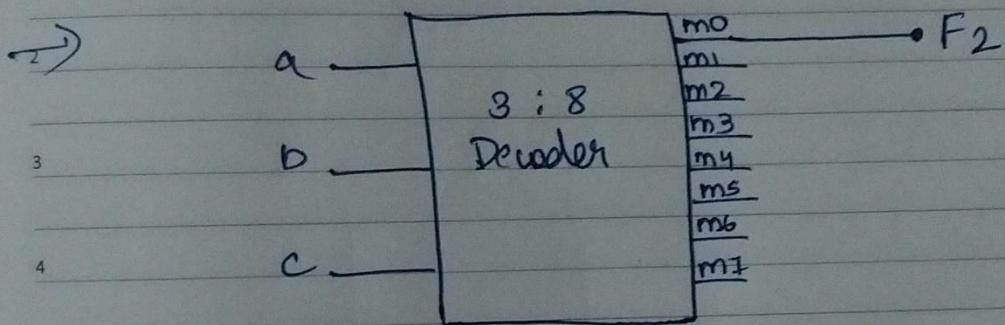
2 $f_4(x, y, z) = \sum m(3, 6, 7)$



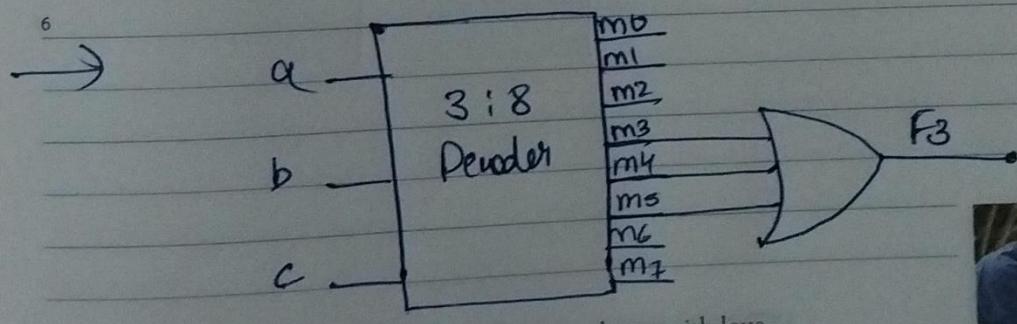
28) i) $F_1(a, b, c) = \sum m(1, 2, 6)$
 \rightarrow Write truth table



ii) $F_2(a, b, c) = \sum m(0)$ (write table)



iii) $F_3(a, b, c) = \sum m(3, 4, 5)$ (write truth table)



» *Forwarded*

A 4-to-2 Encoder is a combinational circuit that converts 4 input lines into 2 output lines. It is a simplified version of a binary encoder where each input corresponds to a specific binary-coded output.

Working of 4x2 Encoder:

- It has 4 input lines (Y_0, Y_1, Y_2, Y_3) and 2 output lines (A_1, A_0).
- The encoder outputs the binary representation of the highest-priority active (1) input.

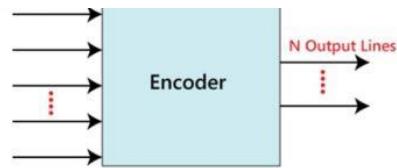
6:43 pm

» *Forwarded*

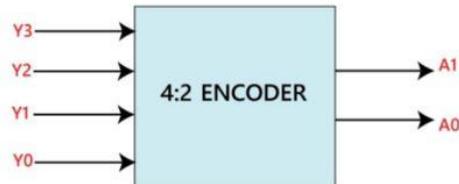
Logic Circuit:

- The circuit consists of two OR gates:
 - The first OR gate takes inputs from Y_2 and Y_3 to generate A_1 .
 - The second OR gate takes inputs from Y_1 and Y_3 to generate A_0 .

6:43 pm



4 to 2 Line Encoder



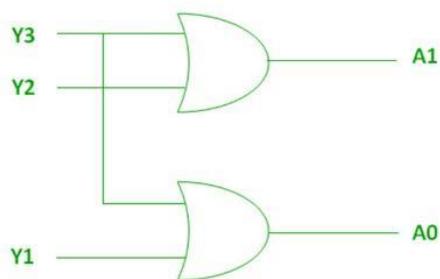
Truth Table:

INPUTS				OUTPUTS	
Y3	Y2	Y1	Y0	A1	A0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

4 to 2 Line Encoder

$$\begin{aligned}A_1 &= Y_3 + Y_2 \\A_0 &= Y_3 + Y_1\end{aligned}$$

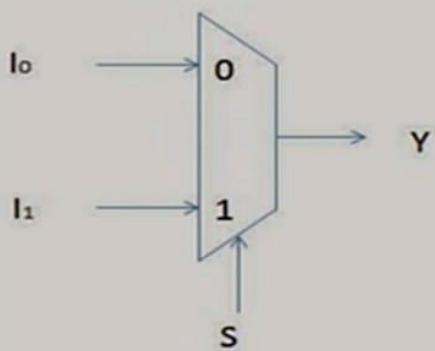
Logic Diagram



8x3 Encoder

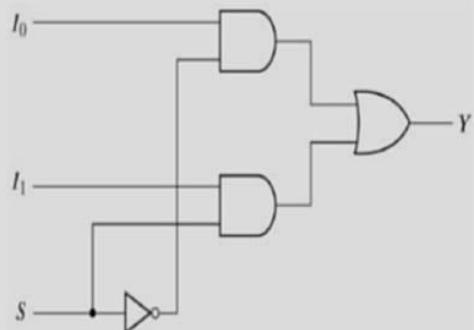
32.

2-to-1 Mux



S	Y
0	I_0
1	I_1

$$Y = \bar{S}I_0 + S I_1$$

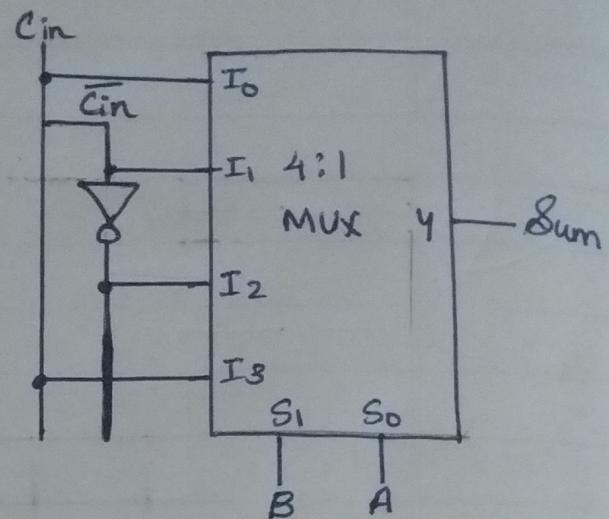


33) Full Adder using 4×1 MUX.

For Sum :

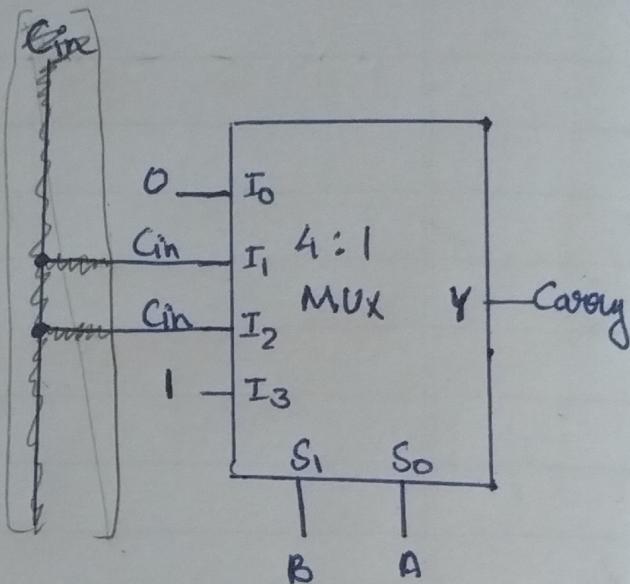
33

	B	A	Cin	Sum	
I_0	0	0	0	0	$\{ \}$ Cin
	0	0	1	1	
I_1	0	1	0	1	$\{ \}$ \bar{Cin}
	0	1	1	0	
I_2	1	0	0	1	$\{ \}$ \bar{Cin}
	1	0	1	0	
I_3	1	1	0	0	$\{ \}$ Cin
	1	1	1	1	



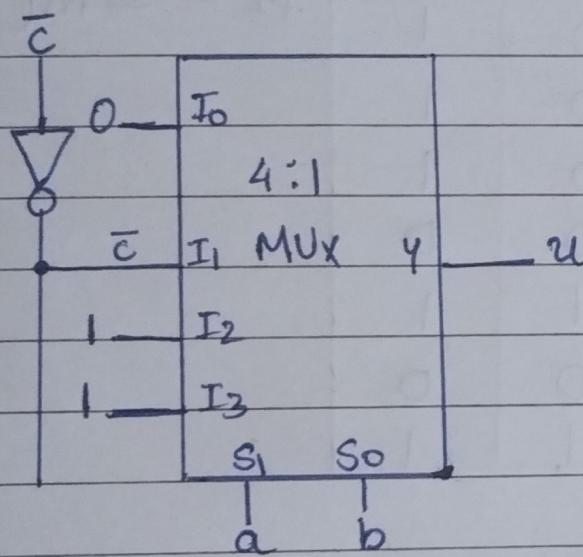
For Carry

	B	A	Cin	Carry	
I_0	0	0	0	0	$\{ \}$ 0
	0	0	1	0	
I_1	0	1	0	0	$\{ \}$ Cin
	0	1	1	1	
I_2	1	0	0	0	$\{ \}$ Cin
	1	0	1	1	
I_3	1	1	0	1	$\{ \}$ 1
	1	1	1	1	



34) Implement $u = a + bc'$ using 4:1 MUX

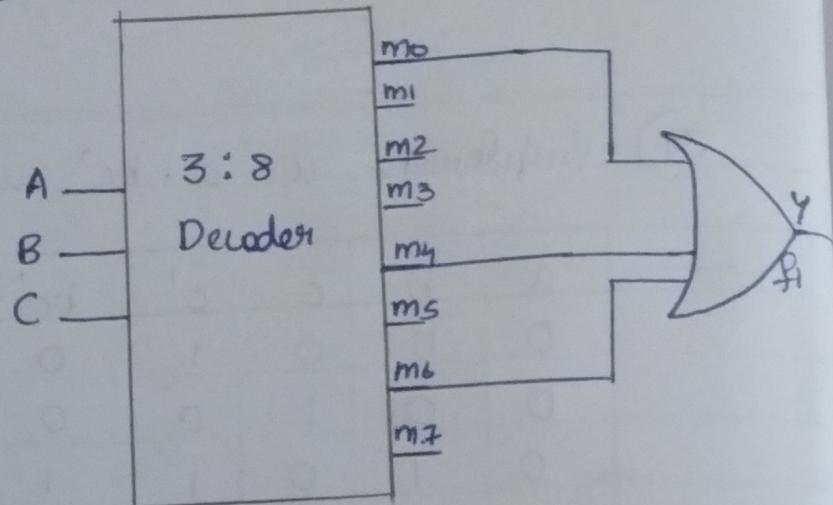
$\overline{S_1}$	S_0	a	b	c	\overline{c}'	bc'	$u = a + bc'$
0	0	0	0	1	1	0	0
0	0	0	1	0	0	0	0
0	1	0	0	1	1	1	1
0	1	1	0	0	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	1
1	1	0	0	1	1	1	1
1	1	1	0	0	1	1	1



$$85) f_1(A, B, C) = \sum m(0, 4, 6) \quad f_2 = A' + AC \quad 3:8 \text{ Decoder}$$

35)

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

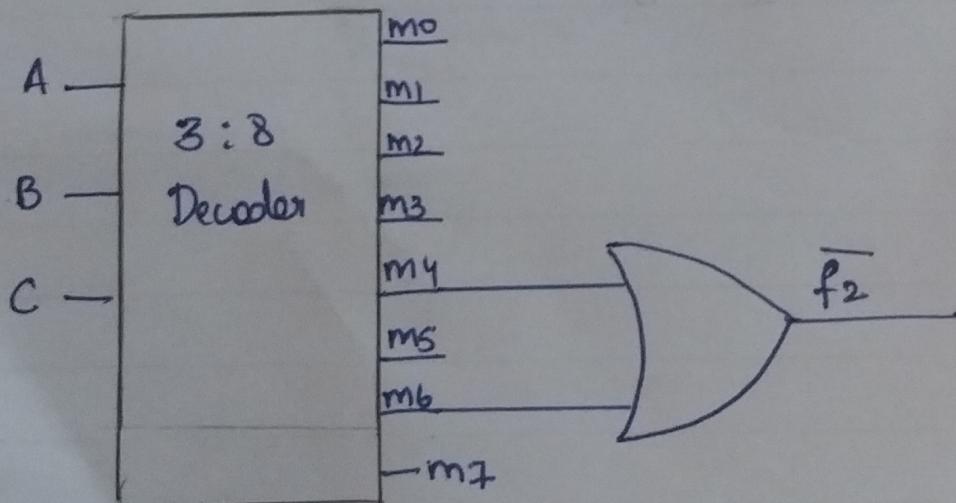


$$\textcircled{ii} \quad f_2 = A' + AC$$

A	B	C	A'	AC	$A' + AC$
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	0	0	0
1	1	1	0	1	1

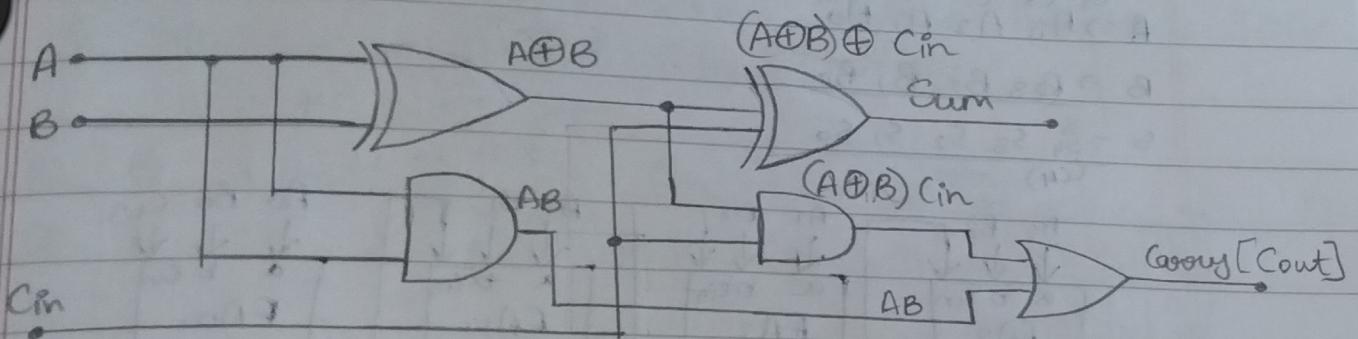
$$f_2 = \sum m(0, 1, 2, 3, 5, 7)$$

$$\bar{f}_2 = \sum m(4, 6)$$



37)

Full Adder using Half Adder



$$\begin{aligned}
 \text{Cout} &= (A \oplus B) \cdot \text{Cin} + AB \\
 &= (\bar{A}B + \bar{B}A) \cdot \text{Cin} + AB \\
 &= \bar{A}\bar{B}\text{Cin} + \bar{A}B\text{Cin} + AB \\
 &= A(\bar{B}\text{Cin} + B) + \bar{A}B\text{Cin} \\
 &= A(\bar{B} + \text{Cin}) + \bar{A}B\text{Cin} \\
 &= \bar{A}\bar{B} + A\text{Cin} + \bar{A}B\text{Cin} \\
 &= B(A + \bar{A}\text{Cin}) + A\text{Cin} \\
 &= B(A + \bar{A})(A + \text{Cin}) + A\text{Cin} \\
 &= B(A + \text{Cin}) + A\text{Cin}
 \end{aligned}$$

$$\text{Cout} = \underline{\overline{AB + BCin + ACin}}$$

$$\text{Sum} = \underline{\overline{(A \oplus B) \oplus Cin}}$$

38)

$$F(A, B, C, D) = \sum m(6, 7, 9, 10, 13) \quad 8:1 \text{ MUX}$$

	A	B	C	D	F	0	0	\bar{D}	I ₀
m ₀	0	0	0	0	0	0	0	0	I ₁
m ₁	0	0	0	1	0	0	0	0	I ₂
m ₂	0	0	1	0	0	0	0	1	I ₃
m ₃	0	0	1	1	0	0	0	0	I ₄
m ₄	0	1	0	0	0	0	0	0	I ₅ 8:1 MUX
m ₅	0	1	0	1	0	0	0	0	I ₆
m ₆	0	1	1	0	1	0	1	0	I ₇
m ₇	0	1	1	1	1	0	0	0	
m ₈	1	0	0	0	0	0	0	0	S ₂
m ₉	1	0	0	1	1	0	0	0	S ₁
m ₁₀	1	0	1	0	1	0	0	0	S ₀
m ₁₁	1	0	1	1	0	0	0	0	A
m ₁₂	1	1	0	0	0	0	0	0	B
m ₁₃	1	1	0	1	1	0	0	0	C
m ₁₄	1	1	1	0	0	0	0	0	
m ₁₅	1	1	1	1	0	0	0	0	

EXTRA QUESTIONS FROM PPT

Question Bank

- 1. Design half-adder and full-adder circuits using logic gates.

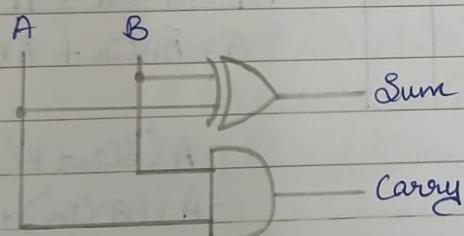
1. Half Adder

Defⁿ: HA takes 2 bit i/p and gives SUM & CARRY.
HA is a combinational logic circuit which acts which adds two i/p bits and generates sum and carry.

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

$S = \bar{A}B + A\bar{B}$
 $\underline{S = A \oplus B}$

$C = AB$



→ Full Adder

Defⁿ: FA is a combinational logic circuit which adds three i/p bits and generates sum and carry.

Inputs			Outputs	
A	B	C-in	C-out	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

A \ BCin	00	01	11	10
0	0	0	1	0
1	1	0	1	1

$$C_{out} = \underline{AB + BC_{int} + AC_{in}}$$

A \ BCin	00	01	11	10
0	0	1	0	1
1	1	0	1	0

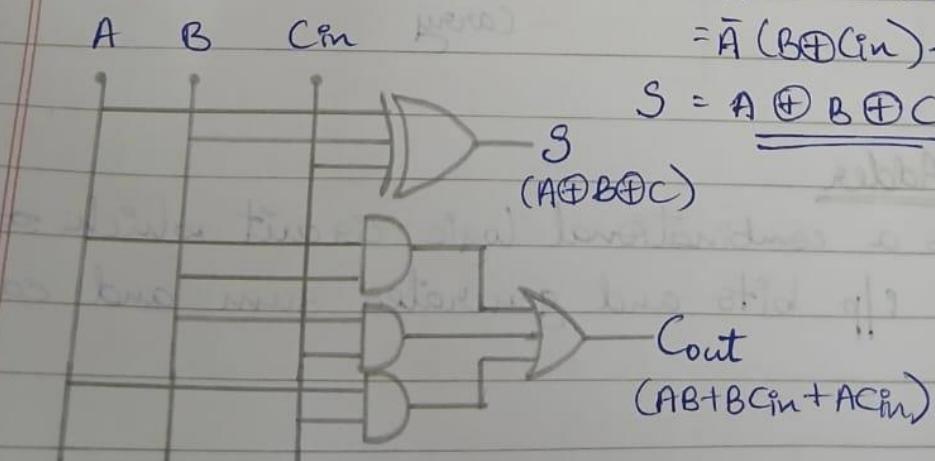
$$S = \bar{A}\bar{B}\bar{C}_{in} + \bar{A}\bar{B}C_{int} + A\bar{B}\bar{C}_{in}$$

$$+ ABC_{in}$$

$$= \bar{A}(\bar{B}C_{int} + B\bar{C}_{in}) + A(\bar{B}C_{int} + B)$$

$$= \bar{A}(B \oplus C_{in}) + A(\bar{B} \oplus C_{in})$$

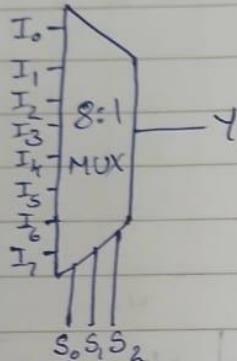
$$S = \underline{A \oplus B \oplus C_{in}}$$



- 15. Explain the working of a multiplexer having 3 select lines.
 - 16. Implement a full subtractor using 8:1 multiplexer.
-

- 8:1 MUX

15.



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

$$Y = \bar{S}_2 \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_2 \bar{S}_1 S_0 I_1 + \bar{S}_2 S_1 \bar{S}_0 I_2 + \bar{S}_2 S_1 S_0 I_3 + S_2 \bar{S}_1 \bar{S}_0 I_4 + \\ S_2 \bar{S}_1 S_0 I_5 + S_2 S_1 \bar{S}_0 I_6 + S_2 S_1 S_0 I_7$$

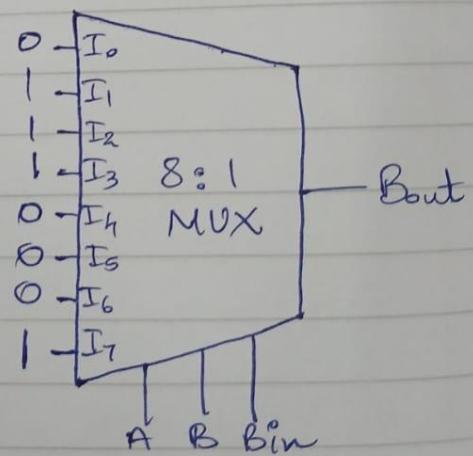
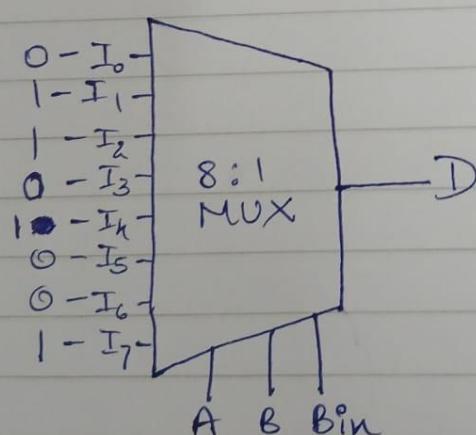
8:1

An ^{8:1} multiplexer is a combinational logic circuit which has 2^3 inputs, 3 select lines and an output. This has $2^3 = 8$ different combinations namely [000, 001, 010, 011, 100, 101, 110, 111].

There is only 1 output. The three select lines determine which of the 8 data inputs is connected to the output.

16. Full Subtractor

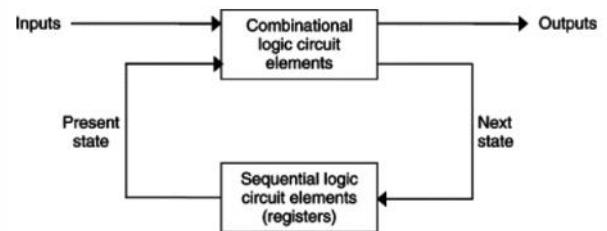
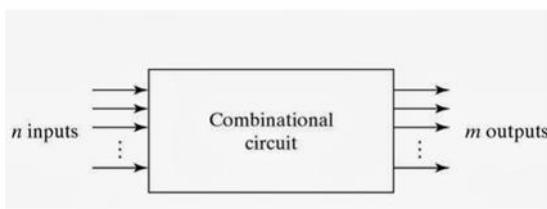
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



Combinational v/s Sequential

1

Output only depends on the present input	Output depends on present input and past output
Memory element is absent	Memory element is present
No clock signal is applied	Clock signal is required



Difference between Latches and Flip Flops

sno	Flip Flops	Latch
1]	It is a Edge triggered device.	It is a Level triggered device.
2]	It requires clock to function.	It doesn't require clock to function.
3]	It is made up of Latches and logic gates.	It is made up of only logic gate blocks.
4]	It checks input continuously but changes the output only when triggered by clock.	It checks input continuously and changes the output immediately w.r.t. input.
5]	Circuit Diagram: 	Circuit Diagram:

7

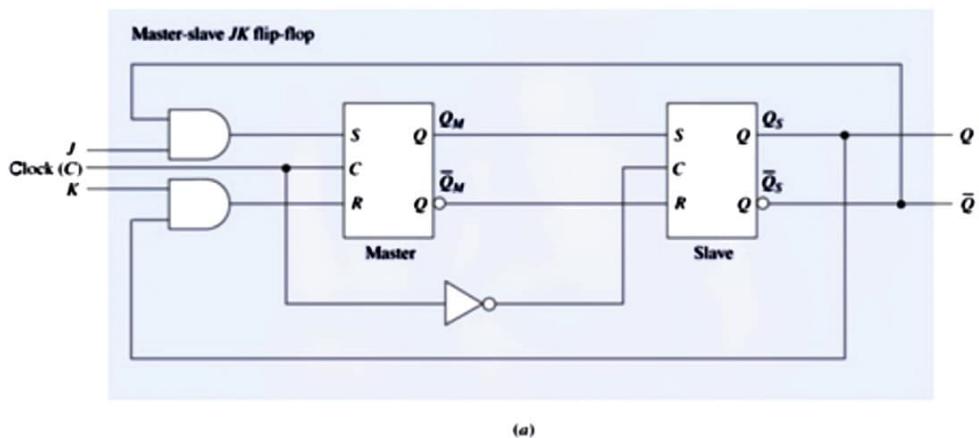
Master-Slave JK Flip-Flop

- In this J corresponds to S and K corresponds to R inputs.
- J=K=1 causes the flip-flop to toggle from the current state.
 - If present state is 1 then next state is 0
 - If present state is 0 then next state is 1
- In this 2 AND gates are used to sense and steer the state of the slave [in addition to SR MS flip-flop]

19EC302: Digital System Design

142

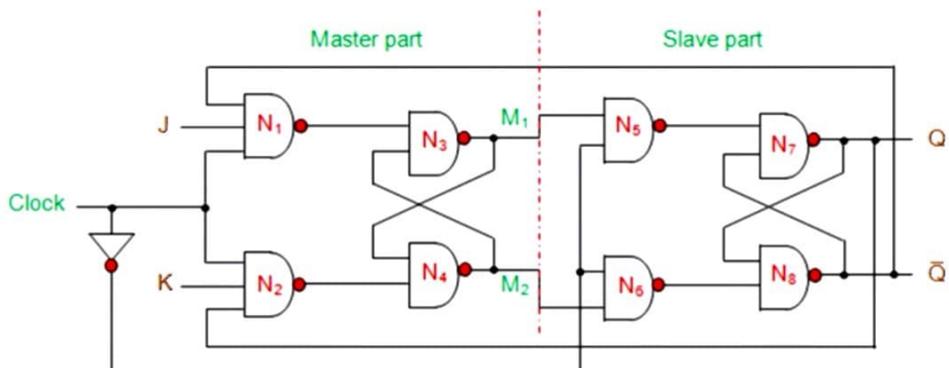
Master-Slave JK Flip-Flop

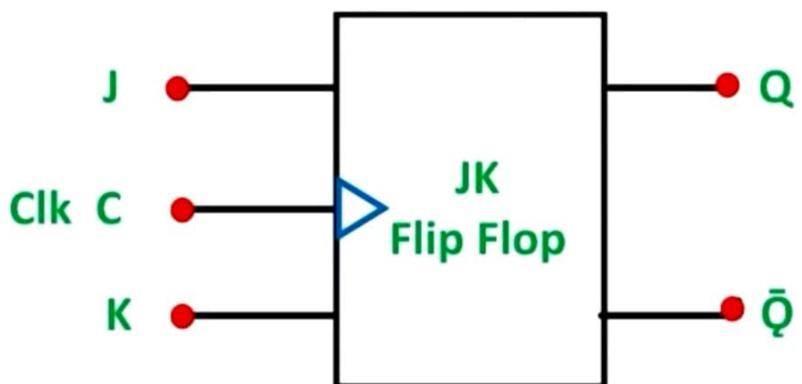


19EC302: Digital System Design

143

Master Slave Flip Flop using NAND gates

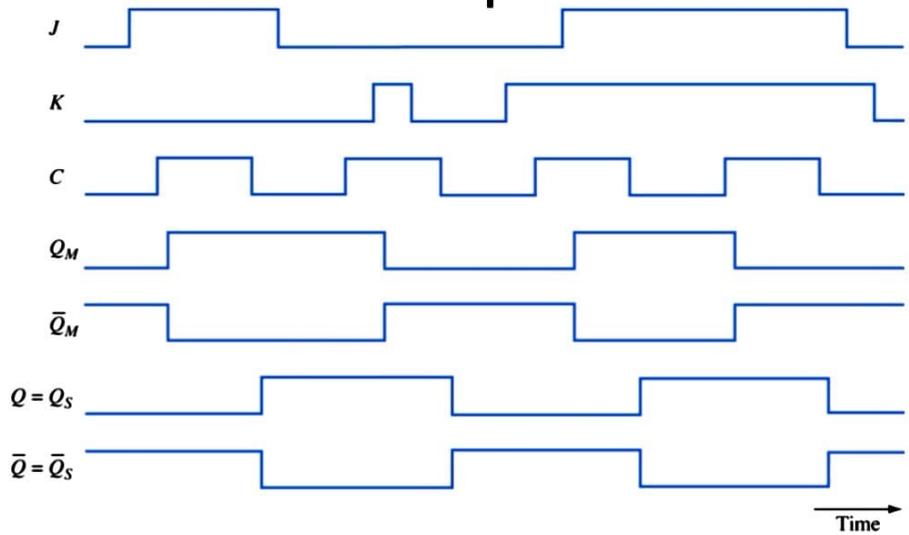




Circuit Globe

J	K	CLK	<i>Q</i>	\bar{Q}	Comment
0	0	↑	<i>Q</i>	\bar{Q}	Latch
1	0	↑	1	0	SET
0	1	↑	0	1	RESET
1	1	↑	\bar{Q}	<i>Q</i>	TOGGLE
<i>X</i>	<i>X</i>	Any Thing Else	<i>Q</i>	\bar{Q}	NO Change !

Timing Diagram for JK MS Flip-Flop



Characteristic Equations

- Algebraic description of the next state table of a flip-flop.
- Obtained by constructing the K-Map for Q^+ in terms of present state and information input variables.

Function tables of SR, JK, D and T flip-flops

S	R	Q^+
0	0	Q'
0	1	0
1	0	1
1	1	-

(a)

J	K	Q^+
0	0	Q
0	1	0
1	0	1
1	1	\bar{Q}

(b)

D	Q^+
0	0
1	1

(c)

T	Q^+
0	Q
1	\bar{Q}

(d)

8]

Characteristic Equations

- Algebraic description of the next state table of a flip-flop.
- Obtained by constructing the K-Map for Q^+ in terms of present state and information input variables.

Function tables of SR, JK, D and T flip-flops

S	R	Q^\pm
0	0	Q^-
0	1	0
1	0	1
1	1	-

(a)

J	K	Q^+
0	0	Q^-
0	1	0
1	0	1
1	1	\bar{Q}

(b)

D	Q^+
0	0
1	1

(c)

T	Q^+
0	Q^-
1	\bar{Q}

(d)

Next state table of SR flip-flop

Row no.	S	R	Q	Q^+
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	-
7	1	1	1	-

Characteristic equation of SR flip-flop

		RQ			
		00	01	11	10
S		0	1	0	0
0		0	1	3	2
1		1	1	-	-
		4	5	7	6

$Q^+ = S + \bar{R} Q$
subject to $SR = 0$

Karnaugh map for next state table of SR flip-flop

Next state table of JK flip-flop

Row no.	J	K	Q	Q^+
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Characteristic equation of JK flip-flop

		KQ				
		00	01	11	10	
J		0	0	1	0	0
1	0	1	1	0	1	2
	1	4	5	7	6	

$$Q^+ = J\bar{Q} + \bar{K}Q$$

Next state table of D flip-flop

Row no.	D	Q	Q^+
0	0	0	0
1	0	1	0
2	1	0	1
3	1	1	1

Characteristic equation of D flip-flop

		Q	
		0	1
D		0	0
1	0	0	1
	1	1	1
		2	3

$$Q^+ = D$$

Next state table of T flip-flop

Row no.	T	Q	Q^+
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

Characteristic equation of T flip-flop

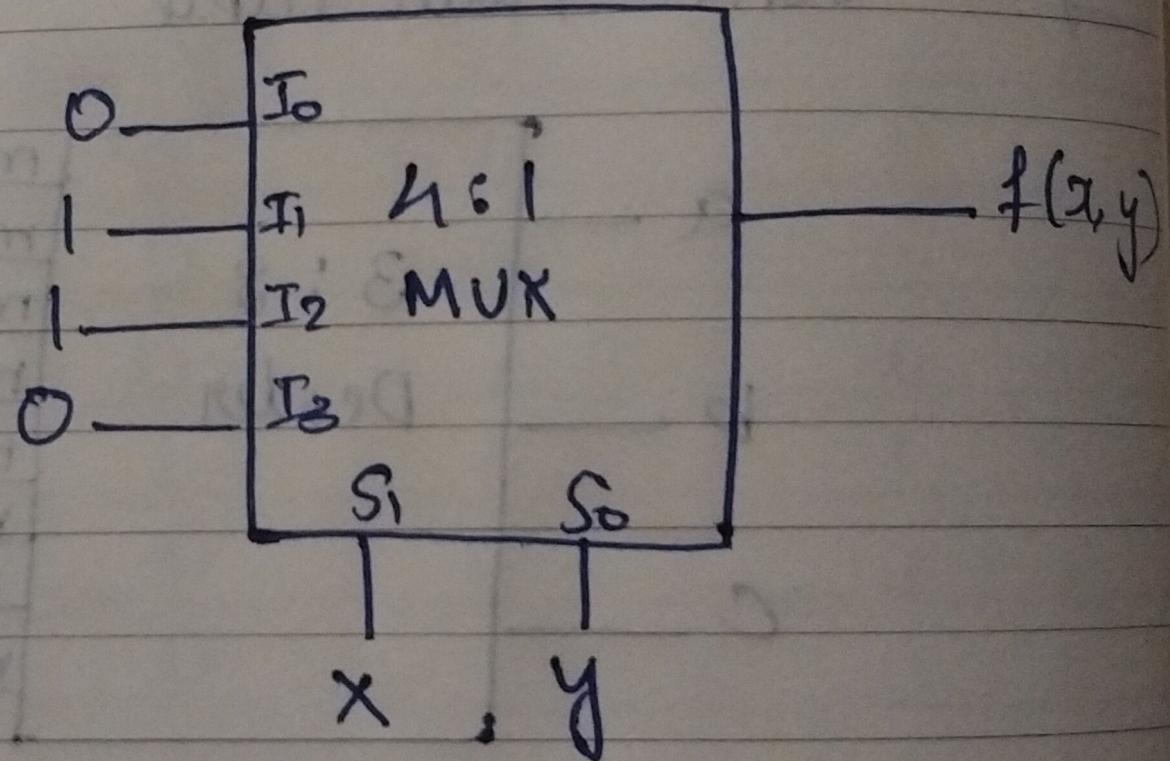
		Q	
		0	1
T	0	0	1
	1	0	1
		2	3

$$Q^+ = \bar{T}Q + T\bar{Q} = T \oplus Q$$

23) i))

WR 11 (07/0-207) FEBRUARY 2021
 $f(x, y) = \sum m(1, 2) [4:1 \text{ MUX}]$

x	y	f
0	0	0
0	1	1
1	0	1
1	1	0

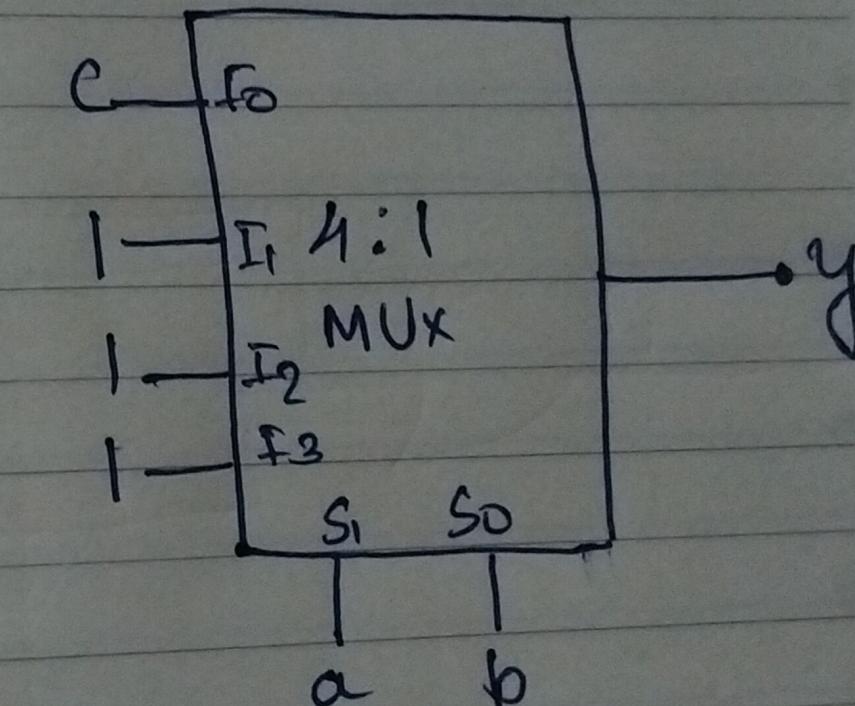


iii)

$$y = f(a, b, c) = a + b + c$$

 \rightarrow

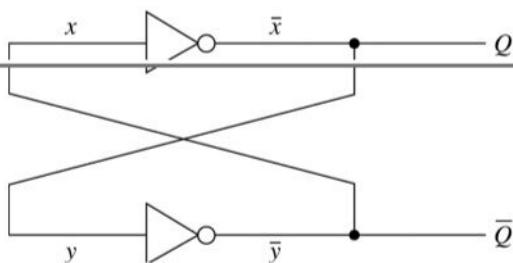
	a	b	c	y
10	0	0	0	0
11	0	0	1	1
12	0	1	0	1
1	0	1	1	1
	1	0	0	1
	1	0	1	1
	1	1	0	1



25)

Basic Bistable element

- Consists of two cross coupled inverters with two outputs named Q and \bar{Q}
 - Assume $x=0$ initially $\Rightarrow Q=y=1 \Rightarrow \bar{Q}=x=0$.
 - Assume $x=1$ initially $\Rightarrow Q=y=0 \Rightarrow \bar{Q}=x=1$.
- The circuit is stable in both the levels.



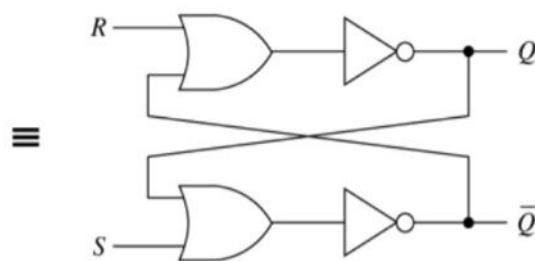
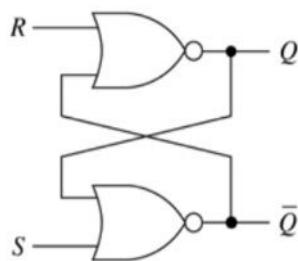
Bistable Element

- It is used to store binary symbols.
- Stored symbol is referred to as *content* or *state* of the element
- When the device is storing '1' it is said to be 'set' or in '1-state'. When storing '0' it is said to be 'reset' or in '0-state'

25)

R-S Latch [summary]

- Consists of two cross-coupled NOR gates.
- 2 inputs [R-Reset, S-Set] and 2 outputs



(a)

Inputs		Outputs	
S	R	Q^+	\bar{Q}^+
0	0	Q	\bar{Q}
0	1	0	1
1	0	1	0
1	1	0^*	0^*

*Unpredictable behavior
will result if inputs
return to 0 simultaneously

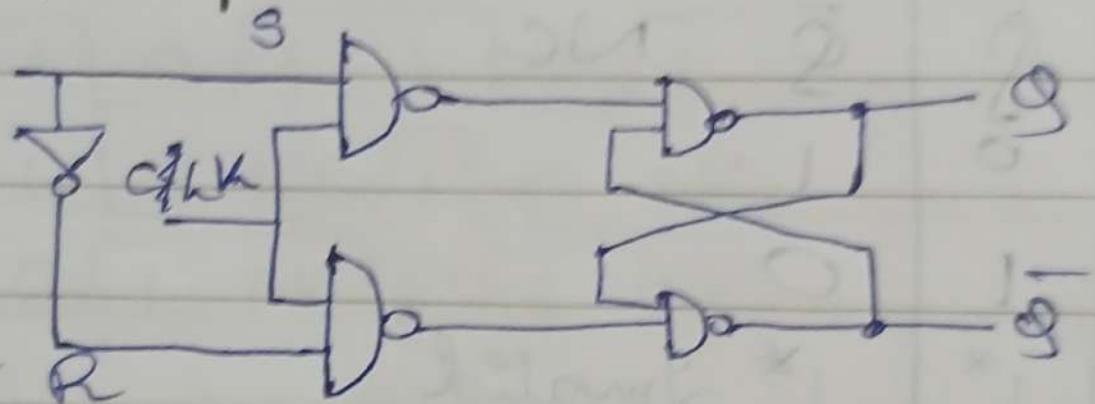
(b)

(c)
19EC302: Digital System Design

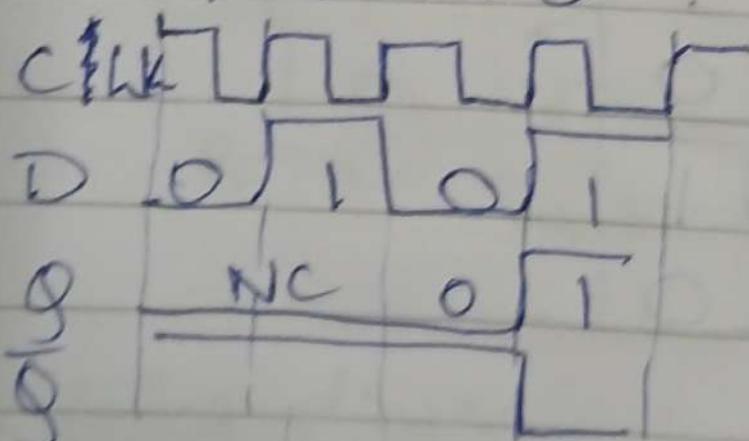
→ D Flip Flop

31.

$D=1$



Clk	D	Q^+	\bar{Q}^+
0	X	0	0
1	0	0	1
1	1	1	0



2.* The Basic Bistable Element :-

A circuit which can indefinitely store a '1' or '0' is called the basic bistable element.

- * It consists of cross coupled inverters with two outputs named Q and \bar{Q} .

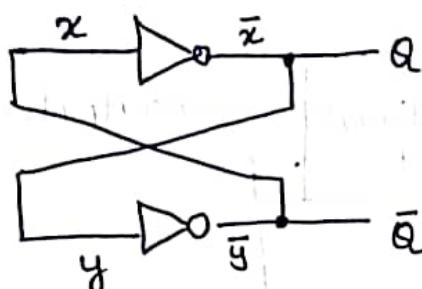


fig: The basic bistable element.

Let us assume that when power is switched on to the circuit,

input x is set to 1, therefore $\bar{x}=0$, $Q=0$.

This implies $y=0$ & $\bar{y}=\bar{Q}=1$. So, $Q=0$, $\bar{Q}=1$ and circuit continues in this state until power to the circuit is switched off.

Similarly when power is ON, if $x=0$, $\bar{x}=1$ & $Q=1$, this implies $y=1$, $\bar{y}=0$ & $\bar{Q}=0$.

Now $Q=1$ & $\bar{Q}=0$ is other stable state of the circuit in which it indefinitely remains until power to the circuit is switched off.

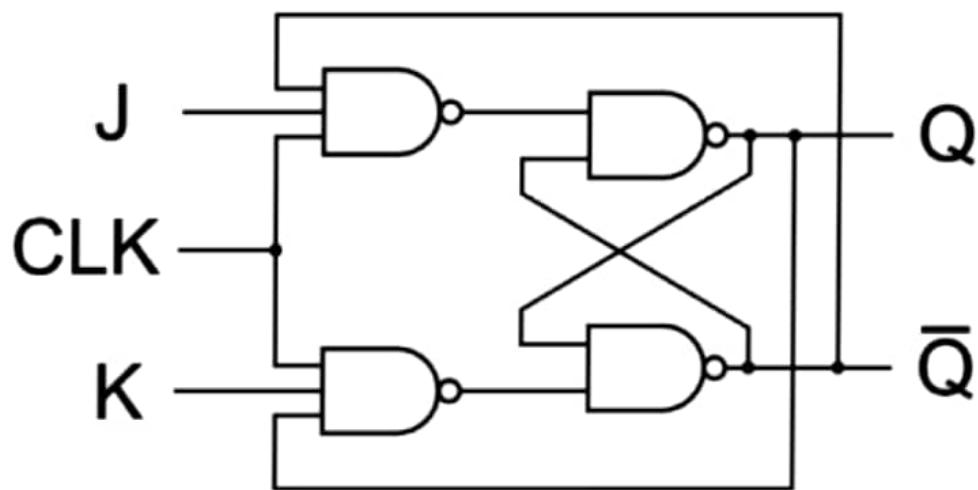
- * It is used to store binary symbols.

- * Stored symbol is referred to as content or state of the element.

- * When the device is storing '1' is said to be 'set' or in '1-state'.

- * When storing '0' is said to be 'reset' or in '0-state'.

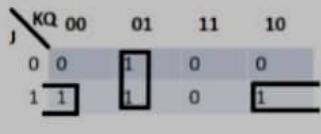
3.



Characteristic Equation of JK flip-flop

Function Table		
J	K	Q^*
0	0	Q
0	1	0
1	0	1
1	1	\bar{Q}

Next State Table			
J	K	Q	Q^*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



$$\text{Characteristic Equation: } Q^* = J\bar{Q} + \bar{K}Q$$

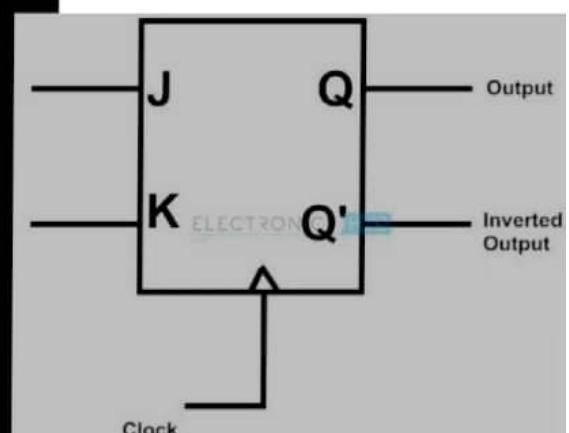
K-map

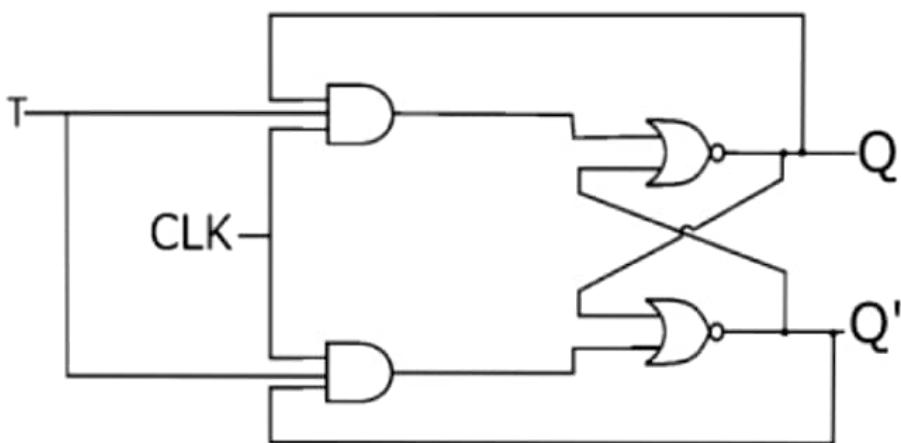
A **JK flip-flop** is like a memory switch that stores a 0 or 1 and changes when given a signal. It works like an improved SR flip-flop, solving the problem of an invalid state.

How it Works:

- **J = 0, K = 0 → No change** (Keeps the last value)
- **J = 0, K = 1 → Sets output to 0**
- **J = 1, K = 0 → Sets output to 1**
- **J = 1, K = 1 → Flips output (0 becomes 1, 1 becomes 0) on each clock pulse**

It only changes when the **clock** is active. This makes it useful in digital circuits for counting and memory storage.

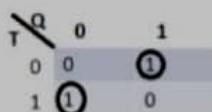




Characteristic Equation of T flip-flop

Function Table	
T	Q^+
0	Q
1	\bar{Q}

Next State Table		
T	Q	Q'
0	0	0
0	1	1
1	0	1
1	1	0



K-map

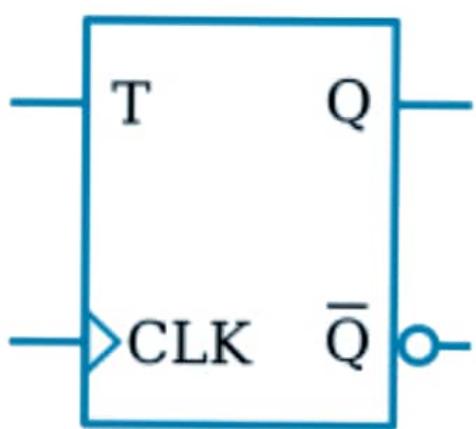
$$\text{Characteristic Equation} \quad Q^+ = T\bar{Q} + \bar{T}Q$$

A **T (Toggle) flip-flop** is a special type of JK flip-flop where both inputs (J and K) are tied together to form a single **T (Toggle)** input.

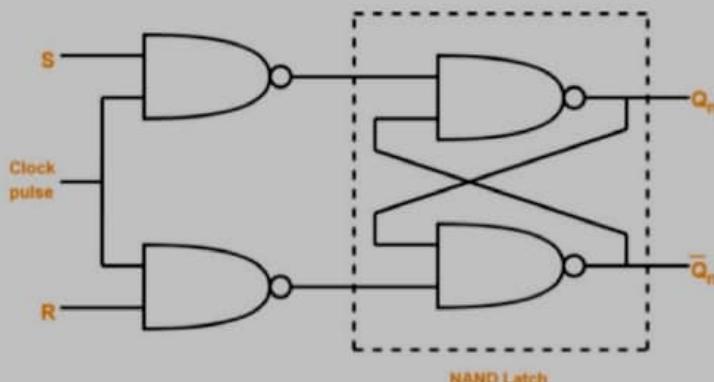
How It Works:

- $T = 0 \rightarrow$ **No change** (Keeps the last value)
- $T = 1 \rightarrow$ **Toggles output** (If it was 0, it becomes 1; if it was 1, it becomes 0)

The **T flip-flop only changes state when the clock is active**, making it useful in **counters** and **frequency dividers**.



SR flip-flop

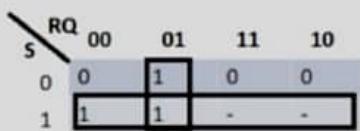


SR Flip Flop Using NAND Latch

Characteristic Equation of SR flip-flop

Function Table		
S	R	Q^+
0	0	Q
0	1	0
1	0	1
1	1	-

Next State Table			
S	R	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-



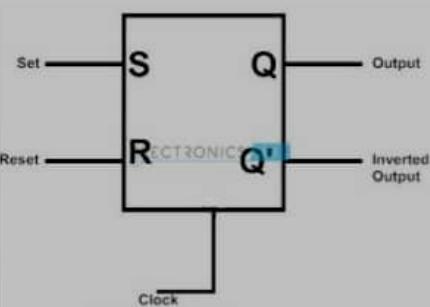
K-map

Characteristic Equation $Q^+ = S + \bar{R}Q$

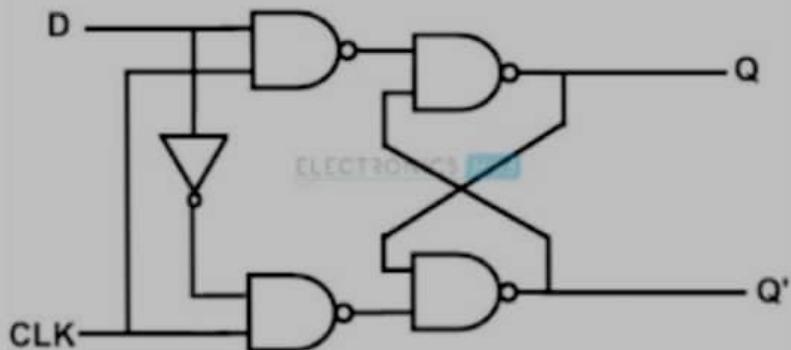
Operation:

1. $S = 1, R = 0 \rightarrow Q = 1$ (Set state)
2. $S = 0, R = 1 \rightarrow Q = 0$ (Reset state)
3. $S = 0, R = 0 \rightarrow$ **No change** (Previous state remains)
4. $S = 1, R = 1 \rightarrow$ **Invalid state** (Not allowed in NOR-based SR flip-flop)

Symbol of SR flip-flop



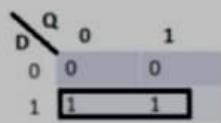
D flip-flop



Characteristic Equation of D flip-flop

Function Table	
D	Q'
0	0
1	1

Next State Table		
D	Q	Q'
0	0	0
0	1	0
1	0	1
1	1	1



$$\text{Characteristic Equation} \quad Q^+ = D$$

K-map

A **D (Data) flip-flop** is a memory circuit that stores a single bit. It ensures that the output follows the input with a delay controlled by a clock signal.

How It Works:

- $D = 0 \rightarrow Q$ becomes 0
- $D = 1 \rightarrow Q$ becomes 1

The output **Q** always follows **D** but only changes when the **clock is active**. This makes the **D flip-flop** useful in data storage and registers.

Symbol of D flip-flop

