DIGITAL SYSTEM DESIGN

Dec. 2019 / Jan. 2020

Module 1

1. a. Write the truth table of the logic circuit having and inputs a, b and c and an output $Y = ab\bar{c} + \bar{a}bc + abc$. Also simplify the Boolean expression and implement the logic circuit using NAND gates only. (06 Marks)

Ans: Given, $Y = ab\bar{c} + \bar{a}bc + abc$ → Sum of Product (SOP)

$$= \Sigma m(110, 011, 111)$$
$$= \Sigma m(6, 3, 7)$$

$A = 1$, $\bar{A} = 0$ } let

Truth Table:

a \ bc	00	01	11	10
0	0	0	1	0
1	0	0	1	1

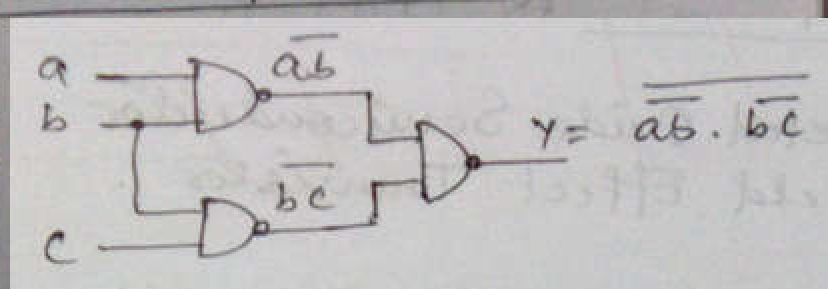
a	b	c	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Simplified Boolean expression:

$$\boxed{Y = bc + ab}$$

Logic circuit using NAND gate only:

$$Y = \overline{\overline{bc + ab}}$$
$$= \overline{\overline{bc} \cdot \overline{ab}}$$
$$= \overline{\overline{ab} \cdot \overline{bc}}$$



6. Minimize the following multiple output functions using K-map. (10 marks)

i) $f_1(a,b,c,d) = \sum m(1,5,7,8,9,10,11,13,15)$

ii) $f_2(a,b,c,d) = \sum m(1,2,6,7,8,13,14,15) + \sum d(3,5,12)$

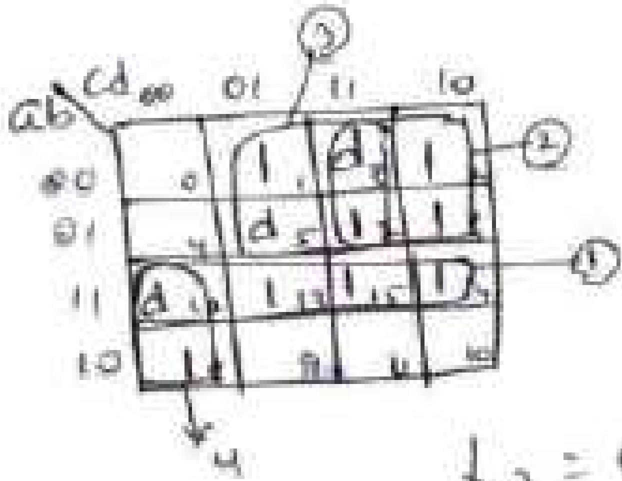
<u>Ans:</u> I) $f_1(a,b,c,d) = \sum m(1,5,7,8,9,10,11,13,15)$
$\hookrightarrow$ SOP (A=0, A=1) let
$\hookrightarrow$ 4 variable

<u>K-map simplification</u>:

ab \ cd	00	01	11	10
00	0 (0)	1 (1)	0 (3)	0 (2)
01	0 (4)	1 (5)	1 (7)	0 (6)
11	0 (12)	1 (13)	1 (15)	0 (14)
10	1 (8)	1 (9)	1 (11)	1 (10)

$f_1(a,b,c,d)$
$= \bar{c}d + bd + a\bar{b}$

ii) $f_2(a,b,c,d) = \sum m(1,2,6,7,8,13,14,15) + \sum d(3,5,12)$
$\hookrightarrow$ 4 variable $\hookrightarrow$ SOP



1 → ab

3 → $\bar{a}c$

2 → $\bar{a}d$

4 $a\bar{c}\bar{d}$

$$f_2 = ab + \bar{a}c + \bar{a}d + a\bar{c}\bar{d}$$

c. Define Canonical Minterm form and canonical Maxterm form. (04 marks)

Each individual term in standard SOP form is called **minterm** and each individual term in standard POS form is called **maxterm**. The concept of minterms and maxterms allows us to introduce a very convenient shorthand notations to express logical functions. Table 1.3.1 gives the minterms and maxterms for a three literal/variable logical function where the number of minterms as well as maxterms is $2^3 = 8$. In general, for an n-variable logical function there are 2^n minterms and an equal number of maxterms.

Variables			Minterms	Maxterms
A	B	C	m_i	M_i
0	0	0	$\bar{A}\,\bar{B}\,\bar{C} = m_0$	$A + B + C = M_0$
0	0	1	$\bar{A}\,\bar{B}\,C = m_1$	$A + B + \bar{C} = M_1$
0	1	0	$\bar{A}\,B\,\bar{C} = m_2$	$A + \bar{B} + C = M_2$
0	1	1	$\bar{A}\,B\,C = m_3$	$A + \bar{B} + \bar{C} = M_3$
1	0	0	$A\,\bar{B}\,\bar{C} = m_4$	$\bar{A} + B + C = M_4$
1	0	1	$A\,\bar{B}\,C = m_5$	$\bar{A} + B + \bar{C} = M_5$
1	1	0	$A\,B\,\bar{C} = m_6$	$\bar{A} + \bar{B} + C = M_6$
1	1	1	$A\,B\,C = m_7$	$\bar{A} + \bar{B} + \bar{C} = M_7$

Table 1.3.1 Minterms and maxterms for three variables

As shown in Table 1.3.1 each minterm is represented by m_i and each maxterm is represented by M_i, where the subscript i is the decimal number equivalent of the natural binary number. With these shorthand notations logical function can be represented as follows :

1. $f(A, B, C) = \bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,\bar{B}\,C + \bar{A}\,B\,C + A\,B\,\bar{C}$

$$= m_0 + m_1 + m_3 + m_6$$

$$= \sum m(0, 1, 3, 6)$$

2. $f(A, B, C) = (A + B + \bar{C})\,(A + \bar{B} + \bar{C})\,(\bar{A} + \bar{B} + C)$

$$= M_1 \bullet M_3 \bullet M_6$$

$$= \Pi\, M\,(1, 3, 6)$$

where $\sum$ denotes **sum of product** while Π denotes **product of sum**.

We know that logical expression can be represented in the truth table form. It is possible to write logic expression in standard SOP or POS form corresponding to a given truth table. The logic expression corresponding to a given truth table can be written in a standard sum of products form by writing one product term for each input combination that produces an output of 1.

A	B	C	Y	
0	0	0	0	
0	0	1	0	
0	1	0	1	← $\bar{A}B\bar{C}$
0	1	1	1	← $\bar{A}BC$
1	0	0	0	
1	0	1	0	
1	1	0	1	← $AB\bar{C}$
1	1	1	0	

Table 1.3.2

These product terms are ORed together to create the standard sum of products. The product terms are expressed by writing complement of a variable when it appears as an input 0, and the variable itself when it appears as an input 1. Consider, for example, the truth Table 1.3.2.

The product corresponding to input combination 010 is $\overline{A}B\overline{C}$, the product corresponding to input combination 011 is $\overline{A}BC$ and product corresponding to input combination 110 is $AB\overline{C}$. Thus the standard sum of products form is

$$f(A, B, C) = \overline{A}B\overline{C} + \overline{A}BC + AB\overline{C}$$

$$= m_2 + m_3 + m_6$$

The logic expression corresponding to a truth table can also be written in a standard product of sums form by writing one sum term for each output 0. The sum terms are expressed by writing complement of a variable when it appears as an input 1 and the variable itself when it appears as an input 0. Consider, for example, the truth Table 1.3.3.

A	B	C	Y	
0	0	0	1	
0	0	1	1	
0	1	0	0	$\leftarrow A + \overline{B} + C$
0	1	1	1	
1	0	0	1	
1	0	1	0	$\leftarrow \overline{A} + B + \overline{C}$
1	1	0	1	
1	1	1	1	

Table 1.3.3

The sum corresponding to input combinations 010 is $A + \overline{B} + C$, and the sum corresponding to input 101 is $\overline{A} + B + \overline{C}$. Thus, the standard product of sum form is

$$f(A, B, C) = (A + \overline{B} + C)(\overline{A} + B + \overline{C})$$

$$= M_2 \bullet M_5$$

OR

2. a) Convert the following Boolean function into their proper canonical form in decimal notation. (08 marks)

(i) $f = \bar{a}b + bc$ (ii) $f = (\bar{x}+y)(y+\bar{z})$

Ans: (i) $f = \bar{a}b + bc$ $\longrightarrow$ {SOP, let $A=1$, $\bar{A}=0$}

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

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

$= \underset{0\,1\,1}{\bar{a}bc} + \underset{0\,1\,0}{\bar{a}b\bar{c}} + \underset{1\,1\,1}{abc}$

$= \sum m(011, 010, 111)$

$= \sum m(3, 2, 7)$

$= \sum m(2, 3, 7)$ $= \pi(0,1,4,5,6)$

(ii) $f = (\bar{x}+y)(y+\bar{z})$ $\longrightarrow$ {POS, let $A=0$, $\bar{A}=1$}

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

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

$= \underset{1\,0\,0}{(\bar{x}+y+z)}\underset{1\,0\,1}{(\bar{x}+y+\bar{z})}\underset{0\,0\,1}{(x+y+\bar{z})}$

$= \prod M(100, 101, 001)$

$= \prod M(4, 5, 1)$

$= \prod M(1, 4, 5)$ $= \Sigma(0,2,3,6,7)$

(b) Simplify using Quine-Mccluskey minimization technique for the following function. (12 marks)

$f(w,x,y,z) = \Sigma(0,1,4,5,9,11,13,15)$

Ans: Given, $f(w,x,y,z) = \Sigma(0,1,4,5,9,11,13,15)$ $\rightarrow$ SOP [let $A=1$, $\bar{A}=0$]

Step 1

Minterm	w	x	y	z
0 →	0	0	0	0
1 →	0	0	0	1
4 →	0	1	0	0
5 →	0	1	0	1
9 →	1	0	0	1
11 →	1	0	1	1
13 →	1	1	0	1
15 →	1	1	1	1

Step 2

Grp	Minterm	w	x	y	z
0	0	0	0	0	0 ✓
1	1	0	0	0	1 ✓
	4	0	1	0	0 ✓
2	5	0	1	0	1 ✓
	9	1	0	0	1 ✓
3	11	1	0	1	1 ✓
	13	1	1	0	1 ✓
4	15	1	1	1	1 ✓

Step 3

Grp	Minterm	w	x	y	z
0	0,1	0	0	0	–
	0,4	0	–	0	0 ✓
1	1,5	0	–	0	1 ✓
	1,9	–	0	0	1
2	4,5	0	1	0	–
	~~4,9~~	–			
3	5,13	–	1	0	1
	9,11	1	0	–	1 ✓
	9,13	1	–	0	1 ✓
4	11,15	1	–	1	1 ✓
	13,15	1	1	–	1 ✓

Step 4

Grp	Minterm	W	x	Y	z
0	0,4,1,5	0	–	0	–
1	9,11,13,15	1	–	–	1
	9,13,11,15	1	–	–	1

$\therefore \boxed{f(w,x,y,z) = \bar{W}\bar{Y} + WZ}$

Prime Implicants	Binary Representation
0,4,1,5	0 – 0 – $\rightarrow \bar{W}\bar{Y}$
9,11,13,15	1 – – 1 $\rightarrow$ WZ

Prime Implicants	m_0	m_1	m_4	m_5	m_9	m_{11}	m_{13}	m_{15}
$\bar{W}\bar{Y} \rightarrow 0,4,1,5$	⊙	⊙	⊙	⊙				
$WZ \rightarrow 9,11,13,15$					⊙	⊙	⊙	⊙

Module-2

3) a) Design a combinational circuit that will multiply two 2-bit numbers. (12 Marks)

a	b	c	d	w	x	y	z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

i/p, o/p – 2M

$w = \Sigma(15)$

$x = \Sigma(10, 11, 14)$

$y = \Sigma(6, 7, 9, 11, 13, 14)$

$z = \Sigma(5, 7, 13, 15)$

$w = abcd$

$x = a\bar{b}c + ac\bar{d}$

$y = a\bar{c}d + a\bar{b}d + \bar{a}bc + bc\bar{d}$ → 8m

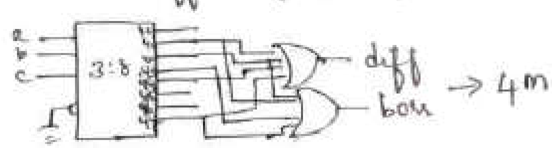
$z = bd$

Implementation (Design) → 2m

8+2+2 = 12

b) Implement full subtractor using 3:8 line decoder with active high outputs and active low enable input. (08 Marks)

T.T → 2m diff = $\Sigma(1,2,4,7)$ borr = $\Sigma(1,2,3,7)$ → 2m

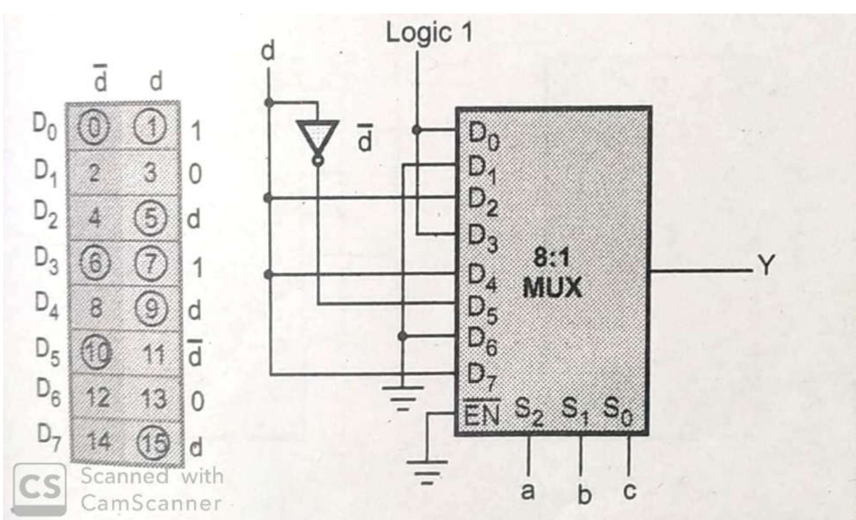


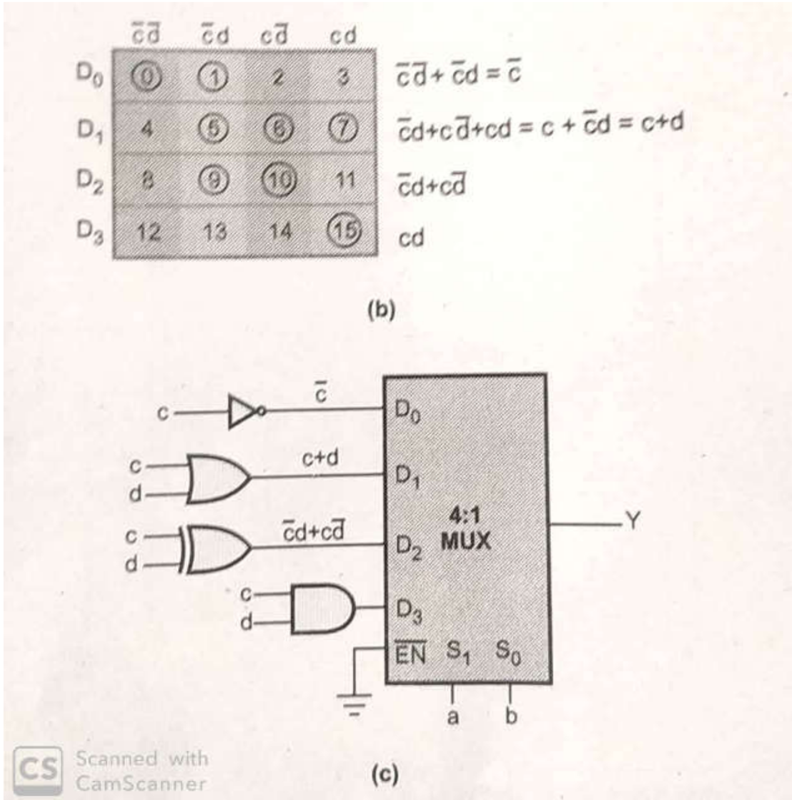
→ 4m

4+2+2 = 08

OR

4) a) Implement following using 8 to 1 MUX with a, b, c as select lines f(a,b,c,d)=∑(0,1,5,6,7,9,10,15). (08 Marks)





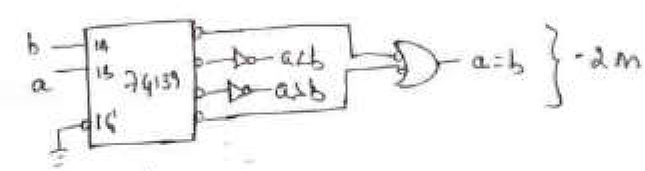
b) Implement a 1-bit comparator using 2:4 decoder 74139. (04 Marks)

I/P's	O/P's		
a b	a=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

$(a=b) = \Sigma(0,3)$

$a>b = \Sigma(2)$

$a<b = \Sigma(1)$ } -> 2m



c) Design a priority encoder for a system with three inputs, with the middle bit with highest priority encoding to 10, the MSB with the next priority encoding to 11, while the LSB with the least priority encoding to 01. (08 Marks)

I/P's			O/P's	
a	b	c	y	z
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0

→ 4M

K-map

$$\left.\begin{aligned} y &= a+b \\ z &= a\bar{b}+\bar{b}c \end{aligned}\right\} - 2M$$

Implementation - 2M

Module-3

5) a) With a neat diagram, explain the workig of master-slave JK flip-flop along with waveforms. (10 Marks)

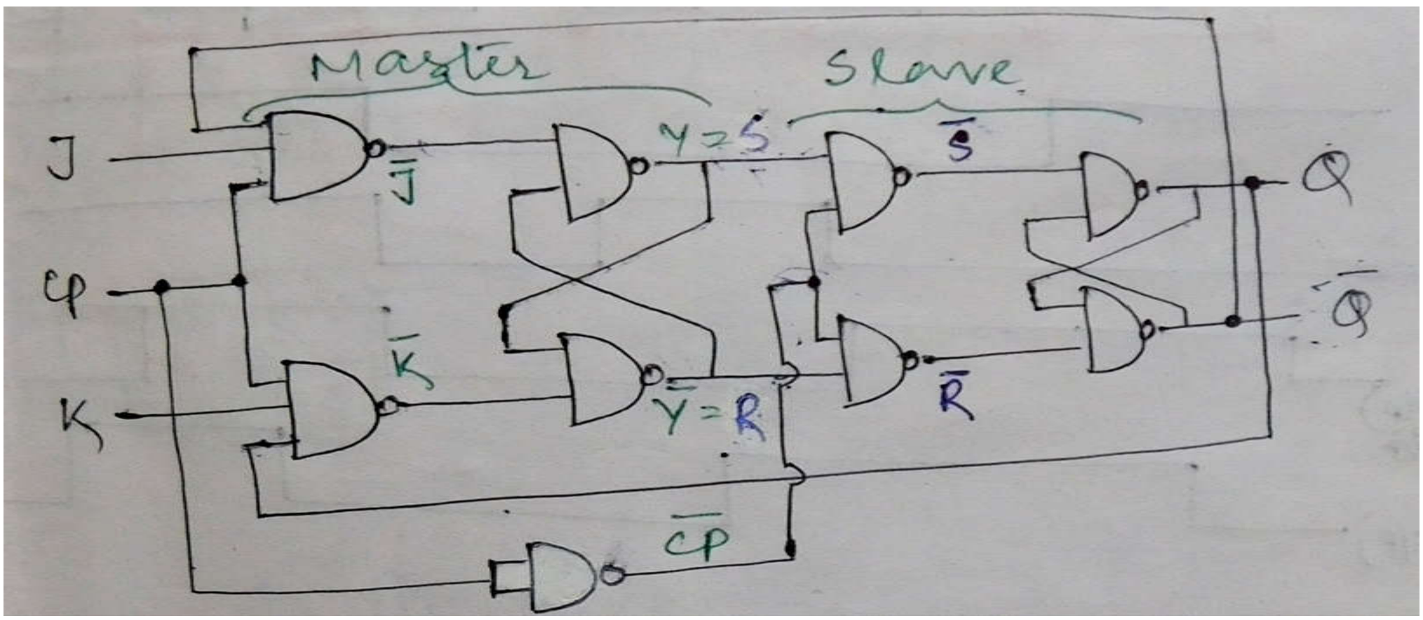


fig. shows the JK master slave flip flop. which consists of 2 flip-flops named as one is Master another is slave. Slave always follows master. Both the flip-flops are positive level triggered but an inverter is connected to the clock input of the slave flip-flop which forces it to trigger at the negative level. Therefore, the information present at the J and K inputs is transmitted to the o/p of master flip-flop on the positive clock pulse and it is held there until the negative clock pulse occurs, after which it is allowed to pass through to the output of the slave FF means slave flip-flop uses the SR input at the negative clock pulse to determine its o/p state Q and $\overline{Q}$.

Case(I): When, $J = K = 0$, the o/p of master remains same at the +ve clock. Thus the o/p of slave also remains same at the -ve clock.

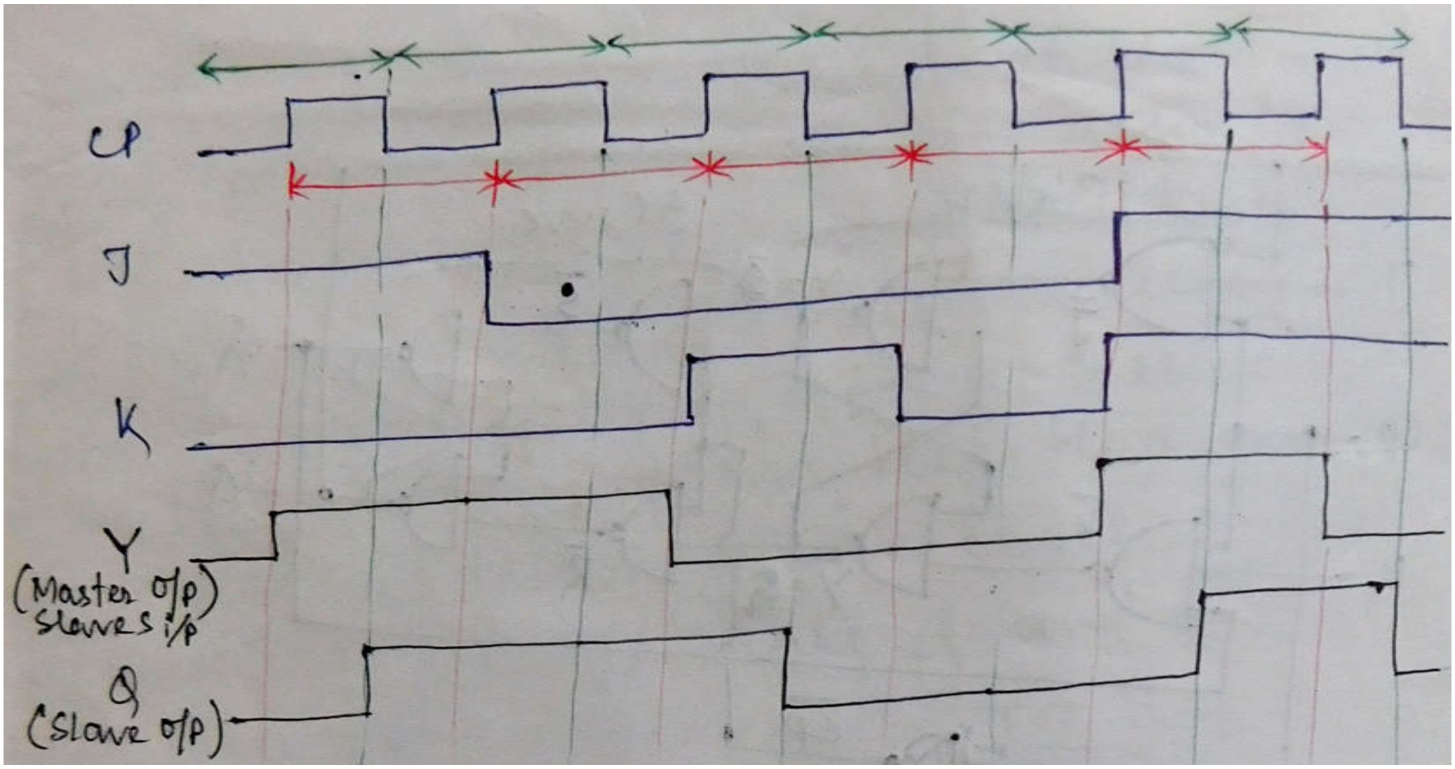
TT

	Clk	J K	Q_n
(I)	$\sqcap$	0 0	Q (NC)
(II)	$\sqcap$	0 1	0 (RESET)
(III)	$\sqcap$	1 0	1 (SET)
(IV)	$\sqcap$	1 1	$\bar{Q}$ (Toggle)

Case(II): When, $J = 0$ & $K = 1$, the master resets on the +ve clock. The high $\bar{Y}$ o/p of master goes to the R i/p of the slave. So, at -ve clock slave resets, again copying the action of the master.

Case(III): When, $J = 1$ & $K = 0$, the master sets on the positive clock. The high Y o/p of master drives the S i/p of the slave, so at -ve clock slave sets, copying the action of master.

Case(IV): When, $J = K = 1$, master toggles on the positive clock and slave then copies the o/p of master on the negative clock.



b) Explain switch debouncer using SR latch with waveforms. (10 Marks)

Ans. In a digital system, usually push button keys are used as a interfacing key. When these push button keys are pressed, then it bounces a few times, means closing and opening of the contact are randomly changed before providing a steady reading. So reading taken during this bouncing period may be faulty. This problem is known as key debounce which is undesirable and it must be avoided.

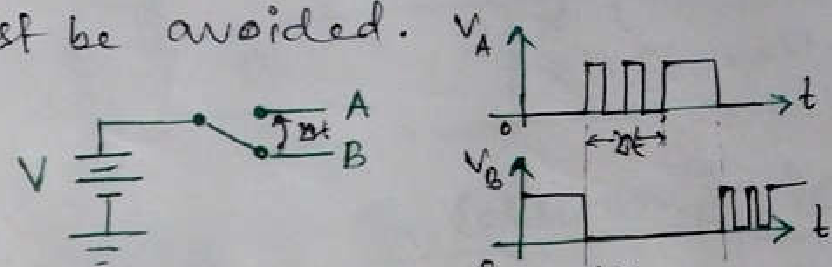


fig 1 Effect of key debounce.

One way to avoid key debounce problem is to use SR Latch. The circuit which is used to avoid keybounce with SR latch is called a switch debouncer. Here, Fig 1(a), shows a switch which is connected to B terminal so, B is 1 and A is 0. When it moves from B to A then during Δt time period it will bounces and a toggle condition is occured. So two pull down resistors are connected across the i/p terminal of SR latch as shown in fig. 2. When key is at position A then S=1 and Q=1 mean V. When transition happens between A to B then

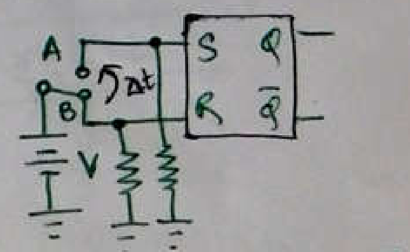


Fig 2. Switch debouncer

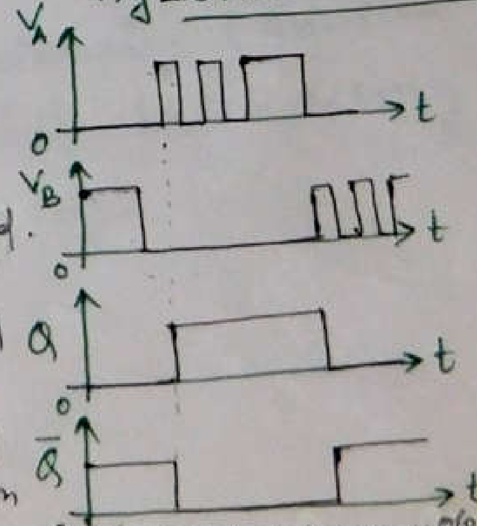


Fig: Switch debouncer o/p

it will take some time (let n sec.). During that time position A and B both are in high impedance state (means low current will flow). Due to these resistors (pull down) these high impedances are pulled down to zero. So whenever the contact point is in between A and B then i/p of SR latch will be zero which will maintain the steady state o/p of the latch.

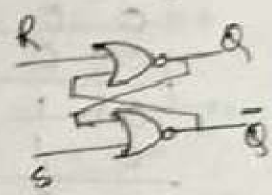
OR

6) a) Write the characteristic equation of SR, JK, D and T flip-flops. (08 Marks)

Ans: **Characteristics equation of SR flip-flop:**

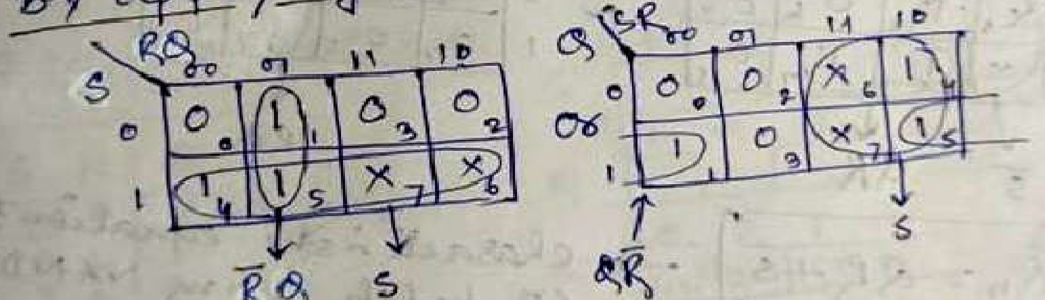
Truth Table of SR latch using NOR logic is given below:

I/P S	R	Q (Memory)	Q_n (next state)
0	0	0	0 (No change)
0	0	1	1 (No change)
0	1	0	0 (Reset)
0	1	1	0 (Reset)
1	0	0	1 (set)
1	0	1	1 (set)
1	1	0	X (Invalid)
1	1	1	X (Invalid)



Logic Diagram

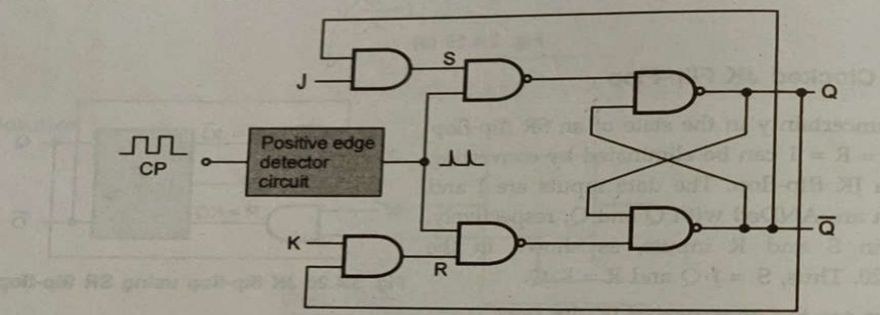
By applying Q_n column in K-map →



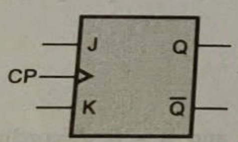
$\therefore \boxed{Q_n = \bar{R}Q + S}$ → characteristic Equation for SR Latch using NOR logic.

Characteristics equation of JK flip-flop:

The Fig. 3.4.21 shows the logic symbol, truth table and timing diagram of positive edge triggered JK flip-flop.



(a) Clocked JK flip-flop



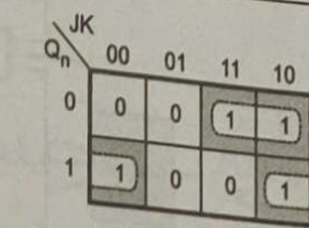
(b) Logic symbol

Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$\equiv$

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

(c) Truth table



$$Q_{n+1} = \overline{Q}_n J + Q_n \overline{K} = J\overline{Q}_n + \overline{K} Q_n$$

Fig. 3.4.21 (d) Characteristics equation

Characteristics equation of D flip-flop:

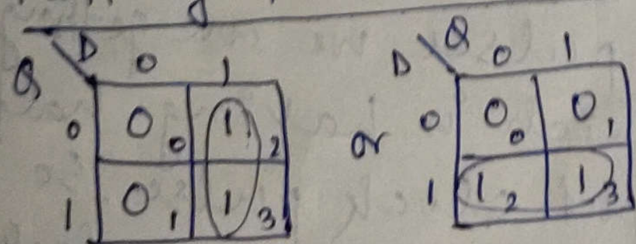
Characteristic Table:

D	Q	Qn	
0	0	0	} RESET
0	1	0	
1	0	1	} SET
1	1	1	

Excitation Table:

Q	Qn	D
0	0	0
0	1	1
1	0	0
1	1	1

Plotting Qn in K-map:

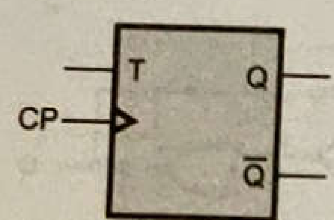


$\therefore \boxed{Q_n = D} \longrightarrow$ characteristic equation of D FF.

So Qn function follows D i/p at positive going edge of the clock pulse.

Characteristics equation of T flip-flop:

Digital System Design



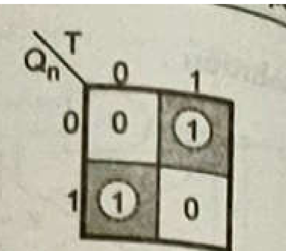
(a) Logic symbol

Q_n	T	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

$\equiv$

T	Q_{n+1}
0	Q_n
1	$\overline{Q_n}$

b) Truth table



$\therefore Q_{n+1} = T\overline{Q}_n + \overline{T}Q_n$

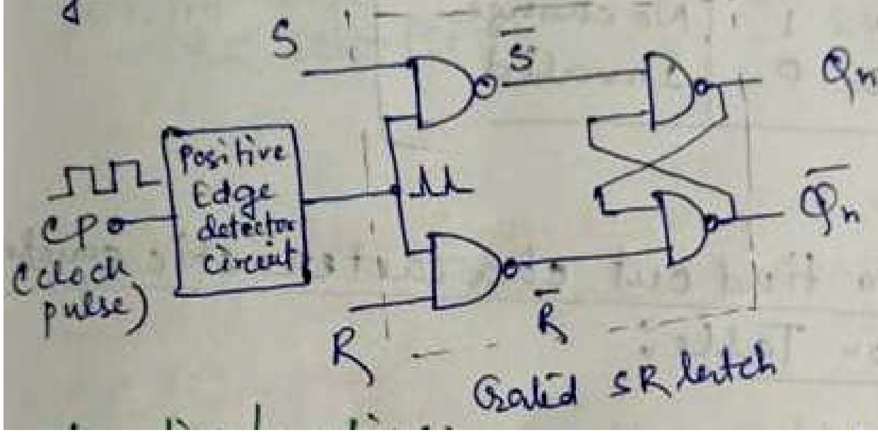
(c) Characteristic equation

b) Differentiate sequential logic circuit and combinational logic circuit. (04 Marks) circuits.

Combinational Circuits	Sequential Circuits
(I) In these circuits, the output variables are at all times dependent on the combination of i/p variables.	(I) In these circuits, the output variables depend not only on the present input variables but they also depend upon the past history of these input variables.
(II) Memory unit is not required in these circuits.	(II) Memory unit is required to store the past of input variables in these circuits.
(III) These are faster in speed because the delay between input and output is due to propagation delay of gates.	(III) These are slower than the combinational circuits.
(IV) These circuits are easy to design.	(IV) These circuits are harder to design.
(V) Example: Parallel Adder	(V) Example: Serial Adder.

c) Explain the operation of SR latch with an example. (08 Marks)

Ans) Logic Diagram of SR flip-flop using NAND gates are given below.



TT (Truth Table)

clk	S	R	Q_n
0	X	X	Q (No change)
↑	0	0	Q (No change)
↑	0	1	0 (RESET)
↑	1	0	1 (SET)
↑	1	1	X (Invalid)

Operation/ working:

Case (I): When, $CP = 0$ means, no clock pulse is applied, then input will not effect the circuit so memory (Q) element will be shown at the o/p (Q_n = next state) means 'no change' i.e 1st row of Truth Table.

Case (II): When, $S = R = 0$ and $CP = 1$ means clk pulse is applied then again like SR gated latch at o/p it will show No change i.e $Q_n = 0$. This indicates 2nd row of Truth Table.

Case (III): When, $S = 0$, $R = 1$ & clock pulse is applied then at o/p 'RESET' ($Q_n = 0$) condition will be reflected. This indicates the 3rd row of TT.

Case (IV): When, $S = 1$, $R = 0$ & clock pulse is applied then at o/p it will show 'SET' ($Q_n = 1$) condition. This indicates the 4rth row of TT.

Case (V): When, $S = R = 1$ & clock pulse is applied then at o/p it will show undefined or invalid which indicates 5th row of TT.

Module-4

7) a) Design a 4-bit register using positive edge triggered D-flip-flop to operate as indicated in the tablebelow: (12 Marks)

Mode Select		Data line selected	Register Operation
a_1	a_0		
0	0	d_0	Hold
0	1	d_1	Shift right
1	0	d_2	Shift left
1	1	d_3	Parallel load

Ans:

A register capable of shifting in one direction only is a unidirectional shift register. A register capable of shifting in both directions is a bidirectional shift register. If the register has both shifts (right shift and left shift) and parallel load capabilities, it is referred to as **Universal shift register.**

The Fig. 4.4.1 shows the 4-bit universal shift register. It has all the capabilities listed above. It consists of four flip-flops and four multiplexers. The four multiplexers have two common selection inputs S_1 and S_0, and they select appropriate input for D flip-flop. The Table 4.4.1 shows the register operation depending on the selection inputs of multiplexers. When $S_1S_0 = 00$, input 0 is selected and the present value of the register is applied to the D inputs of the flip-flops. This results no change in the register value. When $S_1S_0 = 01$, input 1 is selected and circuit connections are such that it operates as a right shift register. When $S_1S_0 = 10$, input 2 is selected and circuit connections are such that it operates as a left shift register. Finally, when $S_1S_0 = 11$, the binary information on the parallel input lines is transferred into the register simultaneously and it is a parallel load operation.

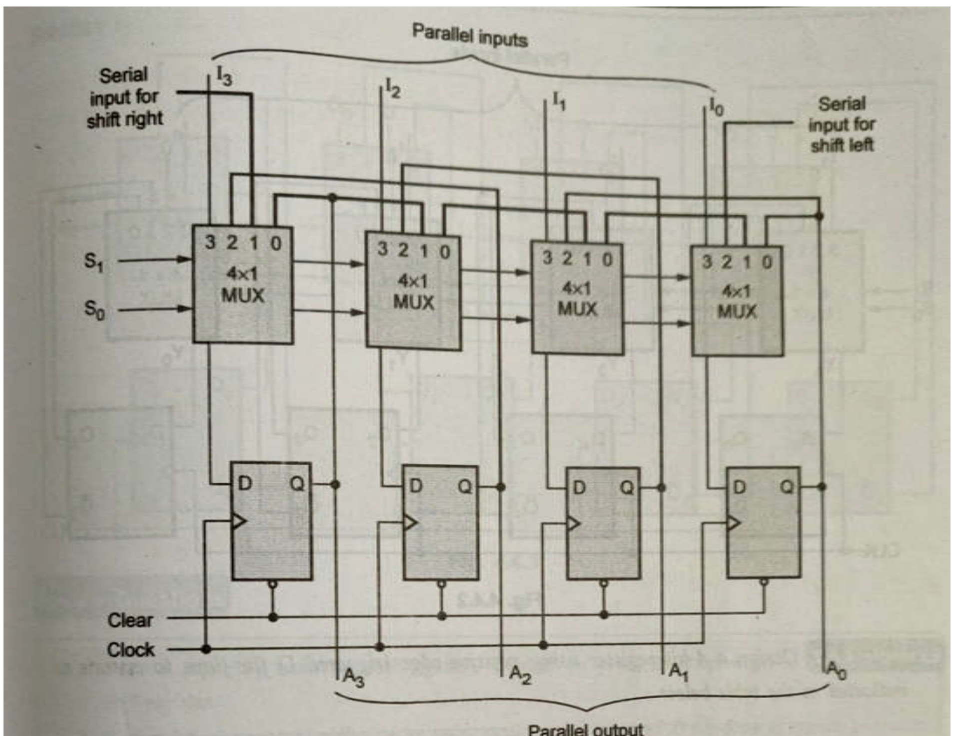


Fig. 4.4.1 4-bit universal shift register

Mode Select		Data line selected	Register Operation
a_1	a_0		
0	0	d_0	Hold
0	1	d_1	Shift right
1	0	d_2	Shift left
1	1	d_3	Parallel load

b) Design a 4-bit mod-8 Johnson counter and also write the count sequence table. (08 Marks)

In a Johnson counter, the Q output of each stage of flip-flop is connected to the D input of the next stage. The single exception is that the complement output of the last flip-flop is connected back to the D-input of the first flip-flop as shown in Fig. 5.4.9.

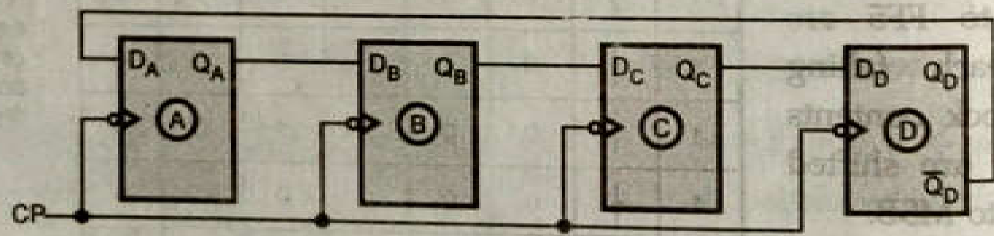


Fig. 5.4.9 Four-bit Johnson counter

As shown in Fig. 5.4.9 there is a feedback from the rightmost flip-flop complement output to the leftmost flip-flop input. This arrangement produces a unique sequence of states.

Initially, the register (all flip-flops) is cleared. So all the outputs, Q_A, Q_B, Q_C, Q_D are zero. The output of last stage, Q_D is zero. Therefore complement output of last stage, $\overline{Q}_D$ is one. This is connected back to the D input of first stage. So D_A is one. The first falling clock edge produces $Q_A = 1$ and $Q_B = 0, Q_C = 0$, $Q_D = 0$ since D_B, D_C, D_D are zero. The next clock pulse produces $Q_A = 1$, $Q_B = 1$, $Q_C = 0$, $Q_D = 0$. The sequence of states is summarized in Table 5.4.2. After 8 states the same sequence is repeated.

Clock pulse	Q_A	Q_B	Q_C	Q_D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

Table 5.4.2 Four-bit Johnson sequence

In this case, four-bit register is used. So the four-bit sequence has a total of eight states. Fig. 5.4.10 gives the timing sequence for a four-bit Johnson counter.

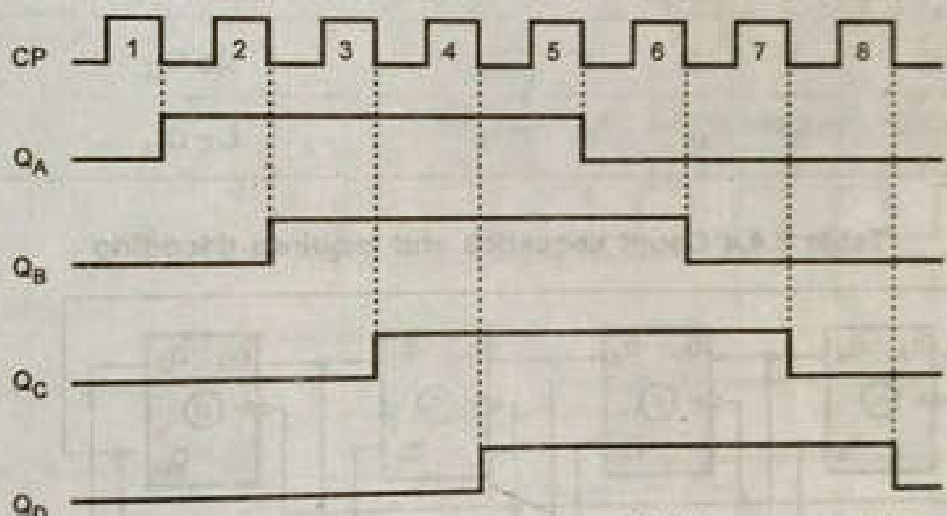
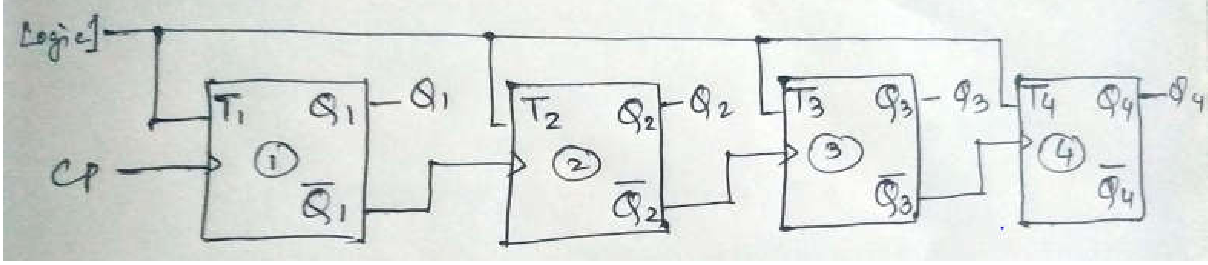


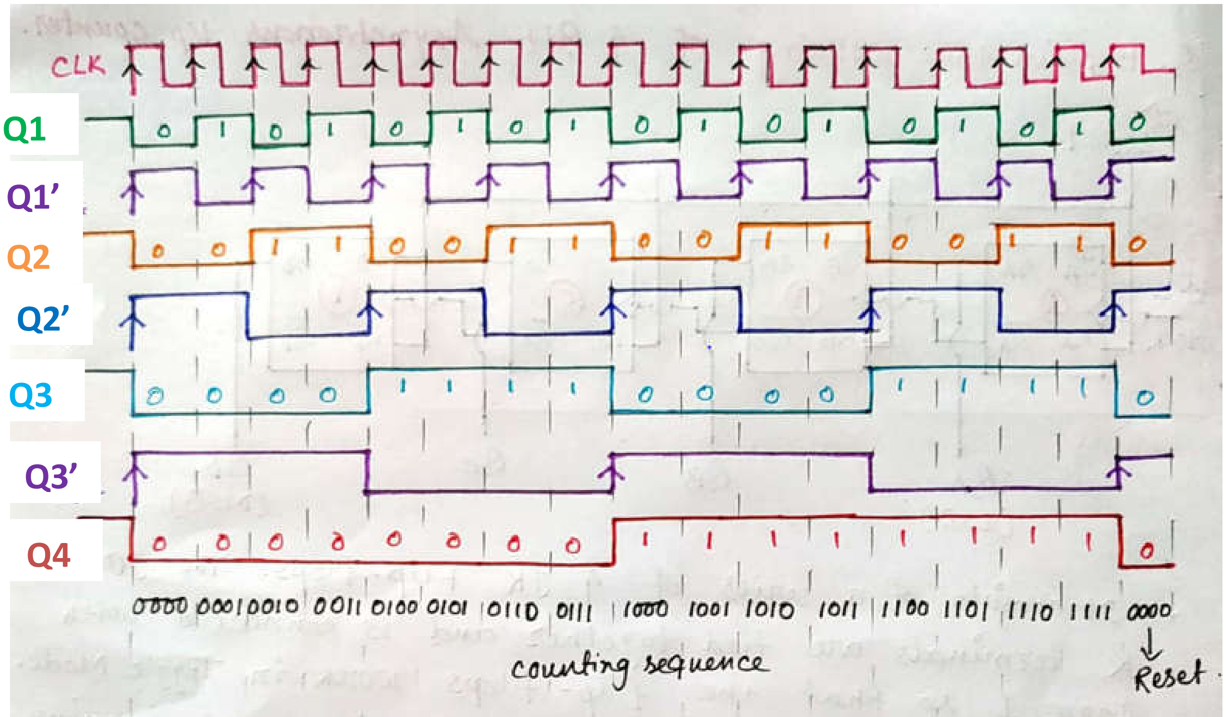
Fig. 5.4.10 Timing sequence for a four-bit Johnson counter

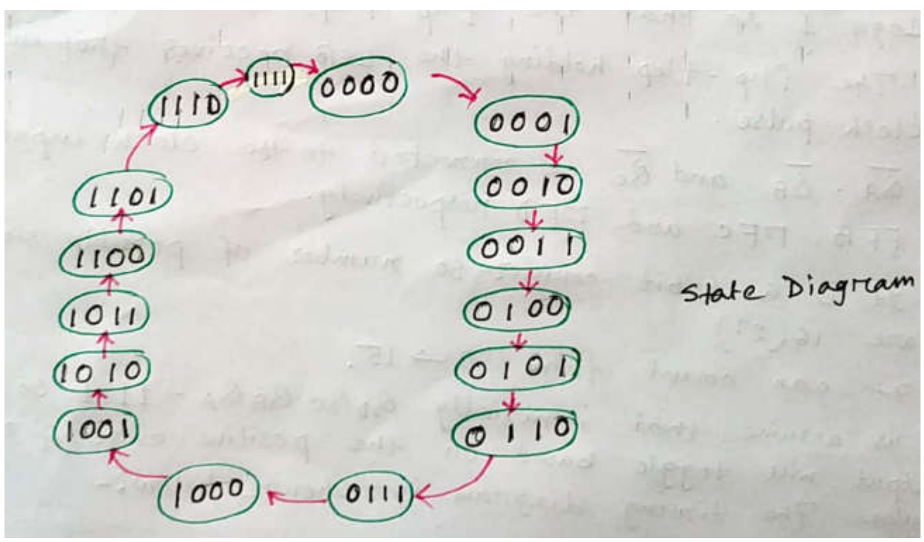
OR

8) a) Design a 4-bit binary ripple up counter using positive edge triggered t-flipflop with a count enable line. Write the counting sequence and relevant timing diagram.

Ans: 4-bit binary ripple up counter using positive edge triggered t- flip-flop.

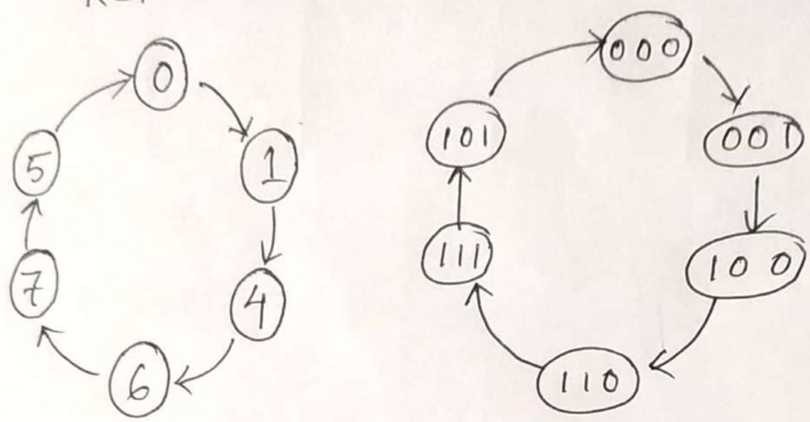



CLK
Q1
Q1’
Q2
Q2’
Q3
Q3’
Q4
0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111 0000
counting sequence
Reset


0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
State Diagram

(b) Design a synchronous counter to count the sequence 0, 1, 4, 6, 7, 5 and repeatusing positive edge triggered JK flip-flops.

⇒ Required numbers of flip-flops = 3.



Here invalid states are 2, 3. We are considering that when the counter reaches to any invalid states, then the counter will again start counting from innitial point that is 0.

Excitation Table for JK flipflop.

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Excitation Table for counter:-

Q_A	Q_B	Q_C	Q_{A+1}	Q_{B+1}	Q_{C+1}	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	1	0	0	1	X	0	X	X	1
0	1	0	0	0	0	0	X	X	1	0	X
0	1	1	0	0	0	0	X	X	1	X	1
1	0	0	1	1	0	X	0	1	X	0	X
1	0	1	0	0	0	X	1	0	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	1	0	1	X	0	X	1	X	0

Scanned with CamScanner

K Map simplification.

J_A

$Q_C \backslash Q_A Q_B$	00	01	11	10
0			X	X
1	1		X	X

$J_A = \overline{Q_B}\, Q_C$

K_A

$Q_C \backslash Q_A Q_B$	00	01	11	10
0	X	X		
1	X	X		1

$K_A = \overline{Q_B}\, Q_C$

J_B

$Q_C \backslash Q_A Q_B$	00	01	11	10
0		X	X	1
1		X	X	

$J_B = Q_A Q_C$

Scanned with CamScanner

K_B

$Q_C \backslash Q_A Q_B$	00	01	11	10
0	X	1		X
1	X	1	1	X

$K_B = \overline{Q_A} + Q_C$

J_C

$Q_C \backslash Q_A Q_B$	00	01	11	10
0	1		1	
1	X	X	X	X

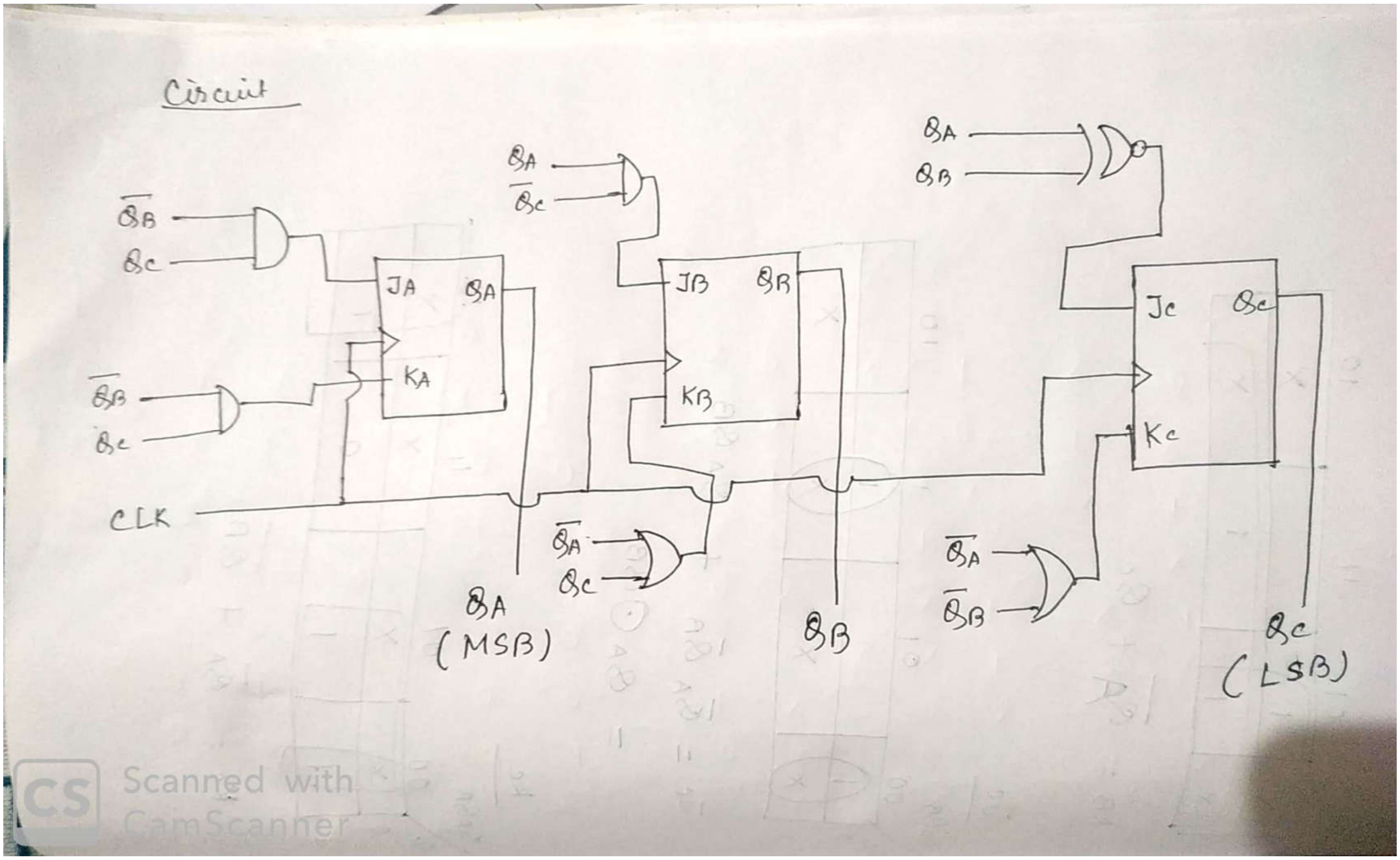
$J_C = \overline{Q_A}\,\overline{Q_B} + Q_A Q_B$

$= Q_A \odot Q_B$

K_C

$Q_C \backslash Q_A Q_B$	00	01	11	10
0	X	X	X	X
1	1	1	0	1

$K_C = \overline{Q_A} + \overline{Q_B}$


Circuit
$\overline{Q_B}$
Q_C
JA
QA
KA
$\overline{Q_B}$
Q_C
CLK
Q_A
$\overline{Q_C}$
JB
QB
KB
$\overline{Q_A}$
Q_C
QA
(MSB)
QB
Q_A
Q_B
Jc
Qc
Kc
$\overline{Q_A}$
$\overline{Q_B}$
Qc
(LSB)

Module-4

9) a) Explain Mealy and Moore model in a sequential circuit analysis.

Mealy Model :- When the output of the sequential circuit depends on both the present state of flip-flops and on the inputs, the sequential circuit is referred to as Mealy Model.

The block diagram of Mealy model is shown below:-

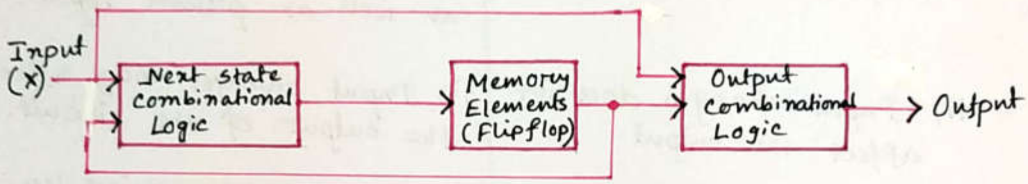
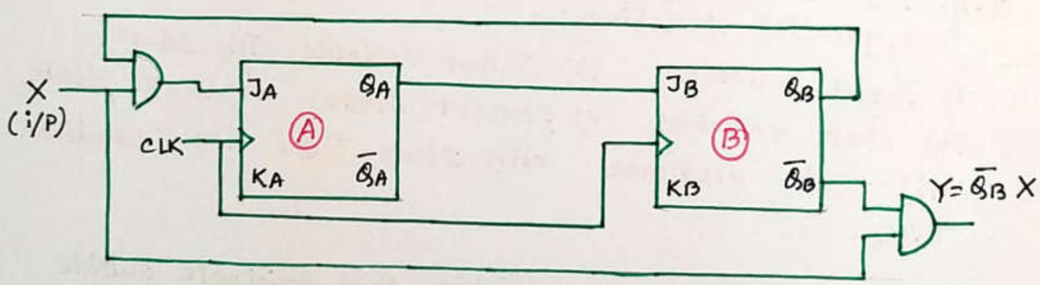


Fig:- Block diagram of Mealy Model.

Example of Mealy Model :-



In the above circuit, the input (X) is used to determine the output of the circuit (Y). So, the output of the above circuit is derived from the combination of present state of flip flops and input of the circuit.

Here, output $\boxed{Y = \bar{Q}_B X}$

State of Flipflop

input

⇒ Moore Model :- When the output of the sequential circuit depends only on the present state of the flip-flop, the sequential circuit is called as Moore model. The block diagram for Moore model is shown below :-

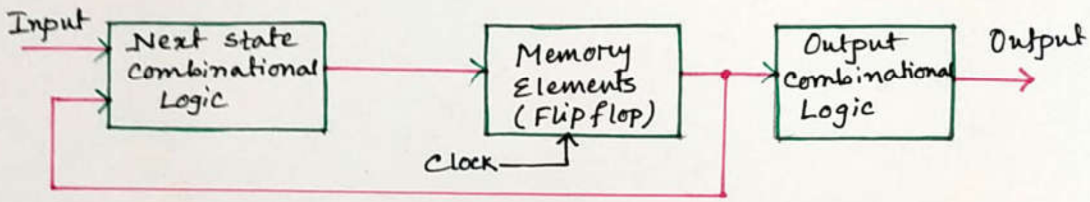
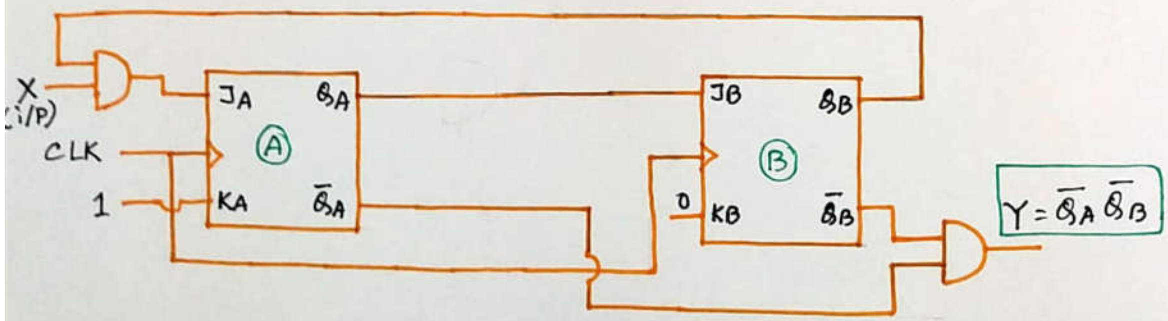


Fig:- Block diagram of Moore Model.

Example of Moore Model :-



In the above circuit, input (X) is used to determine the inputs of the flipflops. It is not used to determine the output. The o/p is derived using the states of flip-flops ($Y = \bar{Q}_A \bar{Q}_B$).

Here, Output $\boxed{Y = \bar{Q}_A \bar{Q}_B}$

State of Flipflop.

b) Design a sequential circuit using D-flip-flop for the state diagram. Show below in Fig.Q9(b). (12 Marks)

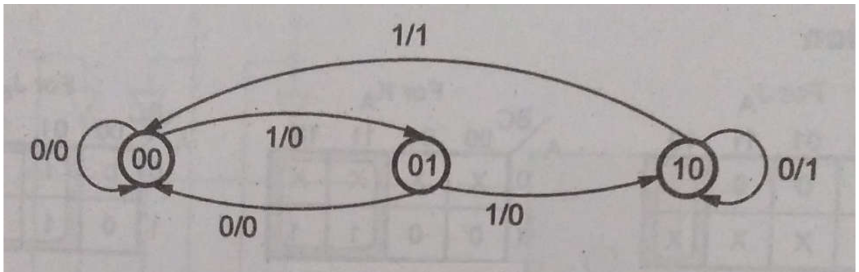
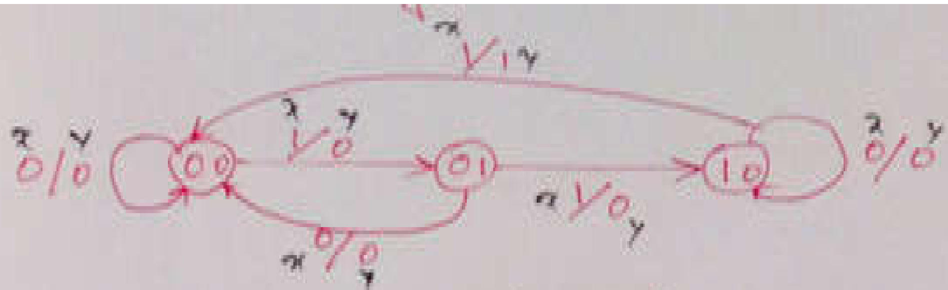


Fig.Q9(b)

Ans.



Ans: No. of states = 3 (valid)

No. of bits in each state = 2, so flip-flop required 2, means = $2^2 = 4$ is total no. of states. So invalid state is 1.

So using D-ff inputs will be D_A, D_B and respective outputs will be Q_A, Q_B. Let the sequential circuit input is x and output is y.

State Table:

PS		Next state				O/P	
		$x=0$		$x=1$		$x=0$	$x=1$
Q_A	Q_B	Q_A^+	Q_B^+	Q_A^+	Q_B^+	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	0	0	1
1	0	1	0	0	0	0	x
1	1	x	x	x	x	x	x

K-map Simplification

$D_A \rightarrow$ i/p

x \ AB	00	01	11	10
0	0	0	X	1
1	0	1	X	0

$D_A = xB + \bar{x}A$

$D_B \rightarrow$ i/p

x \ AB	00	01	11	10
0	0	0	X	0
1	1	0	X	0

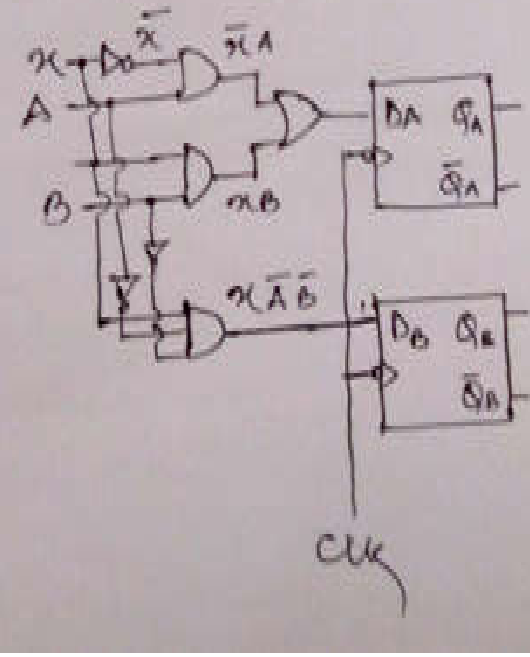
$D_B = x\bar{A}\bar{B}$

O/P $\rightarrow$

x \ AB	00	01	11	10
0	0	0	X	0
1	0	0	X	1

$y = xA$

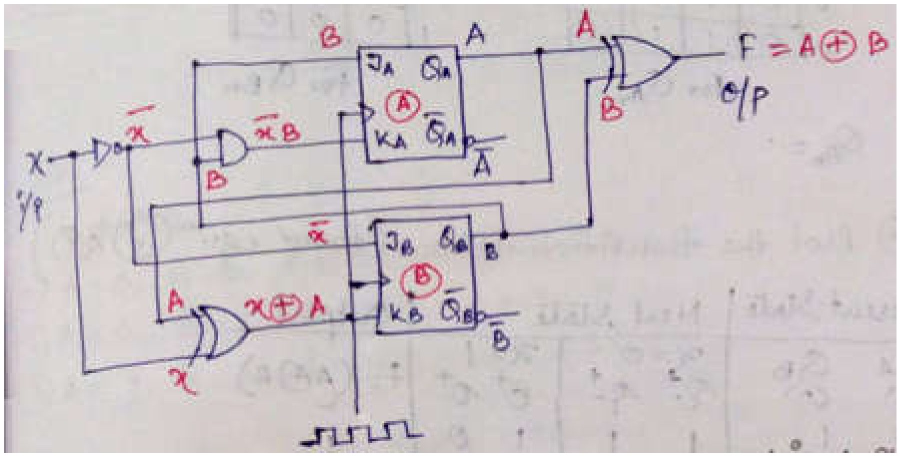
Sequential Circuit



O/P

OR

10) a) Construct the excitation table, transition table, state table and state diagram for the Moore sequential circuit shown in Fig.Q10(a).



Ans: (1) Determine the ff input equations and the output equations from the sequential circuit.

inputs equations, $J_A = B$, $K_A = \bar{x}B$, — (i)

$J_B = \bar{x}$, $K_B = x \oplus A$, — (ii)

output equation, $F = A \oplus B$

(2) characteristics equation of JK flip flop is

(Next State) $Q_n = J\bar{Q} + \bar{K}Q = Q^+$ — (iii)

By putting values from equations (i) and (ii) into equations (iii) we can write.

For (A) flip Flop:

$$Q_A^+ = J_A\bar{Q}_A + \bar{K}_A Q_A$$
$$= B\bar{Q}_A + \overline{\bar{x}B}\,Q_A$$
$$= B\bar{A} + (x+\bar{B})\cdot A \quad \text{— (iv)}$$

$$Q_B^+ = J_B\bar{Q}_B + \bar{K}_B Q_B$$
$$= \bar{x}\bar{Q}_B + \overline{x \oplus A}\cdot Q_B$$
$$= \bar{x}\bar{B} + \overline{x \oplus A}\cdot B \quad \text{— (v)}$$

Transition equations are (iv) and (v)

(7) Plot a nex-step maps for each flip-flop:

$x \backslash AB$	00	01	11	10
0	0	1	1	0
1	0	1	1	1

for Q_{An}

$x \backslash AB$	00	01	11	10
0	1	1	1	0
1	0	0	0	1

for Q_{Bn}

$Q_{An} =$.

(8) Plot the transition table. (from equ^n (iv) & (v))

Present State		Next State				Output
Q_A (A)	Q_B (B)	$x=0$: Q_A^+	$x=0$: Q_B^+	$x=1$: Q_A^+	$x=1$: Q_B^+	$F = (A \oplus B)$
0	0	0	1	0	0	0
0	1	1	1	1	0	1
1	0	1	1	1	0	1
1	1	0	0	1	1	0

Inputs:

$$Q_A^+ = B\bar{A} + (x + \bar{B})A \quad , \quad Q_B^+ = \bar{x}\bar{B} + \overline{x \oplus A} \cdot B$$

(1) $A = 0,\ B = 0,\ x = 0,\ Q_A^+ = 0.\bar{0} + (0 + \bar{0}).0 = 0$

$Q_B^+ = \bar{0}.\bar{0} + (\overline{0 \oplus 0}).0 = 1$

(2) $A = 0,\ B = 1,\ x = 0,\ Q_A^+ = 1.\bar{0} + (0 + \bar{1}).0 = 1$

$Q_B^+ = \bar{0}.\bar{1} + (\overline{0 \oplus 0}).1$

$= 1.0 + (\overline{0.\bar{0} + \bar{0}.0}).1 = 1$

(3) $A = 1,\ B = 0,\ x = 0,\ Q_A^+ = 0.\bar{1} + (0 + \bar{0}).1 = 1$

$Q_B^+ = \bar{0}.\bar{0} + (\overline{0 \oplus 1}).0 = 1$

(4) $A = 1,\ B = 1,\ x = 0,\ Q_A^+ = 1.\bar{1} + (0 + \bar{1}).1 = 0$

$Q_B^+ = \bar{0}.\bar{1} + (\overline{0 \oplus 1}).1 = 0$

(5) $A = 0,\ B = 0,\ x = 1,\ Q_A^+ = 0.\bar{0} + (1 + \bar{0}).0 = 0$

$Q_B^+ = \bar{1}.\bar{0} + (\overline{1 \oplus 0}).0 = 0$

(6) $A = 0,\ B = 1,\ x = 1,\ Q_A^+ = 1.\bar{0} + (1 + \bar{1}).0 = 1$

$Q_B^+ = \bar{0}\,\bar{1} + (\overline{1 \oplus 0}).1$

$= 0 + (\overline{\bar{1}.0 + 1.\bar{0}}).1 = 0$

⑦ $A=1, B=0, x=1, Q_A^+ = 0\cdot\bar{1} + (1+\bar{0})\cdot 1 = 1$

$Q_B^+ = \bar{1}\cdot\bar{0} + (\overline{1\oplus 1})\cdot 0 = 0$

⑧ $A=1, B=1, x=1, Q_A^+ = 1\cdot\bar{1} + (1+\bar{1})\cdot 1 = 1$

$Q_B^+ = \bar{1}\cdot\bar{1} + (\overline{1\oplus 1})\cdot 1 = 0 + (\overline{1\cdot\bar{1}+\bar{1}\cdot 1})1 = 1$

O/P

$f = A\oplus B$

① $A=0, B=0, f = 0\oplus 0 = 0\cdot\bar{0}\oplus\bar{0}\cdot 0 = 0$

② $A=0, B=1, f = 0\oplus 1 = 0\cdot\bar{1} + \bar{0}\cdot 1 = 1$

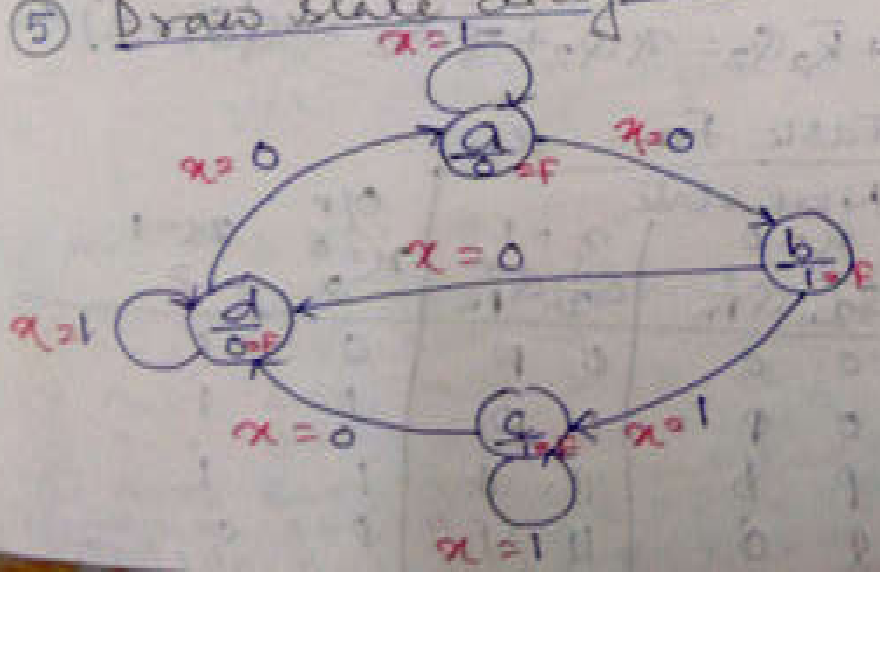
③ $A=1, B=0, f = 1\oplus 0 = 1\cdot\bar{0} + \bar{1}\cdot 0 = 1$

④ $A=1, B=1, f = 1\oplus 1 = 1\cdot\bar{1} + \bar{1}\cdot 1 = 0$

④ Draw the state table. 00 = a, 01 = b, 10 = c, 11 = d

Present state Q_A (A)	Present state Q_B (B)	Next state $x=0$: Q_A^+	Next state $x=0$: Q_B^+	Next state $x=1$: Q_A^+	Next state $x=1$: Q_B^+	Output $f=(A\oplus B)$
a = 0	0	b = 0	1	a = 0	0	0
b = 0	1	d = 1	1	c = 1	0	1
c = 1	0	d = 1	1	c = 1	0	1
d = 1	1	a = 0	0	d = 1	1	0

⑤ Draw state diagram.



b) Write short notes on: i) ROM, ii) RAM, iii) EPROM, iv) Flash Memory.

Read Only Memories (ROM)

- We can't write data in read only memories. It is non-volatile memory i.e. it can hold data even if power is turned off. Generally, ROM is used to store the binary codes for the sequence of instructions you want the computer to carry out and data such as look up tables. This is because this type of information does not change.
- It is important to note that although we give the name RAM to static and dynamic read/write memory devices, that does not mean that the ROMs that we are using are also not random access devices. In fact, most ROMs are accessed randomly with unique addresses.
- The Fig. 6.4.1 shows the typical configuration of a ROM cell. It consists of a transistor T and switch P. The transistor T is driven by the word line. The contents of cell can be read from the cell when word line is logic 1. A logic value 0 is read if the transistor is connected to ground through switch P. If switch P is open, a logic value 1 is read. The bit line is connected through a resistor to the power supply. A sense circuit at the end of the bit line generates the proper output value. Data is stored into a ROM when it is manufactured.
- There are four types of ROM : Masked ROM, PROM, EPROM and EEPROM or E^2PROM.

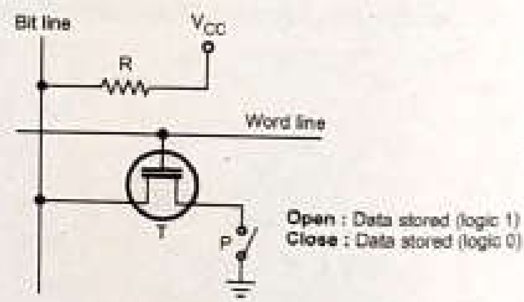


Fig. 6.4.1 ROM cell

Random Access Memory (RAM) / Read-Write Memories

- The memory in which the time taken to transfer information to or from any desired random location is always the same is called **random access memory**.
- There are two types of RAMs :
  - Static RAM
  - Dynamic RAM

1 Static RAM (SRAM)

Memories that consists of circuits capable of retaining their state as long as power is applied are known as **static memories**. These are Random Access Memory (RAM) and hence combinely called **static RAM memories**.

- Fig. 6.3.1 shows the one-bit memory cell for static RAM. The storage part of the cell is modeled by an SR latch with associated gates to form a D latch.

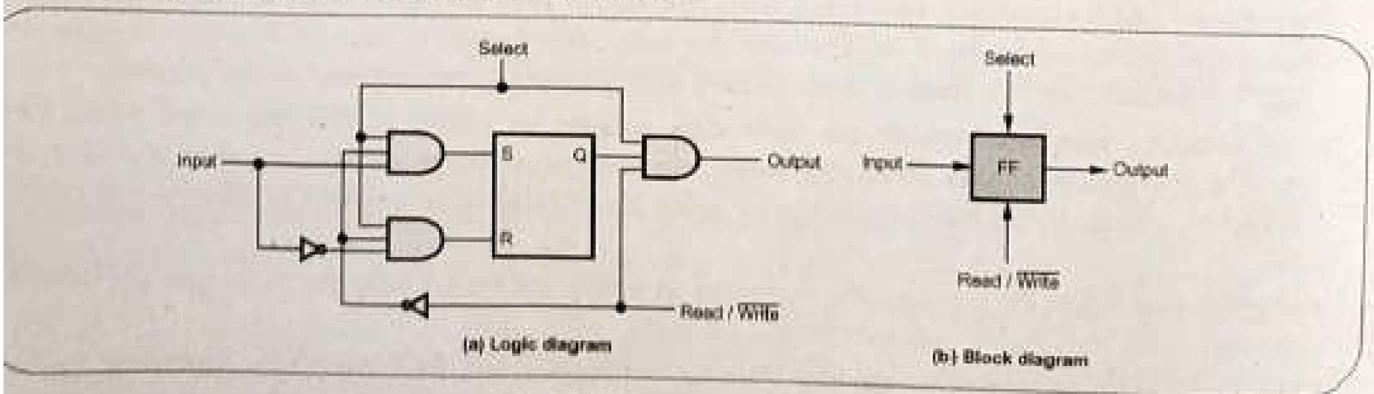


Fig. 6.3.1 One-bit binary cell

- The binary cell is capable of storing one bit in its internal latch.
- The select input enables the cell for reading and writing, and the read/write input determines the operation of the cell when it is selected.
- When Read/$\overline{\text{Write}}$ input is logic 1, read operation is performed; otherwise, write operation is performed.

Dynamic RAM (DRAM)

Dynamic RAM Cell

- Dynamic RAM stores the data as a charge on the capacitor. Fig. 6.3.2 shows the dynamic RAM cell. A dynamic RAM contains thousands of such memory cells. When COLUMN (Sense) and ROW (Control) lines go high, the MOSFET conducts and charges the capacitor. When the COLUMN and ROW lines go low, the MOSFET opens and the capacitor retains its charge. In this way, it stores 1 bit. Since only a single MOSFET and capacitor are needed, the dynamic RAM contains more memory cells as compared to static RAM per unit area.

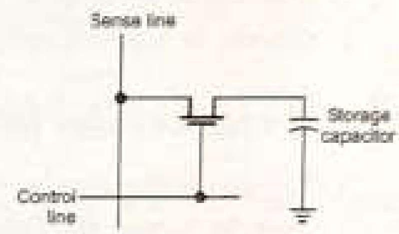


Fig. 6.3.2 Dynamic RAM

- The disadvantage of dynamic RAM is that it needs refreshing of charge on the capacitor after every few milliseconds. This complicates the system design, since it requires the extra hardware to control refreshing of dynamic RAMs.

EPROM (Erasable Programmable Read Only Memory)

- Erasable programmable ROMs use MOS circuitry. They store 1's and 0's as a packet of charge in a buried layer of the IC chip. EPROMs can be programmed by the user with a special EPROM programmer. The important point is that we can erase the stored data in the EPROMs by exposing the chip to ultraviolet light through its quartz window for 15 to 20 minutes, as shown in the Fig. 6.4.4.
- It is not possible to erase selective information, when erased the entire information is lost. The chip can be reprogrammed. This memory is ideally suitable for product development, experimental projects and college laboratories, since this chip can be reused many times, over.

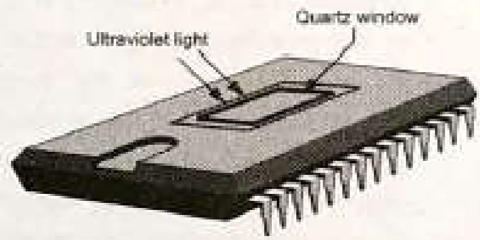


Fig. 6.4.4 EPROM

EPROM Programming :

- When erased each cell in the EPROM contains 1. Data is introduced by selectively programming 0's into the desired bit locations. Although only 0's will be programmed, both 1's and 0's can be presented in the data.
- During programming address and data are applied to address and data pins of the EPROM. When the address and data are stable, program pulse is applied to the program input of the EPROM. The program pulse duration is around 50 ms and its amplitude depends on EPROM IC. It is typically 5.5 V to 25 V. In EPROM, it is possible to program any location at any time - either individually, sequentially, or at random.

Flash Memory

Flash memories are read/write memories. In flash memories it is possible to read the contents of a single cell, but it is only possible to write an entire block of cells. A flash cell is based on a single transistor controlled by trapped charge.

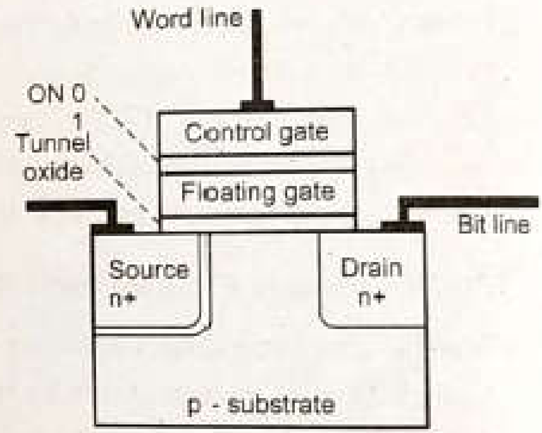


Fig. 6.4.5 Storage transistor with floating gate

A typical flash memory cell uses a floating gate to store a bit by the presence or absence of a charge. If the floating gate is not charged (i.e. neutral), then the device operates almost like a normal MOSFET, a positive charge in the control gate creates a channel in the p-substrate that carries a current from source to drain. If however the floating gate is negatively charged, then this charge shields the channel region somewhat from the control gate and prevents the formation of a channel between source and drain. The **threshold voltage** is the voltage applied to the control gate at which a transistor becomes conductive. The presence or the absence of a charge results in a more positive or more negative threshold voltage. In flash memory, programming (putting electrons into the floating gate) means writing a 0, erasing (removing the charge from the floating gate) means resetting the flash memory contents to 1.

The floating gate is completely surrounded by an isolation layer. This enables the floating gate transistor to be used as non-volatile memory.

Advantages and Drawbacks

Disadvantage : Editing Ability : A computer can read a specific byte from any address on the flash chip, but it can only erase and rewrite in block units.

Disadvantage : Lifetime : Flash memory does not have an infinite lifetime. It can only endure about 100,000 program/erase cycles.

Advantage : Efficiency : Because flash memory stores data without the use of moving mechanical parts, its operation requires less energy than older systems. Flash memory also stores data much more compactly than a hard drive can.