

PROM

④ There are fixed 'AND' array's and programmable OR arrays.

③ SOP fn in the standard form can only be implemented in PROM.

④ It is possible to decode any min-term in PROM.

PLA

'AND' and 'OR' both array's are programmable.

Any SOP fn can be implemented.

We can get any desired min-term by programming the AND & OR matrix.

PAL

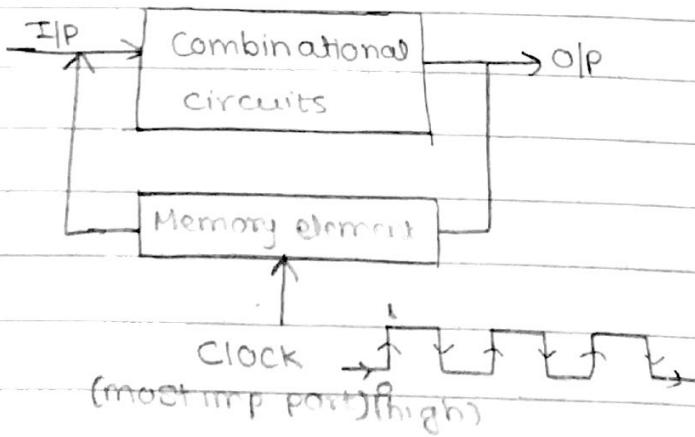
Only the AND array is programmable while the OR array is fixed.

Any SOP fn can be implemented.

We can get any desired min term by programming the AND matrix only.

Imp

SEQUENTIAL CIRCUITS :-



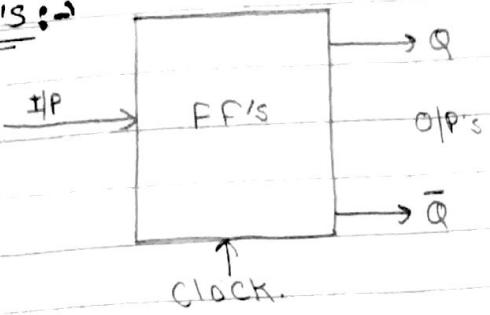
Types of sequential circuits :-

- ① Flip Flops (FFs)
- ② Registers.
- ③ Counters.

① FLIP FLOP'S :-

It can store only one bit of binary information.

① Flip flop's :-



Types of flip-flop's

- ① S-R flip-flop (Set-Reset)
- ② J-K flip-flop.
- ③ T- flip flop (Toggle)
- ④ D- flip-flop (Delay).

Assignment :- (Friday 04/09/15)

1) RAM → Static RAM
→ Dynamic RAM

2) ROM → PROM
→ EPROM
→ E²PROM

02/09/15
→ Present state (Q_n) → last value stored in memory (0/1)

→ Next state (Q_{n+1}) → Current output (0/1)

→ Clock skew →

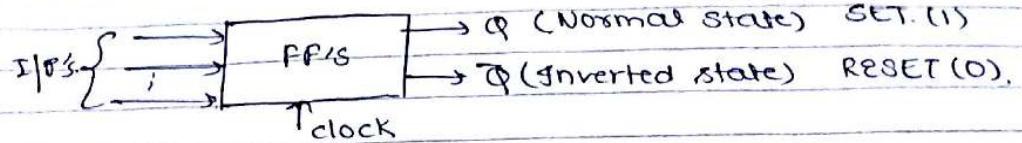
The diff. b/w the 2 clock periods - i.e. time taken by the clock to come from initial to final state

02/09/15

FUP-FLOPS (FF's).

A flip flop has two stable state. Flip flop is basically made up of universal gates that is NAND & NOR gates. It has two stable state which represent its output

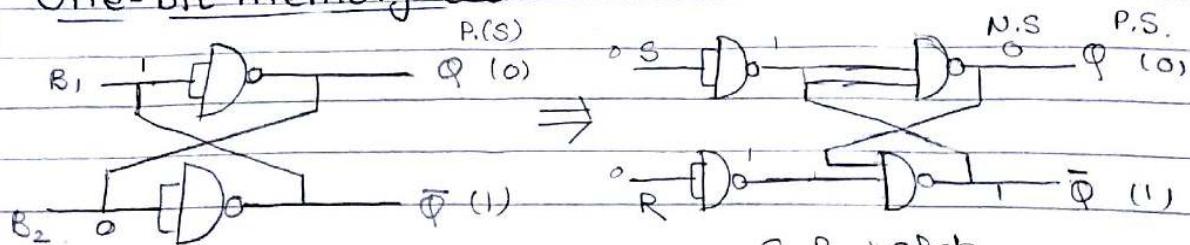
in which normal Q i.e. Q usually represent SET state (1), and \bar{Q} (Inverted Q/P) represent RESET (0) state



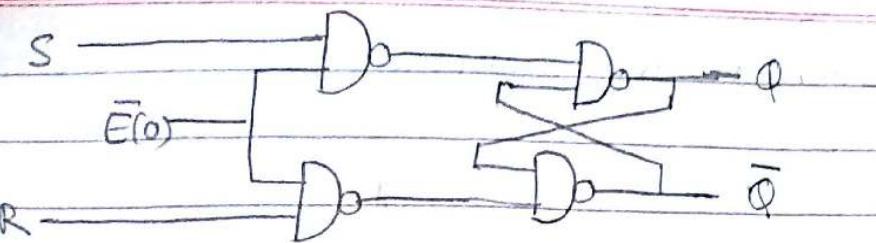
Latch :- Latch refers to non-clock flip-flop.

Gated latch :- $E \mid \text{clock} = 1$

One-bit memory cell :-

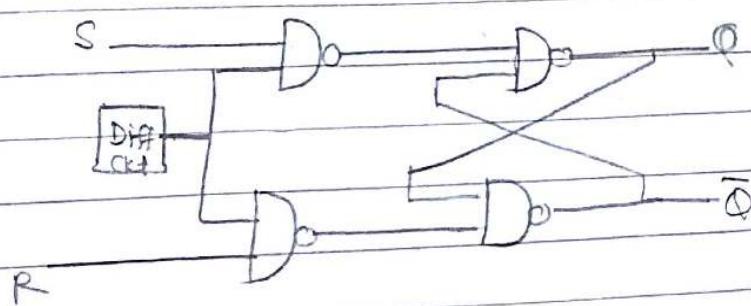


S	R	Q_n	\bar{Q}_n	Q_{n+1}	\bar{Q}_{n+1}	Comments
0	0	0	1	0	1	No change
0	0	1	0	1	0	
0	1	0	1	0	1	Reset
0	1	1	0	0	1	
1	0	0	1	1	0	Set
1	0	1	0	1	0	
1	1	1	0	X	X	Not specify
1	1	0	1	X	X	Active low latch. (NAND gate)



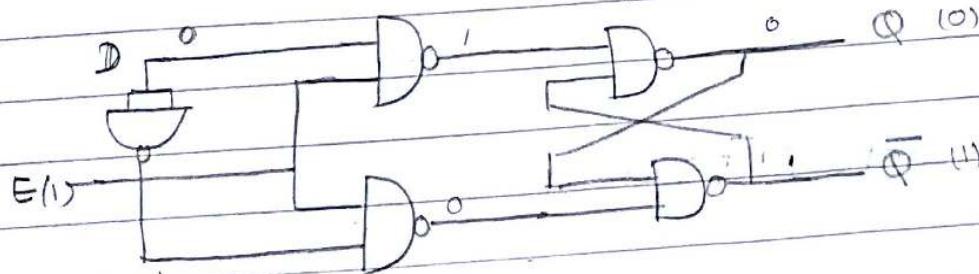
\bar{E}	S	R	Comment
0	0	0	N.C.
0	0	1	Reset
0	1	0	Set
0	1	1	N.A.

GATED S-R Latch. (Edge triggering)

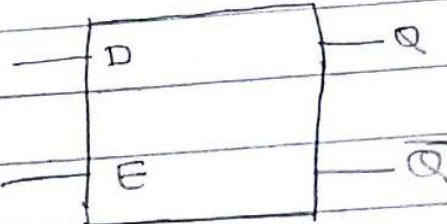


clk	\bar{clk}	S	R	Com
↑	↓	0	0	NC
↑	↓	0	1	R
↑	↓	1	0	S
↑	↓	1	1	N.F.

Gated D-latch (Level triggering D-FF) (Delay FF)

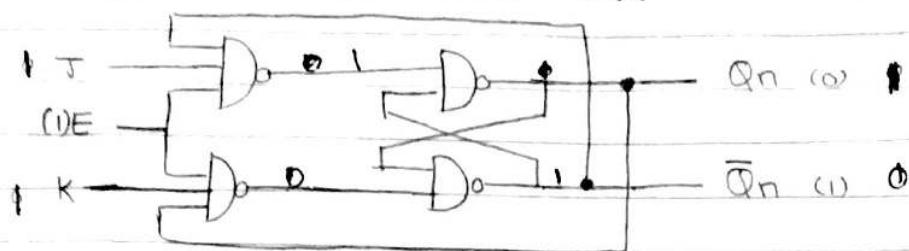


E	D	Comment
0	0	Reset
1	1	Set



09loglis

Gated JK latch (level triggered JKFF)

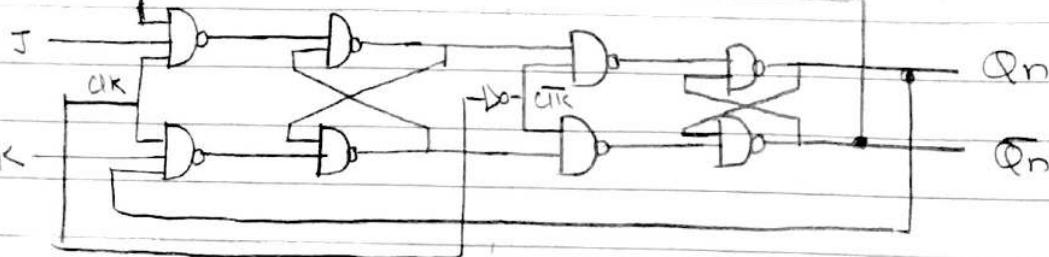


E	J	K	Comments	at t+1
1	0	0	No change	Qn 0 Q̄n+1 1
1	0	1	Reset	Qn 1 Q̄n+1 0
1	1	0	Set	Qn 1 Q̄n+1 0
1	1	1	Toggling	Qn 0 Q̄n+1 1

Race-around condition.

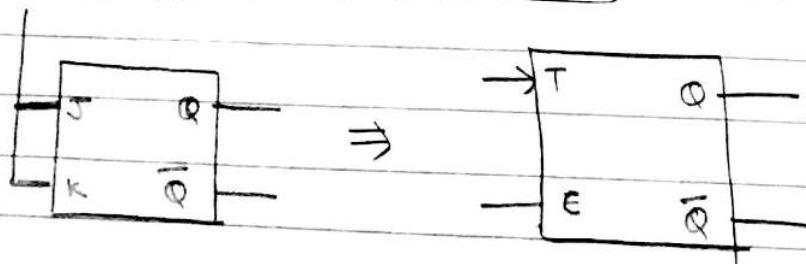
Master

clock



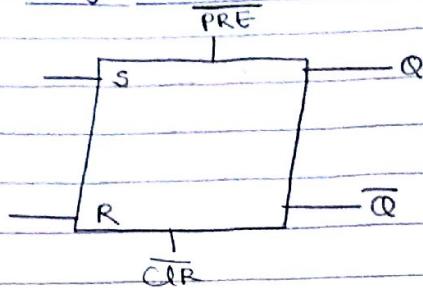
Master slave flip-flop

Gated Toggle latch (LT TFF)



E	T	Comments
1	0	NC
1	1	Toggle

Asynchronous I/P's for ex-nite S/P's



$\overline{PRE} \Rightarrow$ preset = 1 (Pr)
 $\overline{CLR} \Rightarrow$ clear = 0 (Cr)

Conversion of one type FF to another type :-

1) S-R FF

<u>S</u>	<u>R</u>	<u>Q_n</u>	<u>Q_{n+1}</u>	<u>Q_n</u>	<u>Q_{n+1}</u>	<u>S</u>	<u>R</u>
0	0	0	0	0	0	0	X
0	0	1	1	0	1	1	0
0	1	0	0	0	0	0	1
0	1	1	0	1	1	X	0
1	0	0	1	1	0		
1	0	1	1	1	1		
1	1	0	X	1	0		
1	1	1	X	0	1		

Truth table

11log15

Excitation table

2) D-FF

<u>D</u>	<u>Q_n</u>	<u>Q_{n+1}</u>	<u>Q_n</u>	<u>Q_{n+1}</u>	<u>D</u>
0	0	0	0	0	0
0	1	0	0	1	1
1	0	1	1	0	0
1	1	1	1	1	1

Truth-table

Excitation table

3) J-K FF

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

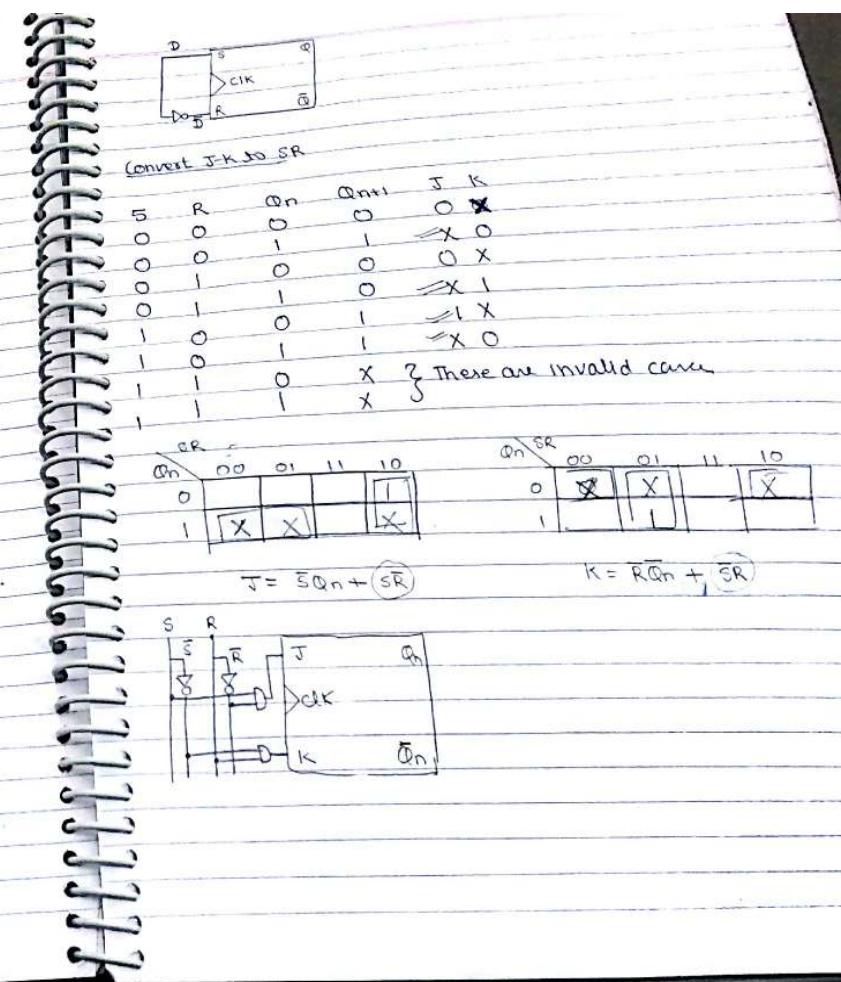
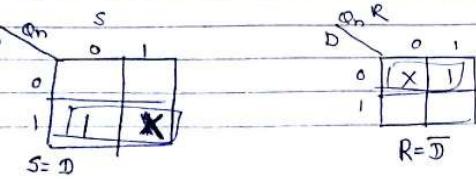
4) T-FF :-

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

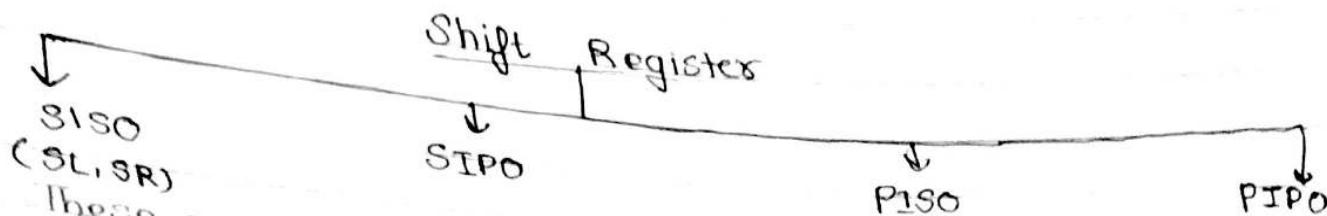
Q_n	00	01	11	10
0	0	0	1	0
1	X	X	X	1

~~S-R FF to D-FF :-~~

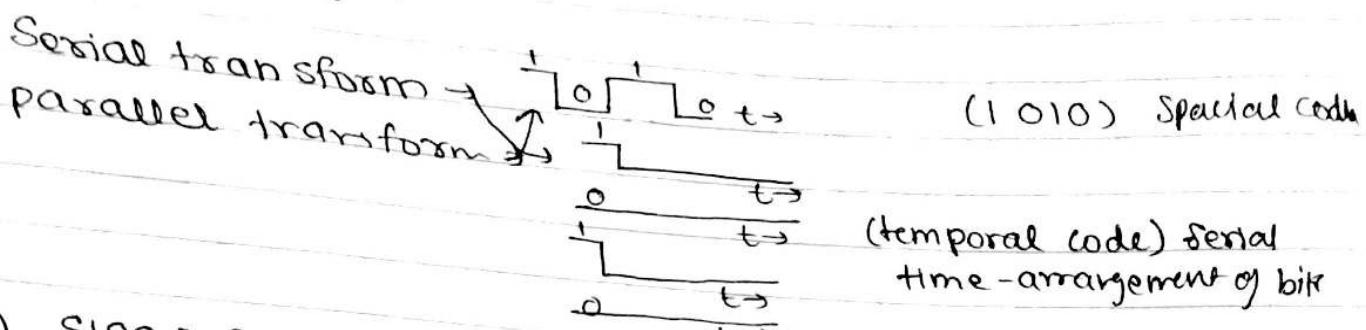
D	Q_n	Q_{n+1}	S	R
0	0	0	0	X
0	1	0	0	1
0	1	1	0	0
1	1	X	0	0



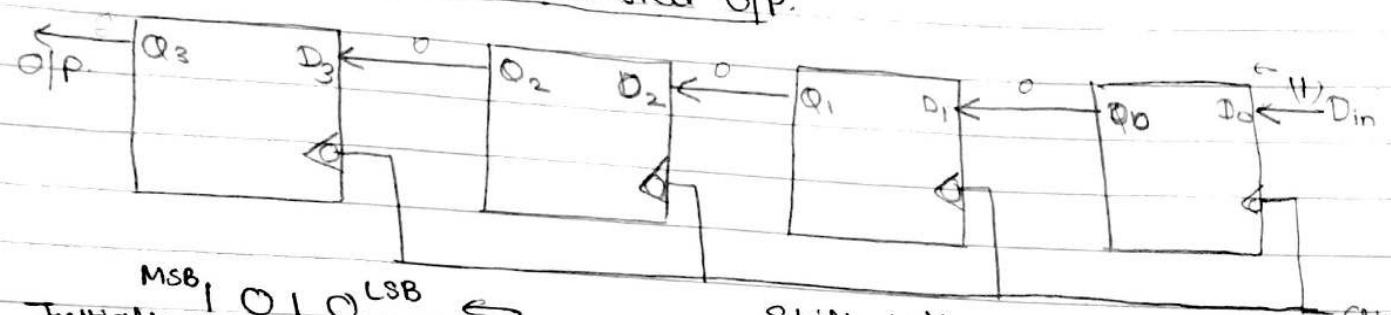
14.10.15



These are used to store more than 1-bit of data using more than 1 flip flop.



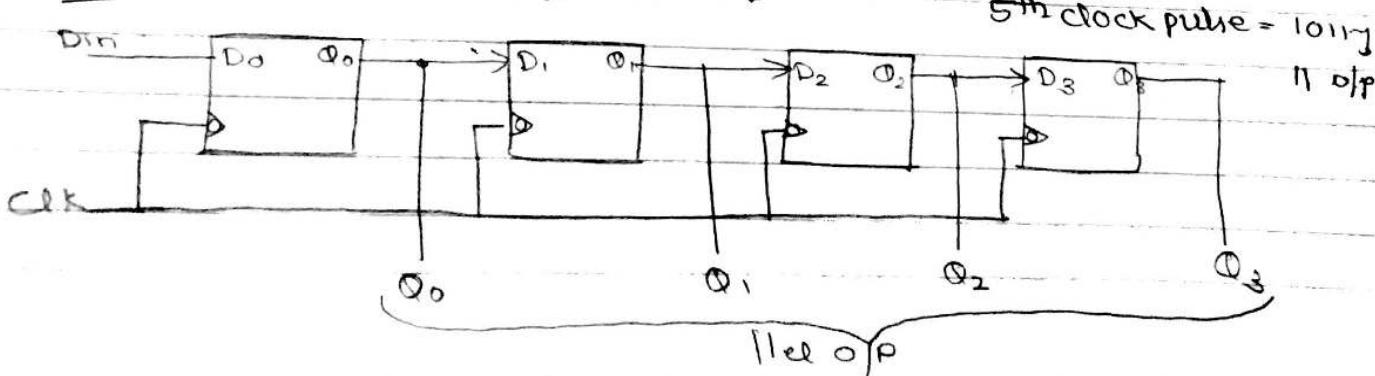
1) SISO \Rightarrow Serial I/P & serial O/P.

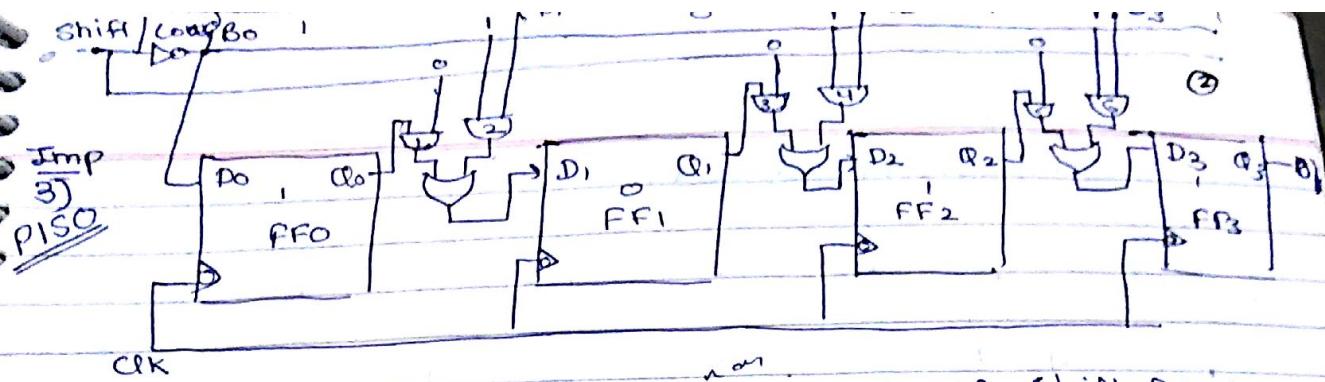


MSB
Initially 0000
1 → 0001
2 → 0010
3 → 0101
4 → 1010

Shift left
Shift right (LSB first enters)

2) SIPO \Rightarrow Serial I/P & parallel O/P





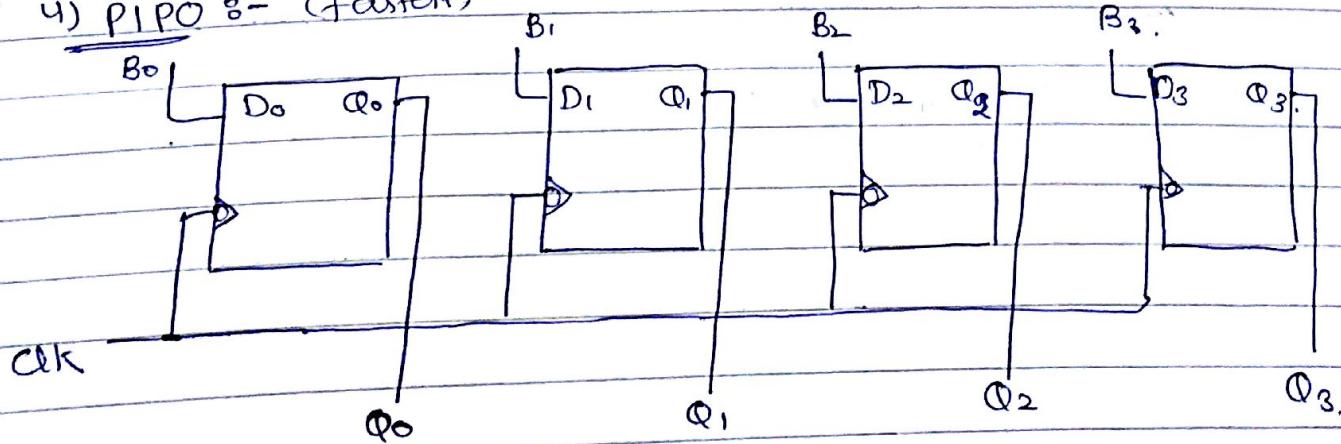
1) Load mode 1011 also known as Bidirectional Shift Register
 Shift $\rightarrow 0$
 Load $\rightarrow 1$

2) Shift mode:-

All of O/P of Ist flip flop will shift to II FF and III FF & so on--.

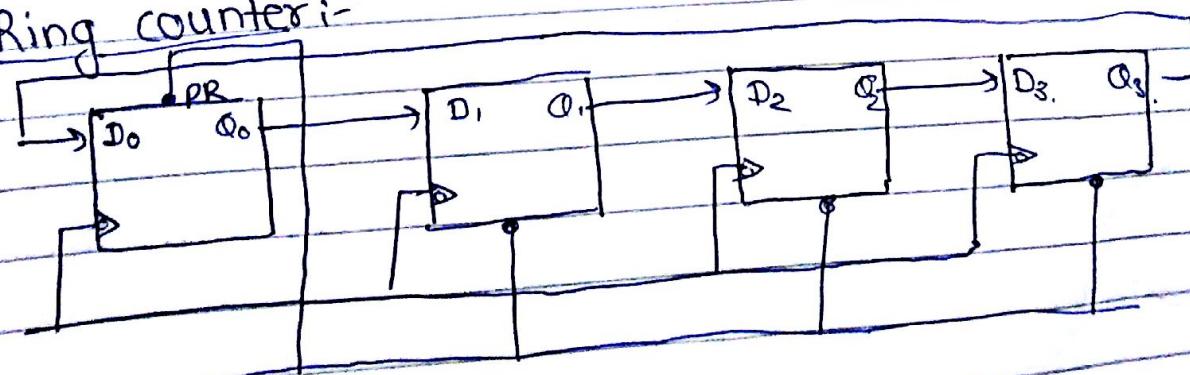
15/09/15

4) PISO 8- (fastest)



Applications of Shift Registers :-

1) Ring counter:-



Clear

Q₃ Q₂ Q₁ Q₀

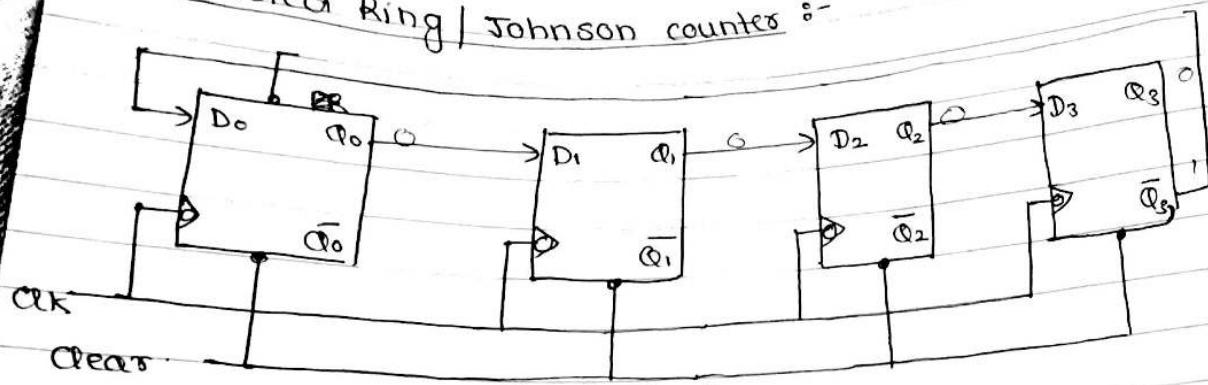
0 0 0 0

PRSet \rightarrow 0 0 0 1

3rd \rightarrow 1000
 4th \rightarrow 0001

3

a) Twisted Ring / Johnson counter :-



	Q ₃	Q ₂	Q ₁	Q ₀
1st →	0	0	0	0
2nd →	0	0	1	1
3rd →	0	1	1	1
4 th →	1	1	1	1
5 th →	1	1	1	0
6 th →	1	1	0	0
7 th →	1	0	0	0
8 th →	0	0	0	0

Design of clocked sequential circuit

Sequential circuits

↓
Asynchronous sequential circuit.

↓
Synchronous sequential circuit.

- These circuits are difficult to design.
- These circuits are easy to design.
- An unclocked flip-flop or latch is used as a memory element.
- A clocked flip-flop is used as a memory element.
- The status of the memory element will change any time as soon as the i/p is changed.
- The status of the memory element is affected only at the active edge of the clock. if the i/p is changed.

Synchronous Sequential circuits:-

Models / Machines used to design synchronous seq. ckt

Moore model

Mealy model

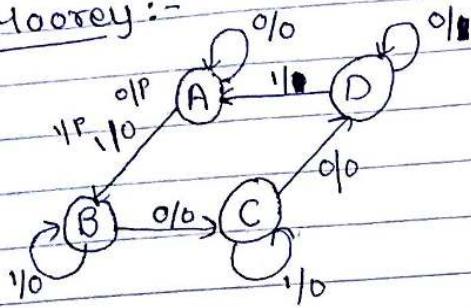
- The final o/p depends only on the present state of memory element.
- The o/p changes only after the active clock edge.
- The implementation of a logic f/n requires more no. of states than the mealy ckt.
- The final o/p depends on the present state of memory element & the external i/p.
- o/p can change in b/w the clock edges if the external i/p changes.
- Implementation of the same logic f/n requires less no. of states than moore ckt.

→ It is less flexible to design.

→ more flexible to design.

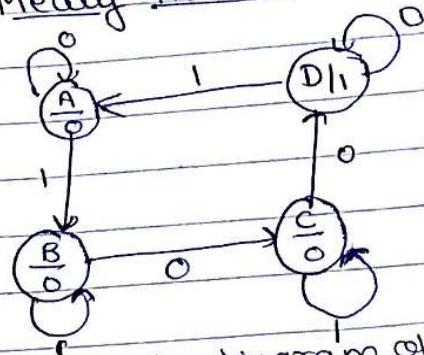
A B C D
1 0 0 1

Moore :-



state diagram of moorey model

Mealy model.

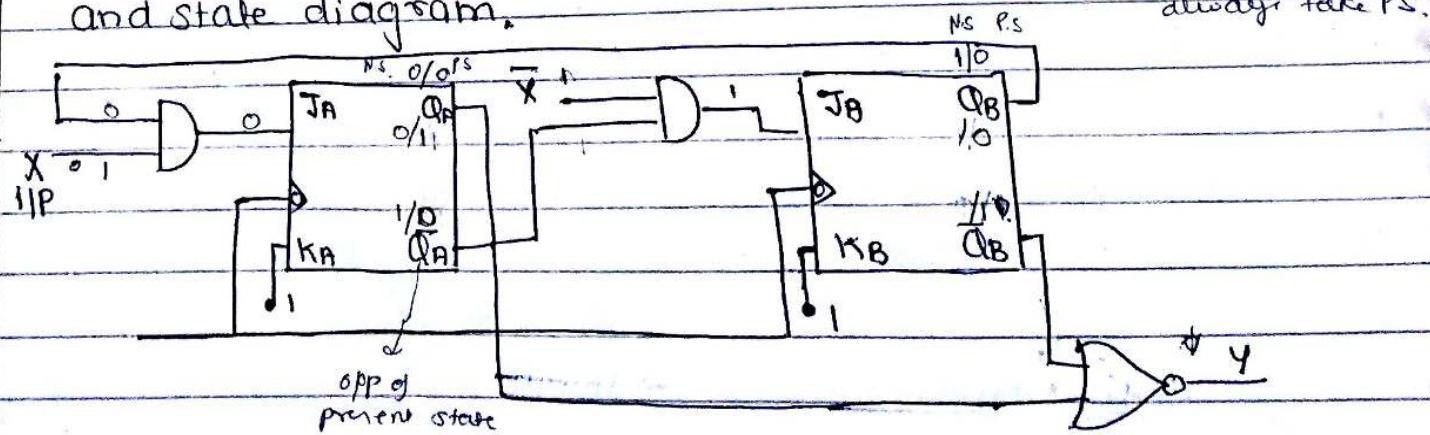


State diagram of mealy mode

STEPS TO DESIGN SYNCHRONOUS SEQUENTIAL CIRCUITS.

- logic diagram
- State table
- State Reduction
- State Assignment
- State diagram.

Q) Identify the type of the circuit given. Draw its state table and state diagram.

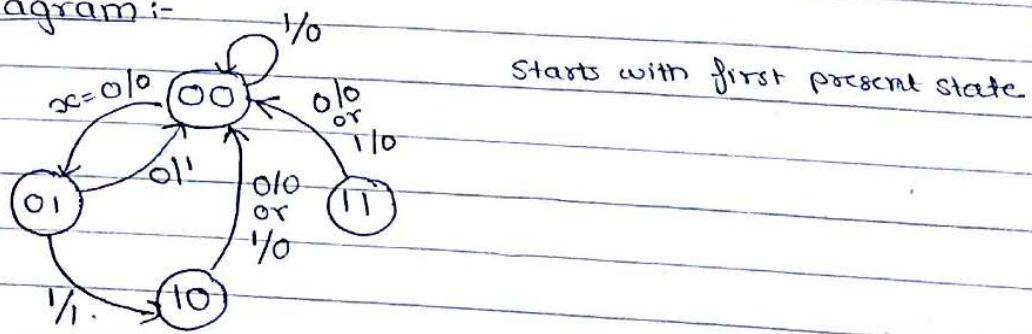


o/p depends only on the present state. ∴ moorey

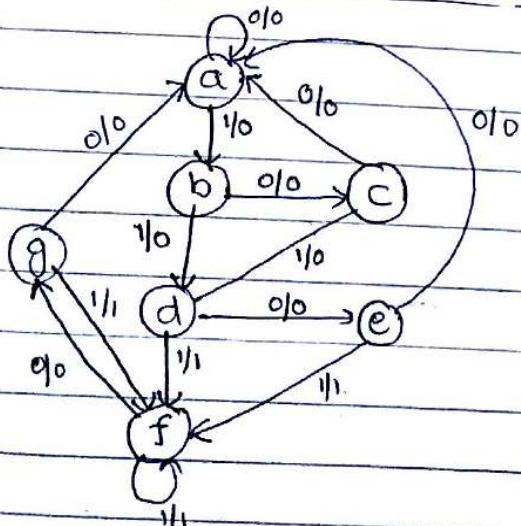
State table :-

P.S		N.S		O/P
A	B	X=0	X=1	(4)
0	0	01	00	0
0	1	00	10	1
1	0	00	00	0
1	1	00	00	0

State diagram :-



State reduction & assignment :-



State table

P.S	N.S		O/P (4)	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

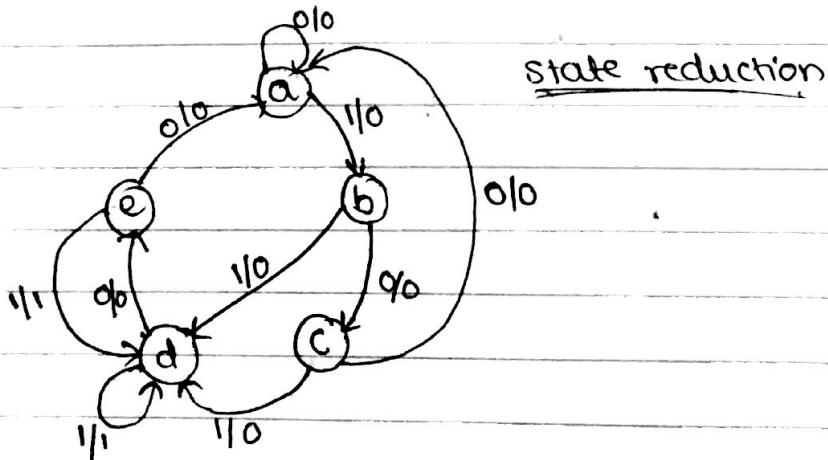
- find two states whose N.S and O/P are same
- Replace all g's by e's.

P.S	N.S $x=0$ $x=1$		O/P (4) $x=0$ $x=1$	
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
$\sim d$	e	f	0	1
e	a	f	0	1
$\checkmark f$	e	f	0	1

Replace $\checkmark f$ by d.

State reduction diagram

P.S	N. S $x=0$ $x=1$		O/P (4) $x=0$ $x=1$	
a	a	b	0	0
b	c	d	0	0
$\checkmark c$	a	d	0	0
d	e	d	0	1
$\checkmark e$	a	d	0	1



State Assignment

Assigning the n-bit no. to each present state.

as a - 000

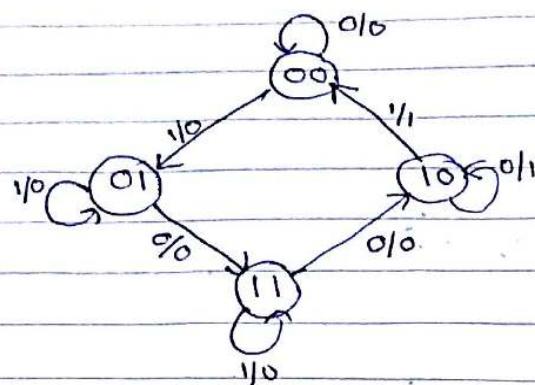
b - 001

c - 010

d - 011

13110115

(Q) For a given state diagram draw the clk & synchronous sequential ckt using any type of flip-flop.

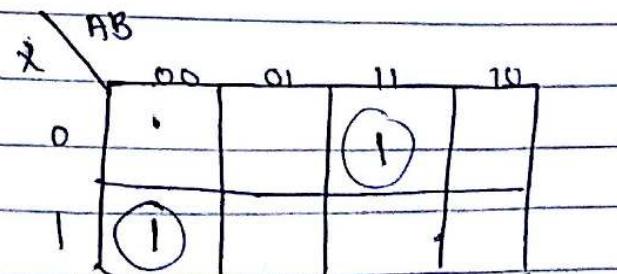
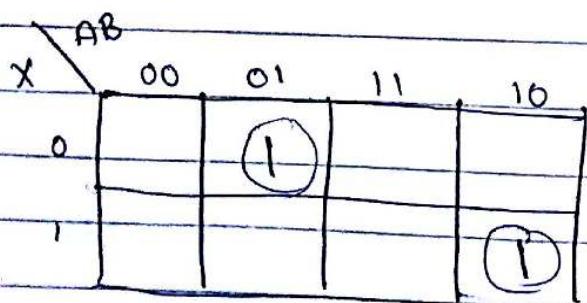


PS	N.S		O/P (4)	
	$x=0$	$x=1$	$x=0$	$x=1$
00	00	01	0	0
01	11	01	0	0
10	10	00	1	1
11	10	11	0	0

Ckt excitation table (expanded state table) :-

P.S	I/P	N.S	ffused (2)		O/P	4	
A	B	(x)	Anti	Bn't	T _A	T _B	
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	0	0
0	1	1	0	1	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	0	1	0	1
1	1	0	1	0	0	1	0
1	1	1	1	1	0	0	0

K-Map



$$T_A = \bar{A}B\bar{x} + A\bar{B}x$$

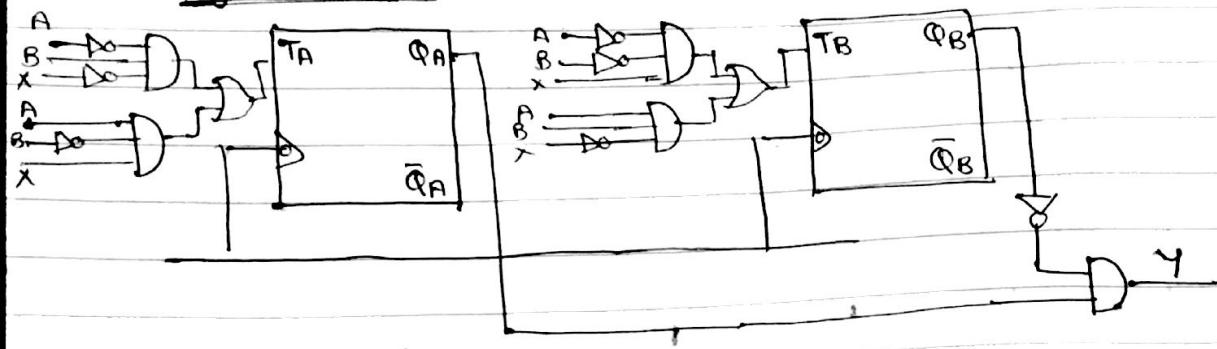
(only AND logic)

$$T_B = \bar{A}\bar{B}x + AB\bar{x}$$

$y =$	$x \setminus AB$	00	01	11	10
0					1
1				1	0

$y = A\bar{B}$

logic diagram



1911015 FINITE STATE MACHINE :- (FSM)

→ Minimization of completely specified sequential ckts :-
 ↳ I/P, O/P, P.S., N.S → given.

- 1) Implication table or graph
- 2) Compatible pairs.
- 3) Merger diagram.
- 4) Maximal compatible pairs / minimal cover table
- 5) Reduced state table

P.S	N.S		O/P	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a	b	0	0
b	e	a	0	0
c	g	f	0	1
d	a	d	1	0
e	a	d	1	0
f	c	b	0	0
g	a	c	1	0

Q) Reduce the given table using implication chart, find merger diagram & min cover table.

→ Implication table : Rules :-

- 1) leave first ^{ps} state for vertical side and leave last present state for horizontal side.
- 2) first check the O/P, if it is not equal then put 'X' mark and no need to check the next state.
- 3) If the O/P is equal then check the next state. if the next state is also equal then put 'V' mark and if N.S. is not equal then write down that state in the box.

	b (d,e) (b,a)	c	d	e	f	g	
b	X	X					✓
c							X X
d	X	X	X				X X X
e	X	X	X				V
f	X (d,c) (a,b)	X	X	X			X V X X X
g	X	X	X	(d,e)	(d,e)	X	X X X V V X
a b c d e f							

→

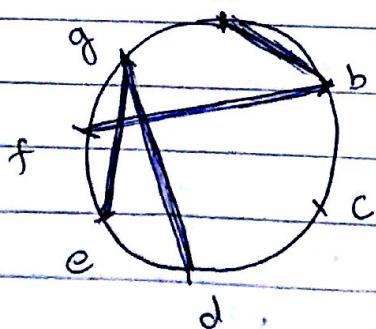
b	c	d	e	f	g
a	X	X	X	X	X
b		X	X		
c					
d			X		
e				X	
f					X
g					

Implication table/graph
I.T

Compatible pairs :-

→ (a,b) (b,f) (e,d) (d,g) (e,g)

Merger diagram



first priority is given to Rhombus
 Second " " " triangle
 third " " " Straight line

(c,d,g) (a,b) (b,f) → maximal compatible pairs

Reduced state table:-

$$e = d = g$$

$$a = b$$

$$b = f$$

P.S.	N.S.		O/P	
	$X=0$	$X=1$	$X=0$	$X=1$
a	d	b	0	0
c	g	f	0	1
e	a	d	1	0

→ Minimization of incompletely specified sequential ckt :- N.S, O/P
 ↳ P.S, I/P given to minimize

P.S.	00	01	11	10 ← S/P.
a	c,-	a,0	b,-	-,-
b	-,-	a,-	b,1	e,-
c	c,0	a,-	-,-	d,-
d	c,-	a,-	b,-	d,0
e	f,-	-,-	b,-	e,1
f	f,1	a,-	-,-	e,-

b	✓				
c	✓	(e,d)			
d	✓	(e,d)	✓		
e	(c,f)	✓	(c,f)	X	
f	(c,f)	✓	X	(c,f) (d,e)	✓
a					
b					
c					
d					
e					

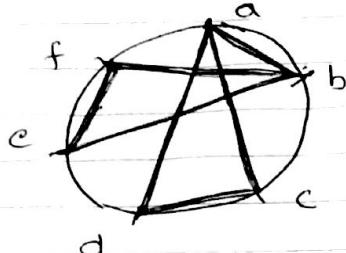
→

b	✓				
c	✓		X		
d	✓		X	✓	
e	X		X	X	
f	X		X	X	✓
a					
b					
c					
d					
e					

Compatible pairs:-

(a,b), (a,c), (a,d), (b,e), (b,f), (c,f), (c,d)

Merge diagram



(a,d,c)

(b,e,f), (a,b) + (a,c), (a,d) \rightarrow maximal compatible pairs.

$$\phi = \phi = f$$

~~$$a = b$$~~
$$a = d = e$$

~~$$c = e$$~~
$$d = \phi$$

~~$$e = f$$~~

Reduced state table

P.S	00	01	11	10
f	f, l	a, -	-, -	e, -

26/10/15 Partition Technique:-

P.S	N.S.z	
	$x=0$	$x=1$
A	C, 0	F, 0
B	D, 1	F, 0
C	E, 0	B, 0
D	B, 1	E, 0
E	D, 0	B, 0
F	D, 1	B, 0

$$P_1 = (A, C, E) \xleftarrow{X=0} (O) O/P$$

$$= (B, D, F) \xleftarrow{X=1} (I) O/P$$

$$P_2 = (A, C, E)$$

P.S. N.S.

$$A \rightarrow C$$

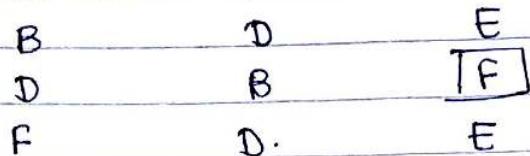
$$C \rightarrow E$$

$$E \rightarrow D$$

$$(C, E)(D) \rightarrow (B)(D)(D)$$

$$P_3 = (B, D, F) \quad (X=1)$$

P.S. N.S.

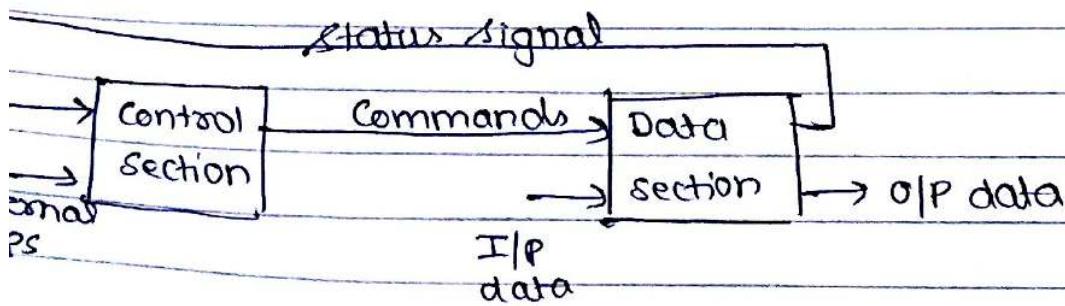


$$(F)(D)(B)$$

ALGORITHM STATE MACHINES (ASM)

4th unit

Any digital system design can be divided into two distinct parts. the part that deals with data processing operation is known as data section and the other that deals with control signal generation and supervision is known as control section.



i.e. the data section manipulates the data in the register