

Module-5 - (Part-1)

Sequential Circuit Design

- * In synchronous or clocked sequential networks, clocked flip-flops are used as memory elements which change their individual states in synchronism with periodic clock signal.
- * : The change in states of flip-flops and change in state of entire circuit occurs at the transition of clock signal.
- * The synchronous or clocked sequential networks are represented by two models :
 1. Mealy Model :- The output depends only on the present state of the flip-flops.
 2. Moore Model :- The output depends on both present state of flip-flops and on the inputs.

\Rightarrow Moore Model:

* When the output of sequential network depends only on the present state of flip-flop, the sequential network is referred to as Moore model.

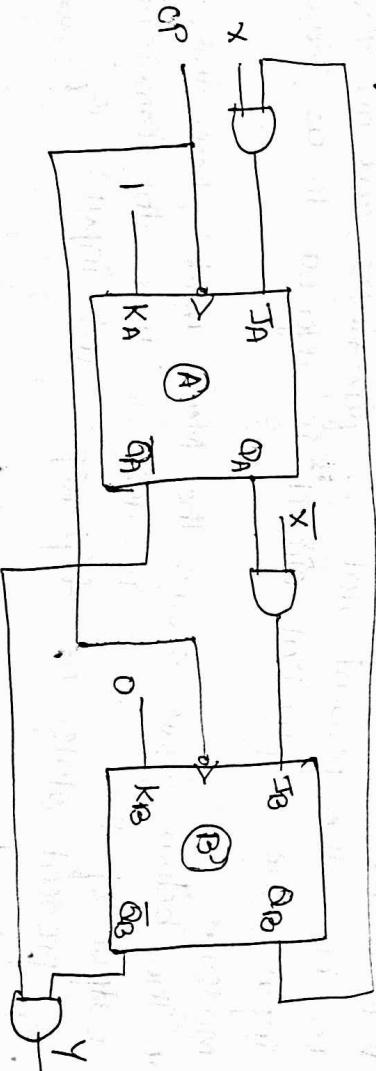


fig:- Example of Moore model

* The above figure shows a sequential network which consists of two JK flip-flops and AND gate. The network has one input X and one output Y .

* In the figure, input X is used to determine the inputs of flip-flops. It is not used to determine the output. The output is derived using only present states of the flip-flops or combination of it.

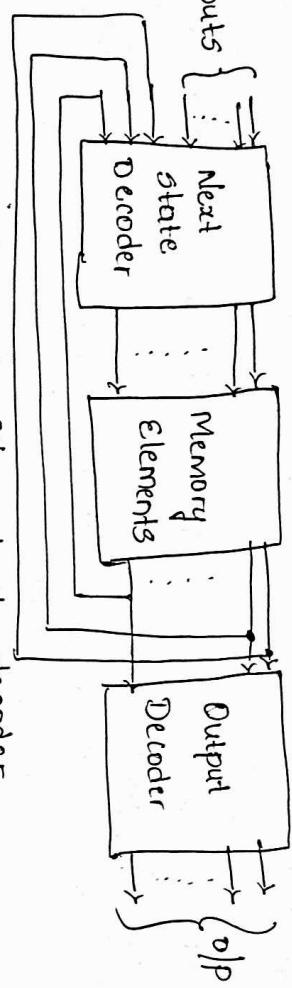
$$Y = Q_A \cdot Q_B$$


Fig:- Moore circuit model with output decoder

- * In the Moore model, as output depends only on present state of flip-flops, it appears only after the clock pulse is applied.

\Rightarrow Mealy model

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

The below figure shows the Mealy model, here the output of the circuit is derived from the combination of present state of flip-flops and inputs of the circuit.

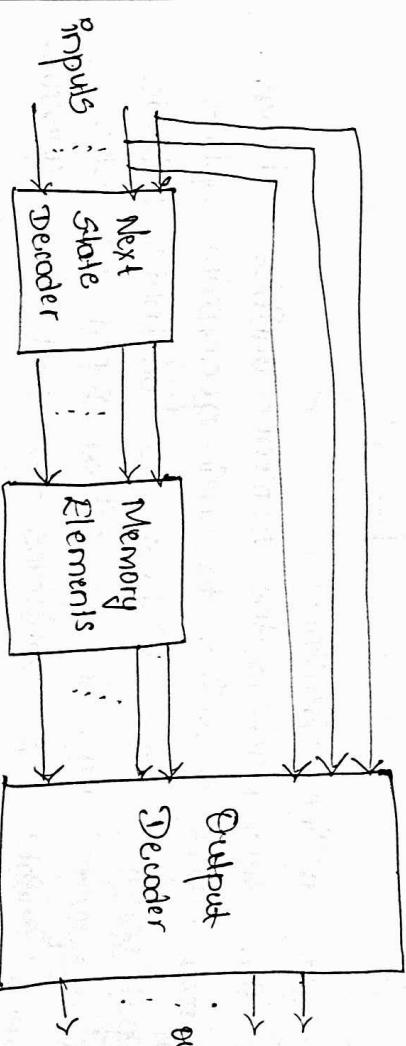
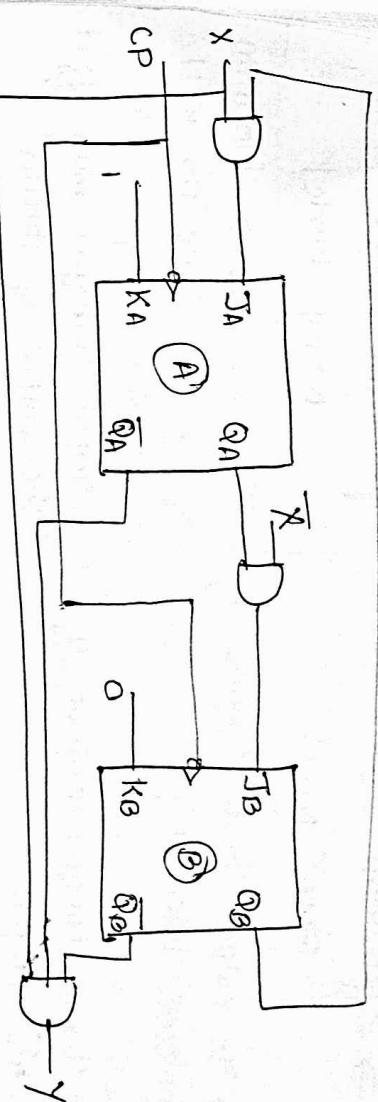


Fig:- Mealy circuit model

- Fig:- Example of Mealy model
- * Here we can observe that changes in the input within the clock pulses cannot affect the state of flip-flop. However they can affect the output of the circuit.
 - * Due to this, if the input variations are not synchronized with the clock, the derived output will also not be synchronized with the clock and we get false output.
 - * The false outputs can be eliminated by allowing input to change only at the active transition of the clock (ex.: high to low).



\Rightarrow Differentiate Moore vs Mealy Circuit models.

Moore Model	Mealy Model
<ul style="list-style-type: none"> 1) Its output is a function of present state only. 2) Input changes does not affect the output. 3) Moore model requires more number of states for implementing same function. 4) There is less hardware requirement for circuit implementation. 5) Synchronous output and state generation. 6) Output is placed on states. 7) Easy to design. 8) They react slower to inputs. <p>\Rightarrow State machine Notation:</p> <ul style="list-style-type: none"> * In the state machine, the Boolean variables have different names according to their generation place. <ol style="list-style-type: none"> 1. <u>Input variable</u>: All variables that originate outside the sequential machine are called input variables. 2. <u>Output variable</u>: All variables that exit the sequential machine are called output variables. 	<ul style="list-style-type: none"> 1) Its output is a function of present state as well as present input. 2) Input changes may affect the output of circuit. 3) It requires less number of states for implementing same function. 4) There is more hardware requirement for circuit implementation. 5) Asynchronous output generation. 6) Output is placed on transitions. 7) Difficult to design. 8) They react faster to inputs.

3. State Variable: The output of flip-flops defines the state of a sequential machine. Therefore, the state variables are flip-flop outputs.

4. Excitation variable: Excitation variables are generated by the input combinational logic operating on the state variables and input variables.

\Rightarrow State and State Variable:-

* The state is defined by the output of flip-flops. In the state machine, state variables and states are related by the expression

$$2^x = Y$$

where,

x = number of state variables (e.g. flip-flops)
 y = max. number of possible states.

\Rightarrow Present State and Next State

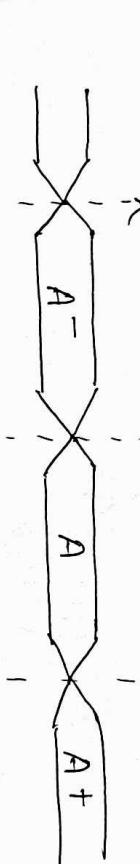
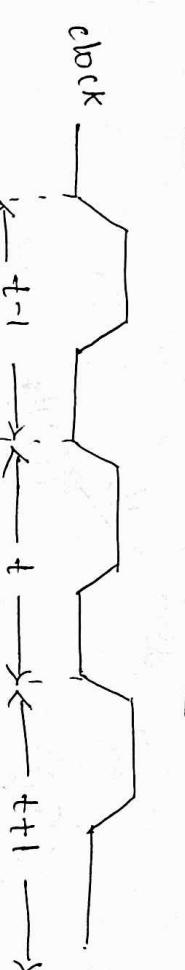


Fig: Illustrating present state and next state

- * In state machines, it is necessary to distinguish state variables before and after the clock pulse.
- * State variable A can be represented as A before

The arrival of clock and as At after the arrival of synchronizing clock pulse.

Present State :- The status of all state variables, at some time t , before the next clock edge represents condition called present state.

Next State :- The status of all state Variables, at some time, $t+1$, represents a condition called next state.

State Diagram :-

* State Diagram is a pictorial representation of a behavior of a sequential circuit.

* The state is represented by the circle and the transition between states is indicated by directed lines connecting the circles.

* A directed line connecting a circle with itself indicates the next state is same as present state.

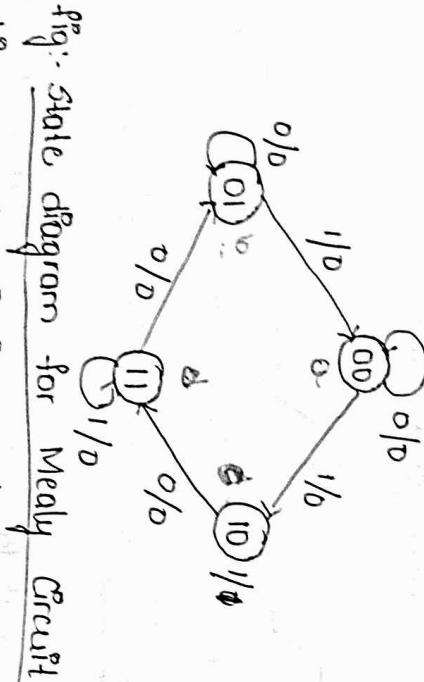


Fig: State diagram for Mealy Circuit

* The binary number inside each circle identifies the state represented by the circle.

* The directed lines are labeled with two binary numbers separated by a symbol ':'.
The output value that causes the state transition is labeled first and output value during the present state is labeled after the symbol ':'.

* The input value that causes the state transition is labeled first and output value during the present state is labeled after the symbol ':'.

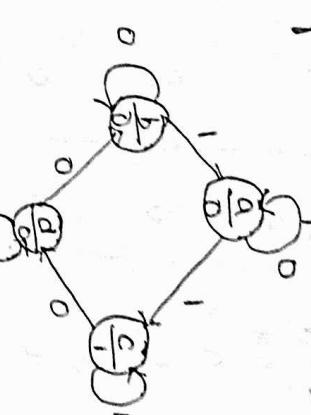


Fig: State diagram for Moore Circuit

* In case of Moore circuit, the directed lines are labeled with only one binary number representing the state of I/P that causes state transition.

* The output state is indicated within the circle below the present state :: o/p state depends only on present state not on input.

State Table for Mealy Circuit

Present State	Next State	Output
---------------	------------	--------

AB	x=0	x=1	x=0	x=1
a	AB	AB	y	y
b	a	c	0	0
c	b	a	0	0
d	c	c	0	0

* State table represents -the relationship b/w I/P, o/P and flip-flop states. Both next state and o/P sections have two columns representing two possible I/P conditions $x=0$ and $x=1$.

→ State Table for Moore Circuit

Present state

Next state

Output

	$x=0$	$x=1$	y
AB	AB	AB	
a	a	c	0
b	b	a	0
c	d	c	1
d	b	d	0

- * In Moore circuit, the O/P section has only one column since O/P does not depend on input.

⇒ Transition table :-

- * The state diagram and state table represent state using symbols or names.
- * In transition table specific state variable values are assigned to each state. Assignment of values to state variables is called state assignment.
- * Transition table also represents relationship between input, output and flip-flop states.

Present state

Next state

Output

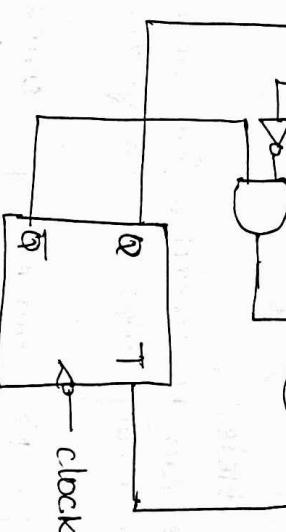
	$x=0$	$x=1$	$x=0$	$x=1$
AB	AB	AB	Y	Y
00	10	00	10	0
01	00	00	0	0
10	01	1	0	0
11	00	10	1	0
10	10	1	0	0
00	10	1	0	0

⇒ Synchronous Sequential Circuit Analysis

5

- * The behaviour of a sequential network is determined from the inputs, the outputs and states of its flip-flops.
- * Both outputs and next state are function of inputs and present state.

- * The analysis of sequential circuit consists of obtaining a table or a diagram for time sequence of inputs, outputs and internal states.



Consider the sequential circuit to be analysed as shown in fig above.

→ Steps to analyze given synchronous sequential circuit.

1. Determine the flip flop input equations and output equations from sequential circuit.

$$x = xQ$$

$$\bar{T} = xQ + \bar{x}\bar{Q}$$

2. Derive the transition equation. The transition equation for T flip-flop is

$$Q^+ = T \oplus Q$$

$$Q^+ = (xQ + \bar{x}\bar{Q}) \oplus Q$$

3. Plot the next step map for each flip-flop

$$\text{for } Q^+ \quad Q^+ = (\alpha Q + \bar{\alpha} \bar{Q}) \oplus Q$$

4. Plot the transition table.

Present state	Next state	Output
$\alpha=0$	$\alpha=1$	$\alpha=0$
Q	Q^+	Z
0	0	0
1	0	0
		1
		0

5. Draw the state table.

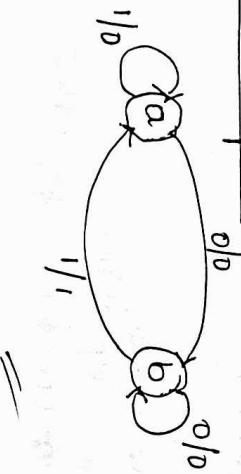
* The transition table can be converted into state table.
Here new symbols to binary codes are assigned.

They are $a=0$, $b=1$.

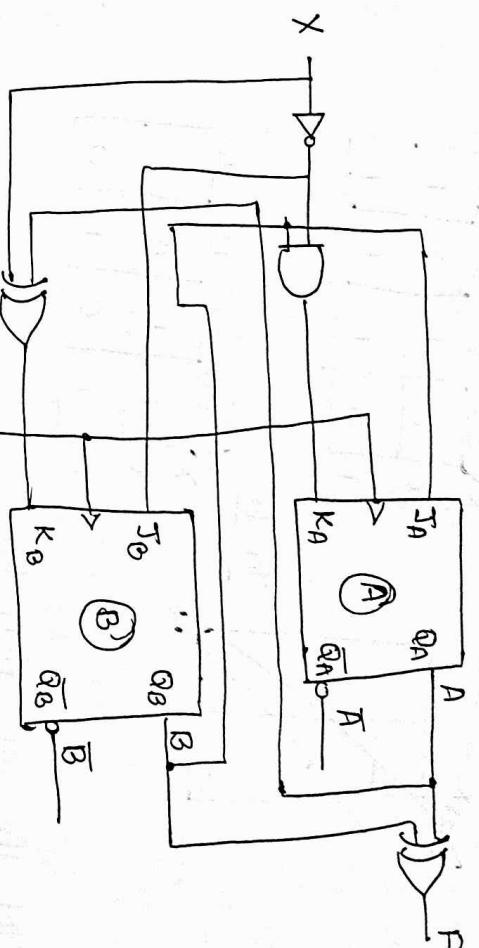
Present state	Next state	Output
$\alpha=0$	$\alpha=1$	$\alpha=0$
a	b	0
b	a	0
		1

b(i)

6. Draw state diagram



Problem-1:
Construct the transition table, state table and state diagram for the Moore sequential circuit given below:



Soln
1. Determine the flip-flop input equations and the output equations from the sequential circuit.

$$F = A \oplus B$$

$$J_A = B$$

$$K_A = \bar{X}$$

$$J_B = \bar{X}$$

$$K_B = X \oplus A$$

2. Define the transition equations.

The transition equations for JK flip-flops can be derived from the characteristic equation of JK flip-flop as follows.

w.k.t. for JK flip-flop

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

$$\therefore A^+ = Q_A^+ = J_A \bar{Q}_A + \bar{K}_A Q_A$$

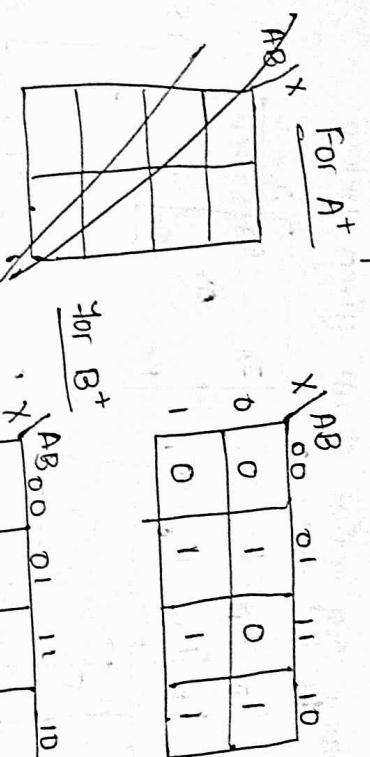
$$= B \bar{Q}_A + \bar{X}\bar{B}Q_A$$

$$= B \bar{Q}_A + (X+\bar{B})Q_A$$

$$= B \bar{A} + (X+\bar{B})A$$

$$= \bar{X}\bar{B} + (X+\bar{X}\bar{A})B$$

3. Plot a next-state maps for each flip-flop.
The next-state maps are:



4. Plot the transition table.

Present state | Next State | Output

A	B	$x=0$	$x=1$	$F = A \oplus B$
		A^+	B^+	
0	0	0	1	0
0	1	1	1	1
1	0	1	1	0
1	1	0	0	1

Transition table

Present state		Next state		Output
$x=0$	$x=1$	A^+	B^+	
0	0	0	1	0
0	1	1	1	1
1	0	1	1	0
1	1	0	0	1

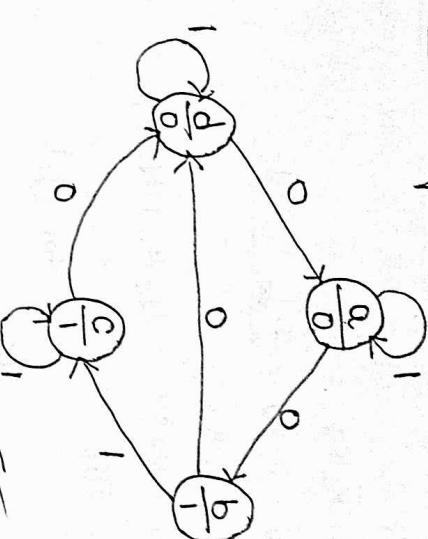
5. Draw the state table:-

By assigning $a=00$, $b=01$, $c=10$ and $d=11$, state table can be drawn.

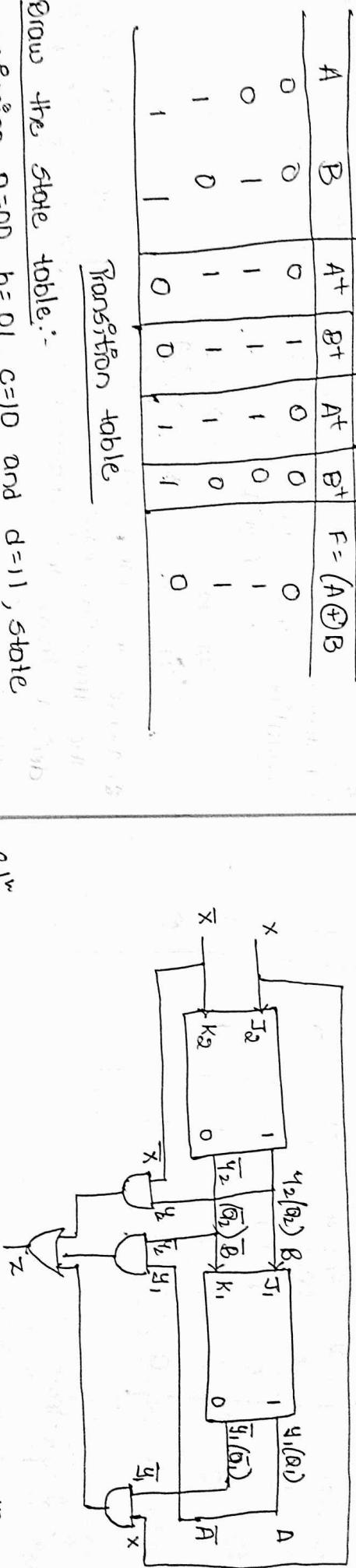
Present state | Next state | Output

Soln:
The flip-flop input equations and output equations are:
 $J_2 = x$ $J_1 = y_2$ $x = \bar{x}y_2 + \bar{y}_2y_1 + \bar{y}_1x$
 $K_2 = \bar{x}$ $J_0 = \bar{y}_0$
 For JK flip-flop

6. Draw state diagram



- Problem 2 :- Analyze the synchronous circuit
 1. Write down the excitation and output functions.
 2. Form the excitation and state tables.
 3. Give a word description of the circuit operation.



Present state		Next state		Output
$x=0$	$x=1$	A^+	B^+	
a	b	b	a	0
b	c	c	c	1
c	d	d	d	1
d	a	a	d	0

For JK flip-flop.

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

$$J_2^+ = Q_2^+ = J_2\bar{Q}_2 + \bar{K}_2Q_2$$

$$= x\bar{Q}_2 + xQ_2 = \underline{x\bar{Q}_2} + \underline{xQ_2}$$

$$J_1^+ = Q_1^+ = J_1\bar{Q}_1 + \bar{K}_1Q_1$$

$$= \bar{Q}_2\bar{Q}_1 + Q_2Q_1 = \bar{Q}_2\bar{Q}_1 + \bar{Q}_2Q_1$$

State maps for each flip-flop:

for \bar{Q}_2^+

x	00	01	11	10
0	0	0	0	0
1	1	1	1	1

for \bar{Q}_1^+

x	00	01	11	10
0	0	0	1	1
1	0	1	1	1

Present state

Next state

Output x

	$x=0$	$x=1$	$x=0$	$x=1$
y_2^+	0	0	0	1
y_1^+	0	0	1	0
y_0^+	0	0	0	1
y^+	0	1	1	0

reached while down counting.

From the problem description w.r.t.

- 10 states are needed.
- Sequential circuit has one input M and two outputs Y and Z.

3. when $M=0$: count up and $M=1$: count down.

To draw state diagram:

Assign - 2 problems

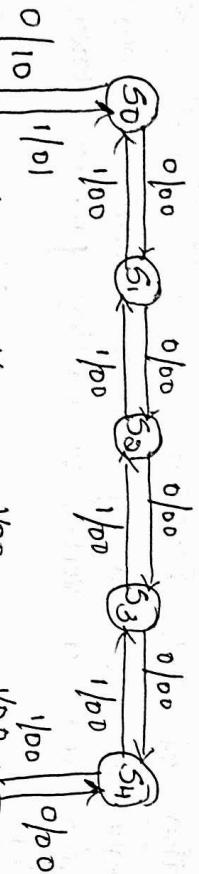
Ex - S.3 and S.4.

1. Up-Down Decade Counter :

In a sequential circuit design, the construction of the state diagram from a verbal or written problem statement is very tedious.

- Ex-1 :- Construct a state diagram for a synchronous decade up/down counter. The mode control input M decides the counting operation. When $M=0$, the counter is to count up; when $M=1$, the counter is to count down. The count is to repeat or cycle after the terminal count i.e. it should go from 9 to 0 while up counting and from 0 to 9 while down counting. The counter output Y should go high if terminal count is reached while up counting and counter Z should go high if terminal count is reached while down counting.

Step-1: The initial state is S_0 . From the table, 0th LSB of excess-3 bit is 1, the BCD bit is 0. If excess-3 bit is 0, then BCD bit is a 1.



The state diagram is a Mealy machine because the two output variables are dependent on both mode input and present state.

Example 2 : Serial Excess 3 to BCD code converter

Design the Mealy state diagram for the sequential ckt that will convert serial excess-3 code into serial binary coded decimal (BCD). Since both excess 3 and BCD are four bit codes the machine is to return to the beginning after four inputs. Output Z should be high for any non-excess-3 input.

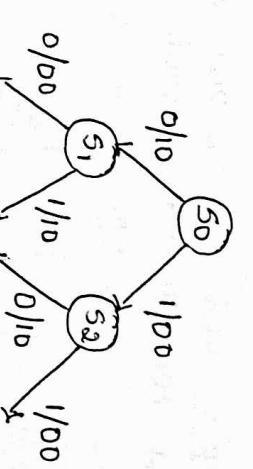
Sol^v Assume first serial input is LSB of excess 3 code. Excess 3 input is labelled E and BCD output B .

E_3	E_2	E_1	E_0	B_3	B_2	B_1	B_0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	1	0
0	1	0	1	0	0	1	0
0	1	1	1	0	1	0	1
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	1
1	0	1	1	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	1	0	0
1	1	1	1	1	1	1	1

Truth Table

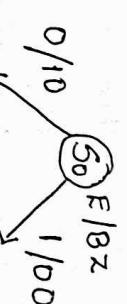
Step-3 : A new state is needed for each transition path.

S/P sequence			O/P
$E_0=0$	$E_1=0$	$E_2=0$	$B_3=1 \quad Z=0$
$E_0=0$	$E_1=0$	$E_2=1$	$B_3=0 \quad Z=0$
$E_0=0$	$E_1=1$	$E_2=0$	$B_3=1 \quad Z=0$
$E_0=0$	$E_1=1$	$E_2=1$	$B_3=0 \quad Z=0$
$E_0=1$	$E_1=0$	$E_2=0$	$B_3=1 \quad Z=0$
$E_0=1$	$E_1=0$	$E_2=1$	$B_3=0 \quad Z=0$
$E_0=1$	$E_1=1$	$E_2=0$	$B_3=1 \quad Z=0$
$E_0=1$	$E_1=1$	$E_2=1$	$B_3=0 \quad Z=0$

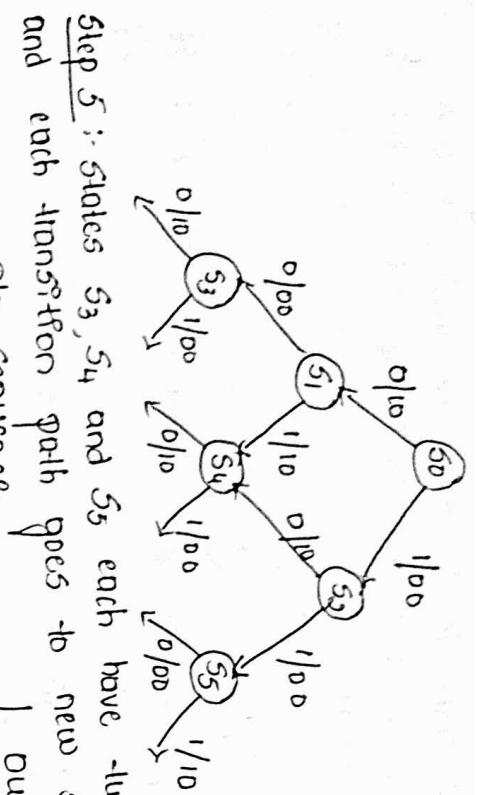


Step-4 : If $E=0$, the system goes to state S_2 .

Step-4 : If $E=1$, the system goes to state S_1 .



Step-4 : Observe S_4 and S_5 for their inputs and outputs. We can see that outputs for S_4 and S_5 are identical for input equal to 0 as well as for input equal to 1.

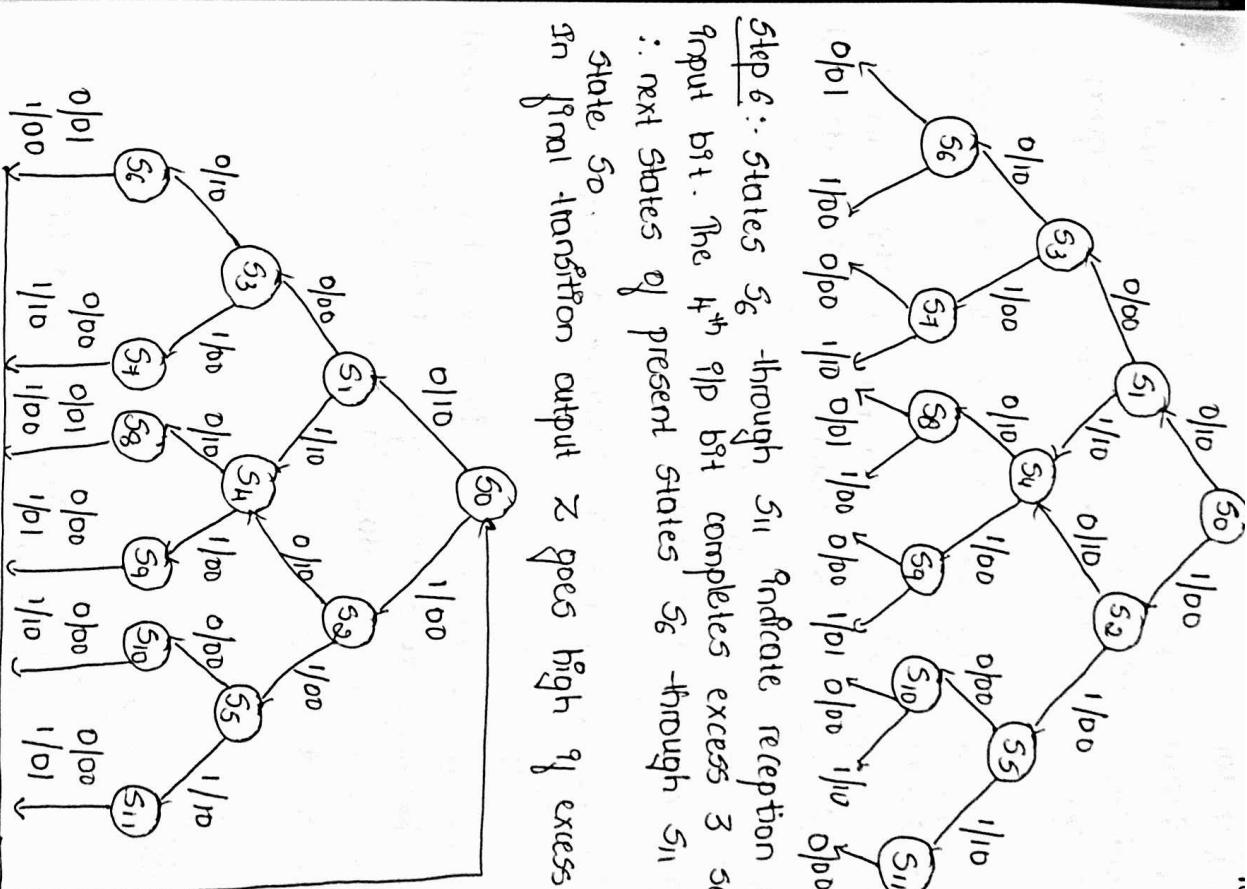


Step 5: States S_3 , S_4 and S_5 each have two transition path and each transition path goes to new state.

S/p sequence

Output

	$E_1 = 0$	$E_2 = 0$	$E_3 = 0$	$B_3 = 0$	$\chi = 1$
$E_0 = 0$	$E_1 = 0$	$E_2 = 0$	$E_3 = 1$	0	0
$E_0 = 0$	$E_1 = 0$	$E_2 = 1$	$E_3 = 0$	0	0
$E_0 = 0$	$E_1 = 0$	$E_2 = 1$	$E_3 = 1$	0	0
$E_0 = 0$	$E_1 = 1$	$E_2 = 0$	$E_3 = 0$	0	0
$E_0 = 0$	$E_1 = 1$	$E_2 = 1$	$E_3 = 1$	0	0
$E_0 = 0$	$E_1 = 1$	$E_2 = 0$	$E_3 = 0$	0	0
$E_0 = 0$	$E_1 = 0$	$E_2 = 1$	$E_3 = 0$	0	0
$E_0 = 0$	$E_1 = 0$	$E_2 = 0$	$E_3 = 1$	0	0
$E_0 = 1$	$E_1 = 0$	$E_2 = 0$	$E_3 = 0$	0	0
$E_0 = 1$	$E_1 = 0$	$E_2 = 0$	$E_3 = 1$	0	0
$E_0 = 1$	$E_1 = 0$	$E_2 = 1$	$E_3 = 0$	0	0
$E_0 = 1$	$E_1 = 1$	$E_2 = 0$	$E_3 = 0$	0	0
$E_0 = 1$	$E_1 = 1$	$E_2 = 1$	$E_3 = 0$	0	0
$E_0 = 1$	$E_1 = 1$	$E_2 = 0$	$E_3 = 1$	0	0
$E_0 = 1$	$E_1 = 0$	$E_2 = 1$	$E_3 = 1$	0	0
$E_0 = 1$	$E_1 = 0$	$E_2 = 0$	$E_3 = 0$	0	0
$E_0 = 1$	$E_1 = 0$	$E_2 = 0$	$E_3 = 1$	0	0
$E_0 = 1$	$E_1 = 1$	$E_2 = 0$	$E_3 = 1$	0	0
$E_0 = 1$	$E_1 = 1$	$E_2 = 1$	$E_3 = 0$	0	0
$E_0 = 1$	$E_1 = 1$	$E_2 = 0$	$E_3 = 0$	0	0
$E_0 = 1$	$E_1 = 0$	$E_2 = 1$	$E_3 = 0$	0	0
$E_0 = 1$	$E_1 = 0$	$E_2 = 0$	$E_3 = 1$	0	0



Step 6: States S_6 through S_{11} indicate reception of 3rd BCD input bit. The 4th 9p bit completes excess 3 sequence. \therefore next states of present states S_6 through S_{11} return to state S_0 .

In final transition output χ goes high if excess 3 is incorrect.

Fig: State diagram serial Excess-3 to BCD converter.

\Rightarrow Sequence Detector;

The specified input sequence can be detected using a sequential machine called Sequence detector. In this circuit output goes high when prescribed input sequence occurs.

Example 1: Design a Mealy type sequence detector to detect a serial input sequence of 101.

Sol: In a Mealy type design, the number of states in the state diagram are equal to the number of bits in desired

Input Sequence. Assume initial state is '0'.

1. slot a:  \rightarrow incorrect first bit
 \downarrow \rightarrow first bit in sequence detected

State 'a' checks for 1. when input is 1, we have detected

first bit in the sequence \rightarrow hence '0' to next state. When input is 0, remain in state '0' because bit 0 is not first bit in the sequence.

3. State b :

```

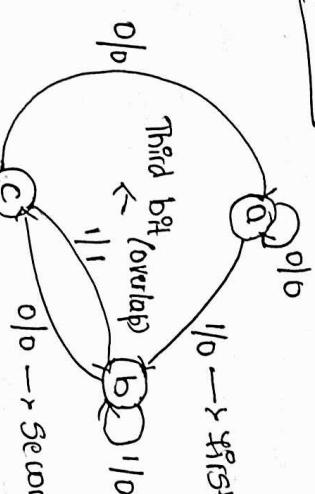
graph LR
    a((a)) -- "0/b" --> b((b))
    b -- "1/b" --> a
    a -- "incorrect second bit" --> b

```

When input is 1 → same state, bcz it is not second bit in sequence.
When input is 0 → second bit detected go to next

since there are 3 states, we need 2 flip-flops.
 Assign state $a = 00$, $b = 01$, $c = 10$. Draw excitation table.

Present State	Next State	Output	
	$x=0$	$x=1$	
a	a	b	0
b	c	b	0
c	a	b	0
		0	0
		0	1



bit detected

Present State		Next State		Output x	
A	B	A ⁺	B ⁺	x = 0	x = 1
0	0	0 0	0 1	0	0
0	1	1 0	0 1	0	0
1	0	0 0	0 1	0	1

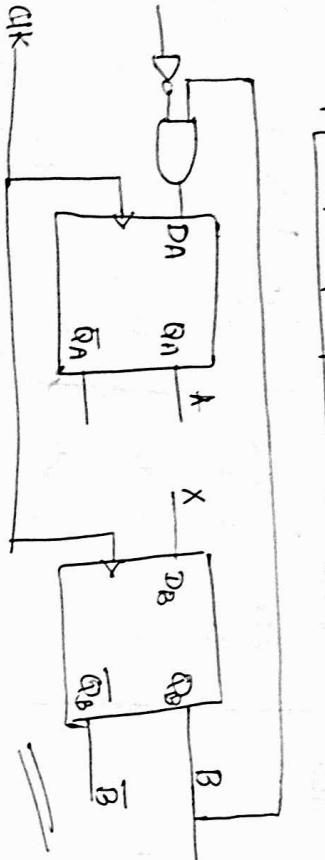
$$A^+ = B \overline{X}$$

<u>for</u>	<u>B^t</u>
X	AB
0	00
0	00
1	11
1	10

$$B^t = X$$

<u>for</u>	<u>X</u>
X	\overline{AB}
0	00
0	00
1	11
1	10

$$X = AX$$



Example 2: Design a Moore type sequence detector to detect a serial input sequence of 101.

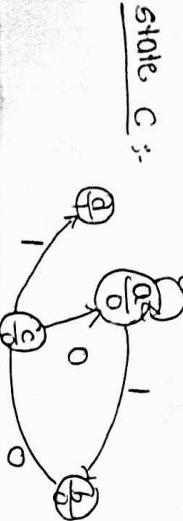
Sol:

State a:



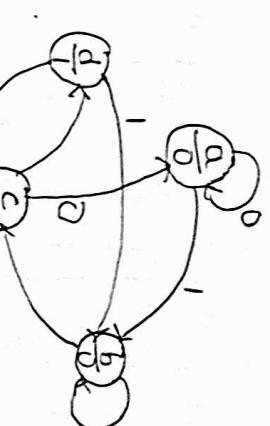
State b: detected a serial input sequence of 101.

State c:



Here, we cannot go back to state b since obj 'in state' b is zero and hence we have to create new state, d with output 1. In case, if input is zero, we have to restart checking of input sequence, and hence we have to return to state a.

State d:



Hence the sequence is detected, this is the last state. When input is 1, we have detected the first bit in next sequence, hence go to state b. When input is 0, we have detected second bit in overlapped sequence, hence we have to go to state c.

State table:

P.S	X=0	X=1	Output Z
a	a	b	0
b	c	b	0
c	a	d	0
d	c	b	1

Explanation table:

P.S	A	B	X=0	X=1	Output
0	0	0	0	1	0
0	0	1	-	0	0
1	1	0	0	1	0
1	0	0	1	1	1

Module -5 (part-2)

Memories

↳ Read / write Memories (RAM)

Logic Diagram

The diagram illustrates a D flip-flop circuit with a clock enable input. The circuit consists of two main components: a D_A flip-flop and a D_B flip-flop.

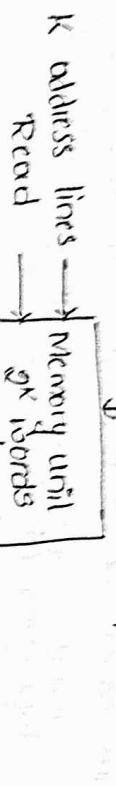
- D_A Flip-Flop:** This component has three inputs: D_A , \overline{Q}_A , and \overline{K} . Its output is Q_A .
- D_B Flip-Flop:** This component has three inputs: D_B , \overline{Q}_B , and \overline{B} . Its output is Q_B .

The connections between the components are as follows:

- The Q_A output of D_A is connected to the D_B input of D_B.
- The \overline{Q}_B output of D_B is connected to the \overline{K} input of D_A.
- The \overline{Q}_A output of D_A is connected to the \overline{B} input of D_B.
- The \overline{K} input of D_A is connected to the \overline{D}_A input of D_A.
- The \overline{B} input of D_B is connected to the \overline{D}_B input of D_B.
- The D_A input of D_A is connected to the Clk input through an inverter.
- The D_B input of D_B is connected to the Clk input through an inverter.
- The Clk input is also connected to the \overline{Q}_A output of D_A through an inverter.
- The Clk input is also connected to the \overline{Q}_B output of D_B through an inverter.

- A memory unit stores binary information in groups of bits called words. A word in memory is an entity of bits that move in and out of storage as a unit.
 - RAM is a volatile memory used to hold instructions and data of currently running programs. If the power is lost, the data is lost.
 - Communication between memory and its environment is achieved through data input and output lines, address selection lines and control lines that specify direction of transfer.
 - A block diagram of memory unit is shown in the figure below.

→ Read Operation



↓ n data input lines

K address lines → Memory unit
Read → 2^k words
n data output lines

Fig: Block diagram of a memory unit.

- * The n data input lines provide the information to be stored in memory and n data output lines supply the information coming out of memory.
- * The K address lines specify the particular word chosen among many available.
- * The two control inputs specify the direction of transfer desired:-
- ↳ Write input causes binary data to be transferred into the memory.
- ↳ Read input causes binary data to be transferred out of memory.

→ Write Operation

The write signal specifies a transfer data into the memory. The steps to transfer are as follows:-

1. Apply binary address of desired word to the address lines.

2. Apply the data bits that must be stored in memory to data input lines.

3. Activate the write input.

The memory will then take bits from input data lines and store them in the word specified by address lines.

The steps that must be taken for the purpose of transferring a stored word out of memory are as follows:-

1. Apply the binary address of desired word to the address lines.
2. Activate the read input.

→ Read Only Memories (ROM)

- * Read only memory (ROM) is a type of non-volatile memory used in computers and other electronic devices.
- * Data stored in ROM cannot be electronically modified after the manufacture of memory device is done.
- * ROM is useful for storing software that is rarely changed during the life of the system i.e. firmware.
- * ROM is essentially a memory device in which permanent binary information is stored.
- * ROM is a non-volatile memory, the data once programmed will stay within the unit even when power is turned off and on again.
- * The block diagram of ROM consisting of K inputs and n outputs is shown below:-

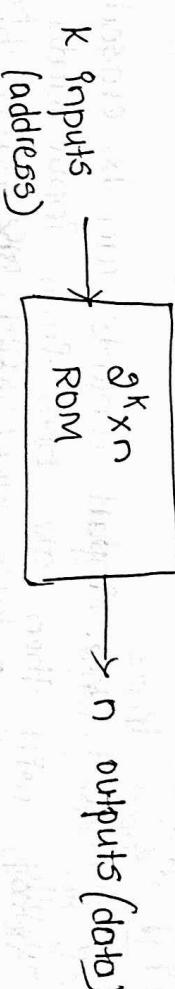


Fig: ROM block diagram

- * The inputs provide the address for memory and the outputs give the data bits of stored word that is selected by the address.
 - * The number of words in ROM is determined from K address input lines are needed to specify & words.
 - * ROM does not have data inputs because it does not have a write operation.
- Types:
- 1) Mask ROM - is a read only memory whose contents are programmed by the integrated circuit manufacturer (factory programmed). Advantage - cost is less and per bit mask ROM is more compact than any other kind of semi-conductor memory.
 - 2) Programmable read only memory (PROM):- It is a one time programmable ROM (OTR).
 - 3) Erasable programmable read only memory (EPROM):- It can be re-programmed and can be erased by exposure to strong ultraviolet light (for 10 minutes or longer).
 - 4) Electrically erasable programmable read only memory (EEPROM):- It is similar to EPROM but contents can be erased electrically.
 - * Flash memory is a type of EEPROM, memory can be erased and rewritten faster.

- ⇒ PROM (Programmable read only memory) (3)
- * PROM is a form of digital memory where the contents can be changed once after manufacture of the device.
 - * It is a type of ROM, after manufacture, the data is permanent and cannot be changed.
 - * PROMs are used in digital electronic devices to store permanent data, usually low level programs such as firmware.
 - * The difference from standard ROM is that data is written into a ROM during manufacture, while PROM, the data is programmed into them after manufacture.
 - * PROMs are manufactured blank and depending on the technology, can be programmed at wafer.
 - * Blank PROM chips are programmed by plugging them into a device called a PROM programmer.
 - * A typical PROM comes with all bits reading as "1". Burning a fuse bit during programming causes the bit to be read as "0" by blowing the fuses which is irreversible process.
 - * These devices uses high voltages to permanently destroy or create internal links (fuses or anti-fuses) within the chip.
 - * PROM can be programmed only once.

⇒ Difference b/w RAM and ROM

RAM	ROM
1) RAM stands for Random access memory.	1) ROM stands for Read only memory.
2) RAM data is volatile.	2) ROM data is non-volatile (permanent).
3) Data can be modified.	3) Data cannot be modified.
4) CPU can access data stored on RAM.	4) Data to be copied from ROM to RAM, so that CPU can access its data.
5) RAM is faster memory.	5) ROM is slower than RAM.
6) RAM is more expensive.	6) ROM is cheap than RAM.
7) Types :- 1) DRAM 2) SRAM	7) Types :- 1) PROM 2) EEPROM 3) EEPROM
8) Used for both read and write.	8) Used only for read operation.
9) Large size with higher capacity.	9) Small size with less capacity.
10) Used in CPU cache, primary memory.	10) Used in firmware, microcontrollers.

⇒ EPROM (Erasable programmable read only memory)

<p>* Erasable programmable read only memory is a type of programmable read only memory (PROM) that retains its data when its power supply is switched off.</p> <p>& It is an array of floating gate transistors individually programmed by an electronic device that supplies higher voltages than those normally used in digital circuits.</p> <p>* Once programmed, an EPROM can be erased by exposing it to strong ultraviolet light source.</p> <p>* EPROMs are easily recognizable by transparent fused quartz window on top of package, through which the silicon chip is visible and which permits exposure to ultraviolet light during erasing.</p>
<p>Fig:- Cross-section of a floating gate transistor.</p> <pre> graph TD C[control gate] --- FG[floating gate] FG --- S[source] FG --- D[drain] </pre> <p>The diagram illustrates a floating gate transistor structure. It features a central channel region. Above the channel is a floating gate electrode, which is connected to a control gate electrode on the left side. On the right side, there is a drain electrode, and at the bottom, there is a source electrode.</p>

other parts of integrated circuit. A control gate electrode is deposited and further oxide covers it.

- * The programming process is not electrically reversible. To erase the data stored in the array of transistors, ultraviolet light is directed onto the die. Photons of UV light cause ionization within the silicon dioxide which allows the stored charge on floating gate to escape.
- * Since the whole memory array is exposed, all the memory is erased at the same time.

Application:- When rapid upgrades of firmware needs considered.

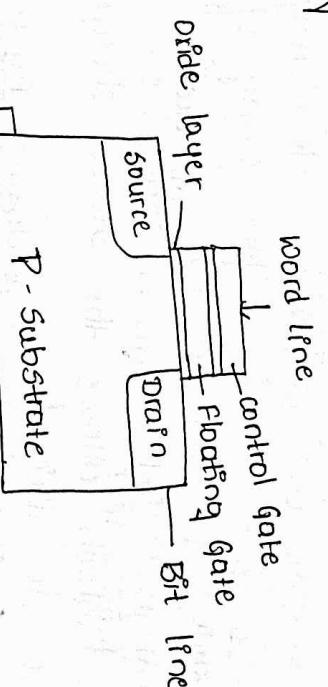
Flash memory

- * Flash memory is non-volatile, that no power is needed to maintain the information stored in the chip.
- * Flash memory offers fast read access times and has better shock resistance than other hard disks, and they find their major applications in battery powered devices.

Operation:

- * The information is stored in flash memory as an array of floating gate transistors called cells, each of which stores one bit of information.
- * Modern flash memory devices - referred to as multi level cell devices, can store more than 1 bit per cell, by varying the number of electrons placed on floating gate of a cell.

Flash memory uses memory cells similar to an EEPROM, but with much thinner, precisely grown oxide between floating gate and source.



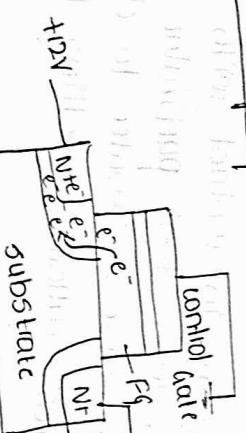
- * The floating gate can come in contact with word line only through control gate. When that particular link is closed, the cell will have value '1'. In order to change the value to '0', a process called tunnelling has to be done.
- * This charge cancels the electric field from control gate and thus causes to modify threshold voltage (V_t) of the cell.
- * During read process, a voltage that is below V_t is applied to control gate. This depends whether the channel should be conducting or insulating, which is in turn controlled by floating gate charge.
- * This causes the channel to know the current flow and hence binary code is formed.

p control gate (word line)



When +ve voltage is applied to drain, the electrons from source move to the floating gate.

→ flash memory operation (when no electrons FG becomes 1)



When +ve voltage is applied to source, with both Drain and control gate grounded, the electrons flow from floating gate to the source terminal.

When +ve voltage is applied to source, with both Drain and control gate grounded, the electrons flow from floating gate to the source terminal.

When +ve voltage is applied to source, with both Drain and control gate grounded, the electrons flow from floating gate to the source terminal.