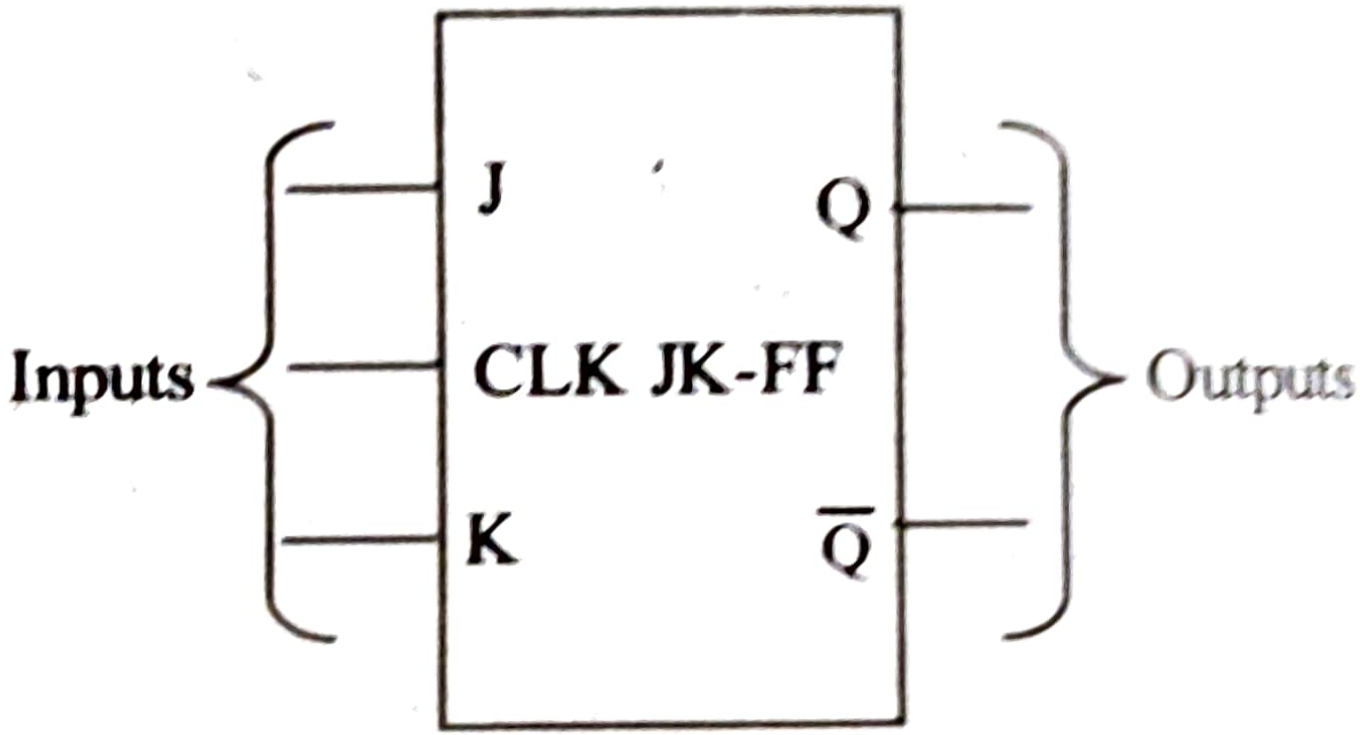
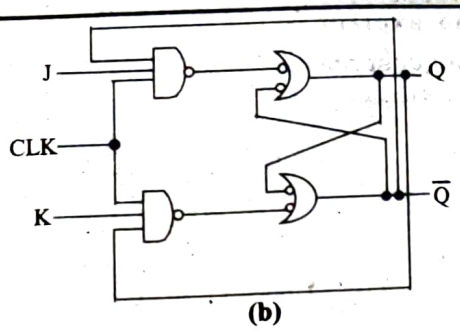
Q25. Explain the operation of JK flip-flop with neat diagram.

Ans:

The logic symbol and circuit diagram of *JK*-flip-flop is as shown in figure (1).



(a)



(b)

Figure (1)

Where, J, K – Data inputs

CLK – Clock input

Q – Normal output

$\overline{Q}$ – Complementary output.

The truth table of *JK*-flip flop is as shown in table (1).

	Inputs			Output	
Operating modes	CLK	J	K	Q	Effect on output Q
Hold		0	0	No change	No change/disable
Reset		0	1	0	Reset (i.e., cleared to 0)
Set		1	0	1	Set to 1
Toggle		1	1	Toggle	Changes to opposite state (i.e., from 0 to 1 (or) 1 to 0)

Table (1)

Working: The operation of JK-flip-flop is described as follows,

Case (i)

If both the inputs of *JK*-flip-flop are '0' (i.e., $J = K = 0$), then the flip-flop enters into hold mode. In this mode, output remains same as that of the previous state (i.e., no change in the output).

Case (ii)

If $J = 0, K = 1$, then the flip flop enters into reset mode. In reset mode, the data inputs reset the output 'Q'. Thus, the outputs are $Q = 0, \overline{Q} = 1$.

Case (iii)

If $J = 1, K = 0$, then the flip flop enters into set mode. In this mode, the data inputs set the output 'Q'. Thus, the outputs are $Q = 1, \overline{Q} = 0$.

Case (iv)

If both the inputs of *JK* -flip flop are '1' (i.e., $J = K = 1$), then the flip-flop enters into toggle mode, in which output continuously shifts between logic '0' and logic '1' for complete clock pulse. Thus, uncertain output is obtained.

Timing Diagram: The timing diagram of *JK*-flip-flop is as shown in figure (2).

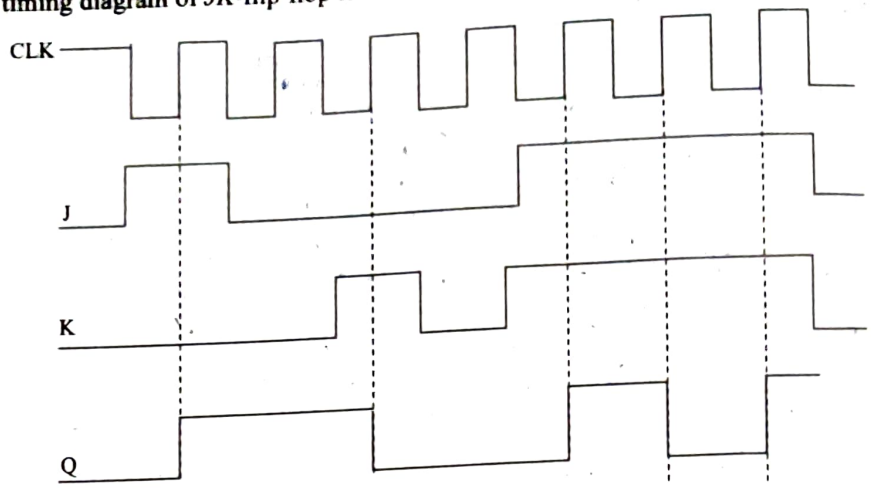


Figure (2): Timing Diagram

Q26. Explain T flip-flop along with its truth table.

Ans:

T-Flip-flop or Toggle Flip -flop can be obtained from JK flip-flop, if both *J* and *K* inputs are combined together. The diagram of *T* flip-flop is as shown in figure (1).

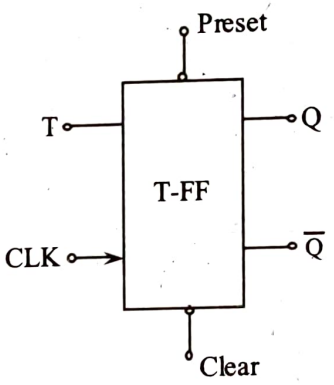


Figure (1)

The circuit diagram of *T*-flip-flop is as shown in figure (2).

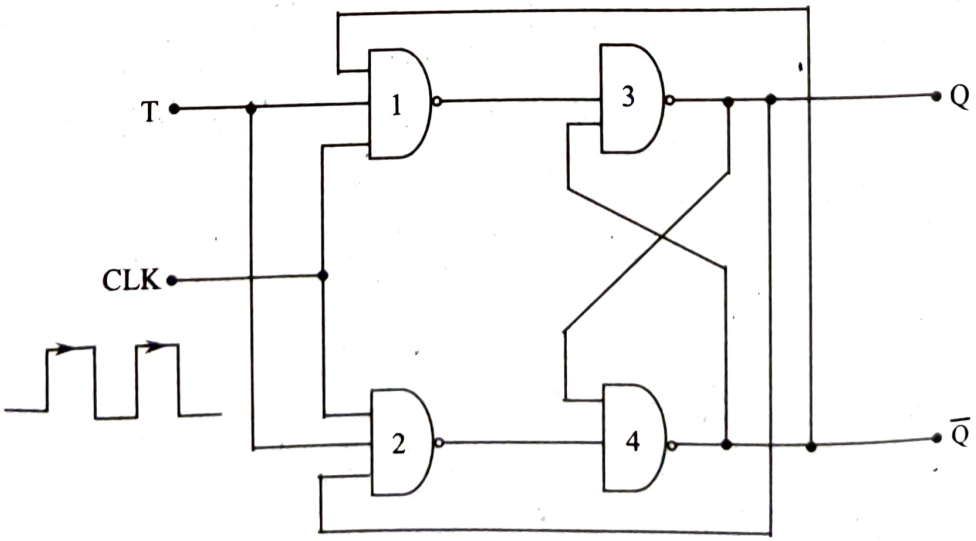


Figure (2): Circuit Diagram

The truth table of *T*-flip-flop is as shown in table.

Clock	Input (T)	Output (Q)
0	1	No change
0	0	
1	0	Q_n
1	1	Q_n

Table

From the above truth table, it is clear that *T*-flip-flop produces toggled output, when *T* input and clock signal are high. i.e., for, CLK = 1, input (*T*) = 1, output = $\overline{Q}_n$. In other case i.e., when *T*-input is low, there is no change in the output, it maintains the previous state.

Timing Diagram: The timing diagram of *T*-flip-flop is as shown in figure (3).

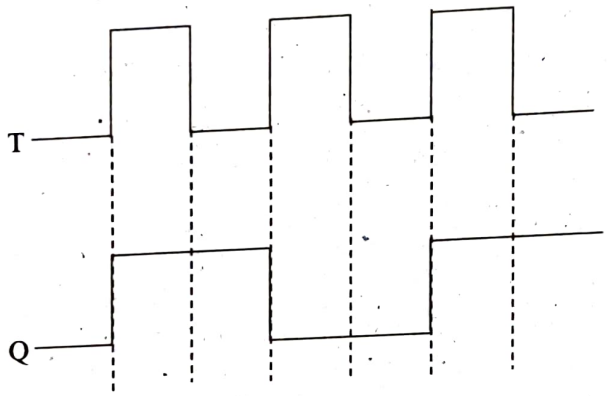


Figure (3): Timing Diagram

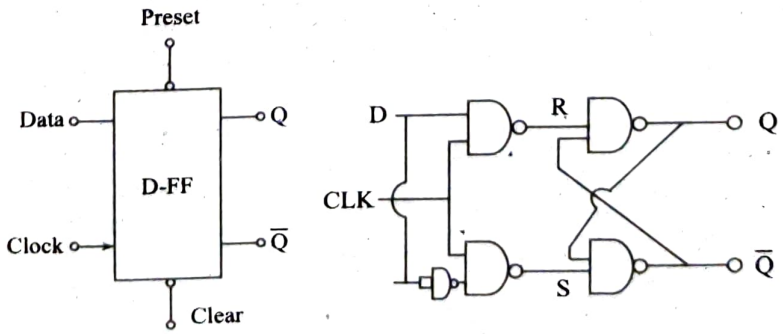
Application

T-flip-flop is particularly used to construct up-down counters.

Q27. Explain the operation of D-flip flop with the help of truth table.

Ans:

D-flip-flop (or) delay flip-flop is the modified form of *SR* (or) *JK*-flip-flop. It has single input (i.e., *D*) and two outputs (i.e., *Q* and $\overline{Q}$). The logic and circuit diagrams of *D*-flip-flop is as shown in figure (1).



(a) Logic symbol **(b) Circuit diagram**

Figure (1)

Here, J, K – Data inputs

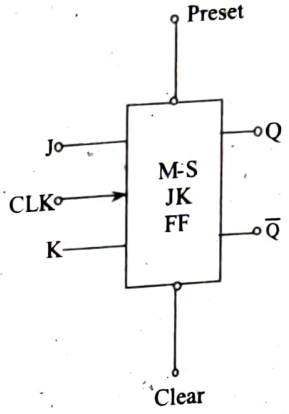
CLK – Clock inputs

$\overline{CLK}$ – Inverter clock input

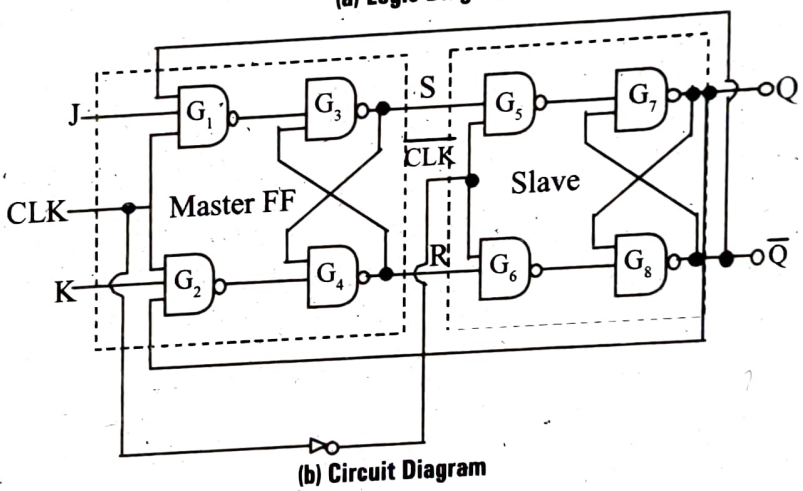
Q – Output

$\overline{Q}$ – Complementary output.

The logic and circuit diagrams of master slave *JK*-flip-flop is as shown in figure (2).



(a) Logic Diagram



(b) Circuit Diagram

Figure (2)

The truth table of master slave *JK*-flip-flop is shown in table.

PR	CLR	CLK	J	K	Q
0	0	×	×	×	×
0	1	×	×	×	1
1	0	×	×	×	0
1	1	×	0	0	NC
1	1	⎍	0	1	Reset
1	1	⎍	1	0	Set
1	1	⎍	1	1	Toggle

Table

The truth table of *D*-flip-flop is shown in table.

Input		Output	Operation
CLK	D	Q_n	
0	0	Q_{n-1}	No change
0	1	Q_{n-1}	
1	0	0	Data Storage
1	1	1	

Table

Operation: The output of the D-flip-flop depends on the clock signal applied at its input.

Case (i)

When clock signal is low (i.e., clock = 0), there is no change in the output.

Case (ii)

If clock signal is high (i.e., clock = 1), data storage takes place.

For $D = 0$; Reset = High, $Q = 0$

For $D = 1$; Set = High, $Q = 1$

This indicates that the input data appears at the output after some delay i.e., at the end of the clock pulse. Thus, it is referred to as delay flip-flop.

Timing Diagram: The timing diagram of *D*-flip-flop is as shown in figure (2).

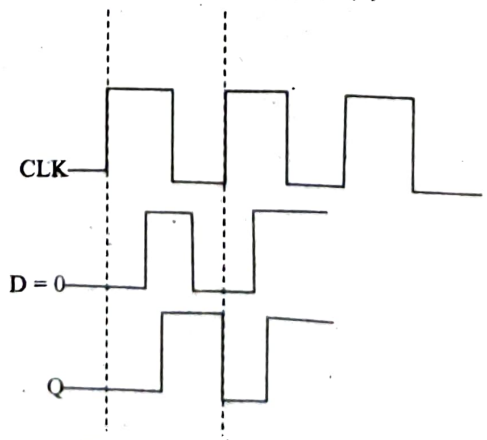


Figure (2): Timing Diagram

Applications

1. It is used as a delay device
2. It is also used as a latch to store 1-bit of binary information.

Q29. Discuss in detail about the pulse triggered S-R flip-flop. Also draw the output waveform of this flip flop and explain it with an example.

Ans: Model Paper-II, Q10

A master slave (or) a pulse triggerd *SR*-flip-flop is a combination of two *SR*-flip-flops, with feedback connection from output to input. The first flip-flop is referred as master and second flip-flop is referred as slave. The logic diagram of master-slave *SR*-flip-flop is as shown in figure (1).

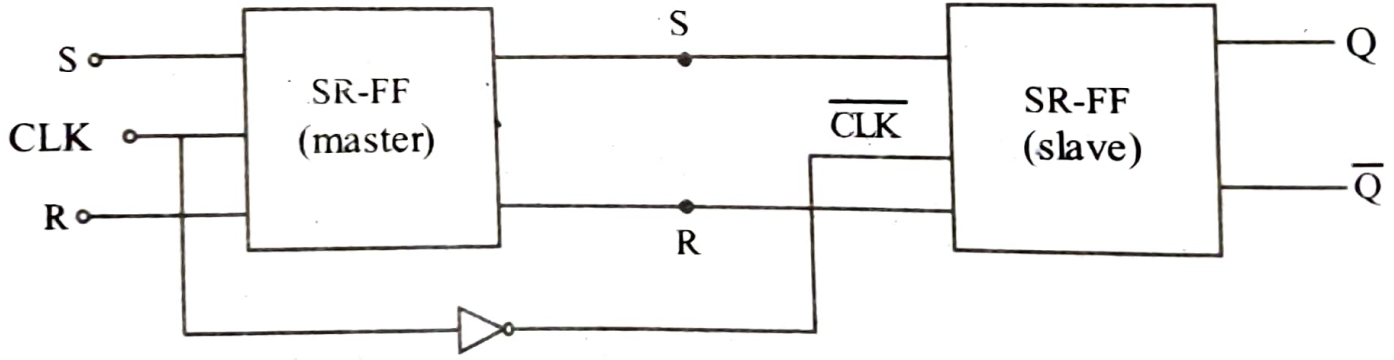


Figure (1)

Here, S, R – Data inputs

CLK – Clock inputs

$\overline{CLK}$ – Inverter clock input

Q – Output

$\overline{Q}$ – Complementary output.

The logic diagram of master slave SR-flip-flop is as shown in figure (2).

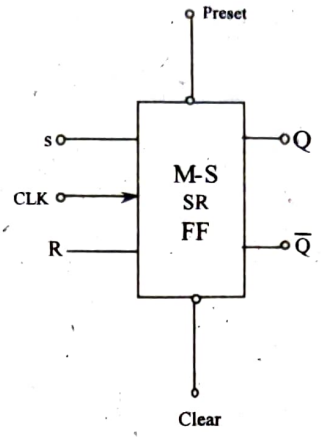


Figure (2)

Design and Operation of Master-slave RS Flip-flop: A master-slave RS flip-flop consists of,

1. A master RS flip-flop
2. A slave RS flip-flop
3. An inverter.

Figure (3) illustrates the circuit diagram of a RS master slave flip-flop.

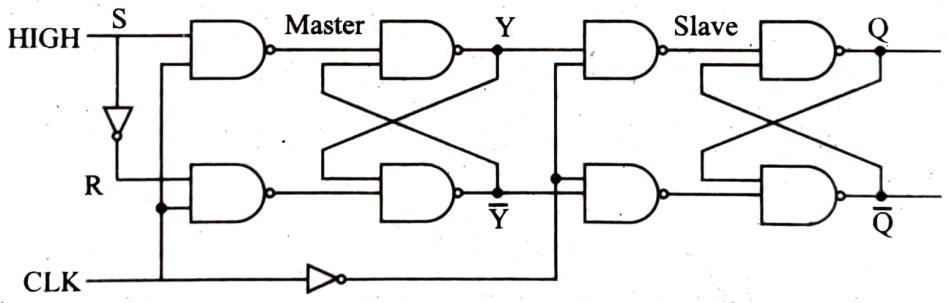


Figure (3)

The slave flip-flop is enabled when the clock pulse 'CLK' is 0 (i.e., when the output of the inverter is 1, it serves as a clock input to the slave flip-flop).

Thus, $Q = Y$ and $\overline{Q} = \overline{Y}$.

When the clock pulse 'CLK' becomes 1, the master flip-flop becomes enabled and the information is passed from the external R and S to the master flip-flop. The slave flip-flop is disabled as long as the clock pulse 'CLK' is 1. As the 'CLK' becomes '0' the master flip-flop gets disabled.

The operation of the master-slave flip-flop can be demonstrated by a truth table as shown below.

R	S	Clk	Q	Comments
0	1	0	0	Clear state
0	1	1	1	Set state

Table

The operation of master slave *SR* flip-flop is described as follows,

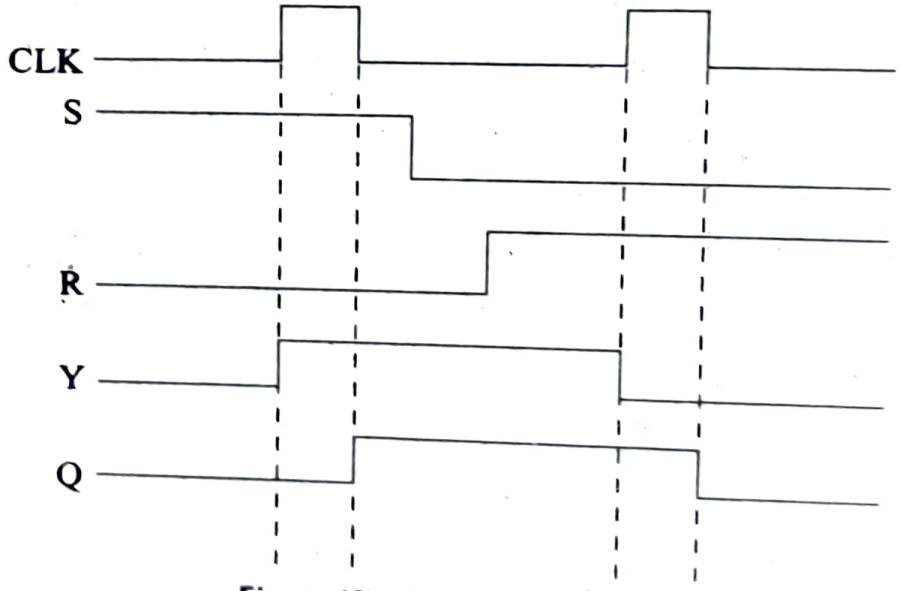
Case (i)

When $R = 0$, $S = 1$ and CLK = 0, the master flip flop enters into clear state and produces output $y = 0$ and $\bar{y} = 1$. These outputs are fed to the slave flip-flop when clock signal goes low. The *S* and *R* inputs force slave flip-flop to reset and produce outputs $Q = 0$; $\overline{Q} = 1$.

Case (ii)

When $R = 0$, $S = 1$ and CLK = 1, the master flip-flop enters into set state and produces output $y = 1$ and $\bar{y} = 0$. But the output remains constant and when the clock signal goes low, slave flip-flop becomes active and set the output to $Q = 1$; $\overline{Q} = 0$.

The timing diagram of master slave SR flip flop is as shown in figure (4).



Q19. What is a sequential circuit? Explain its types.

Ans: A logic circuit whose output at any instant of time not only depends on current input, but also on past output is known as a sequential circuit. This include memory elements and combinational circuits.

Depending on the timing of signals, sequential circuits are classified as two types. They are.

1. Synchronous sequential circuit
2. Asynchronous sequential circuit.

1. Synchronous Sequential Circuit

A circuit is said to be synchronous circuit if its output depends on the input signals only at discrete interval of time i.e., memory device undergoes a change only at a discrete time interval. These circuits use flip-flops as their memory device for storing the binary information. A timing element known as clock generator is needed for synchronization and provides a periodic pulse train signals.

Figure (1) represents the circuit diagram of a synchronous sequential circuit.

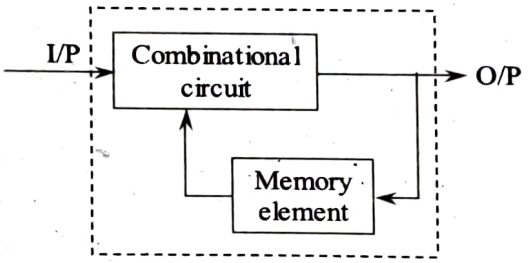


Figure (1): Block Diagram of Synchronous Sequential Circuit

It can be observed from figure (1) that the output can be obtained from combinational circuit or flip flop or both.

As the output is only obtained at discrete instant of time with the input, the state of flip-flop i.e., the memory device changes at active state of clock pulses and remains unaffected when the clock pulse is not active.

Synchronous sequential circuits are also called clocked-sequential circuits. If more than one flip-flop is used, then they have a common clock pulse. These circuits are simple to design and have limited speed of operation due to time delay.

2. Asynchronous Sequential Circuit

A circuit is said to be asynchronous sequential circuit if its output depends on the input signals at all instants of time i.e., the output changes accordingly with the input. Such circuits uses time delay latches, gate devices as their memory element.

Asynchronous sequential circuits does not require clock pulses and are called as combinational circuit with feedback.

At the time of unclock, the change in input is simultaneous for clocked sequential circuits as shown in figure (2).

These circuits are more difficult to design and the speed of operation is high.

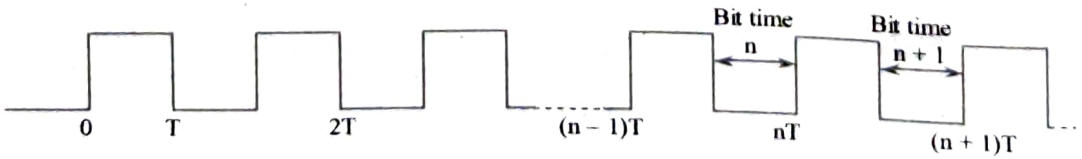


Figure (2): Timing Diagram of Clock Pulse

Figure

Q2. Distinguish between combinational logic and sequential logic circuits.

Ans:

Model Paper-I, Q1(i)

The comparison between sequential and combinational circuits is mentioned below.

	Combinational Logic Circuits		Sequential Logic Circuits
1.	Output depends only on the present input.	1.	Output depends on the present input and past output also.
2.	Easier to design.	2.	Comparatively harder to design.
3.	Speed of operation is high.	3.	Speed of operation is comparatively low.
4.	Memory unit is not required.	4.	Memory unit is required to store the past outputs.
5.	Example: Parallel adder.	5.	Example: Serial adder.

Q3. Compare synchronous and asynchronous sequential circuits.

Ans:

The comparison between synchronous and asynchronous sequential circuits is mentioned below:

Comparison of RAM, ROM, PROM, EPROM

Memory Type	Writable	Erasable Length	Speed	Relative Size	Performance	Cost Per Byte
RAM	Possible to write any number of times	Complete data can be erased	Fast to read and write	Small	Volatile	Costly
ROM	Not possible	–	Fast to read	Large	Permanent	Moderate
PROM	Possible to write but only once	–	Fast to read	Large	Permanent	Moderate
EPROM	Possible to write any number of times	Complete data can be erased	Fast to read and write	Very Large	Persistent	Moderate

Q30. Write the characteristic and excitation tables for JK, RS, T and D flip-flops.

Ans:

JK-Flip Flop: The block diagram of a *JK* flip-flop is as shown in figure (1).

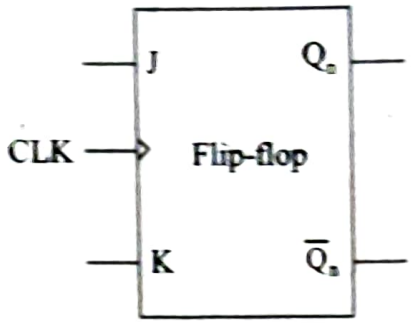


Figure (1)

The characteristic table of a *JK* flip-flop is given as shown in table (1).

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

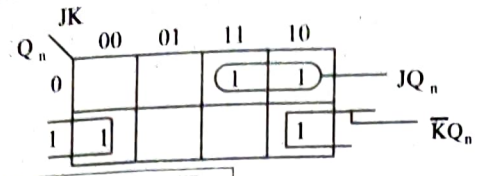
Table (1): Characteristic Table

The excitation table of a *JK* flip-flop is given as shown in table (2).

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

Table (2): Excitation Table of JK Flip-flop

From the table (2), characteristic equation can be obtained using k-map as,



$$\therefore Q_{n+1} = JQ_n + \overline{K}Q_n$$

RS-Flip-flop or SR- Flip-flop: The block diagram of *RS* flip-flop is as shown in figure (2).

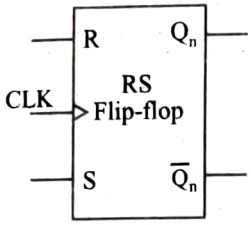


Figure (2): RS Filp-flop

The characteristic table of *RS* flip-flop is shown in table (3).

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	?

Table (3): Characteristic Table

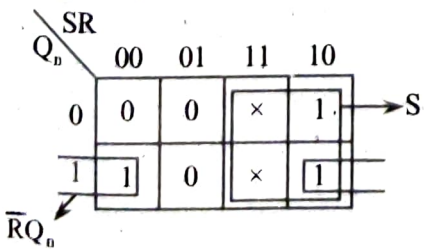
The excitation table for *RS* flip-flop is shown in table (4).

Q_n	Q_{n+1}	S	R
0	0	0	×
0	1	1	0
1	0	0	1
1	1	×	0

Where, × - don't care

Table (4): Excitation Table of RS Flip-flop

From the table (4), characteristic equation can be obtained by using *K*-map as,



$$\therefore Q_{n+1} = S + \overline{R}Q_n$$

T-Flip-Flop: The block diagram of *T*-flip-flop is as shown in figure (3).

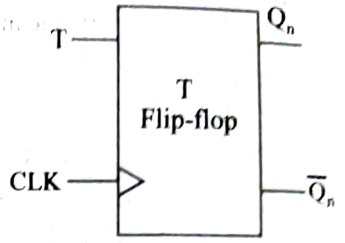


Figure (3)

The characteristic table of *T* flip-flop is shown in table (5).

T	Q_{n+1}
0	(Q_n)
1	$(\overline{Q}_n)$

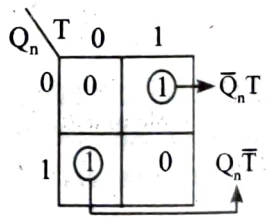
Table (5): Characteristic Table of T Flip-flop

The excitation table for *T* flip-flop is shown in table (6).

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

Table (6): Excitation Table of T Flip-flop

From the table (6), characteristic equation can be obtained by using *K*-map as,



From the *K*-map, characteristic equation is obtained as,

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

D Flip-flop: The block diagram of *D* flip-flop is as shown in figure (4).

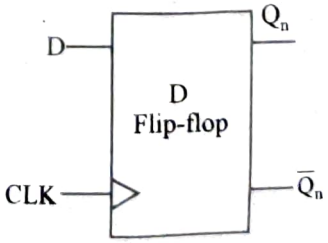


Figure (4): D Flip-flop

The characteristic table of *D* flip-flop is shown in table (7).

D	Q_{n+1}
0	0
1	1

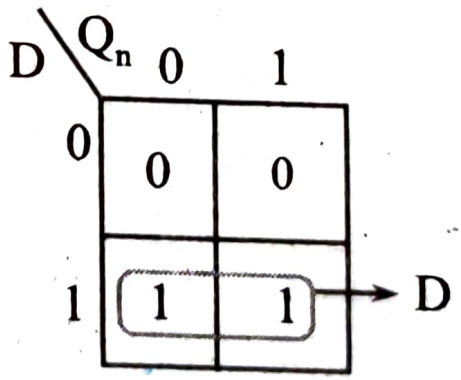
Table (7): Characteristic of D Flip-flop

The excitation table for *D* flip-flop is shown in table (8).

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

Table (8): Excitation Table of D Flip-flop

From the table (8), characteristic equation can be obtained as,



Hence, the characteristic equation is,

$$\boxed{Q_{n+1} = D}$$

Q39. Draw the logic diagram of Mealy model and explain its operation.

Ans:

The logic diagram of a Mealy model is as shown in figure (1).

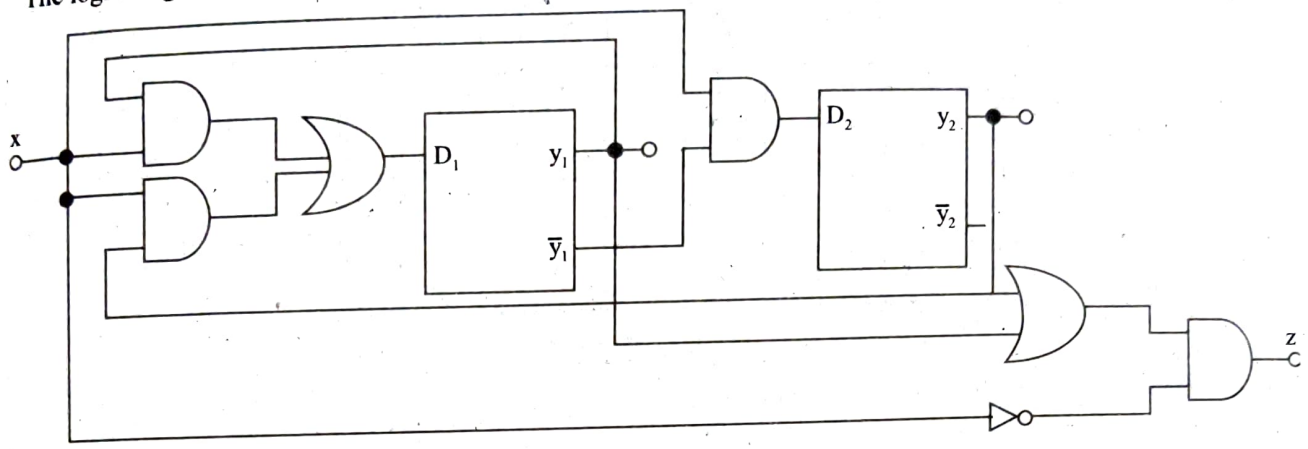


Figure (1): Logic Diagram of a Mealy Model

It consists of two *D*-flip-flops, an input (x) and an output (z).

During clock pulse, any variations in the input directly affect the output of the circuit. As a result, the input variations and output remains unsynchronized with the clock, thereby producing false outputs.

Generally, state equation helps in determining the behaviour of clocked sequential circuit. Here, input to '*D*' describes the value of next state.

The state equations for Mealy model can be obtained as,

$$y_1(t+1) = y_1(t)\,x(t) + y_2(t)\,x(t) \quad \ldots (1)$$

$$y_2(t+1) = \overline{y}_1(t)x(t) \quad \ldots (2)$$

The output equation (z) is,

$$z(t) = \{y_1(t) + y_2(t)\}\overline{x}(t) \quad \ldots (3)$$

Where,

$x(t)$ – Present input

$z(t)$ – Present output.

$y(t+1)$ – Next state of the flip-flop.

Let,

$Y_1(t) = y_1(t+1)$ and

$Y_2(t) = y_2(t+1)$

Then equations (1), (2) and (3) can be written as,

$$Y_1 = y_1x + y_2x \quad \ldots (4)$$

$$Y_2 = \overline{y}_1x \quad \ldots (5)$$

$$z = (y_1 + y_2)\overline{x} \quad \ldots (6)$$

The state table for Mealy model can be obtained using equations (4), (5) and (6) as shown in table.

PS		NS				output	
		x = 0		x = 1		x = 0	x = 1
y_1	y_2	Y_1	Y_2	Y_1	Y_2	z	z
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

Table: State Table

Its corresponding state diagram is shown in figure (2).

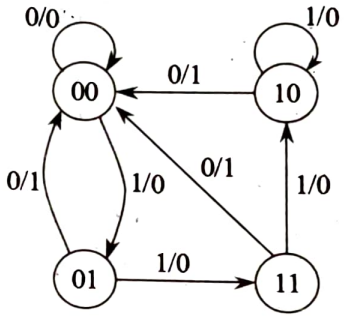


Figure (2): State Diagram

Q40. Draw and explain the logic diagram of Moore model.

Ans:

The logic diagram of Moore model is illustrated as shown in figure (1).

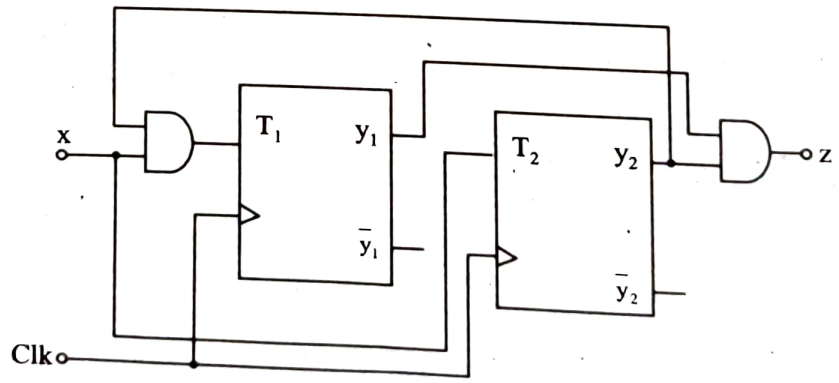


Figure (1): Logic Diagram of a Moore Model

It consists of two T flip-flops, an input (x) and an output (z).

In Moore model, the output depends only on the present state.

From figure (1), the input equations (T_1 and T_2) and output equation (z) are obtained as,

$$T_1 = y_2 x$$

$$T_2 = x$$

$$z = y_1 y_2$$

Generally, the characteristic equation of a T flip-flop can be written as,

$$Q(t+1) = T\overline{Q} + \overline{T}Q \quad \ldots (1)$$

On substituting the corresponding values of T_1 and T_2 in equation (1), the state equations for Moore model are obtained as,

$$y_1(t+1) = Y_1 = (y_2 x) \oplus y_1 = \overline{(y_2 x)} y_1 + (y_2 x)\overline{y}_1 = y_1\overline{y}_2 + y_1\overline{x} + \overline{y}_1 y_2 x \quad \ldots (2)$$

$$y_2(t+1) = Y_2 = x \oplus y_2 = x\overline{y}_2 + \overline{x}y_2 \quad \ldots (3)$$

$$\text{Output } z = y_1 y_2 \quad \ldots (4)$$

The state table for Moore model can be obtained using equations (2), (3) and (4) as shown in table.

NS						
PS		x = 0		x = 1		O/P
y_1	y_2	Y_1	Y_2	Y_1	Y_2	z
0	0	0	0	0	1	0
0	1	0	1	1	0	0
1	0	1	0	1	1	0
1	1	1	1	0	0	1

Table: State Table

The corresponding state diagram is shown in figure (2).

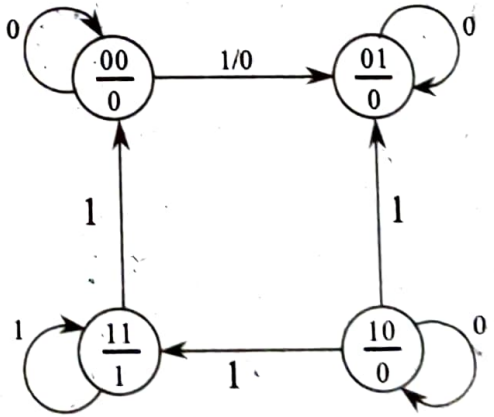


Figure (2): State Diagram

Q48. Design a ripple counter considering one example.

Ans:

Ripple Counter

Ripple counter is an asynchronous counter in which the clocks of all flip-flops are not connected.

Example

BCD Counter

BCD counter is also known as mod-10 counter or decade counter. BCD counter counts binary coded decimal from 0000 to 1001. When the 11^{th} clock pulse is applied, the next state of register becomes 0000 due to external logic gate. Since BCD doesn't have a regular pattern, it becomes all the more important to go through the sequential circuit design procedure to derive a BCD synchronous counter circuit.

The state diagram of a BCD ripple counter is as shown in figure (1).

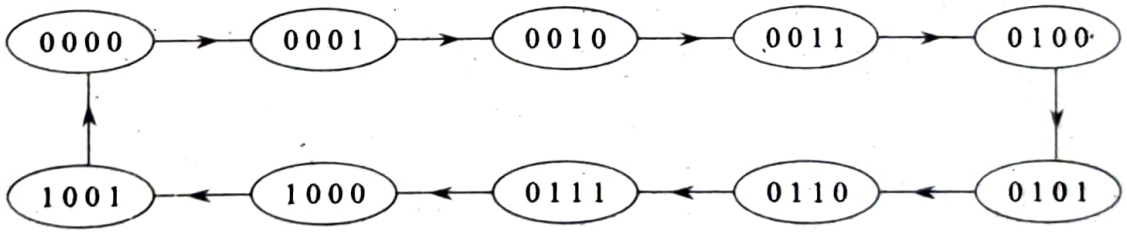


Figure (1): State Diagram

A D-flip-flop is used. The truth table of D-FF is shown in table (1).

Count pulse	D	Q
0	×	×
1	0	0
1	1	1

Table (1)

From this, the function table of the ripple counter is written as shown in table (2).

Count Pulse	Present state				Next state				Output				
	A	B	C	D	A	B	C	D	D_A	D_B	D_C	D_D	y
0	0	0	0	0	0	0	0	1	0	0	0	1	0
1	0	0	0	1	0	0	1	0	0	0	1	0	0
2	0	0	1	0	0	0	1	1	0	0	1	1	0
3	0	0	1	1	0	1	0	0	0	1	0	0	0
4	0	1	0	0	0	1	0	1	0	1	0	1	0
5	0	1	0	1	0	1	1	0	0	1	1	0	0
6	0	1	1	0	0	1	1	1	0	1	1	1	0
7	0	1	1	1	1	0	0	0	1	0	0	0	0
8	1	0	0	0	1	0	0	1	0	0	0	1	0
9	1	0	0	1	0	0	0	0	0	0	0	0	1

Table (2): State Transition Table

The expressions for D_A, D_B, D_C, D_D can be obtained by using the *K*-map simplification technique as,

For D_A

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	×	×	×	×
10	0	0	×	×

$$\therefore \boxed{D_A = BCD}$$

For D_B

AB \ CD	00	01	11	10
00			1	
01	1	1		1
11	×	×	×	×
10			×	×

$$\therefore \boxed{D_B = \overline{C}B + B\overline{D} = B(\overline{C} + \overline{D})} \quad \ldots(1)$$

For D_C

AB \ CD	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	×	×	×	×
10	0	0	×	×

$$\therefore \boxed{D_C = C\overline{D} + \overline{A}\,\overline{C}\,\overline{D}} \quad \ldots(2)$$

For D_D

AB \ CD	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	×	×	×	×
10	1	0	×	×

$$\therefore \boxed{D_D = \overline{D}} \quad \ldots (4)$$

The implementation of BCD ripple counter using above expressions requires four *D*-flip-flop as shown in figure (2).

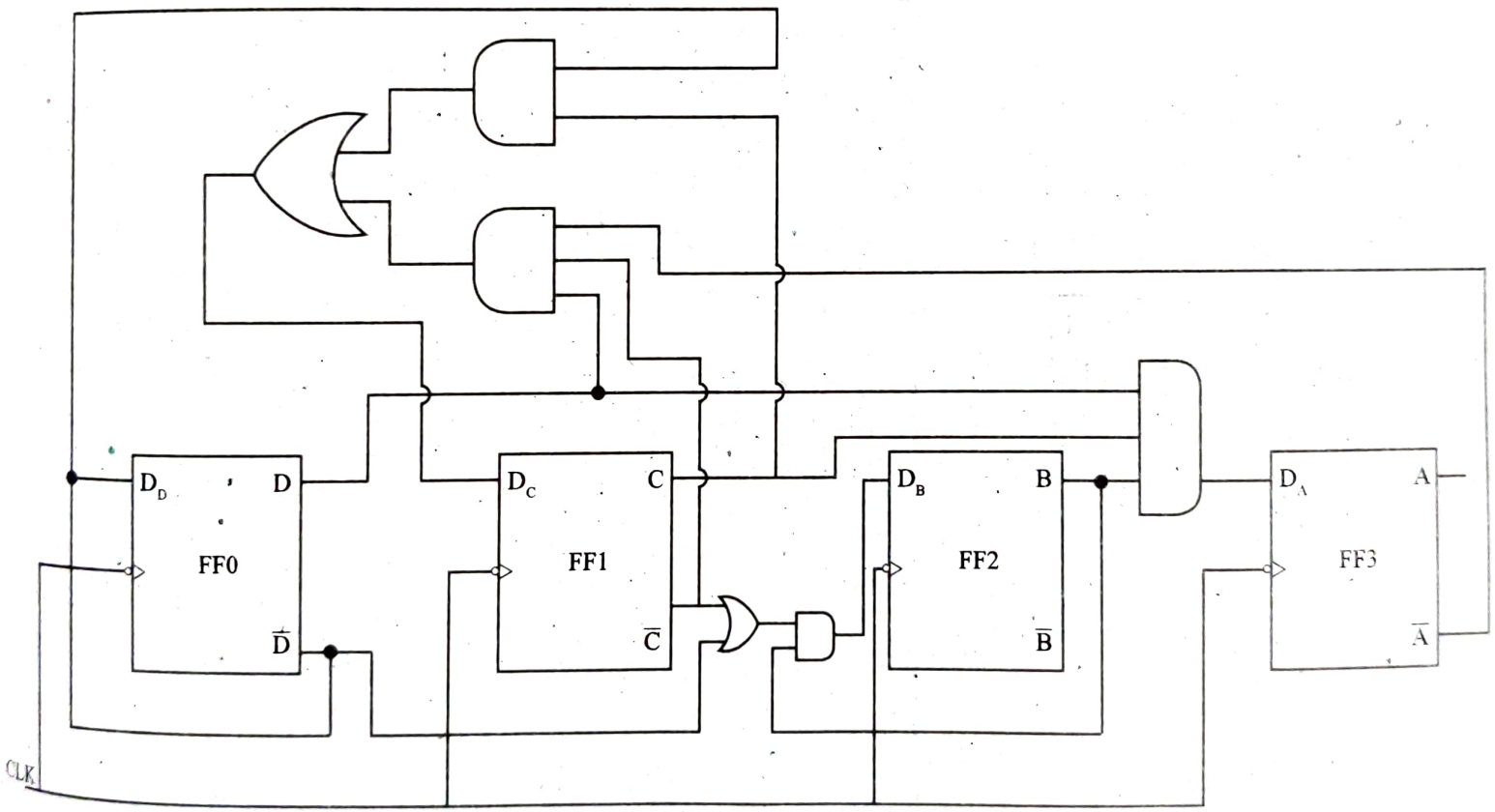
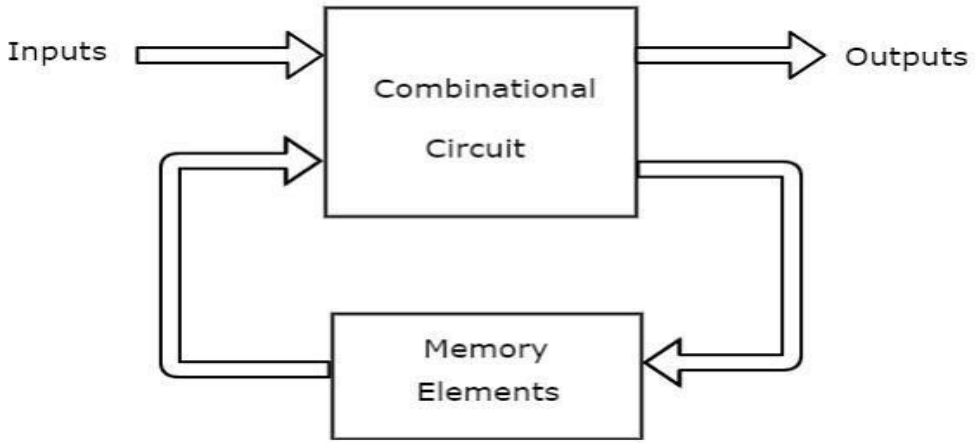


Figure (2): BCD Counter

UNIT - V

SEQUENTIAL CIRCUITS

The following figure shows the **block diagram** of sequential circuit.



This sequential circuit contains a set of inputs and output(s). The output(s) of sequential circuit depends not only on the combination of present inputs but also on the previous output(s). Previous output is nothing but the **present state**. Therefore, sequential circuits contain combinational circuits along with memory (storage) elements. Some sequential circuits may not contain combinational circuits, but only memory elements.

Following table shows the **differences** between combinational circuits and sequential circuits.

Combinational Circuits	**Sequential Circuits**
Outputs depend only on present inputs.	Outputs depend on both present inputs and present state.
Feedback path is not present.	Feedback path is present.
Memory elements are not required.	Memory elements are required.

Clock signal is not required.	Clock signal is required.
Easy to design.	Difficult to design.

Types of Sequential Circuits

Following are the two types of sequential circuits −

- Asynchronous sequential circuits
- Synchronous sequential circuits

Asynchronous sequential circuits

If some or all the outputs of a sequential circuit do not change (affect) with respect to active transition of clock signal, then that sequential circuit is called as **Asynchronous sequential circuit**.Therefore, most of the outputs of asynchronous sequential circuits are **not in synchronous** with either only positive edges or only negative edges of clock signal.

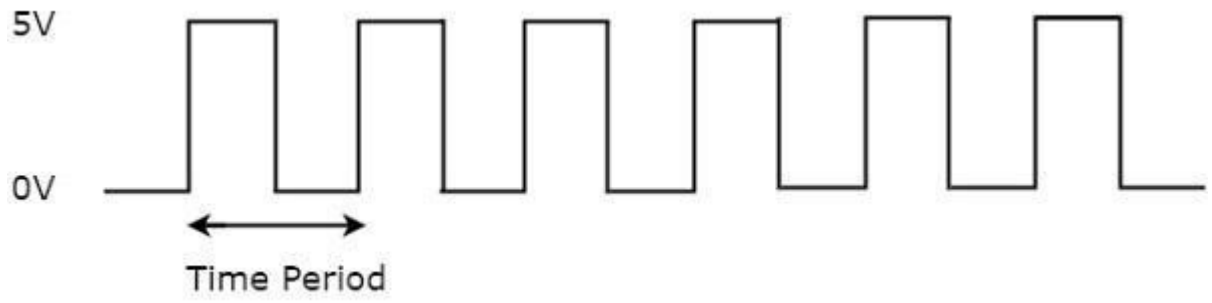
Synchronous sequential circuits

If all the outputs of a sequential circuit change (affect) with respect to active transition of clock signal, then that sequential circuit is called as **Synchronous sequential circuit**. That means, all the outputs of synchronous sequential circuits change (affect) at the same time. Therefore, the outputs of synchronous sequential circuits are in synchronous with either only positive edges or only negative edges of clock signal.

Clock Signal and Triggering

In this section, let us discuss about the clock signal and types of triggering one by one. **Clock signal**

Clock signal is a periodic signal and its ON time and OFF time need not be the same. We can represent the clock signal as a **square wave**, when both its ON time and OFF time are same. This clock signal is shown in the following figure.



This signal stays at logic High (5V) for some time and stays at logic Low (0V) for equal amount of time. This pattern repeats with some time period. In this case, the **time period** will be equal to either twice of ON time or twice of OFF time.

Types of Triggering

Following are the two possible types of triggering that are used in sequential circuits.

- Level triggering
- Edge triggering

Level triggering

There are two levels, namely logic High and logic Low in clock signal. Following are the two **types of level triggering**.

- Positive level triggering
- Negative level

triggering Edge triggering

There are two types of transitions that occur in clock signal. That means, the clock signal transitions either from Logic Low to Logic High or Logic High to Logic Low.

Following are the two **types of edge triggering** based on the transitions of clock signal.

- Positive edge triggering
- Negative edge triggering

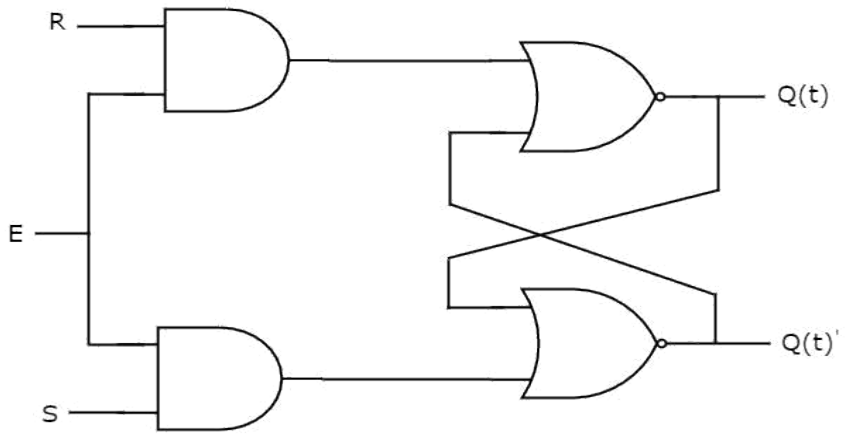
There are two types of memory elements based on the type of triggering that is suitable to operate it.

- Latches
- Flip-flops

Latches operate with enable signal, which is **level sensitive**. Whereas, flip-flops are edge sensitive. We will discuss about flip-flops in next chapter. Now, let us discuss about SR Latch & D Latch one by one.

SR Latch

SR Latch is also called as **Set Reset Latch**. This latch affects the outputs as long as the enable, E is maintained at „1". The **circuit diagram** of SR Latch is shown in the following figure.



This circuit has two inputs S & R and two outputs Q(t) & Q(t)‟. The **upper NOR gate** has two inputs R & complement of present state, Q(t)‟ and produces next state, Q(t+1) when enable, E is „1‟.

Similarly, the **lower NOR gate** has two inputs S & present state, Q(t) and produces complement of next state, Q(t+1)‟ when enable, E is „1‟.

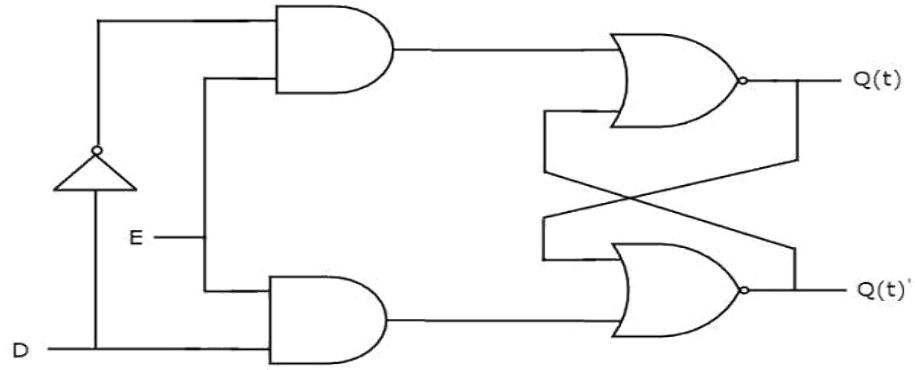
We know that a **2-input NOR gate** produces an output, which is the complement of another input when one of the input is „0‟. Similarly, it produces „0‟ output, when one of the input is „1‟.

The following table shows the **state table** of SR latch.

S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	-

D Latch

There is one drawback of SR Latch. That is the next state value can‟t be predicted when both the inputs S & R are one. So, we can overcome this difficulty by D Latch. It is also called as Data Latch. The **circuit diagram** of D Latch is shown in the following figure.



The following table shows the **state table** of D latch.

D	Q(t+1)
0	0
1	1

In first method, **cascade two latches** in such a way that the first latch is enabled for every positive clock pulse and second latch is enabled for every negative clock pulse. So that the combination of these two latches become a flip-flop.

In second method, we can directly implement the flip-flop, which is edge sensitive. In this chapter, let us discuss the following **flip-flops** using second method.

- SR Flip-Flop
- D Flip-Flop
- JK Flip-Flop
- T Flip-Flop

SR Flip-Flop

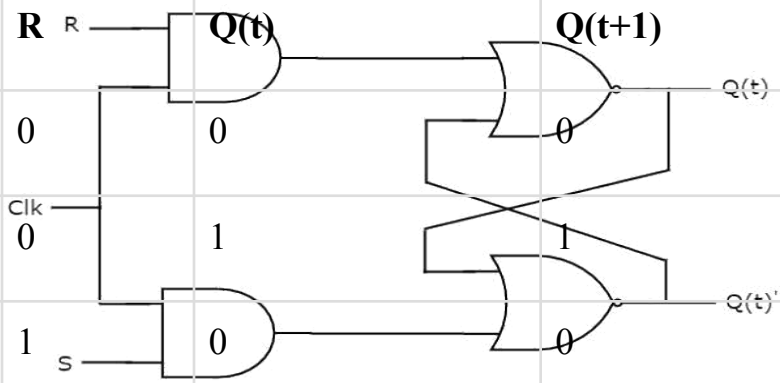
SR flip-flop operates with only positive clock transitions or negative clock transitions. Whereas, SR latch operates with enable signal. The **circuit diagram** of SR flip-flop is shown in the following figure.

The following table shows the **state table** of SR flip-flop.

S	R	Q(t+1)
0	0	Q(t+1)
0	1	0
1	0	1
1	1	-

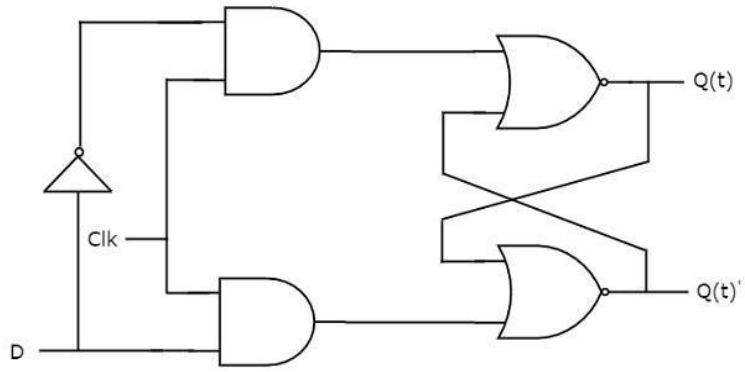
The following table shows the **characteristic table** of SR flip-flop.

Present Inputs		**Present State**	**Next State**
S	**R**	**Q(t)**	**Q(t+1)**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X



D Flip-Flop

D flip-flop operates with only positive clock transitions or negative clock transitions. Whereas, D latch operates with enable signal. That means, the output of D flip-flop is insensitive to the changes in the input, D except for active transition of the clock signal. The **circuit diagram** of D flip-flop is shown in the following figure.



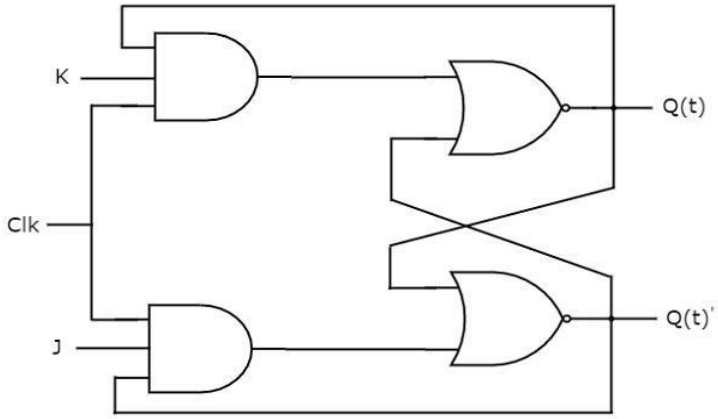
The following table shows the **state table** of D flip-flop.

D	**Q(t+1)**
0	0
0	1

From the above state table, we can directly write the next state equation as Q(t+1)=D

JK Flip-Flop

JK flip-flop is the modified version of SR flip-flop. It operates with only positive clock transitions or negative clock transitions. The **circuit diagram** of JK flip-flop is shown in the following figure.



The following table shows the **state table** of JK flip-flop.

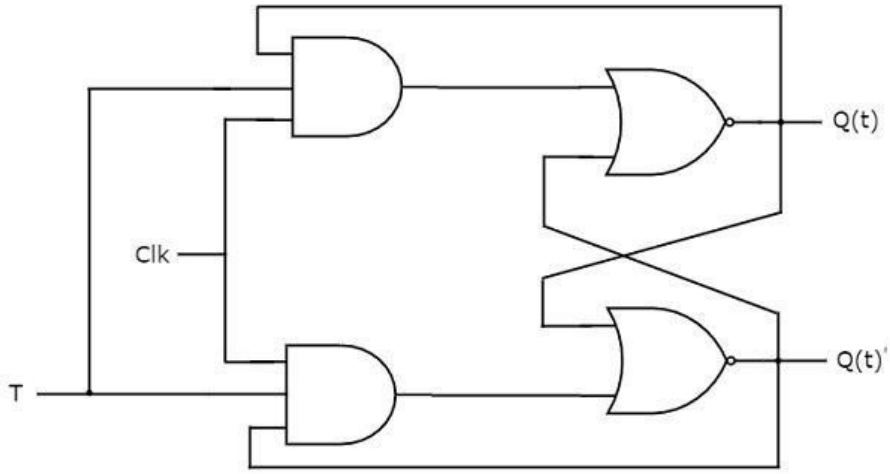
J	**K**	**Q(t+1)**
0	0	Q(t)
0	1	0
1	0	1
1	1	Q(t)'

The following table shows the **characteristic table** of JK flip-flop.

Present Inputs		**Present State**	**Next State**
J	**K**	**Q(t)**	**Q(t+1)**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

T Flip-Flop

T flip-flop is the simplified version of JK flip-flop. It is obtained by connecting the same input „T" to both inputs of JK flip-flop. It operates with only positive clock transitions or negative clock transitions. The **circuit diagram** of T flip-flop is shown in the following figure.



The following table shows the **state table** of T flip-flop.

D	Q(t+1)
0	Q(t)
1	Q(t)"

The following table shows the **characteristic table** of T flip-flop.

Inputs	Present State	Next State
T	**Q(t)**	**Q(t+1)**
0	0	0
0	1	1
1	0	1
1	1	0

From the above characteristic table, we can directly write the **next state equation** as

$Q(t+1)=T'Q(t)+TQ(t)'Q(t+1)=T'Q(t)+TQ(t)' \Rightarrow Q(t+1)=T \oplus Q(t)$

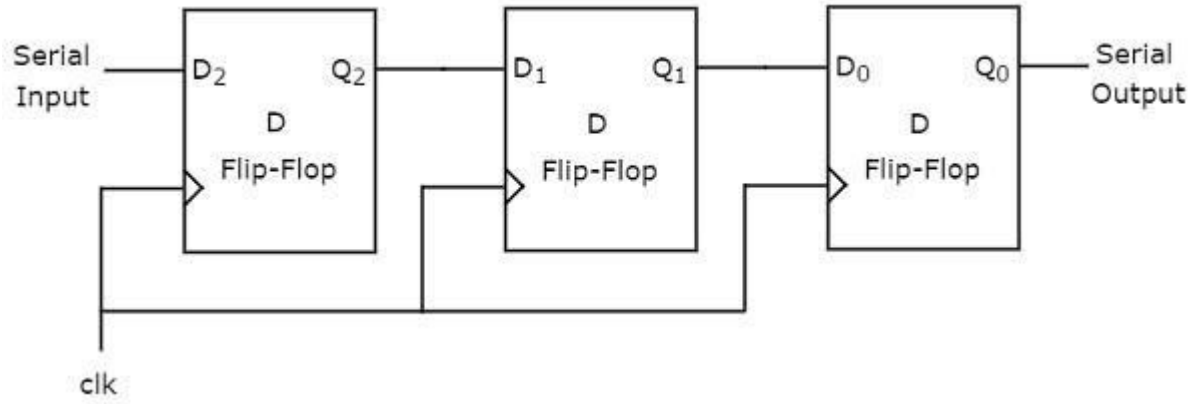
shift register:

If the register is capable of shifting bits either towards right hand side or towards left hand side is known as **shift register**. An „N" bit shift register contains „N" flip-flops. Following are the four types of shift registers based on applying inputs and accessing of outputs.

- Serial In - Serial Out shift register
- Serial In - Parallel Out shift register
- Parallel In - Serial Out shift register
- Parallel In - Parallel Out shift register

Serial In - Serial Out (SISO) Shift Register

The shift register, which allows serial input and produces serial output is known as Serial In – Serial Out **(SISO)** shift register. The **block diagram** of 3-bit SISO shift register is shown in the following figure.

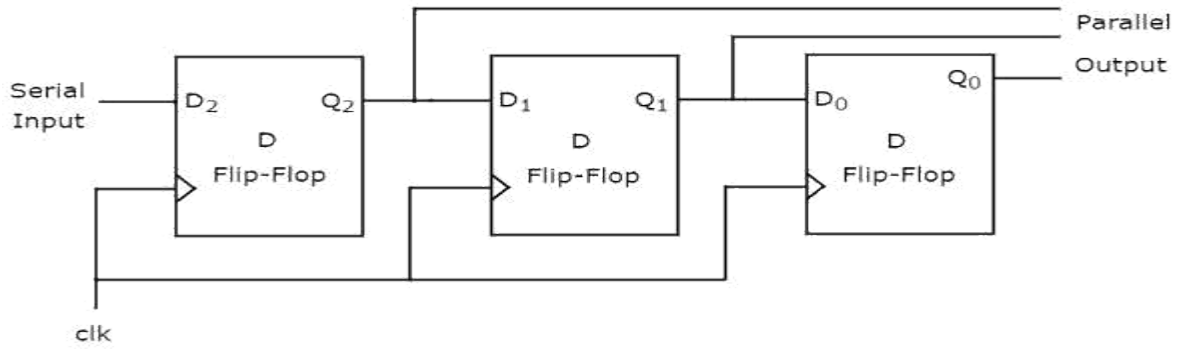


This block diagram consists of three D flip-flops, which are **cascaded**. That means, output of one D flip-flop is connected as the input of next D flip-flop. All these flip-flops are synchronous with each other since, the same clock signal is applied to each one.

In this shift register, we can send the bits serially from the input of left most D flip-flop. Hence, this input is also called as **serial input**. For every positive edge triggering of clock signal, the data shifts from one stage to the next. So, we can receive the bits serially from the output of right most D flip-flop. Hence, this output is also called as **serial output**.

Serial In - Parallel Out (SIPO) Shift Register

The shift register, which allows serial input and produces parallel output is known as Serial In – Parallel Out **(SIPO)** shift register. The **block diagram** of 3-bit SIPO shift register is shown in the following figure.



This circuit consists of three D flip-flops, which are cascaded. That means, output of one D flip-flop is connected as the input of next D flip-flop. All these flip-flops are synchronous with each other since, the same clock signal is applied to each one.

In this shift register, we can send the bits serially from the input of left most D flip-flop. Hence, this input is also called as **serial input**. For every positive edge triggering of clock signal, the data shifts from one stage to the next. In this case, we can access the outputs of each D flip-flop in parallel. So, we will get **parallel outputs** from this shift register.

Design of Asynchronous and Synchronous Circuits:

The synchronous sequential circuits change (affect) their states for every positive (or negative) transition of the clock signal based on the input. So, this behavior of synchronous sequential circuits can be represented in the graphical form and it is known as **state diagram**.

A synchronous sequential circuit is also called as **Finite State Machine** (FSM), if it has finite number of states. There are two types of FSMs.

- Mealy State Machine
- Moore State Machine

Now, let us discuss about these two state machines one by one.

Memory:

READ-ONLY MEMORY

Read-only memory (ROM) is a type of storage medium that permanently stores data on personal computers (PCs) and other electronic devices. It contains the programming needed to start a PC, which is essential for boot-up; it performs major input/output tasks and holds programs or software instructions.

Because ROM is read-only, it cannot be changed; it is permanent and non-volatile, meaning it also holds its memory even when power is removed. By contrast, random access memory (RAM) is volatile; it is lost when power is removed.

There are numerous ROM chips located on the motherboard and a few on expansion boards. The chips are essential for the basic input/output system (BIOS), boot up, reading and writing to peripheral devices, basic data management and the software for basic processes for certain utilities.

RANDOM ACCESS MEMORY

RAM (random access memory) is the place in a computing device where the operating system (OS), application programs and data in current use are kept so they can be quickly reached by the device's processor. RAM is much faster to read from and write to than other kinds of storage in a computer, such as a hard disk drive (HDD), solid-state drive (SSD) or optical drive. Data remains in RAM as long as the computer is running. When the computer is turned off, RAM loses its data. When the computer is turned on again, the OS and other files are once again loaded into RAM, usually from an HDD or SSD.

RAM TYPES

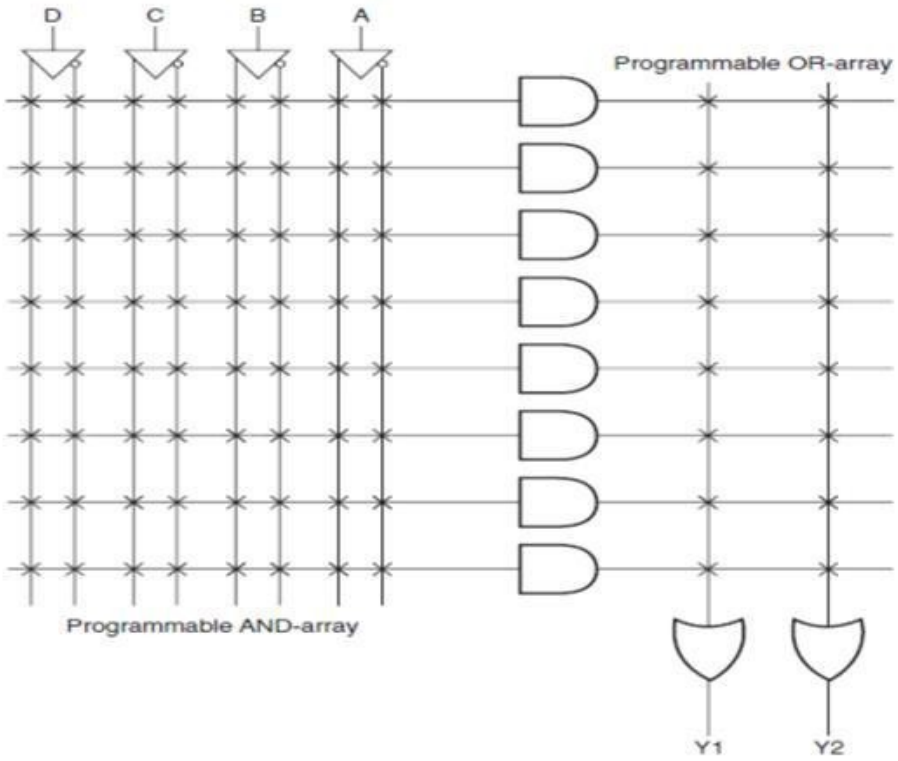
(1)Dynamic random access memory: DRAM is what makes up the typical computing device RAM and, as noted above, requires constant power to hold on to stored data.

(2)Static random access memory.:SRAM doesn't need constant power to hold on to data, but the way the memory chips are made means they are much larger and thousands of times more expensive than an equivalent amount of DRAM.

However, SRAM is significantly faster than DRAM. The price and speed differences mean SRAM is mainly used in small amounts as cache memory inside a device's processor.

PROGRAMMABLE LOGIC ARRAY

A programmable logic array (PLA) has a programmable AND array at the inputs and programmable OR array at the outputs. The PLA has a programmable AND array instead of hard-wired AND array. The number of AND gates in the programmable AND array are usually much less and the number of inputs of each of the OR gates equal to the number of AND gates. The OR gate generates an arbitrary Boolean function of minterms equal to the number of AND gates. Figure below shows the PLA architecture with four input lines, a programmable array of eight AND gates at the input and a programmable array of two OR gates at the output.



ADVANTAGES

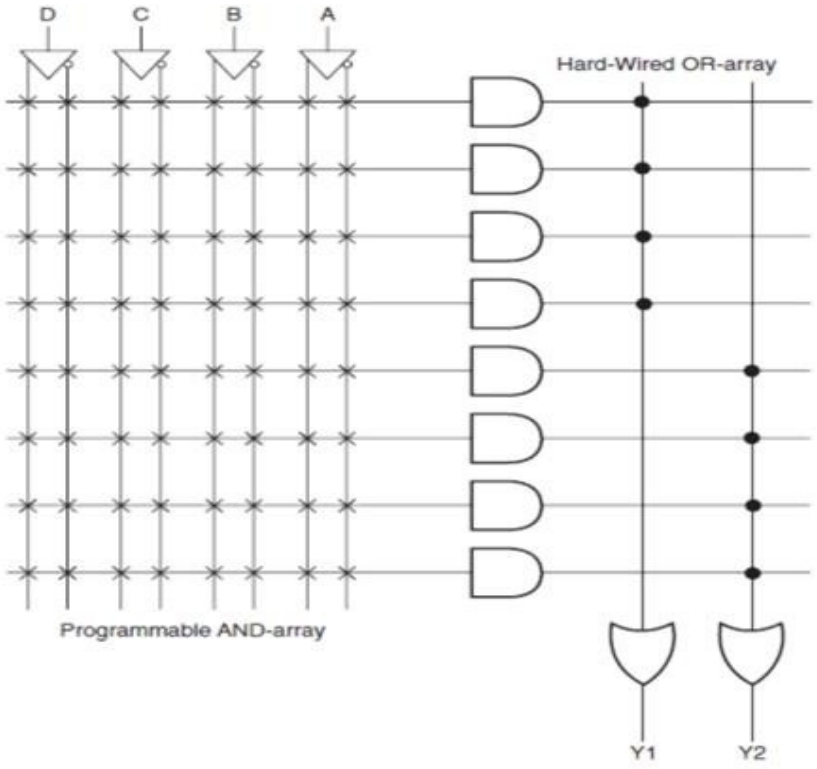
PLA architecture more efficient than a PROM.

DISADVANTAGE

PLA architecture has two sets of programmable fuses due to which PLA devices are difficult to manufacture, program and test.

PROGRAMMABLE ARRAY LOGIC

Programmable array logic (PAL) has a programmable AND array at the input and a fixed OR array at the output. The programmable AND array of a PAL architecture is same as that of the PLA architecture. The number of programmable AND gates in PAL architecture are smaller than the number of minterms. The OR array is fixed and the AND outputs are divided between OR gates.



Memory decoding:

Memory decoding :n The equivalent logic of a binary cell that stores one bit of information is shown below. Read/Write = 0, select = 1, input data to S-R latch Read/Write = 1, select = 1, output data from S-R latch.

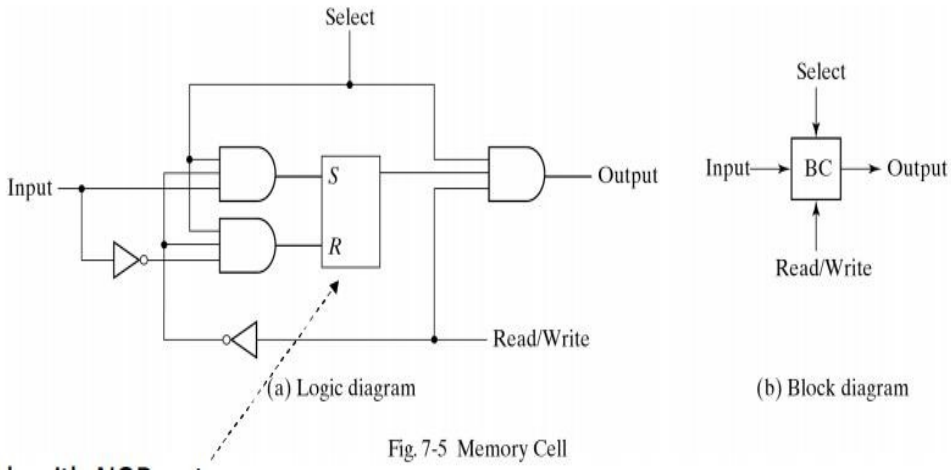


Fig. 7-5 Memory Cell

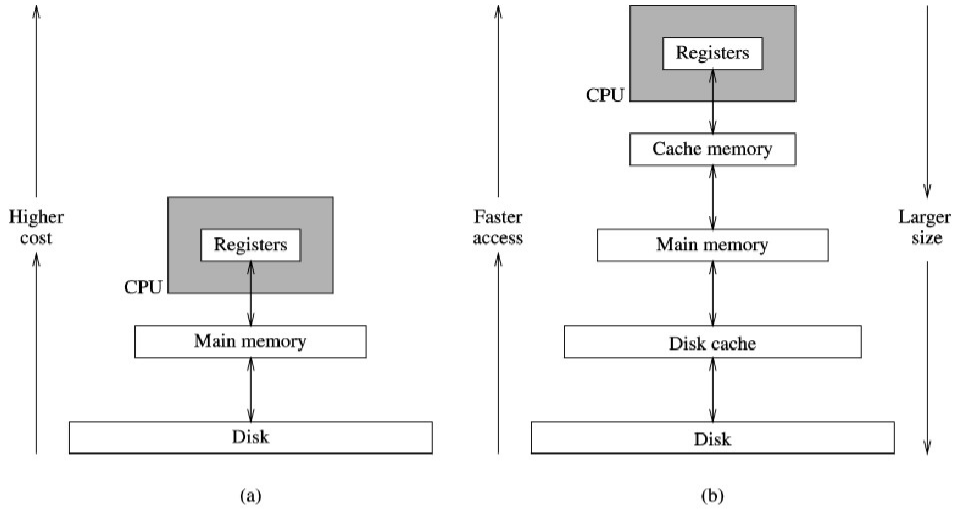
Cache memory:

Cache memory is a small amount of fast memory

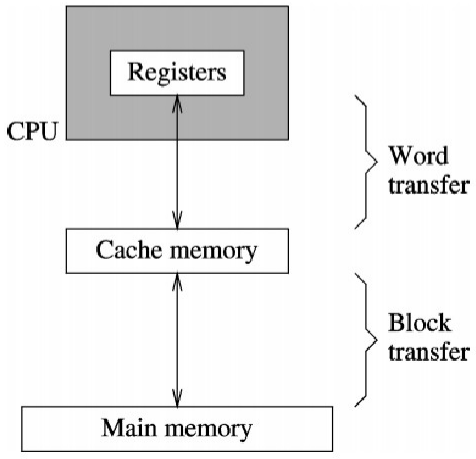
∗Placed between two levels of memory hierarchy » To bridge the gap in access times

– Between processor and main memory (our focus)

– Between main memory and disk (disk cache)
∗Expected to behave like a large amount of fast memory



- Transfer between main memory and cache
  - ∗ In units of blocks
  - ∗ Implements spatial locality
- Transfer between main memory and cache
  - ∗ In units of words
- Need policies for
  - ∗ Block placement
  - ∗ Mapping function
  - ∗ Block replacement
  - ∗ Write policies



- Determines how memory blocks are mapped to cache lines
- Three types
  - Direct mapping
    - » Specifies a single cache line for each memory block
  - Set-associative mapping
    - » Specifies a set of cache lines for each memory block
  - Associative mapping
    - » No restrictions
      - – Any cache line can be used for any memory block

Levels of memory Hierarchy: