UNIT - 3

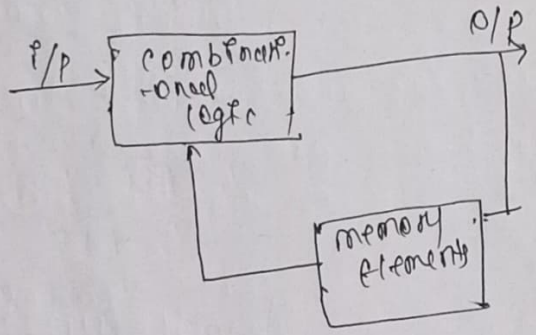
## (1) Sequential logic ckts:

In the case of sequential logic ckt the o/p is the function of present i/p as well as past o/p.

Sequential ckt includes memory element to store the past data.

The flip-flop is basic element of sequential ckt

Using flip flop and combinational logic ckt, any sequential ckt can be designed.




The information stored in memory element at any given time defines the present state and external i/p determine the o/p & the next stage of sequential ckt.

Thus we can specify the sequential ckt of by the time sequence of external i/p

There are two types of sequential ckt:

(1) Synchronous
(2) Asynchronous

### Classification of sequential ckt:

The sequential ckt which are controlled by clock is called synchronous

These ckts are activated when clock signal are present.

The sequential ckt which are not controlled by a clock are called asynchronous sequential ckt.

i.e. the sequential ckts in which the events can take place any time i/p are applied are called asynchronous ckt.

# Comparison b/w synchronous & Asynchronous sequential ckts:

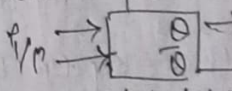
| Synchronous ckt | Vs Asynchronous |
|---|---|
| → In synchronous ckt, the change in i/p signal can affect memory elements upon activation of clock signal. | In asynchronous ckt, change in i/p signal can affect memory element at any instant of time. |
| → memory elements are clocked flip-flop while. | memory elements are either un-clocked flip-flop or time delay |
| ⇒ The max operating speed of clock depend on time delay involved. | since clock is not present, now ckt can operate faster than synchronous. |
| → They are easier to design. | But asynchronous are more difficult to design. |

# flip-flops and latches:

A latch is a sequential device that checks all its i/p continuously. and changes its o/p accordingly at any time independent of clock signal

It refers to non-clocked flip-flops because these flip flop latch on to 1 or 0 immediately upon receiving i/p pulse.

They are not dependent on clock signal for their operation.



General symbol of flip-flop

# flip-flop.

flip-flop is an electronic ckt or device which is used to store a data in binary form.

⇒ flip-flop is an one bit memory device & it can

store either 1 or 0.

flip flop is a sequential device that changes its o/p only when the clock signal is changed.

# Comparison b/w flip flop & latch

| latch | flipflop |
|---|---|
| → latches uses level triggering. | flip-flop use edge triggering |
| → latches are asynchronous i/p | flip-flops are synchronous i/p |
| ⇒ The o/p changes as per i/p till enable is high. | The o/p changes as per i/p on at triggered point. |

# Triggering: Triggering is used to initiate the operation of latches & flip-flop.

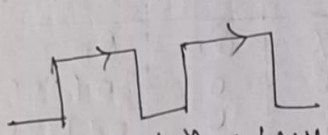
Its main purpose is to synchronise latches or flip-flop.

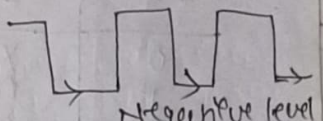
Triggering is classified as:

① level Triggering
② Edge Triggering

① level Triggering:

In level Triggering, i/p signal affect the flip flop only when the clock is at logic 1.



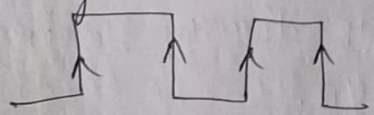



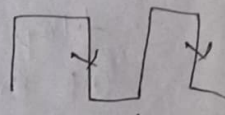

In level triggering, the o/p can change several times in a single clock.

whereas in edge triggering, the o/p will change only once in a single clock.

② Edge Triggering:

In edge triggering, the i/p signal affect the flipflop if they are present in the ⊕ve going & ⊖ve going as of the clock pulse.



# flip-flops

## (1) S-R flip flops:

Truth Table

| Clk | S | R | present state Qn | Next state Qn+1 |
|---|---|---|---|---|
| 0 | X | X | Qn | Qn |
| 1 | 0 | 0 | 0 | 0 (No change) |
| 1 | 0 | 1 | 0 | 0 (Reset) |
| 1 | 1 | 0 | 1 | 1 set |
| 1 | 1 | 1 | X | ⊗ (Don't use this case) |

forbidden condition

0 ↔ 1 changing from JK flip flop

→ The SR flipflop is also known as set-reset flipflop.

→ In SR-flipflop, when the i/p clock or the enable pin is low the o/p of SR-flipflop is disable for any combination of i/p. When the clock pulse is high, the i/p combinations of SR flip flop will directly be propagated at the o/p.

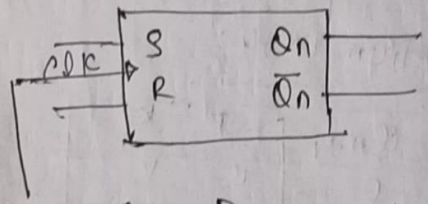
→ In the block diagram of SR-flip-flop, S & R denotes i/p, Qn & $\bar{Q}_n$ represents the complimented & uncomplimented states of flip-flop.

→ Qn+1 denotes the next state.

→ In case of sequential ckt, the o/p at any instant of time depends upon clock signal & present state.

Truth table:

Symbol:




Excitation Table: (Qn, Qn+1 is given)

| Qn | Qn+1 | S | R → To con |
|---|---|---|---|
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | X | 0 |

→ The truth table is invoked when i/p signal is given, Qn & Qn+1 is to be find.

→ Excitation table is invoked when

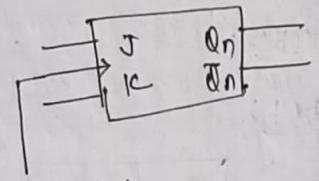
## JK flipflop

Truth table

| present state | | Next state | |
|---|---|---|---|
| clk | J/P | $Q_n$ | $Q_{n+1}$ |
| 0 | X X | $Q_n$ | $Q_n$ |
| 1 | 0 0 | 0 | 0 |
| 1 | 0 1 | 0 | 1 |
| 1 | 1 0 | 1 | $\bar{Q}_n$ |
| 1 | 1 1 | $Q_n$ | $\bar{Q}_n$ |

overcome the forbidden state condition

Symbol:




Excitation Table:

| $Q_n$ | $Q_{n+1}$ | J | K |
|---|---|---|---|
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 0 |

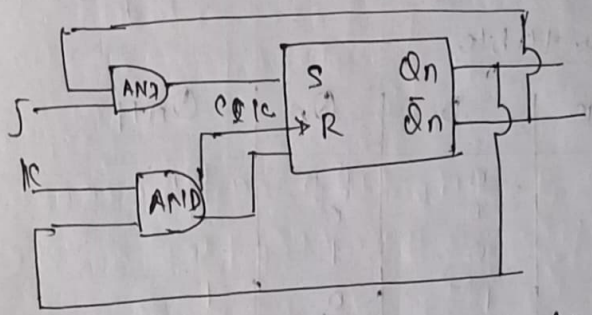
→ A JK flipflop behaves similar to an SR flip flop except for one of the I/P condition.

| clk | J | K | $Q_n$ | $Q_{n+1}$ |
|---|---|---|---|---|
| 1 | 1 | 1 | $Q_n$ | $\bar{Q}_n$ |

i.e., S=R=1 which is not valid in case of SR flipflop.
This condition is valid for JK flop.

→ We can construct a JK flipflop using an SR flip flop & two additional AND gates.

→ construction of JK flip flop using SR flip flop




## # Operation of JK flip flop:

→ I/Ps J & K behave like I/Ps S & R to set & Reset flip flop respectively. When J=K=1 and when clock pulse is 1.
The flipflop O/P toggles i.e, switches to its complemented state. ($\bar{Q}_n$).

→ If $Q_n$ is zero, it switches to 1 ($Q_{n+1}$).

→ Thus, a JK flip flop overcomes the problem of forbidden I/P combination of the SR flip-flop.

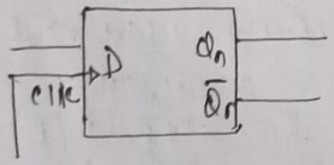
→ NOTE: JK flipflops are known as universal flipflop because it can derive the operating characteristics of all other flipflops using JK flipflop.

# D-flipflops:

Truth table

| clk | D | Qn | Qn+1 |
|---|---|---|---|
| 0 | 0 | Qn | Qn |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Symbol:

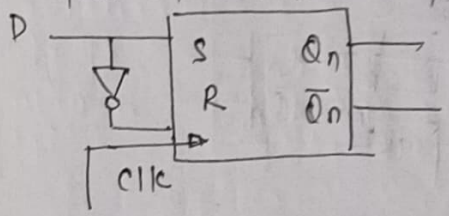



Excitation Table

| clk | Qn | Qn+1 | D |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

→ The logic signal of D-flipflop is shown in fig, ~~the~~ it differs from SR flipflop in the way that it has only one I/P in addition to a clock.

→ It can be constructed from an SR flipflop by placing an inverter between S and R input and assigning the symbol D to S I/P of SR F/F.




# T-flipflop:

Truth Table

| clk | T | Qn | Qn+1 |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

→ General Theory:

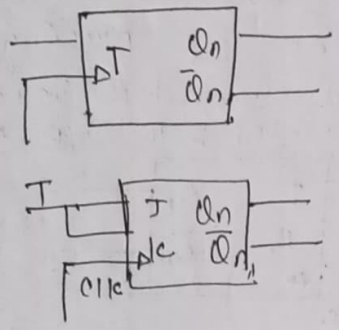
The T-flipflop has a single input as like to D flipflop.

→ T-flip differs from D flipflop in the way that the O/P

of T flip flop toggles when i/p condition is high.

| 1 | 1 | 1 | 0 |
|---|---|---|---|

Symbol:




→ T-flip flop can be constructed using JK flip flop as shown in fig.
When we connect both the i/p of J & K and assign it as one i/p.

Excitation table:

| clk | Qn | Qn+1 | T. |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

# Conversion of flip-flops:

JK flip flop ⟶ SR flip flop

In order to convert one type of flip flop into another type of flip-flop, the following steps must be followed.

Step 1:
Draw the generalise block diagram of the desired i/p & mark the desired i/p of flip flop.

Step 2:
Mark the o/p of desired flip flop in combination with additional combinational ckt connected to the i/p of available flip flop. Then, the o/p of the available flip flop is the o/p of desired flip flop.

Step 3:
Write the characteristic table of the desired flip flop followed by excitation table of given flip flop.

Step-4:

Draw k-map to derive the expression for available flipflops in terms of desired flipflop.

Example:

JK FF ⟷ SR FF → Available flipflop

desired flipflop

① Characteristic table of desired flipflop

② Excitation table of given/available flipflop.

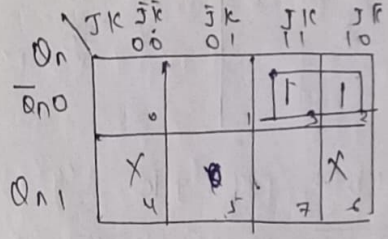
→ Characteristic Table of J-K flipflop

| F/F $Q_n$ | J | K | O/P $Q_{n+1}$ | S | R |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | X |
| 0 | 0 | 1 | 0 | 0 | X |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | X | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |

→ Excitation table of SR flip flop:

| $Q_n$ | $Q_{n+1}$ | S | R |
|---|---|---|---|
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | X | 0 |

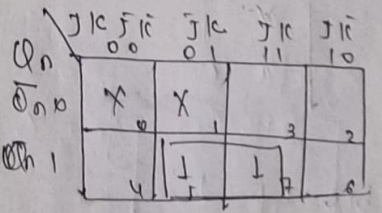
→ k-map for S: (1, 3, 5, 7)




→ 1st preference is pairing of 1 than 1 & X.

$= \bar{Q}_n J$

→ k-map for R




$= Q_n \bar{K}$

0 1
5, 7

② JK → D → Derived variable

Given flip flop

Characteristic table of D flip flop

| $Q_n$ | D | $Q_{n+1}$ | J | K |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | X |
| 0 | 1 | 1 | 1 | X |
| 1 | 0 | 0 | X | 1 |
| 1 | 1 | 1 | X | 0 |

Excitation table of JK

| $Q_n$ | $Q_{n+1}$ | J | K |
|---|---|---|---|
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 0 |

→ Truth table of J:

| $Q_n$ | D | $\bar{D}_0$ | $D_1$ |
|---|---|---|---|
| $\bar{Q}_n$ 0 | 0 | 1 | |
| $Q_n$ 1 | $X_2$ | $X_3$ | |

= D

Truth table of K:

| $Q_n$ | $\bar{D}$ | D |
|---|---|---|
| $\bar{Q}_n$ | X | X |
| $Q_n$ 1 | | |

= $\bar{D}$

→ JK → T

Characteristic table of T flip flop

| $Q_n$ | T | $Q_{n+1}$ | J | K |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | X |
| 0 | 1 | 1 | 1 | X |
| 1 | 1 | 0 | X | 1 |
| 1 | 0 | 1 | X | 0 |

Excitation Table

| $Q_n$ | $Q_{n+1}$ | J | K |
|---|---|---|---|
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 0 |

→ Kmap for J:

| $Q_n$ | T | $\bar{T}$ | T |
|---|---|---|---|
| $\bar{Q}_n$ | | | 1 |
| $Q_n$ | $X_2$ | | $X_3$ |

= T

→ for K:

| $Q_n$ | $\bar{T}$ | T |
|---|---|---|
| $\bar{Q}_n$ | X | X |
| $Q_n$ 1 | | |

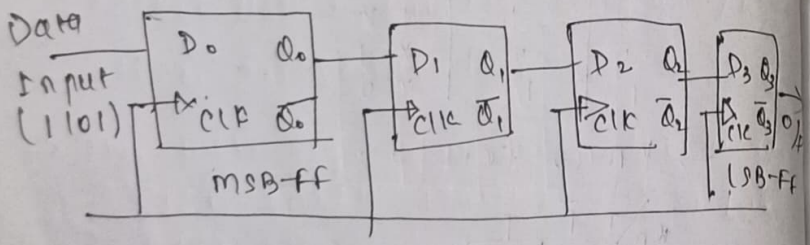
= T

# Register:

→ A register is a digital circuit with two basic functions.
1st → Data storage
2nd → data movement.

→ It is basically a group of flip-flops, logically connected to perform various flip flop.
To store a group of n-bit word the number of flip flop required is n.

→ The simplest possible register is the one that consist of any one flip flop without any external gate as shown in fig.




→ The data can be entered either in SERIAL or in PARALLEL mode.

→ In the serial mode of data entry, 1 bit is added at a time whereas in parallel mode of data entry all the bits are made available simultaneously.

# Shift Register:

A register capable of shifting of binary information entered into it from an external binary source, is called shift register.

→ It is a sequential ckt used to store or shift binary data, either to right (called right shift register) or to left shift.

→ The logical configuration of shift register consist of a chain of flip flop connected in cascade, with the o/p of one flip flop connected to the input of next flip flop.

→ All the flip flops receive a common clock pulse which causes shift from one state to next state.

→ In shift register, each clock pulse shifts the content of register 1 bit position to the right or left

# Classification of shift register:

Shift registers can have a combination of SERIAL or PARALLEL shifted inputs or outputs including

① SERIAL IN, SERIAL OUT (SISO)
② SERIAL IN, PARALLEL OUT (SIPO)
③ PARALLEL IN, SERIAL OUT (PISO)
④ PARALLEL IN, PARALLEL OUT (PIPO)

→ ① SERIAL IN & SERIAL OUT:

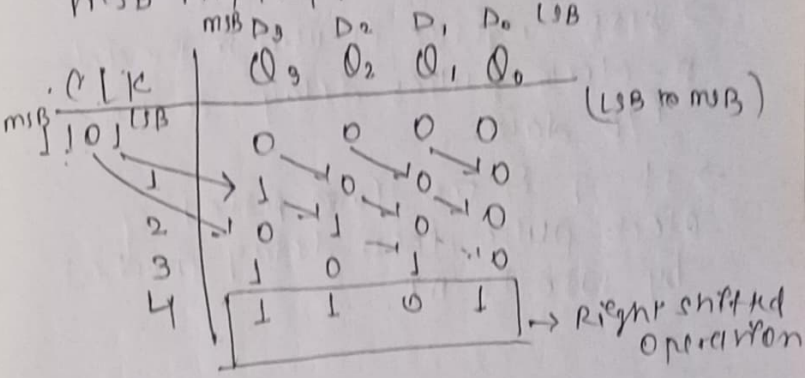
→ The SISO shift register accepts the data serially, one bit at a time on a single I/P line. It produces the stored binary information on its single O/P line in serial form.

→ The prev. fig. represents the 4 bits SISO shift register using D-flip flop.

→ Let the data information 1101 is made available at the I/P.

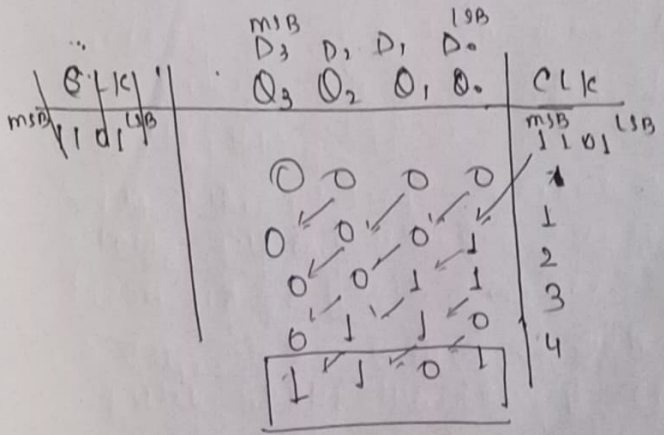
→ Since the shift register is reset so initially all the flipflop O/P are zero.

→ Data 1101 is applied at I/P line. Therefore, in the right shift SISO register, LSB data is applied at the MSB flipflop.




→ Left shift operation of SISO register:

In the left shift SISO register, the data bit is entered from MSB to LSB flipflop.



→ In n-bit register, to enter n-bit data, it requires n-clock pulses in serial form.

→ If n-bit data is stored in SISO register then o/p is taken SERIALLY. For this it requires n-1 pulses.

→ SISO register is used to provide n-clock pulses delay to the i/p data.

→ If T is the time period of clock pulse, then delay provided by SISO is nT.