SWITCHING THEORY AND LOGIC DESIGN

UNIT-4

BY

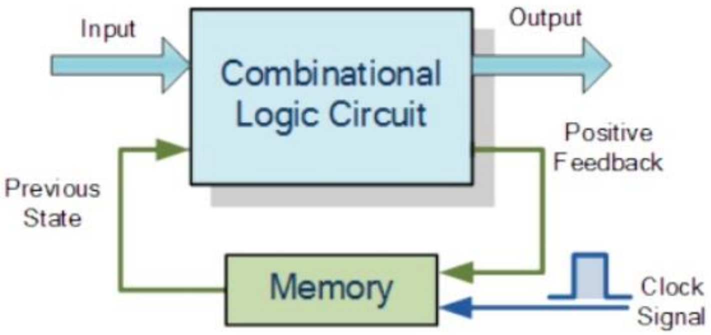
S.SAGAR KRISHNA,

ASSISTANT PROFESSOR,

GAYATRI VIDYA PARISHAD COLLEG OF ENGINEERIN(A)

Sequential Circuits

- Sequential circuit consists of feedback path and several memory elements

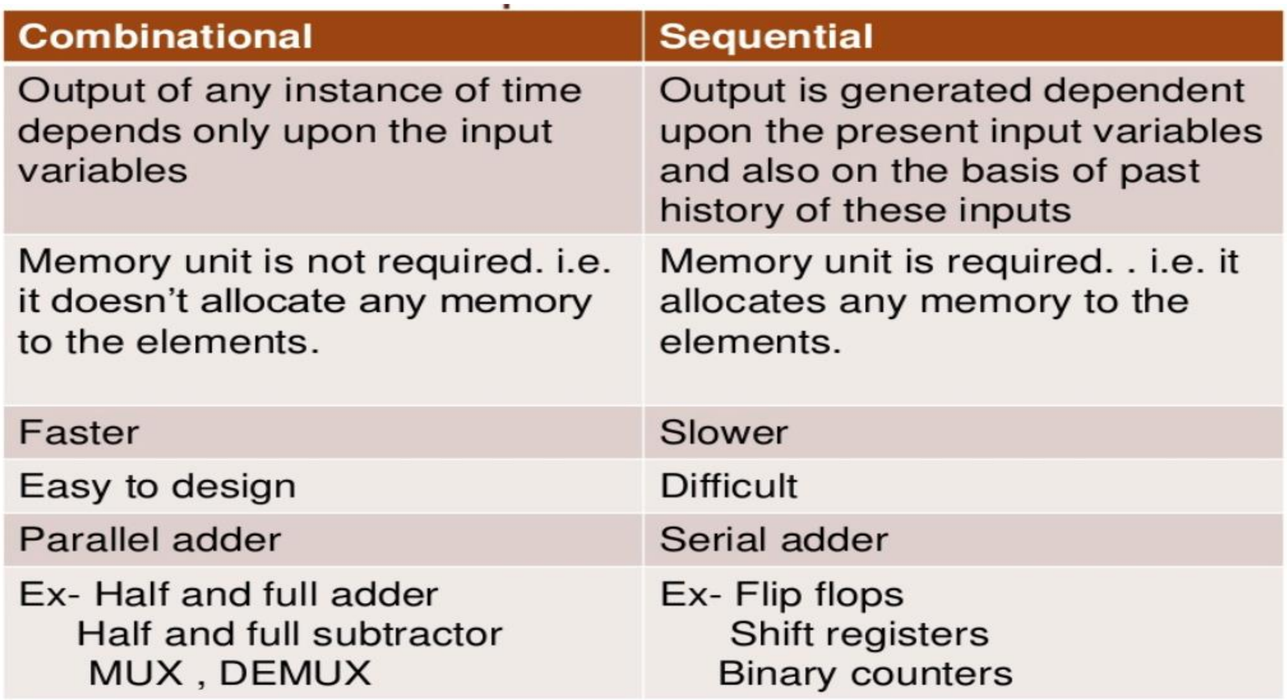


- Sequential circuit = Combinational Logic + Memory Elements

Memory Elements in Sequential Circuits

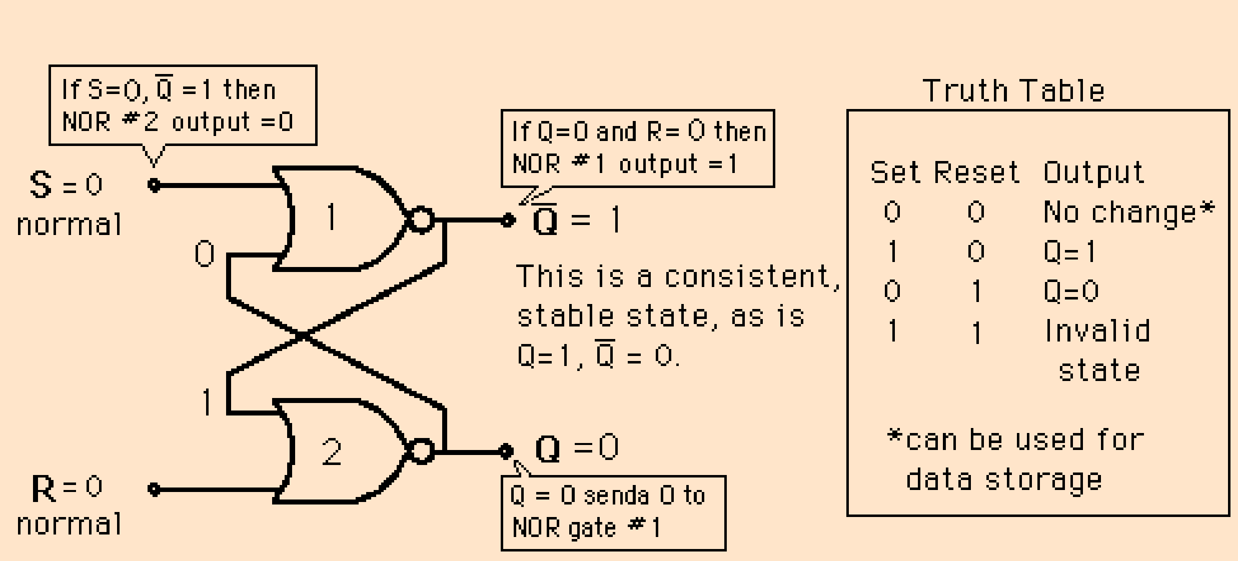
- Memory Element can either be a latch or flip-flop
- Latch output is dependent on input and does not require a clock while flip flop output depends on clock and input.
- One latch or one flip-flop can store 1 bit of information.

Differences between Combinational Circuits and Sequential Circuits



Combinational	Sequential
Output of any instance of time depends only upon the input variables	Output is generated dependent upon the present input variables and also on the basis of past history of these inputs
Memory unit is not required. i.e. it doesn't allocate any memory to the elements.	Memory unit is required. . i.e. it allocates any memory to the elements.
Faster	Slower
Easy to design	Difficult
Parallel adder	Serial adder
Ex- Half and full adder Half and full subtractor MUX , DEMUX	Ex- Flip flops Shift registers Binary counters

SR LATCH USING NOR GATE


If S=0, $\overline{Q}$ =1 then NOR #2 output =0
S = 0
normal
0
1
1
2
If Q=0 and R= 0 then NOR #1 output =1
$\overline{Q}$ = 1
Q =0
Q = 0 senda 0 to NOR gate #1
R = 0
normal

This is a consistent, stable state, as is Q= 1, $\overline{Q}$ = 0.

Truth Table

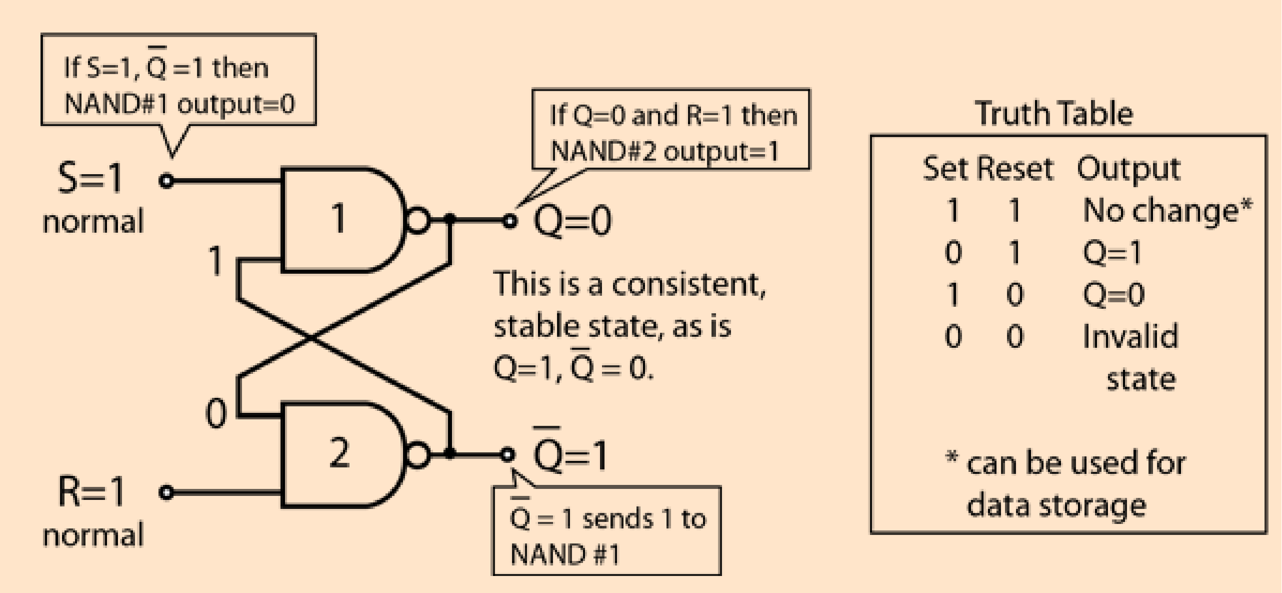
Set	Reset	Output
0	0	No change*
1	0	Q= 1
0	1	Q=0
1	1	Invalid state

*can be used for data storage

SR LATCH USING NOR GATE

- SET=0, RESET=0: has no effect on the output state. Q and Q‘ will remain in whatever state they were prior to the occurrence of this input condition. (No change State)
- SET=0, RESET=1: this will always reset Q=0, where it will remain even after RESET returns to 0
- SET=1, RESET=0: this will always set Q=1, where it will remain even after SET returns to 0.
- SET=1,RESET=1; this condition tries to SET and RESET the latch at the same time, and it produces Q=Q‘=0. here outputs are unpredictable. This input condition should not be used.
- Suppose inputs are applied in this way
  - ➢ S=0,R=0 then Q=0,Q’=1 (No change)
  - ➢ S=0,R=1 then Q=0,Q’=1, (Reset state)
  - ➢ S=1,R=0 then Q=1,Q’=0 (set state)
  - ➢ S=0,R=0 then Q=1,Q’=0(no change)

SR LATCH USING NAND GATE



If S=1, $\overline{Q}$ =1 then NAND#1 output=0

If Q=0 and R=1 then NAND#2 output=1

This is a consistent, stable state, as is Q=1, $\overline{Q}$ = 0.

$\overline{Q}$ = 1 sends 1 to NAND #1

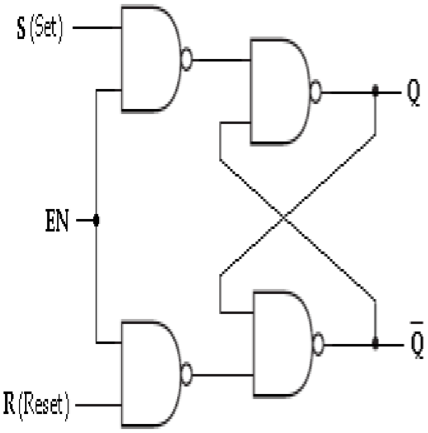
Truth Table

Set	Reset	Output
1	1	No change*
0	1	Q=1
1	0	Q=0
0	0	Invalid state

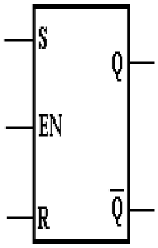
* can be used for data storage

- Here States are reversed

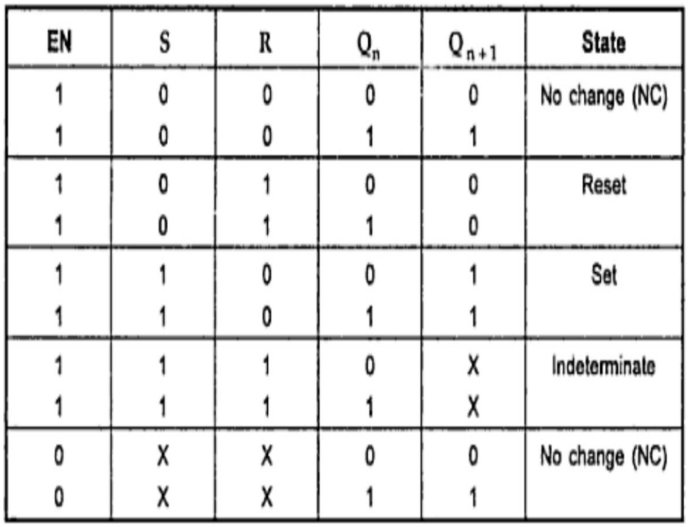
GATED SR LATCH USING NAND GATE



SR Latch with enable input using NAND gates



Logic Symbol



EN	S	R	Q_n	Q_{n+1}	State
1	0	0	0	0	No change (NC)
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	X	Indeterminate
1	1	1	1	X	
0	X	X	0	0	No change (NC)
0	X	X	1	1	

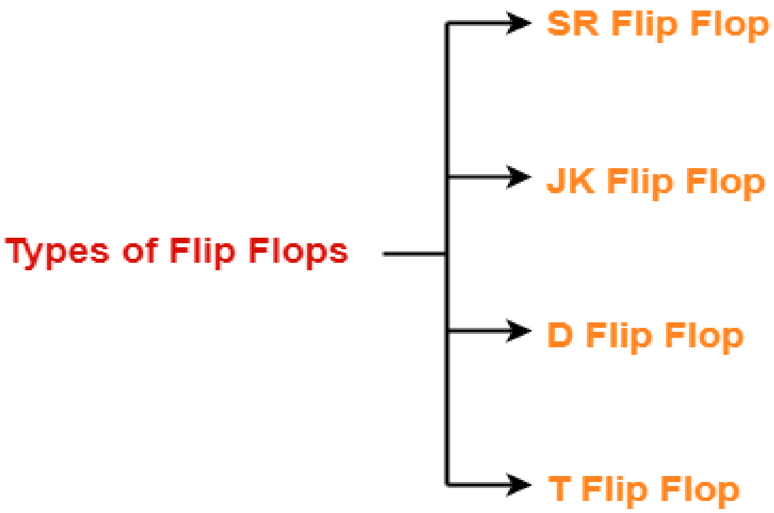
Functional Table

WHEN EN=1 SR LATCH OTHERWISE PREVIOUS STATE IS RETAINED

OTHER LATCHES

- D LATCH
- JK LATCH
- T LATCH
- Working of this will be same as working of flip-flops only difference is latch need not require clock

FLIP-FLOP



Types of Flip Flops

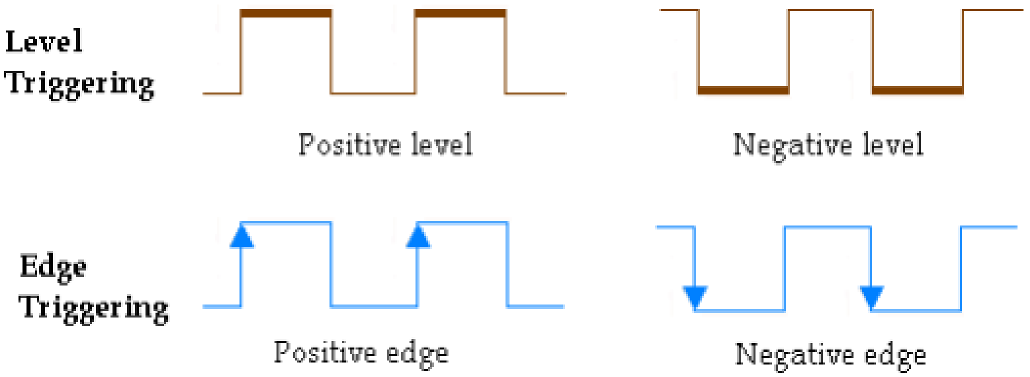
SR Flip Flop

JK Flip Flop

D Flip Flop

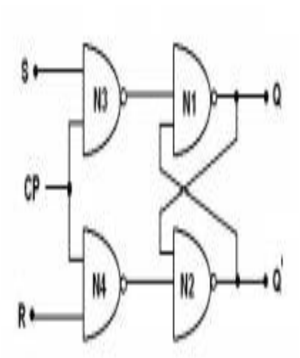
T Flip Flop

TRIGGERING OF FLIP-FLOP

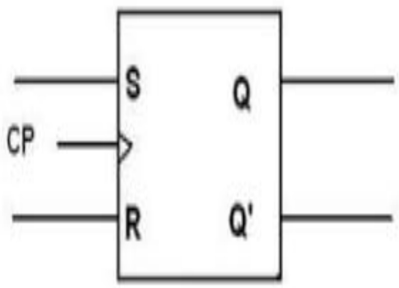


CLOCKS WILL NORMALLY BE EDGE TRIGGERING

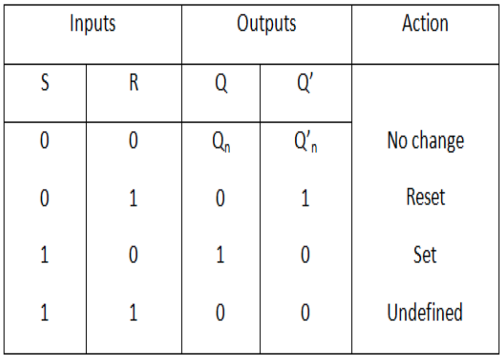
SR FLIP-FLOP



a) Logic diagram



b) Block diagram



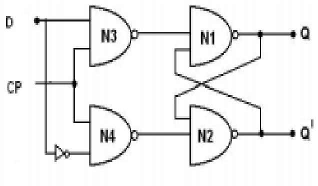
Inputs		Outputs		Action
S	R	Q	Q'	
0	0	Q_n	Q'_n	No change
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Undefined

c) Functional Table

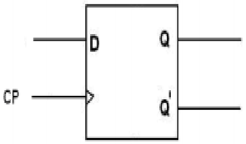
SR FLIP-FLOP

1. When CP=0 the output of N3 and N4 are 1 regardless of the value of S and R. This is given as input to N1 and N2. This makes the previous value of Q and Q‘ unchanged.
2. When CP=1 the information at S and R inputs are allowed to reach the latch and change of state in flip-flop takes place.
3. CP=1, S=1, R=0 gives the SET state i.e., Q=1, Q‘=0.
4. CP=1, S=0, R=1 gives the RESET state i.e., Q=0, Q‘=1.
5. CP=1, S=0, R=0 does not affect the state of flip-flop.
6. CP=1, S=1, R=1 is not allowed, because it is not able to determine the next state. This condition is said to be a race condition.

DFLIP-FLOP

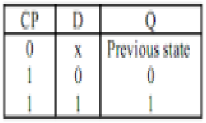


a) Logic diagram



b) Block diagram

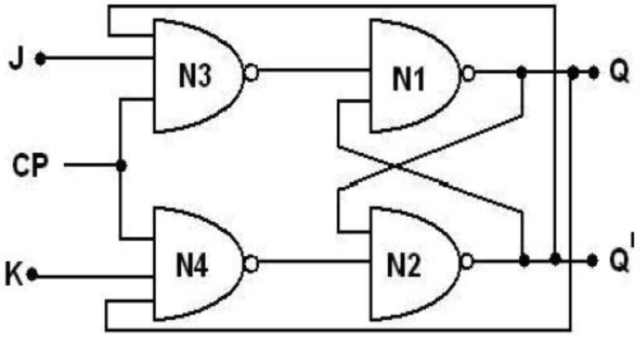
Truth table

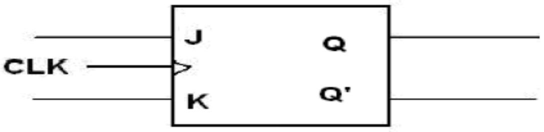


CP	D	Q
0	x	Previous state
1	0	0
1	1	1

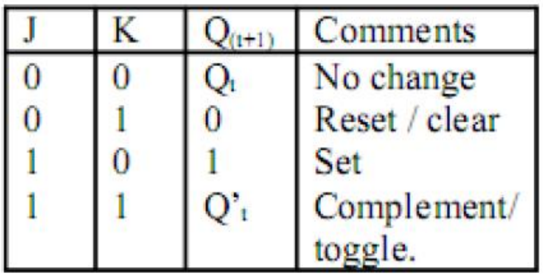
- The D flip-flop is the modified form of R-S flip-flop. S-R flip-flop is converted to D flip-flop by adding an inverter between S and R and only one input D is taken instead of S and R. So one input is D and complement of D is given as another input. The logic diagram and the block diagram of D flip-flop with clocked input
- When the clock is low both the NAND gates (N1 and N2) are disabled and Q retains its last value. When clock is high both the gates are enabled and the input value at D is transferred to its output Q. D flip-flop is also called —Data flip-flop. (Normally D-Flip-Flop is used in the design of Registers)

EDGE TRIGGERED JK FLIP-FLOP





b) Block diagram



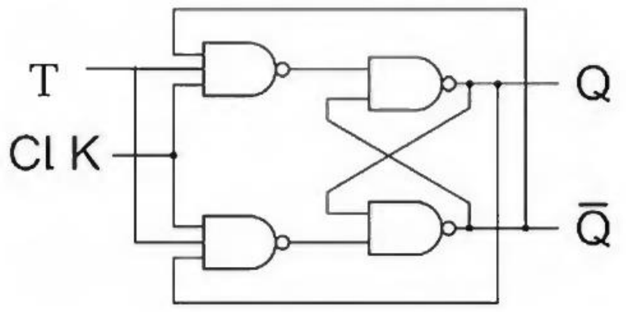
J	K	$Q_{(t+1)}$	Comments
0	0	Q_t	No change
0	1	0	Reset / clear
1	0	1	Set
1	1	Q'_t	Complement/ toggle.

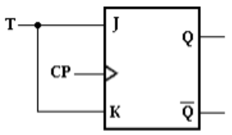
c) Functional Table

EDGE TRIGGERED JK FLIP-FLOP

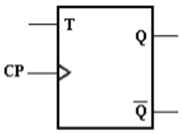
1. When J=0, K=0 then both N3 and N4 will produce high output and the previous value of Q and Q‘ retained as it is(No Change).
2. When J=0, K=1, N3 will get an output as 1 and output of N4 depends on the value of Q. The final output is Q=0, Q‘=1 i.e., reset state
3. When J=1, K=0 the output of N4 is 1 and N3 depends on the value of Q‘. The final output is Q=1 and Q‘=0 i.e., set state
4. When J=1, K=1, If Q=1, Q‘=0 then N4 passes ‘0‘to N2 which produces Q‘=1, Q=0 which is reset state. When J=1, K=1, Q changes to the complement of the last state. The flip-flop is said to be in the toggle state. (This State is normally used in the design of Counters)

EDGE TRIGGERED T FLIP-FLOP



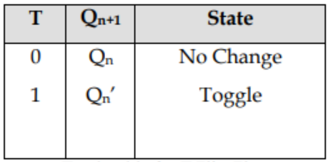


Using JK flipflop



Graphic symbol

It T=0 NS=PS and if T=1 NS=PS'.
NS=NEXT STATE(Q_{N+1})
PS=PREVIOUS STATE(Q_N)

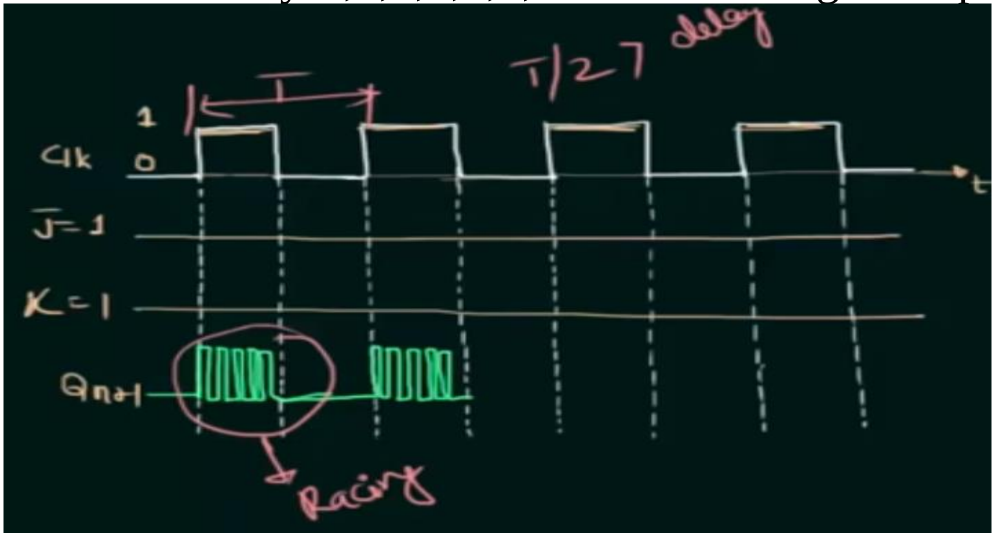


T	Q_{n+1}	State
0	Q_n	No Change
1	Q_n'	Toggle

Truth table for T Flip-Flop

RACE AROUND CONDITION JK FLIP-FLOP

1. In level triggering, if J=1, K=1, the output changes its state continuously 1,0,1,0,1,0,1 without change in input.



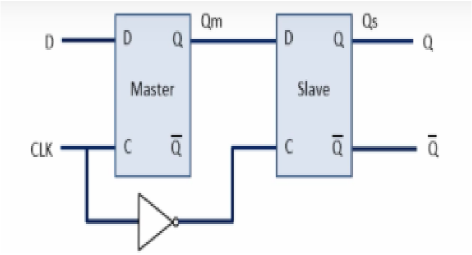
https://www.youtube.com/watch?v=trPGhO7MPnw&itct=CAsQpDAYCSITCLDq0NCn5toCFVTZwQodmDYD3jIGcmVsbWZ1SPD0oZnyg_OXPg%3D%3D&app=desktop

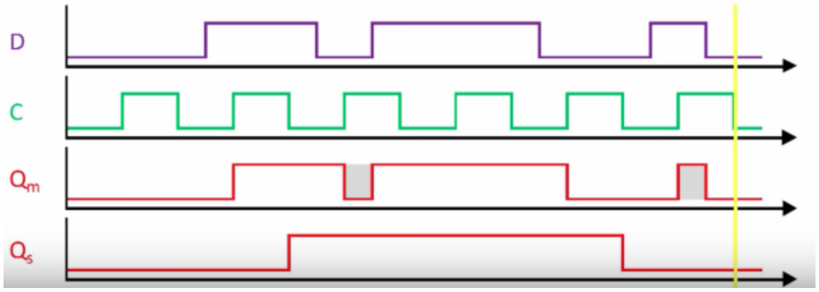
(See this video to understand about it)

To avoid this go for edge triggering

MASTER SLAVE D FLIP-FLOP

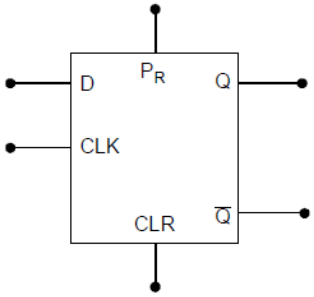
To avoid Race Around Condition we go for Master Slave Flip- Flops. We can use SR or JK or D or T Flip-Flops



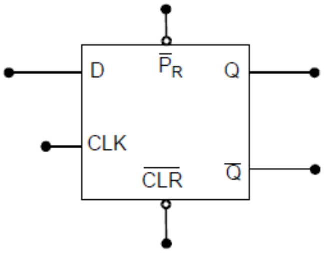


- Master will be active at positive level of the clock Qm=D and Qs=PS(Previous State)
- Slave will be active at the negative level of the clock Qm=PS, and Qs=Qm
- If clk=1 Qm=D but output will still be the previous state(Qs).So here without change in input the output remains same. Hence Race Around Condition is avoided.

D FLIP-FLOP WITH PRESET AND CLEAR



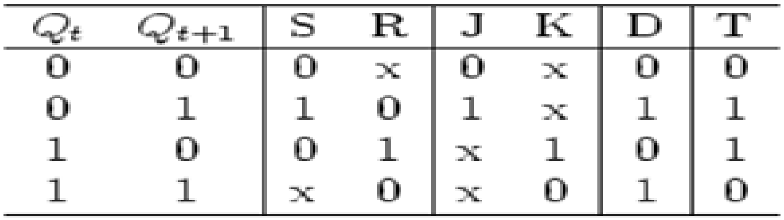
When Preset(Pr)=1 and clear(CLR)=1 it will act as D flip-flop
Otherwise as SR latch



When Preset(Pr)=0 and clear(CLR)=0 it will act as D flip-flop
Otherwise as SR latch

FLIP-FLOP EXCITATION TABLE

Used in conversion of Flip-Flops and design of Synchronous Counter

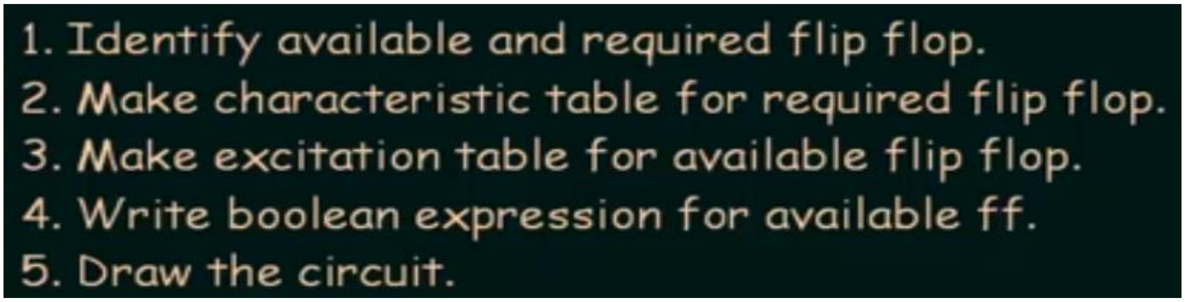


Q_t	Q_{t+1}	S	R	J	K	D	T
0	0	0	x	0	x	0	0
0	1	1	0	1	x	1	1
1	0	0	1	x	1	0	1
1	1	x	0	x	0	1	0

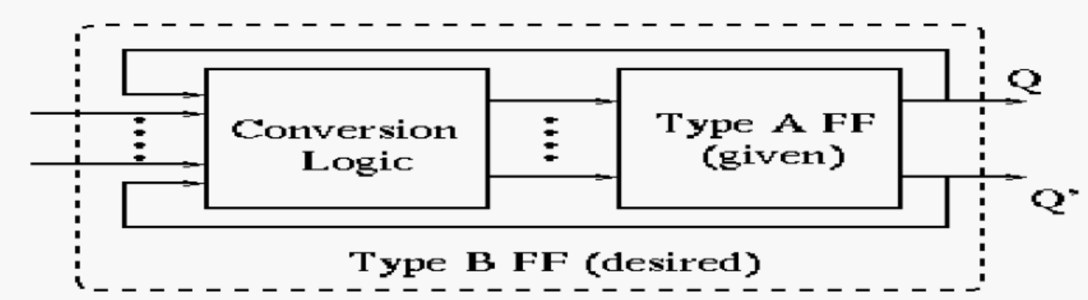
- Depending on output, input is derived
- See this video how excitation table is written for SR Flip-Flop. Same Procedure can be adopted for other flip-flops.
- https://www.youtube.com/watch?v=uiKKRPZbuXA

FLIP-FLOP CONVERSIONS

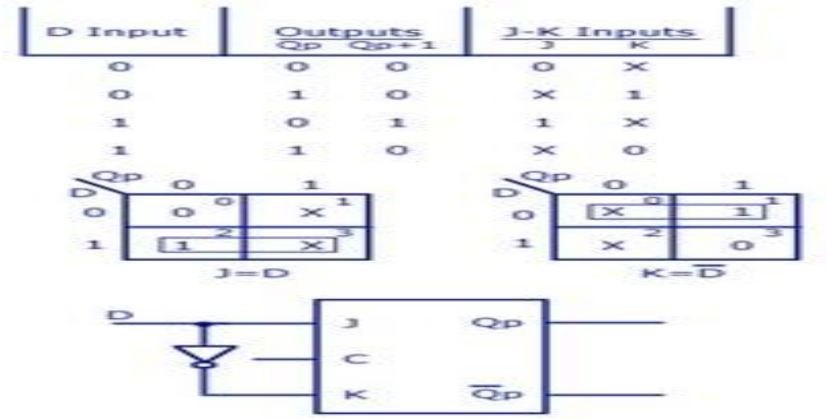
Procedure:



1. Identify available and required flip flop.
2. Make characteristic table for required flip flop.
3. Make excitation table for available flip flop.
4. Write boolean expression for available ff.
5. Draw the circuit.



JK FLIP-FLOP TO D-FLIP-FLOP CONVERSION



D Input	Outputs		J-K Inputs	
	Qp	Qp+1	J	K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	0	X	0

D \ Qp	0	1
0	0^{0}	X^{1}
1	1^{2}	X^{3}

J=D

D \ Qp	0	1
0	X^{0}	1^{1}
1	X^{2}	0^{3}

$K=\overline{D}$

1.Available=JK, Required=D

2.Write characteristic table(functional table) of D flip-flop.

3.Write Excitation table of JK Flip-Flop

4.Write Boolean Expressions for JK Flip-flop

5.Draw the circuit.

https://www.youtube.com/watch?v=ApJ972OYyXQ

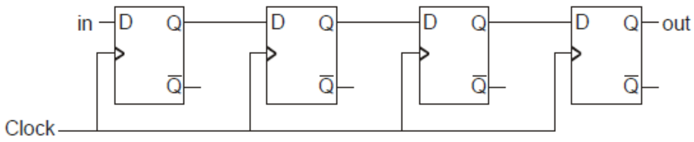
(See this video if procedure is not clear)

SHIFT REGISTERS

Different types of Shift Registers are available

- Serial In - Serial Out Shift Register(SISO)
- Serial In- Parallel Out Shift Register(SIPO)
- Parallel In - Serial Out Shift Register(PISO)
- Parallel In - Parallel Out Shift Register (PIPO)
- Bidirectional Shift Registers for 2 types of Shift
- Universal Shift Register

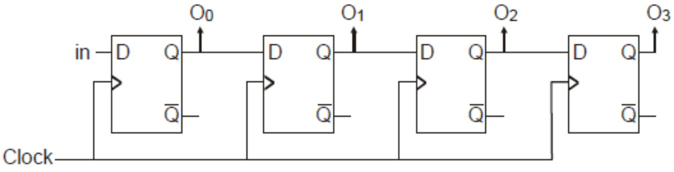
Serial In - Serial Out Shift Register(SISO)



- Serial means single input and serial output means only one output
- The input data is applied sequentially to the D input of the first flip-flop on the left (FF0). During each clock pulse, one bit is transmitted from left to right.
- Assume a data input to be 1. the data input reaches FF3 after 4 clock pulses.

SERIAL IN- PARALLEL OUT SHIFT REGISTERS

- Serial means single input and parallel output means output is taken in parallel
- At each clock, Q_0 =in , Q_1 = PS of Q_0 , Q_2 = PS of Q_1 , Q_3 = PS of Q_2
- After 4 clocks Q_3 =D.



Logic Diagram

No of Clock	Input	Q_0	Q_1	Q_2	Q_3
0	X	0	0	0	0
1	1	1	0	0	0
2	0	0	1	0	0
3	1	1	0	1	0
4	1	1	1	0	1

Functional Table

Similarly other type of Shift Registers

THANK YOU