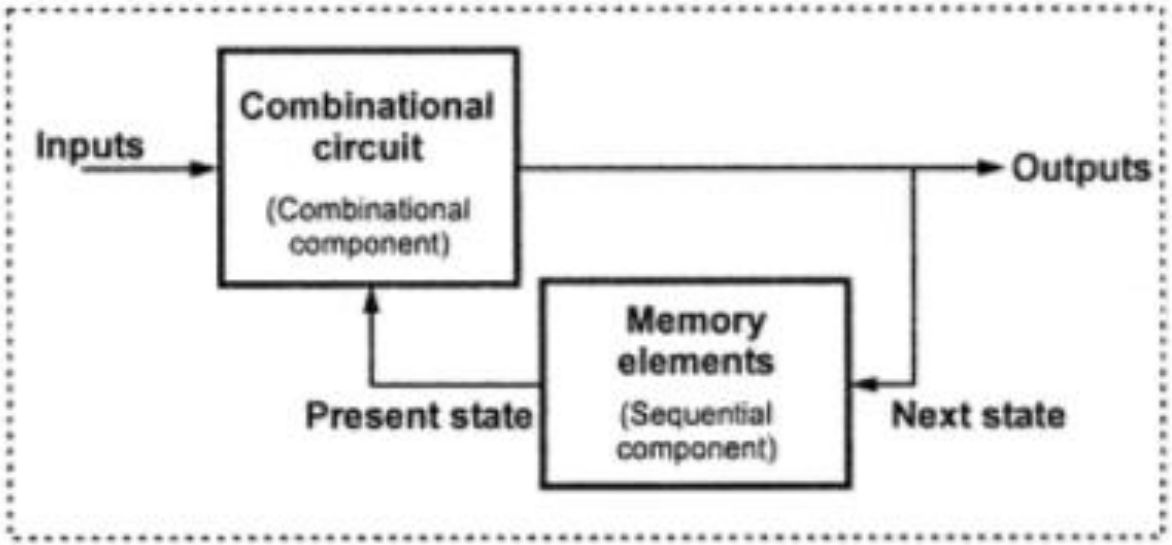
Sequential Circuit

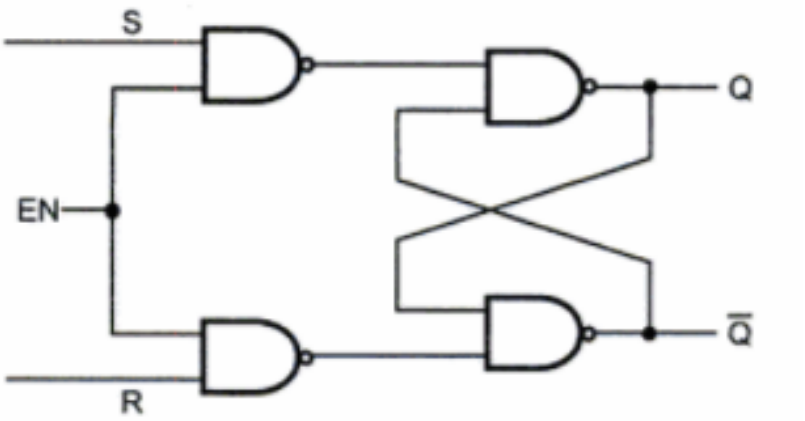
Sr. No.	Combinational circuits	Sequential circuits
1.	In combinational circuits, the output variables are at all times dependent on the combination of input variables.	In sequential circuits, the output variables depend not only on the present input variables but they also depend upon the past history of these input variables.
2.	Memory unit is not required in combinational circuits.	Memory unit is required to store the past history of input variables in the sequential circuit.
3.	Combinational circuits are faster in speed because the delay between input and output is due to propagation delay of gates.	Sequential circuits are slower than the combinational circuits.
4.	Combinational circuits are easy to design.	Sequential circuits are comparatively harder to design.
5.	Parallel adder is a combinational circuit.	Serial adder is a sequential circuit.

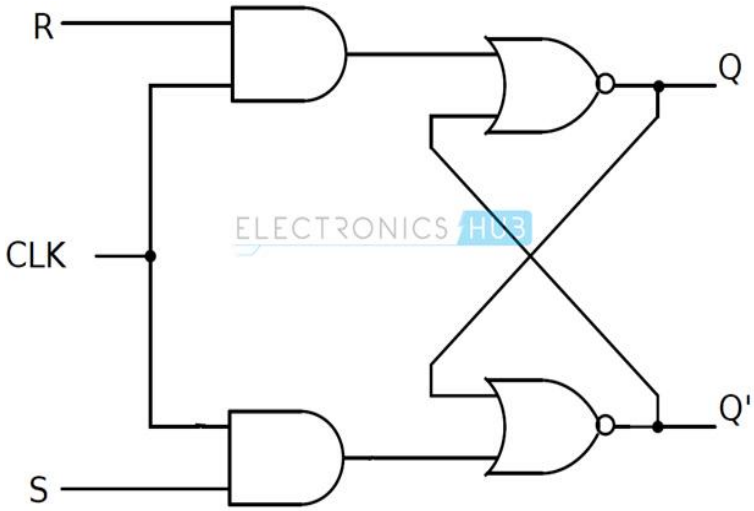
The information stored in the memory elements at any given time defines the present state of the sequential circuit. The present state and the external inputs determine the outputs and the next state of the sequential circuit. Thus we can specify the sequential circuit by a time sequence of external inputs, internal states (present states and next states) and outputs.



Sequential circuit

SR Latch





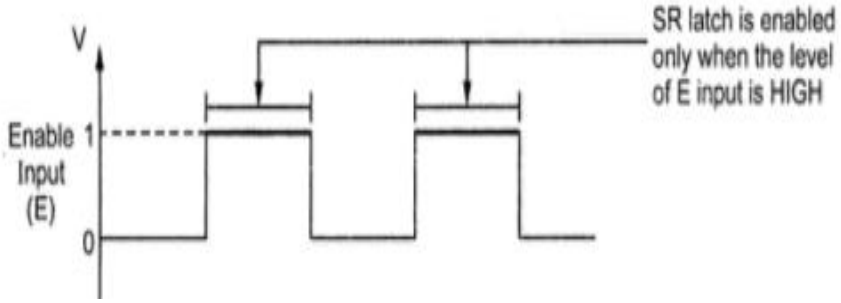
EN	S	R	Q_n	Q_{n+1}	State
1	0	0	0	0	No change (NC)
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	X	Indeterminate
1	1	1	1	X	

EN = CLK

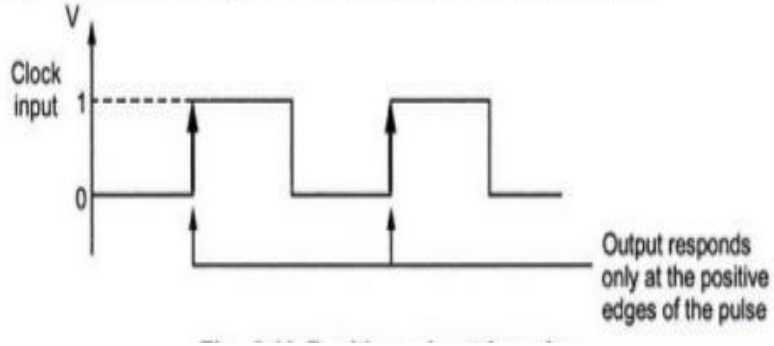
Latches Vs Flip-Flops

Latches and flip-flops are the basic building blocks of the most sequential circuits. The main difference between latches and flip-flops is in the method used for changing their state.

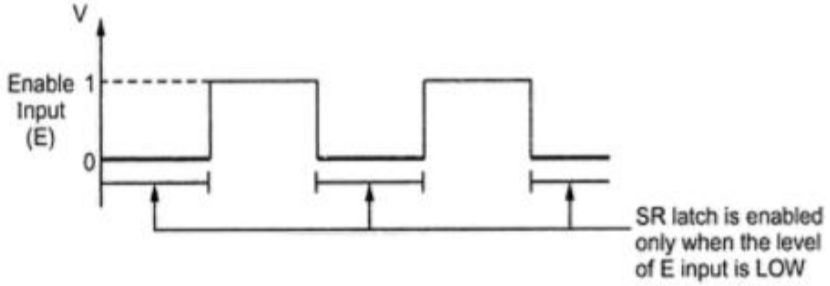
Positive level triggered : The output of latch responds to the input changes only when its enable input is 1 (HIGH).



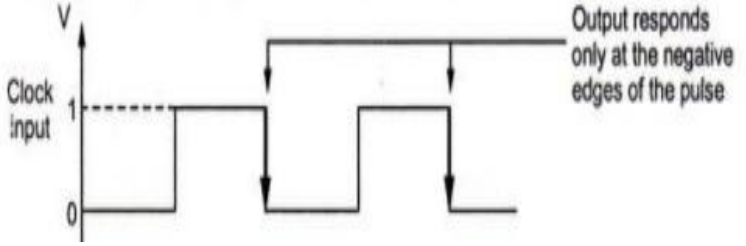
Positive edge triggering : Here, the output responds to the changes in the input only at the positive edge of the clock pulse at the clock input.



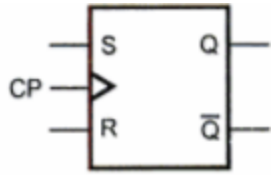
Negative level triggered : The output of latch responds to the input changes only when its enable input is 0 (Low).

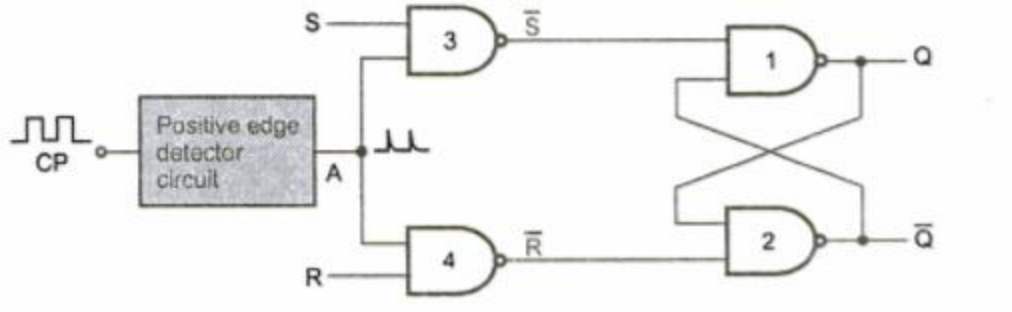


Negative edge triggering : Here, the output responds to the changes in the input only at the negative edge of the clock pulse at the clock input.

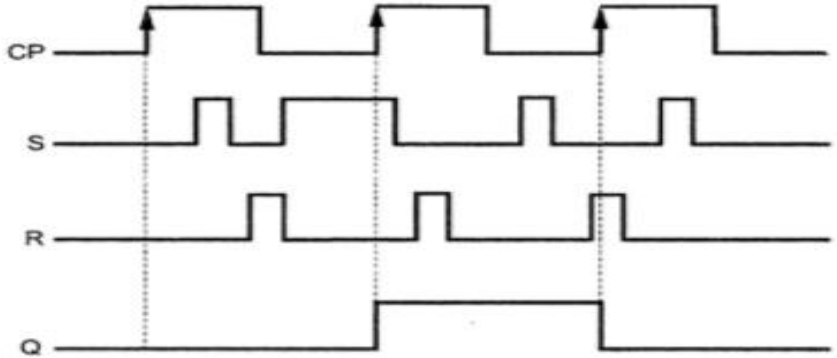


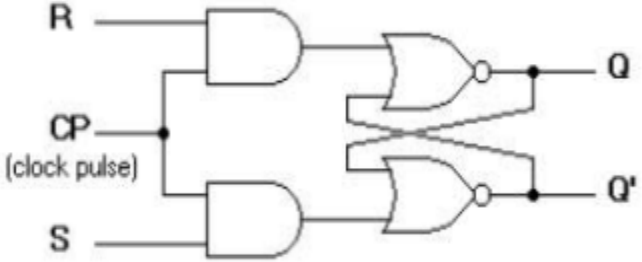
SR flip-flop



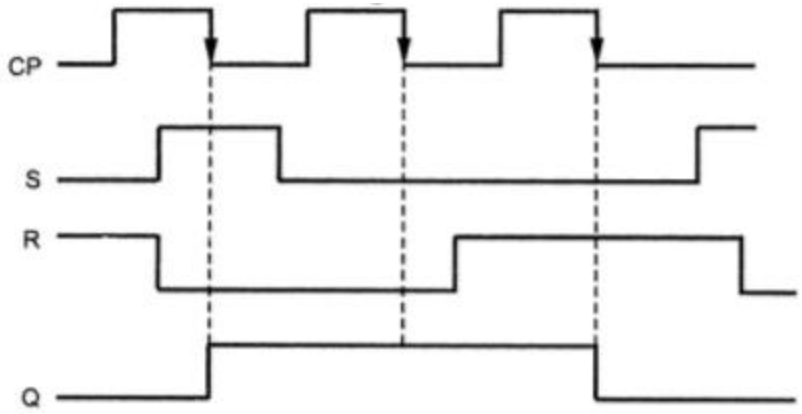


Positive edge triggered SR flip-flop





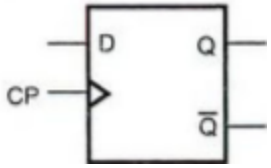
Negative edge triggered SR flip-flop

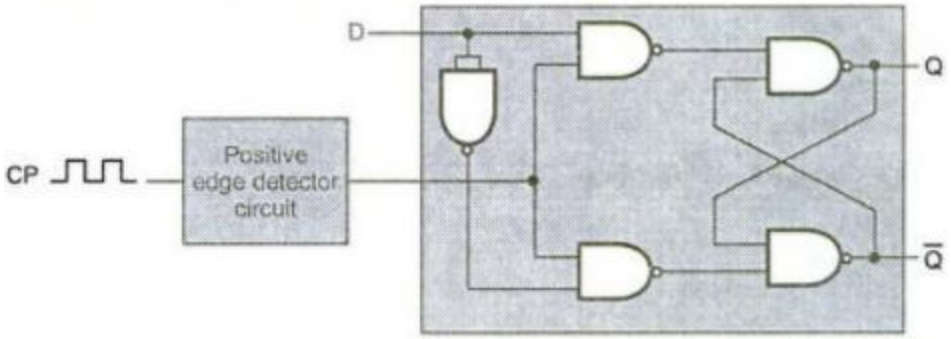


CP	S	R	Q_n	Q_{n+1}	State
↑	0	0	0	0	No change(NC)
↑	0	0	1	1	
↑	0	1	0	0	Reset
↑	0	1	1	0	
↑	1	0	0	1	Set
↑	1	0	1	1	
↑	1	1	0	X	Indeterminate
↑	1	1	1	X	

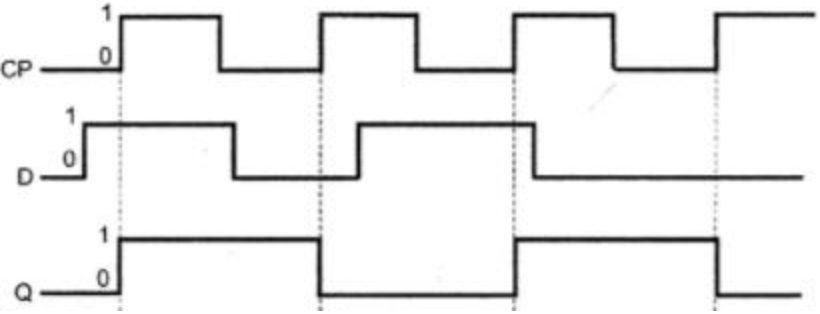
$$Q_{n+1} = S + \overline{R}\,Q_n$$

D Flip-Flop

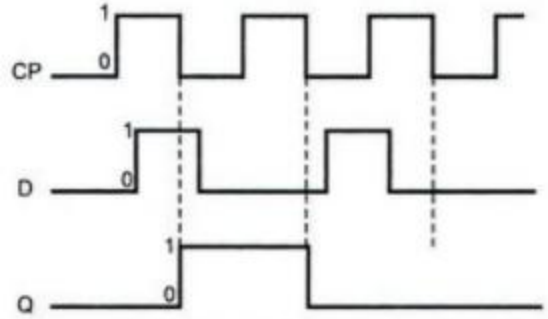




Positive edge triggered D flip-flop



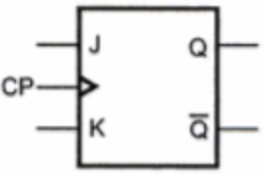
Negative edge triggered D flip-flop

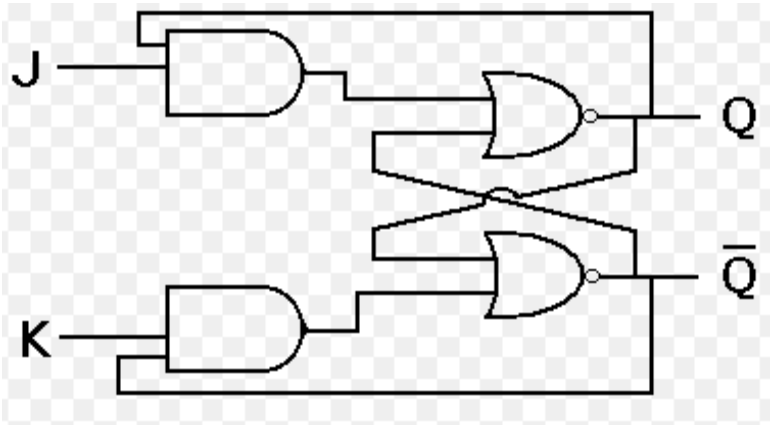


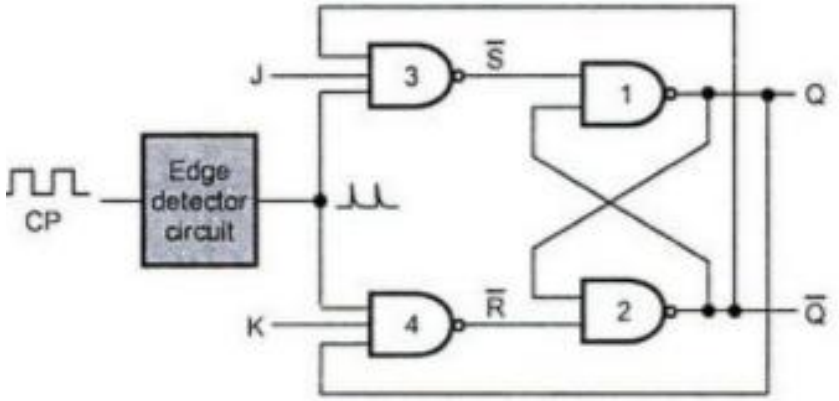
CP	D	Q_{n+1}
↑	0	0
↑	1	1
0	X	Q_n

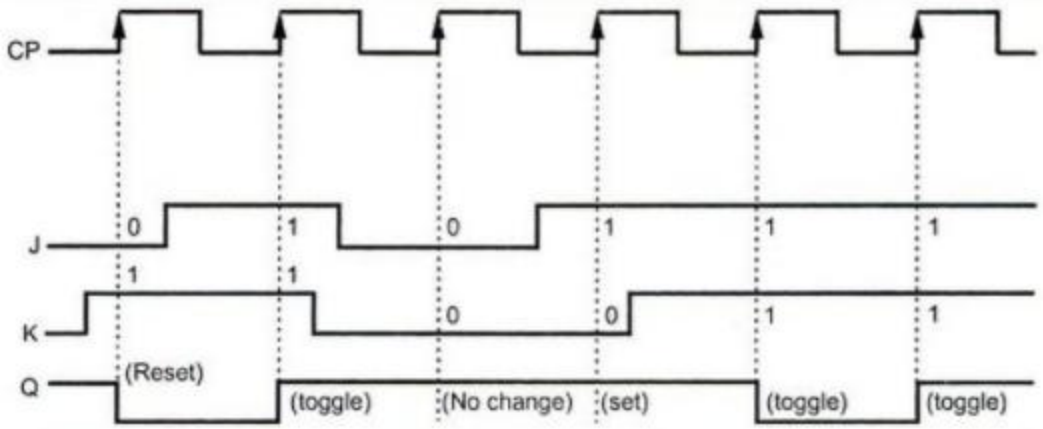
$$Q_{n+1} = D$$

JK Flip-Flop









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

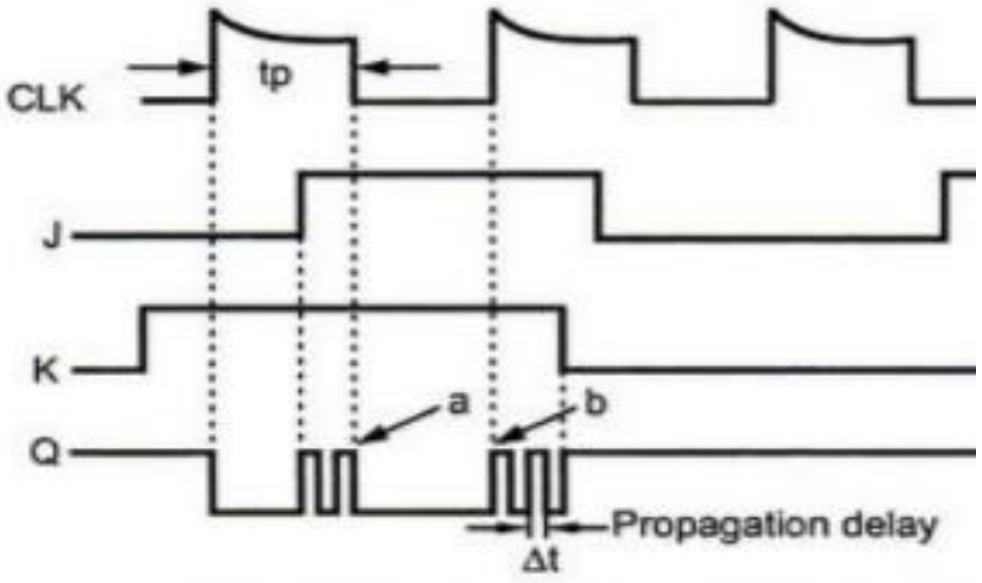
≡

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

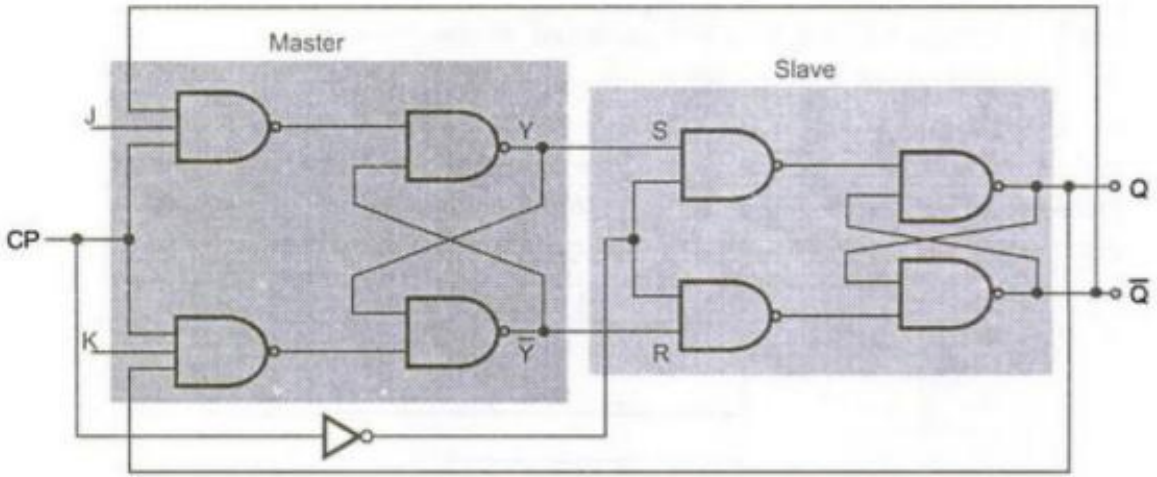
$$Q_{n+1} = J\,\overline{Q}_n + \overline{K}Q_n$$

Race-Around Condition

In JK flip-flop, when J = K = 1, the output toggles (output changes either from 0 to 1 or from 1 to 0). Consider that initially Q = 0 and J = K = 1. After a time interval Δt equal to the propagation delay through two NAND gates in series, the output will change to Q = 1 and after another time interval of Δt the output will change back to Q = 0. This toggling will continue until the flip-flop is enabled and J = K = 1. At the end of clock pulse the flip-flop is disabled and the value of Q is uncertain. This situation is referred to as the **race-around condition.** This is illustrated in Fig. 6.28. This condition exists when $t_p \geq \Delta t$. Thus by keeping $t_p < \Delta t$ we can avoid race around condition.

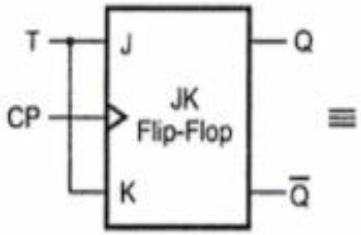


Master-Slave JK Flip-Flop

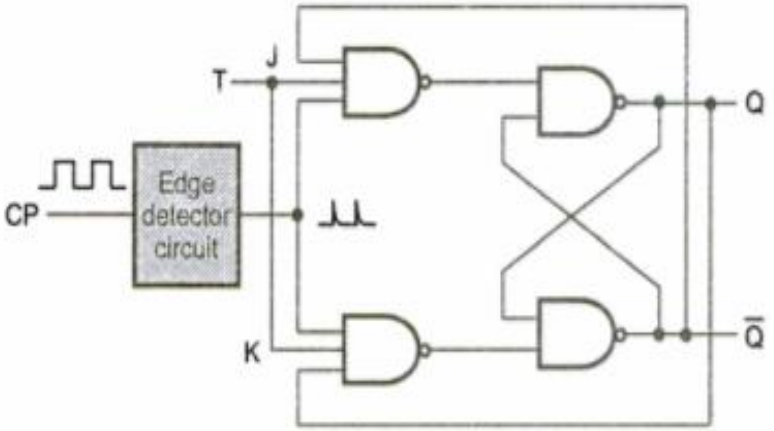


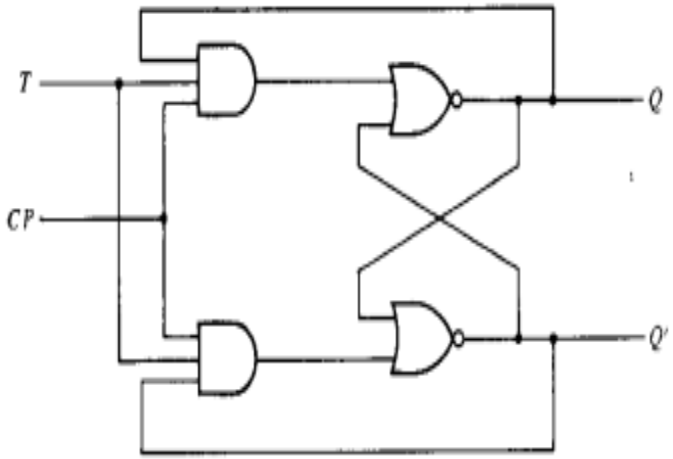
When J = 1 and K = 1, master toggles on the positive clock and slave then copies the output of master on the negative clock. At this instant, feedback inputs to the master flip-flop are complemented but as it is negative half of the clock pulse master flip-flop is inactive. This prevents race around condition.

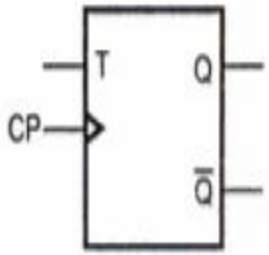
T Flip-Flop



≡



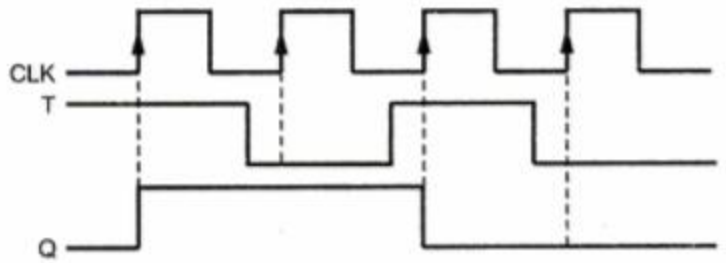




Q_n	T	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

≡

T	Q_{n+1}
0	Q_n
1	$\overline{Q_n}$



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

Excitation table

SR Flip-flop			
Q(t)	Q(t+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

D Flip-flop		
Q(t)	Q(t+1)	DR
0	0	0
0	1	1
1	0	0
1	1	1

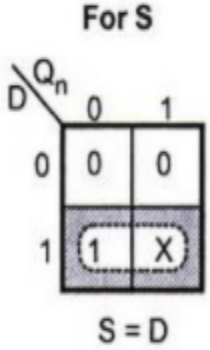
JK flip-flop			
Q(t)	Q(t+1)	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

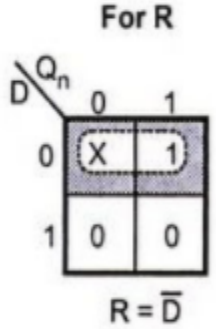
T flip-flop		
Q(t)	Q(t+1)	DR
0	0	0
0	1	1
1	0	1
1	1	0

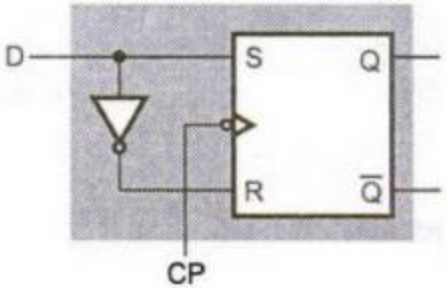
Conversion of Flip-Flops

SR Flip-Flop to D Flip-Flop

Input	Present state	Next state	Flip-flop inputs	
D	Q_n	Q_{n+1}	S	R
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0







SR Flip-Flop to JK Flip-Flop

Inputs		Present state	Next state	Flip-flop inputs	
J	K	Q_n	Q_{n+1}	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

For S

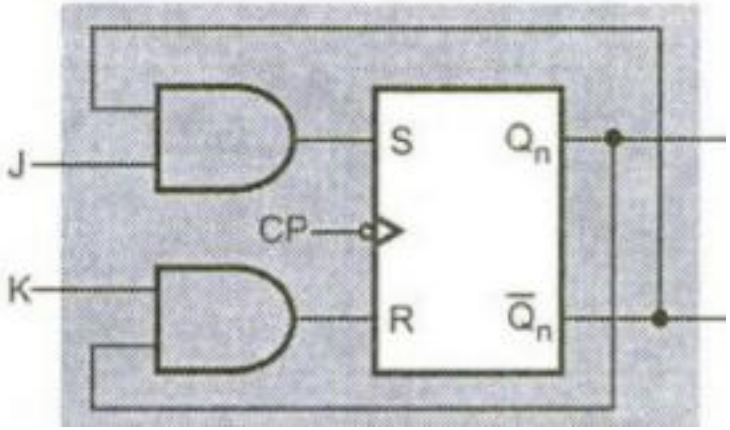
J \ K Q_n	00	01	11	10
0	0	X	0	0
1	1	X	0	1

$S = J\overline{Q}_n$

For R

J \ K Q_n	00	01	11	10
0	X	0	1	X
1	0	0	1	0

$R = K Q_n$



SR Flip-flop to T Flip-flop

Input	Present state	Next state	Flip-flop inputs	
T	Q_n	$Q_n + 1$	S	R
0	0	0	0	X
0	1	1	X	0
1	0	1	1	0
1	1	0	0	1

For S

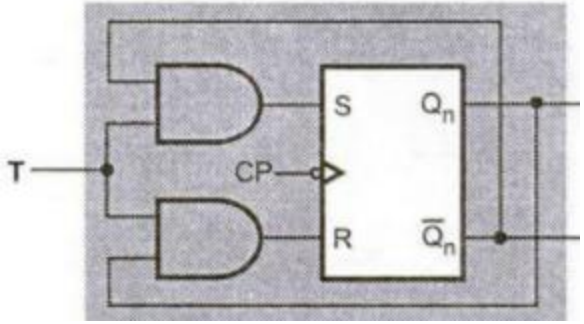
T \ Q_n	0	1
0	0	X
1	1	0

$S = T\overline{Q}_n$

For R

T \ Q_n	0	1
0	X	0
1	0	1

$R = TQ_n$



JK Flip-Flop to T Flip-Flop

Input	Present state	Next state	Flip-flop inputs	
T	Q_n	Q_{n+1}	J_A	K_A
0	0	0	0	X
0	1	1	X	0
1	0	1	1	X
1	1	0	X	1

For J_A

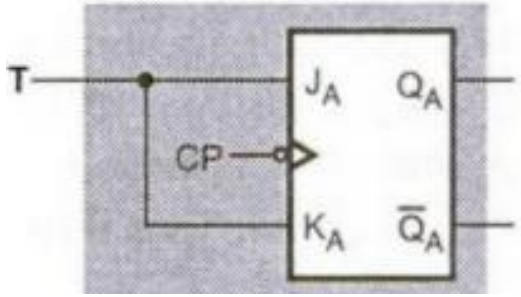
T \ Q_n	0	1
0	0	X
1	1	X

$J_A = T$

For K_A

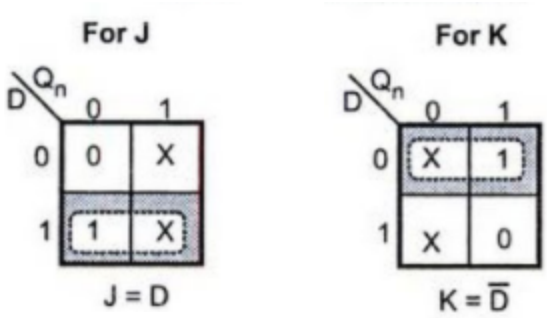
T \ Q_n	0	1
0	X	0
1	X	1

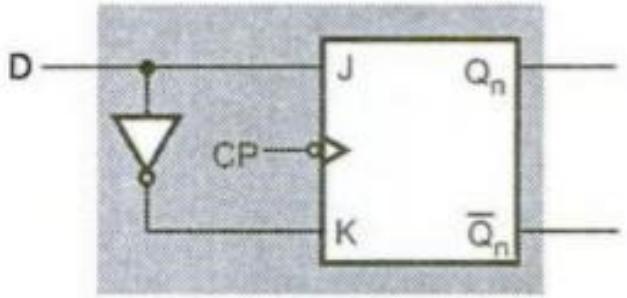
$K_A = T$



JK Flip-Flop to D Flip-Flop

Input	Present state	Next state	Flip-flop inputs	
D	Q_n	Q_{n+1}	J	K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	1	X	0





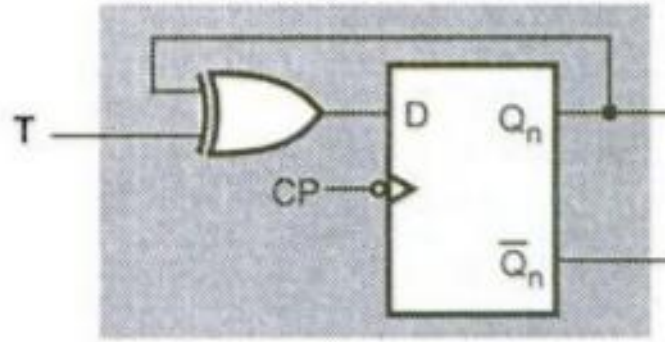
D Flip-Flop to T Flip-Flop

Input	Present state	Next state	Flip-flop input
T	Q_n	Q_{n+1}	D
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

For D

T \ Q_n	0	1
0	0	1
1	1	0

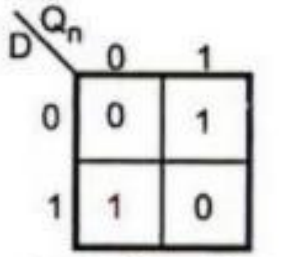
$$D = \bar{T}Q_n + T\bar{Q}_n$$
$$= T \oplus Q_n$$



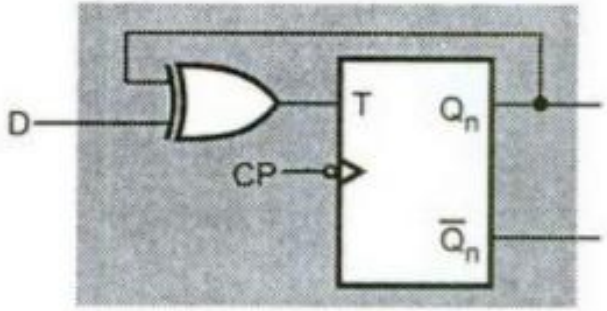
T Flip-Flop to D Flip-Flop

Input	Present state	Next state	Flip-flop input
D	Q_n	Q_{n+1}	T
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

For T

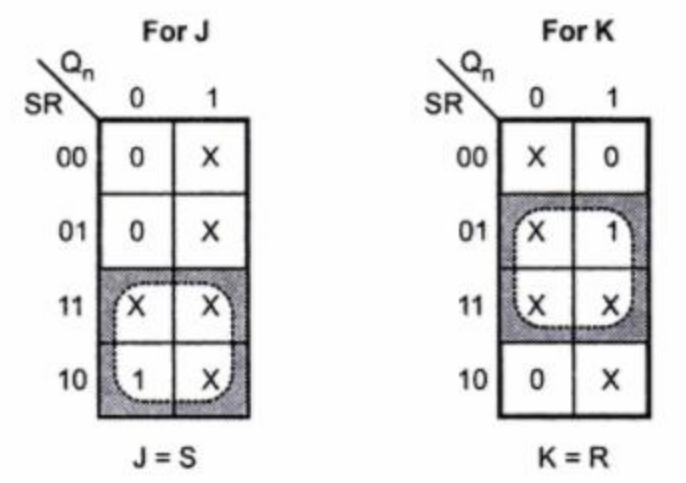


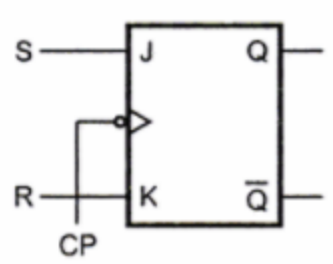
$T = D\,\overline{Q}_n + \overline{D}\,Q_n$



JK Flip-Flop to SR Flip-Flop

Inputs		Present state Q_n	Next state Q_{n+1}	Flip-flop inputs	
S	R			J	K
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	X	1
1	0	0	1	1	X
1	0	1	1	X	0
1	1	0	X	X	X
1	1	1	X	X	X

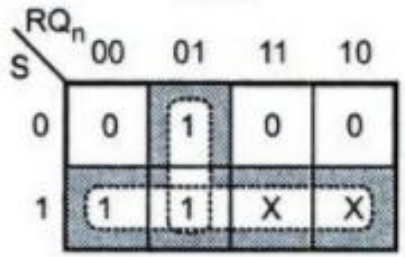




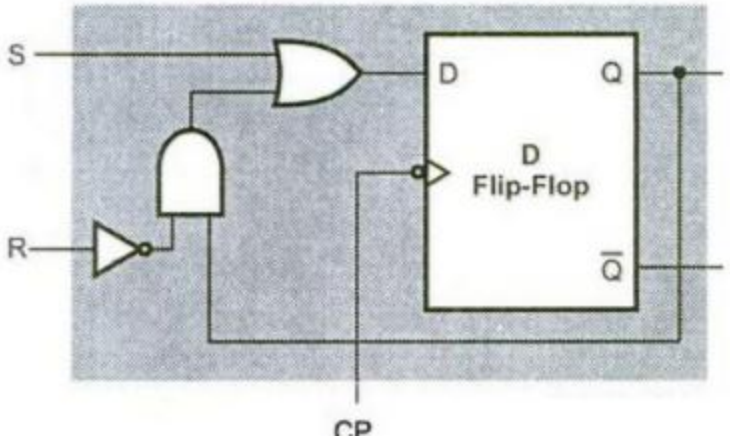
D Flip-Flop to SR Flip-Flop

Inputs		Present state	Next state	Flip-flop input
S	R	Q_n	Q_{n+1}	D
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	X	X
1	1	1	X	X

For D

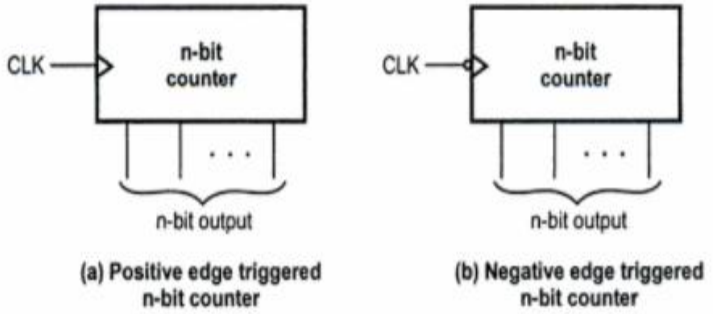


$D = \overline{R}\,Q_n + S$



Counter

A group of flip-flops connected together forms a **register.** A register is used solely for storing and shifting data which is in the form of 1s and/or 0s, entered from an external source. It has no specific sequence of states except in certain very specialized applications. A **counter** is a register capable of counting the number of clock pulses arriving at its clock input. Count represents the number of clock pulses arrived. On arrival of each clock pulse, the counter is incremented by one. In case of down counter, it is decremented by one.



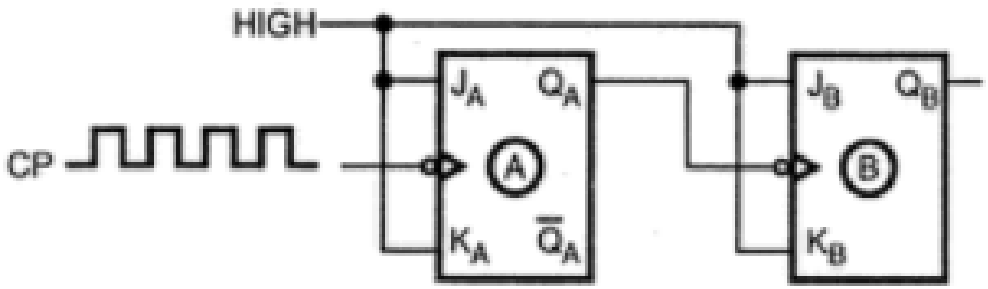
(a) Positive edge triggered n-bit counter

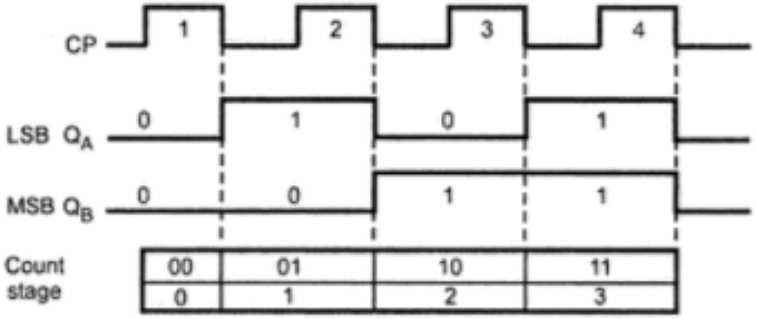
(b) Negative edge triggered n-bit counter

Sr. No.	Asynchronous Counters	Synchronous Counters
1.	In this type of counter flip-flops are connected in such a way that output of first flip-flop drives the clock for the next flip-flop.	In this type there is no connection between output of first flip-flop and clock input of the next flip-flop.
2.	All the flip-flops are not clocked simultaneously.	All the flip-flops are clocked simultaneously.
3.	Logic circuit is very simple even for more number of states.	Design involves complex logic circuit as number of states increases.
4.	Main drawback of these counters is their low speed as the clock is propagated through number of flip-flops before it reaches last flip-flop.	As clock is simultaneously given to all flip-flops there is no problem of propagation delay. Hence they are high speed counters and are preferred when number of flip-flops increases in the given design.

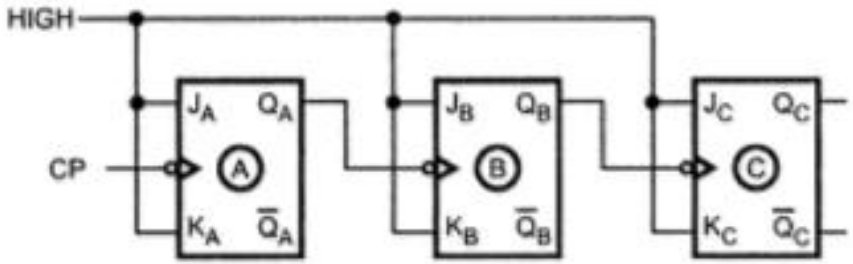
Asynchronous Counters

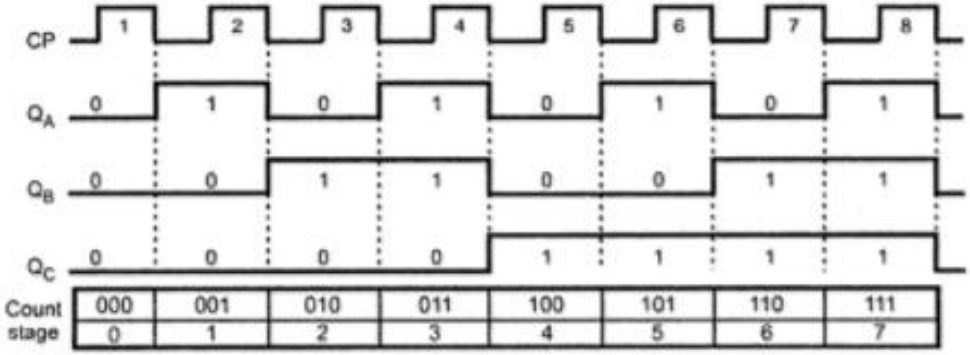
two-bit asynchronous binary counter



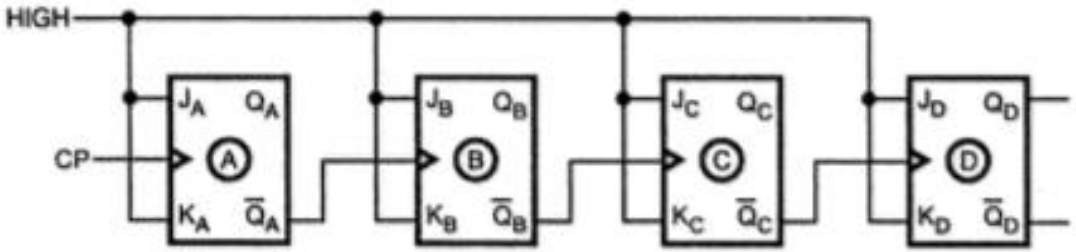


3-bit asynchronous counter





4-stage positive edge triggered ripple counter



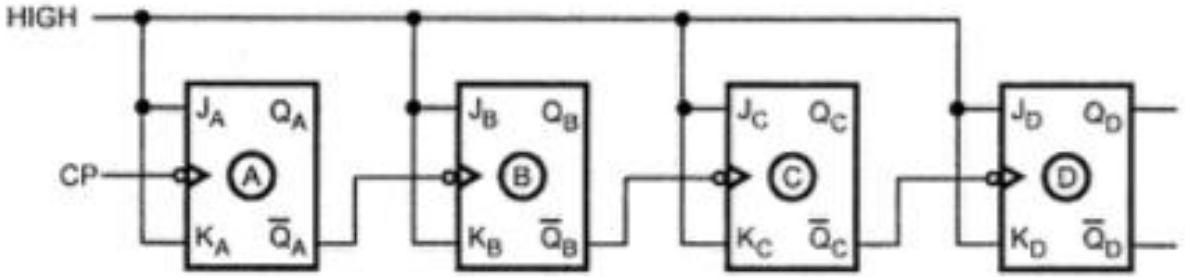
Frequency at output $Q_A = \frac{F_{CLK}}{2}$

Frequency at output $Q_B = \frac{Q_A}{2} = \frac{F_{CLK}}{4}$

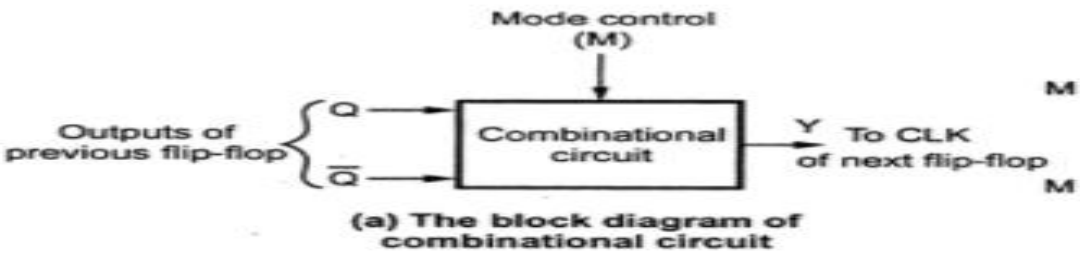
Frequency at output $Q_C = \frac{Q_B}{2} = \frac{Q_A}{4} = \frac{F_{CLK}}{8}$

Frequency at output $Q_D = \frac{Q_C}{2} = \frac{Q_B}{4} = \frac{Q_A}{8} = \frac{F_{CLK}}{16}$

4-bit asynchronous down counter



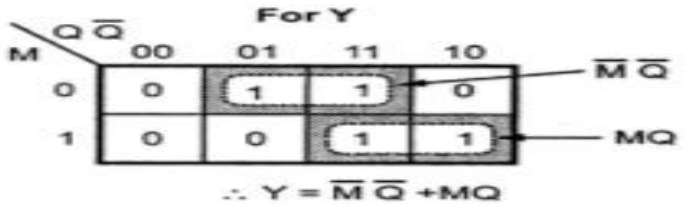
Asynchronous Up/Down Counter

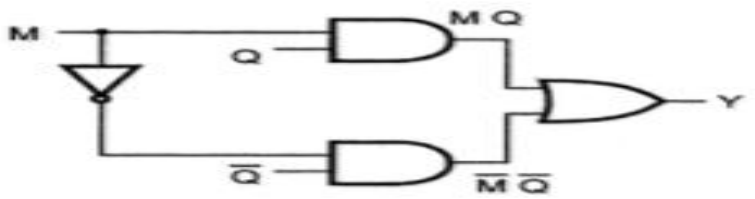


(a) The block diagram of combinational circuit

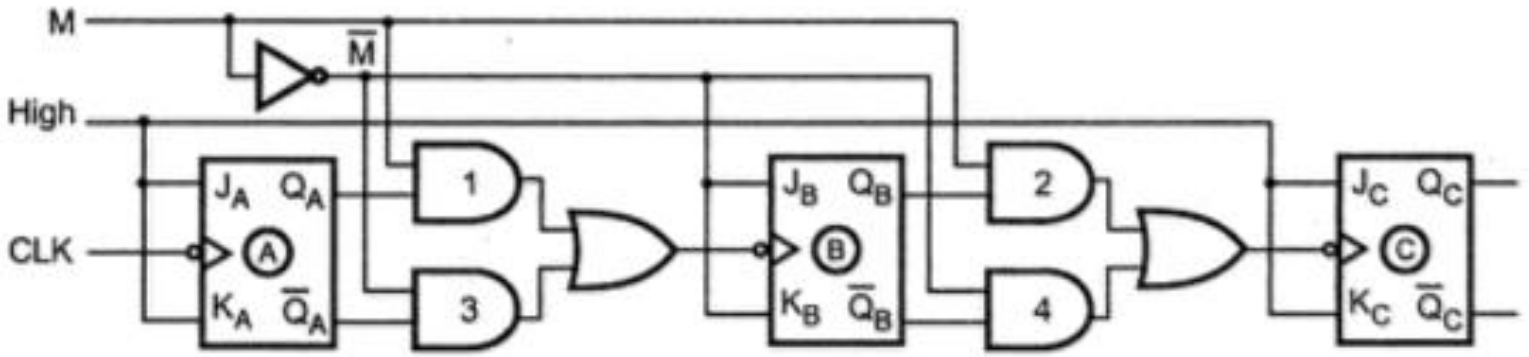
	Inputs M	Q	$\overline{Q}$	Output Y	
M = 0	0	0	0	0	$Y = \overline{Q}$ for down counting
	0	0	1	1	
	0	1	0	0	
	0	1	1	1	
M = 1	1	0	0	0	$Y = Q$ for up counting
	1	0	1	0	
	1	1	0	1	
	1	1	1	1	

(b) Truth table

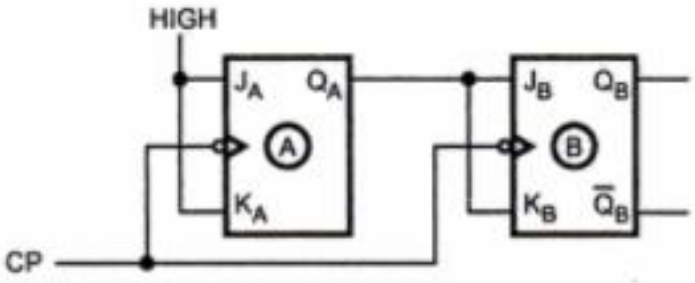




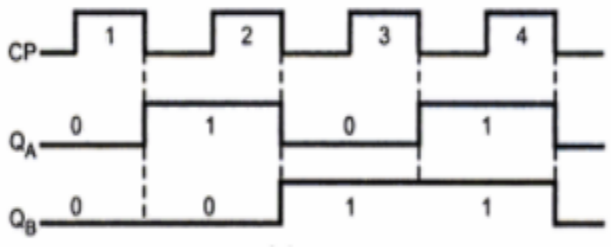
A logic 1 on M enables AND gates 1 and 2 and disables AND gates 3 and 4. This allows the Q_A and Q_B outputs to drive the clock inputs of their respective next stages. So that counter will count up. When M is logic 0, AND gates 1 and 2 are disabled and AND gete 3 and 4 are enabled. This allows the $\overline{Q}_A$ and $\overline{Q}_B$ outputs to drive the clock inputs of their respective next stages so that counter will count down.



Synchronous Counters

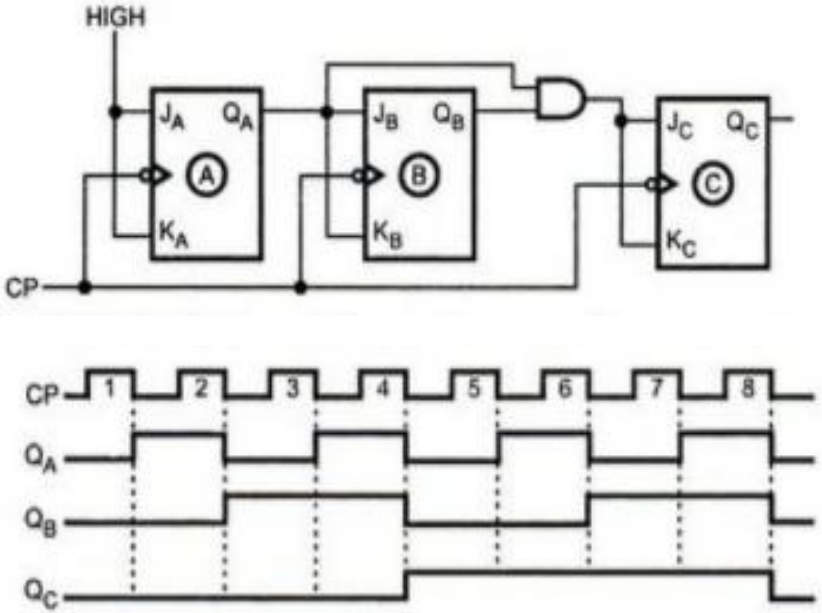


When negative edge of the first clock pulse is applied, flip-flop A will toggle because $J_A = K_A = 1$, whereas flip-flop B output will remain zero because $J_B = K_B = 0$. After first clock pulse $Q_A = 1$ and $Q_B = 0$. At negative going edge of the second clock pulse both flip-flops will toggle because they both have a toggle condition on their J and K inputs ($J_A = K_A = J_B = K_B = 1$). Thus after second clock pulse, $Q_A = 0$ and $Q_B = 1$. At negative going edge of the third clock pulse flip-flop A toggles making $Q_A = 1$, but flip-flop B remains set i.e. $Q_B = 1$. Finally, at the leading edge of the fourth clock pulse both flip-flops toggle as their JK inputs are at logic 1. This results $Q_A = Q_B = 0$ and counter recycled back to its original state. The timing details of



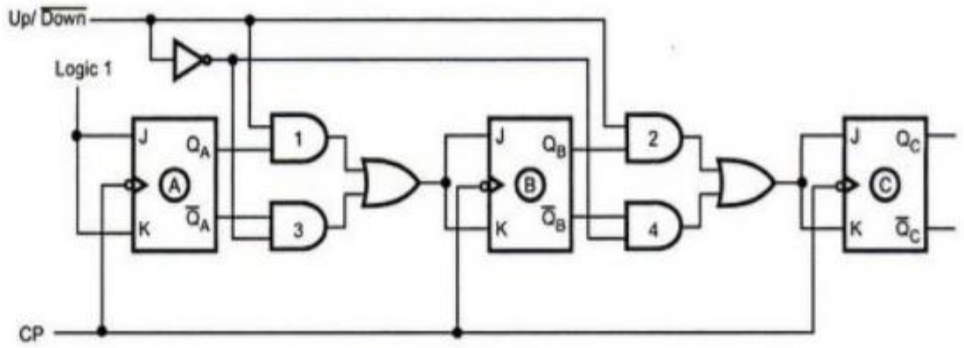
CP	Q_B	Q_A
0	0	0
1	0	1
2	1	0
3	1	1

3-bit Synchronous Binary Up Counter



CP	Q_C	Q_B	Q_A
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

3-bit synchronous/parallel up/down counter



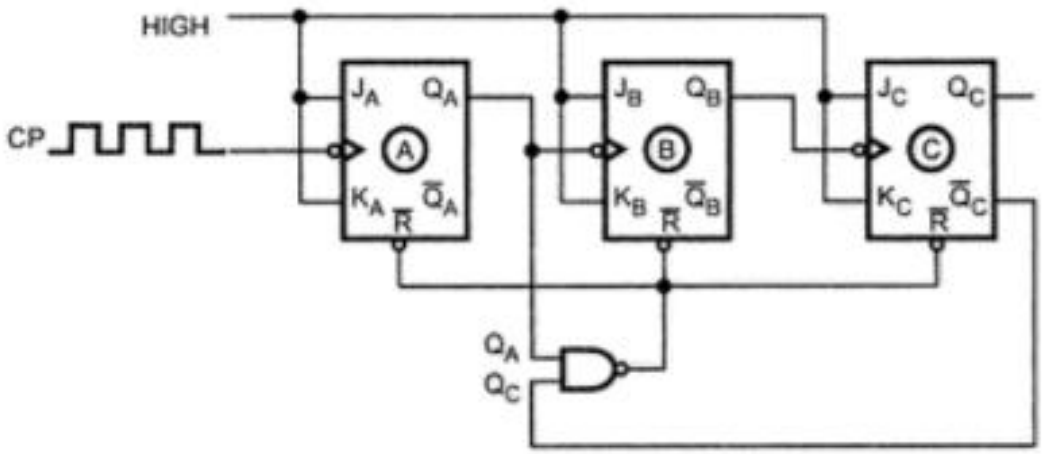
A logic 1 on the Up/$\overline{\text{Down}}$ enables AND gates 1 and 2 and disables AND gates 3 and 4. This allows the Q_A and Q_B outputs through to the J and K inputs of the next flip-flops so that the counter will count up as pulses are applied. When Up/$\overline{\text{Down}}$ line is logic 0, AND gates 1 and 2 are disables and AND gates 3 and 4 are enabled. This allows the $\overline{Q}_A$ and $\overline{Q}_B$ outputs through to the J and K inputs of the next flip-flops so that the counter will count down as pulses are applied.

MOD-5 counter using RESET input

The NAND gate output is connected to the asynchronous RESET inputs of each flip-flop. As long as the NAND output is HIGH, it will have no effect on the counter. When it goes LOW, it will reset all the flip-flops so that counter immediately goes to the 000 state.

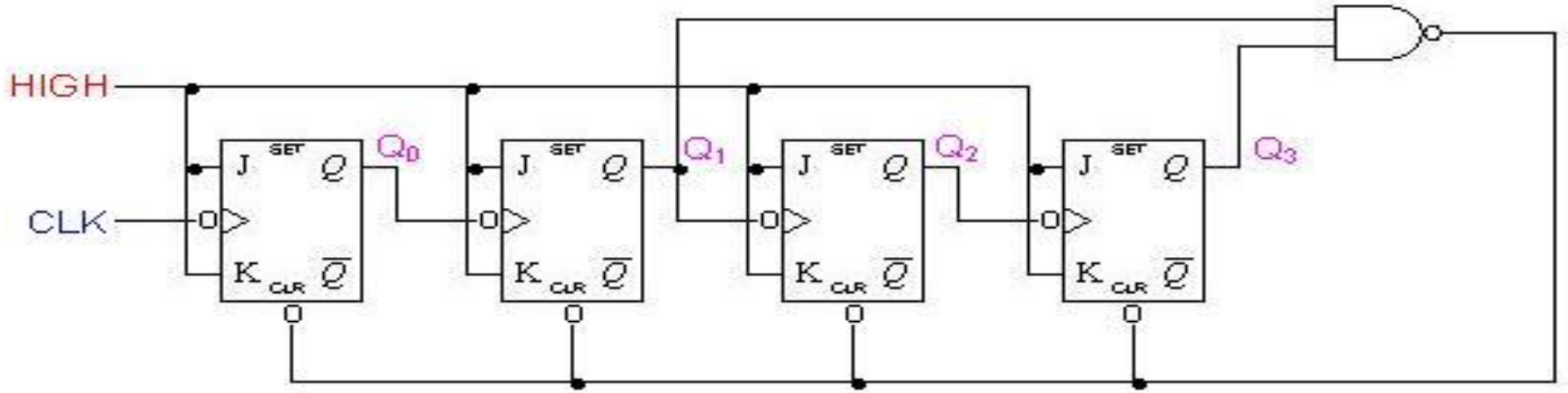
The inputs for the NAND gate are the outputs of the A and C flip-flops, and so the NAND output will go LOW whenever $Q_A = Q_C = 1$. This condition will occur when the counter goes from the 100 state to the 101 state (input pulse 5 on waveforms). The LOW at the NAND output will immediately (generally within a few nanoseconds) reset the counter to the 000 state.

C	B	A
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1



The counting sequence is 000 through 101 therefore, though the counter does go to the 101 state, it remains there for only a few nanoseconds

Decade Counters



Once the counter counts to ten (1010), all the flip-flops are being cleared.

Clock Pulse	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Design of a Synchronous Mod-6 Counter using Clocked JK Flip-Flops

Find number of flip-flops required to build the counter

Flip-flops required are : $2^n \geq N$.

Here $N = 6 \quad \therefore n = 3$

i.e. Three flip-flops are required.

Determine the transition table.

Present state			Next state			Flip-flop inputs					
Q_A	Q_B	Q_C	Q_{A+1}	Q_{B+1}	Q_{C+1}	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	1	0	1	x	0	0	x	1	x
1	0	1	0	0	0	x	1	0	x	x	1
1	1	0	x	x	x	x	x	x	x	x	x
1	1	1	x	x	x	x	x	x	x	x	x

For J_A

Q_A \ Q_BQ_C	00	01	11	10
0	0	0	1	0
1	X	X	X	X

$J_A = Q_B Q_C$

For K_A

Q_A \ Q_BQ_C	00	01	11	10
0	X	X	X	X
1	0	1	X	X

$K_A = Q_C$

For J_B

Q_A \ Q_BQ_C	00	01	11	10
0	0	1	X	X
1	0	0	X	X

$J_B = \overline{Q}_A Q_C$

For K_B

Q_A \ Q_BQ_C	00	01	11	10
0	X	X	1	0
1	X	X	X	X

$K_B = Q_C$

For J_C

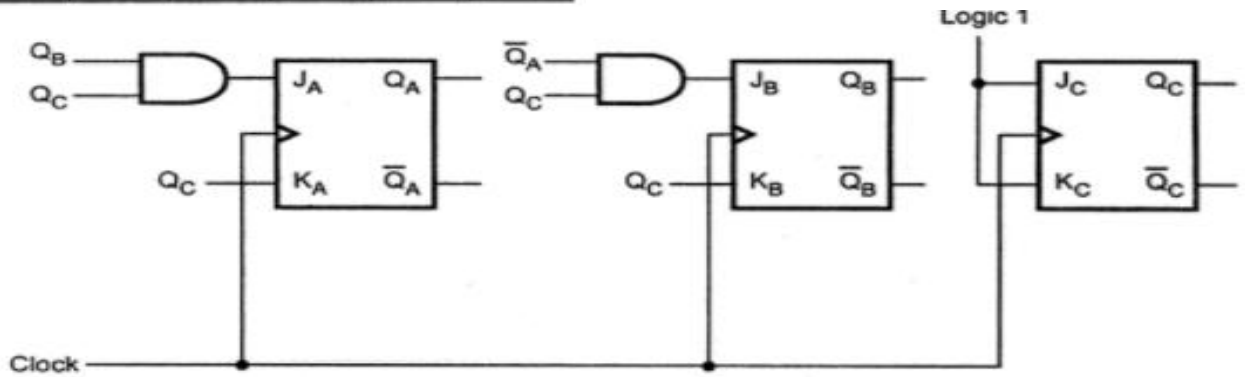
Q_A \ Q_BQ_C	00	01	11	10
0	1	X	X	1
1	1	X	X	X

$J_C = 1$

For K_C

Q_A \ Q_BQ_C	00	01	11	10
0	X	1	1	X
1	X	1	X	X

$K_C = 1$



Design of a Synchronous Mod-6 Counter using Clocked SR Flip-Flop

Present state			Next state			Flip-flop inputs					
Q_A	Q_B	Q_C	Q_{A+1}	Q_{B+1}	Q_{C+1}	S_A	R_A	S_B	R_B	S_C	R_C
0	0	0	0	0	1	0	x	0	x	1	0
0	0	1	0	1	0	0	x	1	0	0	1
0	1	0	0	1	1	0	x	x	0	1	0
0	1	1	1	0	0	1	0	0	1	0	1
1	0	0	1	0	1	x	0	0	x	1	0
1	0	1	0	0	0	0	1	0	x	0	1
1	1	0	x	x	x	x	x	x	x	x	x
1	1	1	x	x	x	x	x	x	x	x	x

For S_A

Q_A \ Q_BQ_C	00	01	11	10
0	0	0	1	0
1	X	0	X	X

$S_A = Q_BQ_C$

For R_A

Q_A \ Q_BQ_C	00	01	11	10
0	X	X	0	X
1	0	1	X	X

$R_A = \overline{Q}_BQ_C$

For S_B

Q_A \ Q_BQ_C	00	01	11	10
0	0	1	X	0
1	0	0	X	X

$S_B = \overline{Q}_AQ_C$

For R_B

Q_A \ Q_BQ_C	00	01	11	10
0	X	0	1	0
1	X	X	X	X

$R_B = Q_BQ_C$

For S_C

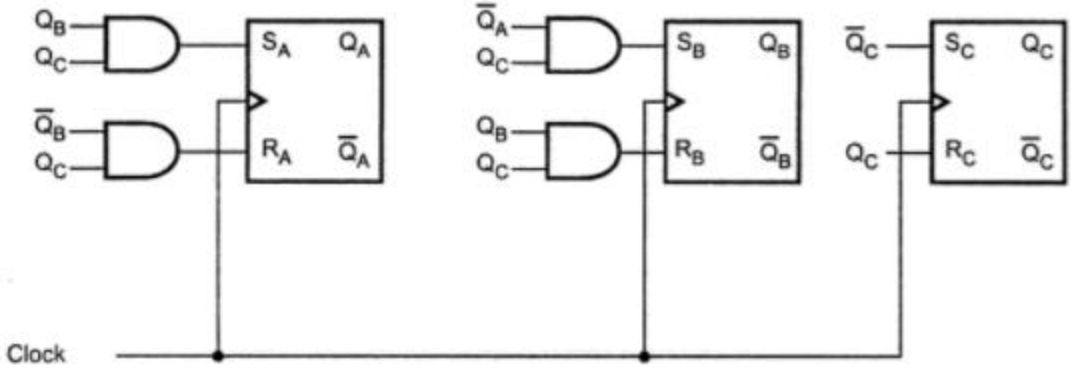
Q_A \ Q_BQ_C	00	01	11	10
0	1	0	0	1
1	1	0	X	X

$S_C = \overline{Q}_C$

For R_C

Q_A \ Q_BQ_C	00	01	11	10
0	0	1	1	0
1	0	1	X	X

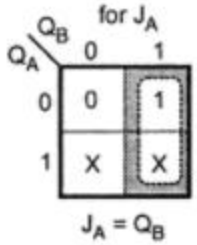
$R_C = Q_C$

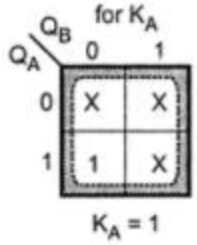


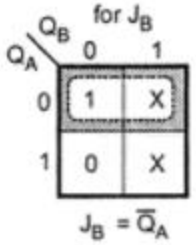
Design synchronous mod-3 counter with the following binary sequence using clocked JK flip-flops.

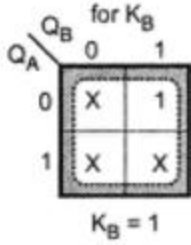
Count sequence : 0, 1, 2, 0, 1, 2,.....

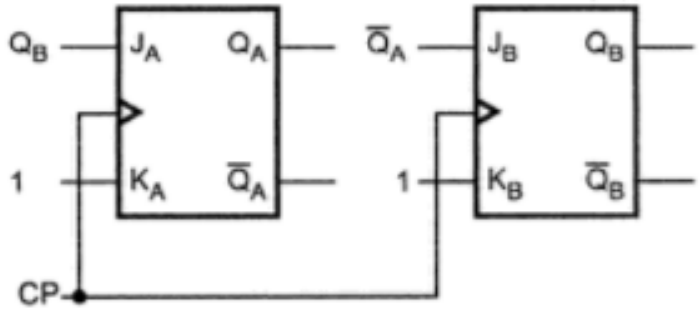
Present state		Next state		Flip-flop inputs			
Q_A	Q_B	Q_{A+1}	Q_{B+1}	J_A	K_A	J_B	K_B
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	0	0	X	1	0	X











The Shift Register

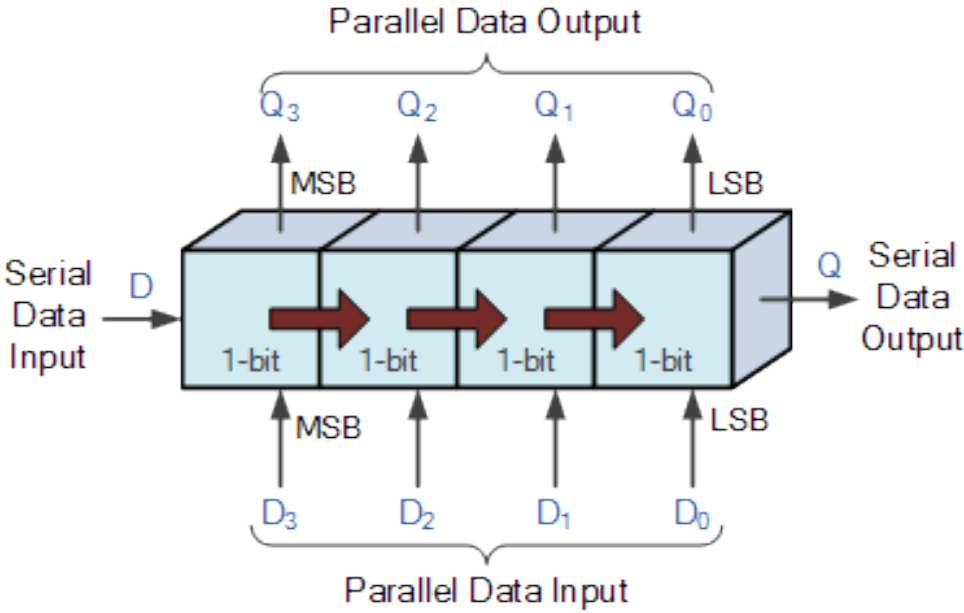
The **Shift Register** used for the storage or the transfer of binary data

Serial-in to Serial-out (SISO) - the data is shifted serially "IN" and "OUT" of the register, one bit at a time in either a left or right direction under clock control.

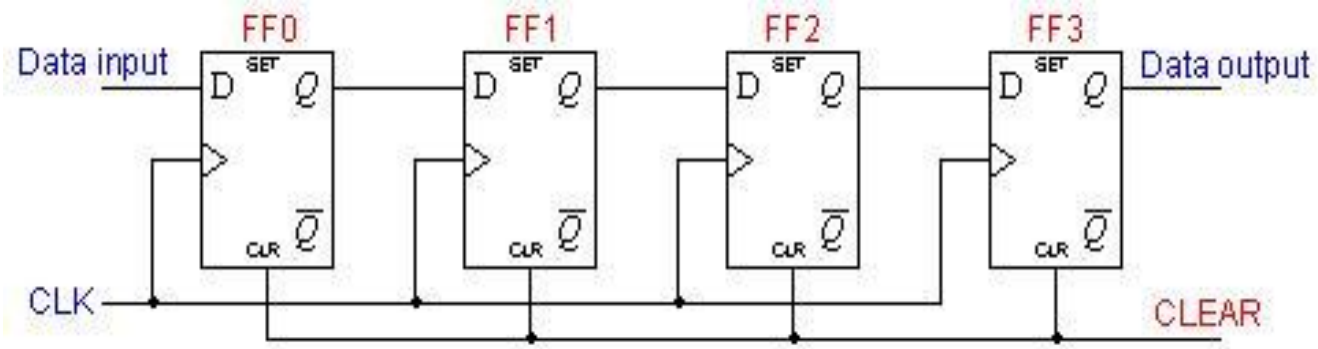
Serial-in to Parallel-out (SIPO) - the register is loaded with serial data, one bit at a time, with the stored data being available at the output in parallel form.

Parallel-in to Serial-out (PISO) - the parallel data is loaded into the register simultaneously and is shifted out of the register serially one bit at a time under clock control.

Parallel-in to Parallel-out (PIPO) - the parallel data is loaded simultaneously into the register, and transferred together to their respective outputs by the same clock pulse.



Serial-in to Serial-out (SISO) Shift Register

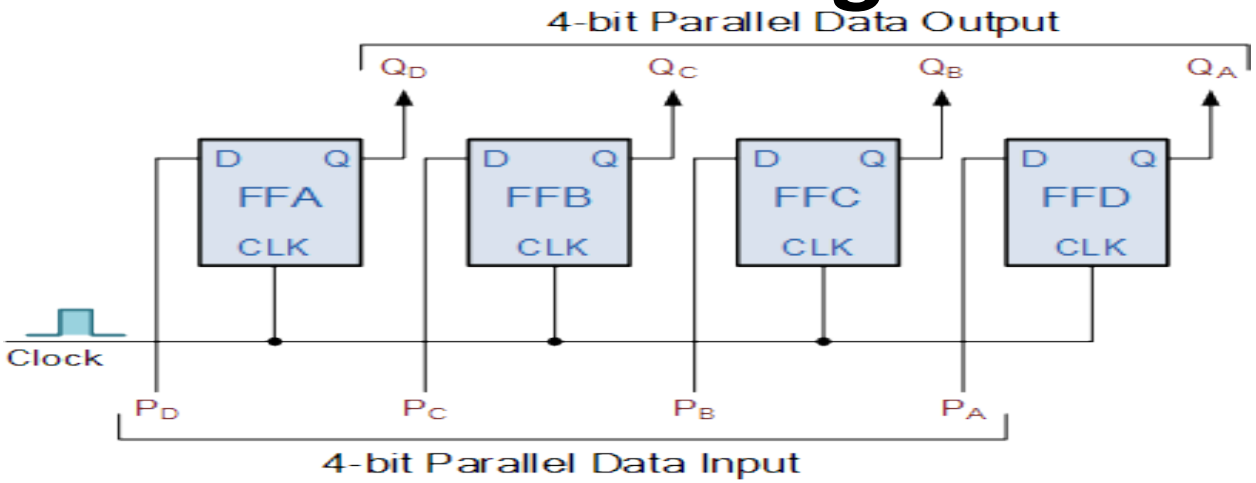


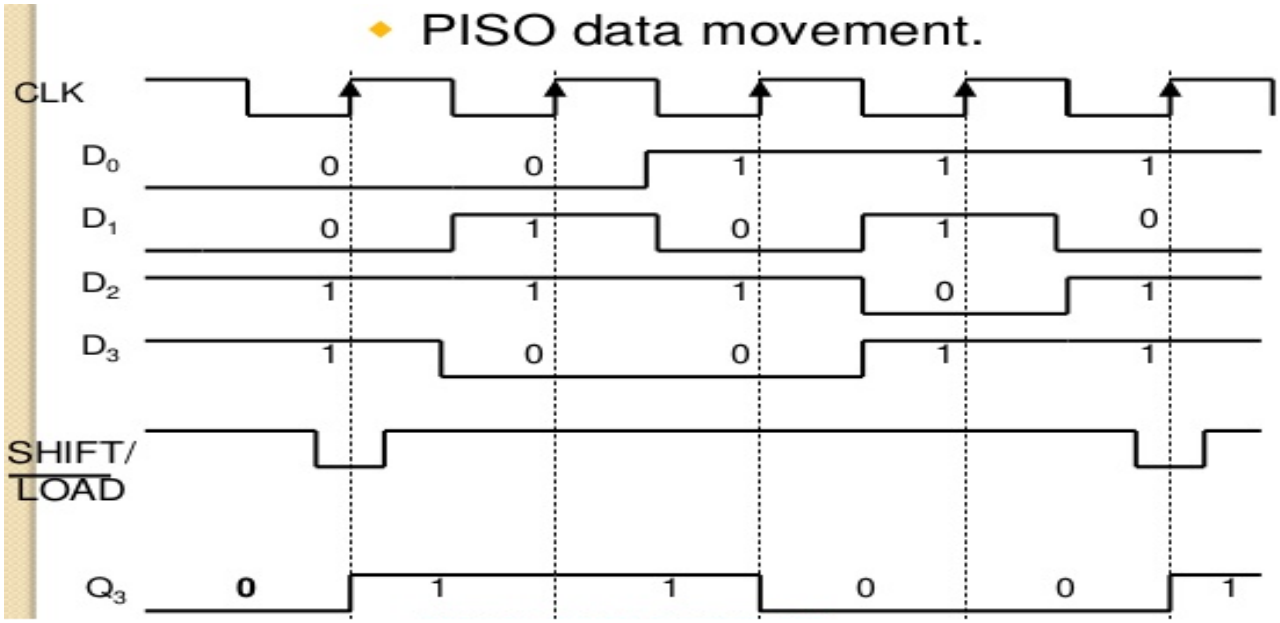
All the FF are reset and a logical input 1011 is applied at the serial input line connected to stage FF0

Operation of the Shift-right Register

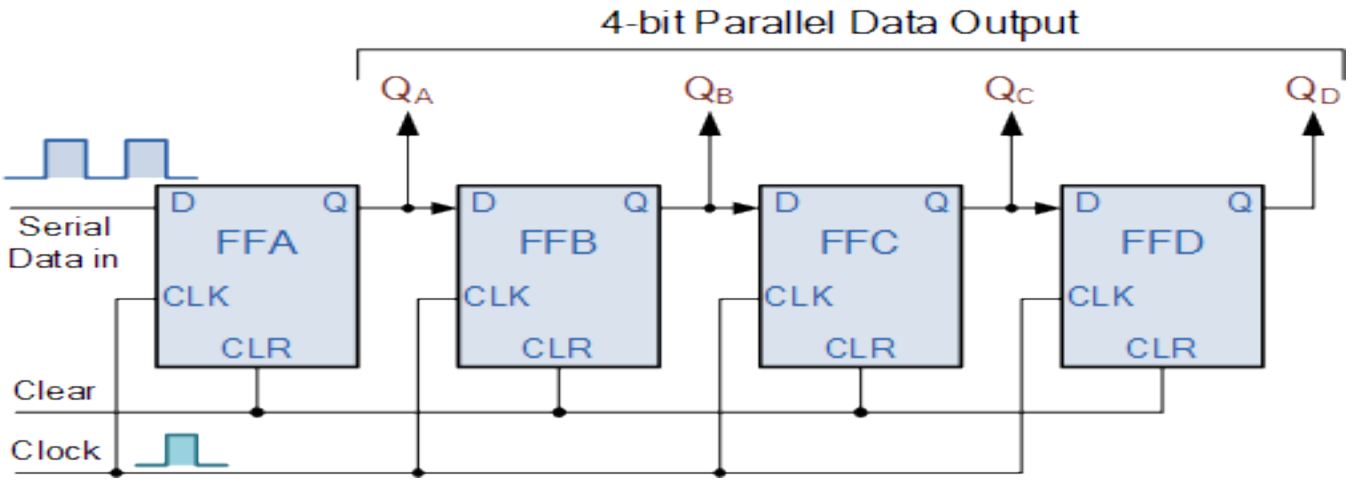
Timing pulse	Q_A	Q_B	Q_C	Q_D	*Serial output at* Q_D
Initial value	0	0	0	0	0
After 1st clock pulse	1	0	0	0	0
After 2nd clock pulse	1	1	0	0	0
After 3rd clock pulse	0	1	1	0	0
After 4th clock pulse	1	0	1	1	1

4-bit Parallel-in to Parallel-out Shift Register

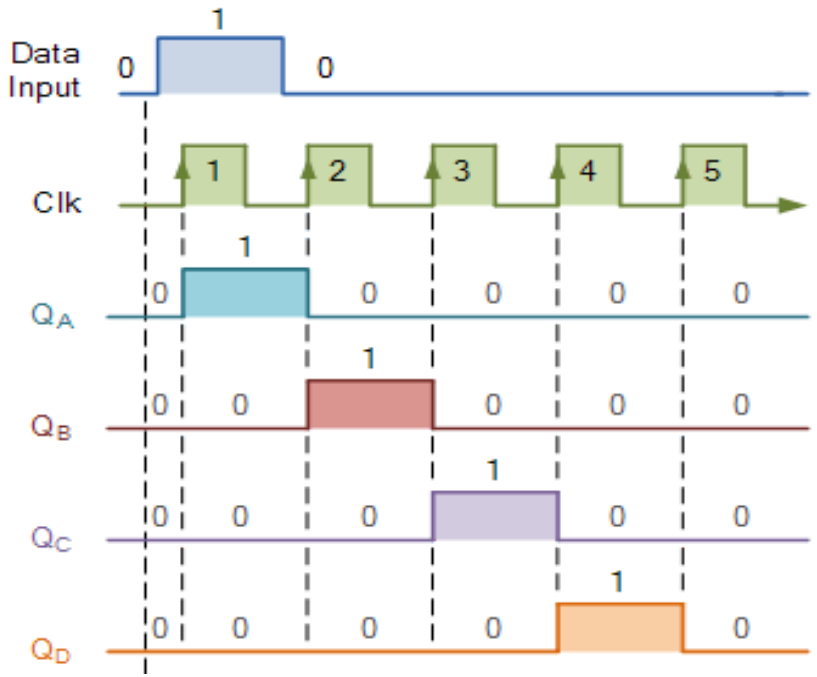




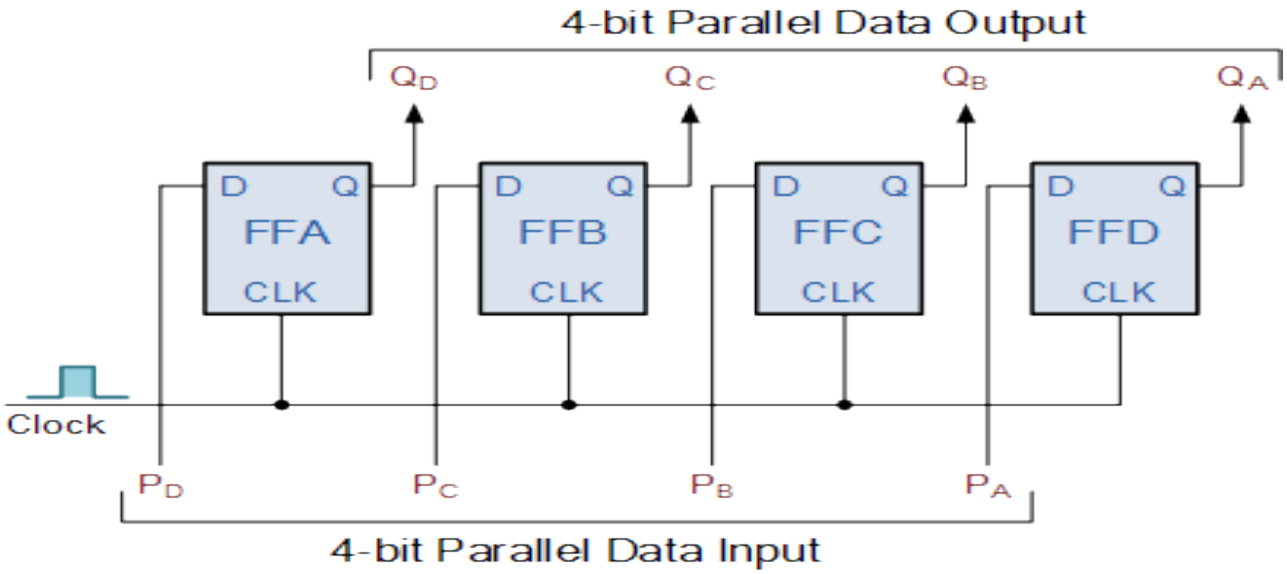
4-bit Serial-in to Parallel-out Shift Register



Clock Pulse No	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	0

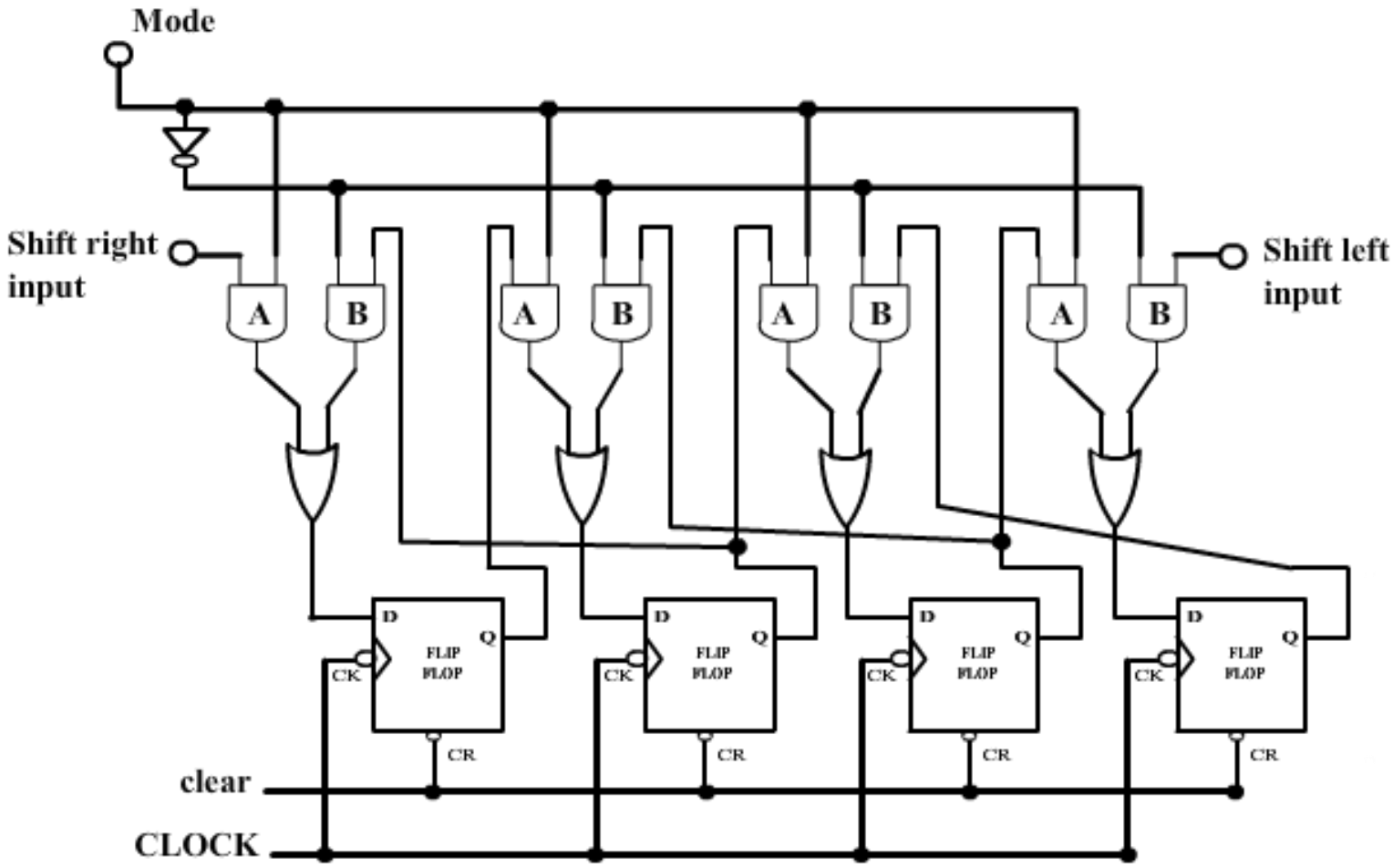


PIPO Shift Register

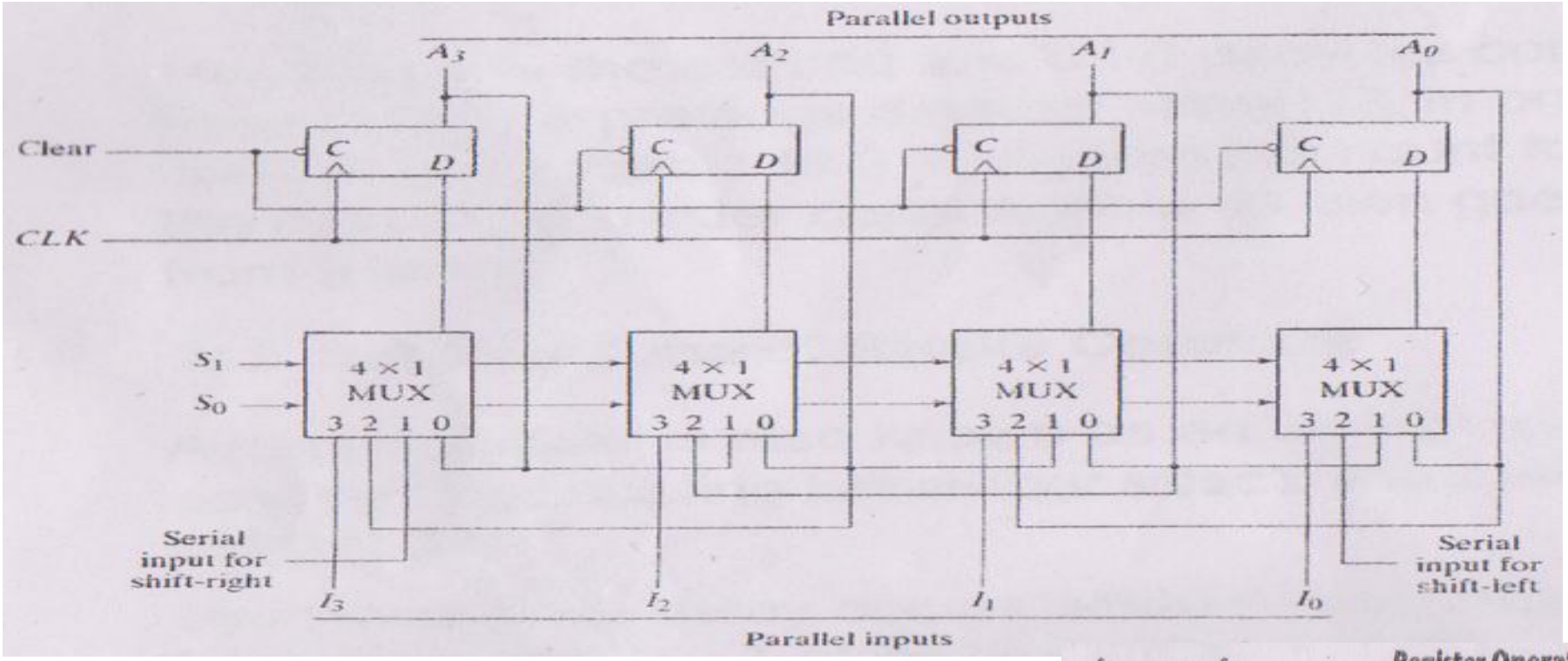


Require 1 clk to load parallel data

Bidirectional shift-register



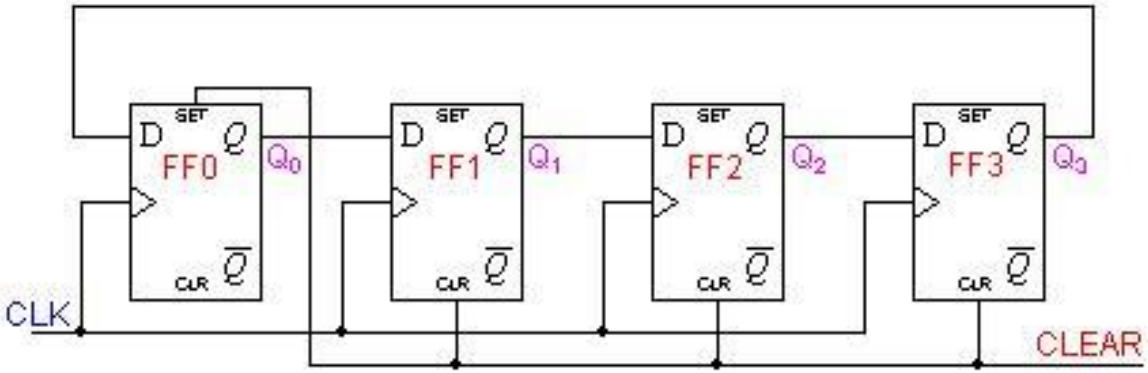
Universal Shift Register



S_1	S_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

It consists of four flip-flops and four multiplexers. The four multiplexers have two common selection inputs S_1 and S_0, and they select appropriate input for D flip-flop. The Table 8.2 shows the register operation depending on the selection inputs of multiplexers. When S_1S_0 = 00, input 0 is selected and the present value of the register is applied to the D inputs of the flip-flops. This results no change in the register value. When S_1S_0 = 01, input 1 is selected and circuit connections are such that it operates as a right shift register. When S_1S_0 = 10, input 2 is selected and circuit connections are such that it operates as a left-shift register. Finally, when S_1S_0 = 11, the binary information on the parallel input lines is transferred into the register simultaneously and it is a parallel load operation.

Ring Counter

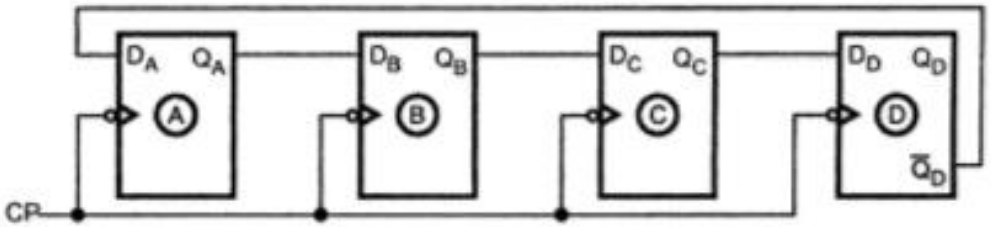


Clock Pulse	Q3	Q2	Q1	Q0
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	1	0	0	0

Last FF output(Q) feed back input to first FF
Count n clock pulse
MOD-N counter
At least 1 binary 1 as initial value

Johnson or Twisting Ring or Switch Tail Counter

In a Johnson counter, the Q output of each stage of flip-flop is connected to the D input of the next stage. The single exception is that the complement output of the last flip-flop is connected back to the D-input of the first flip-flop



Clock pulse	Q_A	Q_B	Q_C	Q_D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

Last FF output(~Q) feed back input to first FF
Count 2n clock pulse
MOD-2N counter