Digital Fundamentals (3130704)

MODULE 3

SEQUENTIAL CIRCUITS AND SYSTEMS

Module 3

- Sequential circuits and systems :A 1-bit memory, the circuit properties of Bistable latch, the clocked SR flip flop
- J- K-T and D types flip flops, applications of flip flops, shift registers, applications of shift registers, serial to parallel converter, parallel to serial converter, ring counter
- sequence generator, ripple(Asynchronous) counters, synchronous counters
- counters design using flip flops, special counter IC's, asynchronous sequential counters, applications of counters

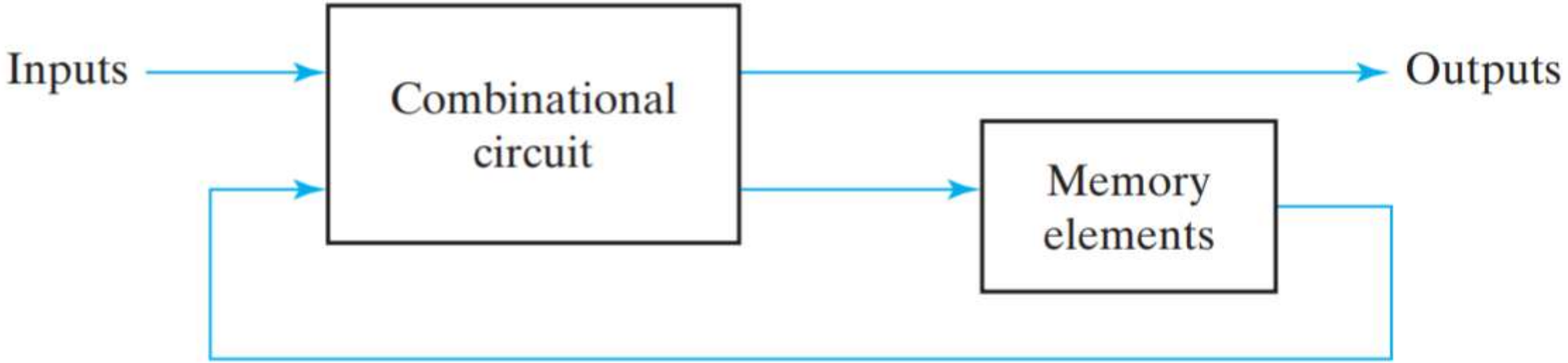
Topics to be covered

- Flip-flops
- Applications of Flip-flops
- Shift registers
- Asynchronous counters
- Synchronous counters
- Sequential counters

Sequential Switching Circuits

- Sequential switching circuits are circuits whose output levels at any instant of time are dependent on the levels present at the inputs at that time and on the state of the circuit, i.e., on the prior input level conditions (i.e. on its past inputs)
- The past history is provided by feedback from the output back to the input.
- Made up of combinational circuits and memory elements.
- Eg. Counters, shift registers, serial adder, etc.

Sequential Switching Circuits



Sequential Circuits v/s Combinational Circuits

- Sequential Circuits

- In sequential circuits, the output variables at any instant of time are dependent on the present input variables and on the present state, i.e., on the past history of the system.
- Memory unit is required to store the past history of the input variables in sequential circuits.
- Sequential circuits are slower than combinational circuits.
- Sequential circuits are comparatively harder to design.

- Combinational Circuits

- In combinational circuits, the output variables at any instant of time are dependent only on the present input variables.
- Memory unit is not required in combinational circuits.
- Combinational circuits are faster because the delay between the input and the output is due to propagation delay of gates only.
- Combinational circuits are easy to design.

Flip-flop

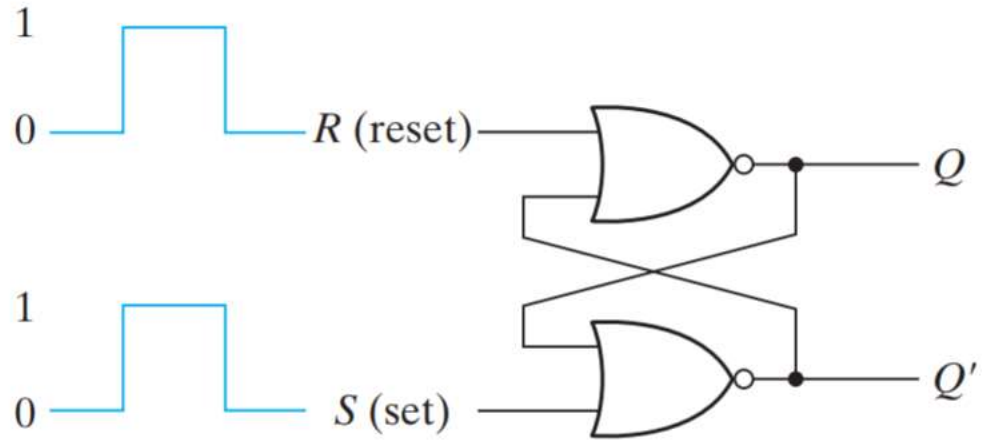
- A flip-flop, known formally as bistable multivibrator, has two stable states.
- It can remain in either of the states indefinitely.
- Its state can be changed by applying the proper triggering signal.

Latch

- Latch is used for certain flip-flop which are *non-clocked.*
- These flip-flops ‘latch on’ to a 1 or a 0 immediately upon receiving the input pulse called SET or RESET.

S-R Flip-Flop (Latch)

- The simplest type of flip-flop is called an S-R latch.
- It has two outputs labelled Q and Q' and two inputs labelled S and R. The state of the latch corresponds to the level of Q (HIGH or LOW, 1 or 0) and Q' is the complement of that state.
- It can be constructed using either two cross-coupled NAND gates or two-cross coupled NOR gates.
- Using two NOR gates, an active-HIGH S-R latch can be constructed and using two NAND gates an active-LOW S-R latch can be constructed.
- The name of the latch, S-R or SET-RESET, is derived from the names of its inputs.

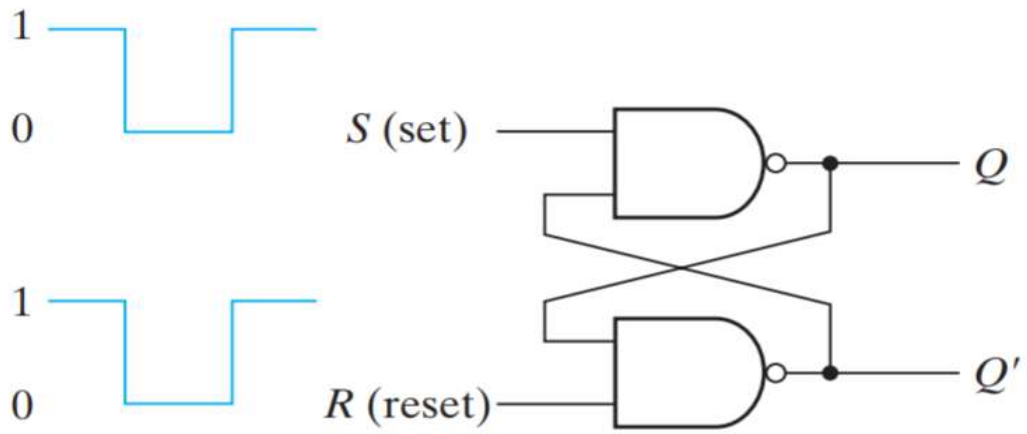


(a) Logic diagram

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	

(b) Function table

SR Latch with NOR Gates

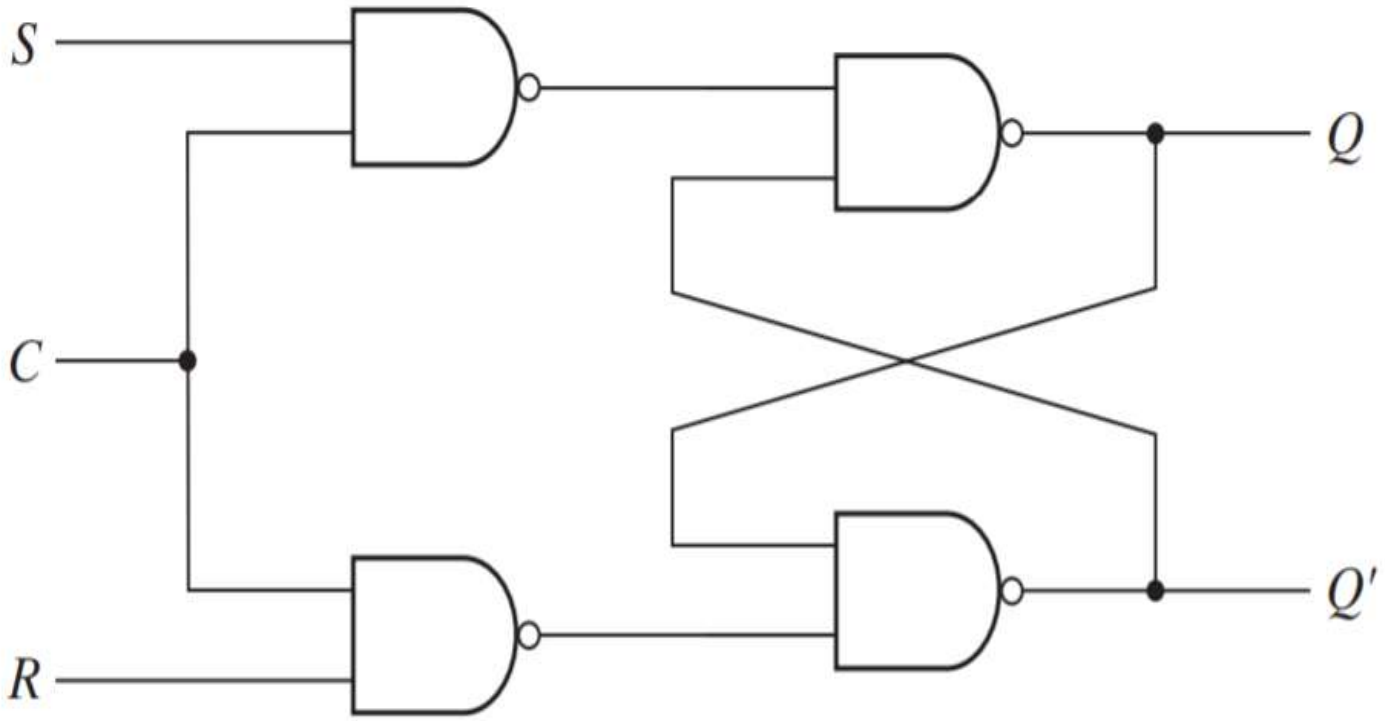


(a) Logic diagram

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	

(b) Function table

SR Latch with NAND Gates

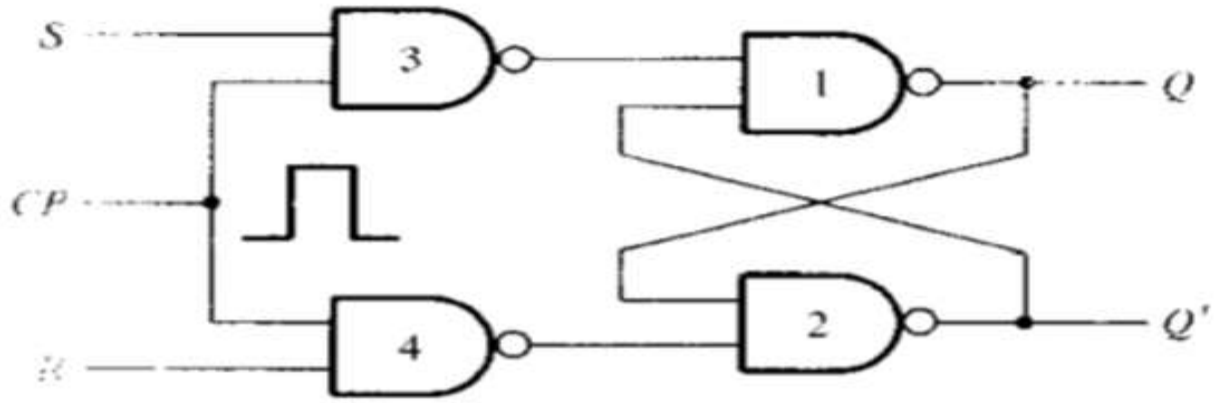


(a) Logic diagram

C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; Reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

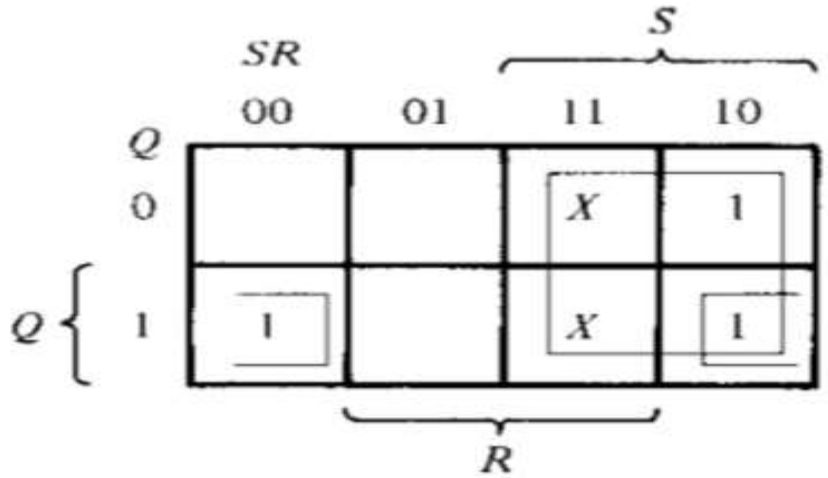
(b) Function table

SR Latch with Control Input



(a) Logic diagram

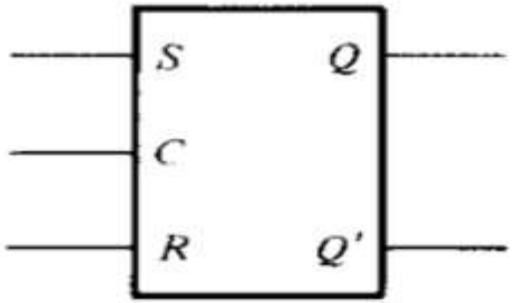
Q	S	R	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	Intermediate
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Intermediate



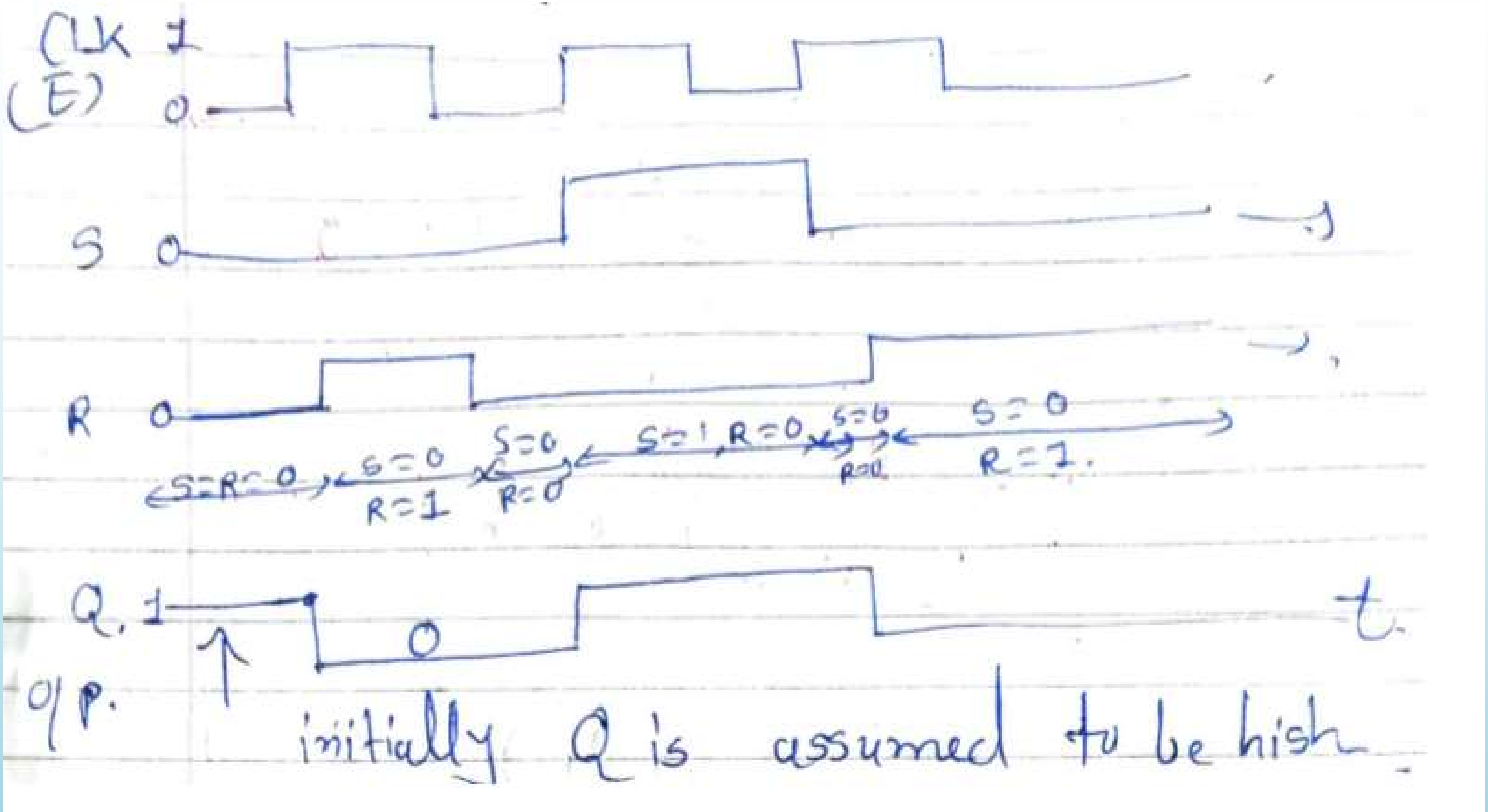
$Q(t+1) = S + R'Q$

$SR = 0$

(c) Characteristic equation



(d) Graphic symbol



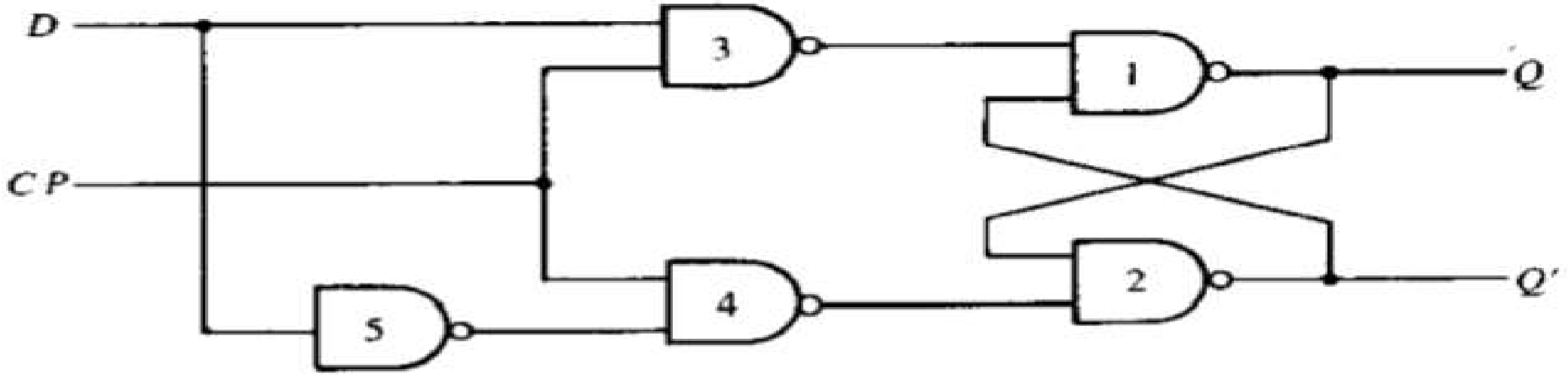
Q.P. initially Q is assumed to be high.

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.

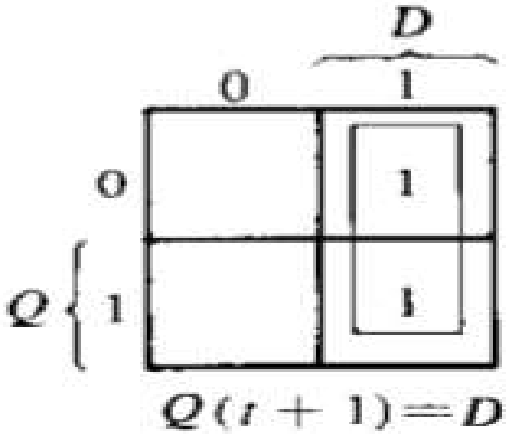
- The circuit diagram of D flip-flop is shown in the following figure.



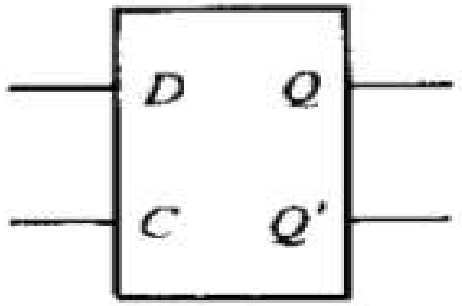
(a) Logic diagram

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

(b) Characteristic table



(c) Characteristic equation

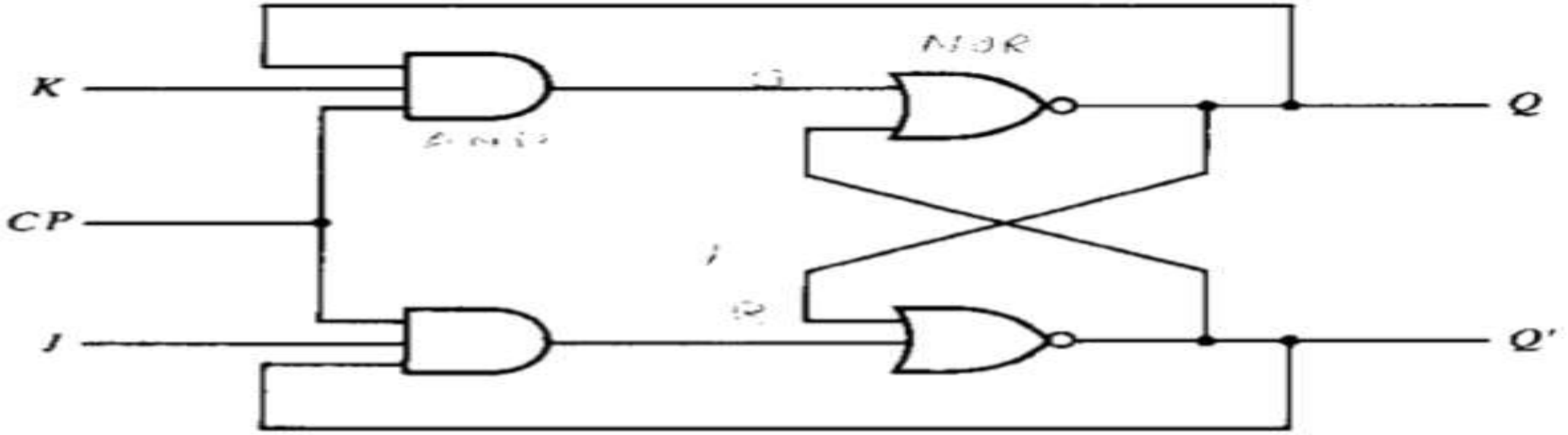


(d) Graphic symbol

FIGURE 6-5
D flip-flop

JK Flip-Flop

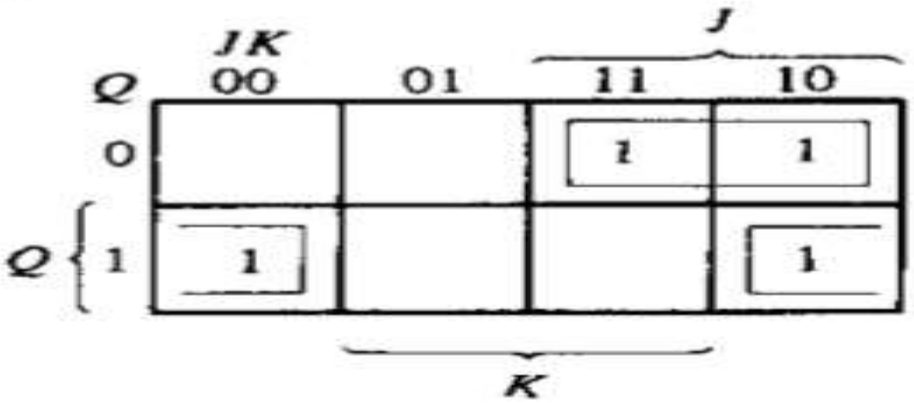
A *JK* flip-flop is a refinement of the *RS* flip-flop in that the indeterminate state of the *RS* type is defined in the *JK* type. Inputs *J* and *K* behave like inputs *S* and *R* to set and clear the flip-flop, respectively. The input marked *J* is for *set* and the input marked *K* is for *reset*. When both inputs *J* and *K* are equal to 1, the flip-flop switches to its complement state, that is, if $Q = 1$, it switches to $Q = 0$, and vice versa.



(a) Logic diagram

Q	J	K	Q(t + 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

(b) Characteristic table

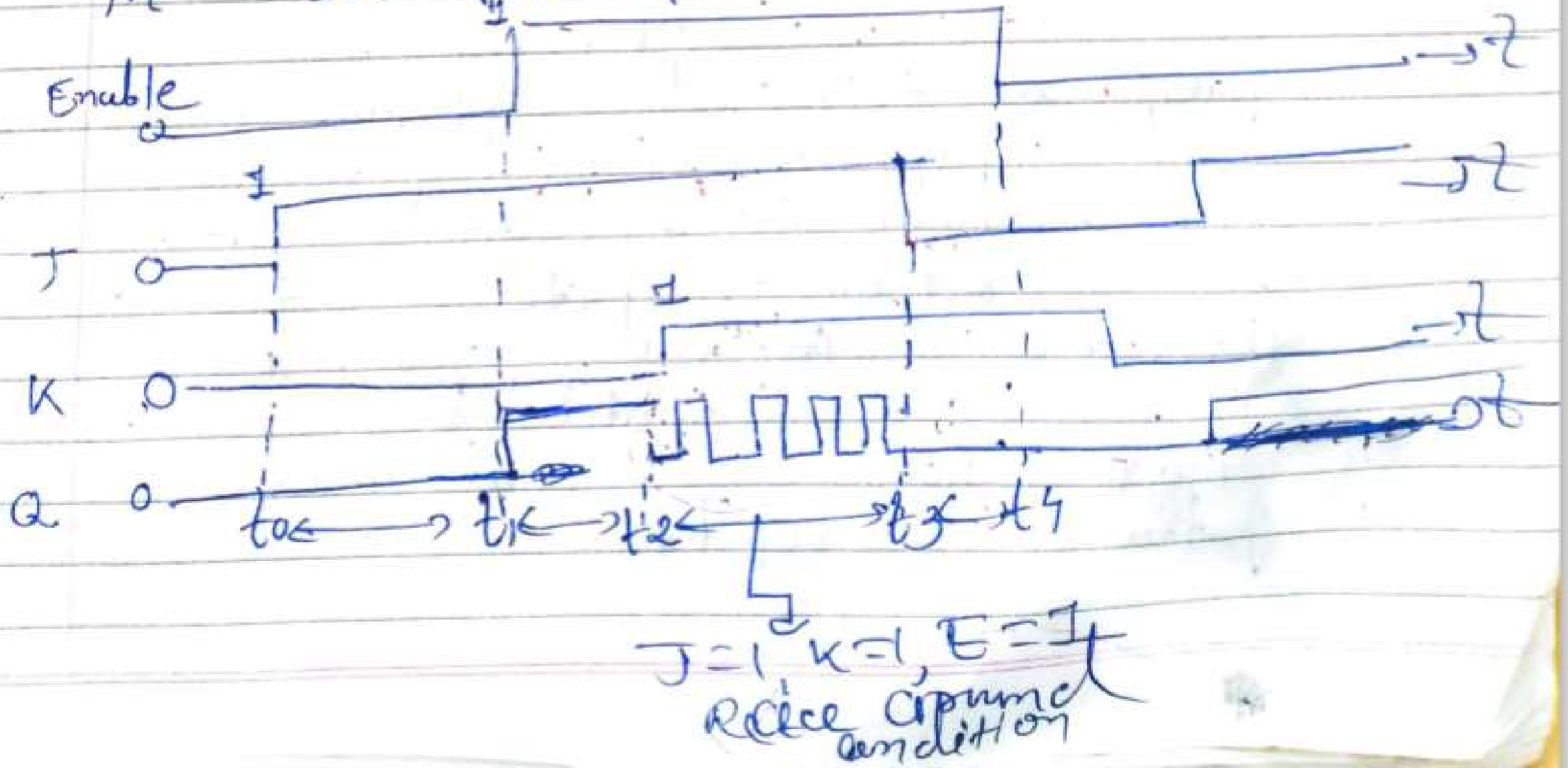


$$Q(t+1) = JQ' + K'Q$$

(c) Characteristic equation

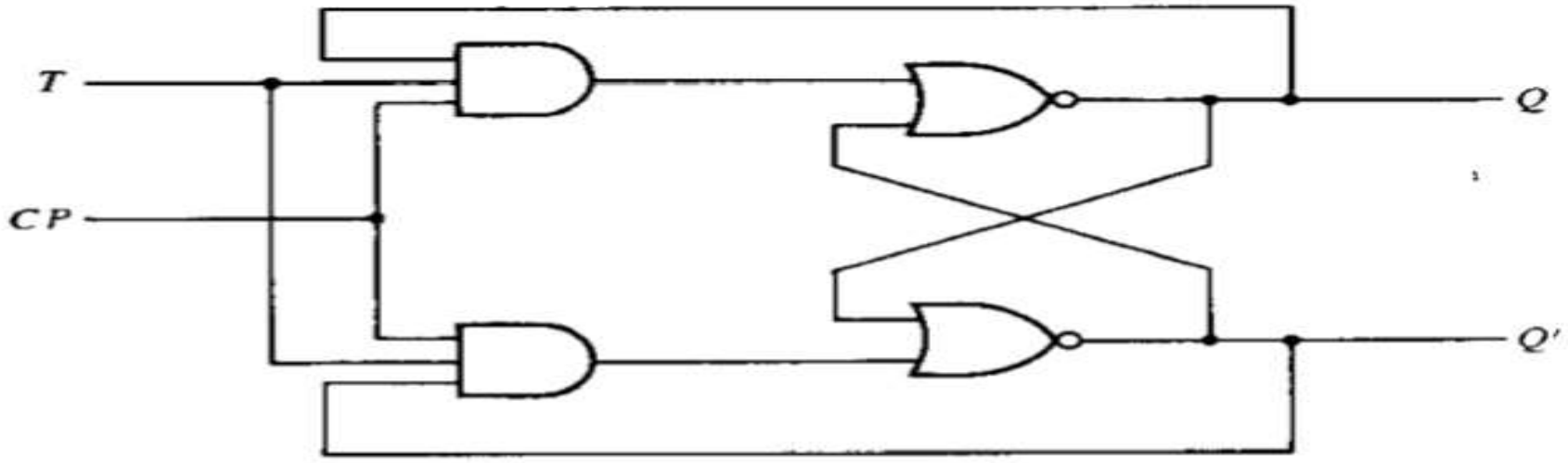
FIGURE 6-6
JK flip-flop

→ J = K = 1 , it is a Race - around Condition in JK flip flop.



T Flip-Flop

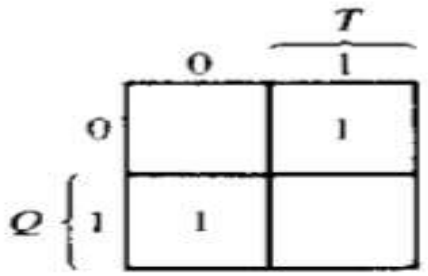
T flip-flop is the simplified version of JK flip-flop. It is obtained by connecting the same input 'T' to both inputs of JK flip-flop. It operates with only positive clock transitions or negative clock transitions. The **circuit diagram** of T flip-flop is shown in the following figure.



(a) Logic diagram

Q	T	Q(t + 1)
0	0	0
0	1	1
1	0	1
1	1	0

(b) Characteristic table



$Q(t+1) = TQ' + T'Q$

(c) Characteristic equation

FIGURE 6-7
T flip-flop

→ How to avoid race around condition?

Race-round Condition in JK latch can be avoided by,

(1) using the edge triggered JK flip-flop.

(2) using the master slave JK flip-flop

Triggering of Flip-Flops

Triggering of flip-flops

state of flip-flop is switched by a momentary change in the input signal. This momentary is called a trigger, and the transition it causes is said to trigger the flip-flop. Clocked flip-flops are triggered by pulses. A pulse start from a initial value of 0, goes momentarily to 1, and after a short time, returns to its initial 0 value. The time interval from the application of the pulse until the o/p transition occurs is a critical factor that needs further investigation.

Depending on which portion of clock signal the latch or flip-flop responds to, we can classify them into two types:

(1) level triggered circuits
(2) Edge triggered circuits.

→ Concept of level triggering

The latch or flip-flop circuits which respond to their inputs, only if their enable input (E) held at an active HIGH or LOW level are called as level triggered latches or flip-flops.

→ Types of level triggered flip-flops.

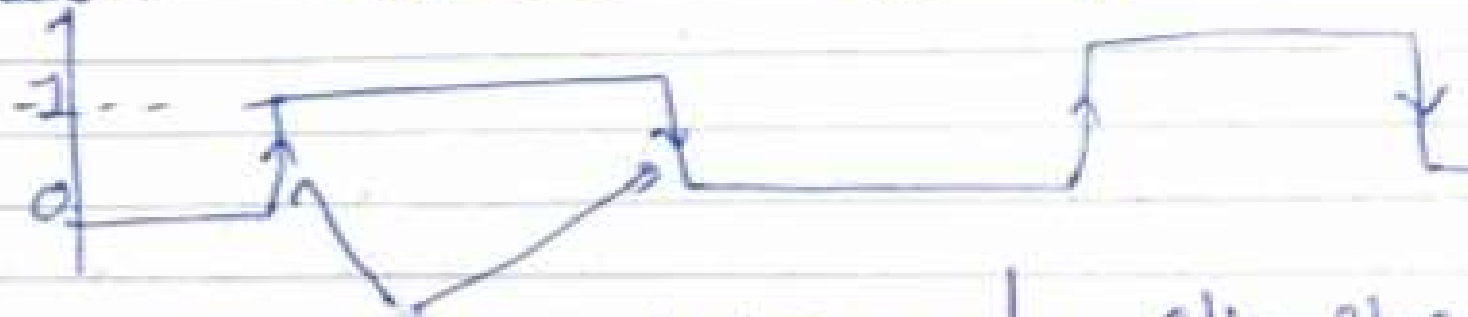
(1) Positive level triggered.
(2) Negative level triggered.

→ if o/ps of S-R flip-flop respond to the input changes, for its clock input at HIGH(1) level then it is called +ve level triggered S-R flip-flop.

→ if the outputs of S-R flip-flop respond to the input changes, for its clock input at Low(0) level, then it is called +ve level triggered S-R FF.

→ Edge triggering

The flip-flops which change their o/p only corresponding to the positive or negative edge of the clock input are called as edge triggered flip-flops.



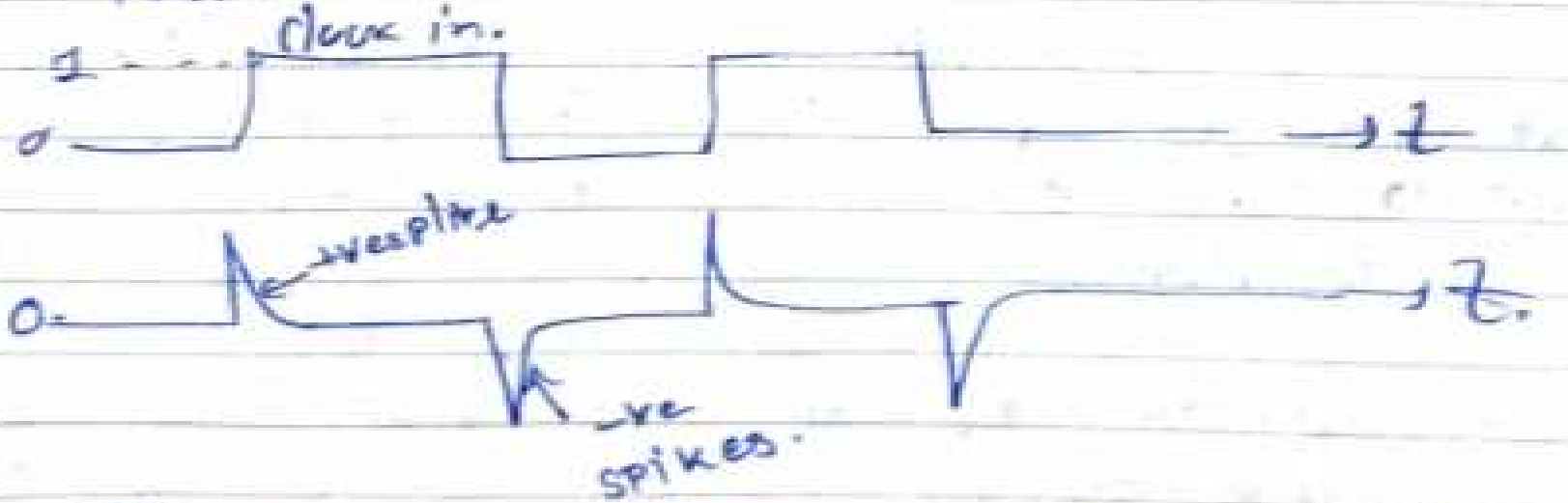
The edge triggered flip-flop samples the inputs at the positive or negative edge and changes its o/p accordingly.

→ Types of edge triggered flip-flops.

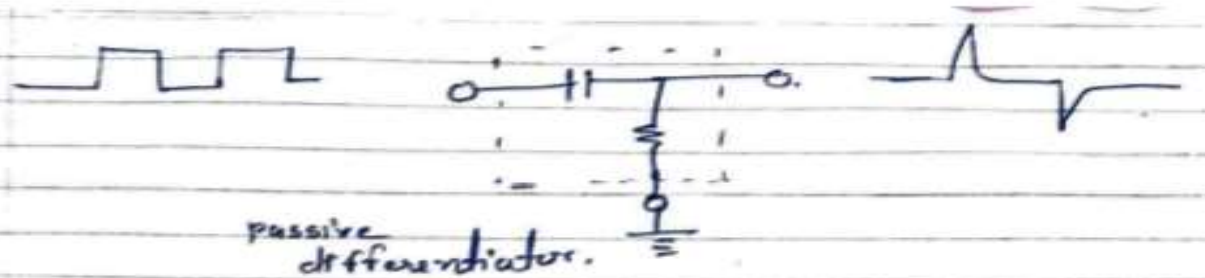
(a) Positive edge triggered.

(b) -ve edge triggered.

→ for the edge triggered flip-flops, it is necessary to apply the clock signal in the form of sharp positive and negative spikes instead of in the form of pulse train.

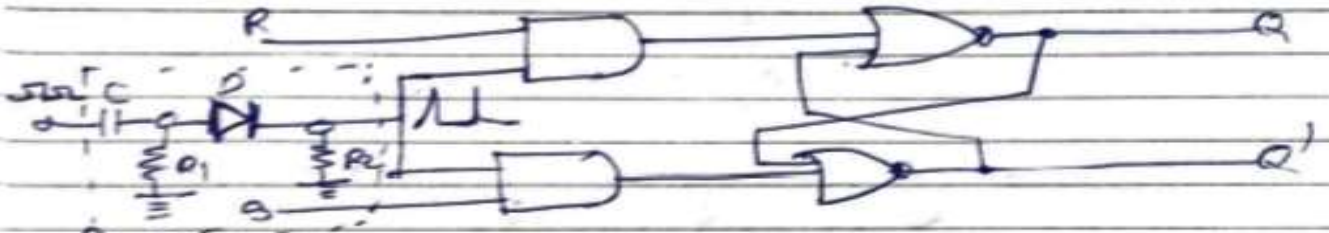


These spikes can be derived from the rectangular clock pulses with the help of a passive differentiator circuit.



Passive differentiator.

→ Positive edge triggered S-R flip-flop.



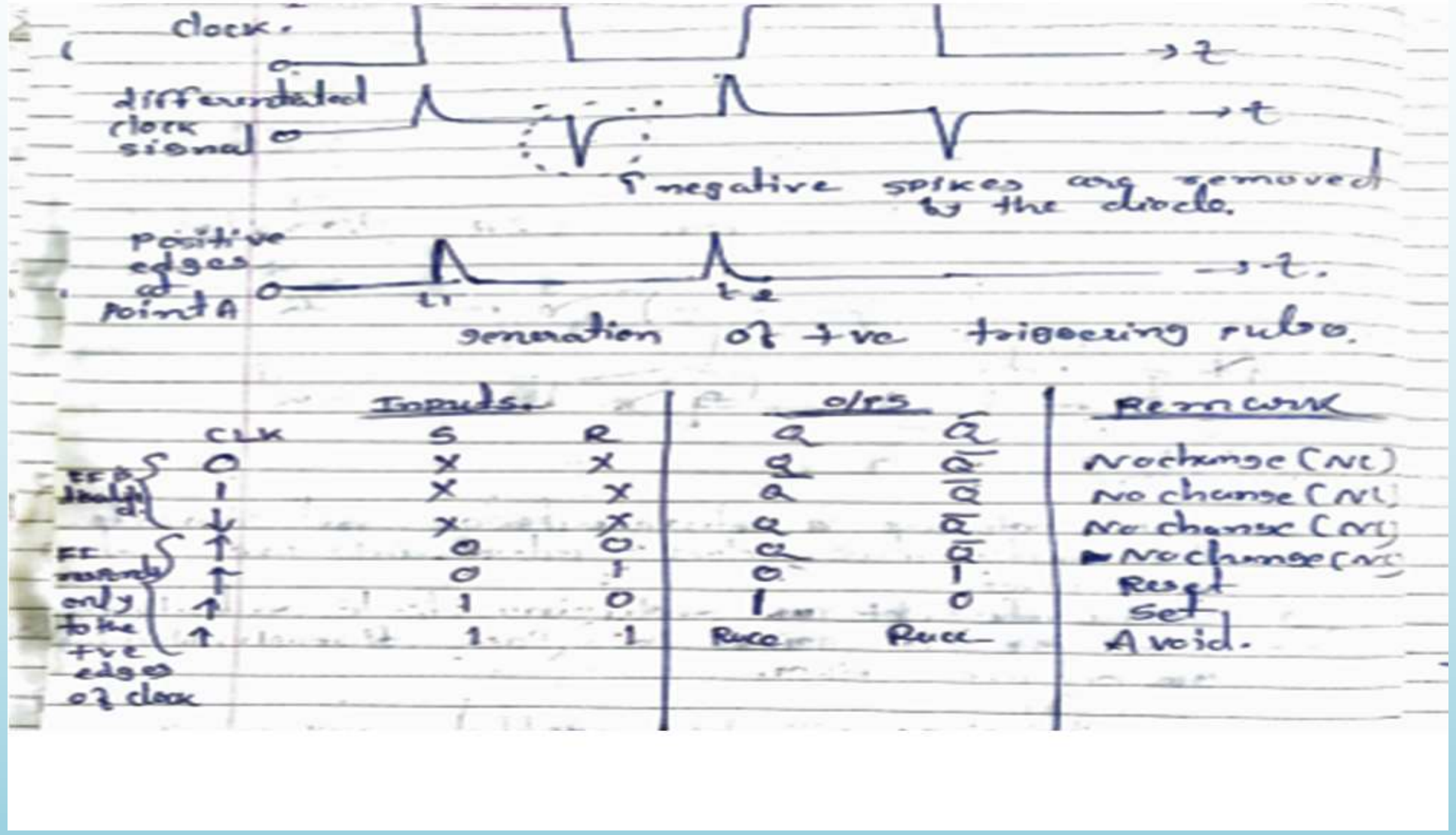
differentiator.

→ C-R_1 acts as a differentiator and converts the rectangular clock pulses into +ve and -ve spikes.

→ The diode acts as rectifying diode and allows only the +ve spikes to pass through, blocking the negative spikes.

→ S-R flip flop will be enabled only for a short duration of time when the positive spike is present at A

→ At these instants, the flip-flop behaviour is exactly same as that of enabled gated S-R latch.



	CLK	Inputs S	R	O/Ps Q	$\bar{Q}$	Remark
FF is disabled	0	X	X	Q	$\bar{Q}$	No change (NC)
	1	X	X	Q	$\bar{Q}$	No change (NC)
	↓	X	X	Q	$\bar{Q}$	No change (NC)
FF responds only to the +ve edges of clock	↑	0	0	Q	$\bar{Q}$	No change (NC)
	↑	0	1	0	1	Reset
	↑	1	0	1	0	Set
	↑	1	1	Race	Race	Avoid.

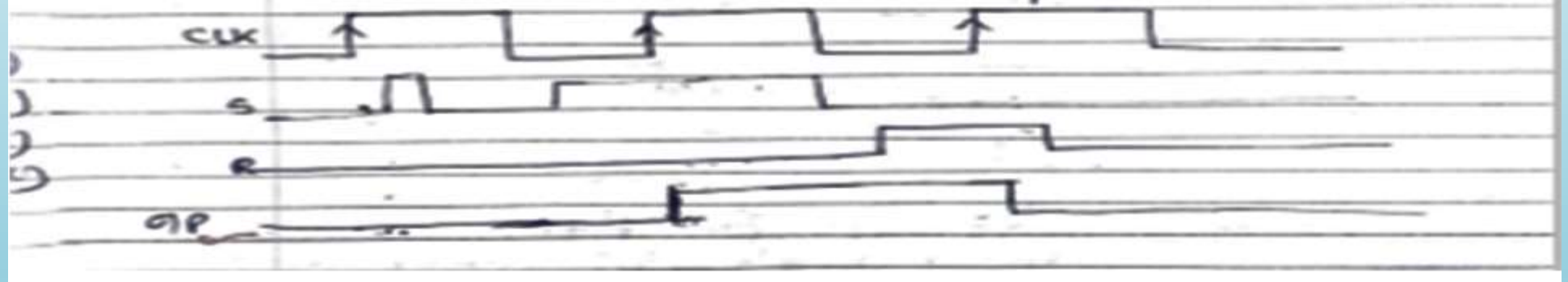
→ negative edge triggered $S\bar{-}R$ flip-flop.

- internally triggered circuit of the negative edge S-R flip flop is same as that of the +ve edge triggered one. only difference the circuit is slightly modified.

inputs			o/ps		state.
CLK	S	R	Q	$\bar{Q}$	
0	X	X	q	$\bar{q}$	No change
1	X	X	q	$\bar{q}$	No change
↑	X	X	q	$\bar{q}$	No change
↓	0	0	q	$\bar{q}$	No change
↓	0	1	0	1	Reset
↓	1	0	1	0	set
↓	1	1	)	)	avoid.

FF is holding

FF remains in its -ve edge

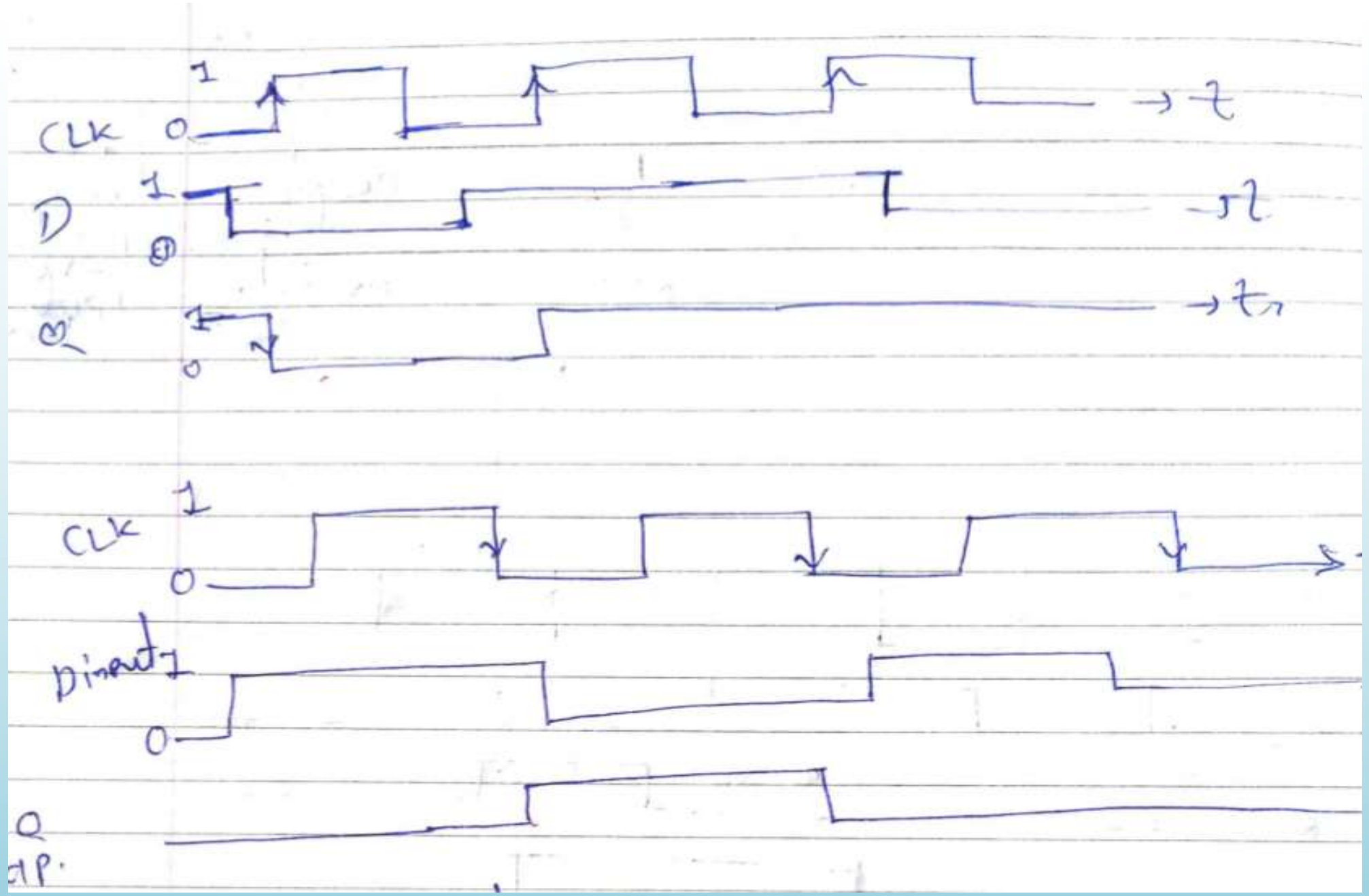


→ for the racing around to take place, it is necessary to have the enable input high along with J=K=1.

→ As the enable input remains high for a long time in JK latch, the problem of multiple toggling arises.

→ But in edge triggered JK flip-flop, the +ve clock pulse is present only for a very short time so multiple toggling can not take place.

So edge triggering avoids the race around condition.

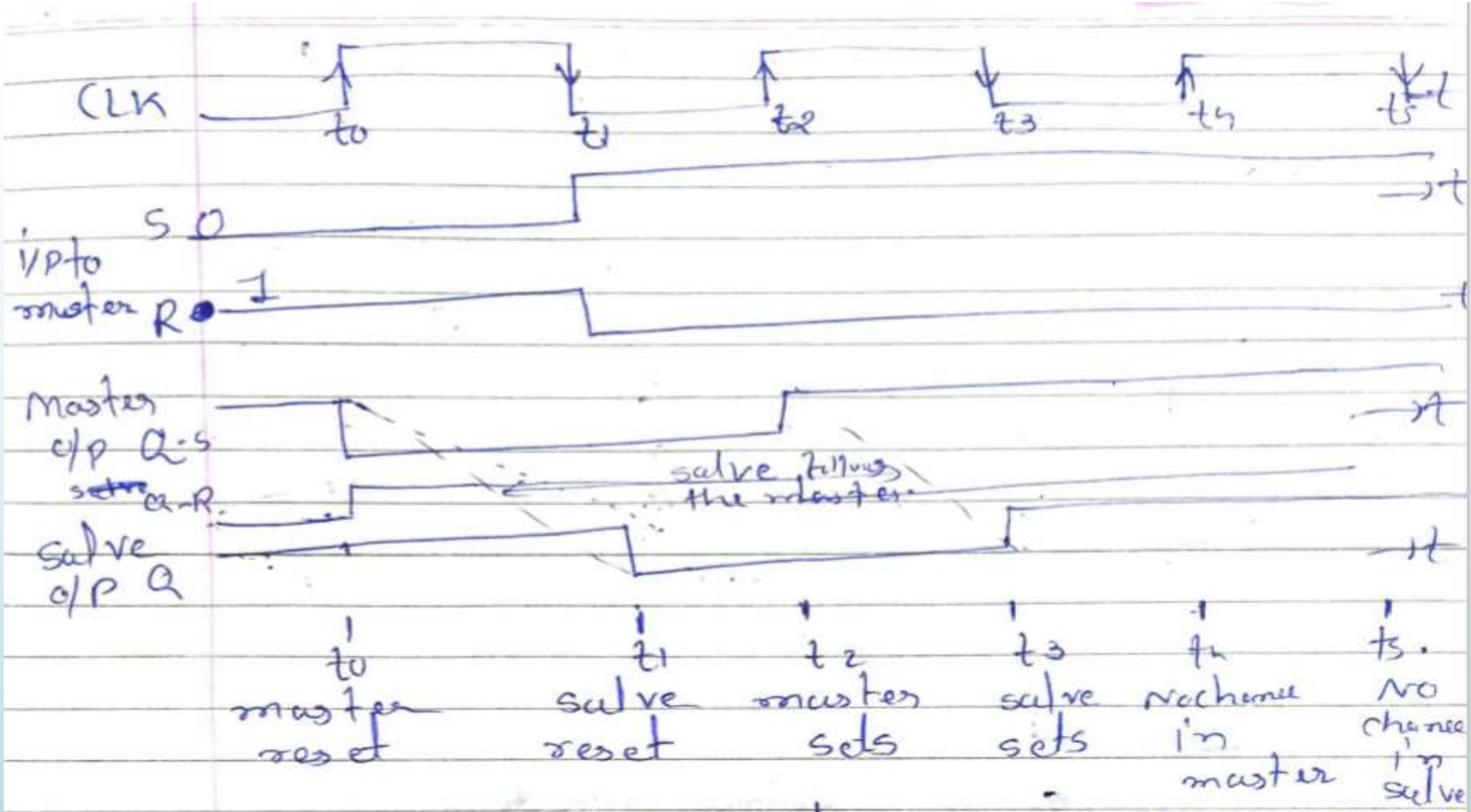

CLK
1
0
D
1
0
Q
1
0
CLK
1
0
Dinput
1
0
Q
o/p.

Master salve SR flip - flop:

→ master slave, SR flip-flop Consists of two level triggered flip flops shown in fig.

positive level triggered master

Negative level triggered salve.



slave has followed the master. In other words the o/p state of master has been transferred to the o/p of salve.

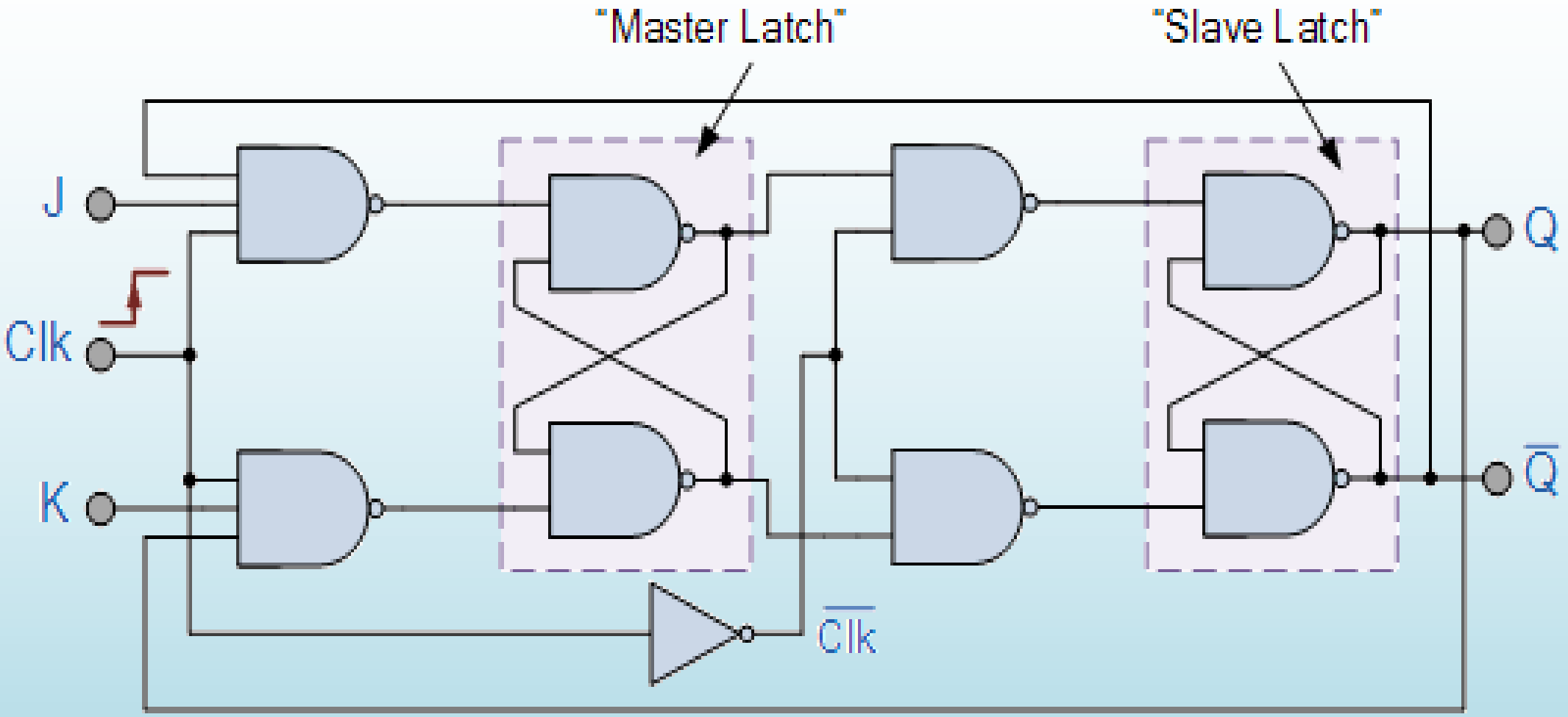
Master slave JK flip-flop.

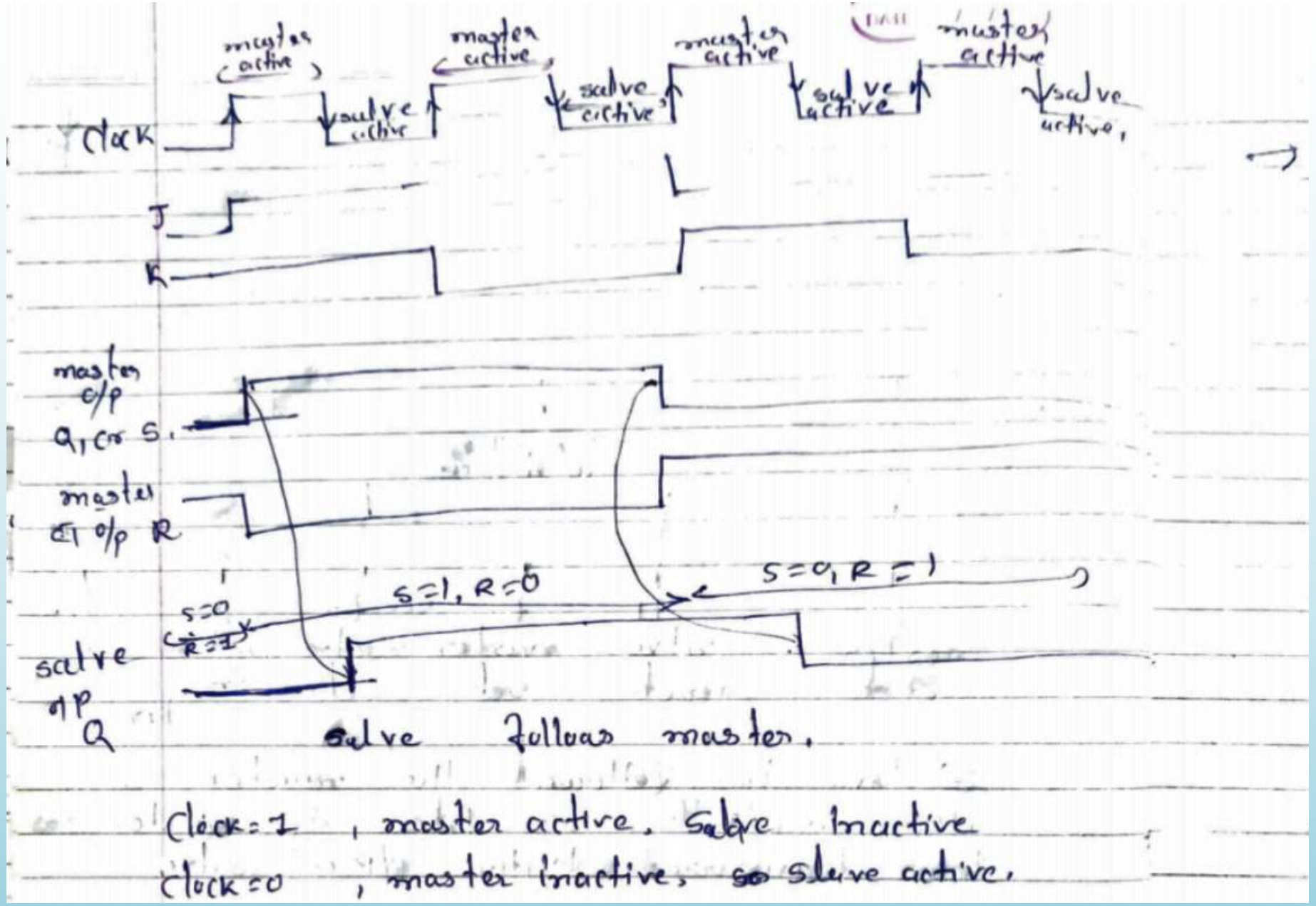
→ it is a combination of clocked JK flip flop and clocked SR flip flop.

→ clocked JK flip-flop acts as the master and the clocked SR latch acts as the slave.

→ master is +ve level triggered, slave is -ve level triggered because of inverter in the clock line.

→ clk = 1, master is active
clk = 0, salve is active.


"Master Latch"
"Slave Latch"
J
Clk
K
Q
$\overline{Q}$
$\overline{Clk}$


master active
slave active
Clock
J
R
master o/p Q1 or S.
master o/p R
S=1, R=0
S=0, R=1
S=0
R=1
slave o/p Q
Slave follows master.
Clock=1, master active. Slave inactive
Clock=0, master inactive, so Slave active.

Clock = 1 , master active, Salve Inactive
Clock = 0 , master inactive, so slave active.

changed o/p are returned back to the master inputs.

→ But since CLK = 0, master is still inactive. So it does not respond to these changed o/ps.

→ This avoids the multiple toggling which leads to the race around condition. Thus master slave flip-flop will avoid the race around condition.

→ salve always follows the master, after a delay of half clock cycle Period.

Flip-Flop Excitation Tables

The characteristic table is useful for analysis and for defining the operation of the flip-flop. It specifies the next state when the inputs and present state are known. During the design process, we usually know the transition from present state to next state and wish to find the flip-flop input conditions that will cause the required transition. For this reason, we need a table that lists the required inputs for a given change of state. Such a list is called an *excitation table*.

Flip-Flop Excitation Tables

$Q(t)$	$Q(t+1)$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

(a) *RS*

$Q(t)$	$Q(t+1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(b) *JK*

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

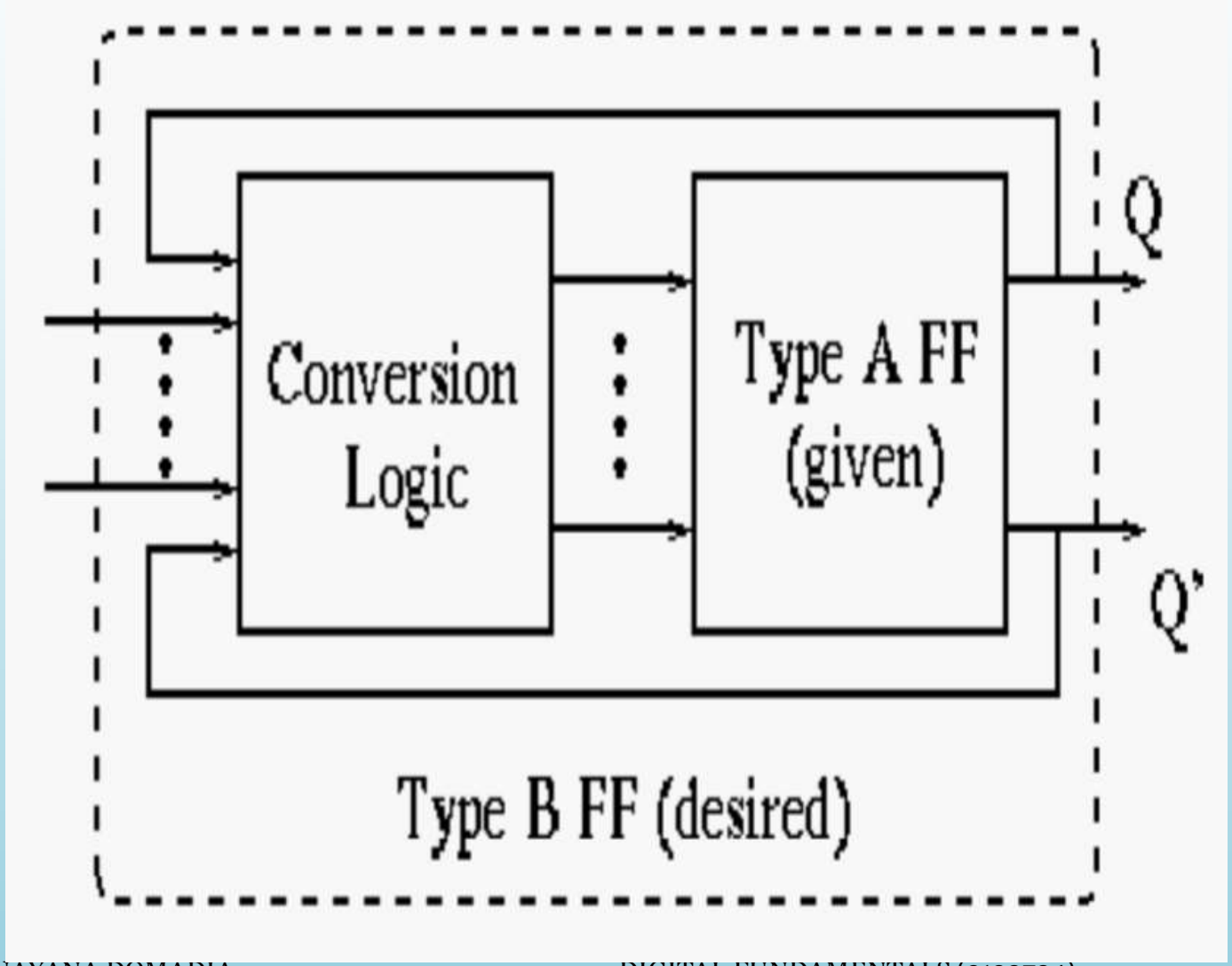
(c) *D*

$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

(b) *T*

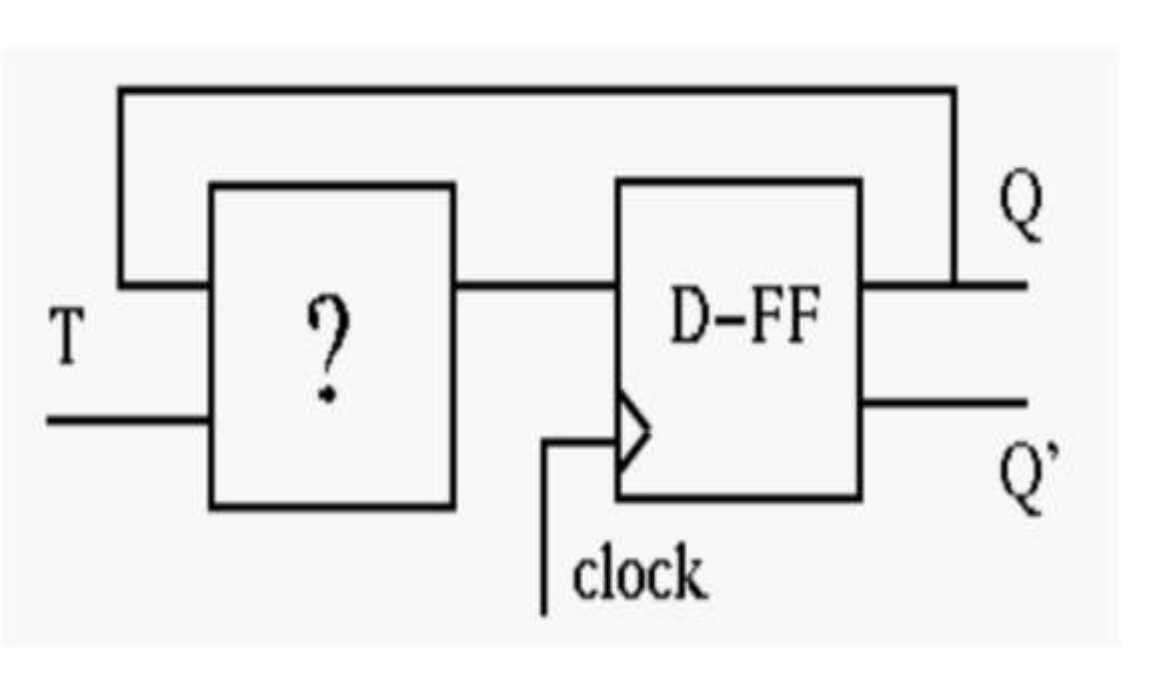
Flip-flop Conversions

The purpose is to convert a given type A FF to a desired type B FF using some conversion logic.



Convert a D-FF to a T-FF

The output of D flip flop should be as the output of T flip flop.



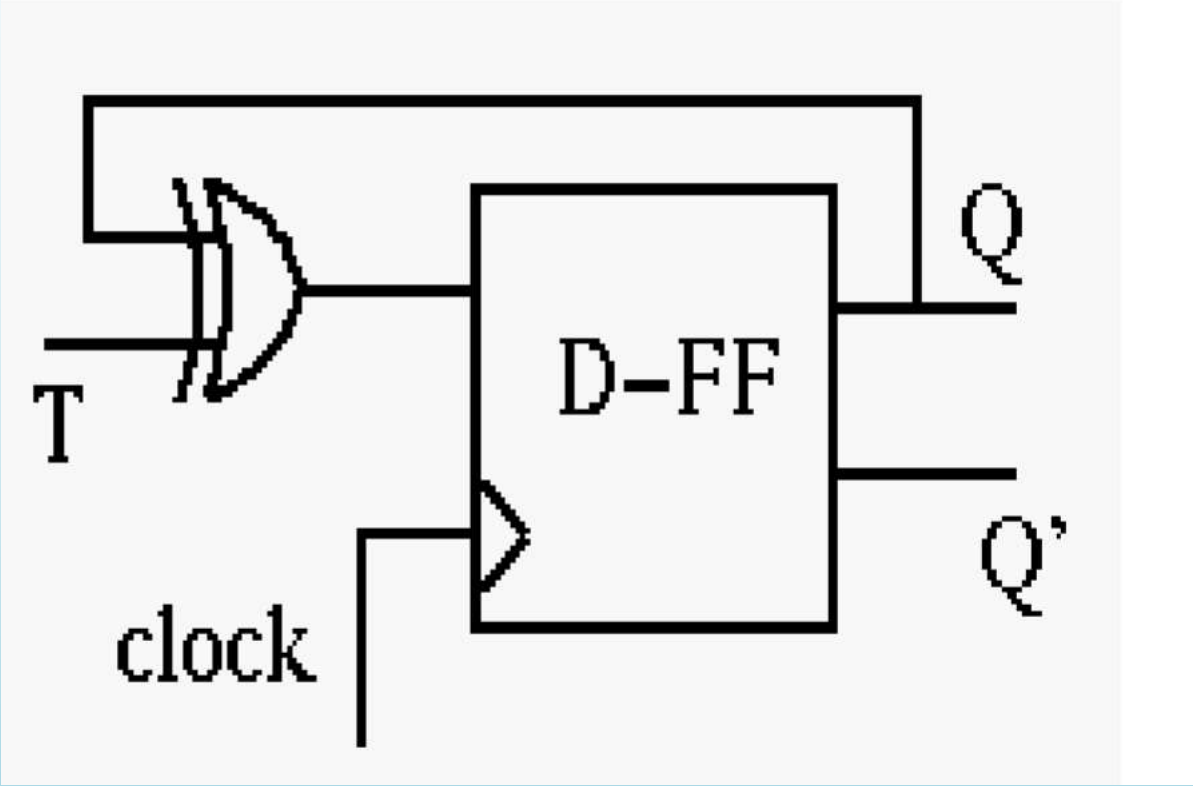
T	Q_n	Q_{n+1}	D
0	0	0	0
1	0	1	1
1	1	0	0
0	1	1	1

Consider the excitation table of T and D Flip flops. Write Down Excitation Table of T, Qn and Qn+1, D.

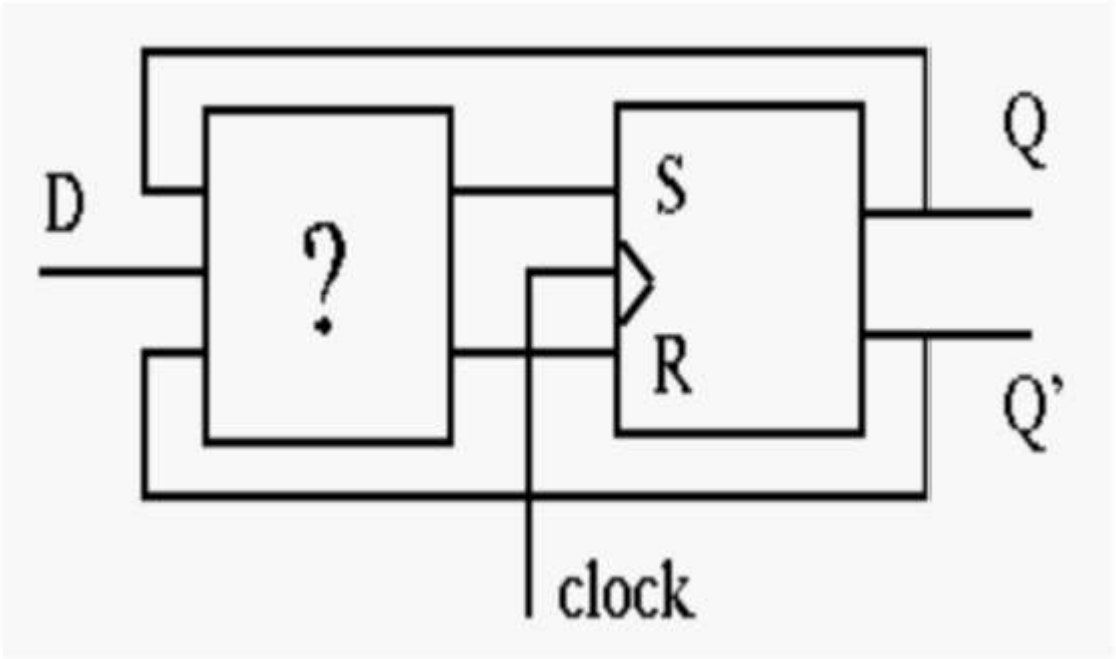
For the K-map, consider T and Qn As Input and D as output. D = TQn' + T'Qn (Ex- OR gate)

Treating as a function of and current FF state Q (Qt), we have:

$$D = T'Q + TQ' = T \oplus Q$$



Convert a RS-FF to a D-FF

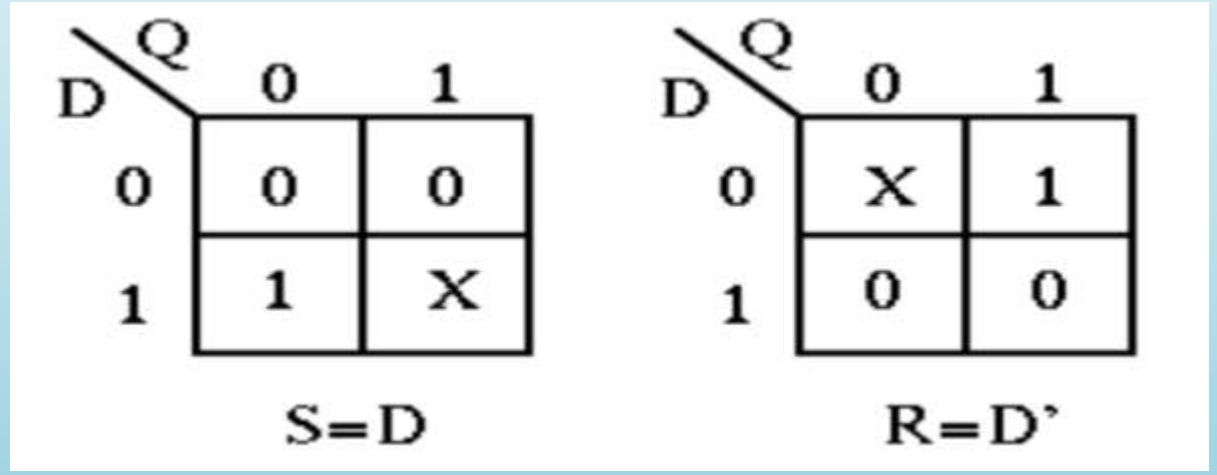


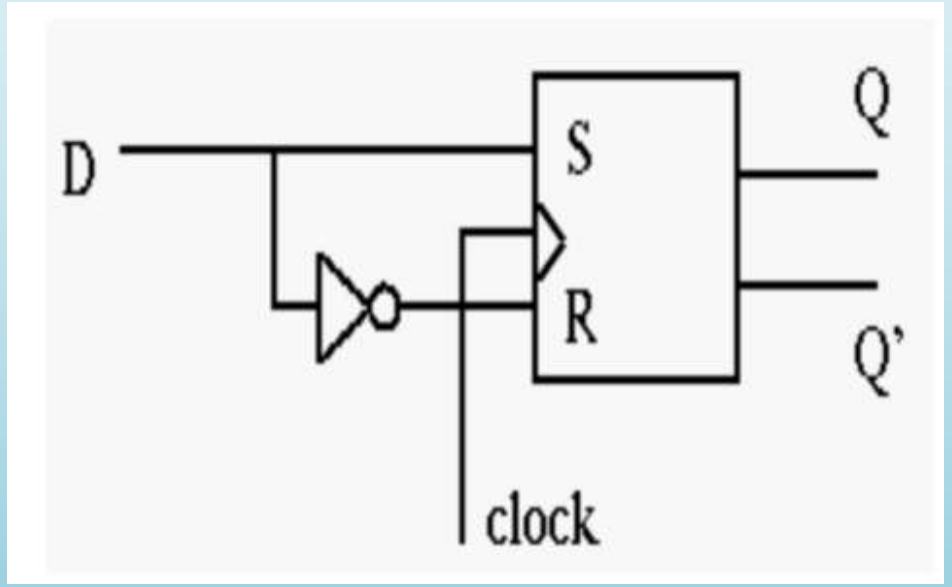
We need to design the circuit to generate the triggering signals S and R as functions of D and Q.

Consider the excitation table

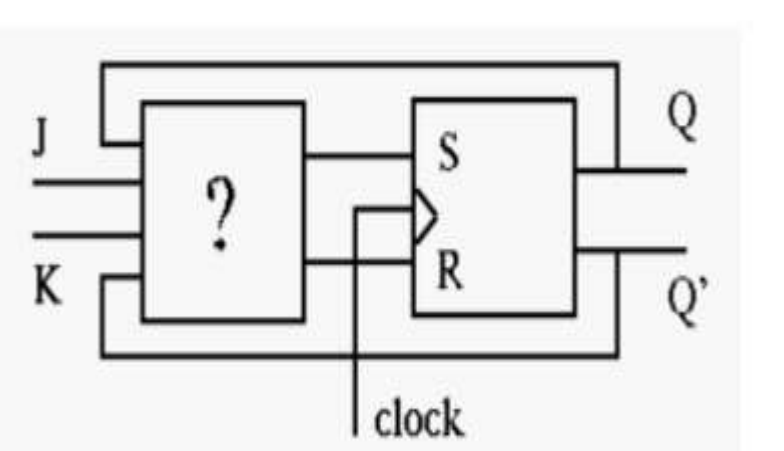
D	Q_t	Q_{t+1}	S	R
0	0	0	0	x
1	0	1	1	0
0	1	0	0	1
1	1	1	x	0

The desired signal S and R can be obtained as functions of D and Q current FF state from the Karnaugh maps:





Convert a RS-FF to a JK-FF



We need to design the circuit to generate the triggering signals S and R as functions of J, K and Q. Consider the excitation table:

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

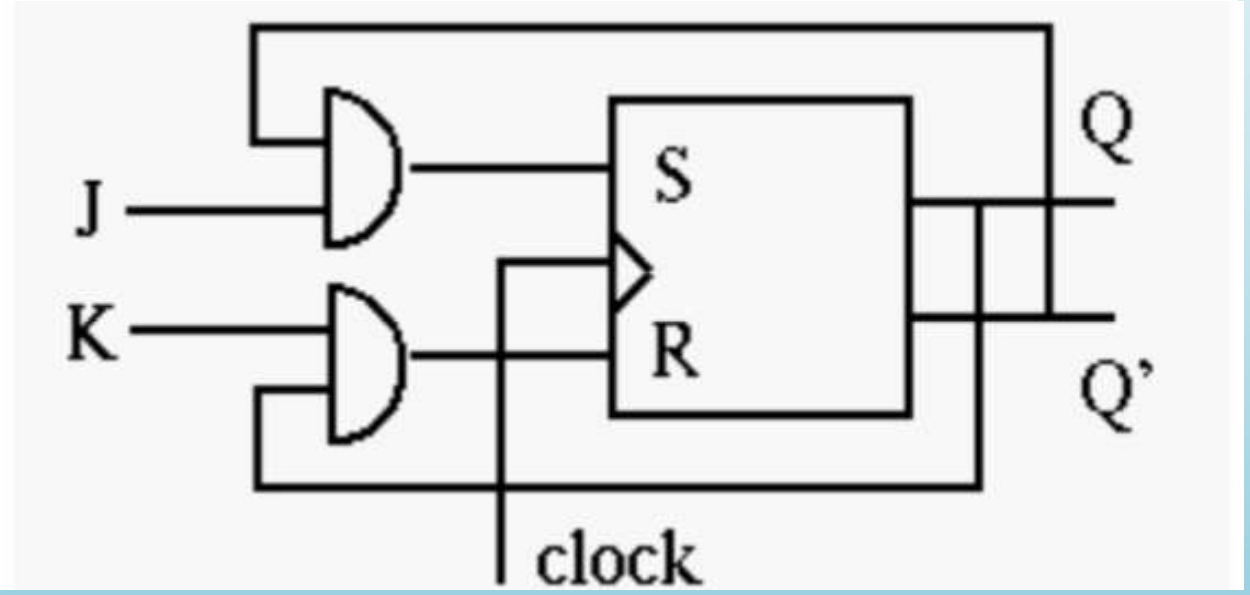
The desired signal S and R as functions of J, K and current FF state Q can be obtained from the Karnaugh maps:

K \ QJ	00	01	11	10
0	0	1	X	X
1	0	1	0	0

S=Q'J

K \ QJ	00	01	11	10
0	X	0	0	0
1	X	0	1	1

R=QK



Registers

- As a flip-flop (FF) can store only one bit of data, a 0 or a 1, it is referred to as a single-bit register.
- A register is a set of FFs used to store binary data.
- The storage capacity of a register is the number of bits (1s and 0s) of digital data it can retain.

Registers

- Loading a register means setting or resetting the individual FFs, i.e. inputting data into the register so that their states correspond to the bits of data to be stored.
- Loading may be serial or parallel.
- In serial loading, data is transferred into the register in serial form i.e. one bit at a time.
- In parallel loading, the data is transferred into the register in parallel form meaning that all the FFs are triggered into their new states at the same time.

Types of Registers

1. Buffer register
2. Shift register
3. Bidirectional shift register
4. Universal shift register

Buffer Register

Buffer registers are a type of registers used to store a binary word. These can be constructed using a series of flip-flops as each flip-flop can store a single bit. This means that in order to store an n-bit binary word one should design an array of n flip-flops. Figure 1 shows a 4 bit synchronous buffer register formed by cascading four positive edge triggered D flip-flops. Here the entire input data word X1 X2 X3 X4 is loaded onto the register at a single clock tick

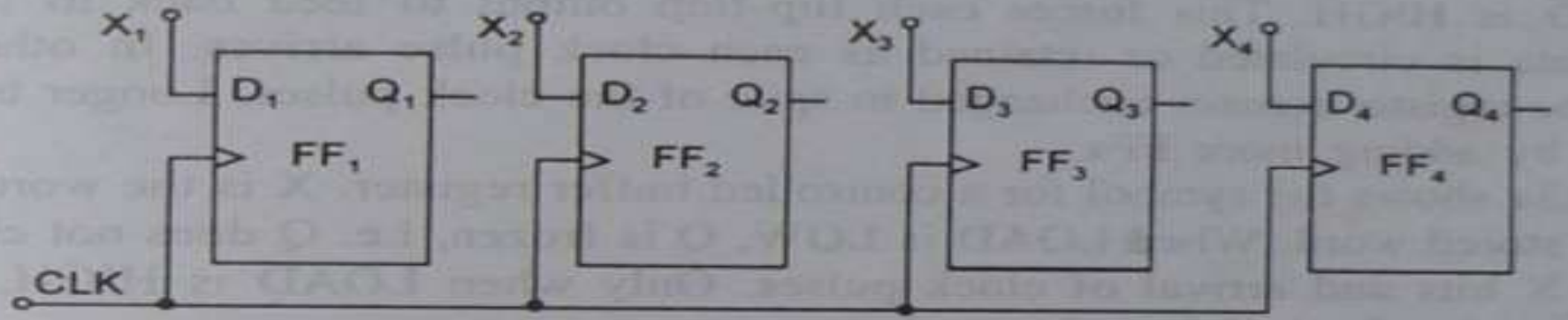


Fig. 9.1 Logic diagram of a 4-bit buffer register.

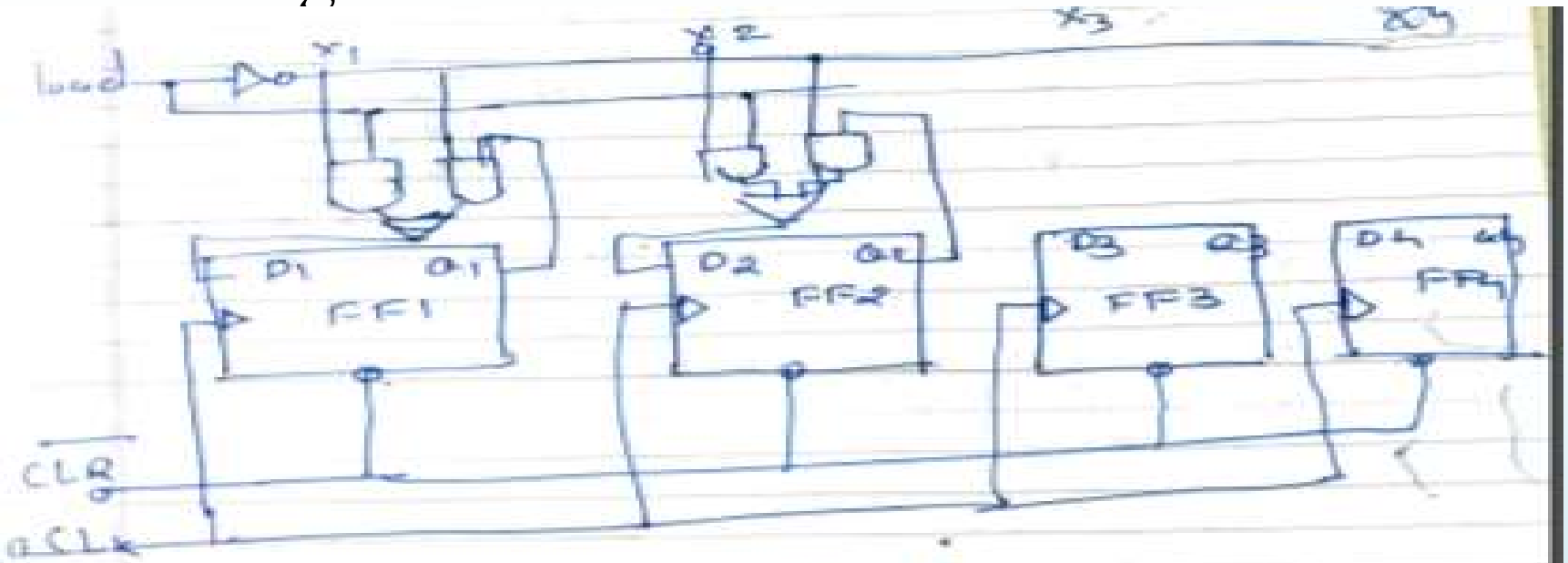
When the positive clock edge arrives, the stored word becomes

$$Q_4\ Q_3\ Q_2\ Q_1 = X_4\ X_3\ X_2\ X_1$$

or

$$Q = X$$

Controlled Buffer Register



→ if $\overline{CLR}$ goes low, output Q = 0000
→ when CLR is High, register is ready for action
now LOAD is control i/p.

LOAD is ↑ register is loaded.

$Q_4\ Q_3\ Q_2\ Q_1 = x_4 x_3 x_2 x_1$

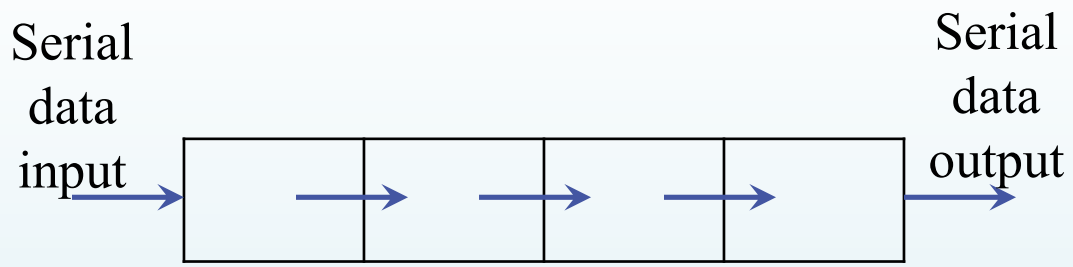
$Q = X$

LOAD is ↓ This forces each flip-flo output to feedback to its data i/p. Therefor data is circulated and remains unchanged in spite of the clock pulse

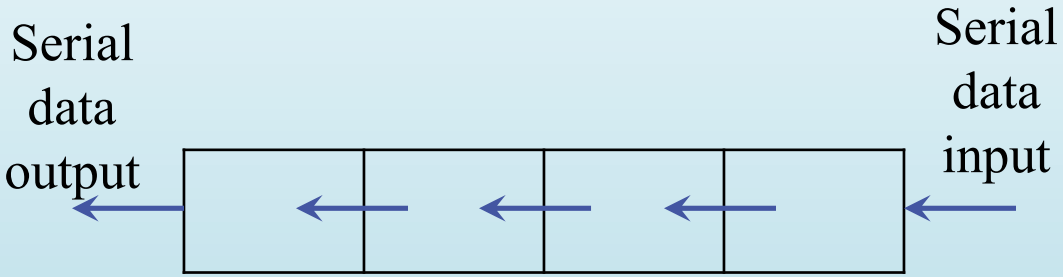
Shift Register

- A number of FFs connected together such that data may be shifted into and shifted out of them is called a shift register.
- Data may be shifted into or out of the register either in serial form or in parallel form.
- So, there are four basic types of shift registers:
  1. serial-in, serial-out
  2. serial-in, parallel out
  3. parallel-in, serial-out
  4. parallel-in, parallel-out
- Data may be rotated left or right. Data may be shifted from left to right or right to left at will, i.e. in a bidirectional way.
- Also, data may be shifted in serially (in either way) or in parallel and shifted out serially (in either way) or in parallel.

Data transmission in shift register

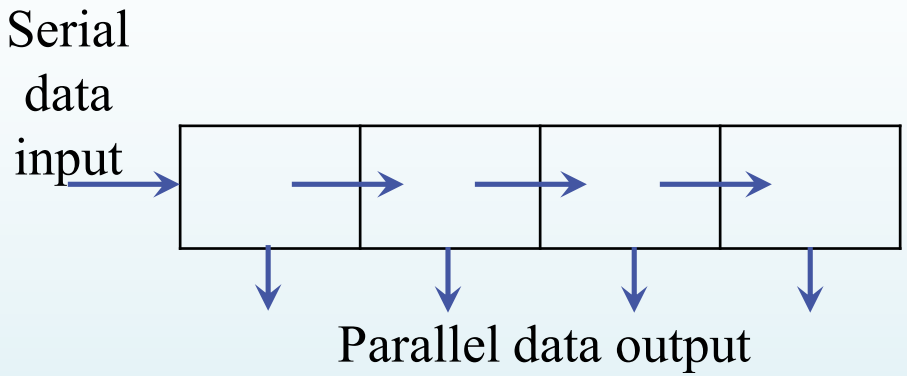


Serial-in, serial-out shift-right, shift register

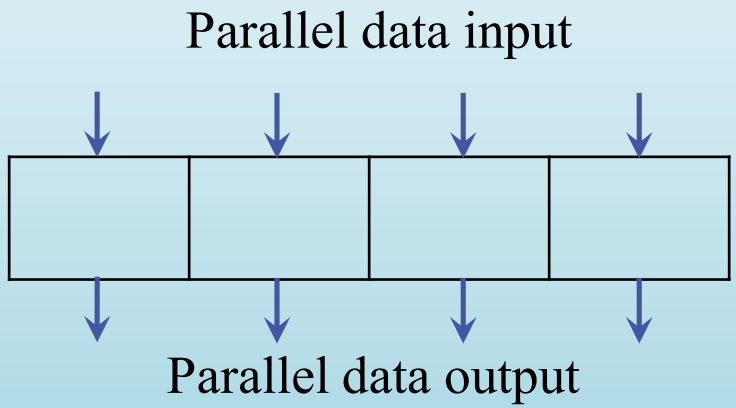


Serial-in, serial-out shift-left, shift register

Data transmission in shift register

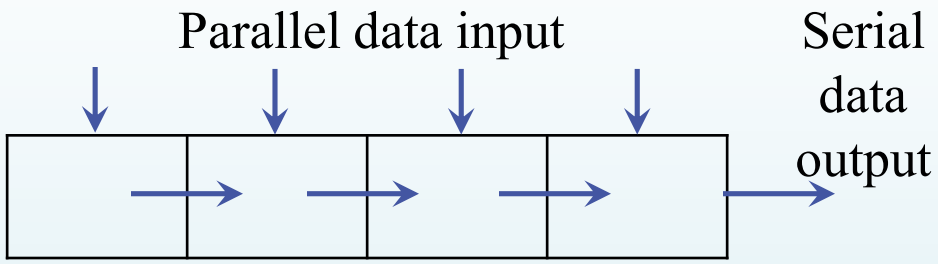


Serial-in, parallel-out, shift register



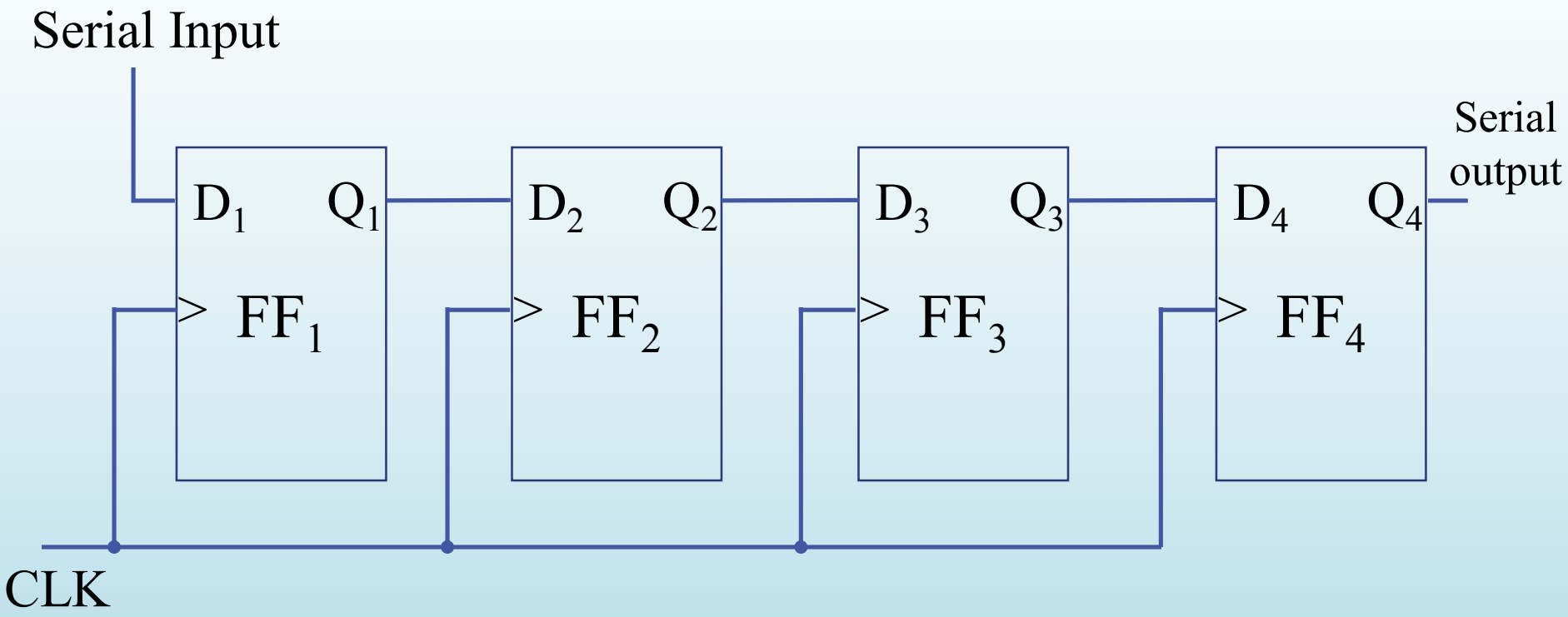
Parallel-in, parallel-out, shift register

Data transmission in shift register



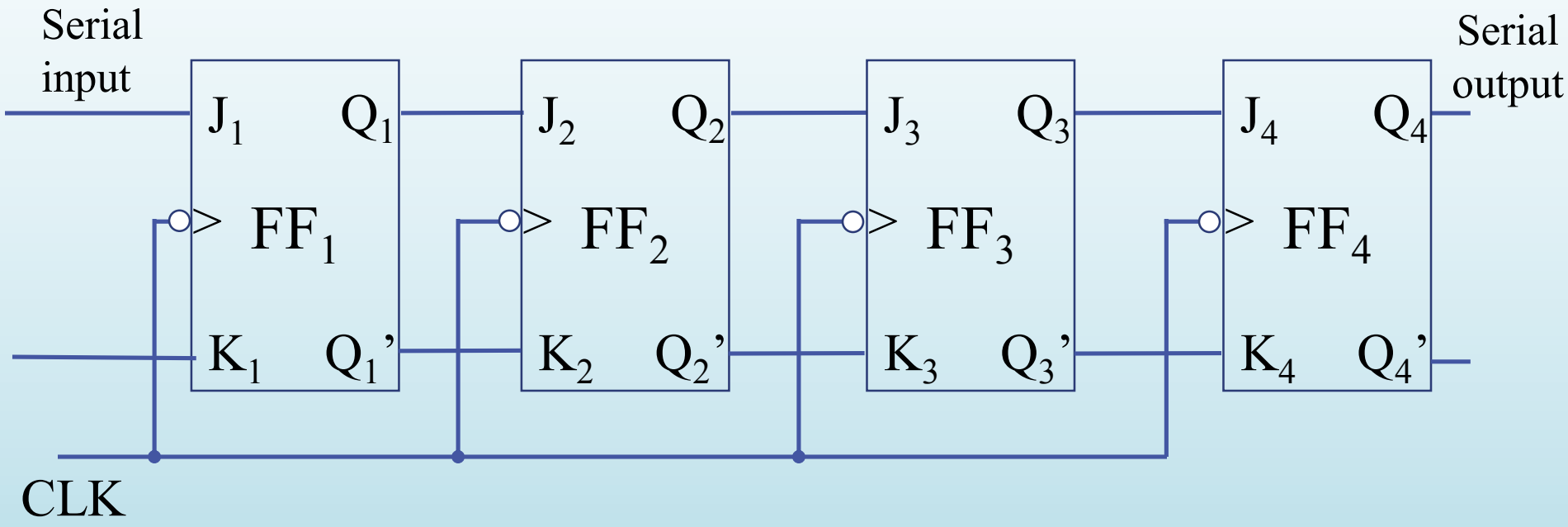
Parallel-in, serial-out, shift register

Serial-in, Serial-out, Shift register

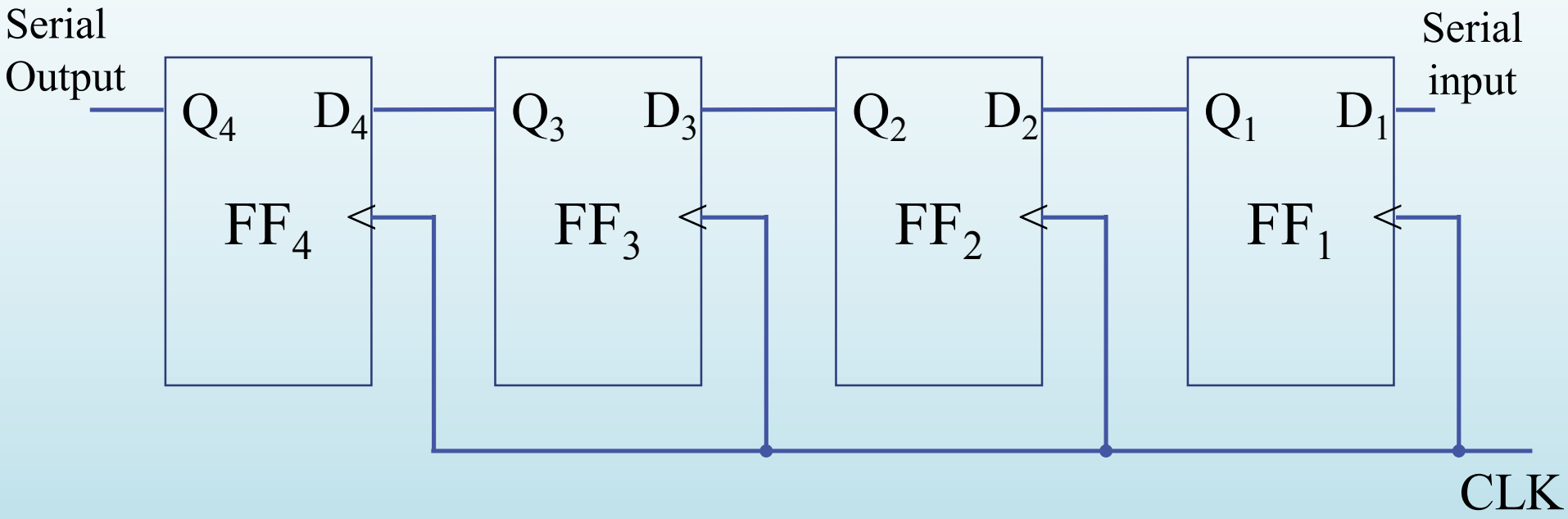


Serial-in, Serial-out, Shift register

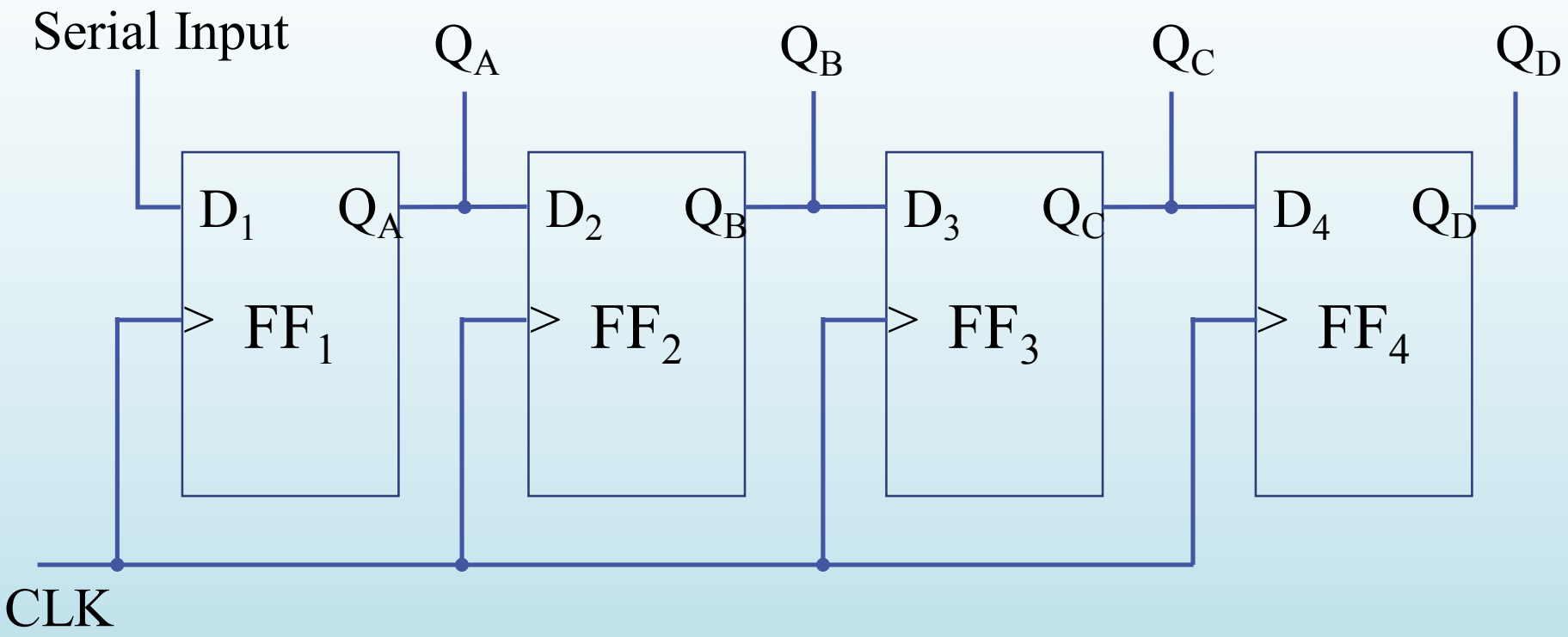
Using J-K Flip Flop



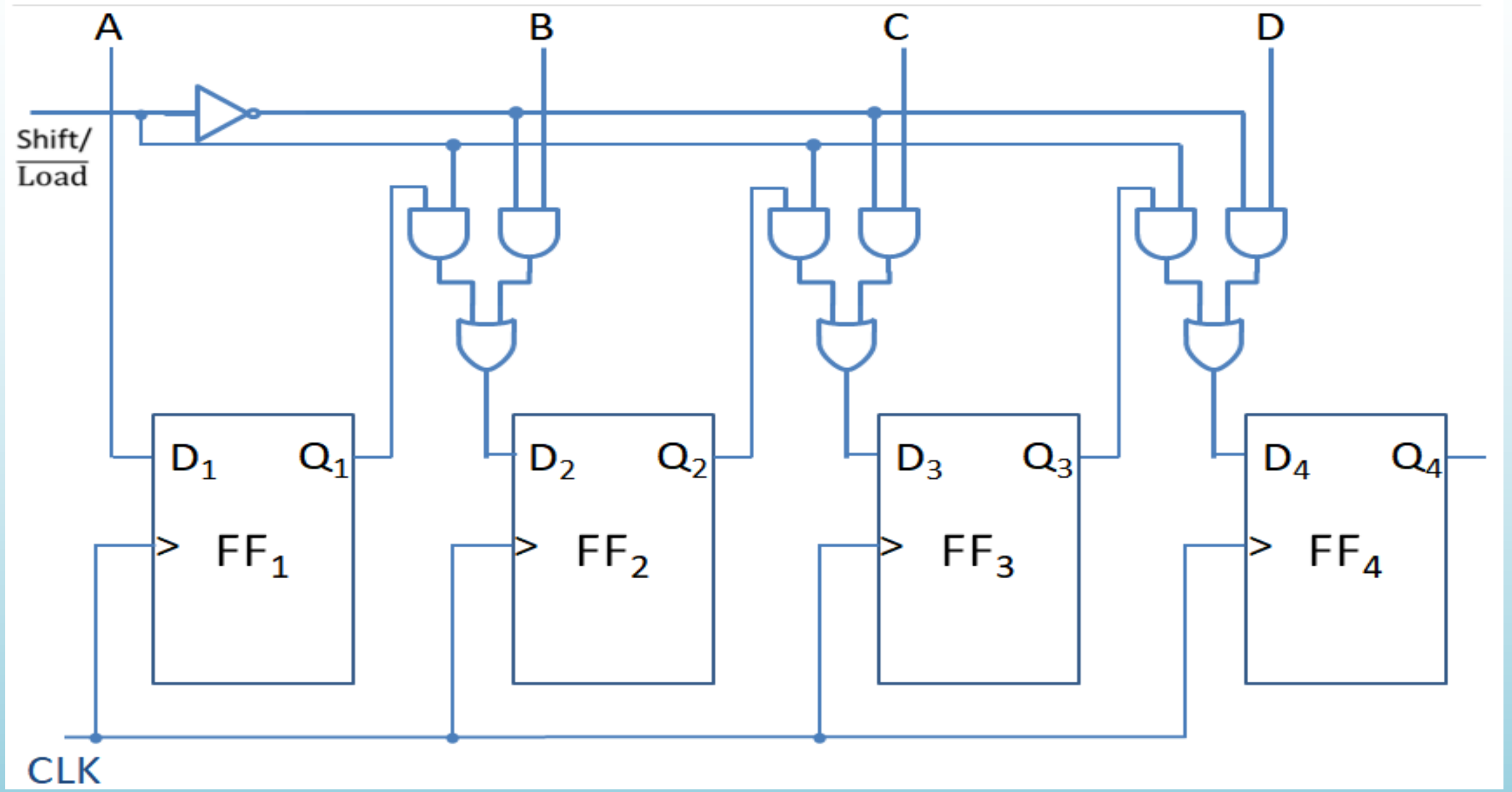
Serial-in, Serial-out, Shift-left, Shift register



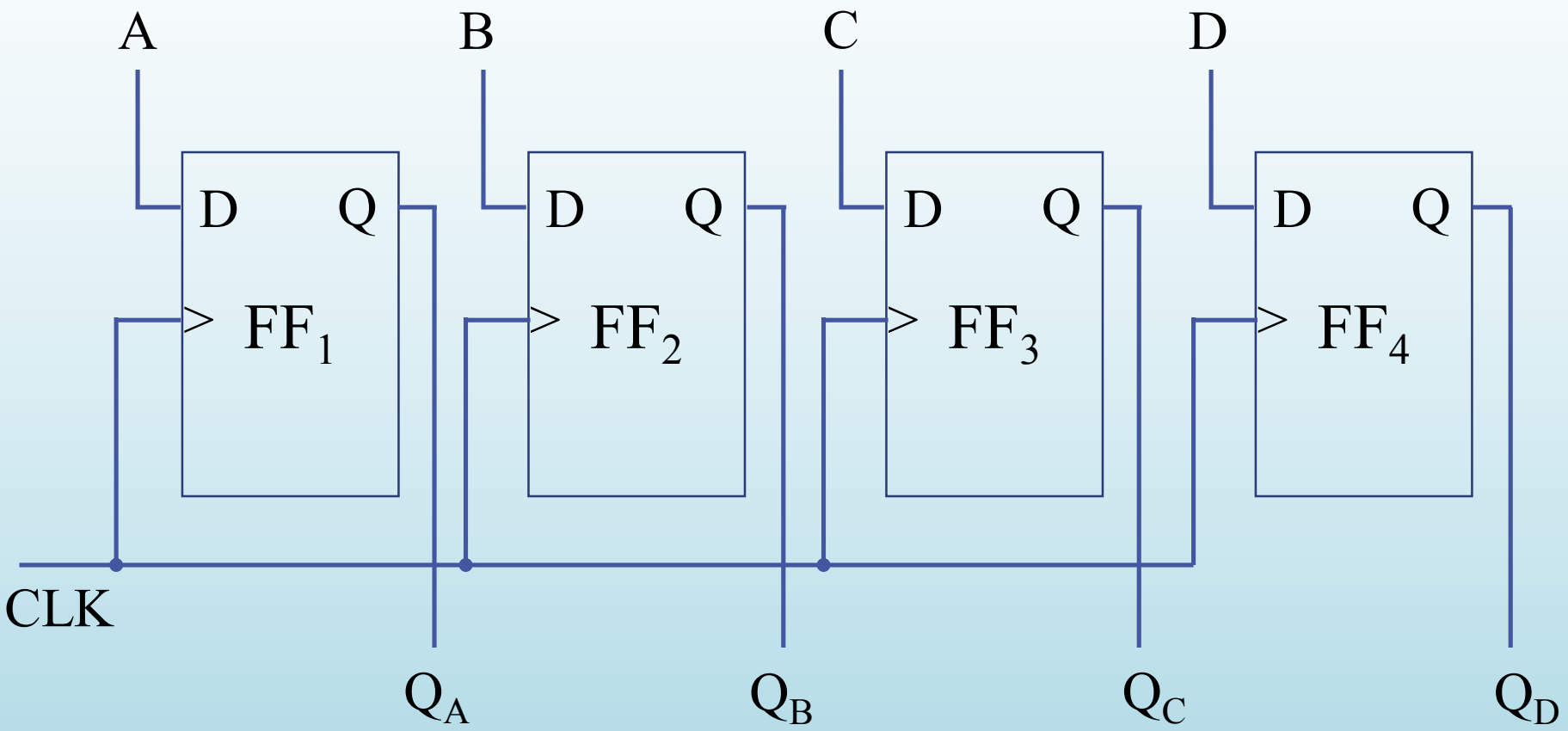
Serial-in, Parallel-out, Shift register



Parallel-in, Serial-out, Shift register

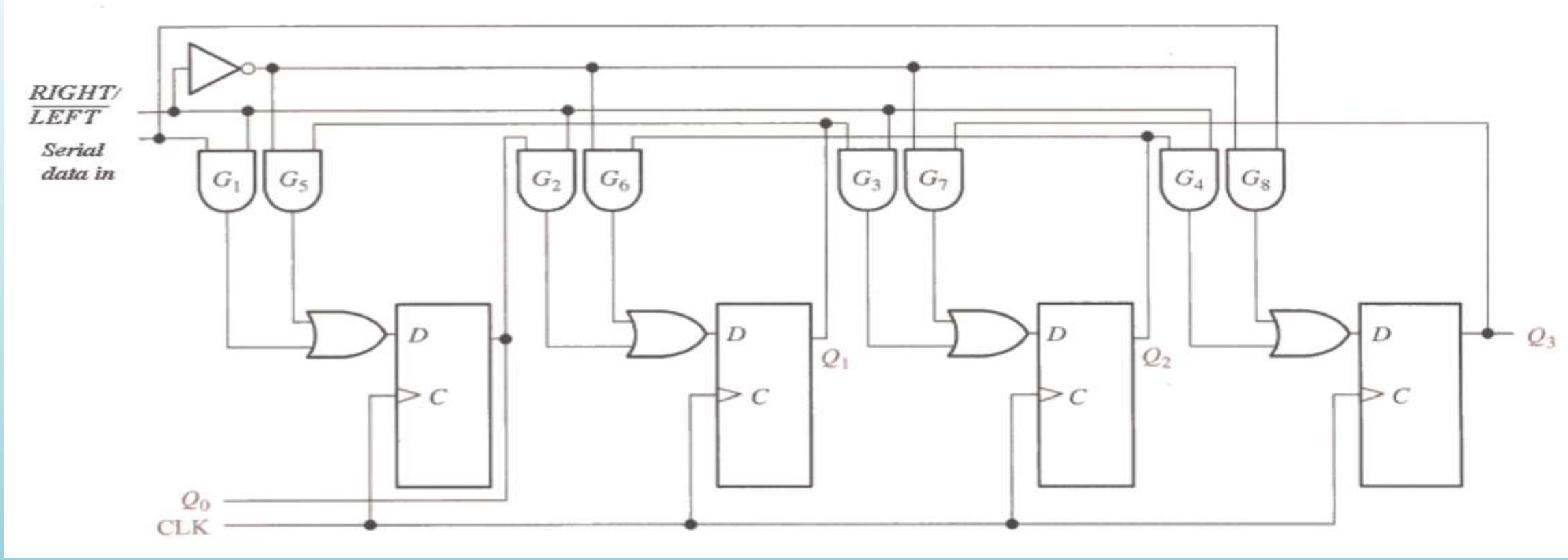


Parallel-in, Parallel-out, Shift register



Bi directional Shift Register

Bidirectional shift registers are the storage devices which are capable of shifting the data either right or left depending on the mode selected. Figure 1 shows an n-bit bidirectional shift register with serial data loading and retrieval capacity



With Right/Left=1- Right operation 1 ,2 ,3,4 are enabled

With Right/Left=0- Left operation 5,6,7,8 are enabled

Universal Shift Register

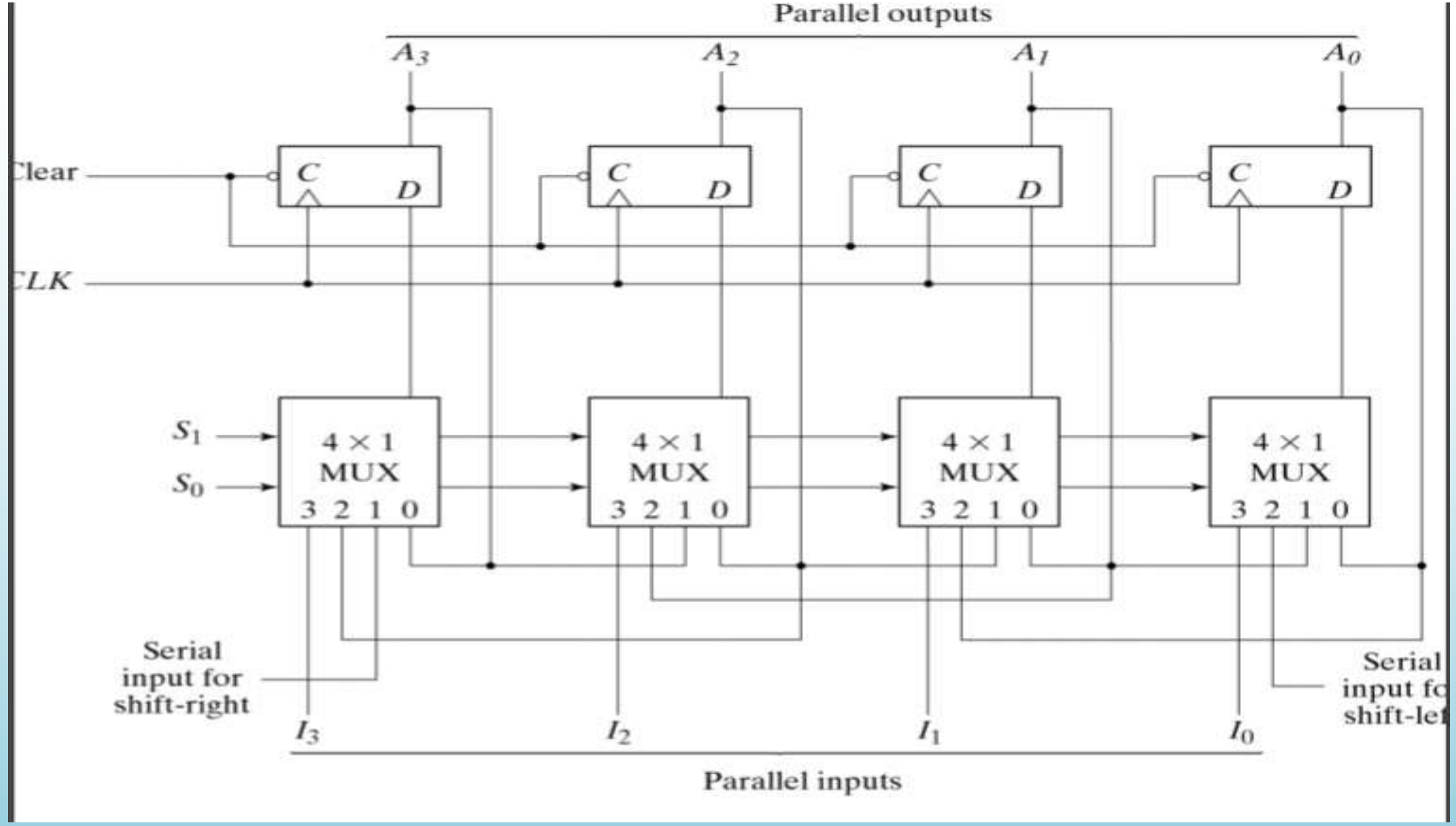
- A clear control to clear the register to 0.
- A clock input to synchronize the operations.
- A shift-right control to enable the shift operation and the serial input and output lines associated with the shift right.
- A shift-left control to enable the shift operation and the serial input and output lines associated with the shift left.

- A parallel-load control to enable a parallel transfer and the n input lines associated with the parallel transfer.

- n parallel output lines.

- A control state that leaves the information in the register unchanged in the presence of the clock.

- If the register has both shifts and parallel load capabilities, it is referred to as a universal shift register.


Parallel outputs
A_3
A_2
A_1
A_0
Clear
CLK
C
D
S_1
S_0
4 × 1
MUX
3 2 1 0
Serial
input for
shift-right
Serial
input fo
shift-lef
I_3
I_2
I_1
I_0
Parallel inputs

Mode Control		*Register* Operation
S_1	S_0	
0	0	No Change
0	1	Shift right
1	0	Shift Left
1	1	Parallel load

Applications of Shift Registers

Registers are used in digital electronic devices like computers as
Temporary data storage
Data transfer
Data manipulation
As counters.

Counter

- "A register that goes through a prescribed sequence of distinct states upon the application of a sequence of input pulses is called a counter."

- The input pulses could be the clock or some other input that occurs when the next step in the count should occur.

- A counter that follows the binary number sequence is called a *binary counter.*
- **Counters are of two types.**
  - Asynchronous or ripple counters.
  - Synchronous counters.

Asynchronous Counters v/s Synchronous Counters

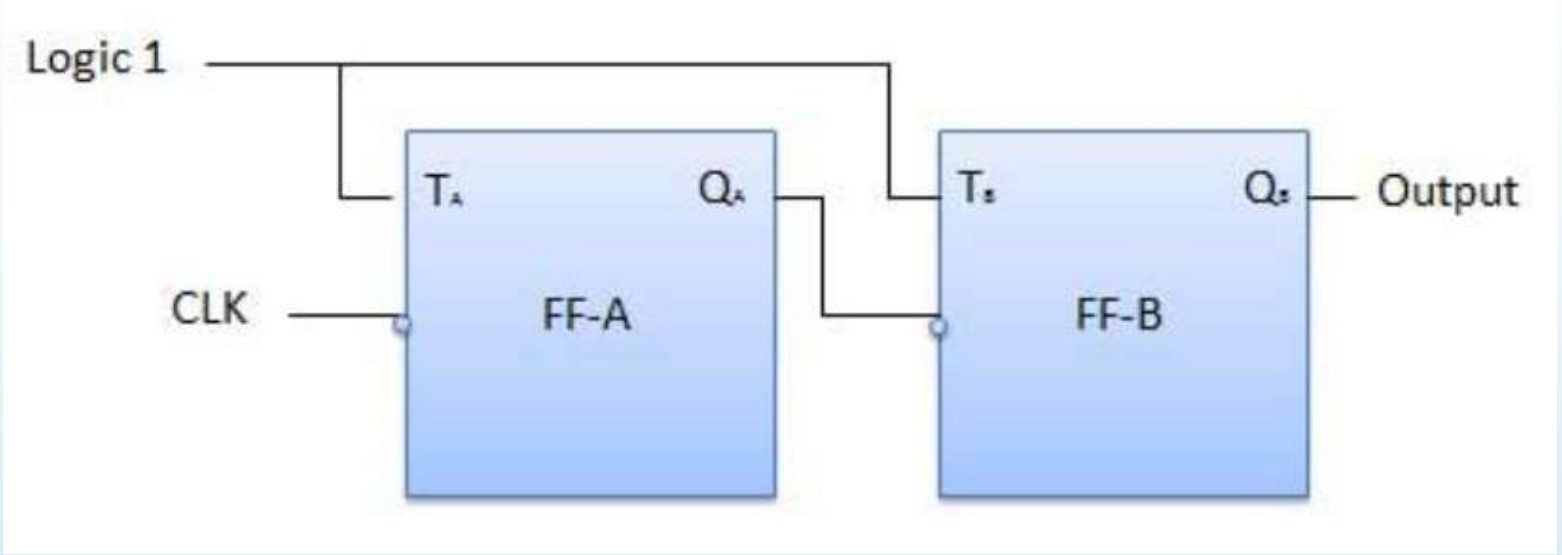
Asynchronous Counters

- In this type of counters FFs are connected in such a way that the output of the first FF drives the clock for the second FF, the output of the second the clock of the third and so on.
- All the FFs are not clocked simultaneously.
- Design and implementation is very simple even for more number of states.
- Main drawback of these counters is their low speed as the clock is propagated through a number of FFs before it reaches the last FF.

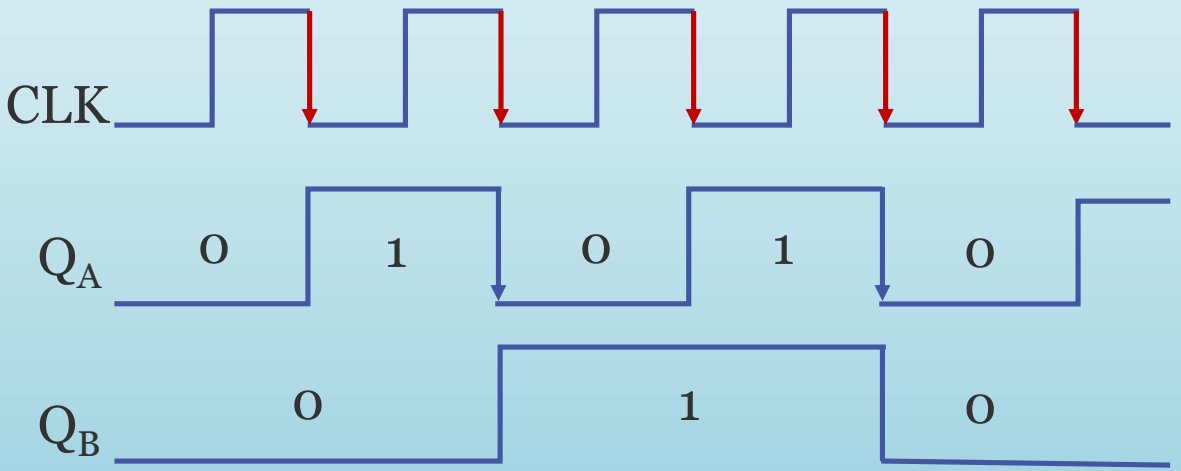
Synchronous Counters

- In this type of counters there is no connection between the output of first FF and clock input of next FF and so on.
- All the FFs are clocked simultaneously.
- Design and implementation becomes tedious and complex as the number of states increases.
- Since clock is applied to all the FFs simultaneously the total propagation delay is equal to the propagation delay of only one FF. Hence they are faster.

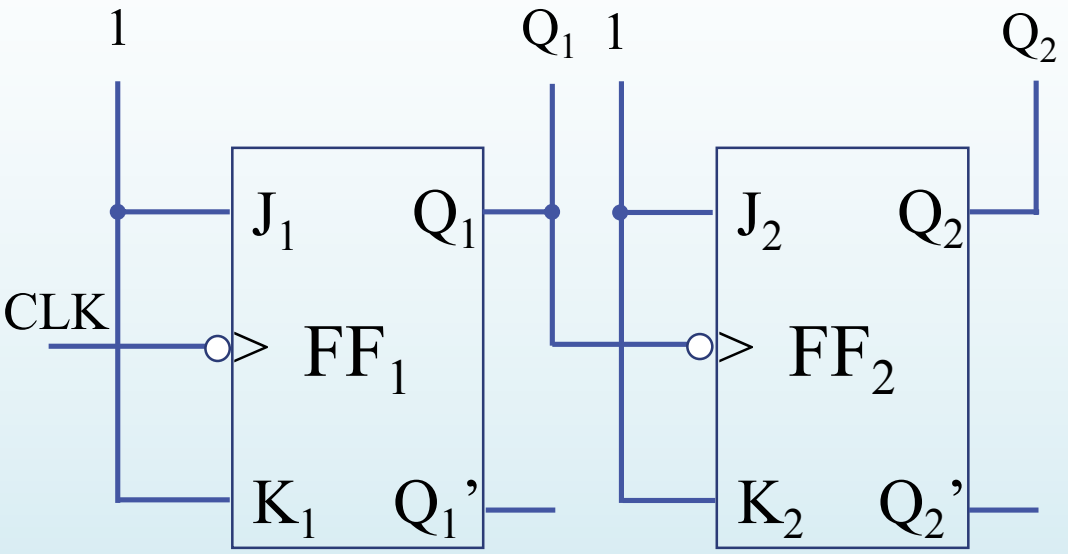
Asynchronous or ripple counters



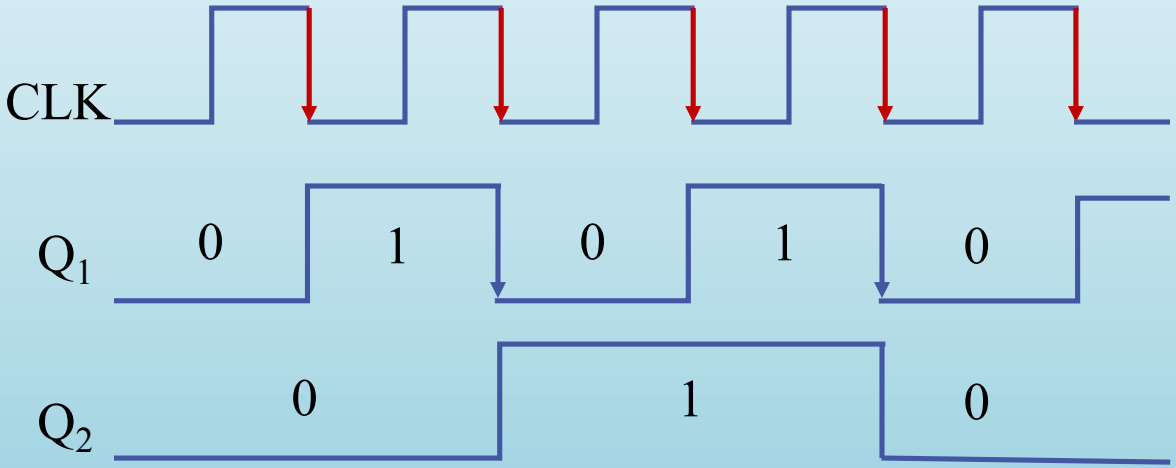
Clock	Counter output		State number	Deciimal Counter output
	Q_B	Q_A		
Initially	0	0	—	0
1st	0	1	1	1
2nd	1	0	2	2
3rd	1	1	3	3
4th	0	0	4	0



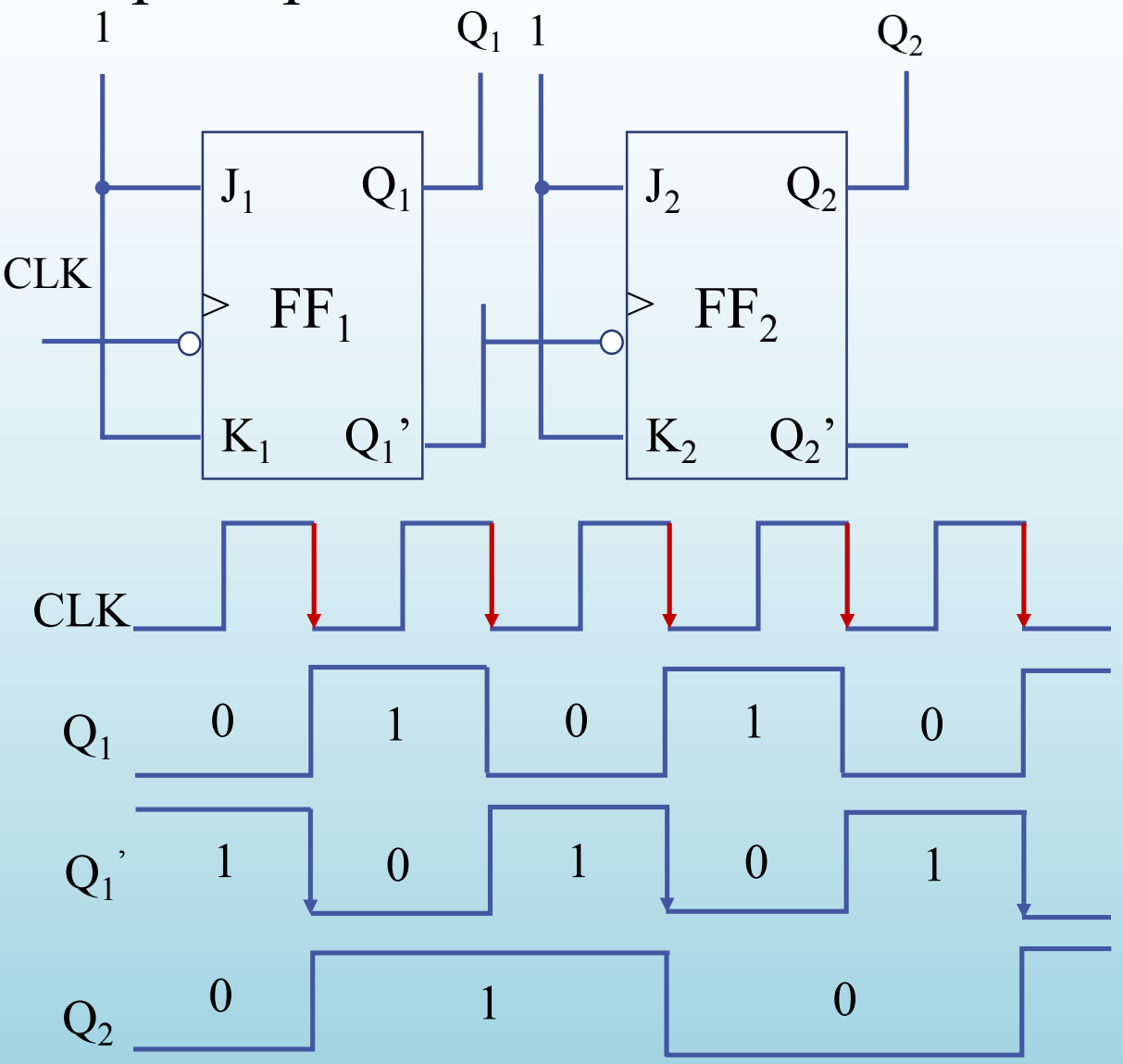
2-bit Ripple Up-Counter using Negative Edge-triggered Flip-Flop



CLK	Counter output	
	Q_2	Q_1
initially	0	0
↓	0	1
↓	1	0
↓	1	1
↓	0	0

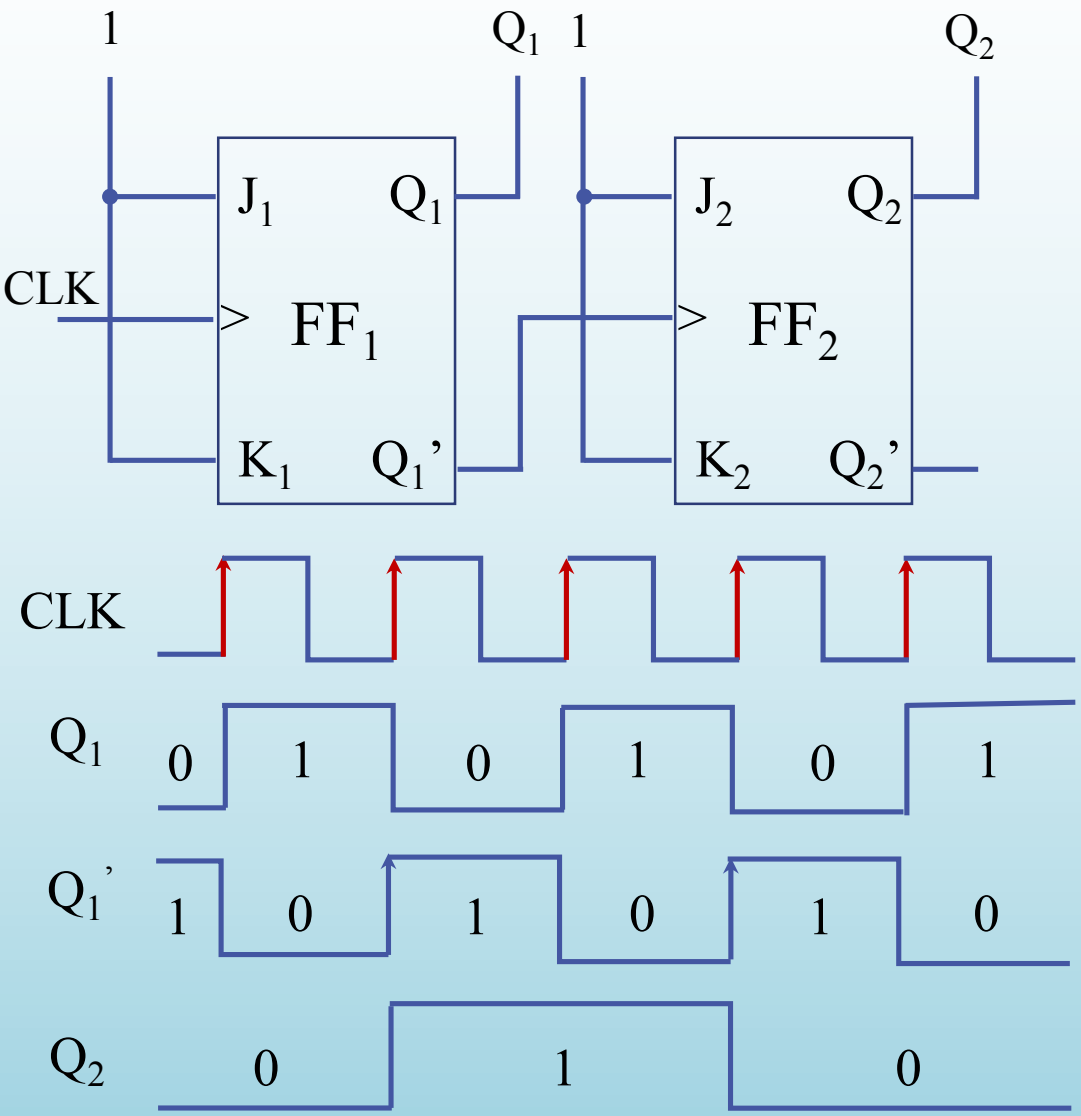


2-bit Ripple Down-Counter using Negative Edge-triggered Flip-Flop



CLK	Present State	
	Q_2	Q_1
	0	0
↓	1	1
↓	1	0
↓	0	1
↓	0	0

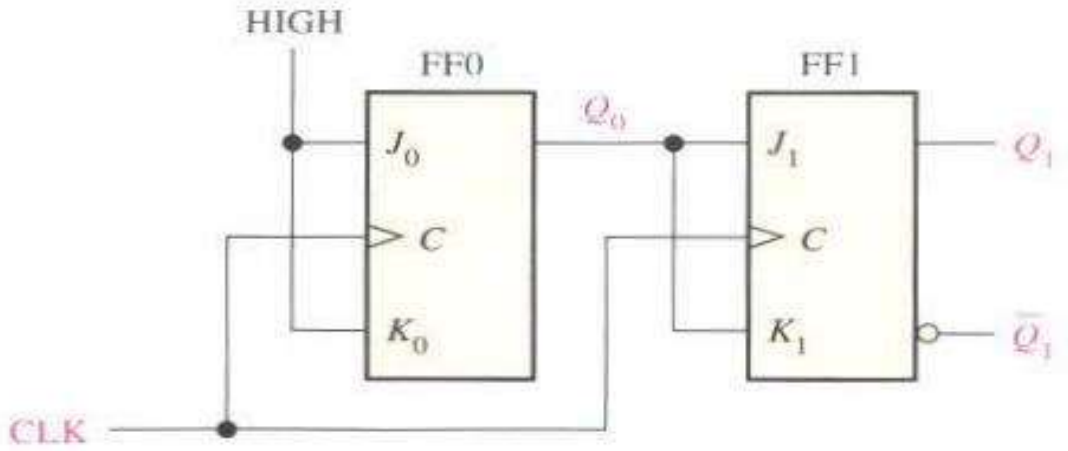
2-bit Ripple Down-Counter using Positive Edge-triggered Flip-Flop

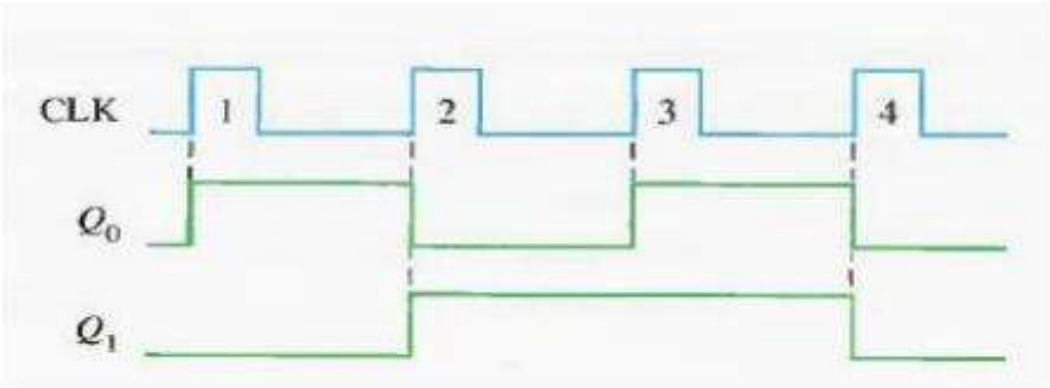


CLK	Present State	
	Q2	Q1
	0	0
	1	1
	1	0
	0	1
	0	0

Synchronous counters

2-bit Synchronous Binary Counter





CLK	Present State	
	Q_1	Q_0
	0	0
↓	0	1
↓	1	0
↓	1	1
↓	0	0

Classification of counters

Depending on the way in which the counting progresses, the synchronous or asynchronous counters are classified as follows −

Up counters
Down counters
Up/Down counters

UP/DOWN Counter

Up counter and down counter is combined together to obtain an UP/DOWN counter. A mode control (M) input is also provided to select either up or down mode. A combinational circuit is required to be designed and used between each pair of flip-flop in order to achieve the up/down operation.

Type of up/down counters
UP/DOWN ripple counters
UP/DOWN synchronous counter

UP/DOWN Ripple Counters

In the UP/DOWN ripple counter all the FFs operate in the toggle mode. So either T flip-flops or JK flip-flops are to be used. The LSB flip-flop receives clock directly. But the clock to every other FF is obtained from (Q = Q bar) output of the previous FF.

UP counting mode (M=0) − The Q output of the preceding FF is connected to the clock of the next stage if up counting is to be achieved. For this mode, the mode select input M is at logic 0 (M=0).

DOWN counting mode (M=1) − If M = 1, then the Q bar output of the preceding FF is connected to the next FF. This will operate the counter in the counting mode.

3-bit binary up/down ripple counter.

- 3-bit − hence three FFs are required.
- UP/DOWN − So a mode control input is essential.

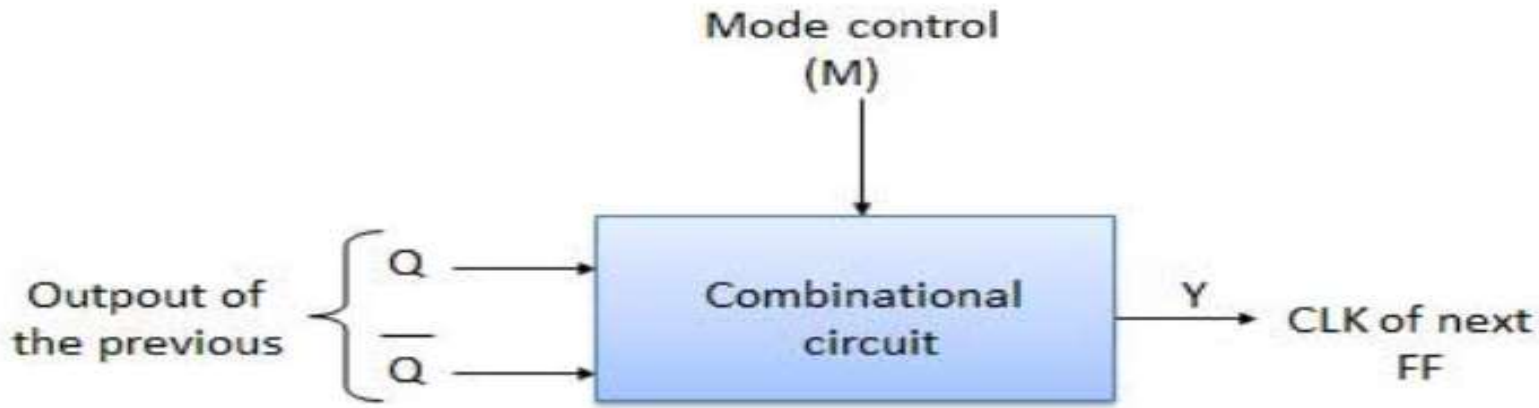
- For a ripple up counter, the Q output of preceding FF is connected to the clock input of the next one.

- For a ripple up counter, the Q output of preceding FF is connected to the clock input of the next one.

- For a ripple down counter, the Q bar output of preceding FF is connected to the clock input of the next one.

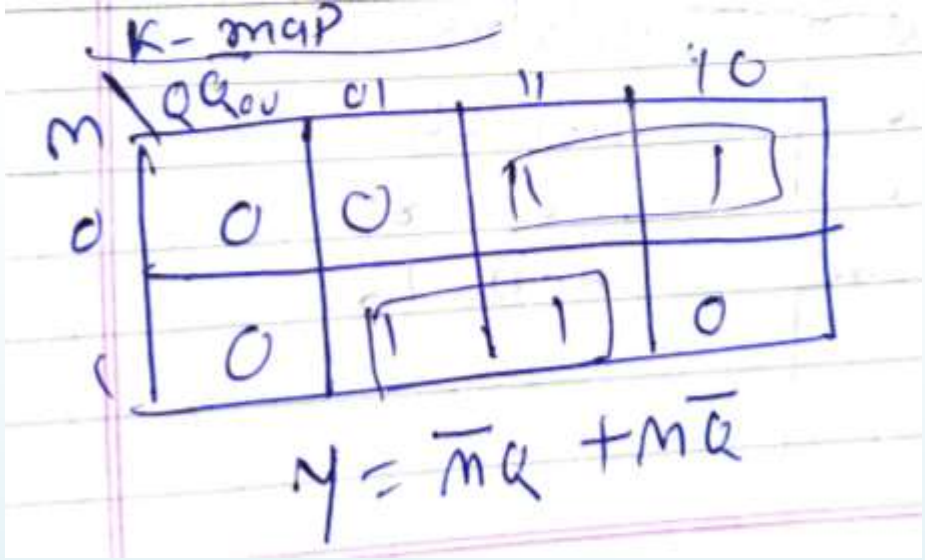
- Let the selection of Q and Q bar output of the preceding FF be controlled by the mode control input M such that, If M = 0, UP counting. So connect Q to CLK. If M = 1, DOWN counting. So connect Q bar to CLK.

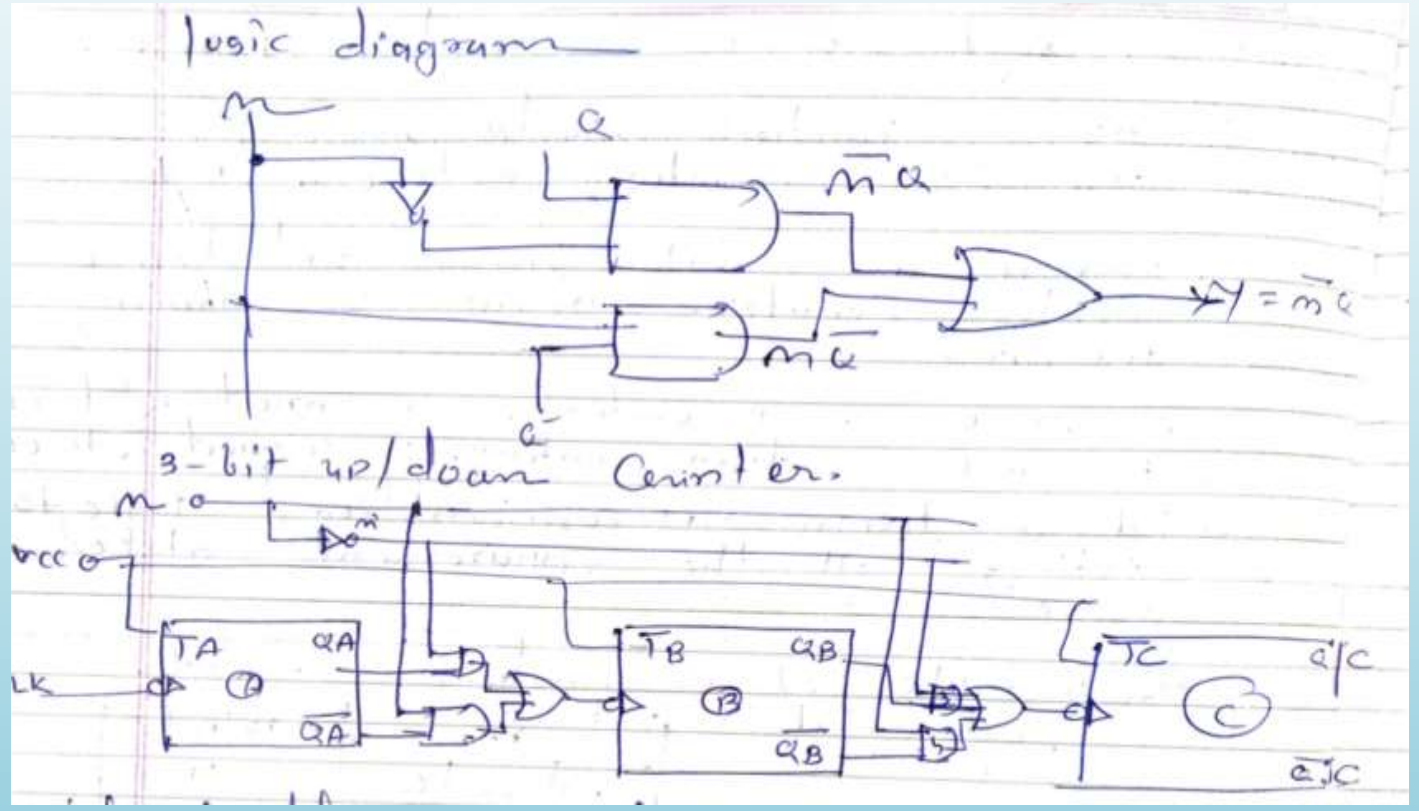
Block Diagram



Truth Table

Inputs			Outputs	
M	Q	$\overline{Q}$	Y	
0	0	0	0	Y = Q for up counter
0	0	1	0	
0	1	0	1	
0	1	1	1	
1	0	0	0	$Y = \overline{Q}$ for up counter
1	0	1	1	
1	1	0	0	
1	1	1	1	


K- map
QQ0 01 11 10
M
0
1
0 0 1 1
0 1 1 0
Y = M̄Q + MQ̄


logic diagram
M
Q
M̄Q
MQ̄
Y = M̄Q + MQ̄
Q̄
3-bit up/down Counter.
M
Vcc
TA
QA
CLK
QA
TB
QB
QB
JC
Q/C
Q̄/C
C

Modulus Counter (MOD-N Counter)

The 2-bit ripple counter is called as MOD-4 counter and 3-bit ripple counter is called as MOD-8 counter. So in general, an n-bit ripple counter is called as modulo-N counter. Where, MOD number = 2^n.

Type of modulus

2-bit up or down (MOD-4)

3-bit up or down (MOD-8)

4-bit up or down (MOD-16)

→ Design a modulo-5 ripple counter using 3-bit ripple counter

Step-1 Draw state diagram

Step-2 write truth table for next logic:

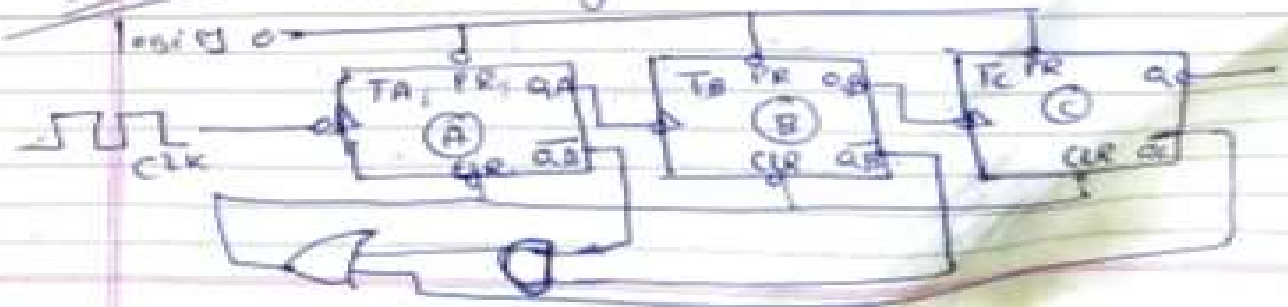
state	flip-flop o/ps			o/p of next logic	
	Q_C	Q_B	Q_A		
0	0	0	0	1	valid state
1	0	0	1	1	
2	0	1	0	1	
3	0	1	1	1	
4	1	0	0	0	invalid state
5	1	0	1	0	
6	1	1	0	0	
7	1	1	1	0	

Step-3 Karnaugh map

Q_C \ $Q_B Q_A$	00	01	11	10
0	1	1	1	1
1	1	0	0	0

$Y = \overline{Q_C} + \overline{Q_B}\,\overline{Q_A}$

Step-4 logic diagram



Design 3-bit Binary Counter

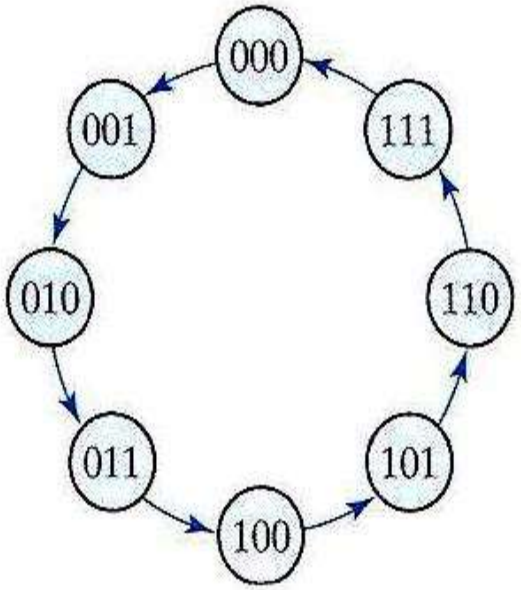


FIGURE 5.32
State diagram of three-bit binary counter

Output State Transitions		Flip-flop inputs
Present State $Q_2\ Q_1\ Q_0$	Next State $Q_2\ Q_1\ Q_0$	$T_2\ T_1\ T_0$
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 1
0 1 0	0 1 1	0 0 1
0 1 1	1 0 0	1 1 1
1 0 0	1 0 1	0 0 1
1 0 1	1 1 0	0 1 1
1 1 0	1 1 1	0 0 1
1 1 1	0 0 0	1 1 1

Excitation table

Next step is to transfer the flip-flop input functions to Karnaugh maps to derive a simplified Boolean expressions,

Q2Q1 \ Q0	0	1
00	0	0
01	0	1
11	0	1
10	0	0

T2 map

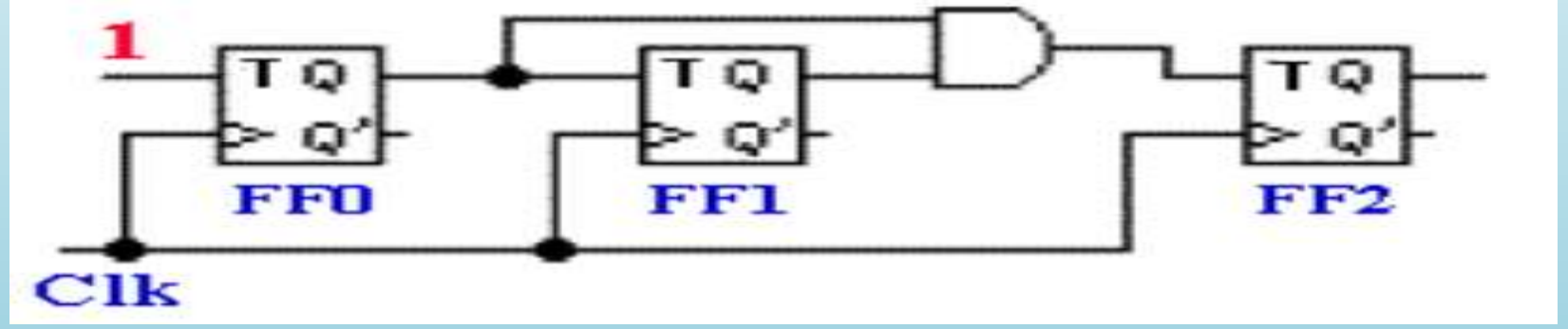
Q2Q1 \ Q0	0	1
00	0	1
01	0	1
11	0	1
10	0	1

T1 map

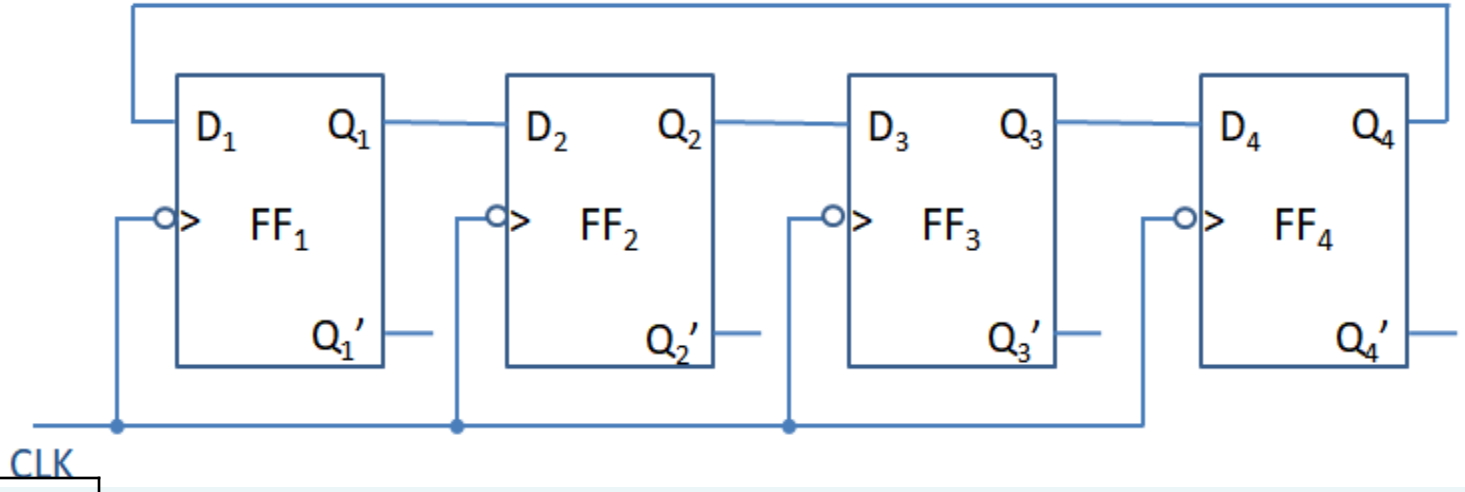
Q2Q1 \ Q0	0	1
00	1	1
01	1	1
11	1	1
10	1	1

T0 map

$T_0 = 1;$ $T_1 = Q_0;$ $T_2 = Q_1 * Q_0$

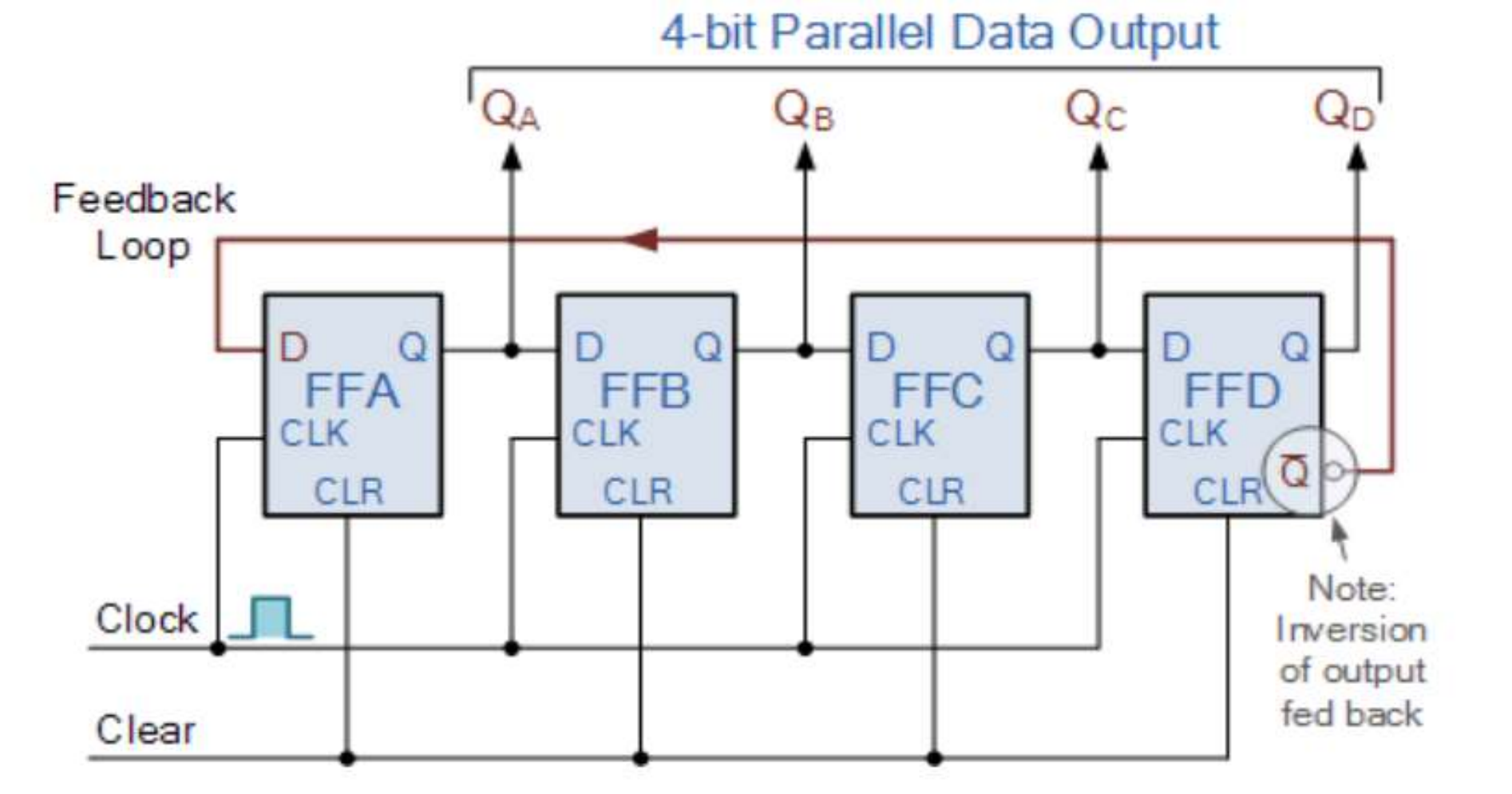


Ring Counter



After pulses	State			
	Q_1	Q_2	Q_3	Q_4
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0
5	0	1	0	0
6	0	0	1	0
7	0	0	0	1

4-bit Johnson Ring Counter



Clock Pulse No	FFA	FFB	FFC	FFD
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

Application of counters

Frequency counters
Digital clock
Time measurement
A to D converter
Frequency divider circuits
Digital triangular wave generator.