APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

LOGIC SYSTEM DESIGN (CST-203)

MODULE 4
NOTES

PREPARED BY : JIBIN EP, ASSISTANT PROFESSOR, EKC TC MANJERI

INDEX

SL No	Topic	Slide Link	Video Link
1	INTRODUCTION TO SEQUENTIAL CIRCUITS	INTRODUCTION TO SEQUENTIAL CIRCUITS	https://youtu.be/9NitpW-Rmzs
2	SR LATCH LATCH VS FLIPFLOP	SR LATCH LATCH VS FLIPFLOP	https://youtu.be/y8NewzE4DqA
3	SR FLIPFLOP	SR FLIPFLOP	https://youtu.be/23C2FGN_wts
4	D FLIPFLOP	D FLIPFLOP	https://youtu.be/tyV1RLUSsAM
5	JK FLIPFLOP	JK FLIPFLOP	https://youtu.be/zuH42Y7dH3M
6	T FLIPFLOP	T FLIPFLOP	https://youtu.be/zuH42Y7dH3M

INDEX

Module IV

Sequential logic circuits:

Flip-flops- SR, JK, T and D. Triggering of flip-flops- Master slave flip- flops, Edge- triggered flip- flops. Excitation table and characteristic equation. Registers- register with parallel load. Counter design: Asynchronous counters- Binary and BCD counters, timing sequences and state diagrams. Synchronous counters- Binary Up- down counter, BCD counter.

MODULE 4
(TOPIC-1)

INTRODUCTION TO SEQUENTIAL CIRCUITS

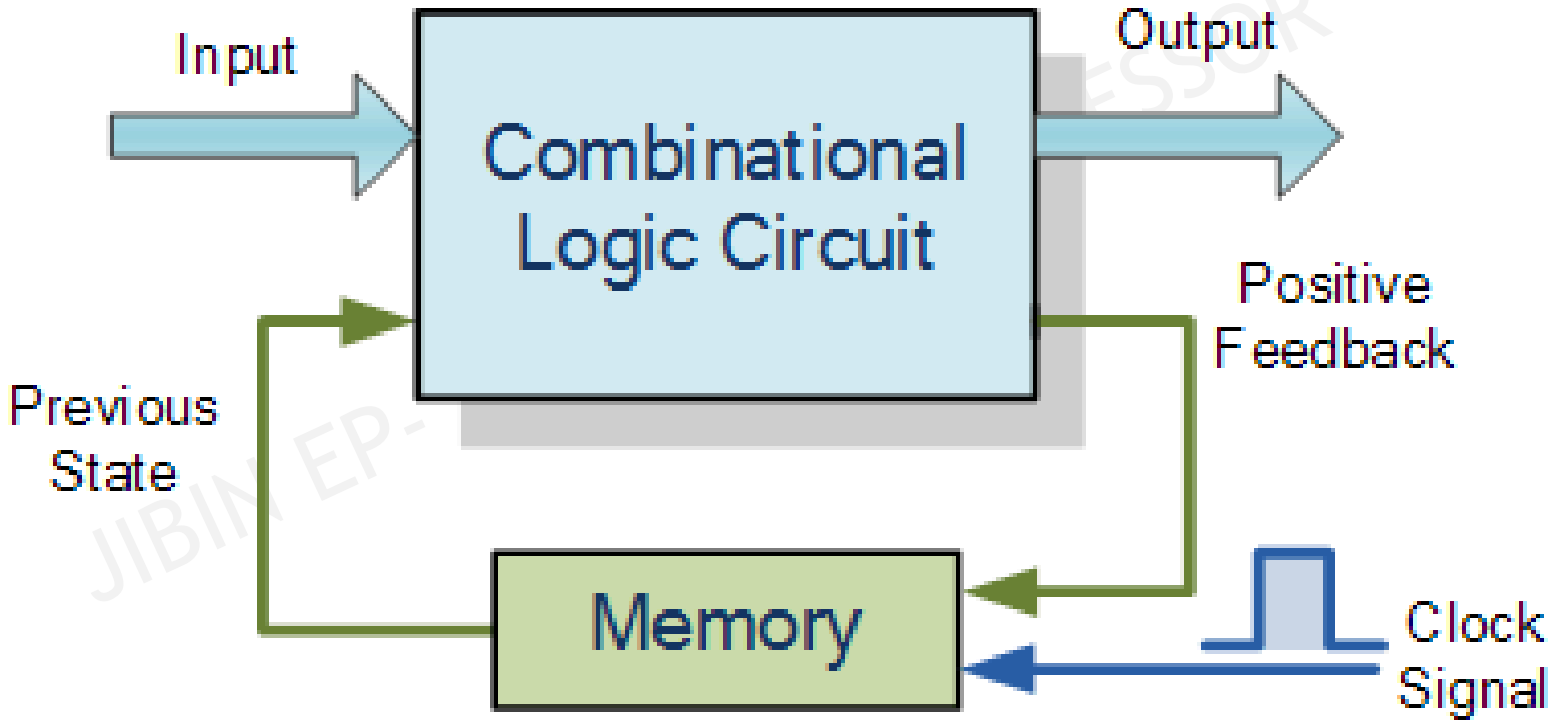
https://youtu.be/9NitpW-Rmz

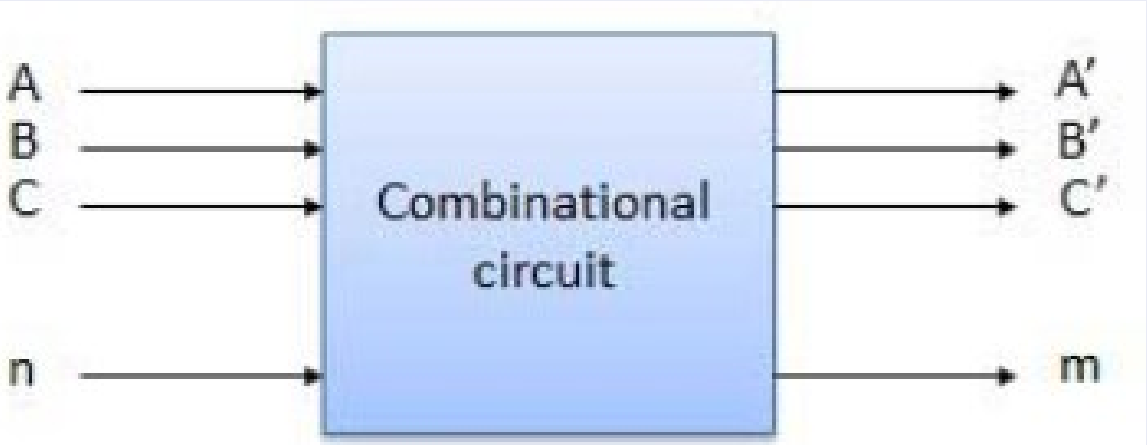
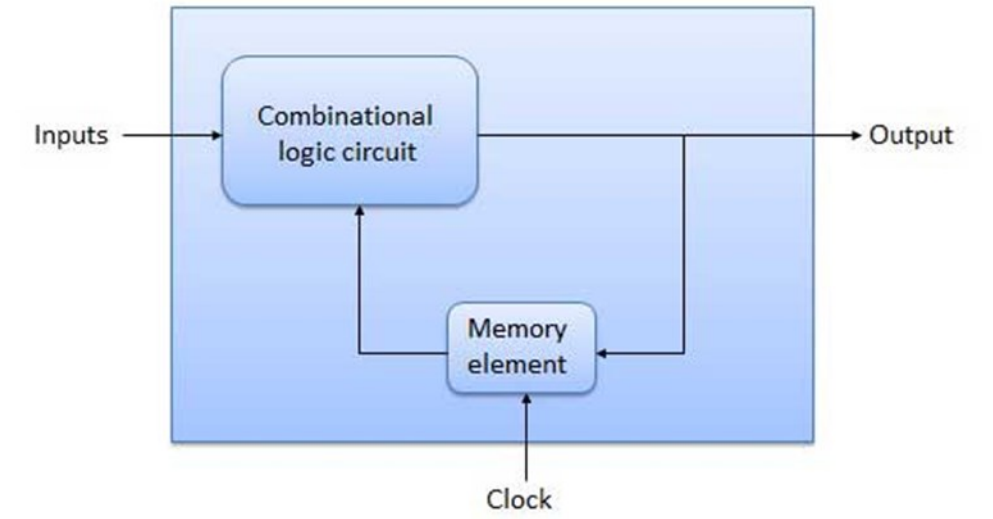
SEQUENTIAL CIRCUIT

- Sequential Circuit is a combinational circuit with Memory
- The output of Sequential Circuit depends upon the Present Input and Past Output [Present State]
- The Information stored in the circuit represent Preset State
- The Present State and Present Input Will define the Output and Next State

In Sequential Circuit the Present Output depends on Present Input as well as Past Output / Past Outputs

SEQUENTIAL LOGIC CIRCUIT

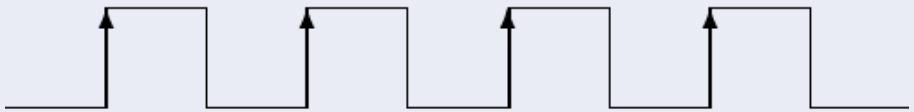
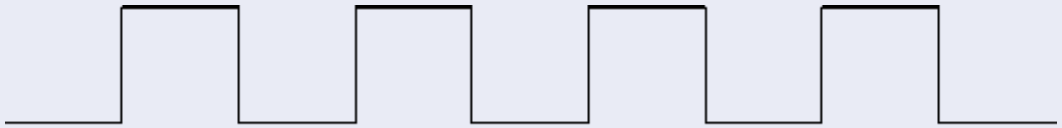


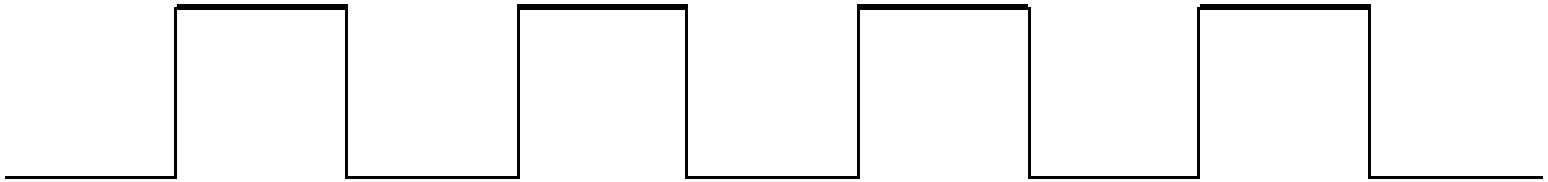
Combinational Circuit	Sequential Circuit
Output is Only depends on the Present Input	Output depends on Present input and Past Output
Memory Element is Absent	Presence of Memory Element
No Clock Signal is Applied	Clock signal is Required
	
Example : Half Adder , Full Adder, Multiplexer	Examples : Flpflop , Counters and Registers

TYPES OF SEQUENTIAL CIRCUIT

- Flipflop
- Counter
- Shift Register

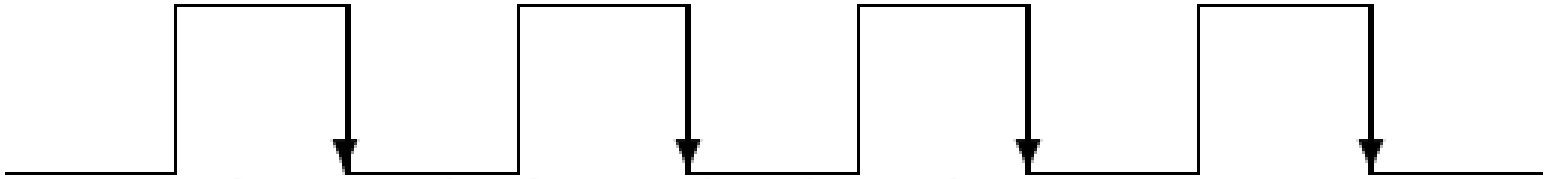
TRIGGERING METHODS

Edge Triggering	Level Triggering
Types of triggering that allows a circuit to become active at positive edge or negative edge of the clock signal	Types of triggering that allows a circuit to become active when the clock pulse is on a particular level
An Event occurs at the rising edge or falling edge	An Event occurs during the high voltage level or low voltage level
Flipflops are edge triggered	Latches are level triggered
	



(a) Level trigger.

(b) Positive-edge trigger.



(c) Negative-edge trigger.

Time ⟶

TYPES OF FLIPFLOPS

- SR FLIPFLOP
- D FLIPFLOP
- JK FLIPFLOP
- T FLIPFLOP
- MASTER SLAVE JK FLIPFLOP

MODULE 4
(TOPIC-2)

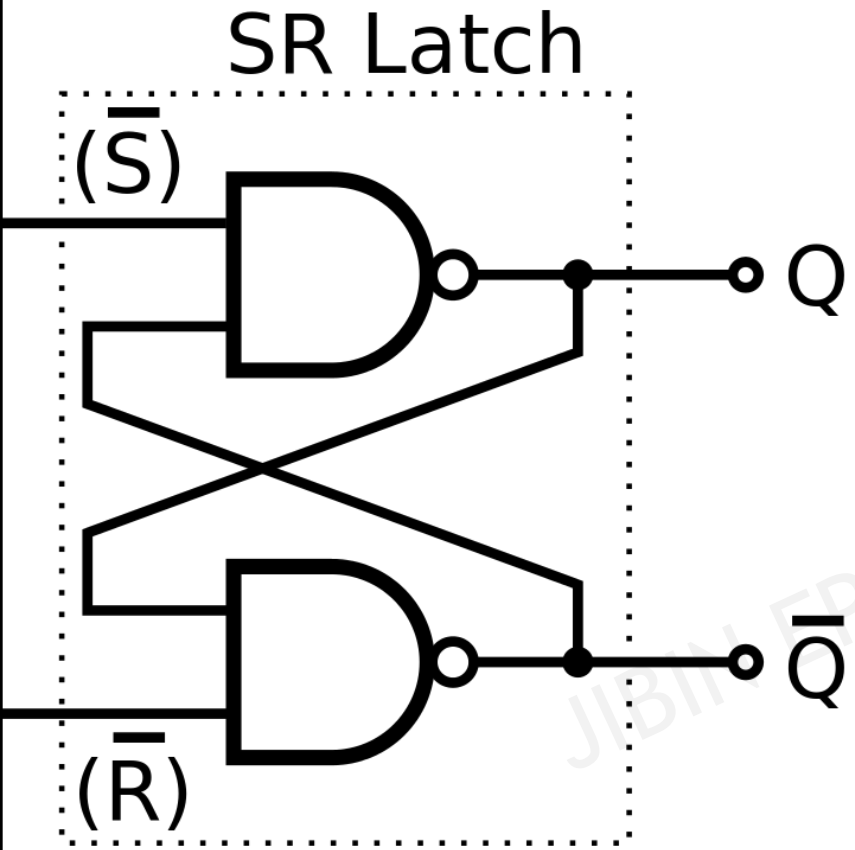
SR LATCH
LATCH VS FLIPFLOP

https://youtu.be/y8NewzE4DqA

Latch	Flipflop
Building block of sequential circuit, Built using Logic Gates	Building block of sequential circuit, built using Latches
Check input continuously and Changes output correspondingly	Check the input continuously and check the output in a continuous manner only with clock signal
Work with only binary inputs	Work with binary input and clock signal
Can not used as register	Can be used as a register
No Clock Signal	FF has clock signal

SR LATCH

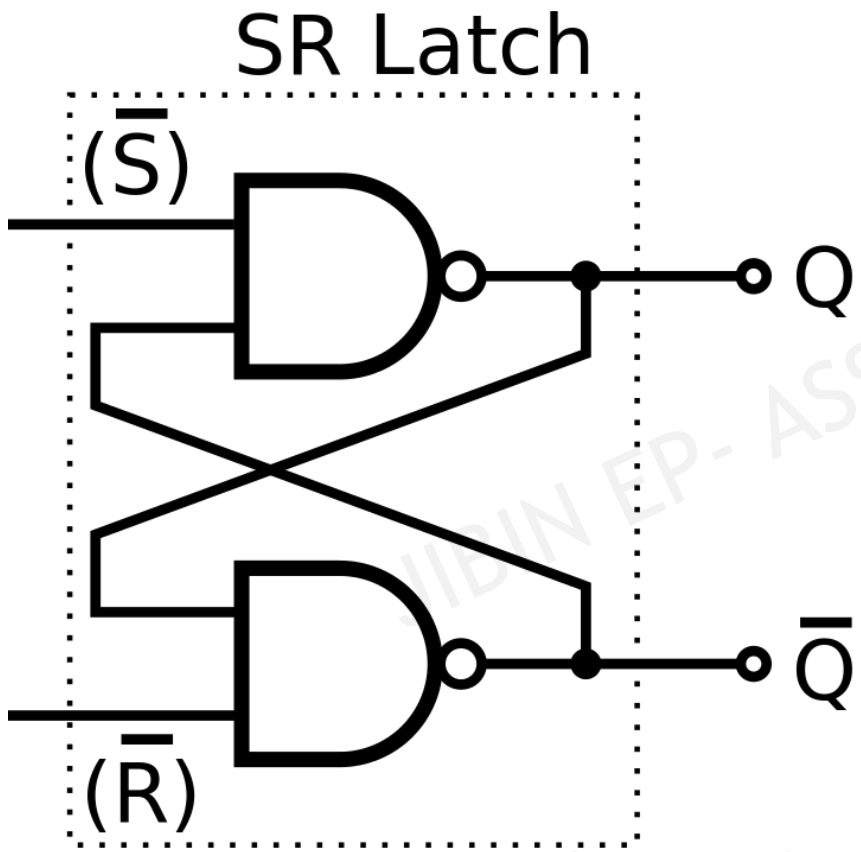
A	B	A **NAND** B
0	0	1
0	1	1
1	0	1
1	1	0



Active Low SR Latch can be constructed using two cross coupled NAND Gate

Suppose initially, $Q = 1$ and $\overline{Q} = 0$. If inputs are $S = 0$ and $R = 0$, G_1 inputs are a 0 (S) and a 0 ($\overline{Q}$). So, its output $Q = 1$. G_2 inputs are a 0 (R) and a 1 (Q) and so its output $\overline{Q} = 1$. Now, suppose $Q = 0$ and $\overline{Q} = 1$; then if inputs are $S = 0$ and $R = 0$, G_1 inputs are a 0 (S) and a 1 ($\overline{Q}$). So, its output $Q = 1$. G_2 inputs are a 0 (R) and a 1 (Q) and so its output $\overline{Q} = 1$. That is, whatever may be the initial state, when the inputs are $S = 0$ and $R = 0$, both Q and $\overline{Q}$ will be equal to 1.

SR LATCH



Case 1 : S=0 , R=0

Invalid State

$(Q = 1, Q^{-} = 1)$

Case 2 : S=0, R=1

$(Q = 1, Q^{-} = 0)$

Set State

Case 3: S=1 , R=0

$(Q = 0\ Q^{-} = 1)$

Reset State

Case 4: S=1 , R=1

Memory State

A	B	A **NAND** B
0	0	1
0	1	1
1	0	1
1	1	0

MODULE 4
(TOPIC-3)

SR FLIPFLOP

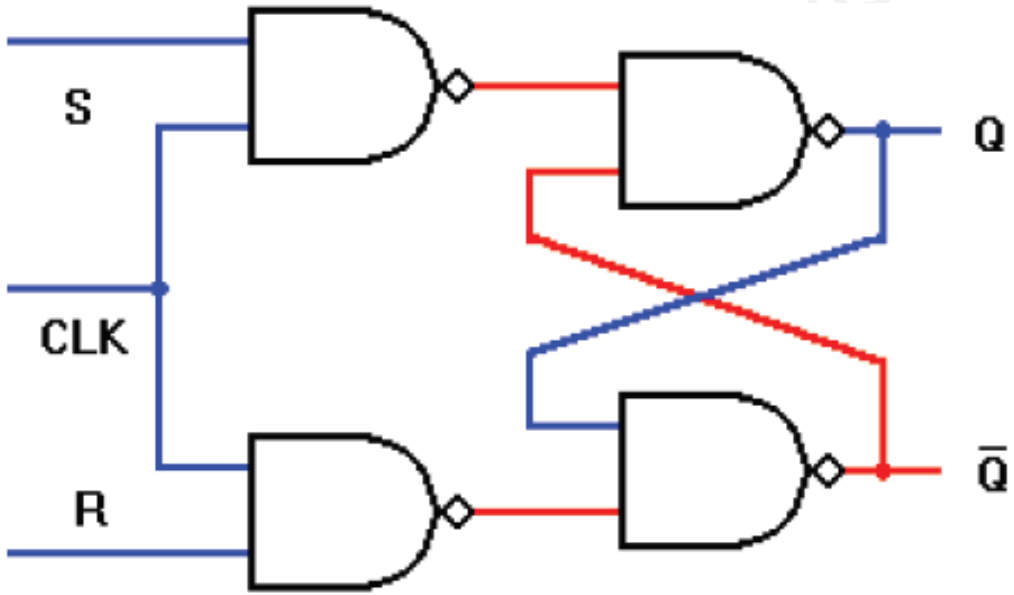
https://youtu.be/23C2FGN_wts

FLIPFLOP CIRCUITS

- Flip flops are actually an application of logic gates. With the help of Boolean logic you can create memory with them.
- Flip flops can also be considered as the most basic idea of a Random Access Memory [RAM]. When a certain input value is given to them, they will be remembered and executed, if the logic gates are designed correctly.
- A higher application of flip flops is helpful in designing better electronic circuits.
- The most commonly used application of flip flops is in the implementation of a feedback circuit. As a memory relies on the feedback concept, flip flops can be used to design it.

SR FLIPFLOP

- In the SR Latch we have seen that output changes occurs immediately after the input changes occurs
- I.e., the latch is sensitive to its S and R inputs at all times
- However it can easily be modified to create a latch that is sensitive to these inputs only when a clock input is active Such a circuit is called SR Flipflop



S	R	QN+1	STATE
0	0	QN	MEMORY
0	1	0	RESET
1	0	1	SET
1	1	X	INVALID

SR FLIPFLOP

S	R	QN+1	STATE
0	0	QN	MEMORY
0	1	0	RESET
1	0	1	SET
1	1	X	INVALID

Case 1 : If $S = R = 0$ and the clock pulse is applied, the output do not change, i.e. $Q_{n+1} = Q_n$. This is indicated in the first row of the truth table.

Case 2 : If $S = 0$, $R = 1$ and the clock pulse is applied, $Q_{n+1} = 0$. This is indicated in the second row of the truth table.

Case 3 : If $S = 1$, $R = 0$ and the clock pulse is applied, $Q_{n+1} = 1$. This is indicated in the third row of the truth table.

Case 4 : If $S = R = 1$ and the clock pulse is applied, the state of the flip-flip is undefined and therefore is indicated as indeterminate in the fourth row of the truth table.

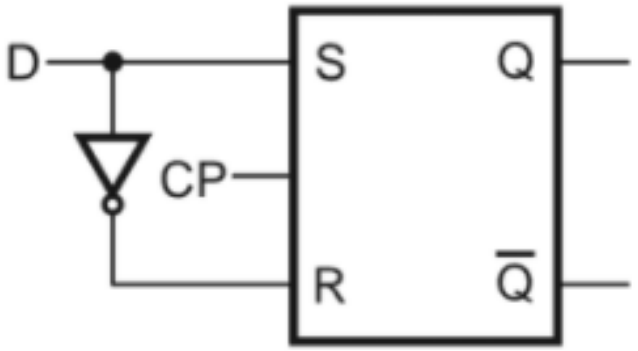
MODULE 4
(TOPIC-4)

D FLIPFLOP

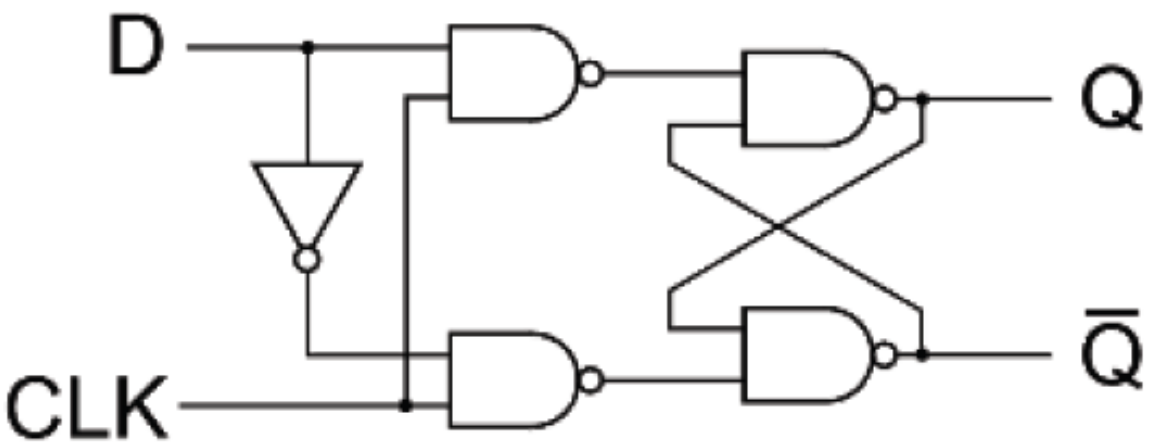
https://youtu.be/tyV1RLUSsAM

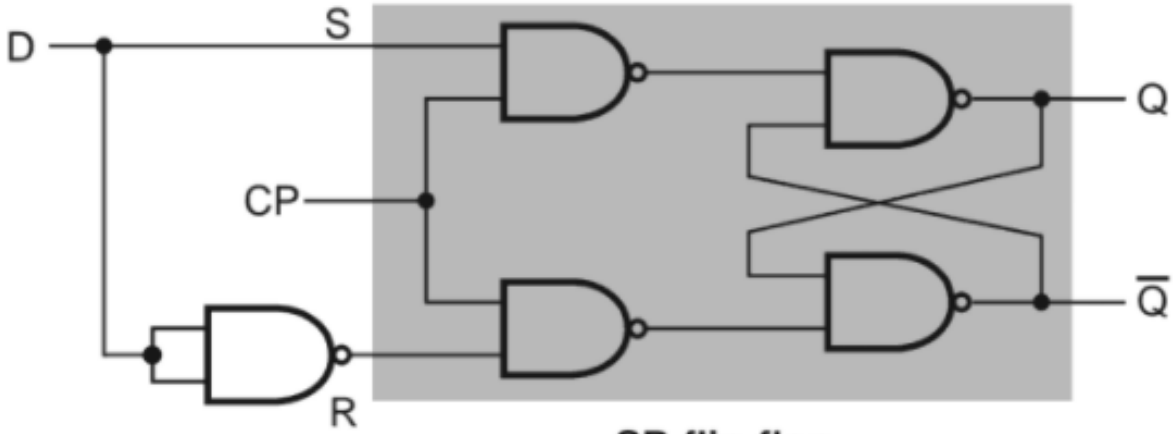
D FF

- The basic building block of D Flipflop is SR Flipflop
- The SR FF has two inputs S and R. The S Input is made high to store 1 in the flipflop and R input made high to store 0 in the flipflop
- Looking at the truth table of SR Flipflop we can realize that when both inputs are same the output either does not change or its invalid
- In many practical applications, these inputs are not required
- These inputs conditions can be avoided by making them complement of each other. This modified SR Flipflop is known as **D Flipflop**



D FF





SR flip-flop

Case 1 : D=0

Here Output Q_{N+1} Will be 0, the condition is called **Reset State**

Case 2 : D=0

Here Output Q_{N+1} Will be 1, the condition is called **Set State**

D	QN+1	STATE
0	0	RESET
1	1	SET

MODULE 4
(TOPIC-5)

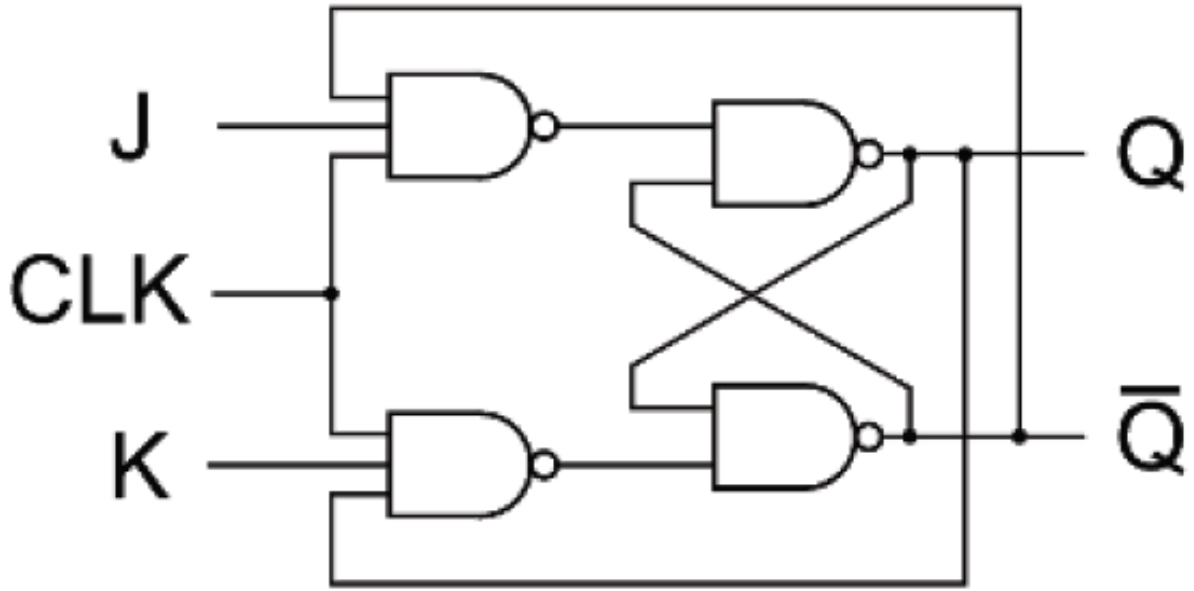
JK FLIPFLOP

https://youtu.be/zuH42Y7dH3M

JK FLIPFLOP

- The unnecessary in the state of SR Flipflop When S=1,R=1 can be eliminated by converting it into JK Flipflop.
- The data input are J and K which are ANDed with $Q^{-} and\ Q$ respectively, to obtain S and R inputs . Thus
-

$$S = J\,Q^{-}$$
$$R = K\,Q$$



JK FLIP FLOP

Case 1 : J = K = 0

When J = K = 0, S = R = 0 and according to truth table of SR flip-flop there is no change in the output.

When inputs J = K = 0, output does not change.

Case 2 : J = 1 and K = 0

Q = 0, $\overline{Q}$ = 1 : When J = 1, K = 0 and Q = 0, S = 1 and R = 0. According to truth table of SR flip-flop it is **set** state and the output Q will be 1.

Q = 1, $\overline{Q}$ = 0 : When J = 1, K = 0 and Q = 1, S = 0 and R = 0. Since SR = 00, there is no change in the output and therefore, Q = 1 and $\overline{Q}$ = 0.

The inputs J = 1 and K = 0, makes Q = 1, i.e. **set** state.

Case 3 : J = 0 and K = 1

Q = 0, $\overline{Q}$ = 1 : When J = 0, K = 1 and Q = 0, S = 0 and R = 0. Since SR = 00, there is no change in the output and therefore, Q = 0 and $\overline{Q}$ = 1.

Q = 1, $\overline{Q}$ = 0 : When J = 0, K = 1 and Q = 1, S = 0 and R = 1. According to truth table of SR flip-flop it is a **reset** state and the output Q will be 0.

The inputs J = 0 and K = 1, makes Q = 0, i.e., **reset** state.

Case 4 : J = K = 1

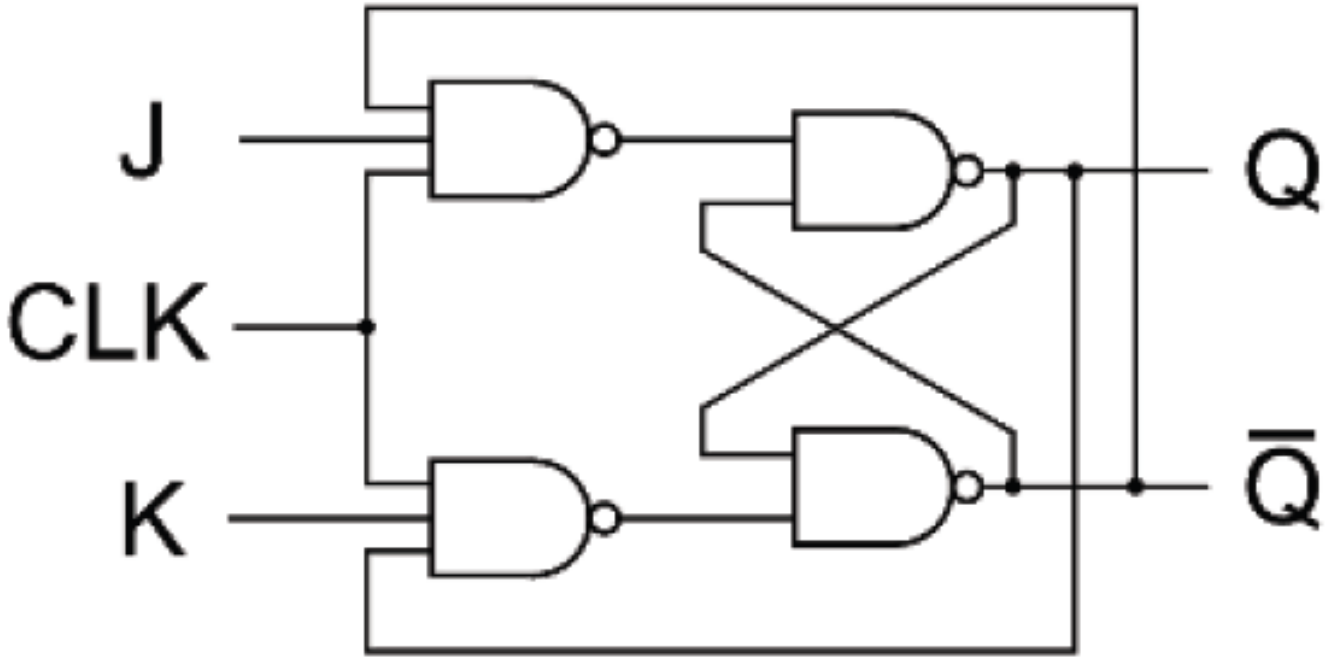
Q = 0, $\overline{Q}$ = 1 : When J = K = 1 and Q = 0, S = 1 and R = 0. According to truth table of SR flip-flop it is a set state and the output Q will be 1.

Q = 1, $\overline{Q}$ = 0 : When J = K = 1 and Q = 1, S = 0 and R = 1. According to truth table of SR flip-flop it is a reset state and the output Q will be 0.

The input J = K = 1, toggles the flip-flop output.

J	K	QN+1	STATE
0	0	QN	MEMORY
0	1	0	RESET
1	0	1	SET
1	1	QN BAR	TOGGLE

JK FLIPFLOP



J	K	QN+1	STATE
0	0	QN	MEMORY
0	1	0	RESET
1	0	1	SET
1	1	QN BAR	TOGGLE

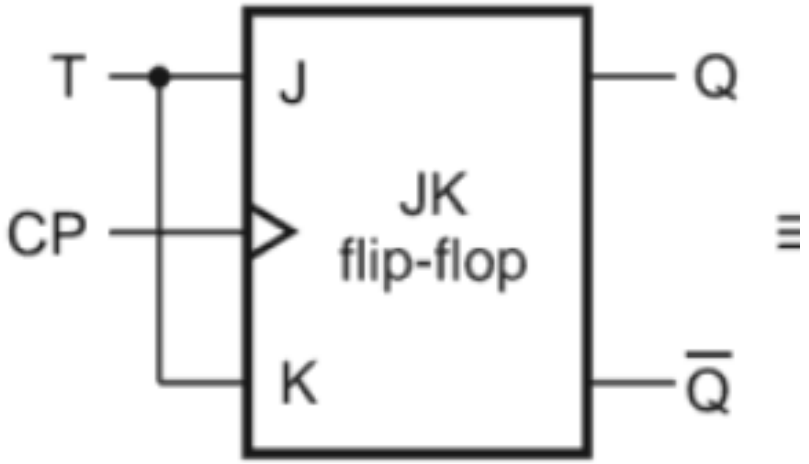
MODULE 4
(TOPIC-6)

T FLIPFLOP

https://youtu.be/zuH42Y7dH3M

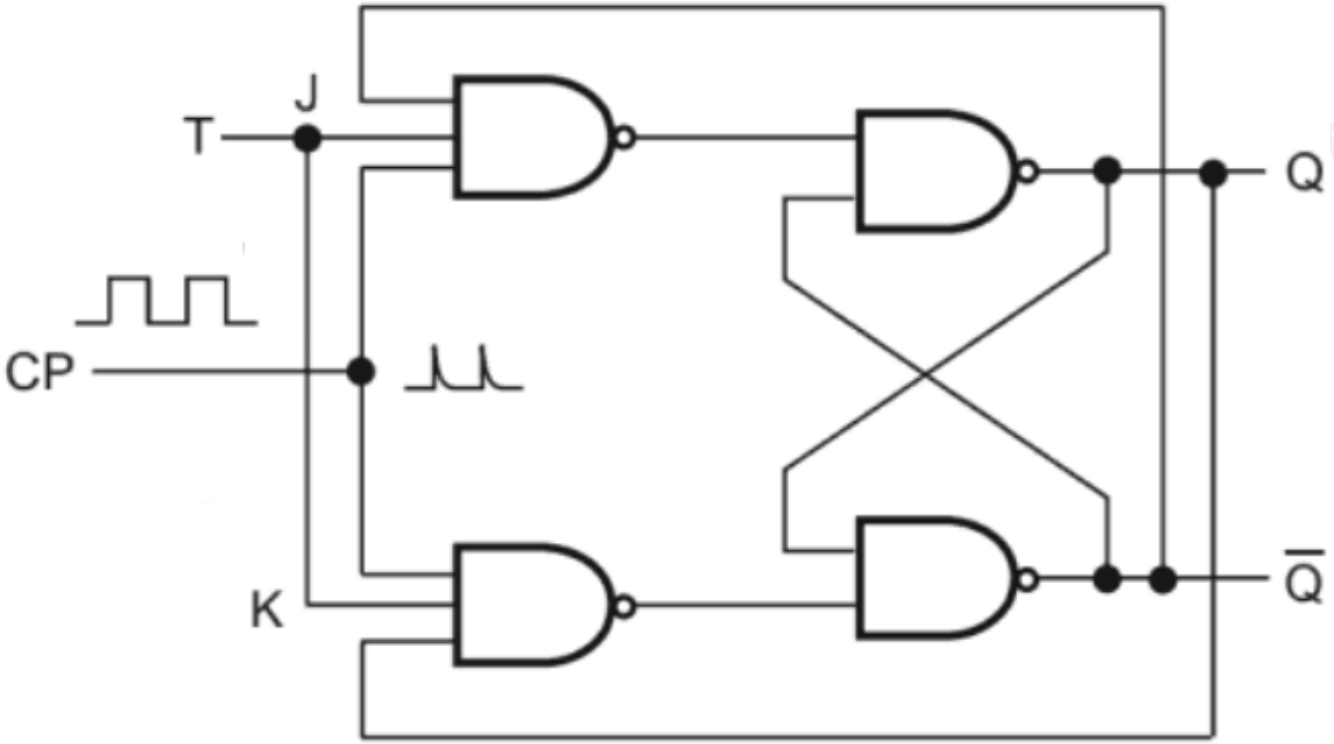
T FLIPFLOP

- T Flipflop is also known as **Toggle Flipflop**. The T Flipflop is the modification of JK Flipflop
- The T Flipflop is obtained from JK Flipflop by connecting both inputs J and K together
- When T=0, J=K=0 , hence there is no change in the output
- When T=1, J=K=1, and hence output Toggles



T FLIPFLOP

When T=0, J=K=0 , hence there is no change in the output
When T=1, J=K=1, and hence output Toggles



T	QN+1	STATE
0	QN	MEMORY
1	QN BAR	TOGGLE

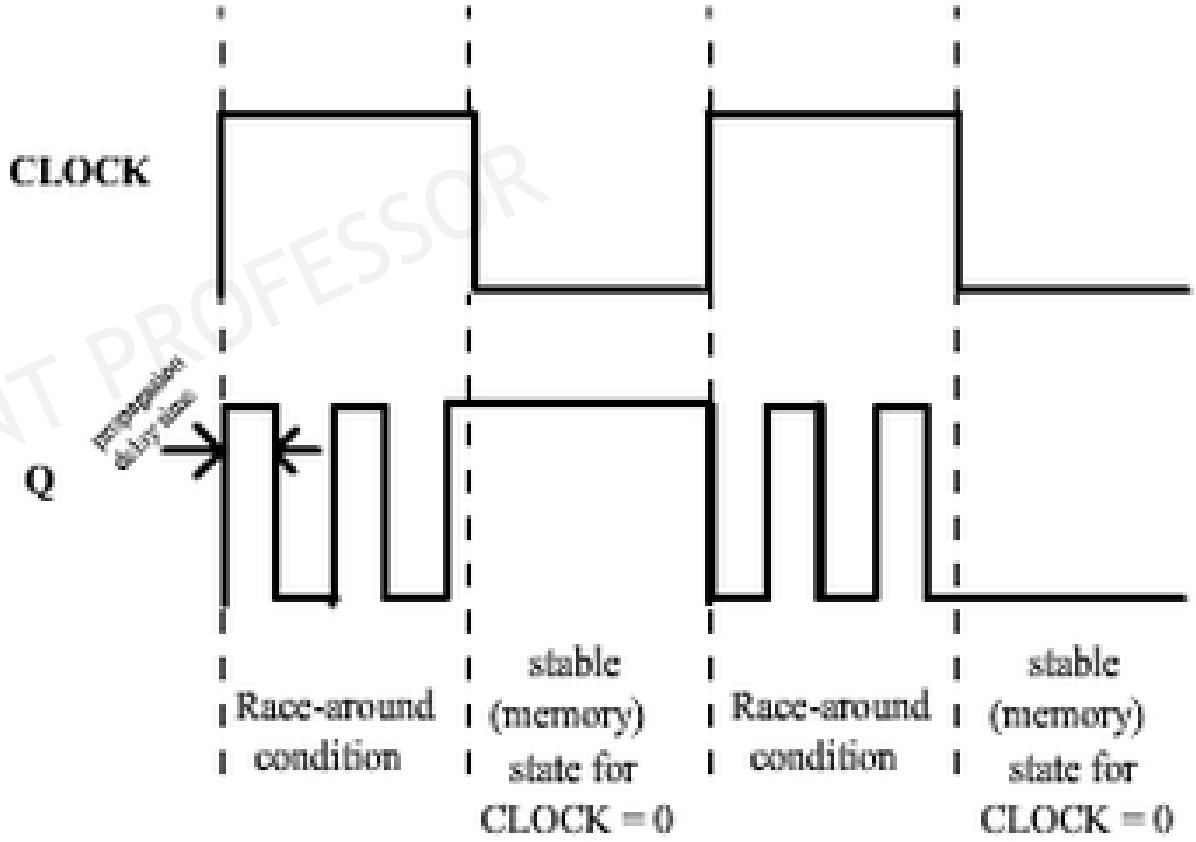
MODULE 4
(TOPIC-7)

RACE AROUND CONDITION
AND
MASTER SLAVE JK FLIPFLOP

https://youtu.be/cRjDzE11Pk4

What is Race Around Condition

- For JK Flipflop, When J=1 and K=1 , clock is too long then the state of FF keeps on Toggle which leads to uncertainty in determining the output state of flipflop
- This Problem is called **Race Around Condition**



Methods of Eliminating Race Around Condition

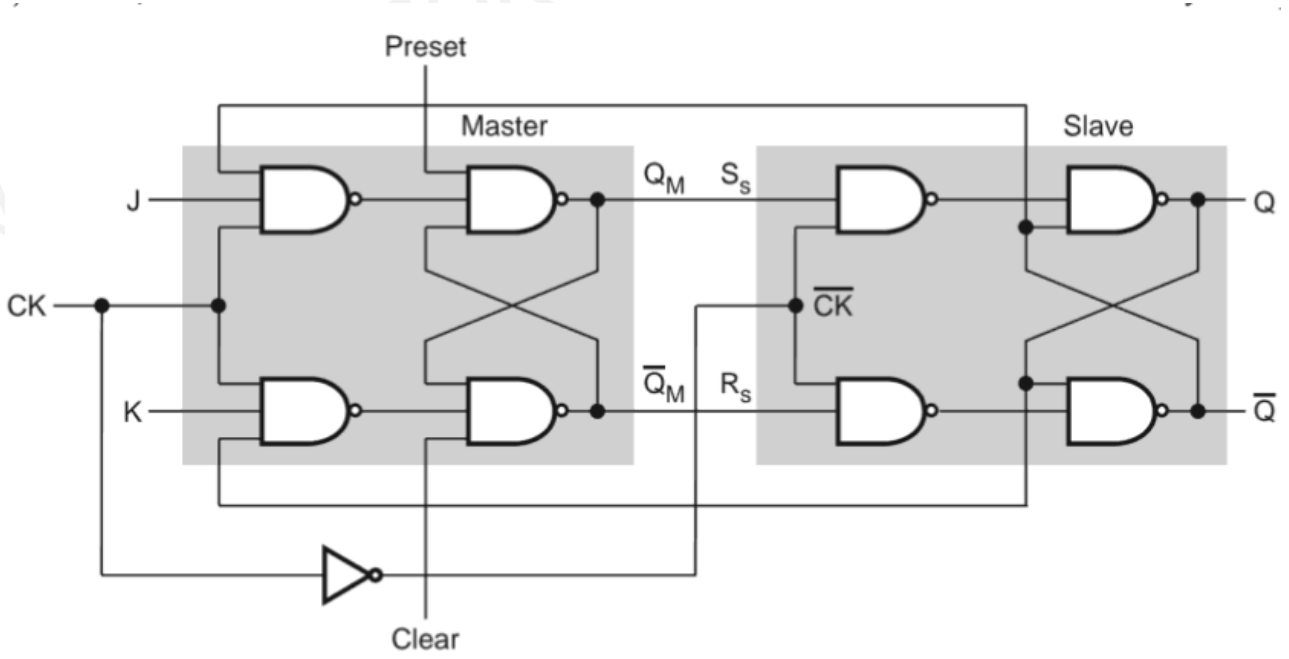
- T/2 < Propagation Delay
  (Half Time Period is less than Propagation Delay)
- Use of Negative Edge Triggering
- Master Slave JK Flipflop

MASTER SLAVE JK FLIPFLOP

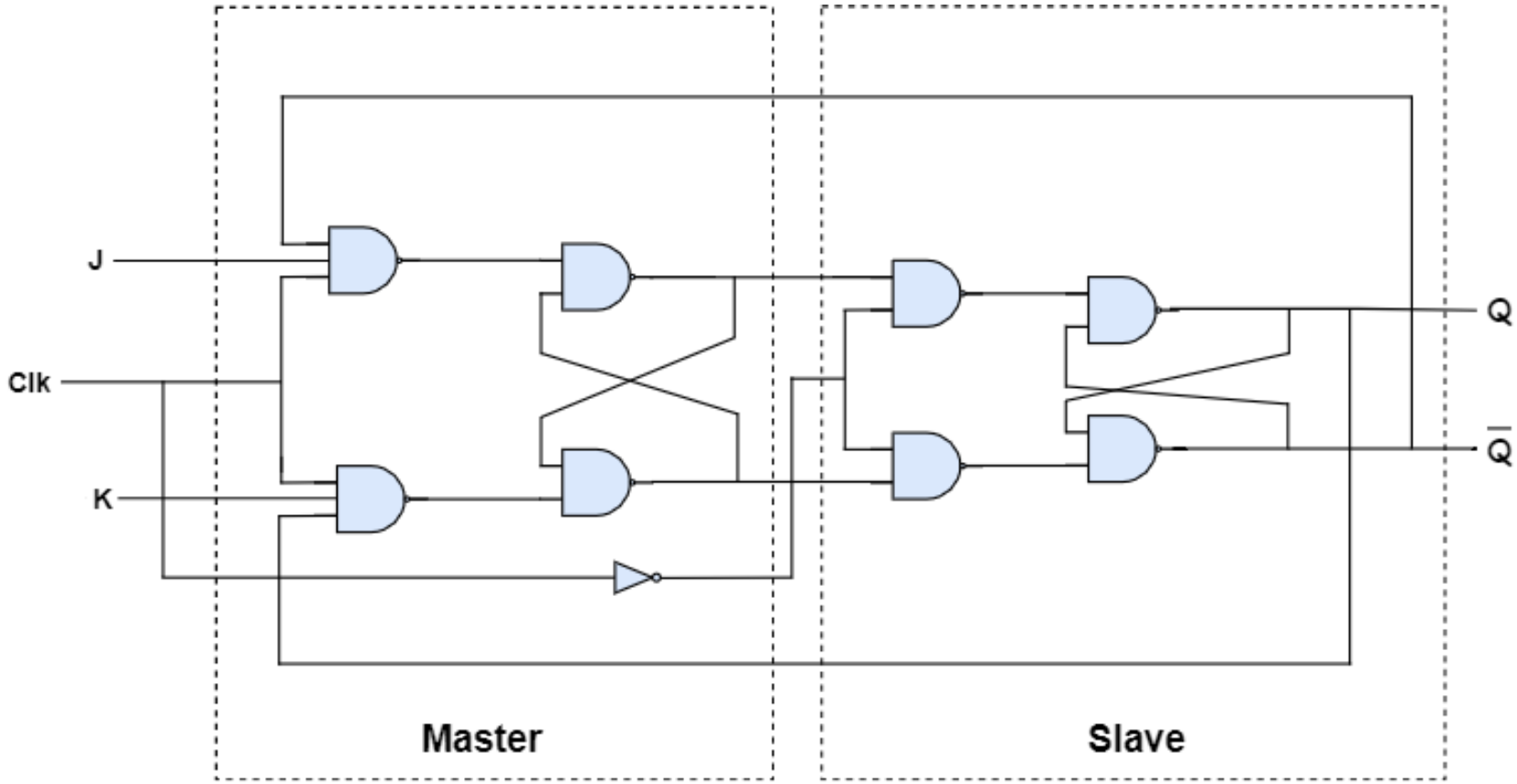
- It Consist of Clocked JK Flipflop as master and Clocked JK Flipflop as slave
- Output of the master FF is fed as an input of slave Flipflop
- Clock is connected directly to master flipflop, but it is connected through inverter to slave flipflop
- In the master slave JK Flipflop the State change occurs when flipflop goes through both positive transition of clock and negative transition of the clock. Thus **race around condition** does not exist in master slave JK Flipflop

MASTER SLAVE JK FLIPFLOP

- Figure shows master slave JK Flipflop. Positive clock pulses are applied to first flipflop and inverted clock pulses are applied to second flipflop
- When CLK=1 first FF is enabled, the second flipflop is inhibited because of its clock is low
- When CLK=0, first FF is Inhibited and Second FF is enabled
- Since the Second flipflop follows first one , it is referred as **slave** and the first one is called **Master**



MASTER SLAVE JK FLIPFLOP



MODULE 3
(TOPIC-8)

TRUTH TABLE , CHARACTERISTIC TABLE, EXCITATION TABLE CHARACTERISTIC EQUATION OF FLIPFLOP

https://youtu.be/nDnfLDjOxv0

SR FLIPFLOP

1

TRUTH TABLE

S	R	QN+1
0	0	QN
0	1	0
1	0	1
1	1	INVALID

2

CHARACTERISTIC TABLE

QN	S	Q	QN+1
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

3

EXCITATION

On	Qn+1	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

D FLIPFLOP

1

TRUTH TABLE	
D	QN+1
0	0
1	1

2

CHARACTERITIC TABLE		
Qn	D	Qn+1
0	0	0
0	1	1
1	0	0
1	1	1

3

EXCITATION TABLE		
Qn	QN+1	T
0	0	0
0	1	1
1	0	1
1	1	0

JK FLIPFLOP

1

TRUTH TABLE

J	K	QN+1
0	0	QN
0	1	0
1	0	1
1	1	QN BAR

3

EXCITATION

On	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

2

CHARACTERISTIC TABLE

QN	J	K	QN+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

T FLIPFLOP

1

TRUTH TABLE	
T	QN+1
0	Qn
1	Qn bar

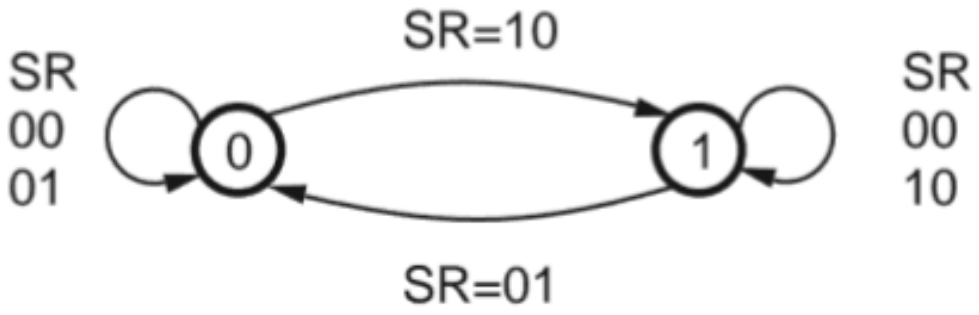
2

CHARACTERITIC TABLE		
Qn	T	Qn+1
0	0	0
0	1	1
1	0	1
1	1	0

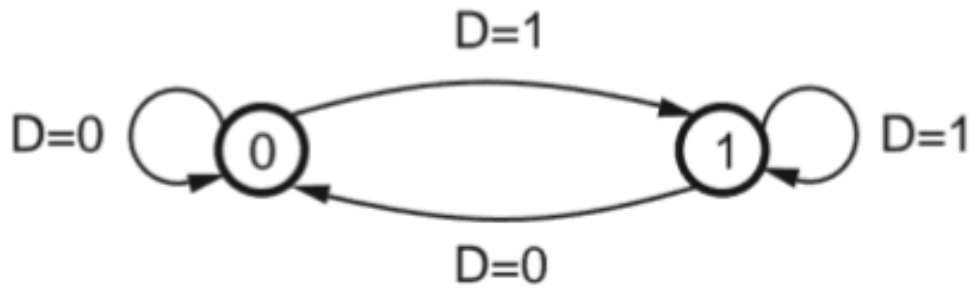
3

EXCITATION TABLE		
Qn	QN+1	T
0	0	0
0	1	1
1	0	1
1	1	0

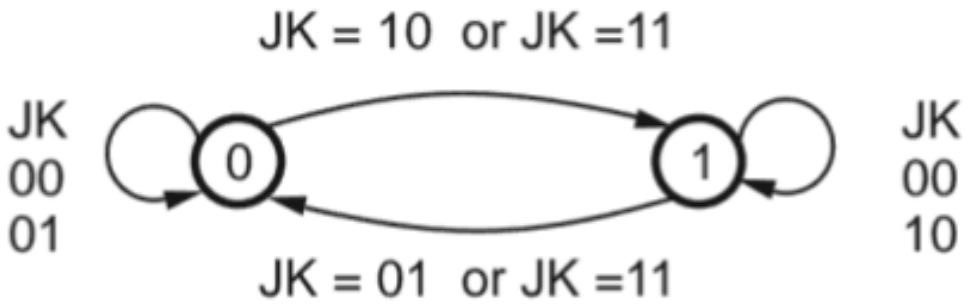
STATE DIAGRAMS



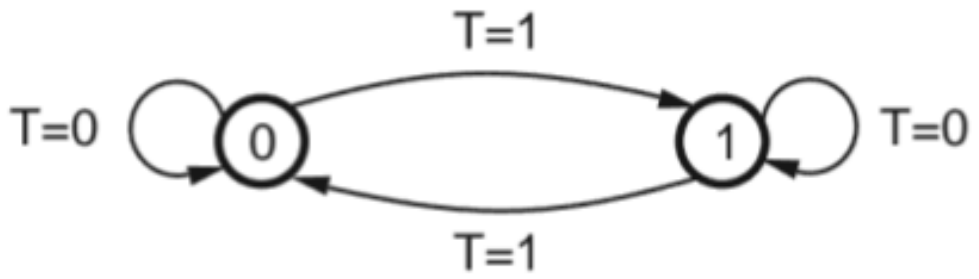
(a) SR flip-flop



(b) D flip-flop



(c) JK flip-flop



(d) T flip-flop

MODULE 4
(TOPIC-9)

INTRODUCTION TO COUNTERS

https://youtu.be/LP4844Om570

COUNTERS

- Counter is a sequential circuit. A digital circuit which is used for a counting pulses is known counter. Counter is the widest application of flip-flops. It is a group of flip-flops with a clock signal applied. Counters are of two types.

ASYNCHRONOUS COUNTER / RIPPLE COUNTER

SYNCHRONOUS COUNTER

Synchronous vs Asynchronous Counter

Synchronous Counter	Asynchronous Counter
In synchronous counter, all flip flops are triggered with same clock simultaneously.	In asynchronous counter, different flip flops are triggered with different clock, not simultaneously.
Synchronous Counter is faster than asynchronous counter in operation.	Asynchronous Counter is slower than synchronous counter in operation.
Synchronous Counter does not produce any decoding errors.	Asynchronous Counter produces decoding error.
Synchronous Counter is also called Parallel Counter.	Asynchronous Counter is also called Serial Counter.
Synchronous Counter designing as well implementation are complex due to increasing the number of states.	Asynchronous Counter designing as well as implementation is very easy.
Synchronous Counter will operate in any desired count sequence.	Asynchronous Counter will operate only in fixed count sequence (UP/DOWN).
In synchronous counter, propagation delay is less.	In asynchronous counter, there is high propagation delay.

MODULE 4
(TOPIC-10)

ASYNCHRONOUS UP AND DOWN COUNTER

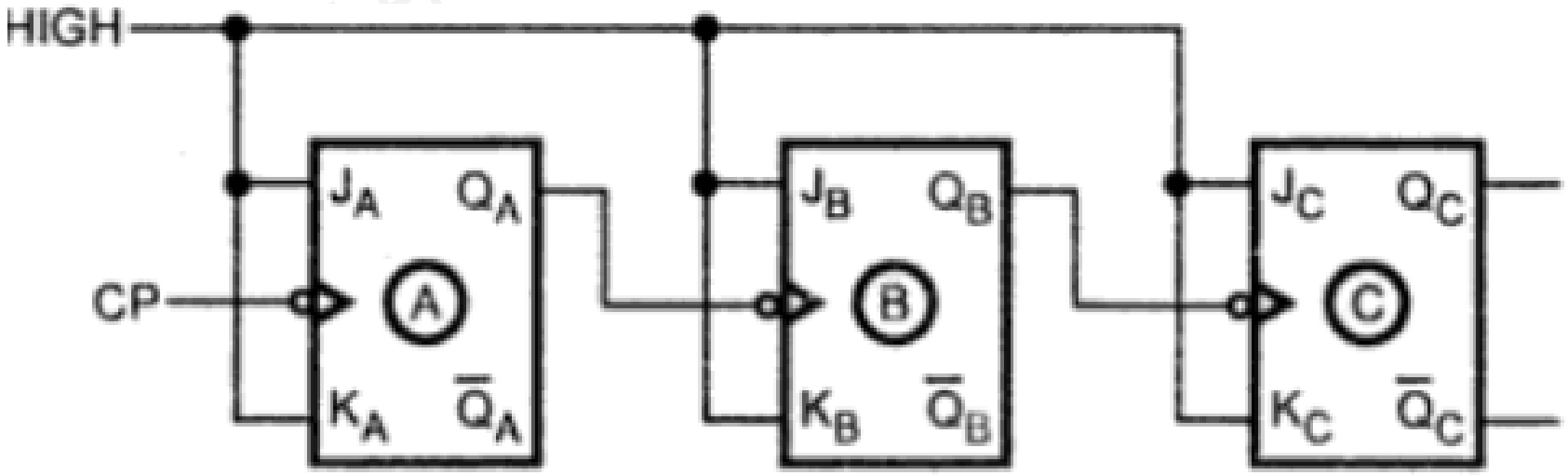
https://youtu.be/LP4844Om570

Asynchronous counter

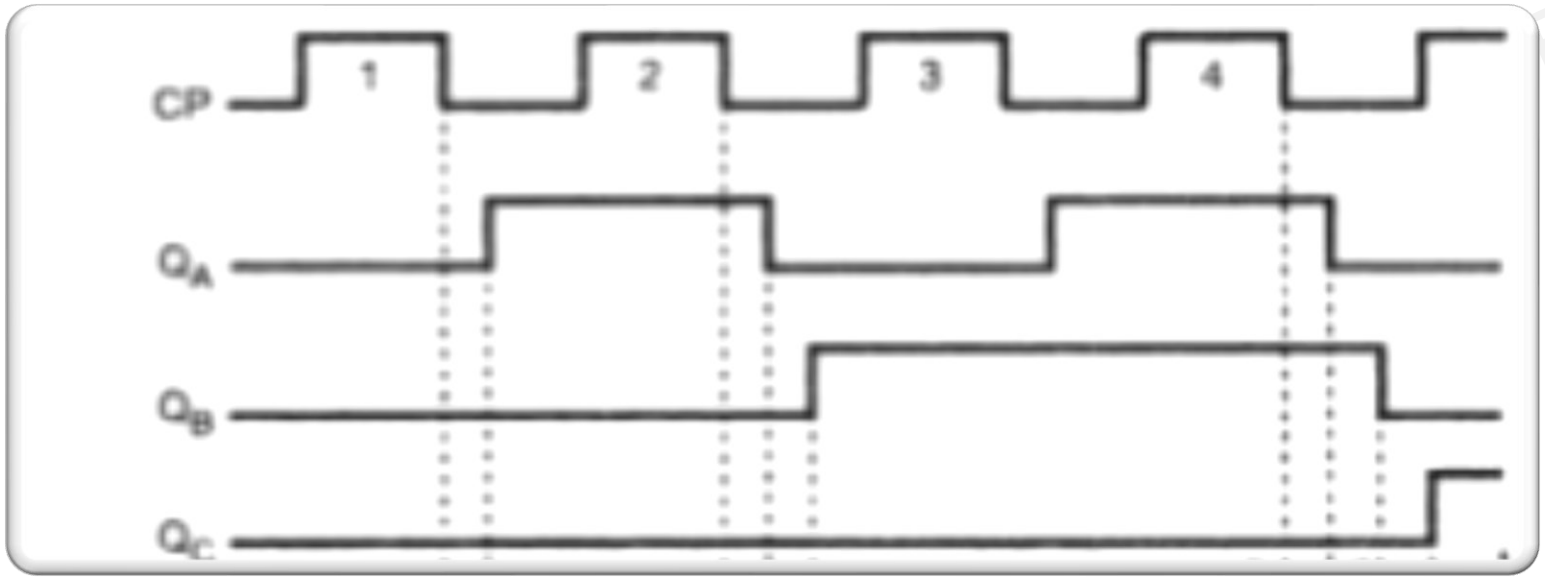
- If the flip-flops do not receive the same clock signal, then that counter is called as **Asynchronous counter**. The output of system clock is applied as clock signal only to first flip-flop. The remaining flip-flops receive the clock signal from output of its previous stage flip-flop. Hence, the outputs of all flip-flops do not change

3 Bit Asynchronous Up Counter

- It Consist of 3 JK Flipflops.
- In order to achieve Toggle condition, all the inputs are connected to logical high input (J=1, K=1)
- The initial clock is applied to first flipflop directly. The clock is negative edge triggered in order to eliminate race around condition
- The output of 1st flipflop is connected to Clock of 2nd flipflop
- The output of 2nd Flipflop is act as clock of 3rd flipflop and so on.



Timing Diagram and Truth table

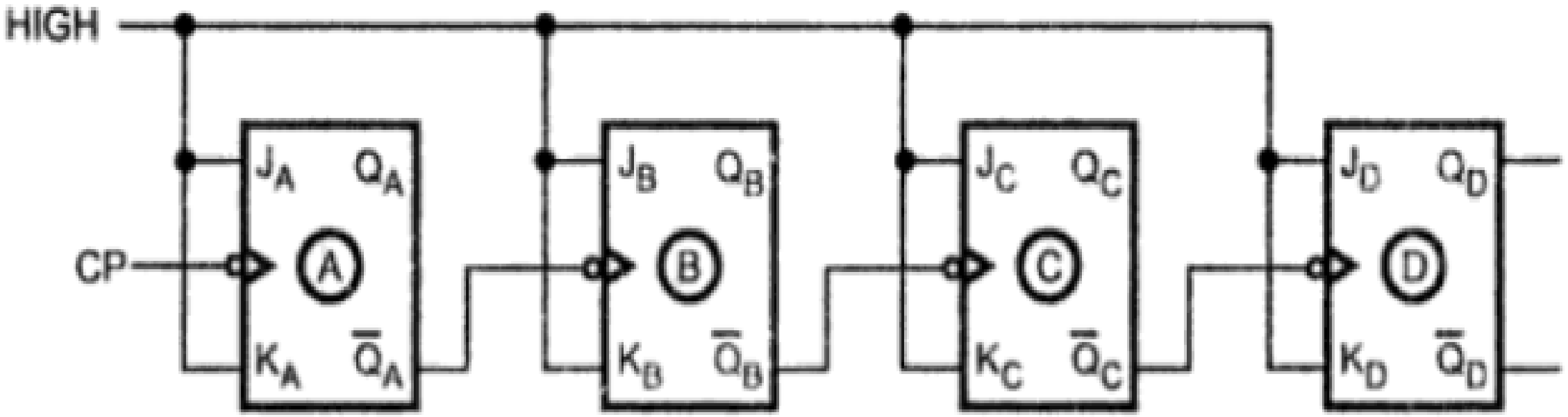


CLOCK	QC	QB	QA
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Asynchronous Down Counter

- The Down Counter will count downwards from a maximum count to zero
- In 4 bit asynchronous down counter, the clock signal is connected to clock of first flipflop
- The clock of remaining flipflop is triggered by QA bar output of previous stage

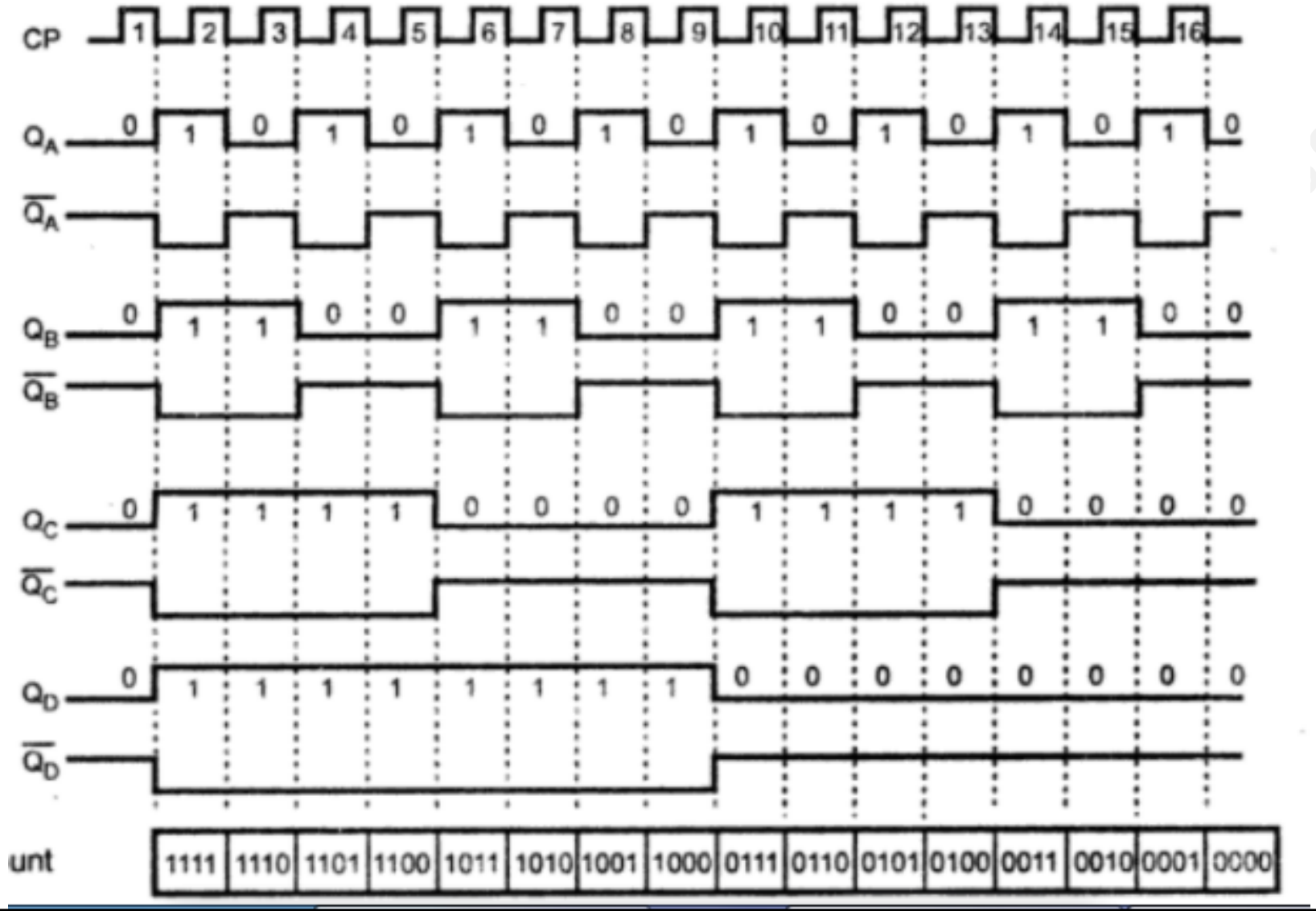
4 Bit Asynchronous Down Counter



Asynchronous Down Counter

- The Down Counter will count downwards from a maximum count to zero
- In 4 bit asynchronous down counter, the clock signal is connected to clock of first flipflop
- The clock of remaining flipflop is triggered by QA bar output of previous stage

Timing Diagram and Truth table



CLOCK	QD	QC	QB	QA
0	1	1	1	1
1	1	1	1	0
2	1	1	0	1
3	1	1	0	0
4	1	0	1	1
5	1	0	1	0
6	1	0	0	1
7	1	0	0	0
8	0	1	1	1
9	0	1	1	0
10	0	1	0	1
11	0	1	0	0
12	0	0	1	1
13	0	0	1	0
14	0	0	0	1
15	0	0	0	0

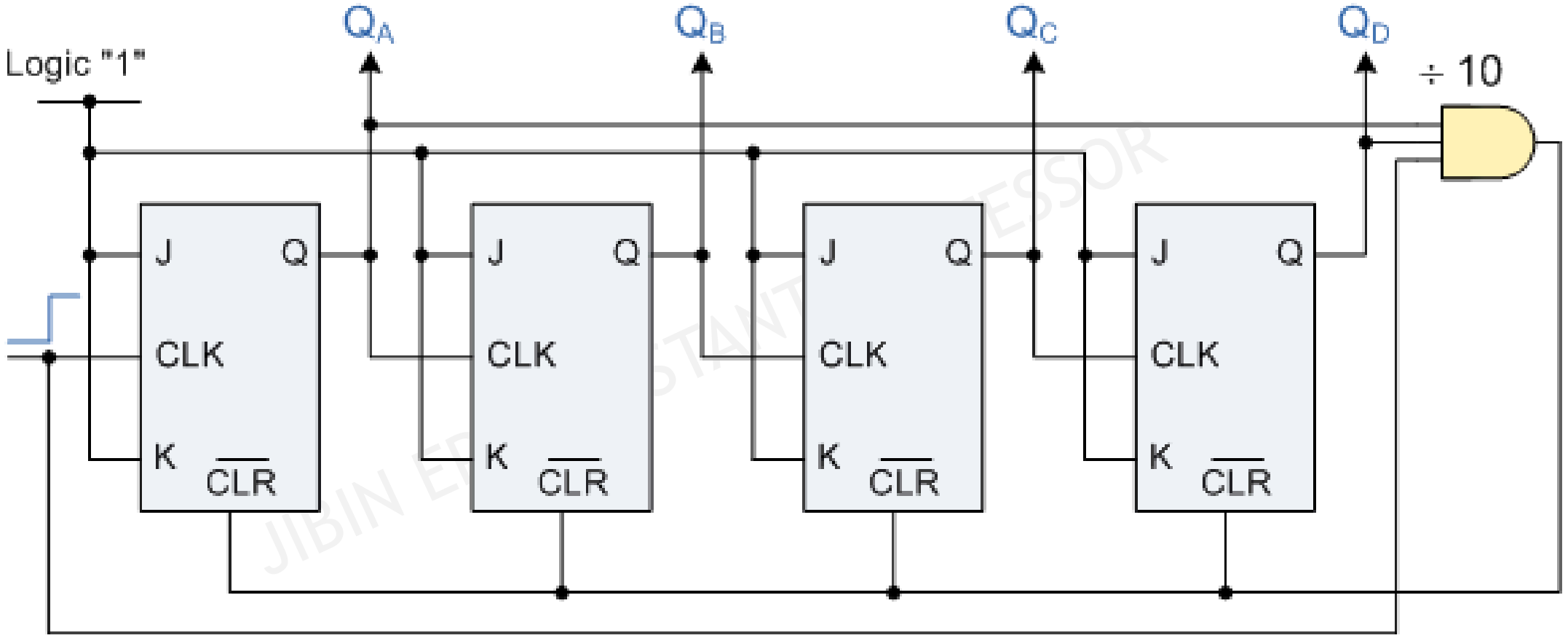
MODULE 4
(TOPIC-11)

MODULO COUNTER

https://youtu.be/LP4844Om570

Asynchronous Decade Counter

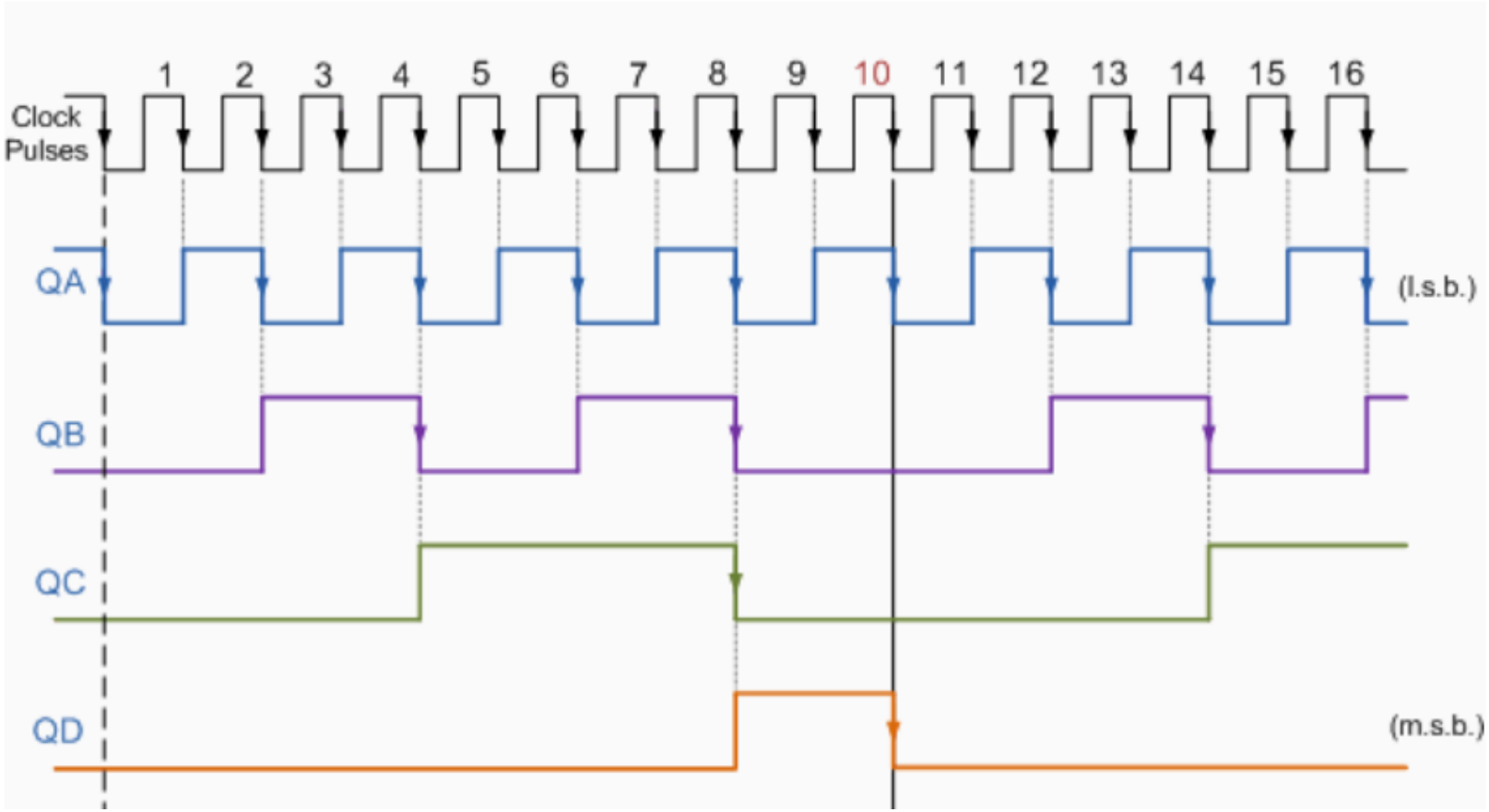
- This type of asynchronous counter counts upwards on each leading edge of the input clock signal starting from "0000" until it reaches an output "1010" (decimal 10). Both outputs QB and QD are now equal to logic "1" and the output from the NAND gate changes state from logic "1" to a logic "0" level and whose output is also connected to the CLEAR (CLR) inputs of all the J-K Flip-flops. This causes all of the Q outputs to be reset back to binary "0000" on the count of 10. Once QB and QD are both equal to logic "0" the output of the NAND gate returns back to a logic level "1" and the counter restarts again from "0000". We now have a decade or Modulo-10 counter



Decade Counter Truth Table

Clock Count	Output bit Pattern				Decimal Value
	QD	QC	QB	QA	
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	2
4	0	0	1	1	3
5	0	1	0	0	4
6	0	1	0	1	5
7	0	1	1	0	6
8	0	1	1	1	7
9	1	0	0	0	8
10	1	0	0	1	9
11	Counter Resets its Outputs back to Zero				

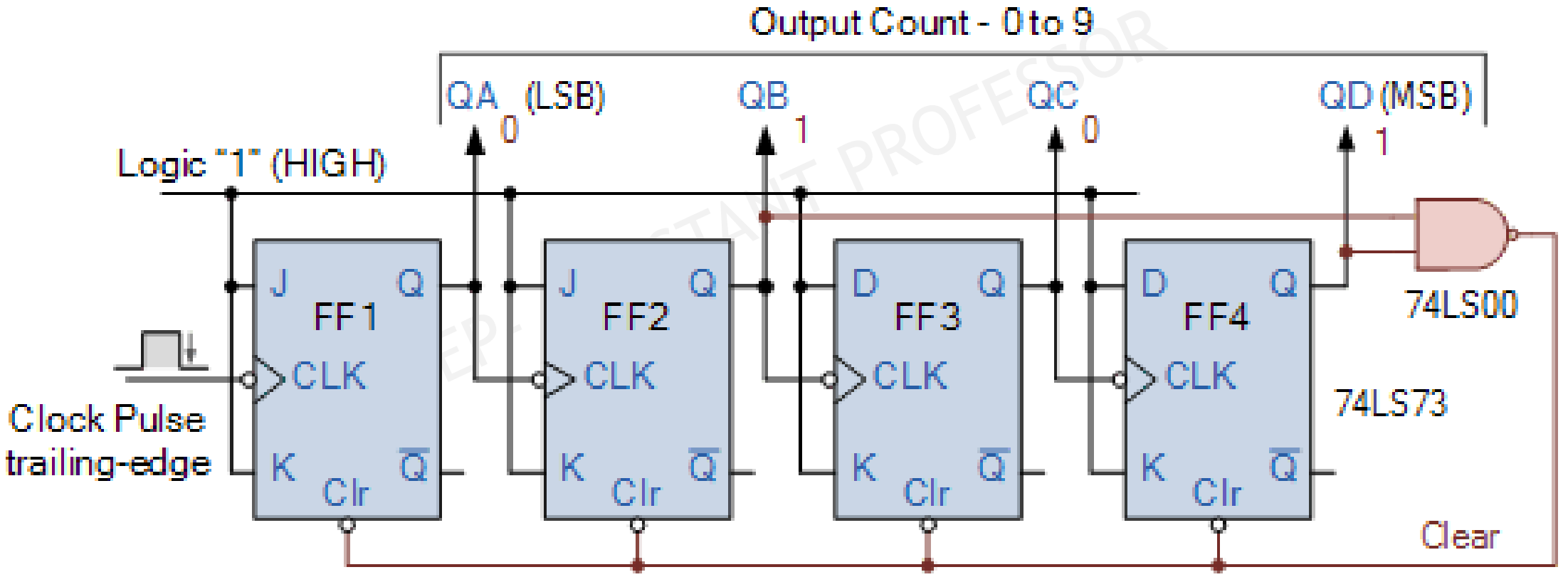
Decade Counter Timing Diagram



MODULO-10 COUNTER (BCD Counter)

- A good example of a modulo-m counter circuit which uses external combinational circuits to produce a counter with a modulus of 10 is the Decade Counter. Decade (divide-by-10) counters such as the TTL 74LS90, have 10 states in its counting sequence making it suitable for human interfacing where a digital display is required.
- The decade counter has four outputs producing a 4-bit binary number and by using external AND and OR gates we can detect the occurrence of the 9th counting state to reset the counter back to zero. As with other mod counters, it receives an input clock pulse, one by one, and counts up from 0 to 9 repeatedly.
- Once it reaches the count 9 (1001 in binary), the counter goes back to 0000 instead of continuing on to 1010. The basic circuit of a decade counter can be made from JK flip-flops that switch state on the negative trailing-edge of the clock signal as shown.

MODULO-10 COUNTER (BCD Counter)



MODULE 4
(TOPIC-12)

SYNCHONOUS COUNTER

https://youtu.be/PF39iLI1r4g
https://youtu.be/pPMXScXy0ZE

Steps to design synchronous counter

Decide the number of flipflops

Excitation table of Flipflop

State diagram and circuit excitation table

Obtain simplified equation using kmap

Draw the logic diagram

Synchronous Counter

- Synchronous Up counter
- Synchronous Down Counter
- Synchronous Up-Down Counter
- Synchronous BCD Counter

JK Excitation Table			
Qn	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

T Excitation Table		
Qn	Qn+1	T
0	0	0
0	1	1
1	0	1
1	1	0

1- Design Mod-5 Synchronous Counter using JK Flipflop

Step 1 : Determine the number of flip-flop needed

Flip-flops required are

$$2^n \geq N$$

Here $N = 5$ $\therefore$ $n = 3$ i.e. three flip-flops are required.

Step 2 : Type of flip-flop to be used : JK

Step 3 : Determine the excitation table for the counter.

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

Table 4.6.1 Excitation table for JK flip-flop

1- Design Mod-5 Synchronous Counter using JK Flipflop

Present State			Next State			Flip-flop Inputs					
Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	J_C	K_C	J_B	K_B	J_A	K_A
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	0	0	0	X	1	0	X	0	X
1	0	1	X	X	X	X	X	X	X	X	X
1	1	0	X	X	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X	X	X

Table 4.6.2 Excitation table for counter

1- Design Mod-5 Synchronous Counter using JK Flipflop

Step 4 : K-Map simplification

For J_C

Q_C \ Q_BQ_A	00	01	11	10
0	0	0	1	0
1	X	X	X	X

$J_C = Q_BQ_A$

For K_C

Q_C \ Q_BQ_A	00	01	11	10
0	X	X	X	X
1	1	X	X	X

$K_C = 1$

For J_B

Q_C \ Q_BQ_A	00	01	11	10
0	0	1	X	X
1	0	X	X	X

$J_B = Q_A$

For K_B

Q_C \ Q_BQ_A	00	01	11	10
0	X	X	1	0
1	X	X	X	X

$K_B = Q_A$

For J_A

Q_C \ Q_BQ_A	00	01	11	10
0	1	X	X	1
1	0	X	X	X

$J_A = \bar{Q}_C$

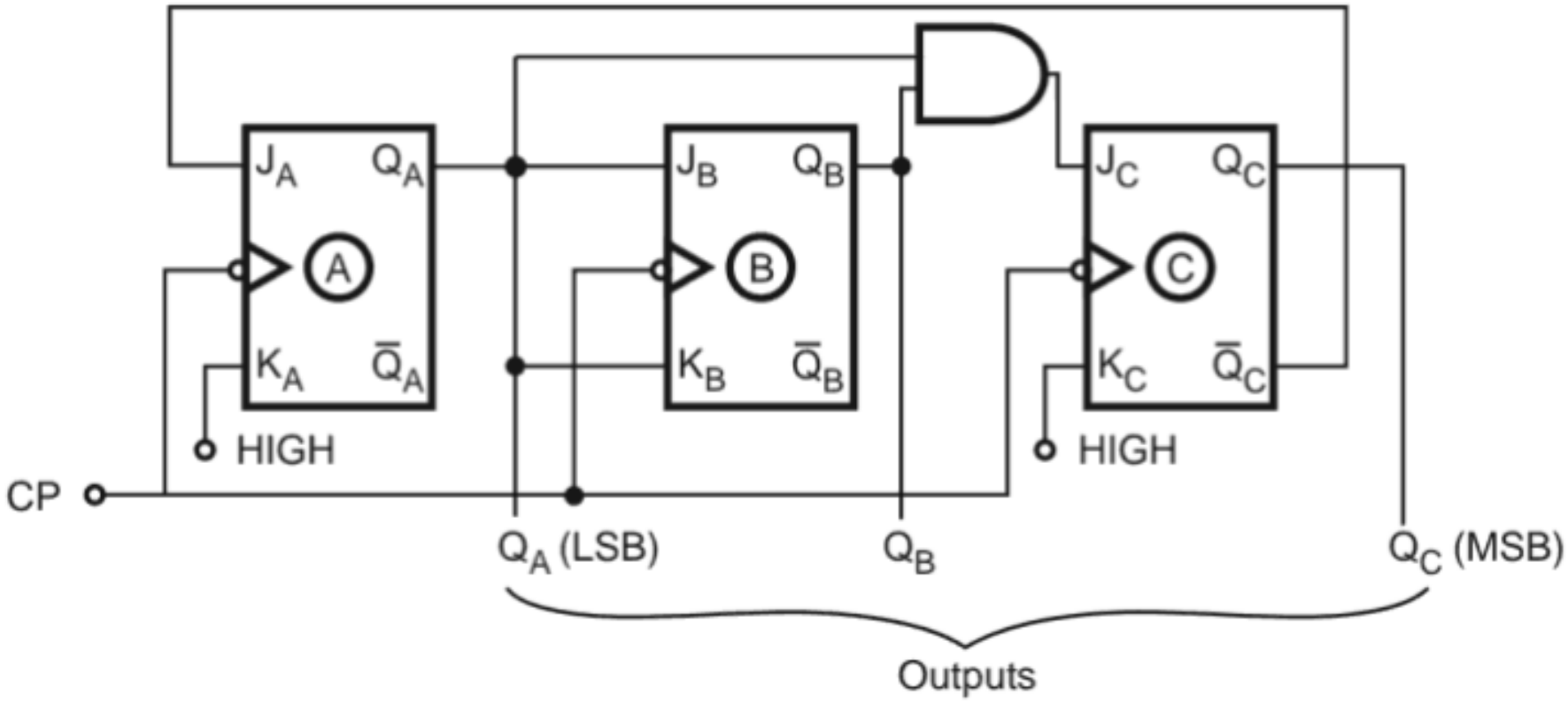
For K_A

Q_C \ Q_BQ_A	00	01	11	10
0	X	1	1	X
1	X	X	X	X

$K_A = 1$

1- Design Mod-5 Synchronous Counter using JK Flipflop

Step 5 : Draw the logic diagram



2- Using Positive Triggering SR Flipflop design a counter counts in the following sequence 000,111,110,101,100,011,010,001,000….

Step 1 : Determine the number of flip-flops needed

We know that $2^n \geq N$. Here, $N = 8$ $\therefore$ $n = 3$

Step 2 : Type of flip-flop to be used : SR

2- Using Positive Triggering SR Flipflop design a counter counts in the following sequence 000,111,110,101,100,011,010,001,000….

Step 3 : Determine the excitation table for counter.

Here, the next state for each present state is written according to given sequence. For example, the next state for the present state 000 is 111.

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

Table 4.6.5 Excitation table of SR flip-flop

Present state			Next state			Flip-flop inputs					
A	B	C	A^+	B^+	C^+	S_A	R_A	S_B	R_B	S_C	R_C
0	0	0	1	1	1	1	0	1	0	1	0
0	0	1	0	0	0	0	x	0	x	0	1
0	1	0	0	0	1	0	x	0	1	1	0
0	1	1	0	1	0	0	x	x	0	0	1
1	0	0	0	1	1	0	1	1	0	1	0
1	0	1	1	0	0	x	0	0	x	0	1
1	1	0	1	0	1	x	0	0	1	1	0
1	1	1	1	1	0	x	0	x	0	0	1

Table 4.6.6 Excitation table for counter

2- Using Positive Triggering SR Flipflop design a counter counts in the following sequence 000,111,110,101,100,011,010,001,000….

Step 4 : K-map simplification.

For S_A

A \ BC	00	01	11	10
0	(1)	0	0	0
1	0	X	X	X

$S_A = \overline{A}\,\overline{B}\,\overline{C}$

For R_A

A \ BC	00	01	11	10
0	0	X	X	X
1	(1)	0	0	0

$R_A = A\,\overline{B}\,\overline{C}$

For S_B

A \ BC	00	01	11	10
0	1	0	X	0
1	1	0	X	0

$S_B = \overline{B}\,\overline{C}$

For R_B

A \ BC	00	01	11	10
0	0	X	0	1
1	0	X	0	1

$R_B = B\,\overline{C}$

For S_C

A \ BC	00	01	11	10
0	1	0	0	1
1	1	0	0	1

$S_C = \overline{C}$

For R_C

A \ BC	00	01	11	10
0	0	1	1	0
1	0	1	1	0

$R_C = C$

2- Using Positive Triggering SR Flipflop design a counter counts in the following sequence 000,111,110,101,100,011,010,001,000….

Step 5 : Draw logic diagram

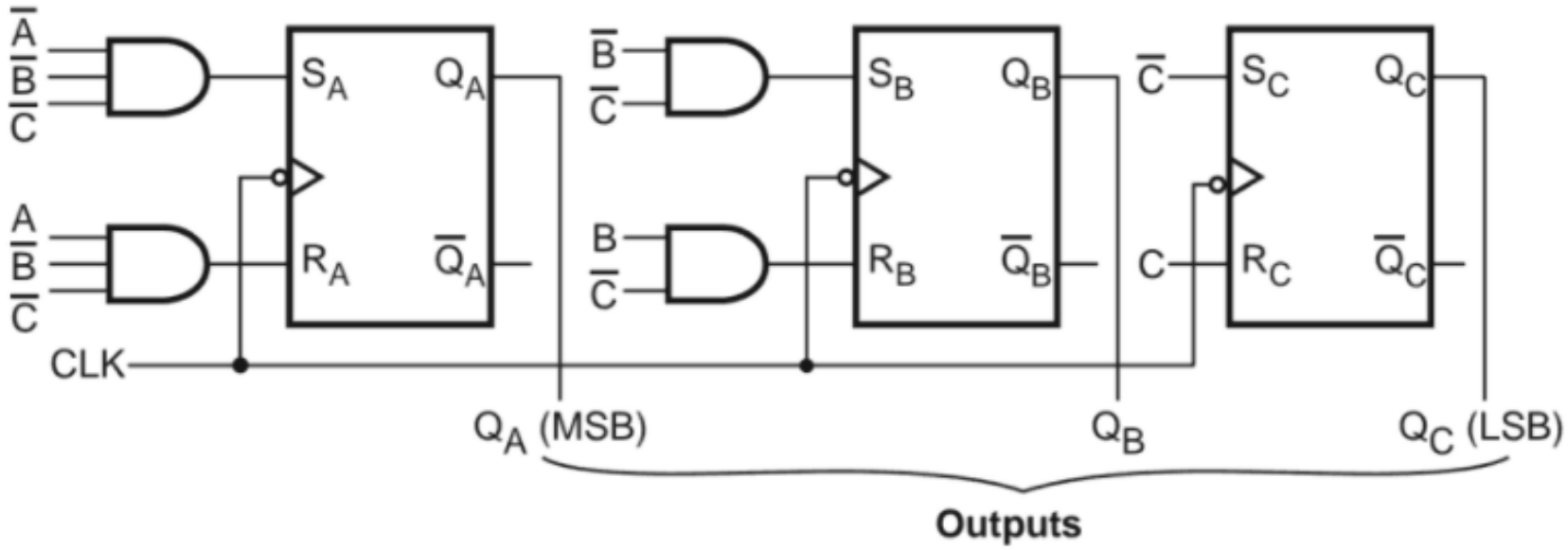


Fig. 4.6.5 (b) Logic diagram

2- Using Positive Triggering SR Flipflop design a counter counts in the following sequence 000,111,110,101,100,011,010,001,000….

3- Design 3 Bit Synchronous UP-Down Counter using T Flipflop

Step 1 : Determine Number of Flipflops

It require 3 T Flipflops

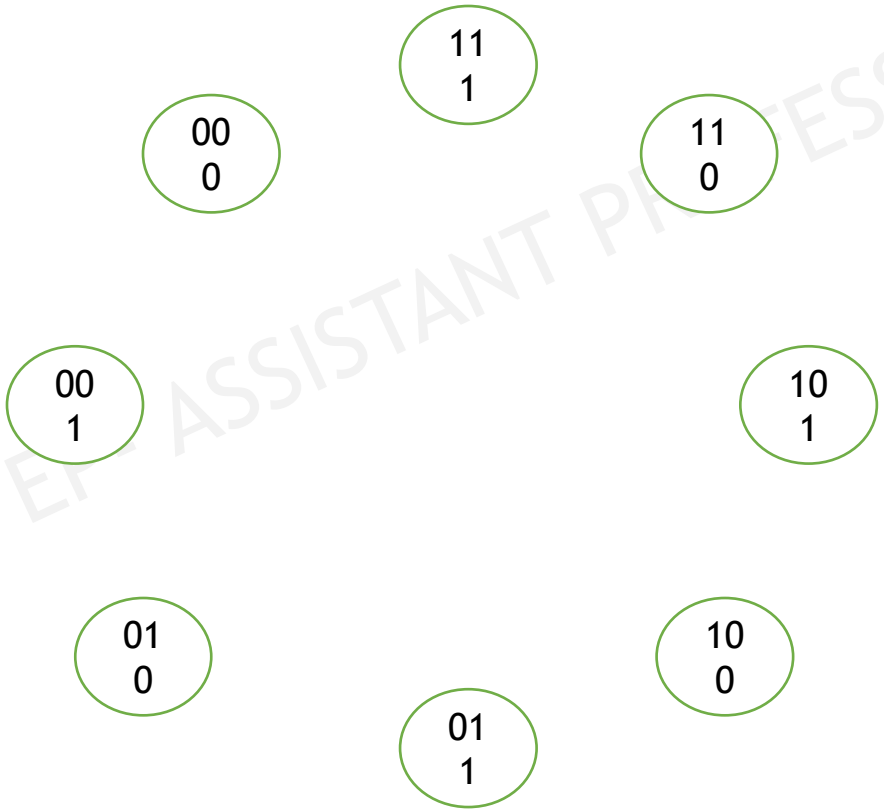
3- Design 3 Bit Synchronous UP-Down Counter using T Flipflop

Step 2 : The Excitation Table

T Excitation Table		
Qn	Qn+1	T
0	0	0
0	1	1
1	0	1
1	1	0

3- Design 3 Bit Synchronous UP-Down Counter using JK Flipflop

Step 3 : Draw the State Diagram



M=0
UPCOUNTER

M=1
DOWN COUNTER

3- Design 3 Bit Synchronous UP-Down Counter using T Flipflop

Step 3B : The Circuit Excitation Table

Mode	Present State			Next State			Required Excitation		
	Q2	Q1	Q0	Q2*	Q1*	Q0*	T2	T1	T0
0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	1	0	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	1	1	1
1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	0	0	0	1
1	0	1	0	0	0	1	0	1	1
1	0	1	1	0	1	0	0	0	1
1	1	0	0	0	1	1	1	1	1
1	1	0	1	1	0	0	0	0	1
1	1	1	0	1	0	1	0	1	1
1	1	1	1	1	1	0	0	0	1

T Excitation Table		
Qn	Qn+1	T
0	0	0
0	1	1
1	0	1
1	1	0

Design 3 Bit Asynchronous UP-Down Counter using T Flipflop

K Map Simplification and Logic Diagram (Kindly Workout)

PREPARED BY

SHASTRA TECHNICAL INSTITUTE

KTU | PSC | UPSC

ശാസ്ത്ര soaring to success

Jibin EP
Assistant Professor
Department of Electronics and Communication
Eranad Knowledge City Technical Campus Manjeı
Contact :9633908979, 70121716607
Email : epjibin@gmail.com, jibin@ekc.edu.in

YouTube

https://www.youtube.com/ShastraTechnicalInstitute