SEQUENTIAL CIRCUITS 1

Unit-4

Topics:

- Classification of sequential circuits
- Basic Flip Flops-Triggering and Excitation Tables
- Conversion of flip flops

INTRODUCTION

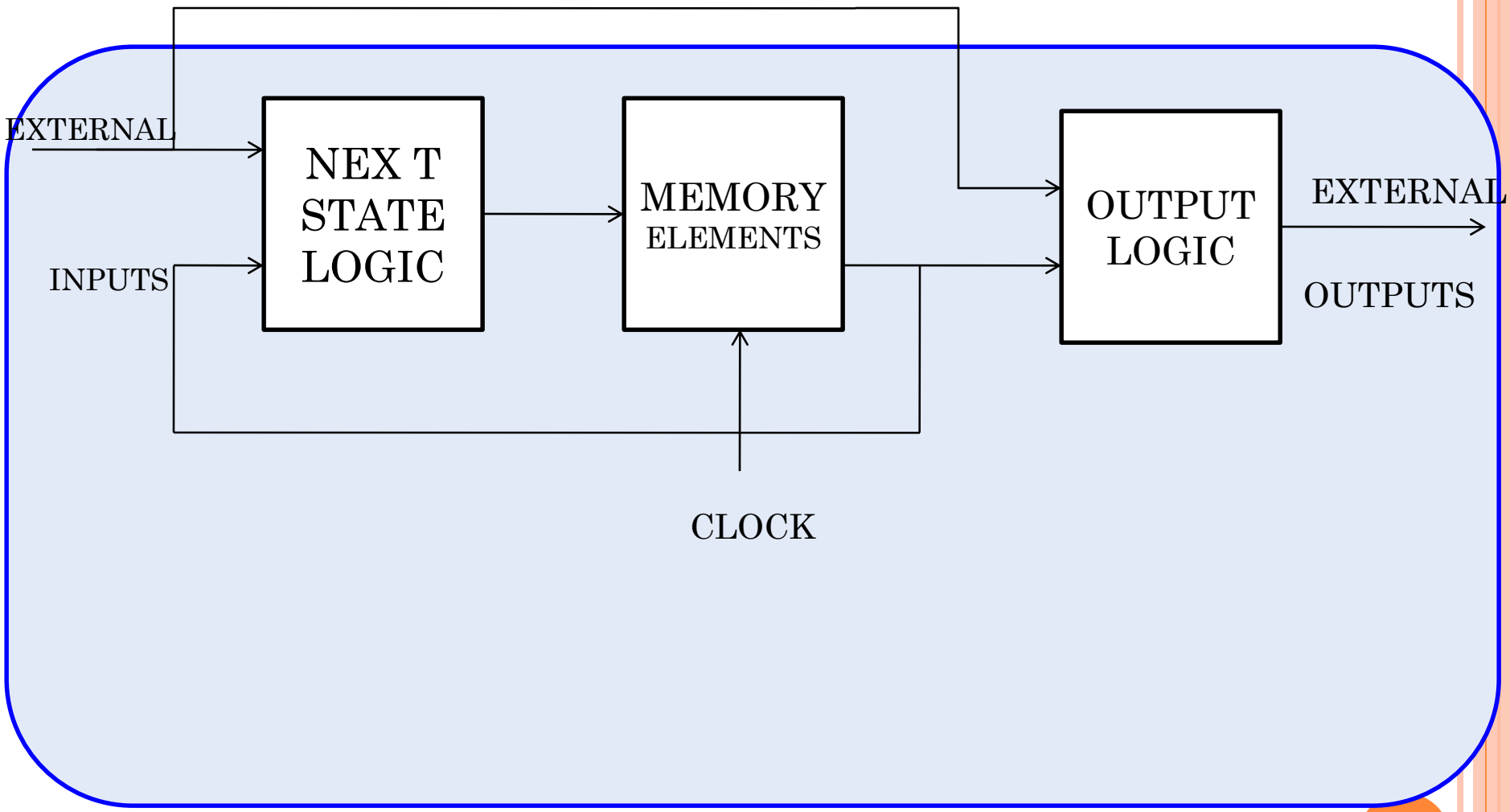
So far, all of the logic circuits we have studied were basically based on the analysis and design of combinational digital circuits.

The other major aspect of a digital system is the *ANALYSIS* and *DESIGN* of sequential digital circuits.

SEQUENTIAL LOGIC CIRCUITS

The main characteristic of combinational logic circuits is that their output values depend on their present input values.

Sequential logic circuits differ from combinational logic circuits because they contain memory elements so that their output values depend on both present and past input values.



Block diagram of a sequential circuit

❖**Combinational Circuits**

– the logic circuits whose outputs at any instant of time depend only on the input signals present at the time.

❖**Sequential Circuits** ***– the logic circuits whose outputs at any instant of time depend the present inputs as well as on the past outputs. It consists of combinational circuits to which memory elements are connected to*** **form a feedback path.**

MEMORY ELEMENTS

- These are the devices capable of restoring binary information within them.
- The binary info stored in memory elements at any given time defines the state of sequential circuit.
- The sequential circuit receives binary information from external inputs. These inputs together with the present state of memory elements determine the binary value at output terminals.

❖***Memory Element*** **–** is a medium in which one bit of information (0 or 1) can be stored or retained until necessary, and thereafter its contents can be replaced by a new value**.**

Synchronous and Asynchronous Sequential Logic:-

- **Synchronous**
  - the timing of all state transitions is controlled by a common clock
  - changes in all variables occur simultaneously
- **Asynchronous**
  - state transitions occur independently of any clock and normally dependent on the timing of transitions in the input variables
  - changes in more than one output do not necessarily occur simultaneously

MASTER CLOCK GENERATOR

- Synchronization is achieved by timing device called master clock generator which generates a periodic train of clock pulses.

CLOCKED SEQUENTIAL CIRCUIT

- Synchronous sequential circuits that use clock pulses in the inputs of the memory elements are called clock sequential circuit. The memory elements used in clocked sequential circuits are called flip-flops.

Clock:-

- A clock is a special device that whose output continuously alternates between 0 and 1.
- The time it takes the clock to change from 1 to 0 and back to 1 is called the clock period, or clock cycle time.
- The clock frequency is the inverse of the clock period. The unit of measurement for frequency is the hertz.
- Clocks are often used to synchronize circuits.

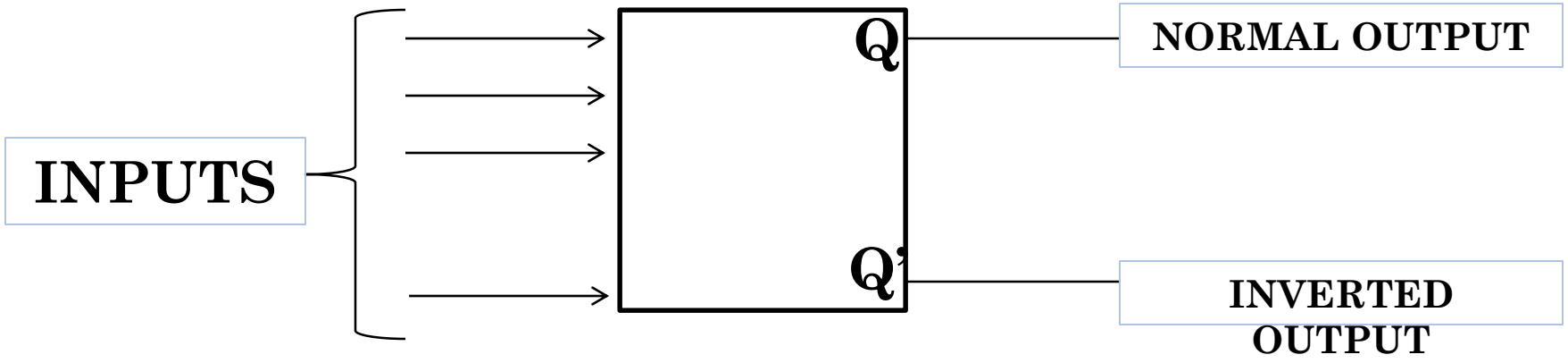
Memory Devices

Latches A *latch* is a memory element whose excitation signals control the state of the device. A latch has two stages *set* and *reset*. *Set* stage sets the output to 1. *Reset* stage set the output to 0.

Flip-flops A *flip-flop* is a memory device that has clock signals control the state of the device.

FLIP-FLOPS

- ***The basic 1-bit digital memory circuit is known as a flip-flop. It can have two states either the 1 state or the 0 state.***
- ***It is also known as a bistable multivibrator. Flip-flops can be obtained by using NAND or NOR gates.***



BLOCK DIAGRAM OF A FLIP-FLOP

TYPES OF FLIP-FLOPS

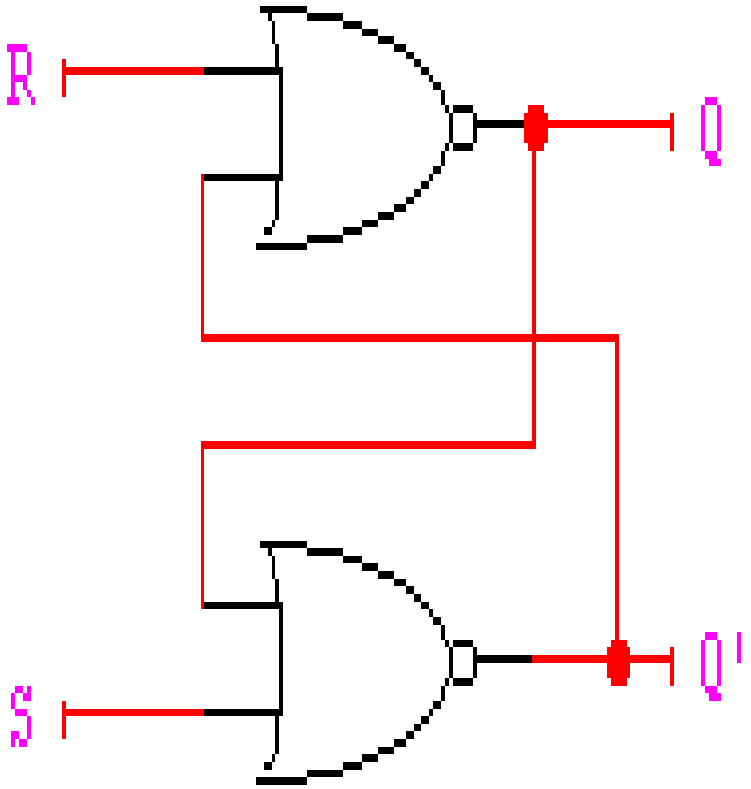
- RS flip-flop
- D flip-flop
- T flip-flop
- JK flip-flop
- Master Slave JK flip-flop

LATCH

RS Latch

The ***RS latch*** is the basic memory element consists of two cross-coupled NOR gates. It has two input signals, *S* set signal and *R* reset signal. It also has two outputs *Q* and *Q′*; and two states, a set state when *Q* = 1 and a reset state when *Q* = 0 (*Q′* = 1).

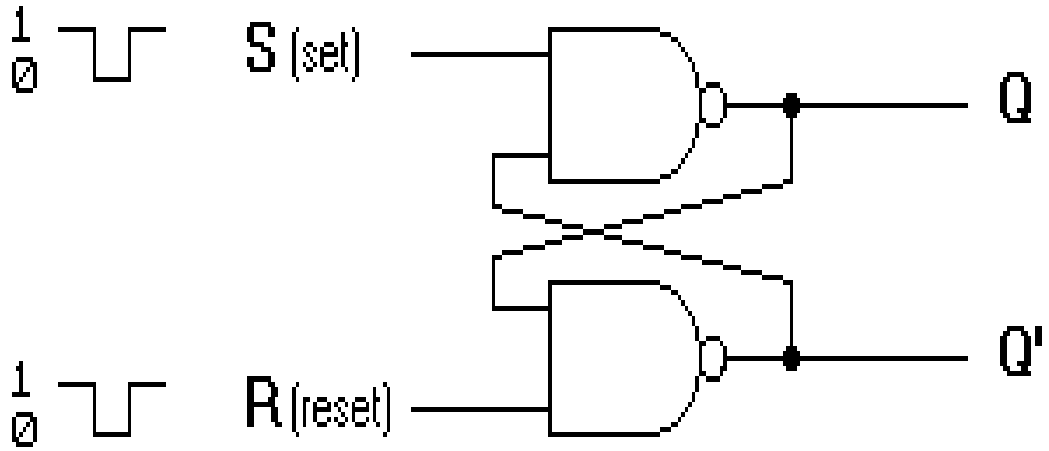
S-R (SET-RESET)



S	*R*	*Q*
0	0	hold
0	1	0 reset
1	0	1 set
1	1	unstable

NAND implementation of Flip Flops

- **Logic Diagram:**

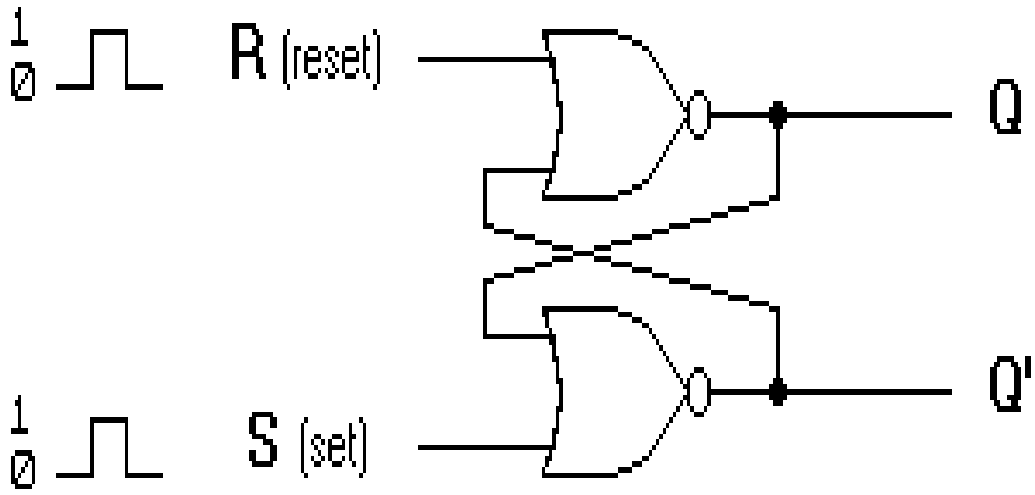


TRUTH TABLE

S	R	Q	Q’
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0
0	0	Invalid Value	

NOR implementation of Flip Flops

- **Logic Diagram:**

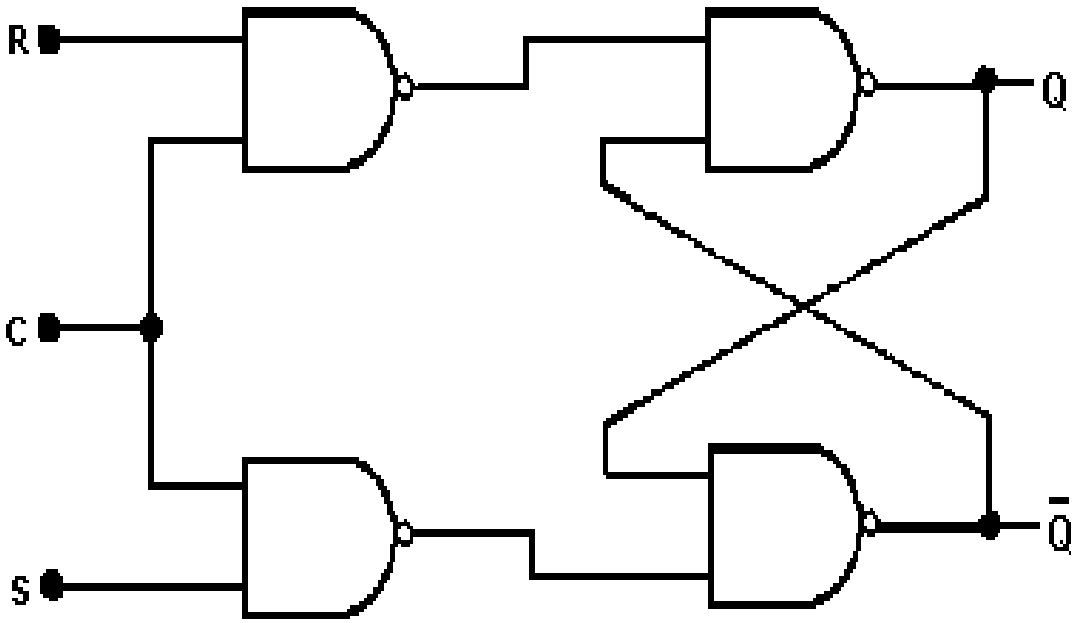


TRUTH TABLE

S	R	Q	Q’
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0 Invalid Value	

Clocked RS Flip-flop

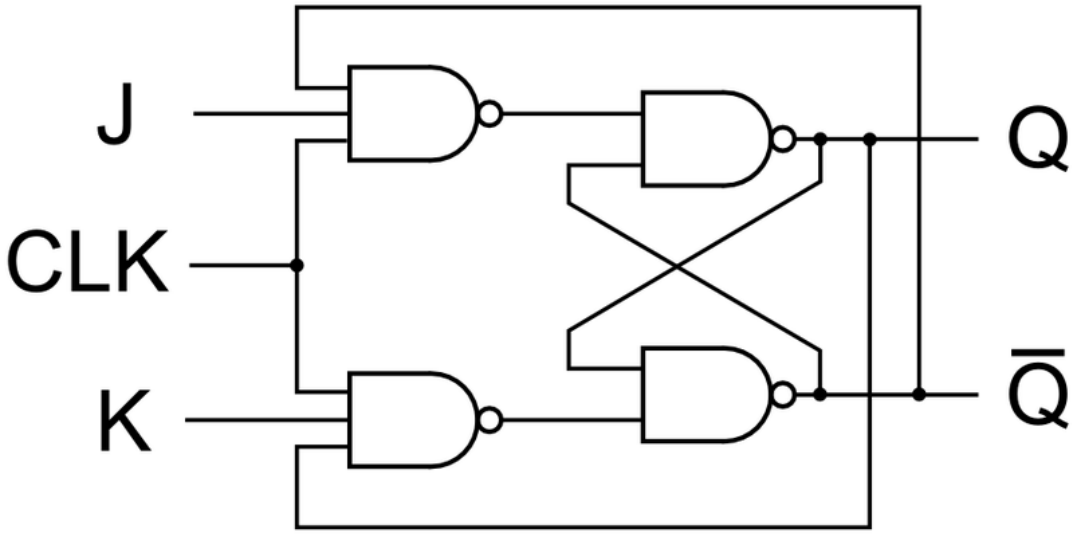
- **Logic Diagram:**



TRUTH TABLE

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

JK FLIP FLOP



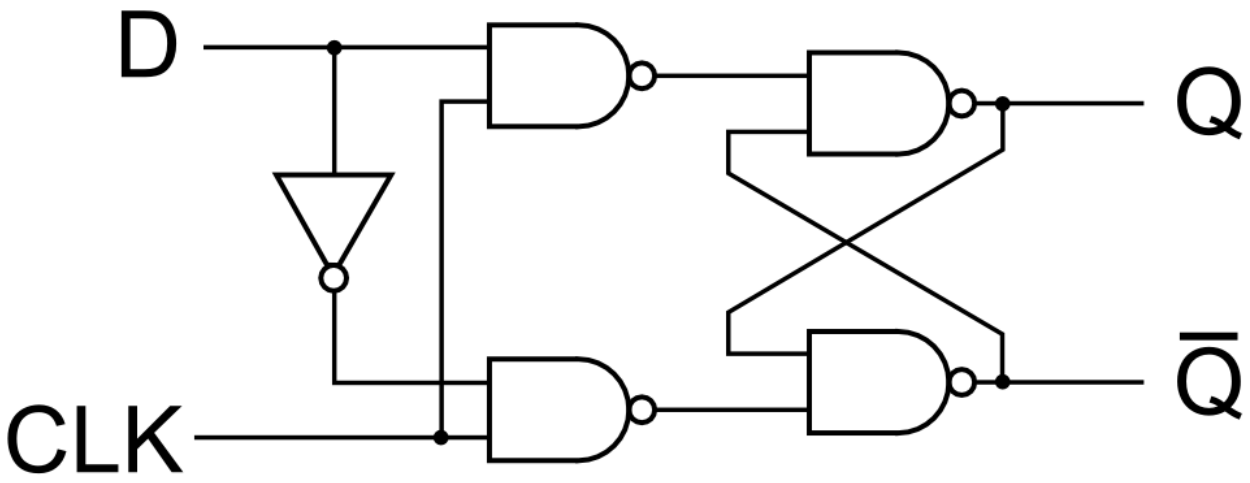
Truth Table

J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q}_0$ (toggles)

D FLIP-FLOP

- *Make S and R complements of each other*
  - *Eliminates 1s catching problem*
  - *Can't just hold previous value (must have new value ready every clock period)*
  - *Value of D just before clock goes low is what is stored in flip-flop*
  - *Can make R-S flip-flop by adding logic to make* $D = S + R' Q$.

D FLIP FLOP



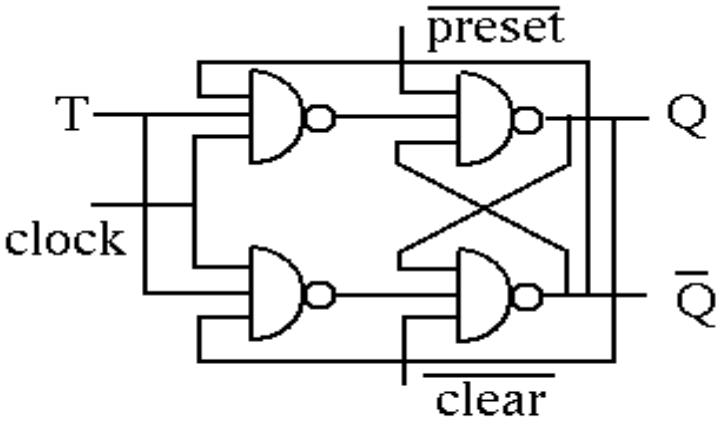
Truth Table

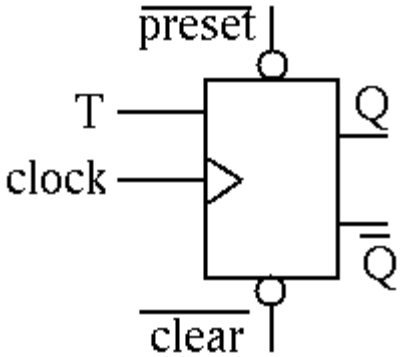
clk	D	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	Q	$\overline{Q}$
1	0	0	1
1	1	1	0

T FLIP-FLOP

With a slight modification of a J-K flip-flop, we can construct a new flip flop called the ***T-FLIPFLOP*** **.** If the two inputs **J and K** of a **J-K flip-flop** are tied together it is referred to as a **T-flip-flop**. Hence a T flip-flop has only one input T and two outputs Q and Q'. The name T flip-flop actually indicates the fact that the flip-flop has the ability to ***TOGGLE*****.** It has actually only two states ***TOGGLE STATE*** and ***MEMORY STATE***.

T FLIP FLOP





TRUTH TABLE

T	Q_{t+1}
0	Q_t
1	$\overline{Q}_t$

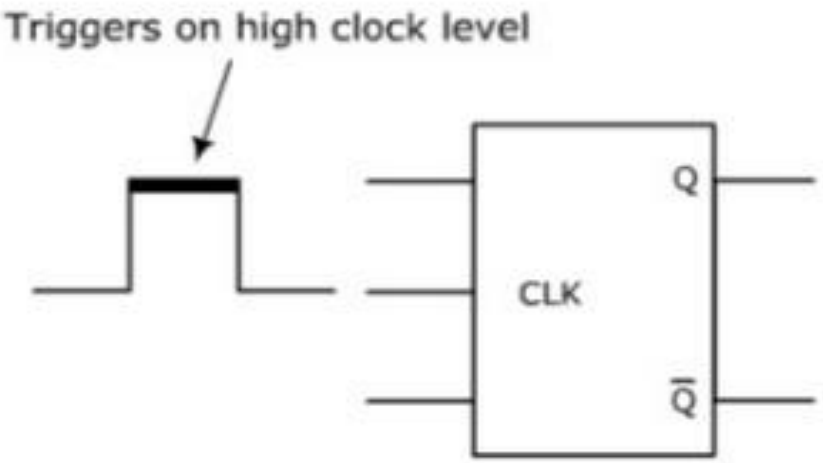
Triggering

- Sequential circuits are dependant on clock pulses applies to their inputs.
- The result of flip-flop responding to a clock input is called **clock pulse triggering**, of which there are four types. Each type responds to a clock pulse in one of four ways :-

1. High level triggering
2. Low level triggering
3. Positive edge triggering
4. Negative edge triggering

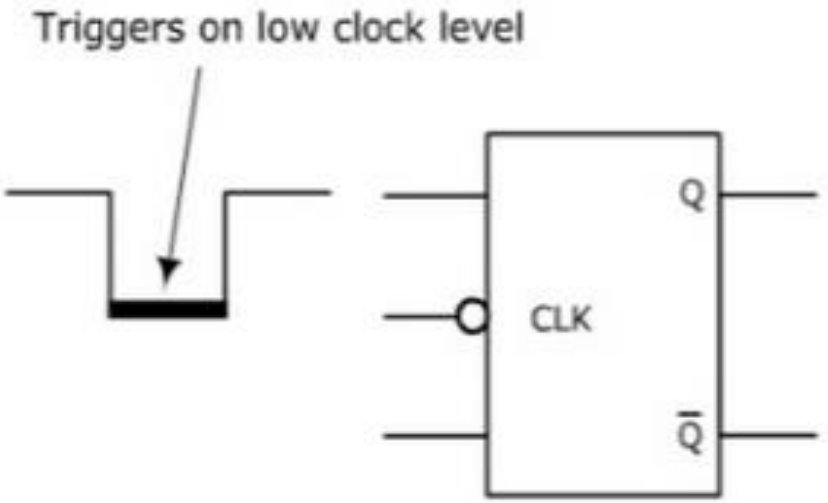
High Level Triggering

One type of flip-flop responds to a clock signal during the time at which it is in the logic High state. This type is identified by a straight lead at the clock input, as shown below.



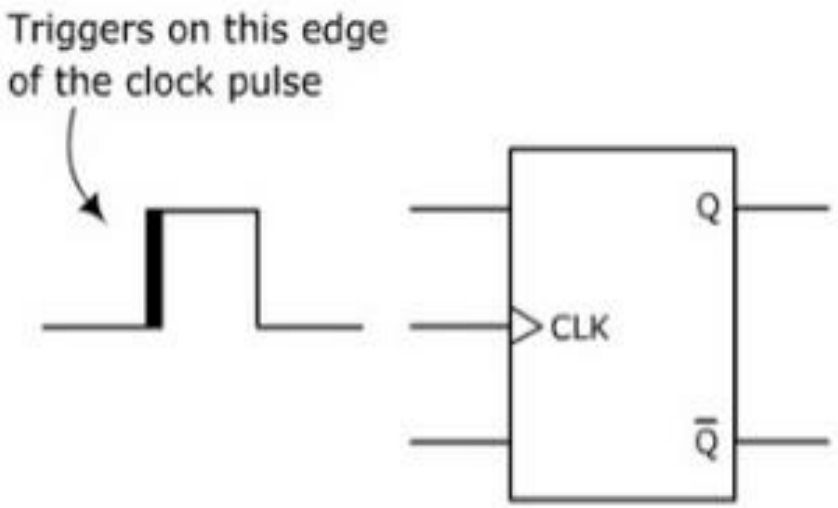
Low Level Triggering

Another type of flip-flop responds to a clock signal during the time at which it is in the logic Low state. This type is identified by a clock input lead with a low-state indicator bubble, as shown below.



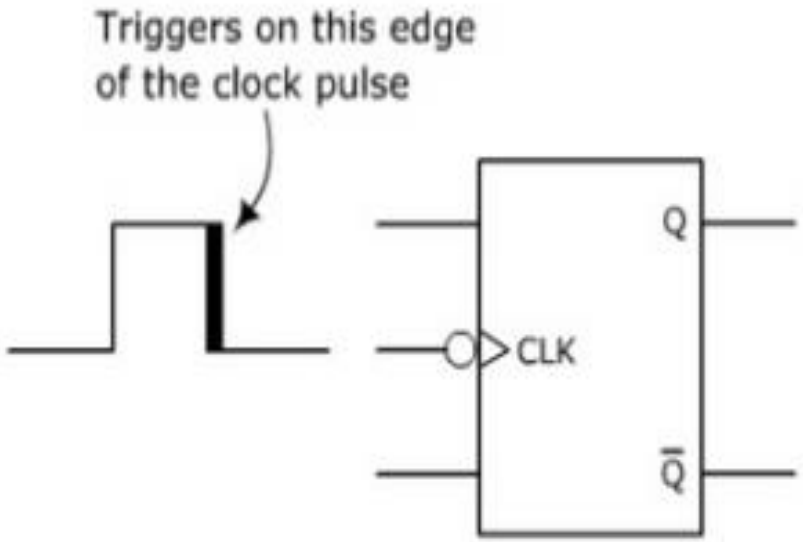
Positive Edge Triggering

A third type of flip-flop responds to a clock signal during the low-to-high transition of a clock pulse. This type is identified by a clock input lead with a triangle, as shown below.



Negative Edge Triggering

The fourth type of flip-flop responds to a clock signal during the high-to-low transition of a clock pulse. This type is identified by a clock input lead with a low-state indicator and triangle, as shown below.



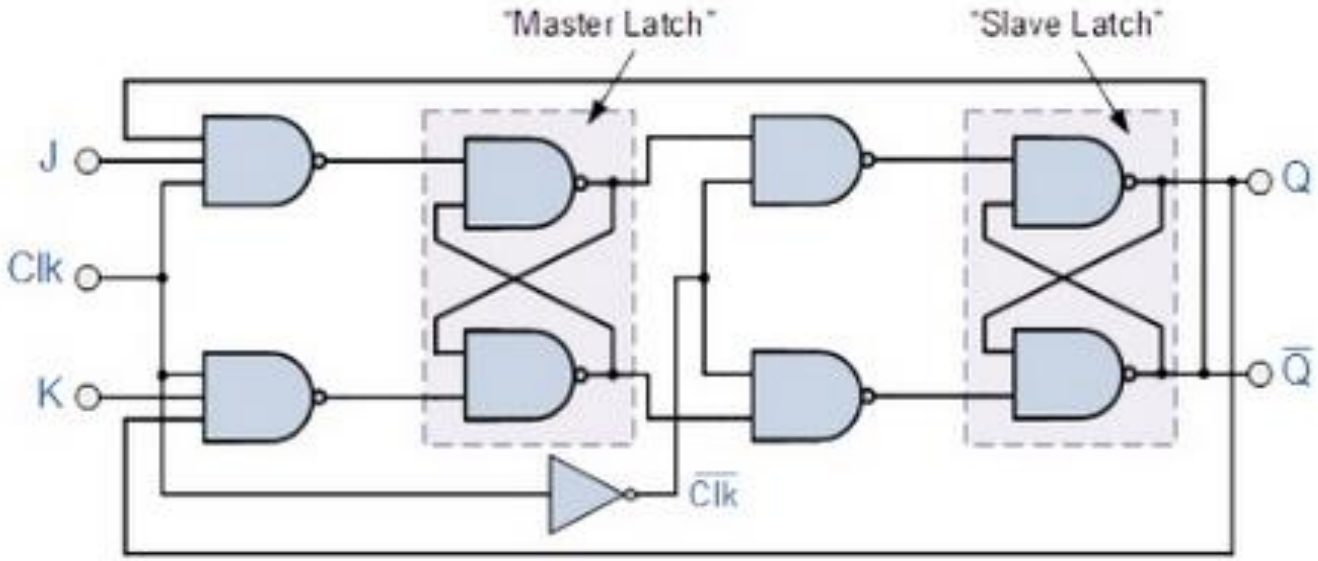
Race around condition

This condition occur when j=k=1 i.e when the latch is in toggle mode.

This can be avoided by

- Using edge triggering J-k flip-flop
- Using master slave flip-flop

Master slave flip-flop



Clk	j	k	Q_n	$\overline{Q_n}$
↑	0	0	No change	
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	$\overline{Q_n}$	Q_n

Toggle

EXCITATION TABLES:

The excitation tables are used for conversion of flip flops.

Excitation table

SR flip-flop:-

Clk	S	R	Q_n	$\overline{Q_n}$
↑	0	0	No change	
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	Race	Race

Excitation table:-

Present state of Q o/p	Next state of Q o/p	S_n Input	R_n input
0	0	0	×
0	1	1	0
1	0	0	1
1	1	×	0

JK flip-flop:-

Clk	J	k	Q_n	$\overline{Q_n}$
↑	0	0	No change	
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	$\overline{Q_n}$	Q_n

Excitation table:-

Present state of Q o/p	Next state of Q o/p	J_n Input	K_n input
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

D flip-flop:-

Clk	D	Q
↑	0	0
↑	1	1

Excitation table:-

Present state of Q o/p	Next state of Q o/p	D_n Input
0	0	0
0	1	1
1	0	0
1	1	1

T flip-flop:-

Clk	T	Q
↑	0	No change
↑	1	toggle

Excitation table:-

Present state of Q o/p	Next state of Q o/p	T_n Input
0	0	0
0	1	1
1	0	1
1	1	0

Conversion Between FlipFlop Types

Example: Use JK-FF to realize D-FF

1) Start transition table for D-FF

2) Create K-maps to express J and K as functions of inputs (D, Q)

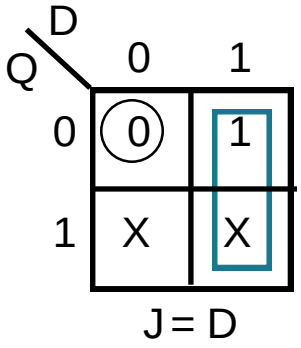
3) Fill in K-maps with appropriate values for J and K
to cause the same state transition as in the D-FF transition table

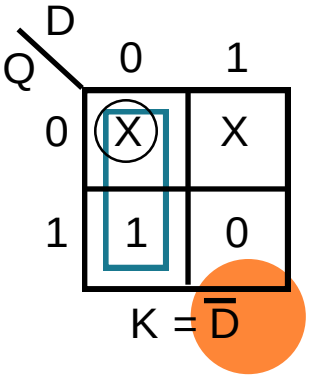
D	Q	Q^+	J	K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	1	X	0

State-Table

Q	Q^+	R	S	J	K	T	D
0	0	X	0	0	X	0	0
0	1	0	1	1	X	1	1
1	0	1	0	X	1	1	0
1	1	0	X	X	0	0	1

e.g.
when D=Q=0, then $Q^+ = 0$
the same transition $Q \rightarrow Q^+$
is realize with J=0, K=X

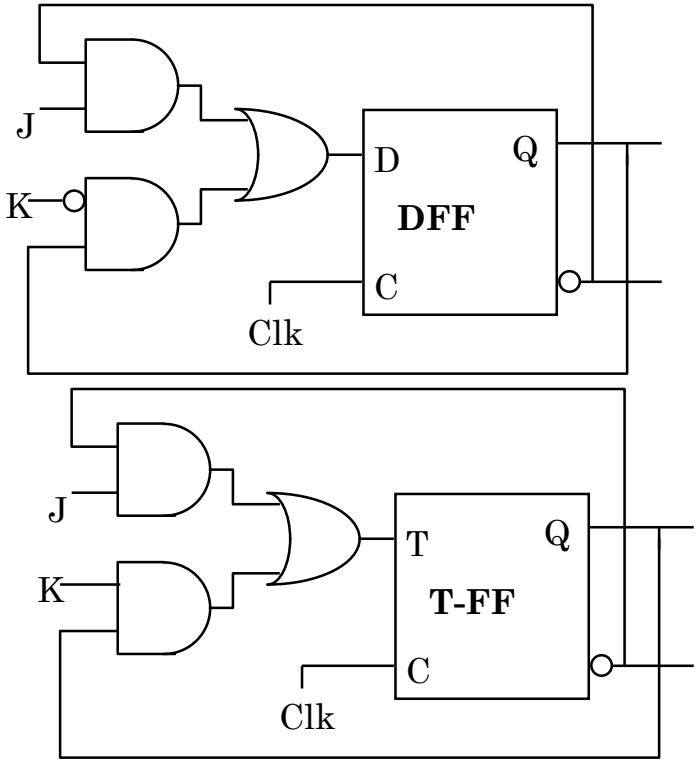


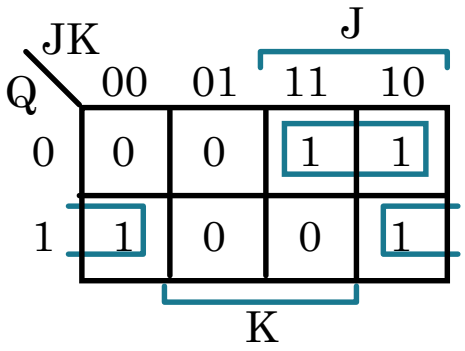


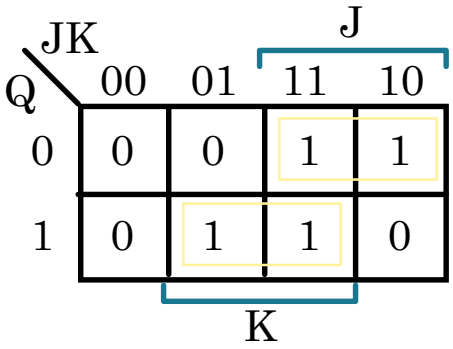
Conversion Between FlipFlops

Another Example: Implement JK-FF using a D-FF

J	K	Q	Q+	D	T
0	0	0	0	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1







d= jQ + Kq

t= jQ + kq