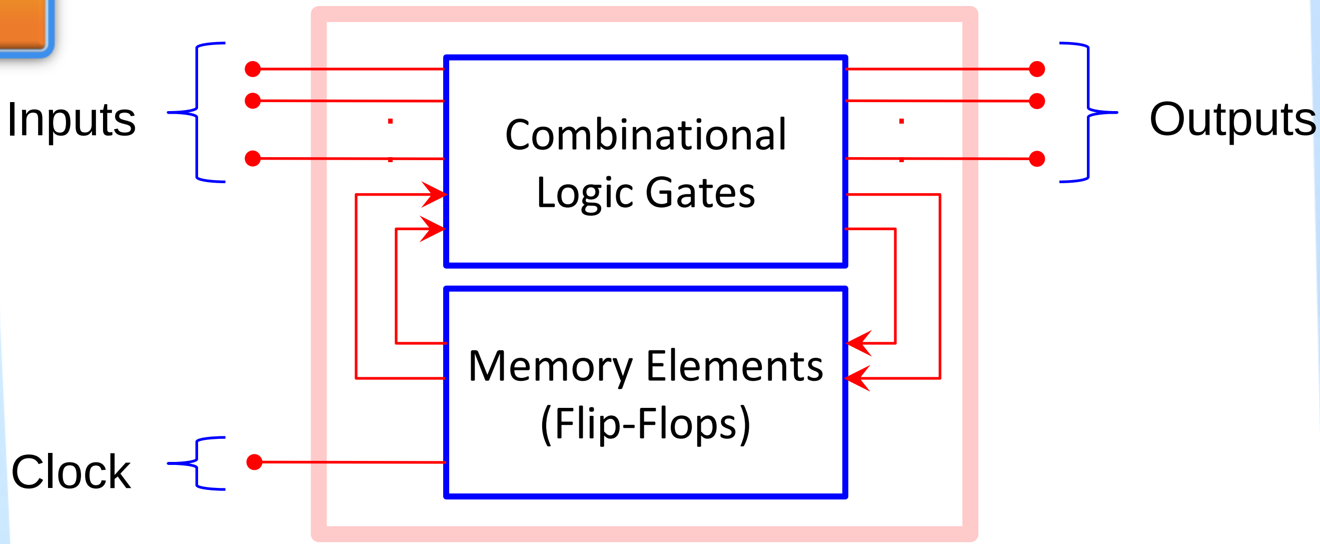
FLIP–FLOPS

SEQUENTIAL CIRCUIT

DIFFERENCES WITH COMBINATIONAL CIRCUIT

Introduction



DIFFERENCES WITH COMBINATIONAL CIRCUIT

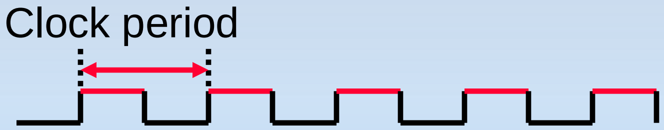
- Sequential logic can have one or more, inputs and one or more outputs.
- However, the outputs are a function of both the present value of the inputs and also the previous output values.
- Sequential logic requires memory to store these previous outputs values.
- Unike combinational logic circuits, a sequential logic circuit uses a clock.
- Examples: Flipflops, Latch, Counters, Shift Registers etc

CLOCK AND FREQUENCY

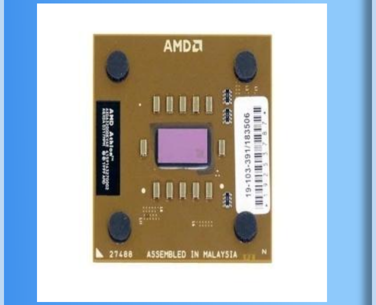
- A signal used to synchronize the operations of an electronic system. Clock pulses are continuous, precisely spaced changes in voltage.
- In electronics and especially synchronous digital circuits, a clock signal is a particular type of signal that oscillates between a high and a low state and is utilized to coordinate actions of circuits.
- Although the word signal has a number of other meanings, the term here is used for "transmitted energy that can carry information".
- A clock signal is produced by a clock generator.

CLOCK AND FREQUENCY

- Frequency is inverse of time period.
- Clock speed refers to the number of pulses per second generated by an oscillator that sets the tempo for the processor.
- 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.
- Clock speed is usually measured in MHz megahertz, or millions of pulses per second or GHz gigahertz, or billions of pulses per second.
- Today's personal computers run at a clock speed in the hundreds of megahertz and some exceed one gigahertz.



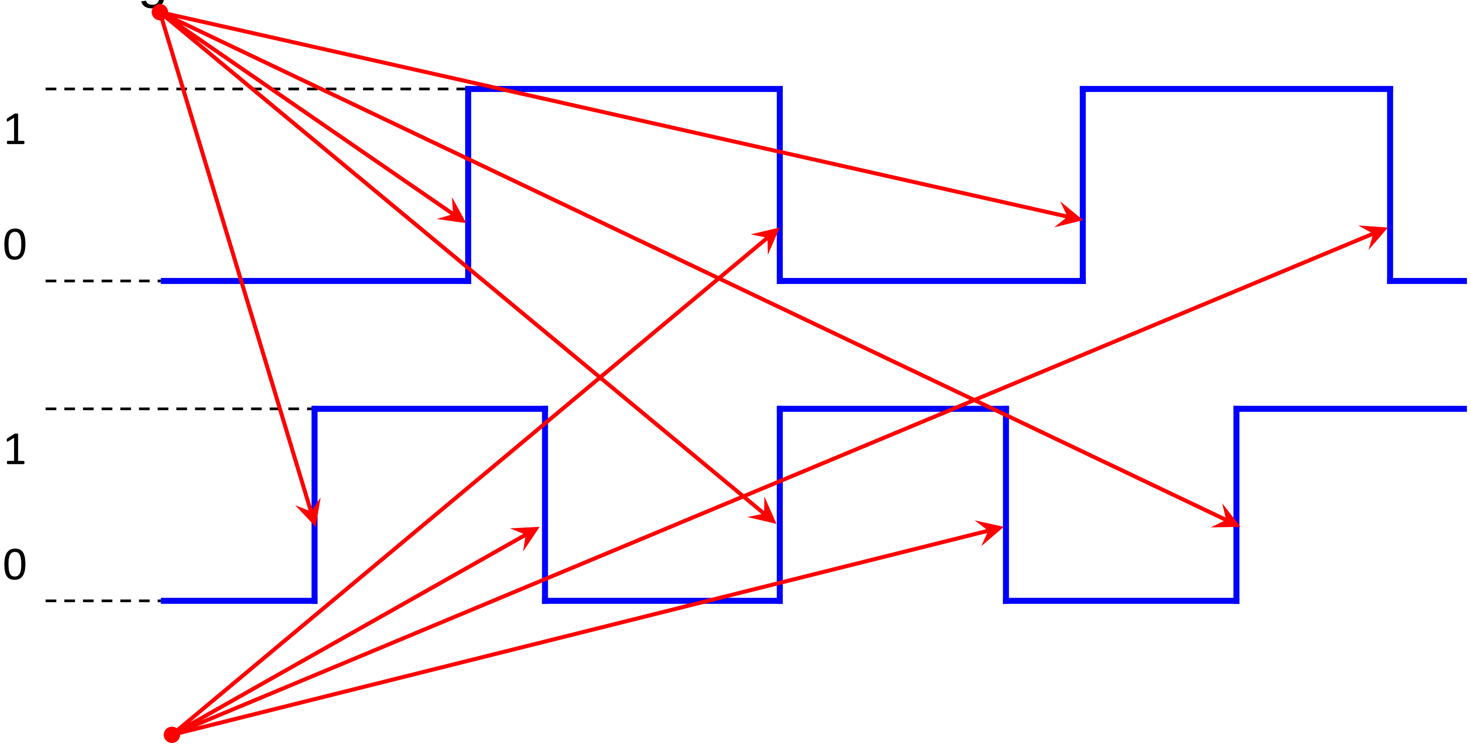
MORE ABOUT CLOCKS



- Clocks are used extensively in computer architecture.
- All processors run with an internal clock.
  - Modern chips run at frequencies up to 3.2 GHz.
  - This works out to a cycle time as little as 0.31 ns!
- Memory modules are often rated by their clock speeds too—examples include “PC133” and “DDR400” memory.
- Be careful...higher frequencies do not always mean faster machines!
  - You also have to consider how much work is actually being done during each clock cycle.

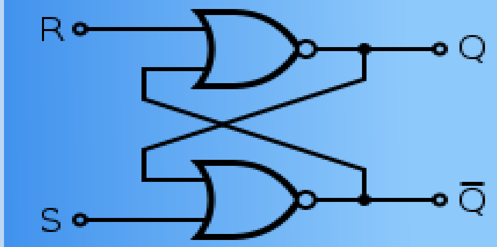
CLOCK AND FREQUENCY

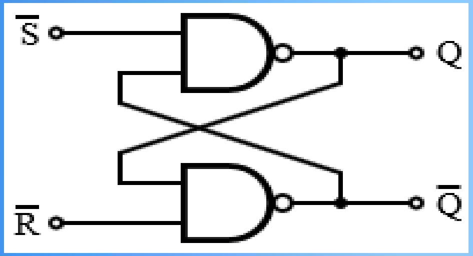
Positive Edge Transition



Negative Edge Transition

LATCH





The primary difference between a flip-flop and latch is the **EN/CLOCK input.**

Latch: Bi-stable memory device with level sensitive triggering (no clock) meaning the latch's output changes on the level (high or low) of the EN input, watches all of its inputs continuously and changes its outputs, independent of a clocking signal.

LATCH: OTHER DIFFERENCES

- It’s difficult to control the timing of latches in a large circuit.
- Clocks tell us when to write to our memory.
- Flip-flops allow us to quickly write the memory at clearly defined times.
- Used together, we can create circuits without worrying about the memory timing.

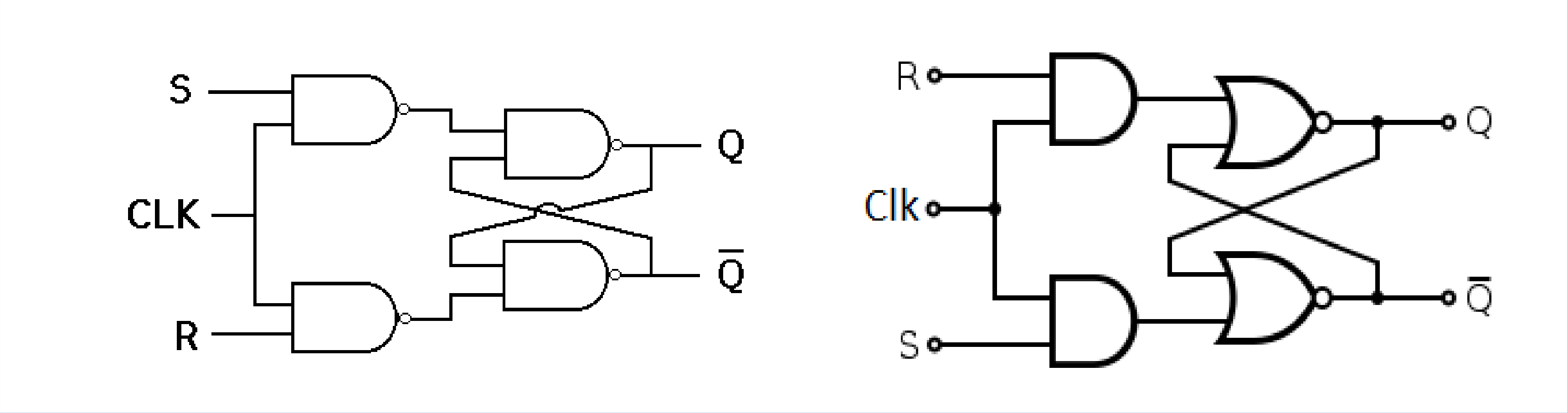
FLIP-FLOPS

Introduction

Flip-Flop: Bi-stable memory device with edge-triggering (with clock) meaning the flip-flop's output changes on the edge (rising or falling) of the CLOCK input, samples its inputs, and changes its output only at times determined by a clocking signal.

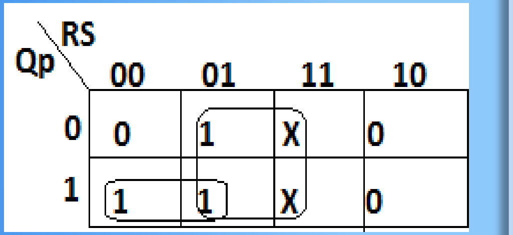
Flip-flops are a fundamental building block of digital electronics systems used in computers, communications, and many other types of systems

RS FLIP-FLOP



- S and R stand for Set and Reset
- Constructed from a pair of cross-coupled NOR logic gates OR NAND Logic gates
- With Clock high, the signals can pass through the input gates to the encapsulated latch. The latch is transparent.
- With Clock low, the latch is closed (opaque) and remains in the state it was left the last time Clock was high

RS FLIP-FLOP: CHARARCTERISTIC TABLE AND EQUATION



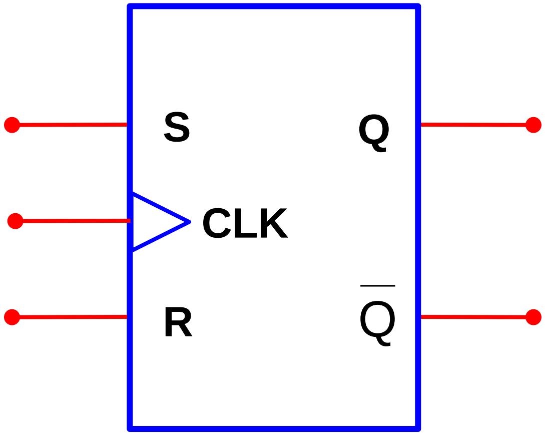
Qp \ RS	00	01	11	10
0	0	1	X	0
1	1	1	X	0

Qn=S+R’Qp

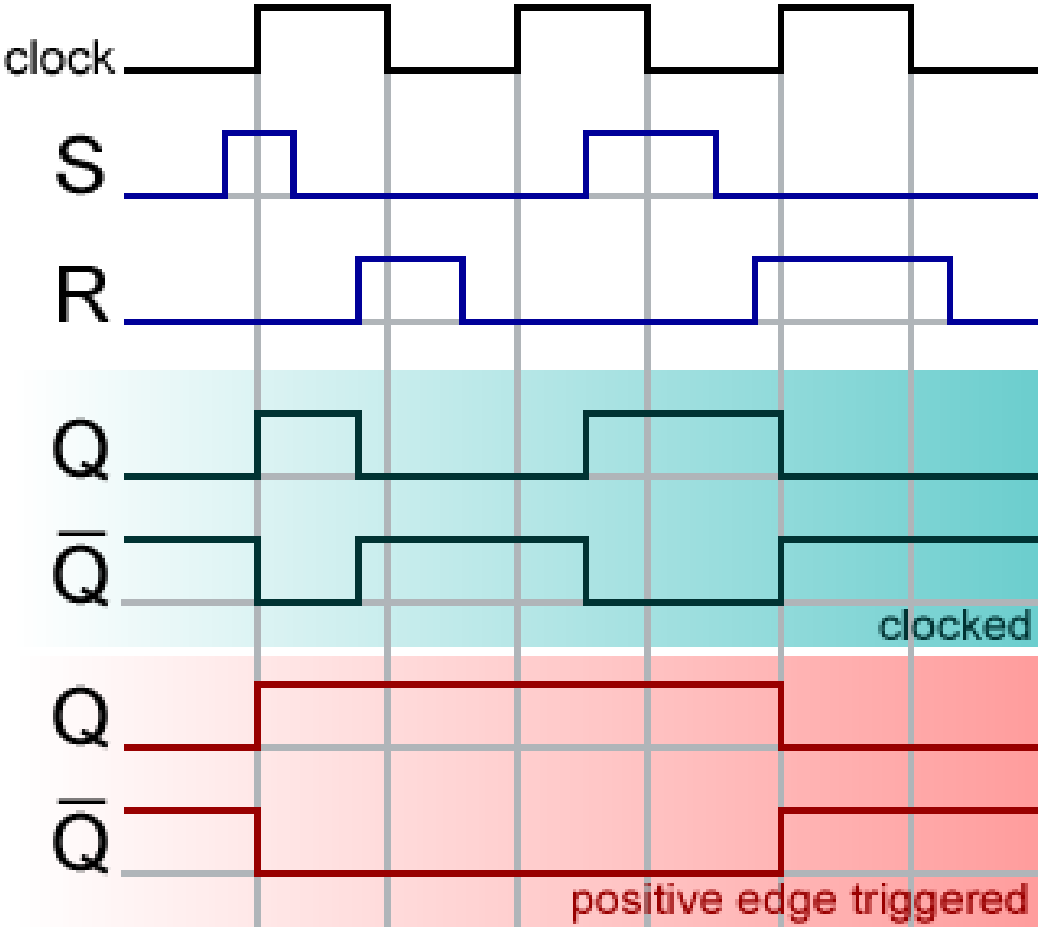
clock	R	S	Qp	Qn
O	X	X	X	No change
1	O	O	O	O
1	O	O	1	1
1	O	1	O	1
1	O	1	1	1
1	1	O	O	O
1	1	O	1	O
1	1	1	O	Race
1	1	1	1	Race

The R = S = 1 combination (in active high case) is called a restricted combination or a forbidden state because, as both NOR gates then output zeros, it breaks the logical equation Q = not Q. Race condition occurs when the flip-flop is unstable.

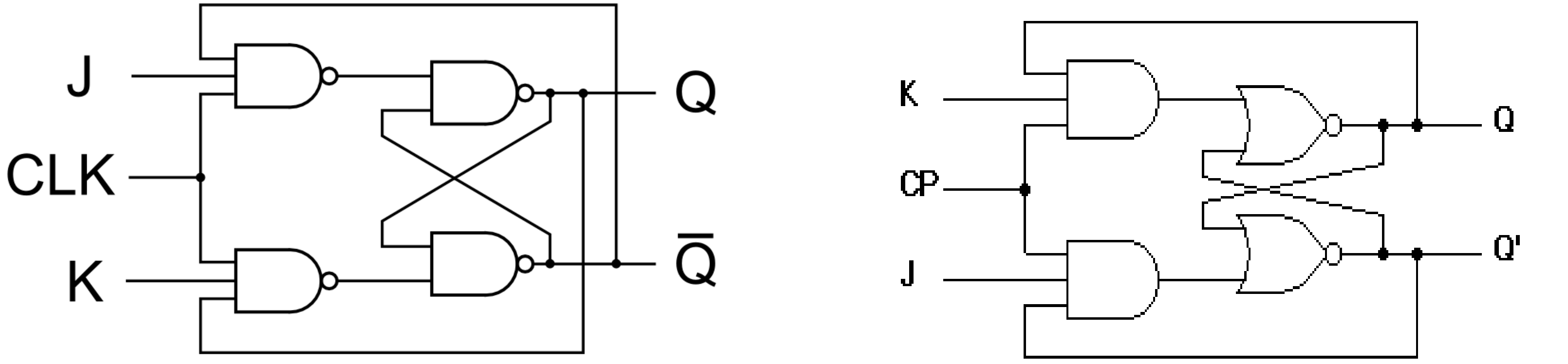
RS FLIP-FLOP: EXCITATION TABLE AND TIMING DIAGRAM



Qp	Qn	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0



JK FLIP-FLOP



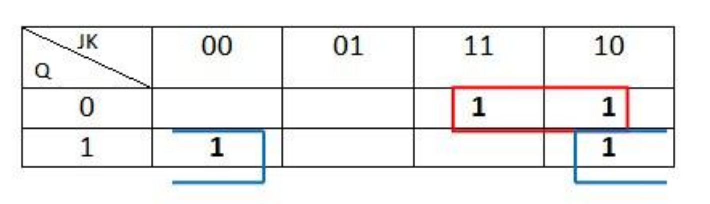
- J and K stand for Indirect Set and Indirect Reset
- Constructed from a pair of cross-coupled NOR logic gates OR NAND Logic gates
- With Clock high, the signals can pass through the input gates to the encapsulated latch. The latch is transparent.
- With Clock low, the latch is closed (opaque) and remains in the state it was left the last time Clock was high
- Modification of SR Flip-flop that removes race around condition

JK FLIP-FLOP: CHARARCTERISTIC TABLE AND EQUATION

J	K	CLK	Q	
0	0	↑	Q_0	No Change
0	1	↑	0	Clear
1	0	↑	1	Set
1	1	↑	$\overline{Q}_0$	Toggle

↑ : Rising Edge of Clock

$\overline{Q}$: Complement of Q



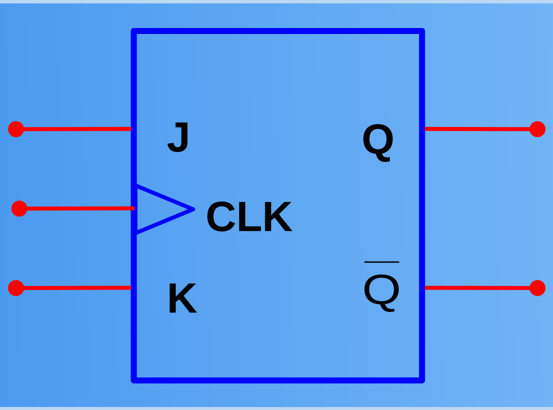
Q \ JK	00	01	11	10
0			1	1
1	1			1

$$Qn=JQ'_p+K'Q_p$$

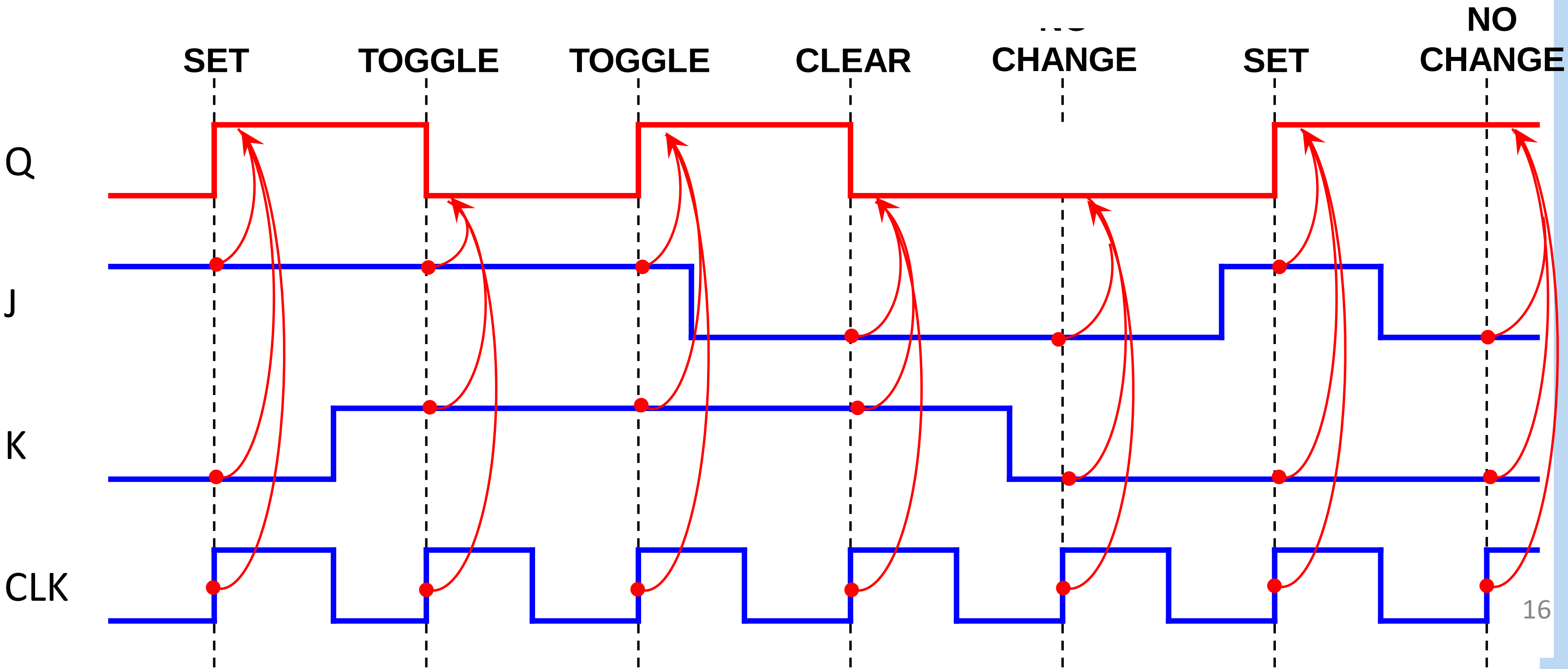
clock	J	K	Qp	Qn
O	X	X	X	No change
1	O	O	O	O
1	O	O	1	1
1	O	1	O	0
1	O	1	1	0
1	1	O	O	1
1	1	O	1	1
1	1	1	O	1
1	1	1	1	0

- The JK flip-flop augments the behavior of the SR flip-flop (J=Set, K=Reset) by interpreting the S = R = 1 condition as a "flip" or toggle command.

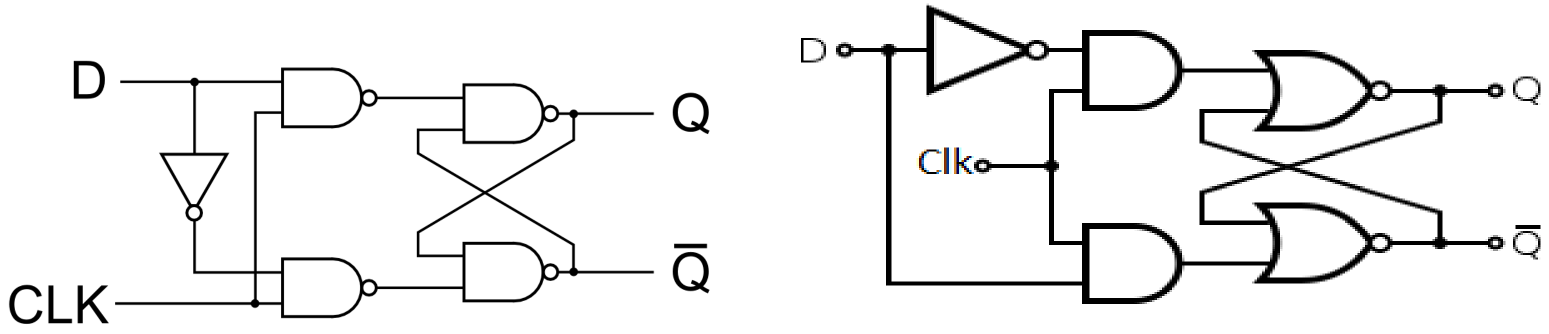
JK FLIP-FLOP: EXCITATION TABLE AND TIMING DIAGRAM



Qp	Qn	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



D FLIP-FLOP

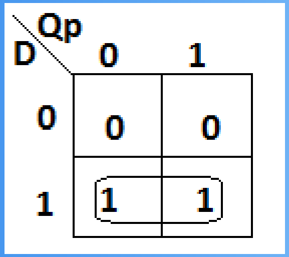


- Single input flip flop
- Also known as a data or delay flip-flop
- Can be viewed as a memory cell
- They form the basis for shift registers, which are an essential part of many electronic devices
- Constructed from a pair of cross-coupled NOR logic gates OR NAND Logic gates
- With Clock high, the signals can pass through the input gates to the encapsulated latch. The latch is transparent.
- With Clock low, the latch is closed (opaque) and remains in the state it was left the last time Clock was high
- Modification of SR Flip-flop that removes race around condition

D FLIP-FLOP: CHARARCTERISTIC TABLE AND EQUATION

D	CLK	Q	$\overline{Q}$
0	↑	0	1
1	↑	1	0

↑ : Rising Edge of Clock

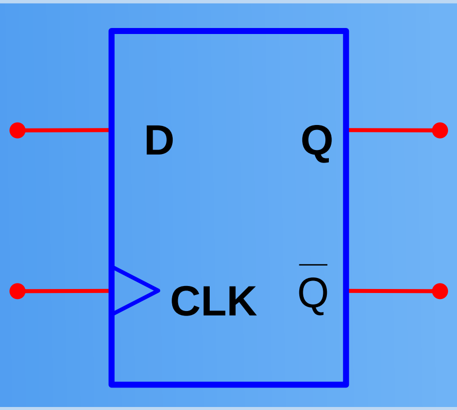


D \ Qp	0	1
0	0	0
1	1	1

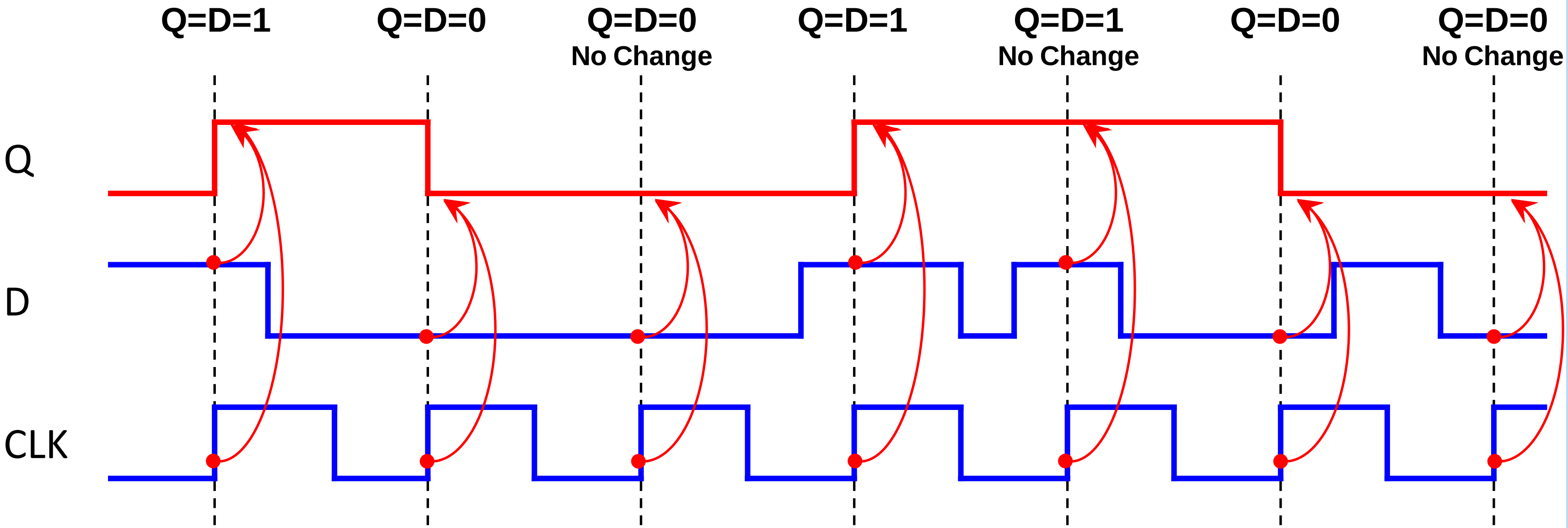
Qn=D

clock	D	Qp	Qn
O	X	X	No change
1	O	O	O
1	O	1	O
1	1	O	1
1	1	1	1

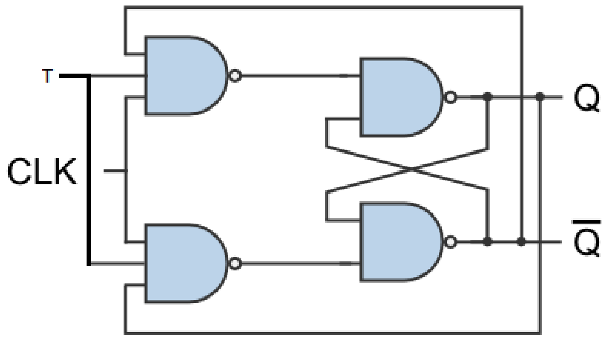
D FLIP-FLOP: EXCITATION TABLE AND TIMING DIAGRAM

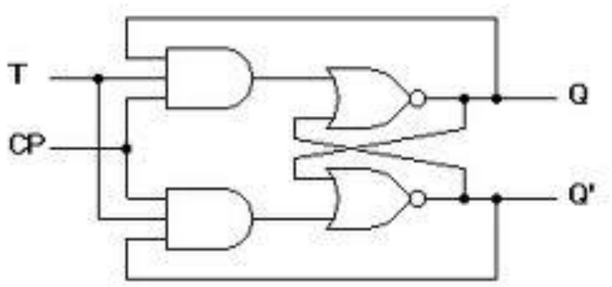


Qp	Qn	D
0	0	0
0	1	1
1	0	0
1	1	1



T FLIP-FLOP



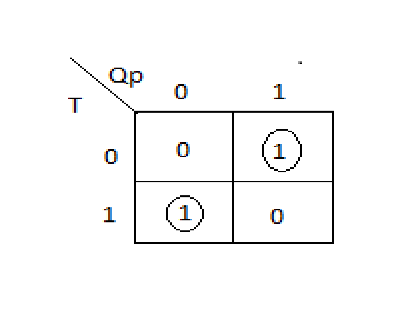


- Single input flip flop
- Also known as a Toggle or Triggered flip-flop
- They form the basis for counters
- Constructed from a pair of cross-coupled NOR logic gates OR NAND Logic gates
- With Clock high, the signals can pass through the input gates to the encapsulated latch. The latch is transparent.
- With Clock low, the latch is closed (opaque) and remains in the state it was left the last time Clock was high
- Modification of JK Flip-flop

T FLIP-FLOP: CHARARCTERISTIC TABLE AND EQUATION

T	CLK	Q	$\overline{Q}$
0	↑	1	0
1	↑	0	1

↑ : Rising Edge of Clock



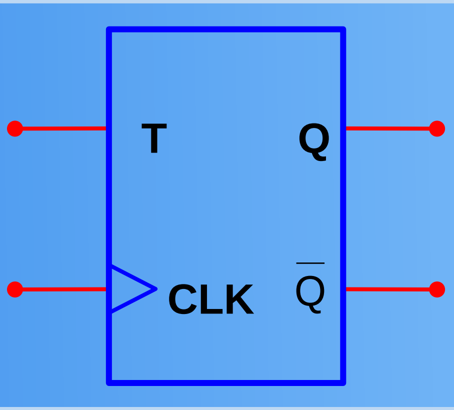
T \ Qp	0	1
0	0	1
1	1	0

$Qn=TQ'_p+T'Q_p$

$Qn=T \text{ XOR } Q_p$

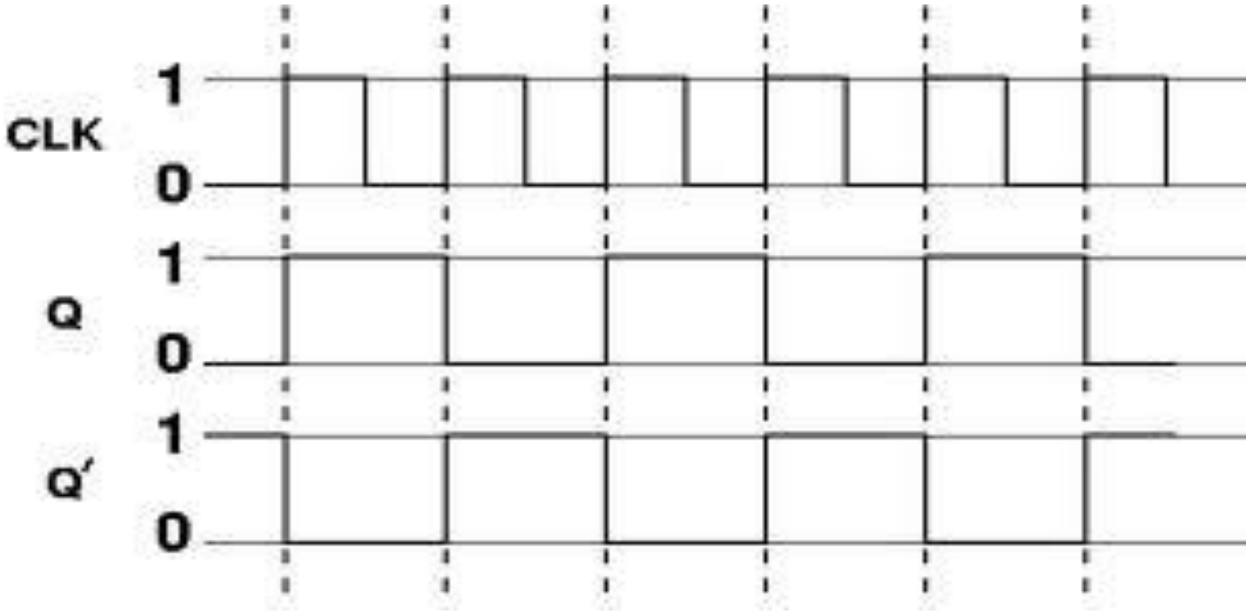
clock	T	Qp	Qn
O	X	X	No change
1	O	O	O
1	O	1	1
1	1	O	1
1	1	1	O

T FLIP-FLOP: EXCITATION TABLE AND TIMING DIAGRAM



Qp	Qn	T
0	0	0
0	1	1
1	0	1
1	1	0

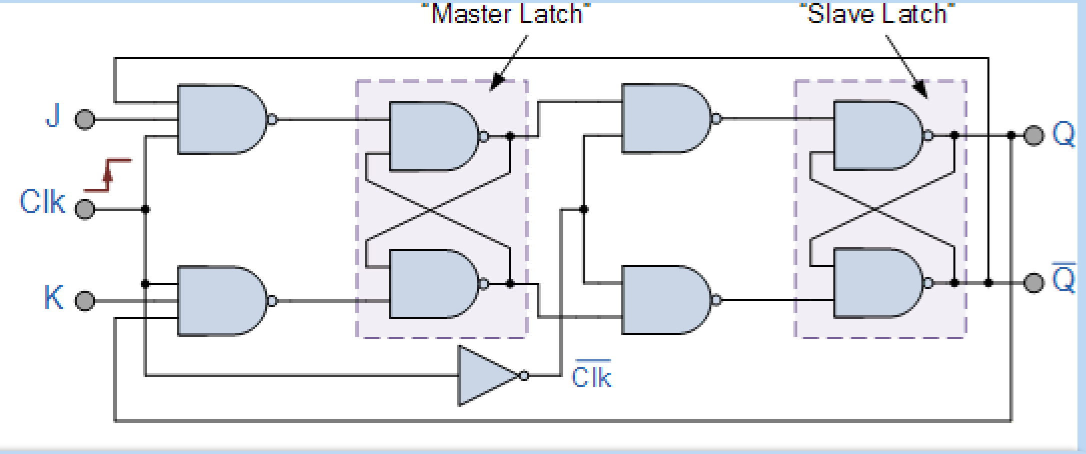
T=1 and clock is in positive edge

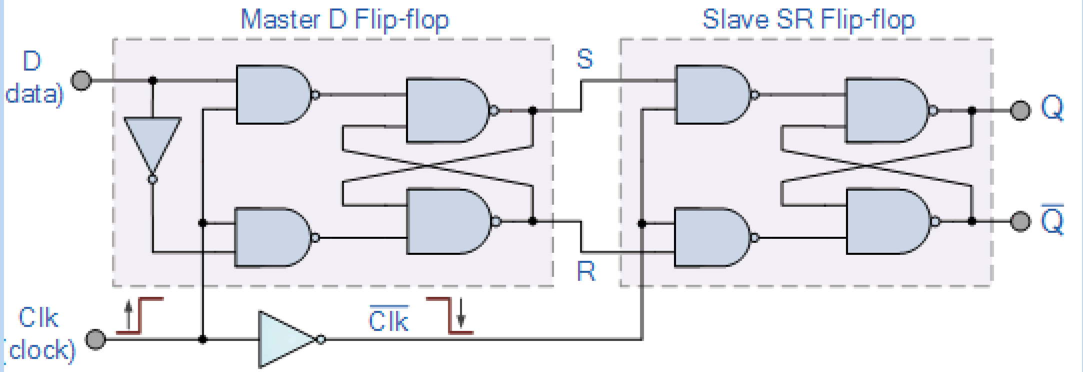


Master Slave Flip-Flop

- ➢ Figure alongside shows Master Slave JK Flipflop and Master Slave D Flipflop
- ➢ Also known as Pulse Triggering flipflop
- ➢ The circuit accepts input data when the clock signal is "HIGH", and passes the data to the output on the falling-edge of the clock signal.

.





Basically two gated D flip-flops connected together in a series configuration with the slave having an inverted clock pulse.

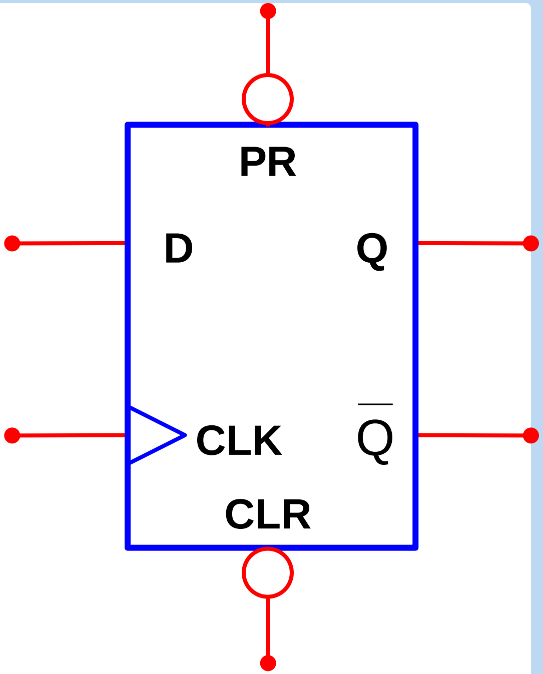
ASYNCHRONOUS INPUTS OF A FLIP-FLOP

- Asynchronous inputs (Preset & Clear) are used to override the clock/data inputs and force the outputs to a predefined state.
- The Preset (PR) input forces the output to:

$$Q = 1 \ \& \ \overline{Q} = 0$$

- The Clear (CLR) input forces the output to:

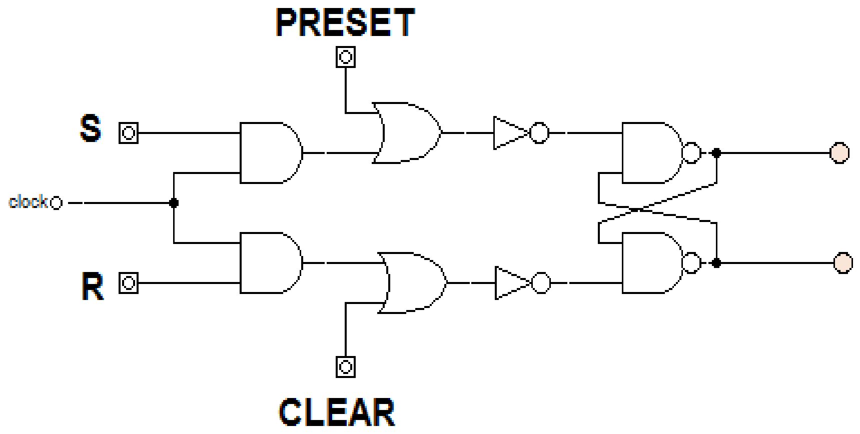
$$Q = 0 \ \& \ \overline{Q} = 1$$

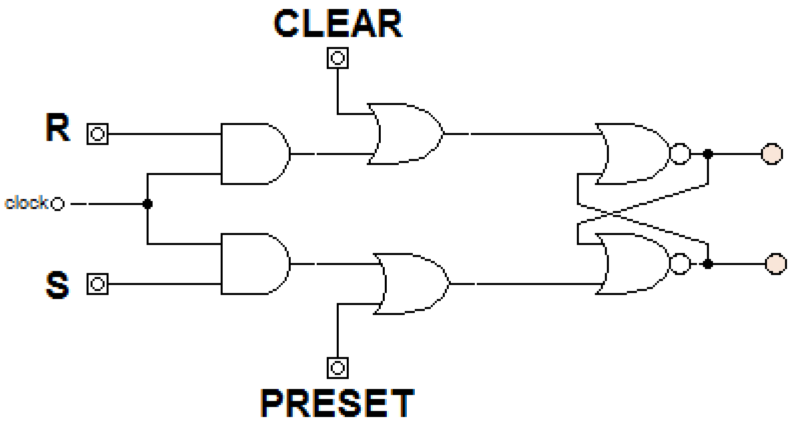


The table below is for an active low consideration

PR PRESET	CLR CLEAR	CLK CLOCK	D DATA	Q	$\overline{Q}$	
1	1	↑	0	0	1	
1	1	↑	1	1	0	
0	1	X	X	1	0	Asynchronous Preset
1	0	X	X	0	1	Asynchronous Clear
0	0	X	X	1	1	ILLEGAL CONDITION

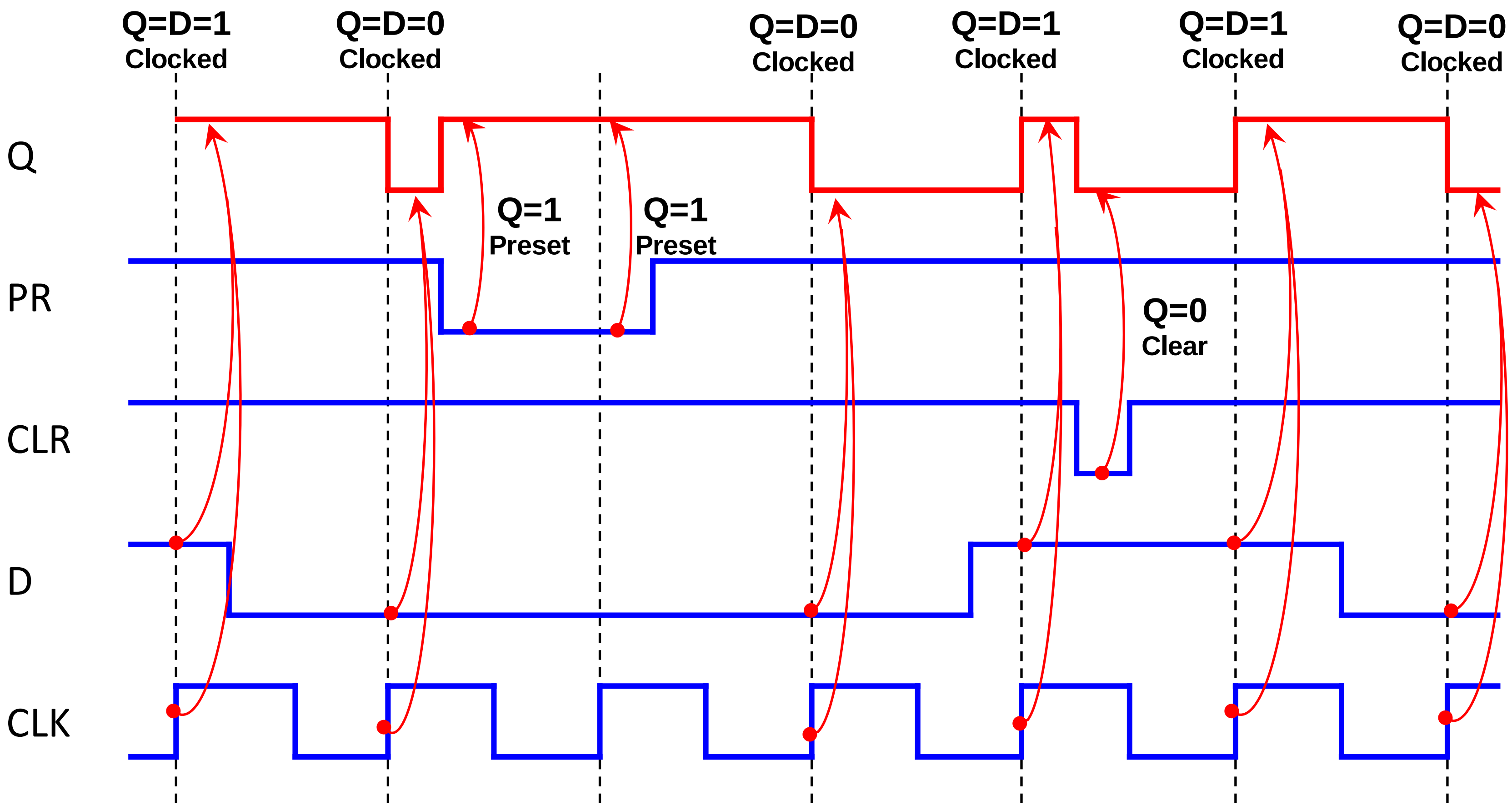
ASYNCHRONOUS INPUTS OF A FLIP-FLOP



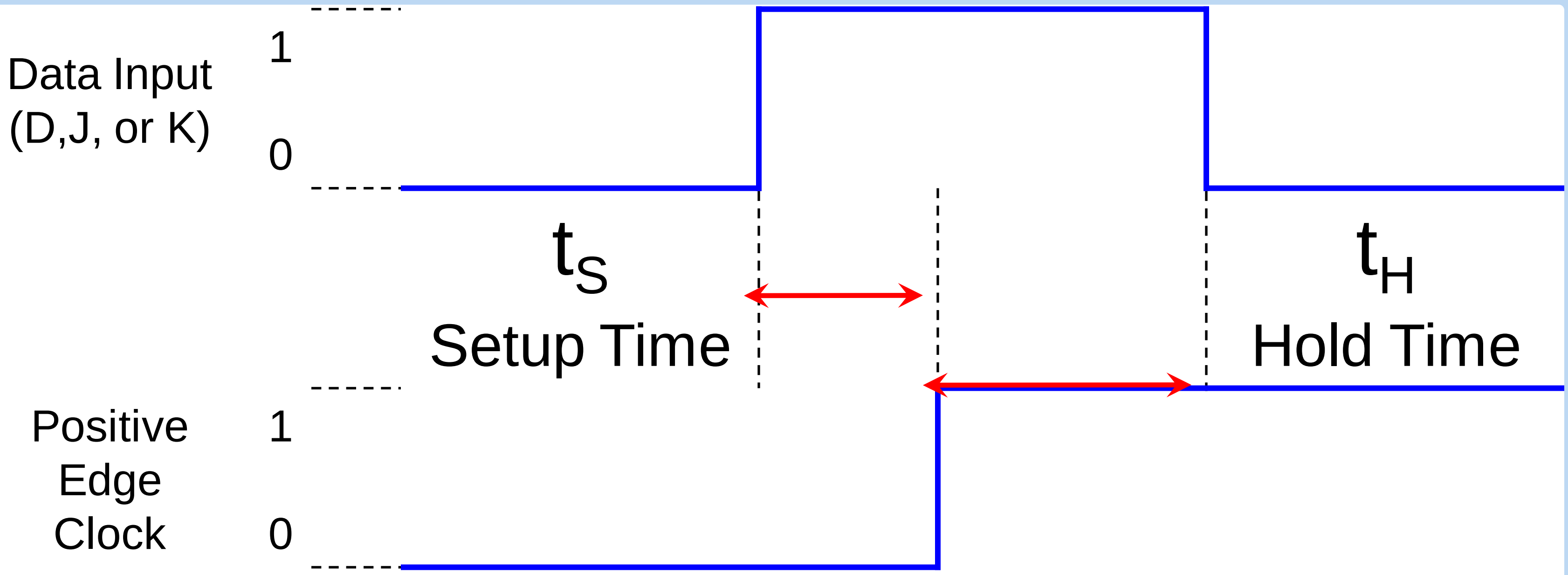


- Flip-flops are subject to a problem called metastability, which can happen when two inputs, such as data and clock or clock and reset, are changing at about the same time.
- When the order is not clear, within appropriate timing constraints, the result is that the output may behave unpredictably, taking many times longer than normal to settle to one state or the other, or even oscillating several times before settling.
- In a computer system, this metastability can cause corruption of data or a program crash, if the state is not stable before another circuit uses its value; in particular, if two different logical paths use the output of a flip-flop, one path can interpret it as a 0 and the other as a 1 when it has not resolved to stable state, putting the machine into an inconsistent state

D FLIP-FLOP: PR & CLR TIMING (Active low consideration)



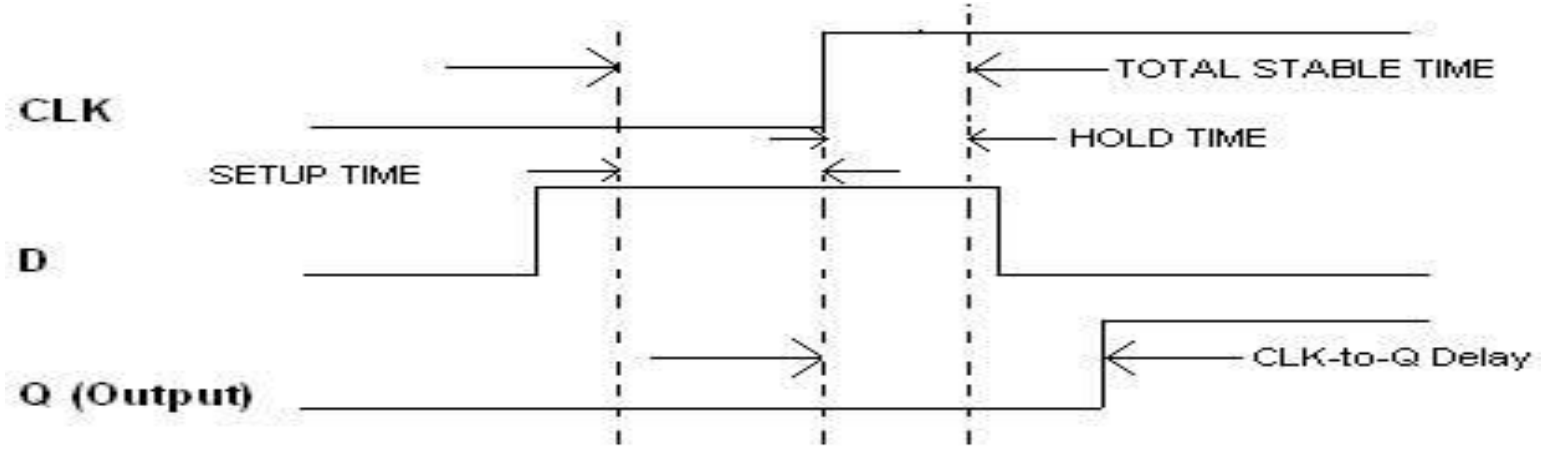
PARAMETERS OF A FLIP-FLOP (Operating Characteristics)



- **Setup Time (t_S):** The time interval before the active transition of the clock signal during which the data input (D, J, or K) must be maintained.

- **Hold Time (t_H):** The time interval after the active transition of the clock signal during which the data input (D, J, or K) must be maintained.

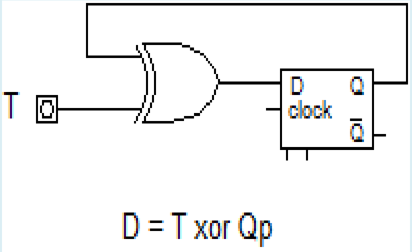
PARAMETERS OF A FLIP-FLOP (Operating Characteristics)



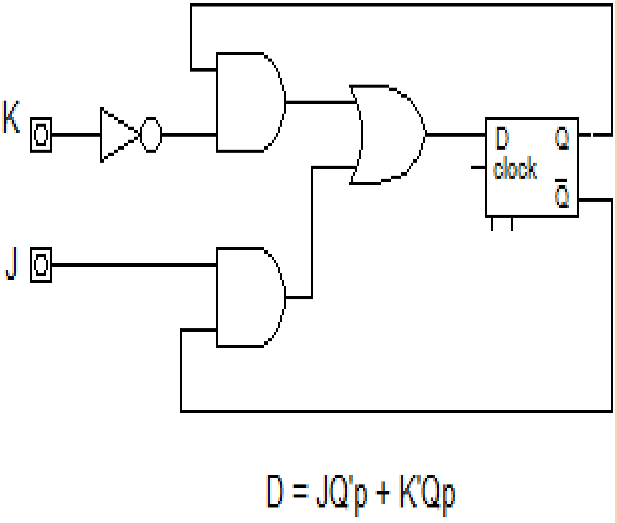
- **PROPAGATION TIME:** It is defined as the time after the clock transition, required for a flip-flop to generate output. This is also called CLOCK TO Q delay($T_{clock\text{-}to\text{-}q}$).

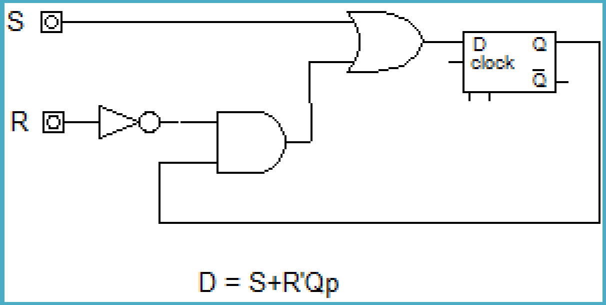
OTHER FLIP-FLOPS FROM D FLIP-FLOP

D to T Flip-Flop



D to JK Flip-Flop

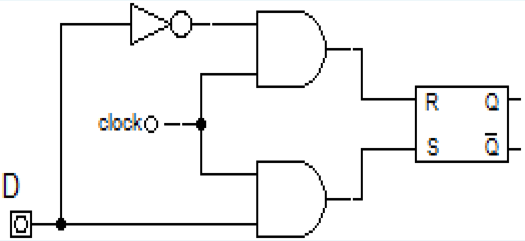




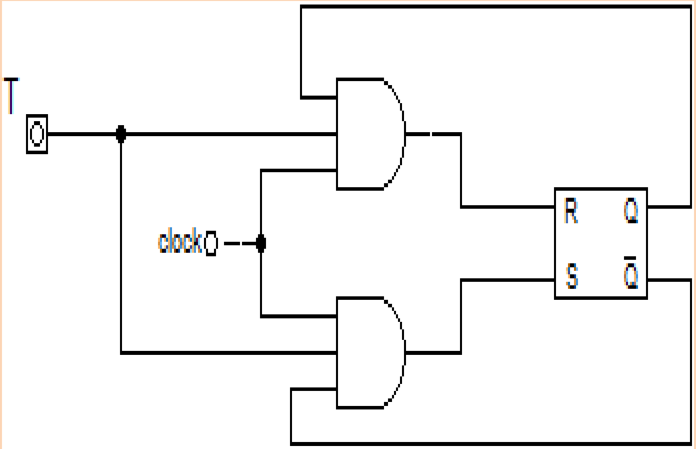
D to SR Flip-Flop

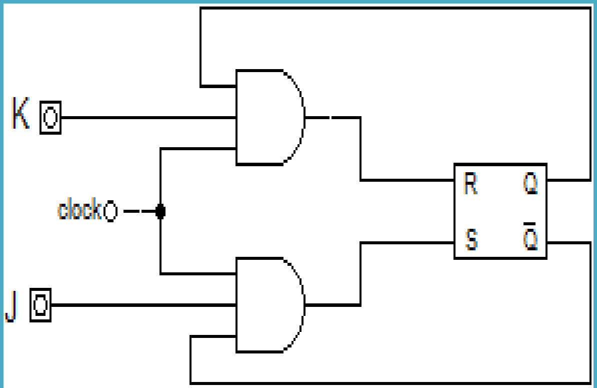
OTHER FLIP-FLOPS FROM SR FLIP-FLOP

SR to D Flip-Flop



SR to T Flip-Flop

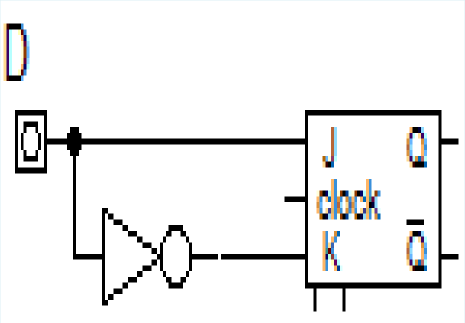




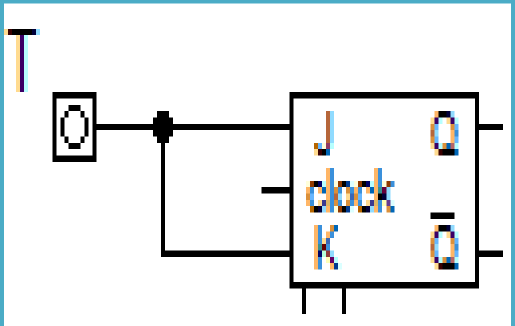
SR to JK Flip-Flop

OTHER FLIP-FLOPS FROM JK FLIP-FLOP

JK to D Flip-Flop



JK to T Flip-Flop



OTHER FLIP-FLOPS FROM JK FLIP-FLOP

JK to SR Flip-Flop

Given is JK (Excitation Table)

Qp	Qn	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Required is SR(Characteristic Table)

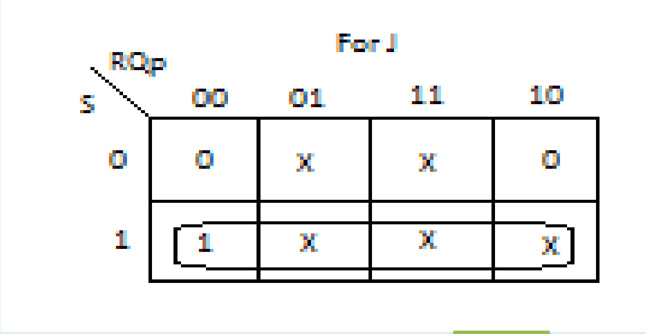
S	R	Qp	Qn
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	RACE
1	1	1	RACE

S	R	Qp	Qn	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	RACE	X	X
1	1	1	RACE	X	X

Diagram is in next slide

OTHER FLIP-FLOPS FROM JK FLIP-FLOP

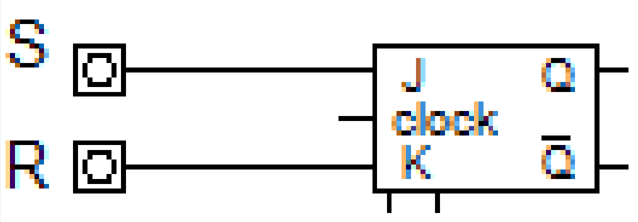
JK to SR Flip-Flop

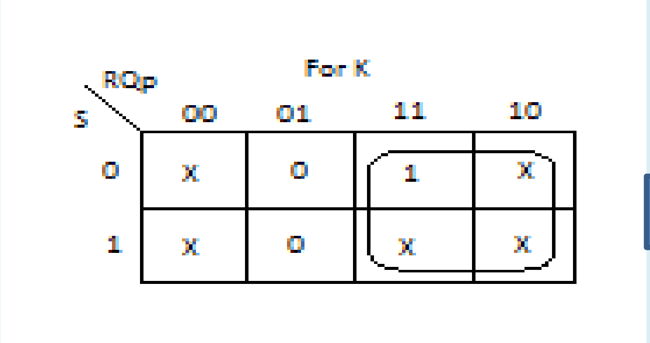


For J

RQp S	00	01	11	10
0	0	X	X	0
1	1	X	X	X

J=S





For K

RQp S	00	01	11	10
0	X	0	1	X
1	X	0	X	X

K=R

OTHER FLIP-FLOPS FROM T FLIP-FLOP

T to D Flip-Flop

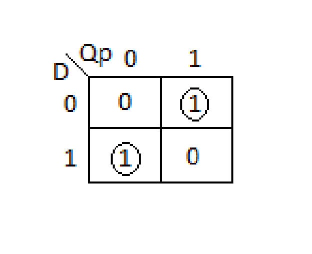
Required is D (Characteristic Table)

D	Qp	Qn
0	0	0
0	1	0
1	0	1
1	1	1

Given is T (Excitation Table)

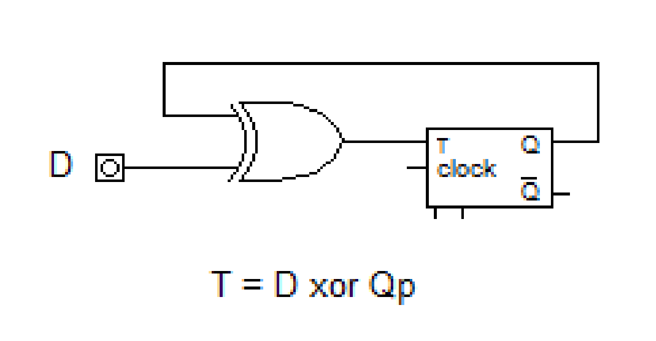
Qp	Qn	T
0	0	0
0	1	1
1	0	1
1	1	0

D	Qp	Qn	T
0	0	0	O
0	1	0	1
1	0	1	1
1	1	1	0



D \ Qp	0	1
0	0	1
1	1	0

→ T = D XOR Qp



T = D xor Qp

OTHER FLIP-FLOPS FROM T FLIP-FLOP

T to JK Flip-Flop

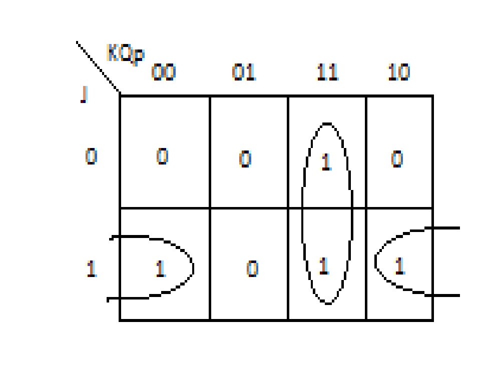
Required is JK(Characteristic Table)

J	K	Qp	Qn
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Qp	Qn	T
0	0	0
0	1	1
1	0	1
1	1	0

Given is T (Excitation Table)

J	K	Qp	Qn	T
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	1



J \ KQp	00	01	11	10
0	0	0	1	0
1	1	0	1	1

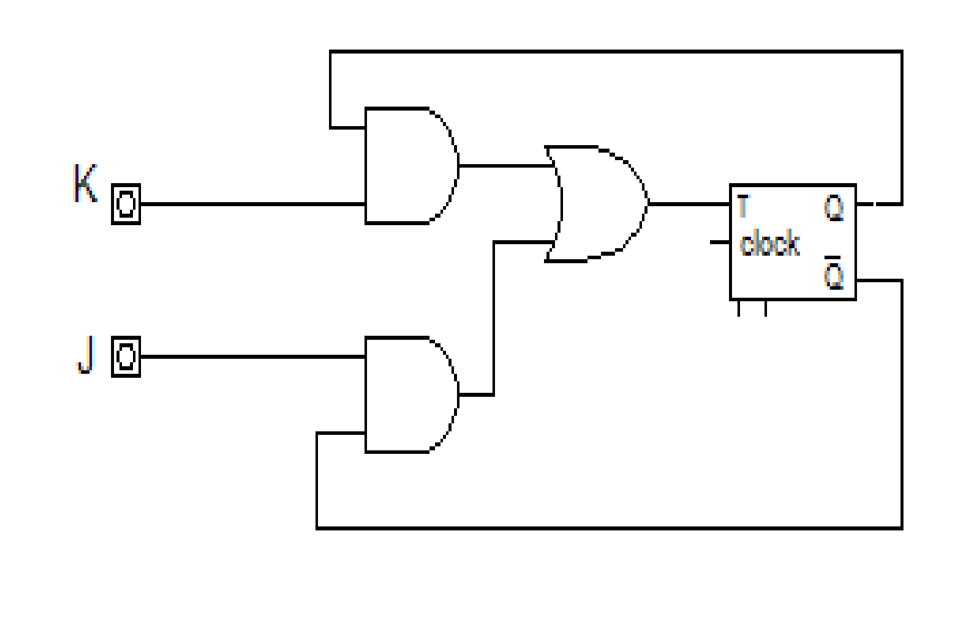
$$T=JQ'_p+KQ_p$$

Diagram is in next slide

OTHER FLIP-FLOPS FROM T FLIP-FLOP

T to JK Flip-Flop

$T = JQ'_p + KQ_p$



OTHER FLIP-FLOPS FROM T FLIP-FLOP

T to SR Flip-Flop

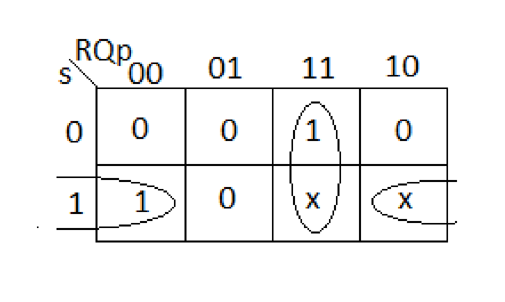
Required is SR(Characteristic Table)

S	R	Qp	Qn
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	RACE
1	1	1	RACE

Given is T (Excitation Table)

Qp	Qn	T
0	0	0
0	1	1
1	0	1
1	1	0

S	R	Qp	Qn	T
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	RACE	X
1	1	1	RACE	X



S \ RQp	00	01	11	10
0	0	0	1	0
1	1	0	x	x

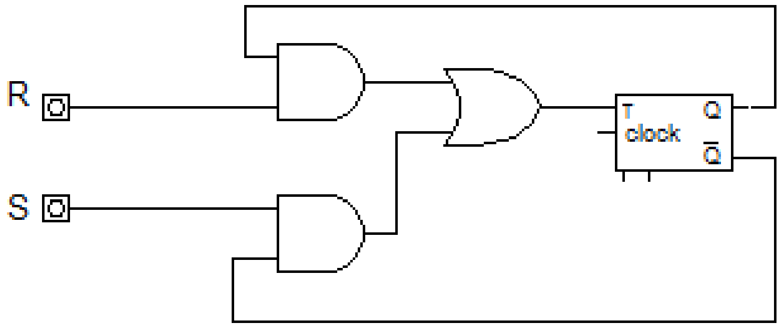
$T=SQ'_p+RQ_p$

Diagram is in next slide

OTHER FLIP-FLOPS FROM T FLIP-FLOP

T to SR Flip-Flop

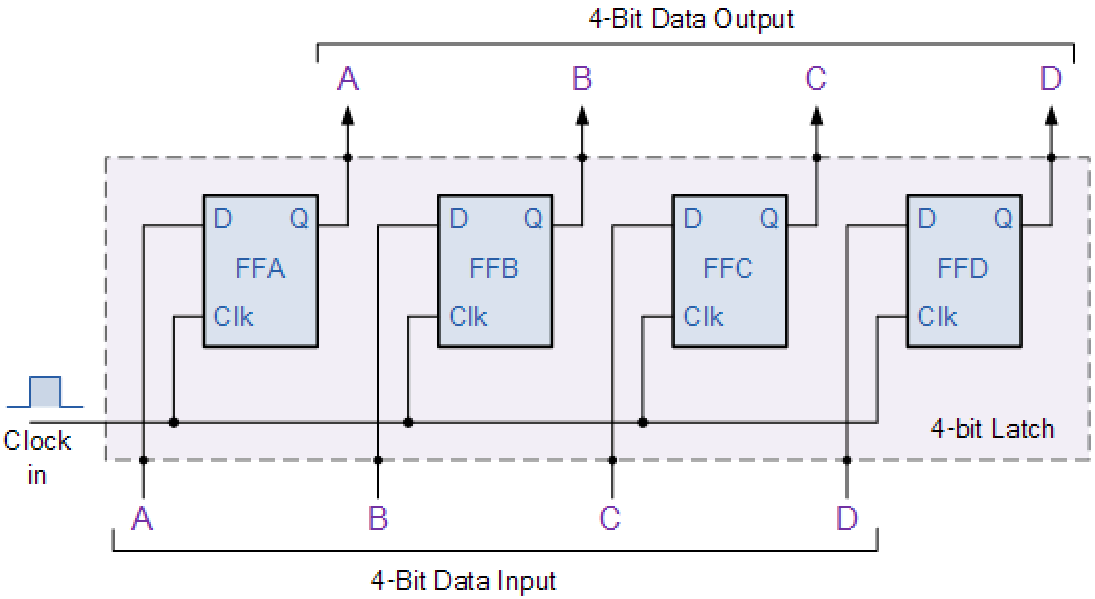
$T=SQ'_p+RQ_p$



T = SQ'p + RQp

BASIC FLIP-FLOP APPLICATIONS

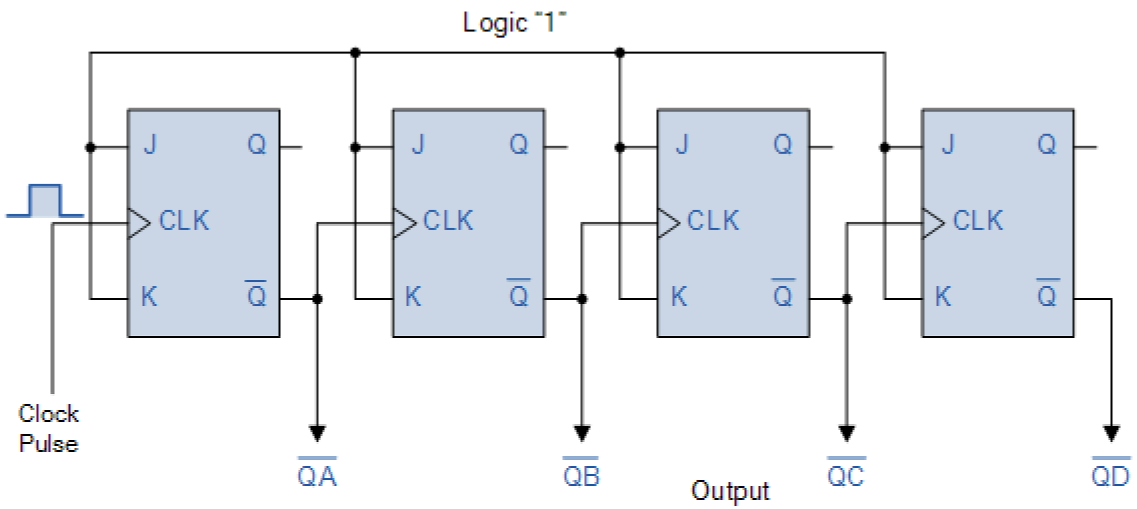
1. Registers



Data storage and movement serially or in parallel way

BASIC FLIP-FLOP APPLICATIONS

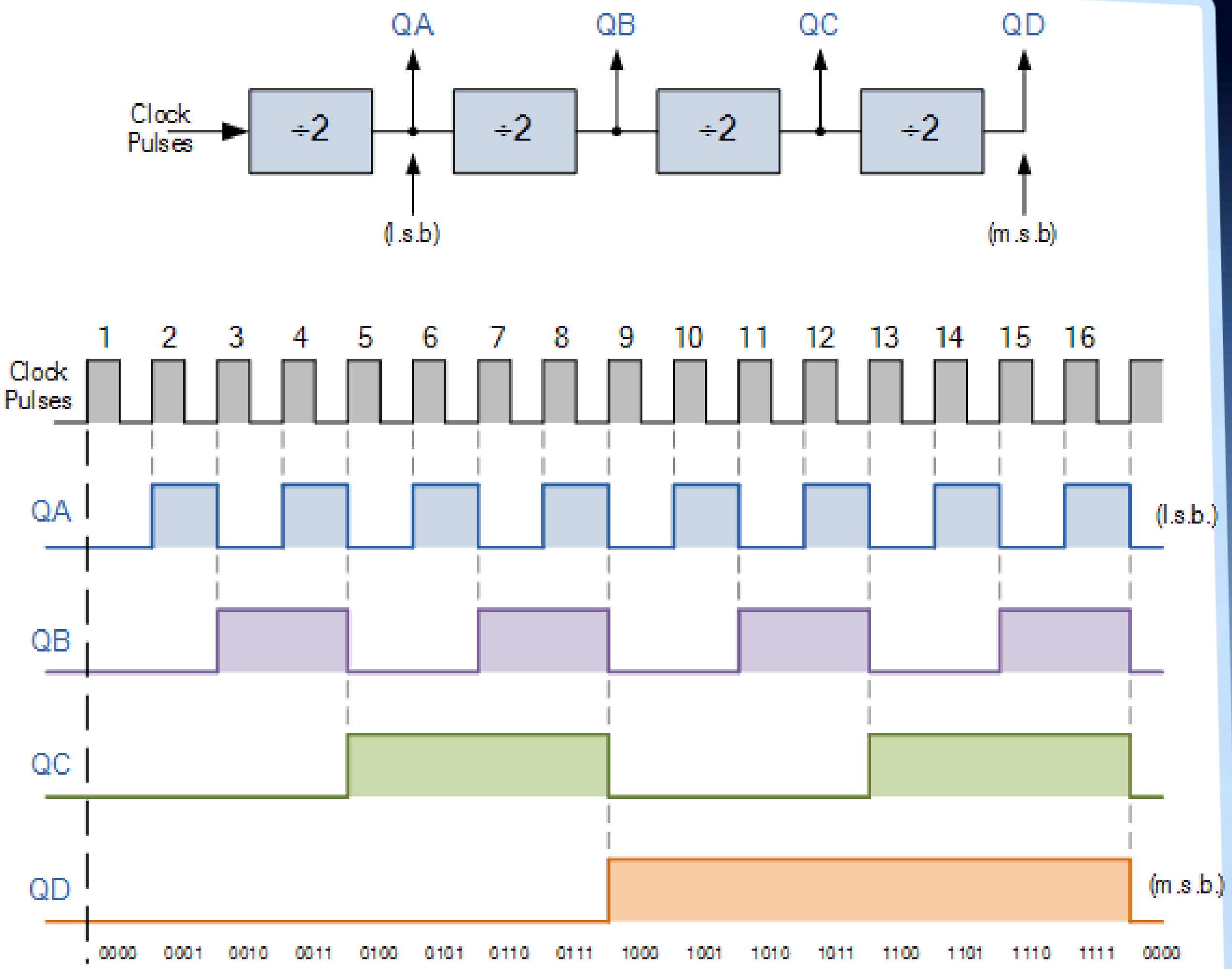
2. Counting



Asynchronous down counting from 15 to 0

BASIC FLIP-FLOP APPLICATIONS

3. Frequency Division



BASIC FLIP-FLOP APPLICATIONS

3. Frequency Division

1 Flip-flop- Divide the clock frequency by 2
2 Flip-flop- Divide the clock frequency by 4
3 Flip-flop- Divide the clock frequency by 8
4 Flip-flop- Divide the clock frequency by 16
5 Flip-flop- Divide the clock frequency by 32
6 Flip-flop- Divide the clock frequency by 64
7 Flip-flop- Divide the clock frequency by 128

↓

n Flip-flop- Divide the clock frequency by 2^n