Digital Electronics & Logic Design

(EC 207)

Dr. Vivek Garg

Department of Electronics Engineering

S. V. National Institute of Technology (SVNIT)
Surat

email: vivekg@eced.svnit.ac.in; vivekgarg0101@gmail.com

Course Outline

- **PN DIODE AND TRANSITOR** **(04 Hours)**

  PN Diode Theory, PN Characteristic and Breakdown Region, PN Diode Application as Rectifier, Zener Diode Theory, Zener Voltage Regulator, Diode as Clamper and Clipper, Photodiode Theory, LED Theory, 7 Segment LED Circuit Diagram and Multi Colour LED, LASER Diode Theory and Applications, Bipolar Junction Transistor Theory, Transistor Symbols And Terminals, Common Collector, Emitter and Base Configurations, Different Biasing Techniques, Concept of Transistor Amplifier, Introduction to FET Transistor And Its Feature.

- **WAVESHAPING CIRCUITS AND OPERATIONAL AMPLIFIER** **(06 Hours)**

  Linear Wave Shaping Circuits, RC High Pass and Low Pass Circuits, RC Integrator and Differentiator Circuits, Nonlinear Wave Shaping Circuits, Two Level Diode Clipper Circuits, Clamping Circuits, Operational Amplifier OP-AMP with Block Diagram, Schematic Symbol of OP-AMP, The 741 Package Style and Pinouts, Specifications of Op-Amp, Inverting and Non-Inverting Amplifier, Voltage Follower Circuit, Multistage OP-AMP Circuit, OP-AMP Averaging Amplifier, OP-AMP Subtractor.

- **BOOLEAN ALGEBRA AND SWITCHING FUNCTIONS** **(04 Hours)**

  Basic Logic Operation and Logic Gates, Truth Table, Basic Postulates and Fundamental Theorems of Boolean Algebra, Standard Representations of Logic Functions- SOP and POS Forms, Simplification of Switching Functions-K-Map and Quine-Mccluskey Tabular Methods, Synthesis of Combinational Logic Circuits.

- **COMBINATIONAL LOGIC CIRCUIT USING MSI INTEGRATED CIRCUITS** **(07 Hours)**

Course Outline

Binary Parallel Adder; BCD Adder; Encoder, Priority Encoder, Decoder; Multiplexer and Demultiplexer Circuits; Implementation of Boolean Functions Using Decoder and Multiplexer; Arithmetic and Logic Unit; BCD to 7-Segment Decoder; Common Anode and Common Cathode 7-Segment Displays; Random Access Memory, Read Only Memory And Erasable Programmable ROMS; Programmable Logic Array (PLA) and Programmable Array Logic (PAL).

- **INTRODUCTION TO SEQUENTIAL LOGIC CIRCUITS** **(04 Hours)**

  Basic Concepts of Sequential Circuits; Cross Coupled SR Flip-Flop Using NAND or NOR Gates; JK Flip-Flop Rise Condition; Clocked Flip-Flop; D-Type and Toggle Flip-Flops; Truth Tables and Excitation Tables for Flip-Flops; Master Slave Configuration; Edge Triggered and Level Triggered Flip-Flops; Elimination of Switch Bounce using Flip-Flops; Flip-Flops with Preset and Clear.

- **SEQUENTIAL LOGIC CIRCUIT DESIGN** **(06 Hours)**

  Basic Concepts of Counters and Registers; Binary Counters; BCD Counters; Up Down Counter; Johnson Counter, Module-N Counter; Design of Counter Using State Diagrams and Table; Sequence Generators; Shift Left and Right Register; Registers With Parallel Load; Serial-In-Parallel-Out (SIPO) And Parallel-In-Serial-Out(PISO); Register using Different Type of Flip-Flop.

- **REGISTER TRANSFER LOGIC** **(04 Hours)**

  Arithmetic, Logic and Shift Micro-Operation; Conditional Control Statements; Fixed-Point and Floating-Point Data; Arithmetic Shifts; Instruction Code and Design Of Simple Computer.

- **PROCESSOR LOGIC DESIGN** **(03 Hours)**

  Processor Organization; Design of Arithmetic Logic Unit; Design of Accumulator.

- **CONTROL LOGIC DESIGN** **(04 Hours)**

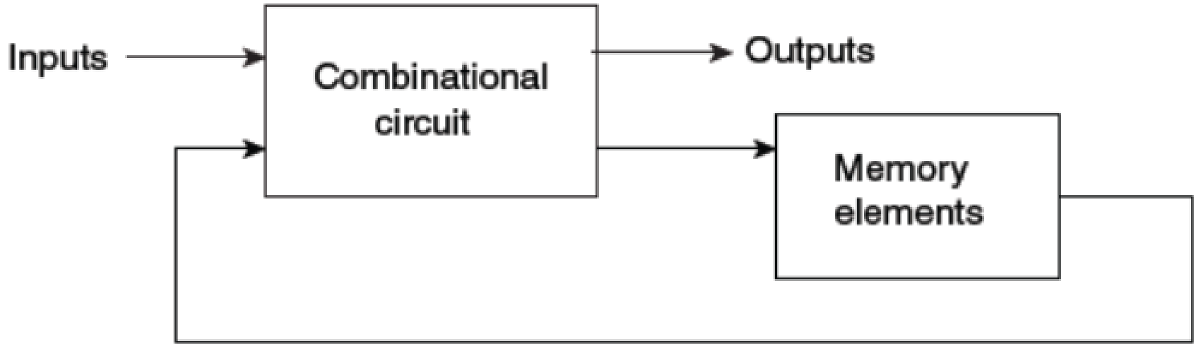
  Control Organization; Hard-Wired Control; Micro Program Control; Control Of Processor Unit; PLA Control.

Course Text and Materials

1. Schilling Donald L. and Belove E., "Electronics Circuits- Discrete and Integrated", 3rd Ed., McGraw-Hill, 1989, Reprint 2008.
2. Millman Jacob, Halkias Christos C. and Parikh C., "Integrated Electronics", 2nd Ed., McGraw-Hill, 2009.
3. Taub H. and Mothibi Suryaprakash, Millman J., "Pulse, Digital and Switching Waveforms", 2nd Ed., McGraw-Hill, 2007.
4. Mano Morris, “Digital Logic and Computer Design”, 5th Ed., Pearson Education, 2005.
5. Lee Samual, “Digital Circuits and Logic Design”, 1st Ed., PHI, 1998.

Digital Logic Circuits

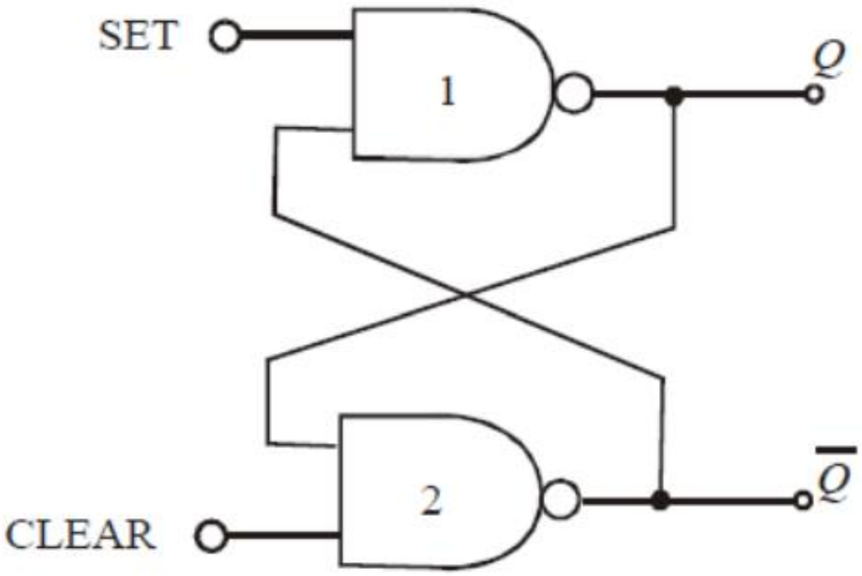
COMBINATIONAL CIRCUITS	SEQUENTIAL CIRCUITS
Output depends only on the present value of the inputs.	Output depends on both the present and previous state values of the inputs
These circuits will not have any memory as their outputs change with the change in the input value.	Sequential circuits have some sort of memory as their output changes according to the previous and present values.
There are no feedbacks involved.	In a sequential circuit the outputs are connected to it as a feedback path.
Used in basic Boolean operations.	Used in the designing of memory devices.
Implemented in: Half adder circuit, full adder circuit, multiplexers, de-multiplexers, decoders and encoders.	Implemented in: RAM, Registers, counters and other state retaining machines.

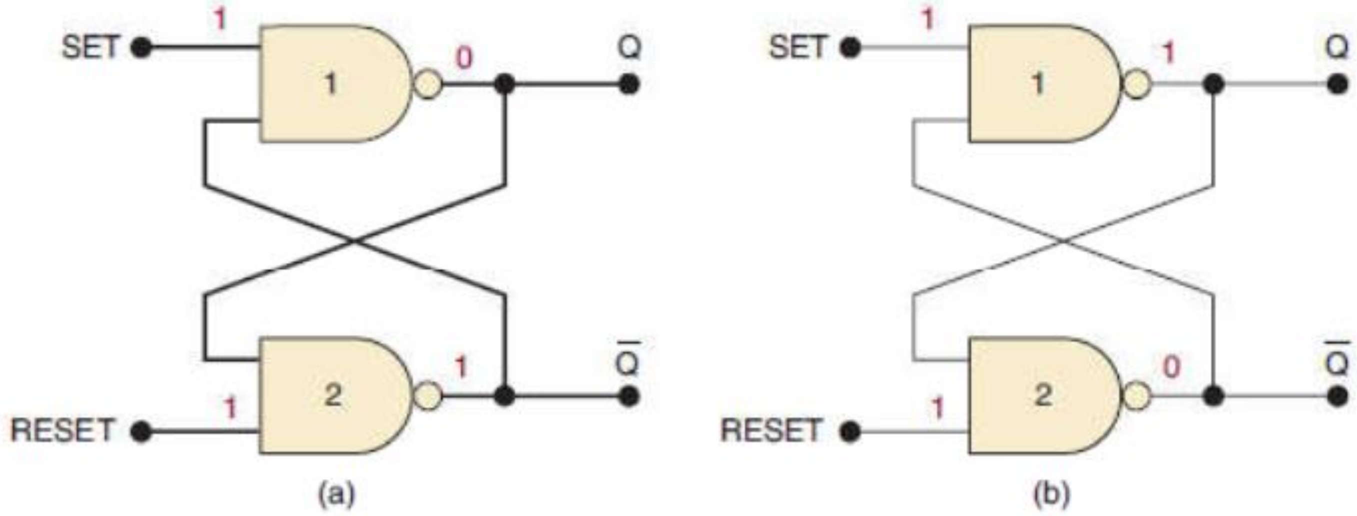


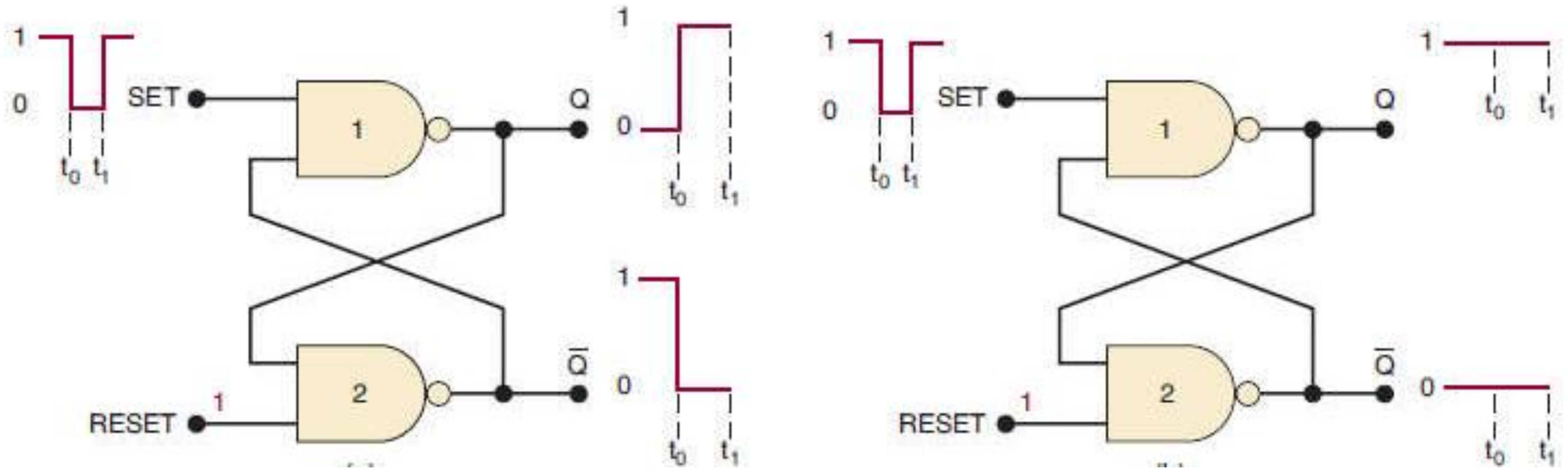
- Basic Memory element of a Digital Computer
- Stores 1 bit of information
- It’s a Bistable device
- Has two outputs, one complement of another (Q and Q’)
- Four Types
  - SR
  - D
  - JK
  - T

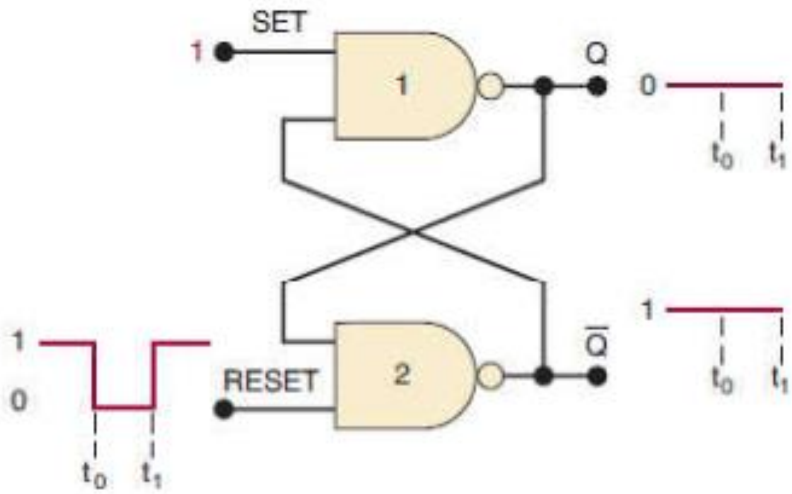
- Most basic type of FF circuit
- Can be constructed using NAND or NOR Gates
- These circuits latch to '1' or '0' immediately upon application of inputs
- Two types
  - NAND Gate Latch (Active Low)
  - NOR Gate Latch (Active High)

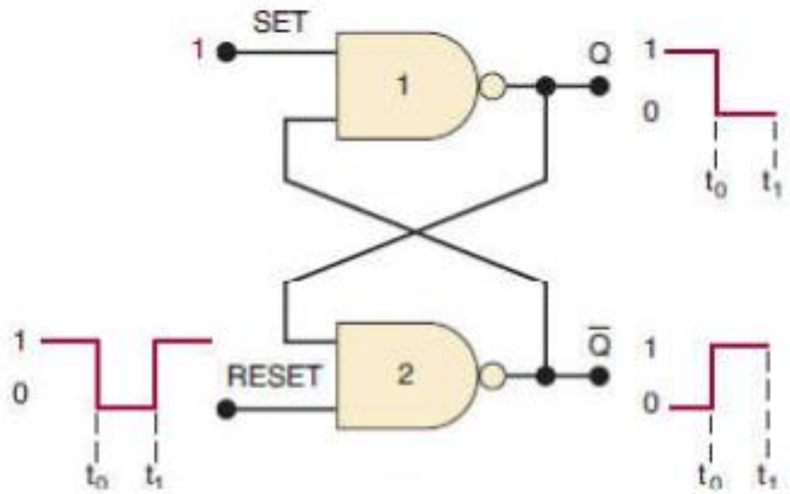
NAND Gate SR Latch



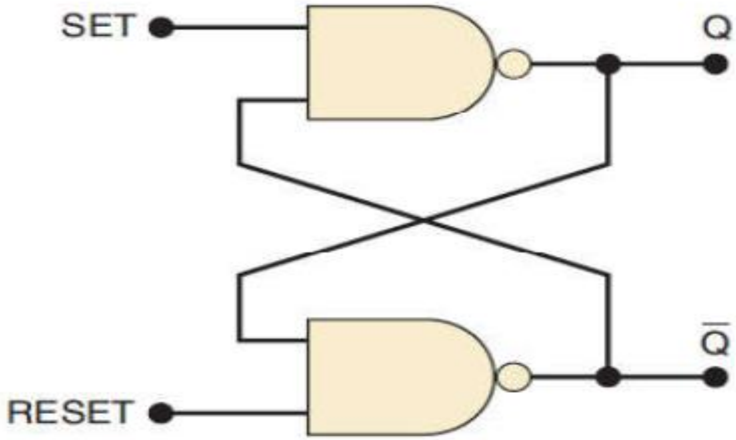

SET
1
1
0
Q
RESET
1
2
1
Q̄
(a)
SET
1
1
1
Q
RESET
1
2
0
Q̄
(b)


1
0
SET
t_0 t_1
1
Q
0
1
2
$\overline{Q}$
RESET
1
0
SET
Q
1
2
$\overline{Q}$
0
RESET


SET
1
1
Q
0
t_0 t_1
1
2
RESET
$\overline{Q}$
1
0
t_0 t_1


SET
1
1
Q
1
0
t_0 t_1
1
0
2
RESET
$\overline{Q}$
1
0
t_0 t_1

NAND Gate SR Latch



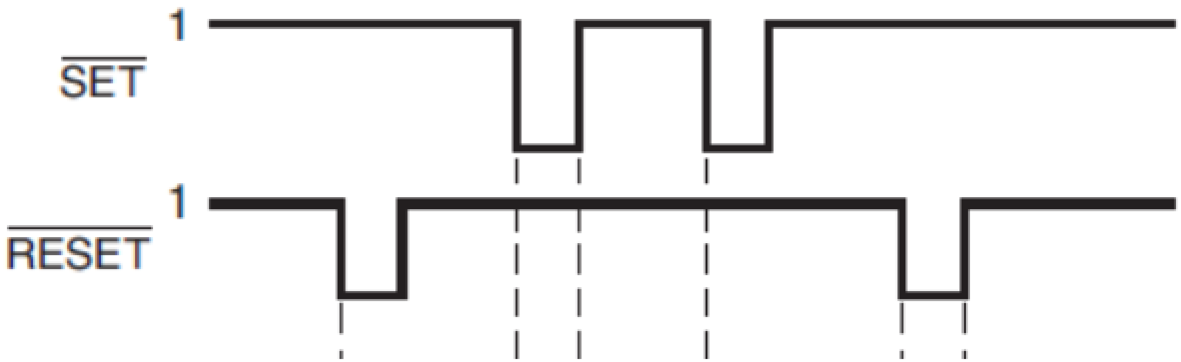
Set	Reset	Output
1	1	No change
0	1	Q = 1
1	0	Q = 0
0	0	Invalid *

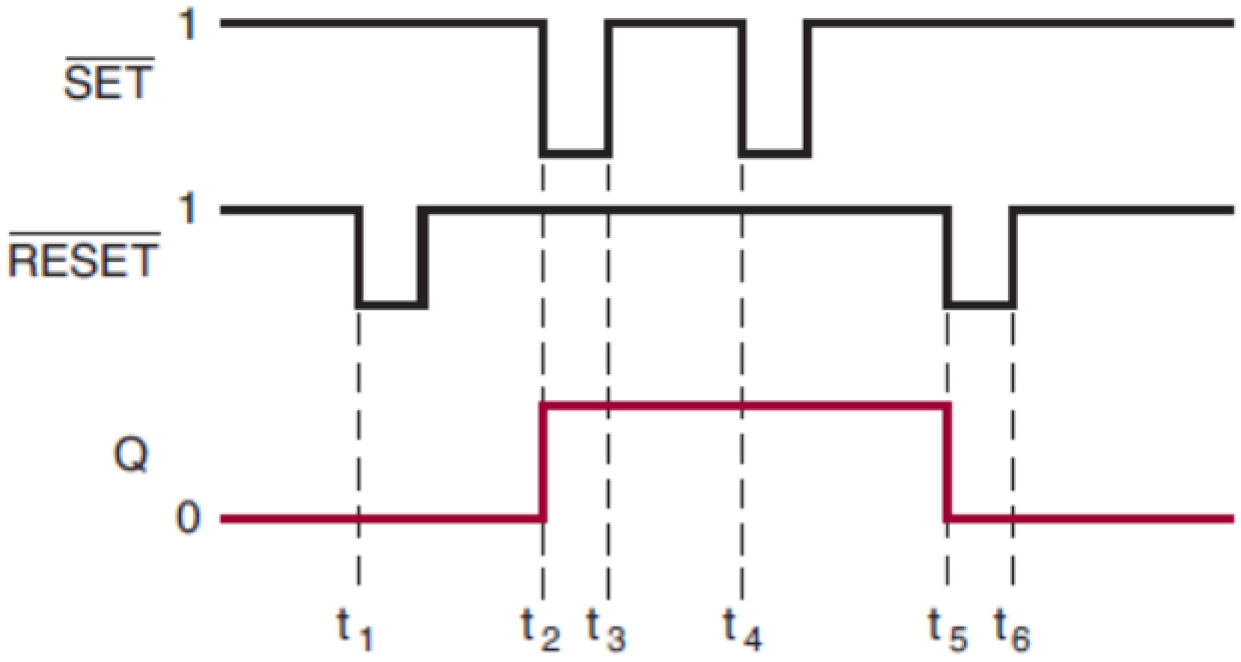
*Produces $Q = \overline{Q} = 1$.

(b)

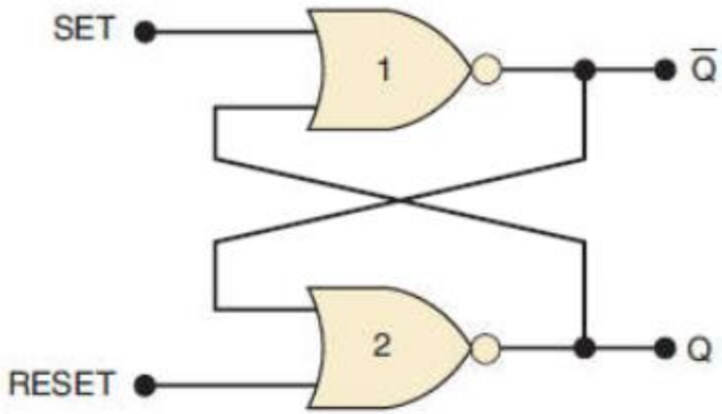
1. SET = RESET = 1. This condition is the normal resting state, and it has no effect on the output state. The Q and $\overline{Q}$ outputs will remain in whatever state they were in prior to this input condition.
2. SET = 0, RESET = 1. This will always cause the output to go to the $Q = 1$ state, where it will remain even after SET returns HIGH. This is called *setting* the latch.
3. SET = 1, RESET = 0. This will always produce the $Q = 0$ state, where the output will remain even after RESET returns HIGH. This is called *clearing* or *resetting* the latch.
4. SET = RESET = 0. This condition tries to set and clear the latch at the same time, and it produces $Q = \overline{Q} = 1$. If the inputs are returned to 1 simultaneously, the resulting state is unpredictable. This input condition should not be used.

Assuming the Q=0 initially, determine the Q waveform for the NAND Latch



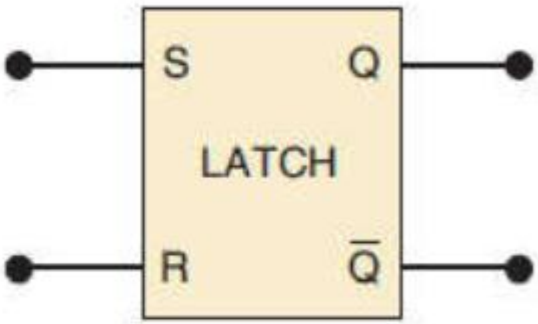

1
$\overline{SET}$
1
$\overline{RESET}$
Q
0
t_1 t_2 t_3 t_4 t_5 t_6

NOR Gate SR Latch



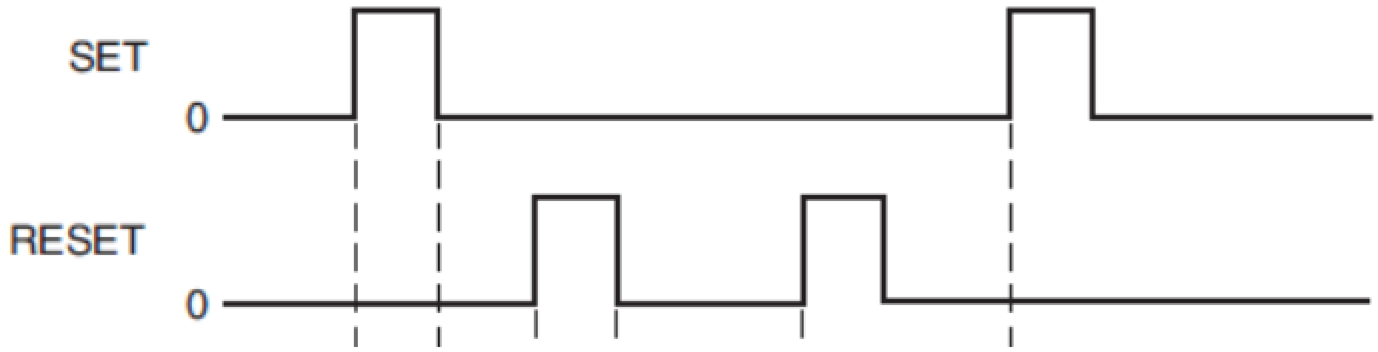
Set	Reset	Output
0	0	No change
1	0	Q = 1
0	1	Q = 0
1	1	Invalid*

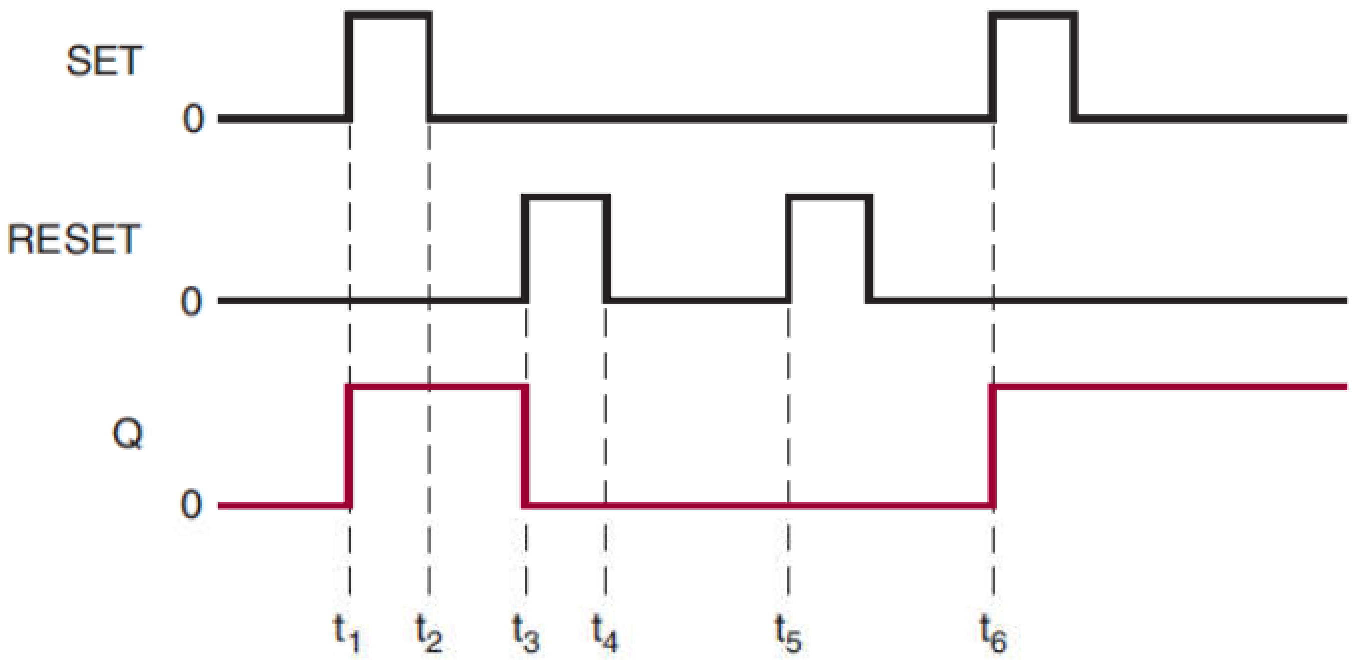
*Produces $Q = \overline{Q} = 0$.



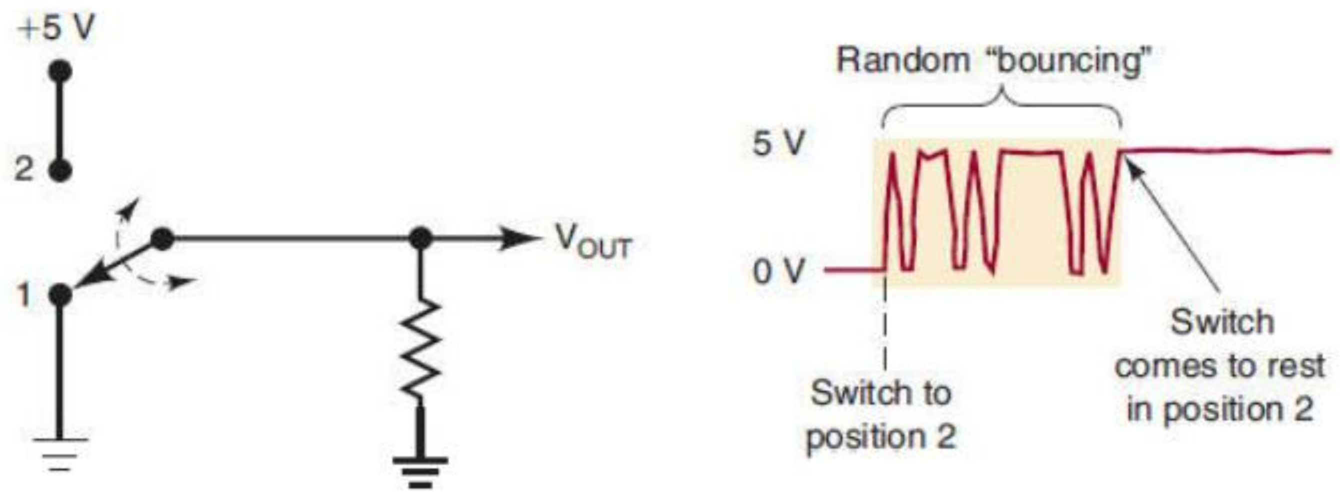
1. SET = RESET = 0. This is the normal resting state for the NOR latch, and it has no effect on the output state. Q and $\overline{Q}$ will remain in whatever state they were in prior to the occurrence of this input condition.
2. SET = 1, RESET = 0. This will always set $Q = 1$, where it will remain even after SET returns to 0.
3. SET = 0, RESET = 1. This will always clear $Q = 0$, where it will remain even after RESET returns to 0.
4. SET = 1, RESET = 1. This condition tries to set and reset the latch at the same time, and it produces $Q = \overline{Q} = 0$. If the inputs are returned to 0 simultaneously, the resulting output state is unpredictable. This input condition should not be used.

Assuming the Q=0 initially, determine the Q waveform for the NOR Latch

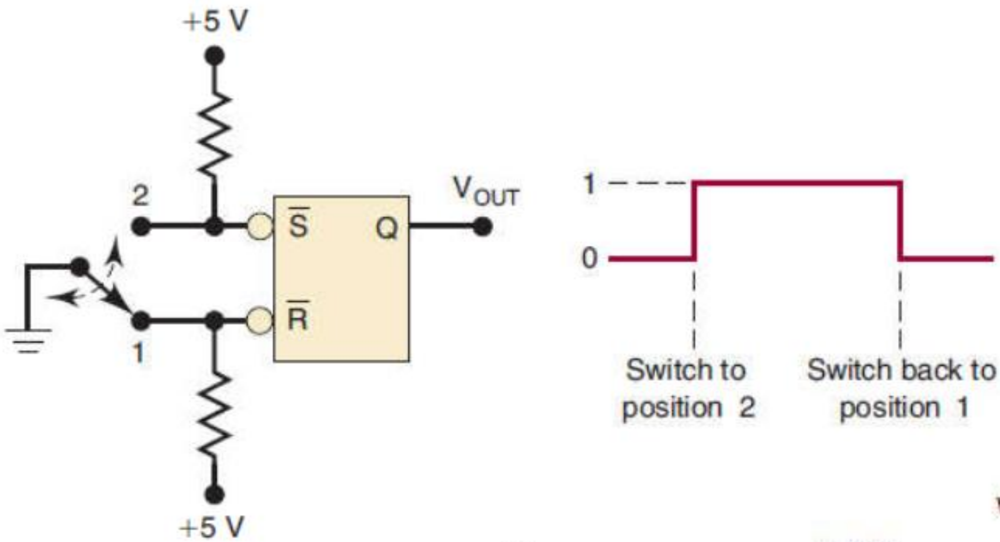


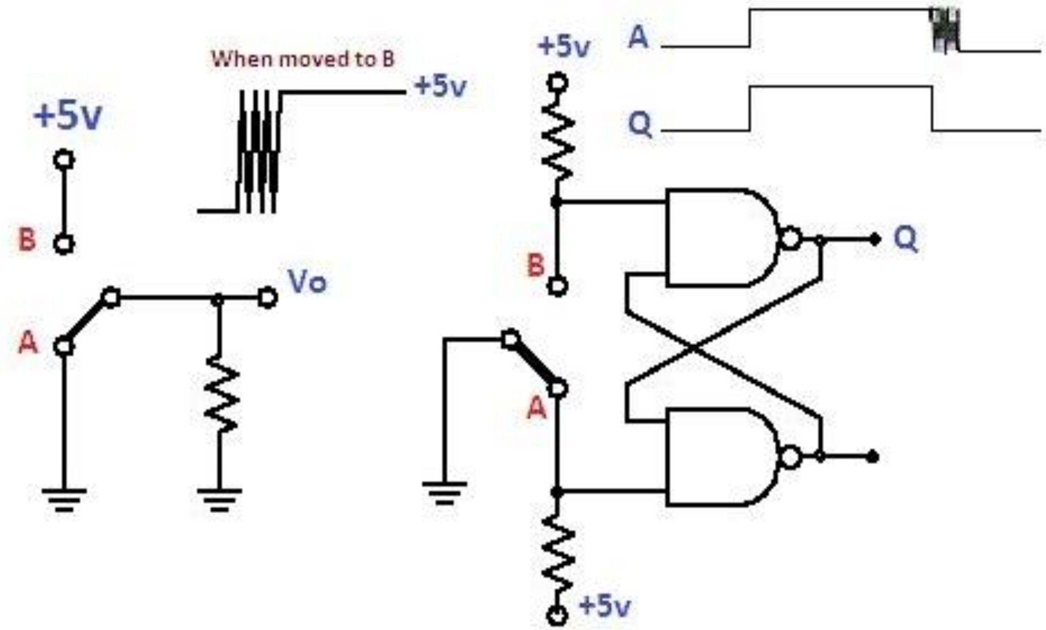

SET
0
RESET
0
Q
0
t_1
t_2
t_3
t_4
t_5
t_6

Switch Debounce

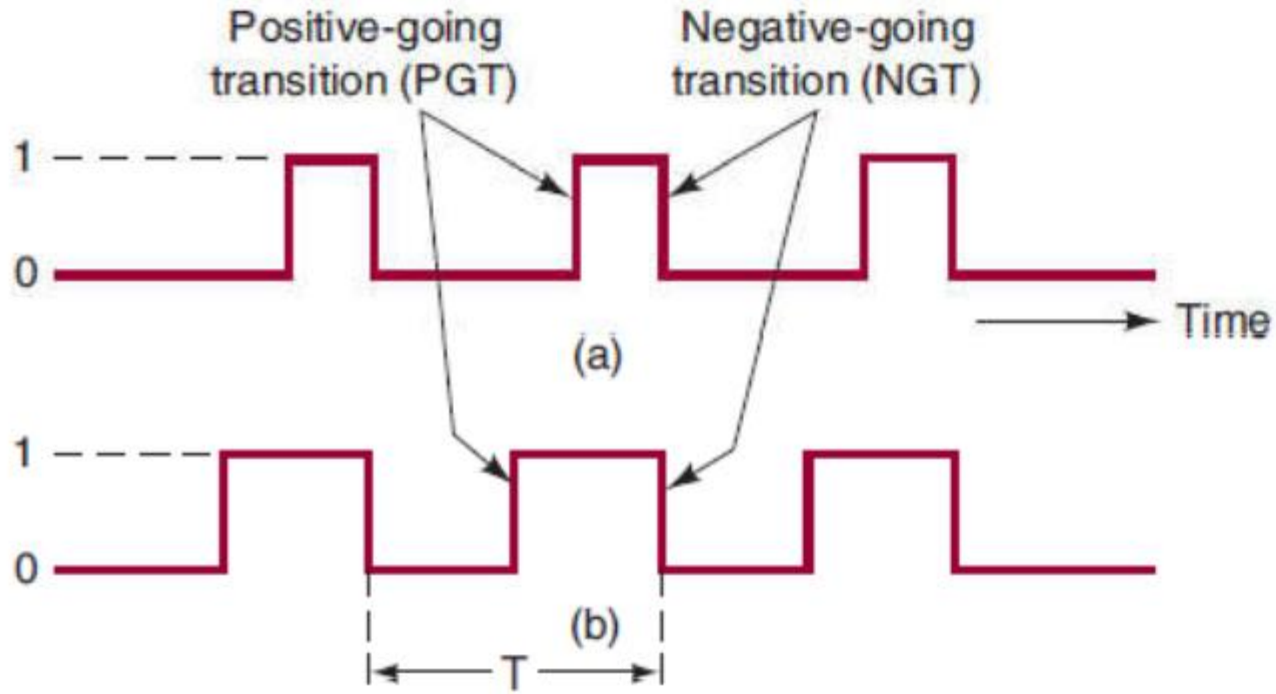


Digital Logic Circuits

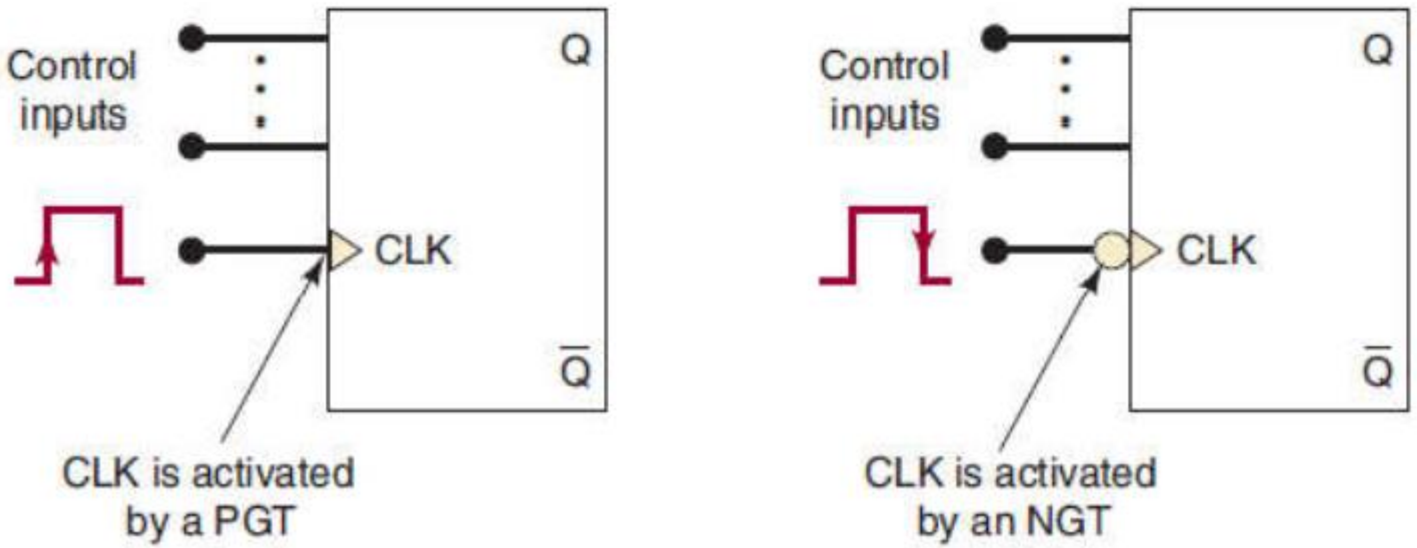




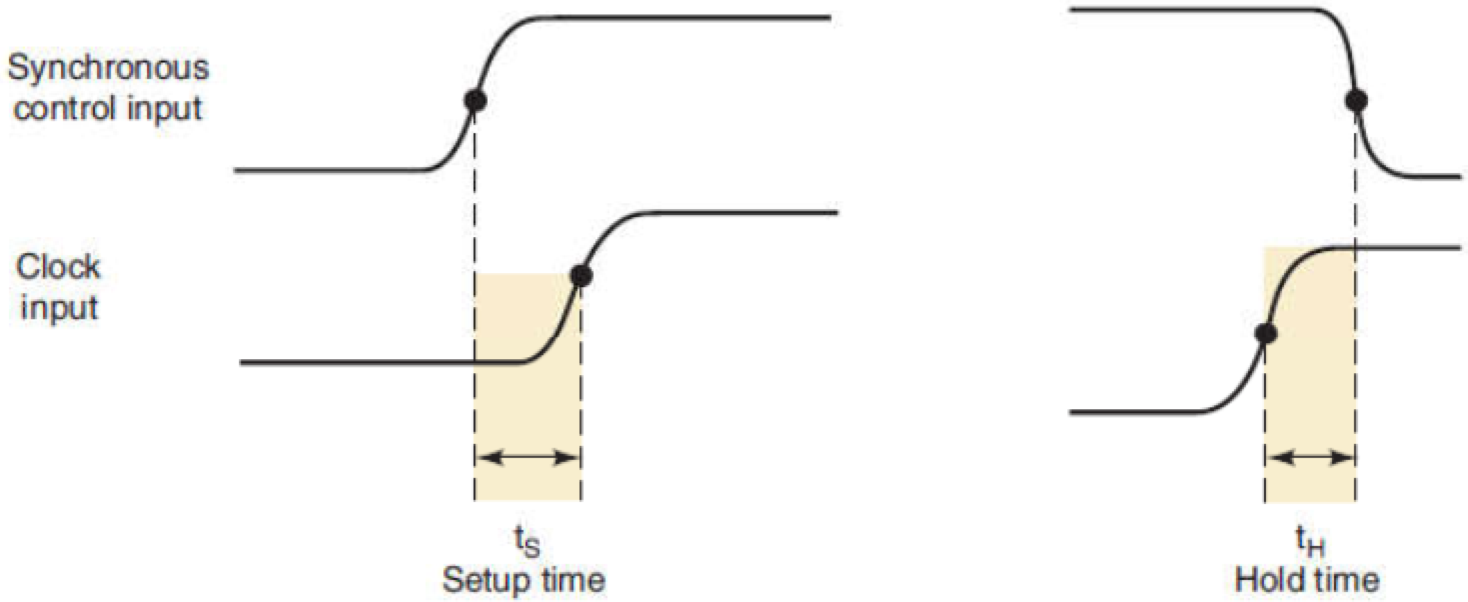
Pulse triggering



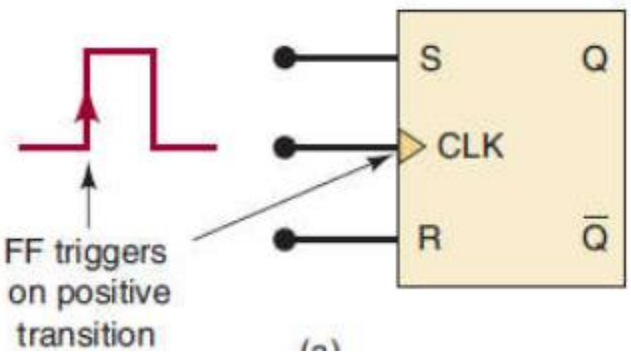
Edge triggering



Setup and Hold time



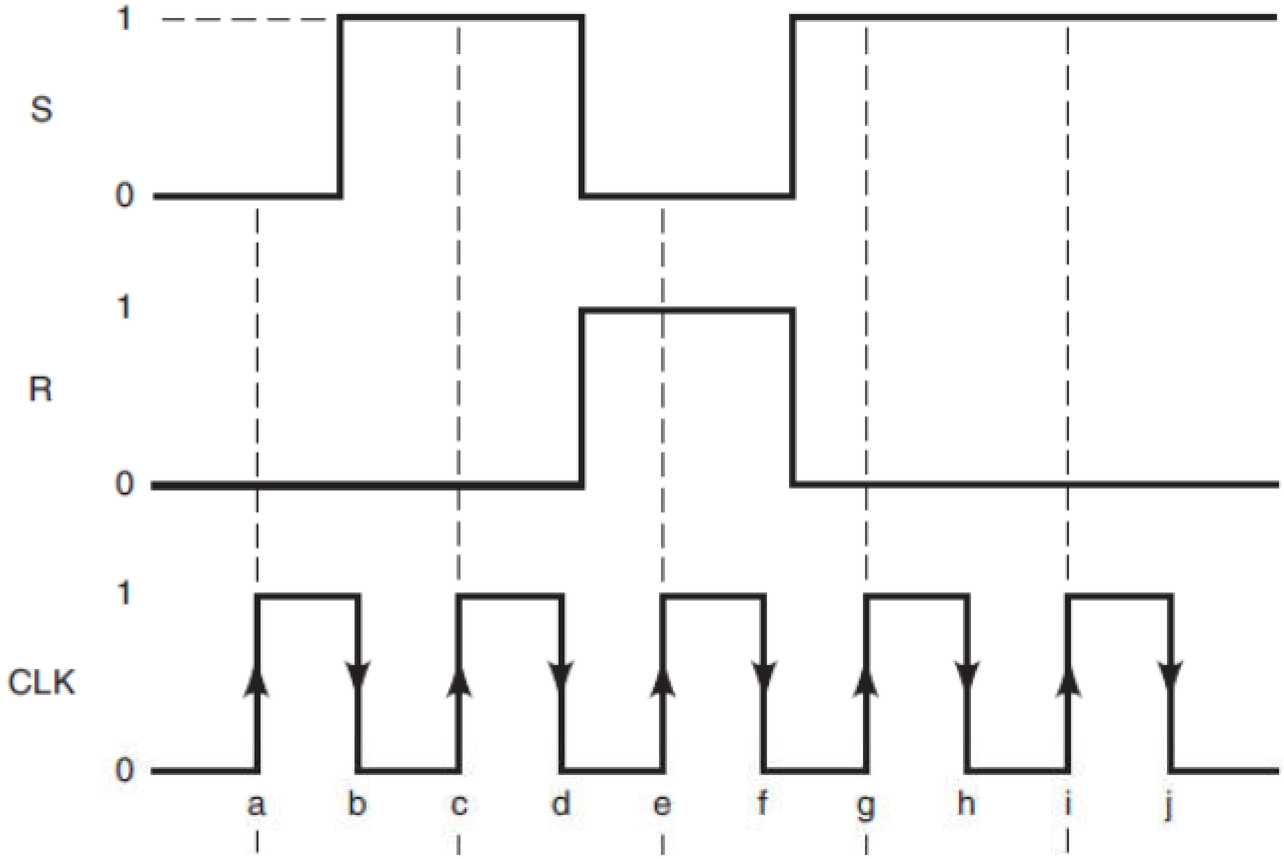
Clocked SR Flip Flop

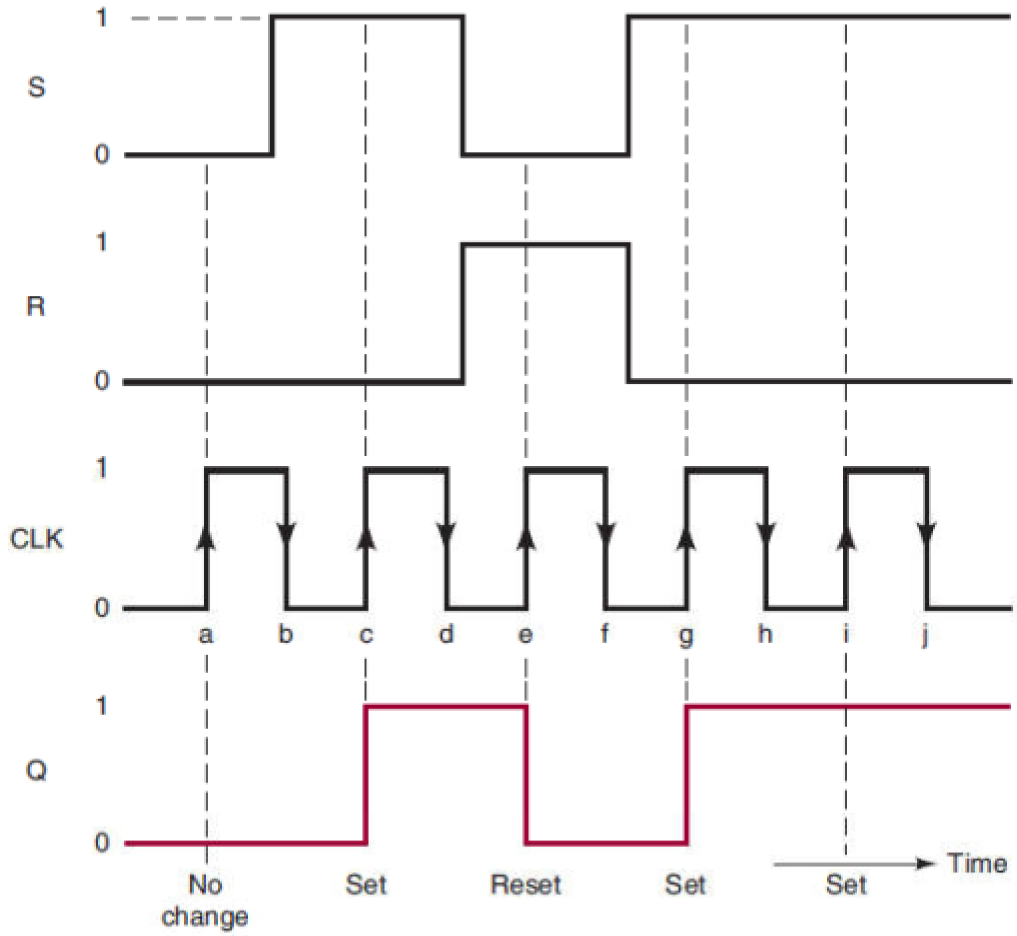


Inputs			Output
S	R	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	Ambiguous

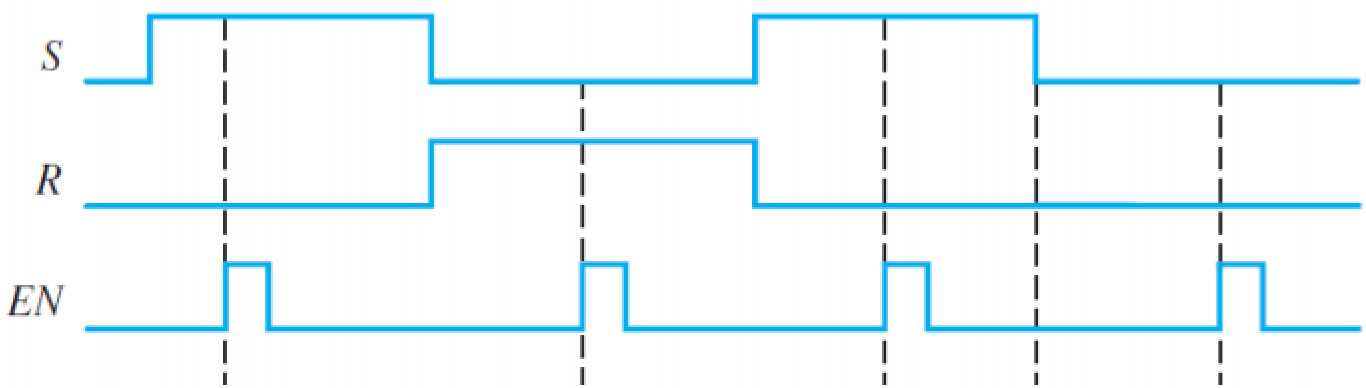
Q_0 is output level prior to ↑ of CLK.
↓ of CLK produces no change in Q.

Problem

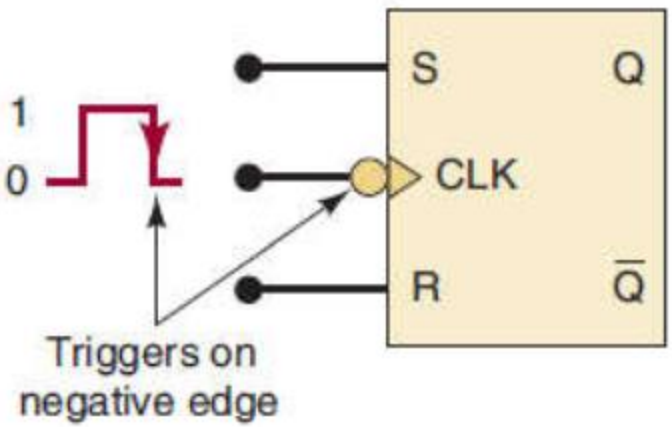



S
R
CLK
Q
1
0
1
0
1
0
1
0
a
b
c
d
e
f
g
h
i
j
No
change
Set
Reset
Set
Set
Time

- Assuming Q=0 initially, Predict the output Waveform For a Gated SR Latch.

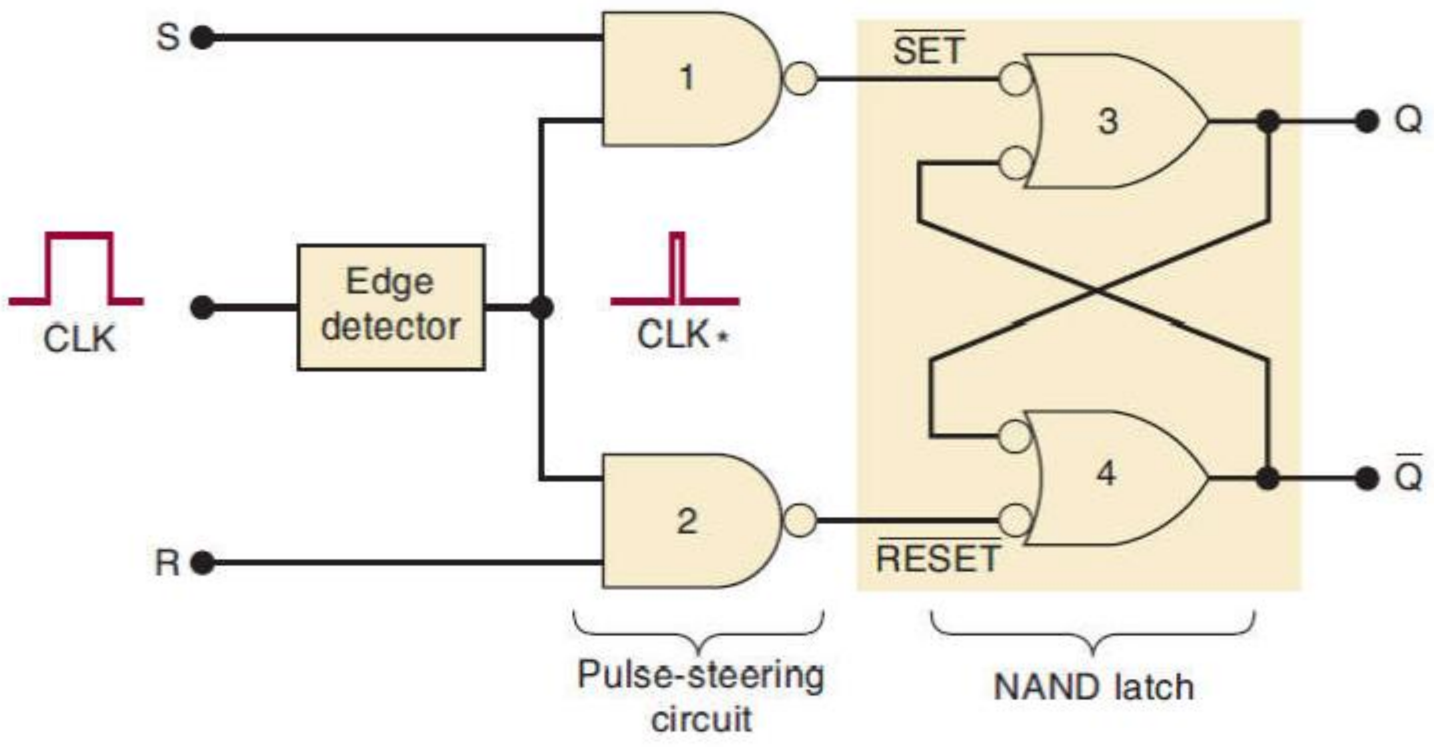


Negative Edge triggered SR Flip Flop

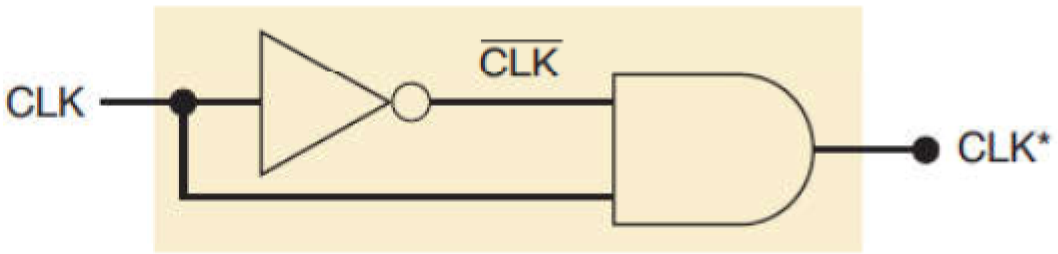


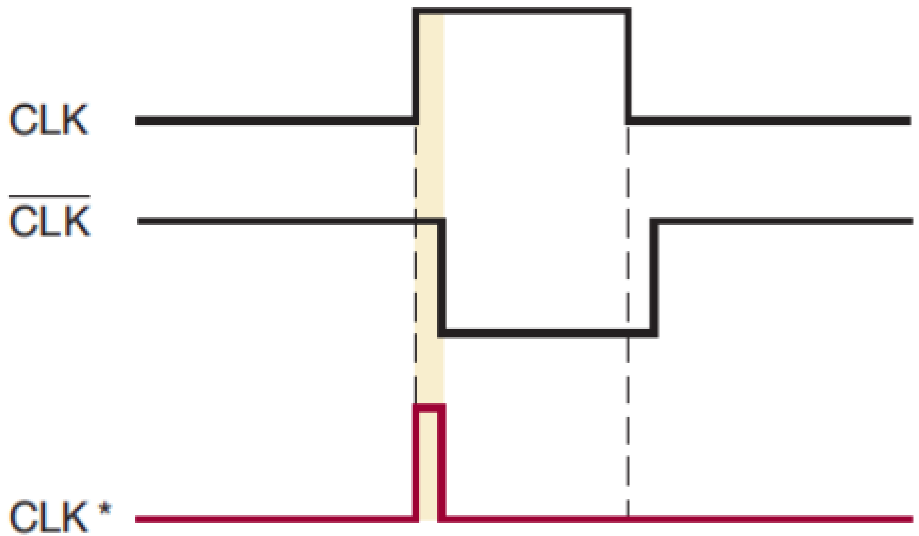
Inputs			Output
S	R	CLK	Q
0	0	↓	Q_0 (no change)
1	0	↓	1
0	1	↓	0
1	1	↓	Ambiguous

Internal Circuit of Clocked SR Flip Flop

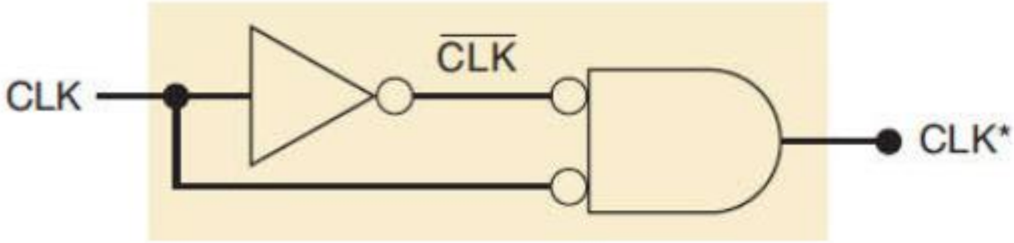


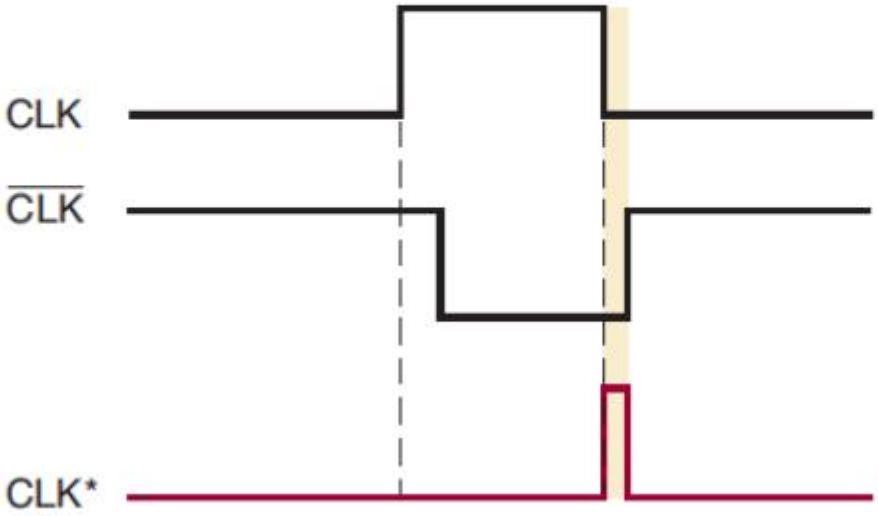
Generation of Positive Edge Triggering



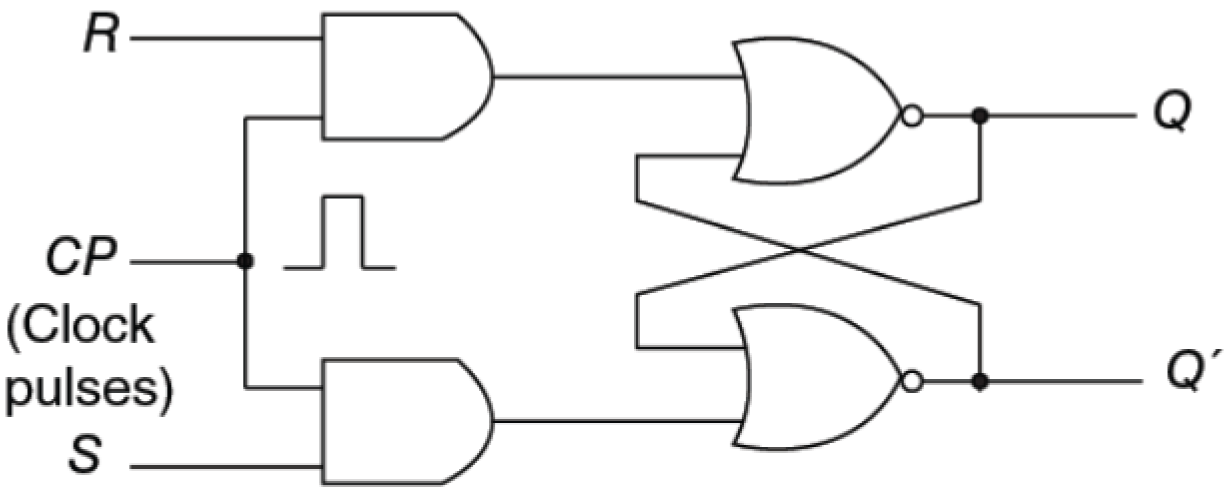


Generation of Negative Edge Triggering

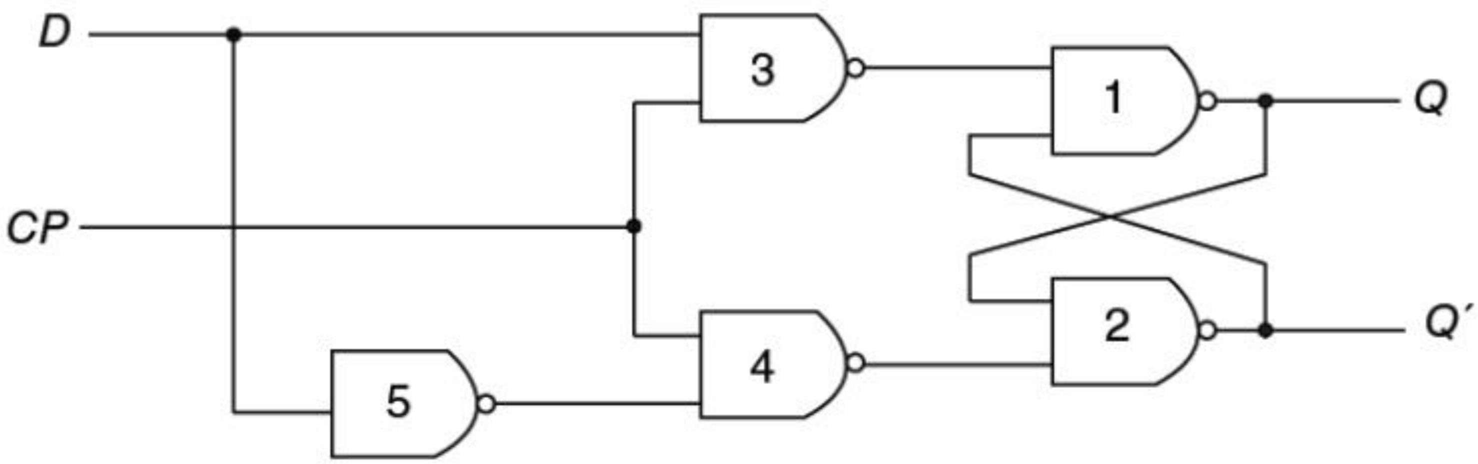


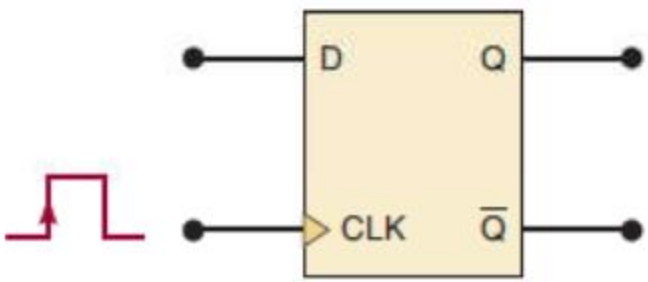


Clocked RS Flip Flop

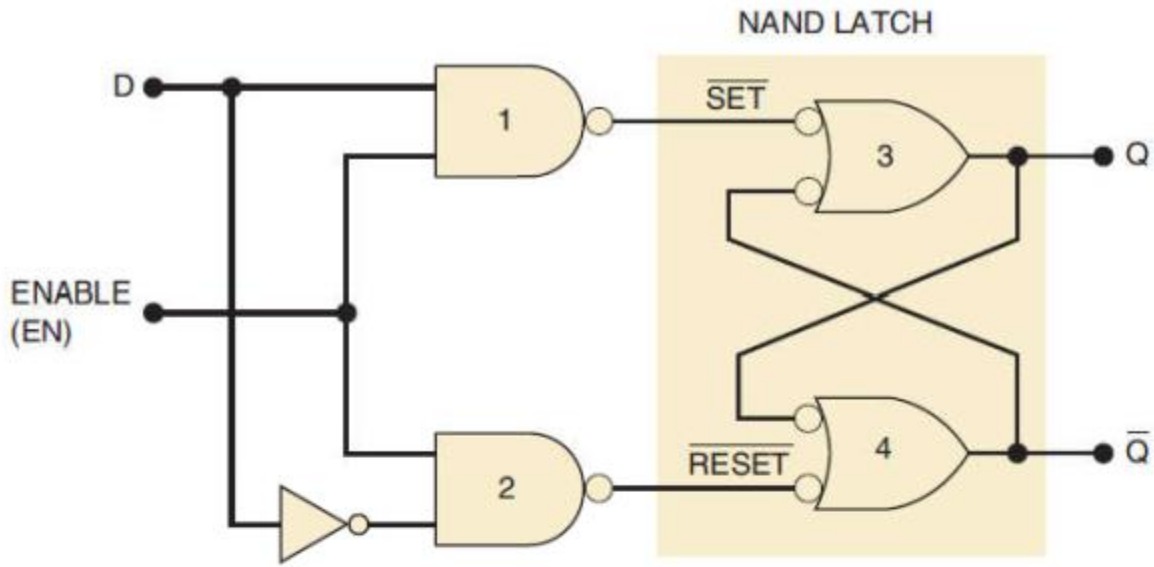


D-Flip Flop



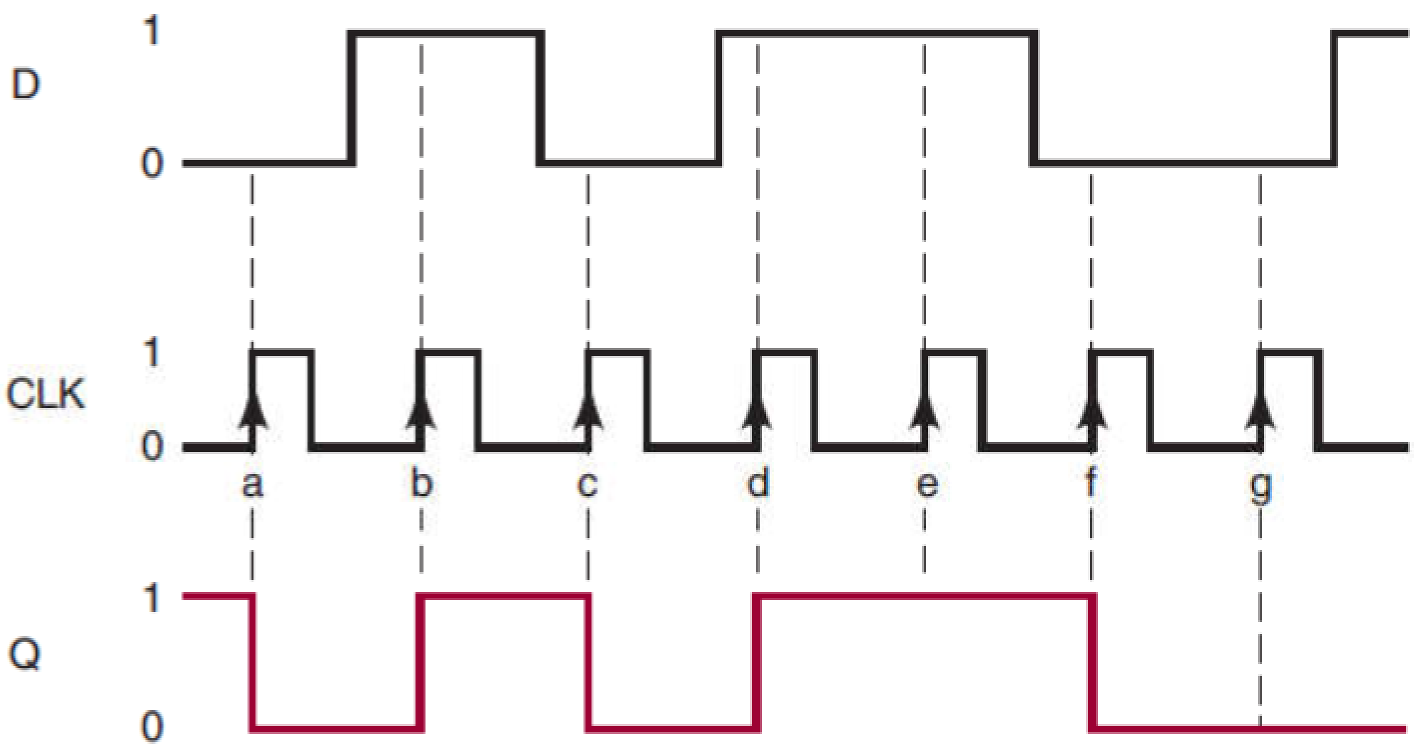


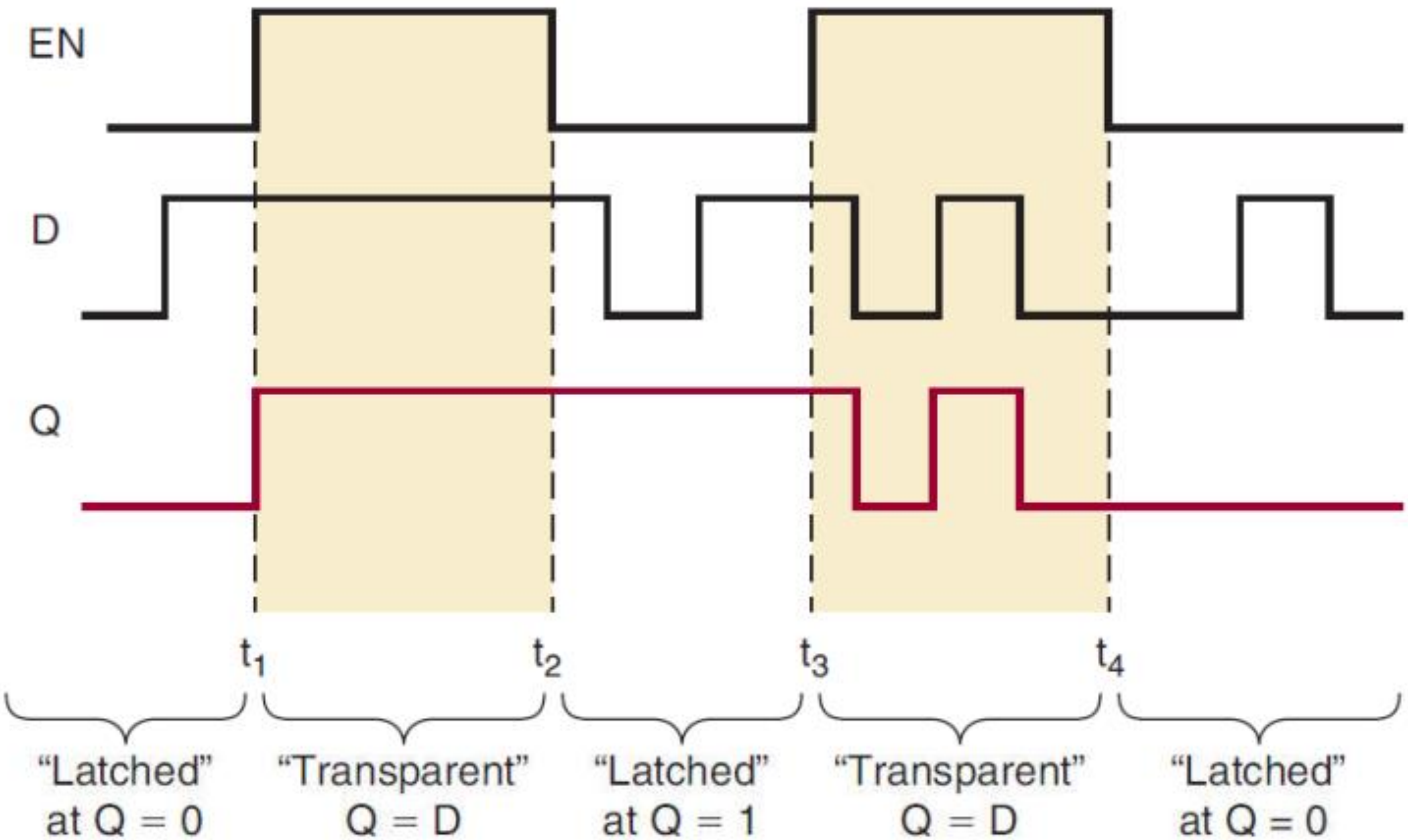
D	CLK	Q
0	↑	0
1	↑	1



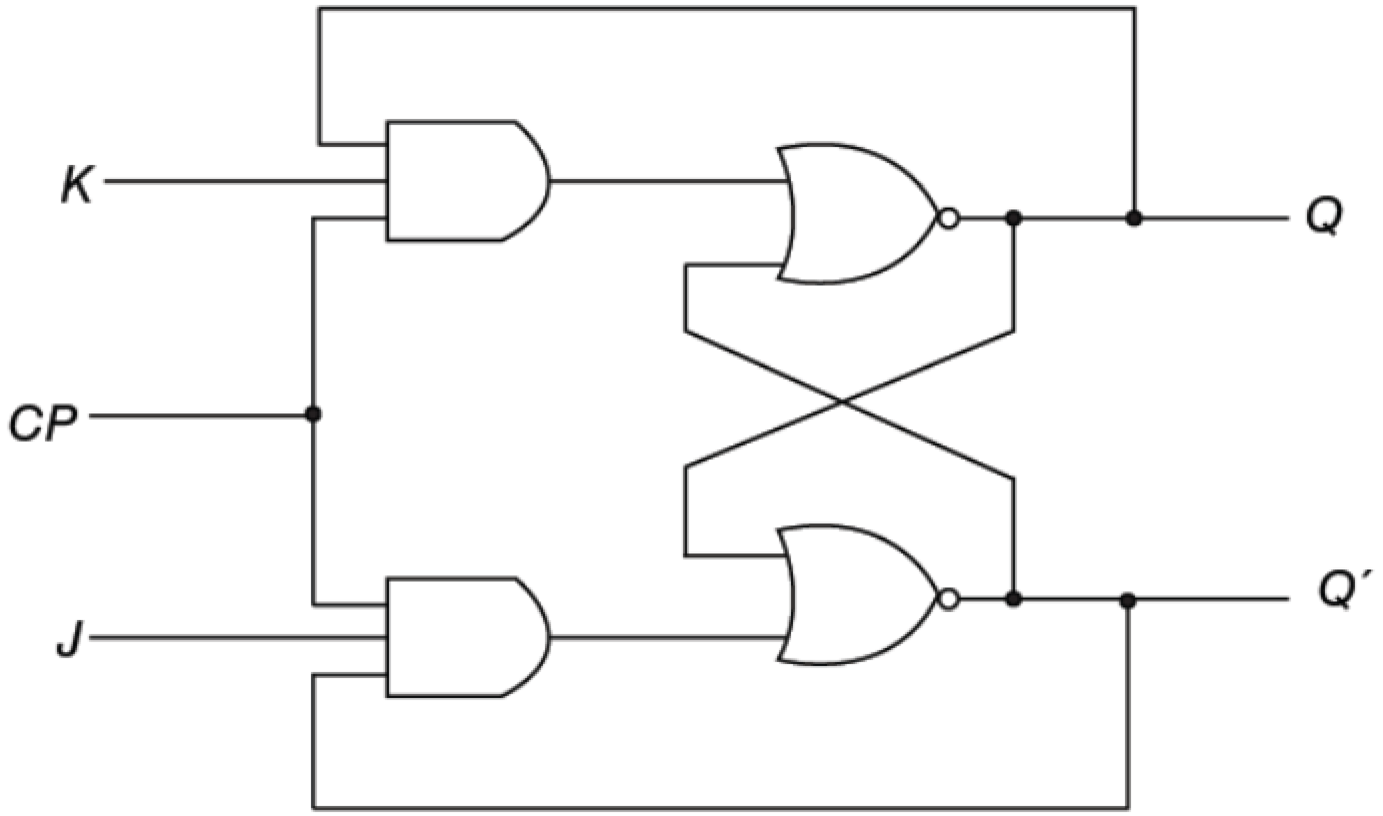
Inputs		Output
EN	D	Q
0	X	Q_0 (no change)
1	0	0
1	1	1

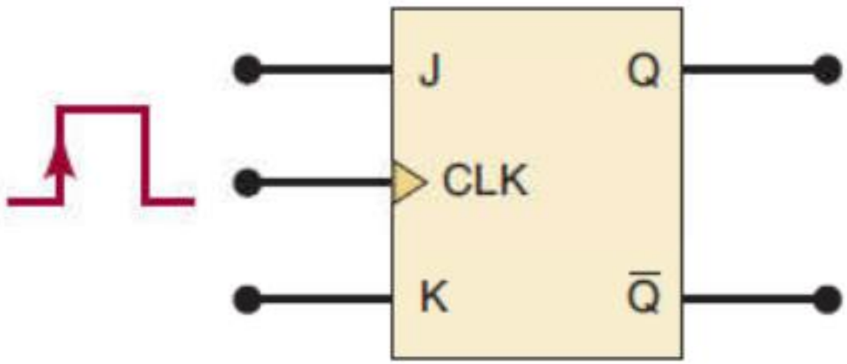
"X" indicates "don't care."
Q_0 is state Q just
prior to EN going LOW.


D
1
0
CLK
1
0
a
b
c
d
e
f
g
Q
1
0


EN
D
Q
t1
t2
t3
t4
"Latched" at Q = 0
"Transparent" Q = D
"Latched" at Q = 1
"Transparent" Q = D
"Latched" at Q = 0

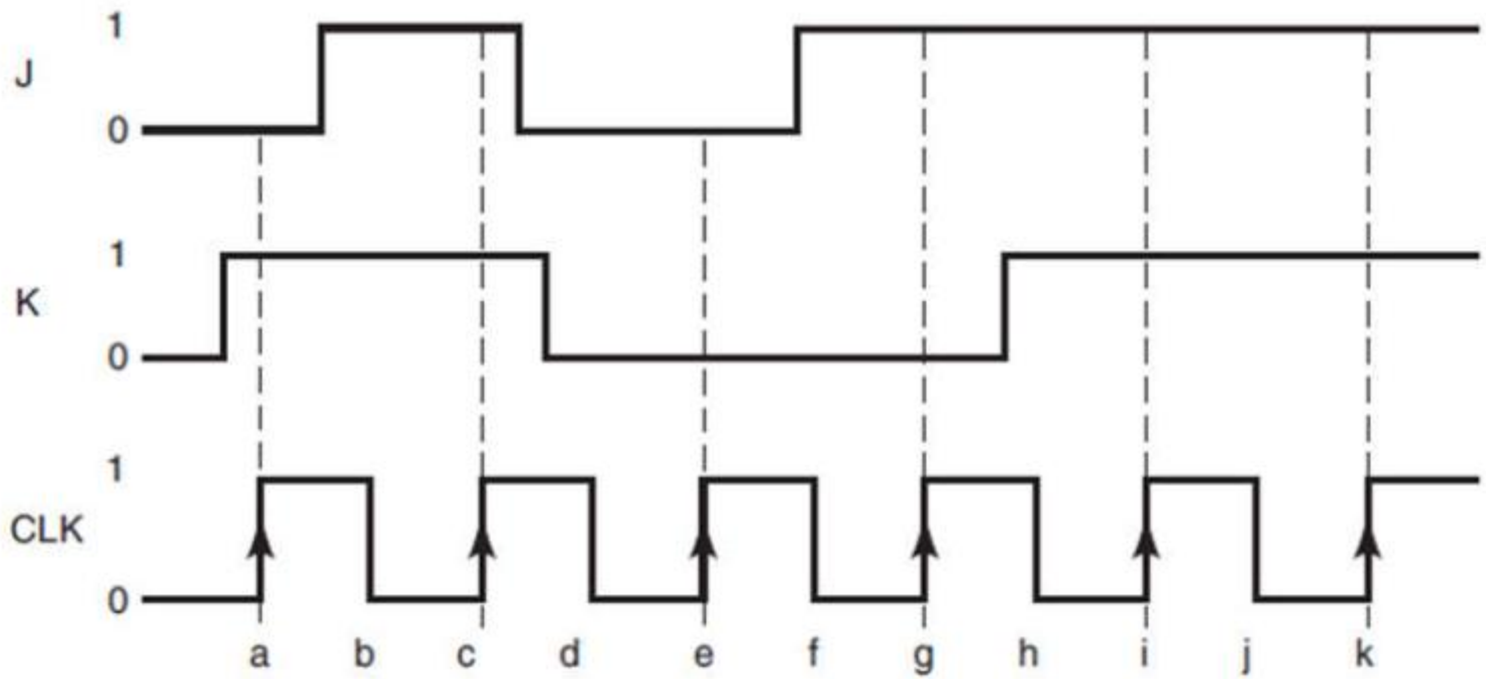
Jack-Kibly (JK) Flip Flop



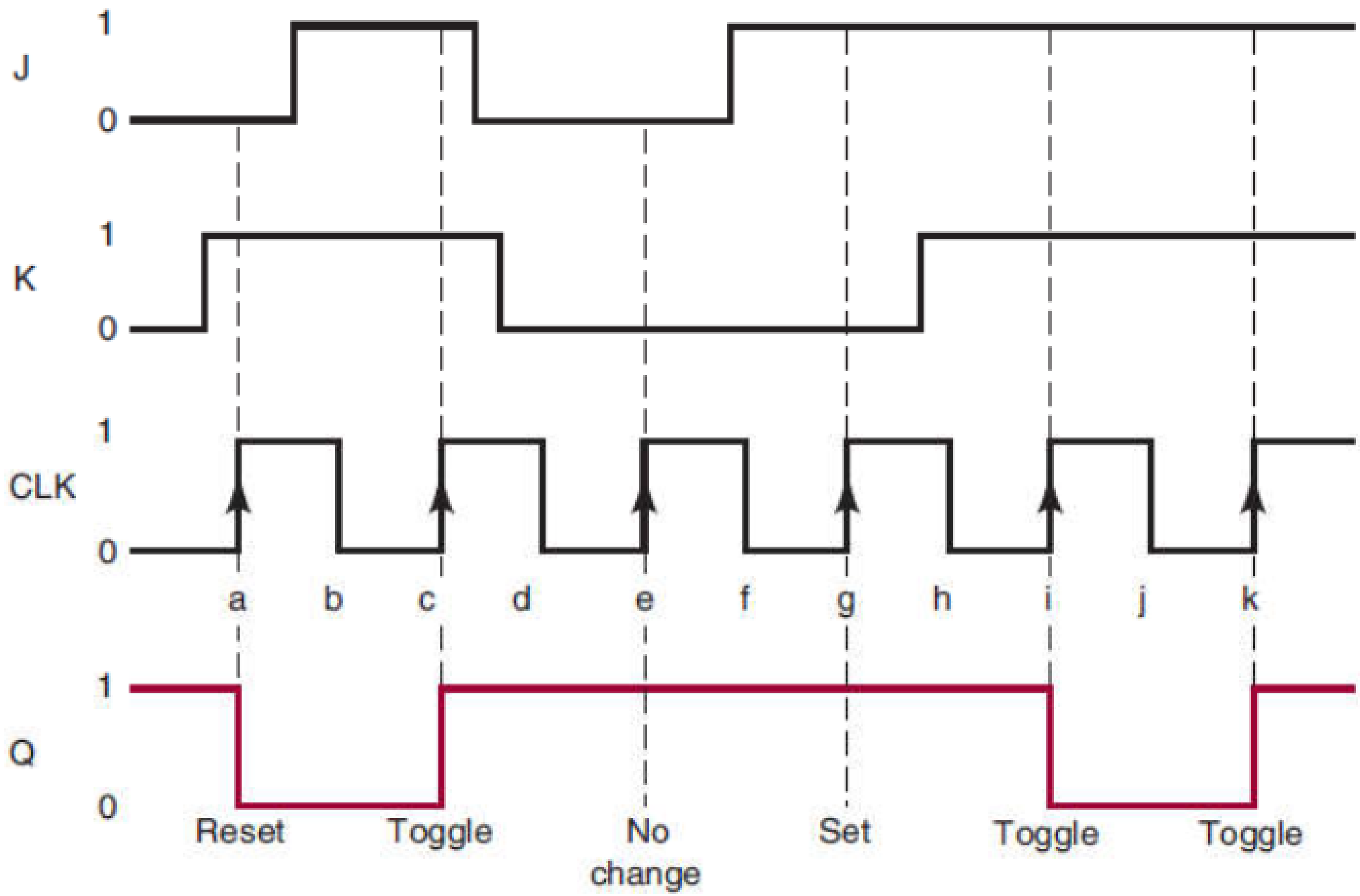


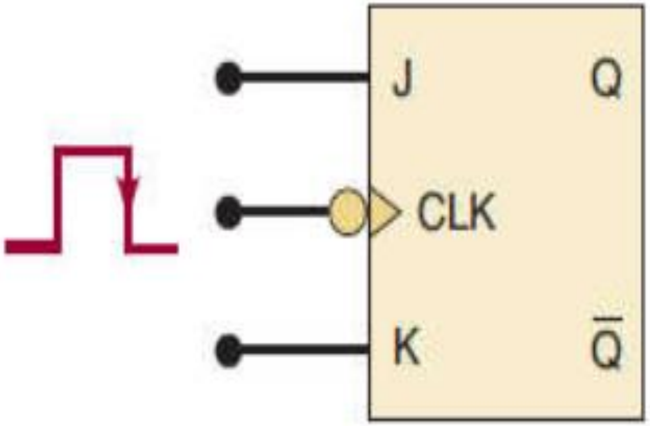
J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q_0}$ (toggles)

Assume Q=1 initially



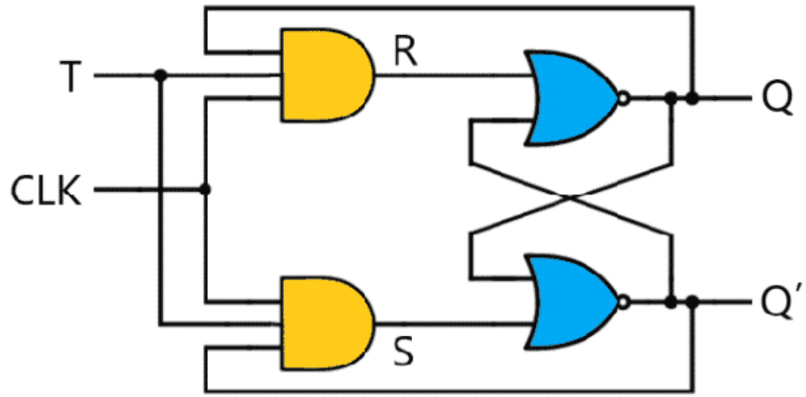
Digital Logic Circuits

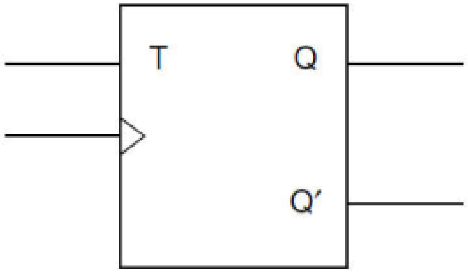


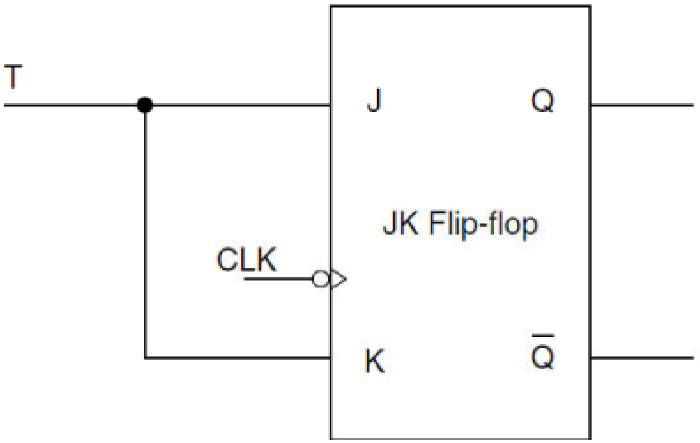


J	K	CLK	Q
0	0	↓	Q_0 (no change)
1	0	↓	1
0	1	↓	0
1	1	↓	$\overline{Q_0}$ (toggles)

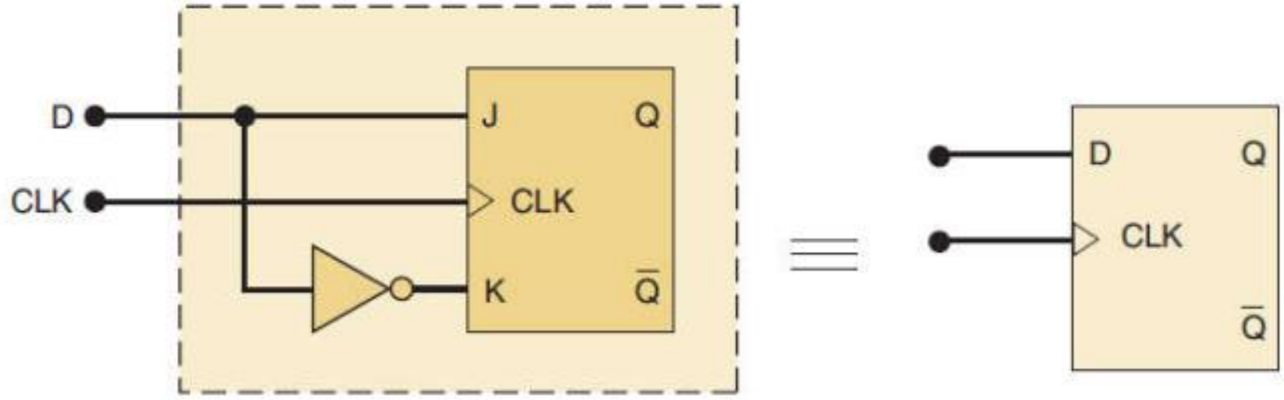
T Flip Flop



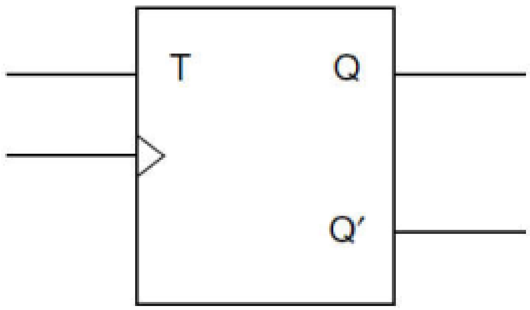


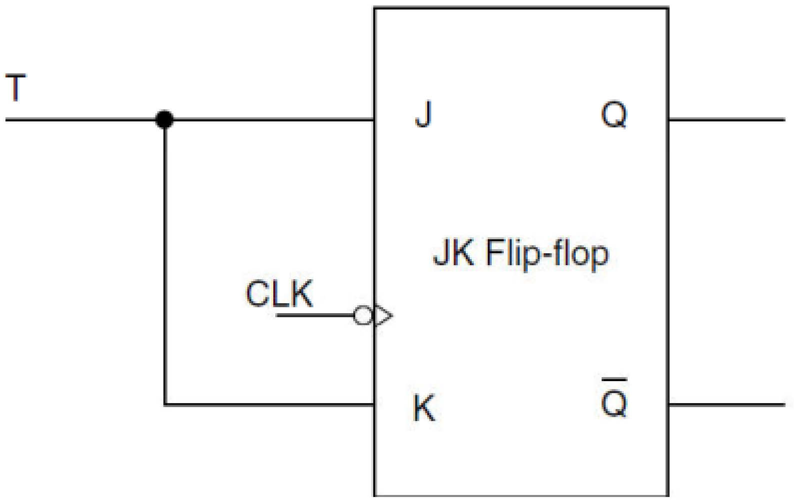


D Flip Flop from JK Flip Flop



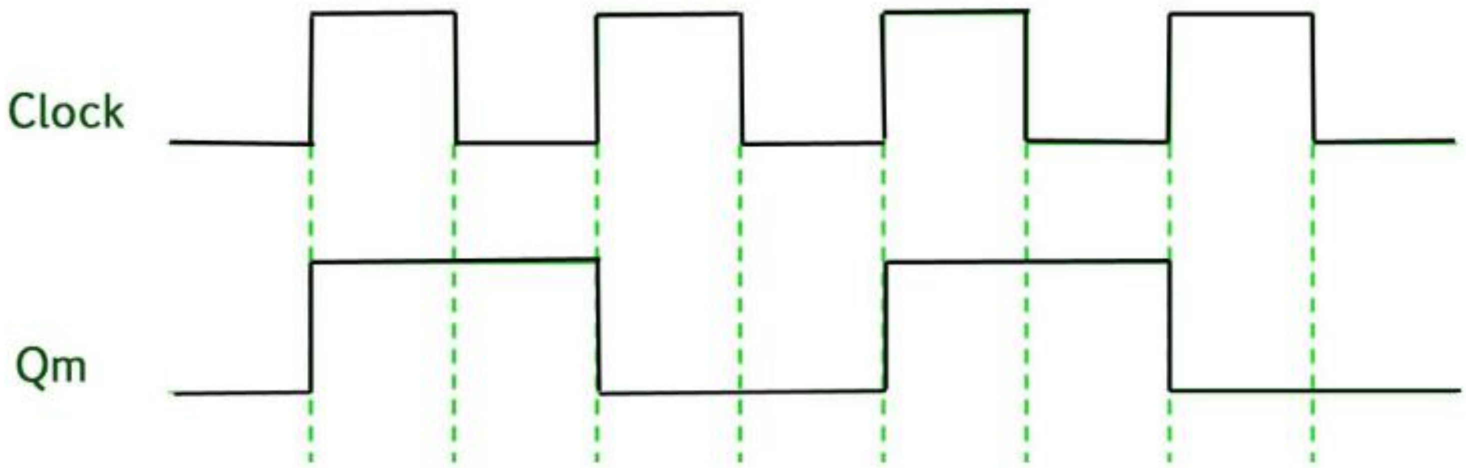
T Flip Flop from JK Flip Flop



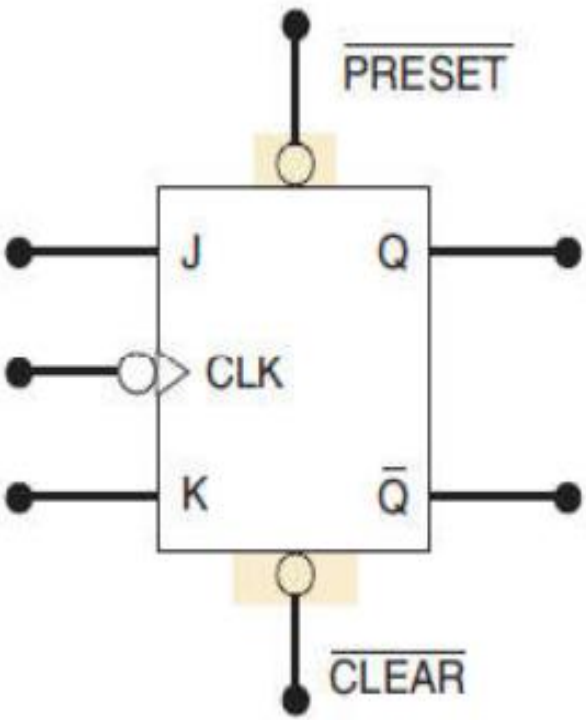


Concept of Frequency Division

- ❑ Flip Flops can be used to divide input frequencies
- ❑ The frequency to be divided is applied as clock

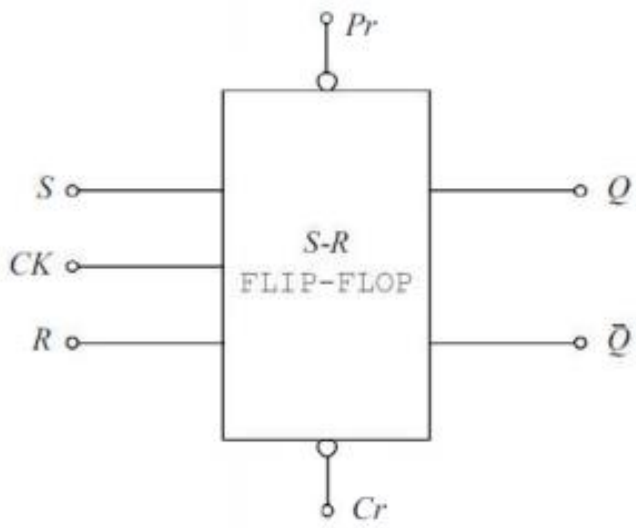


Preset and Clear

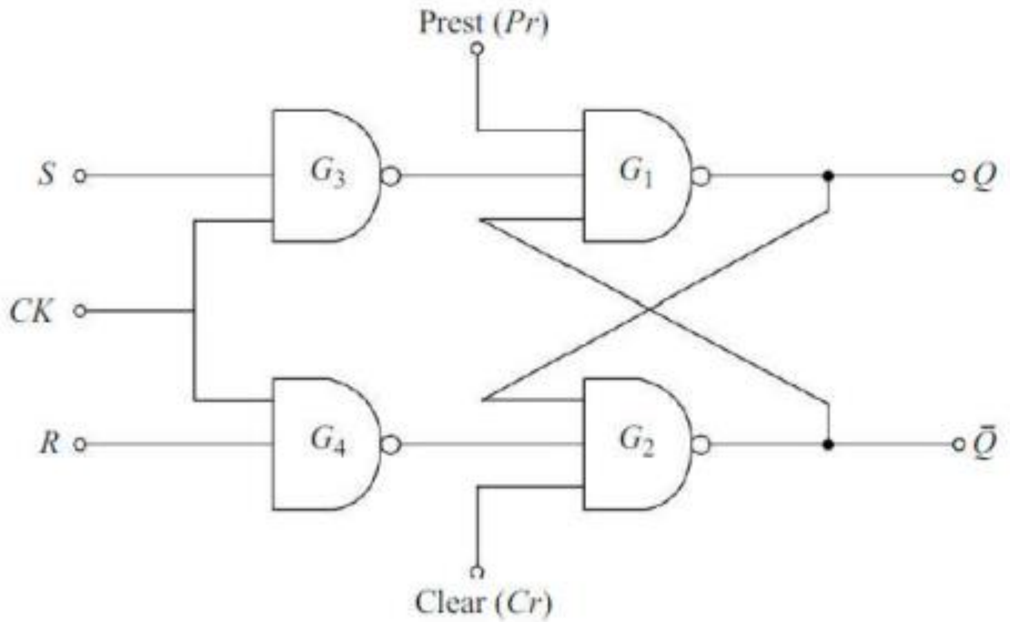


J	K	Clk	$\overline{PRE}$	$\overline{CLR}$	Q
0	0	↓	1	1	Q (no change)
0	1	↓	1	1	0 (Synch reset)
1	0	↓	1	1	1 (Synch set)
1	1	↓	1	1	$\overline{Q}$ (Synch toggle)
x	x	x	1	1	Q (no change)
x	x	x	1	0	0 (asynch clear)
x	x	x	0	1	1 (asynch preset)
x	x	x	0	0	(Invalid)

SR Flip Flop

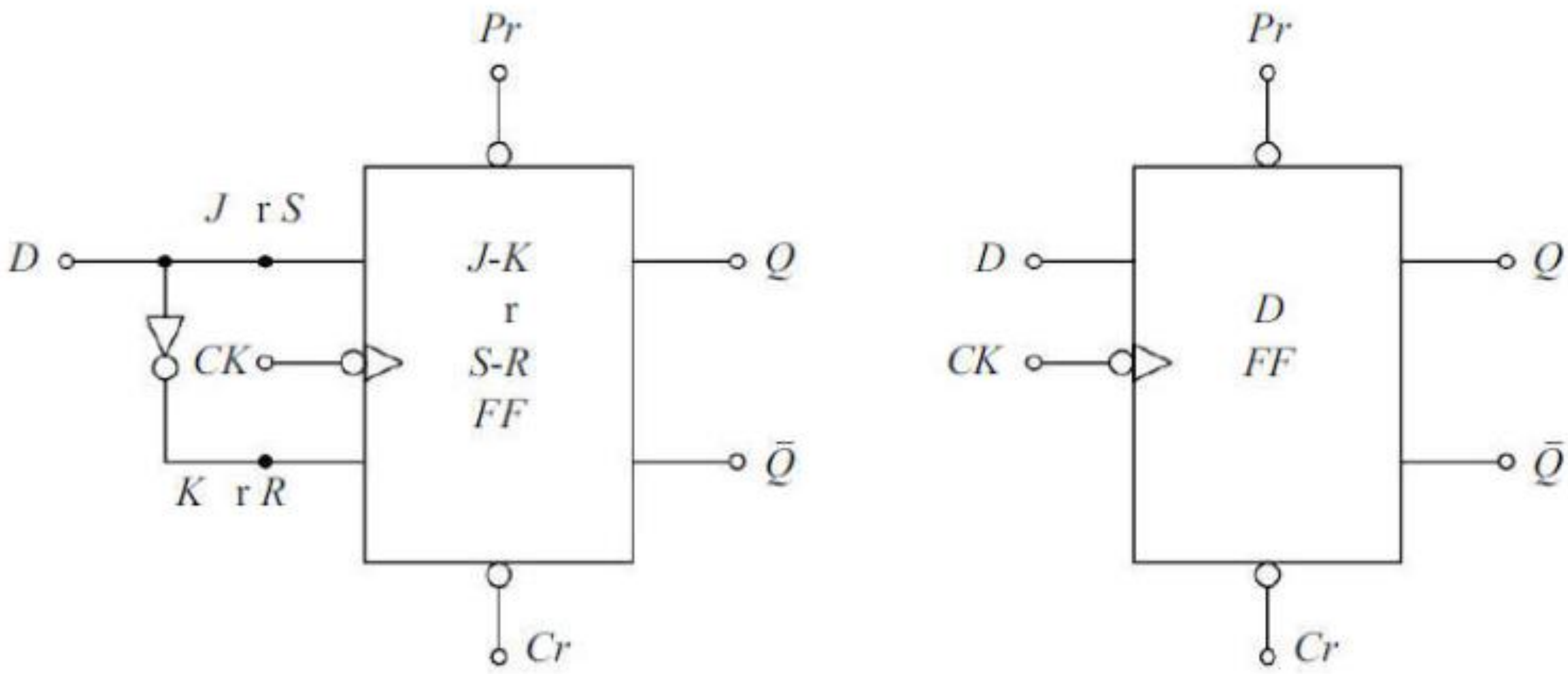


Pr
S
CK
R
S-R
FLIP-FLOP
Q
$\bar{Q}$
Cr

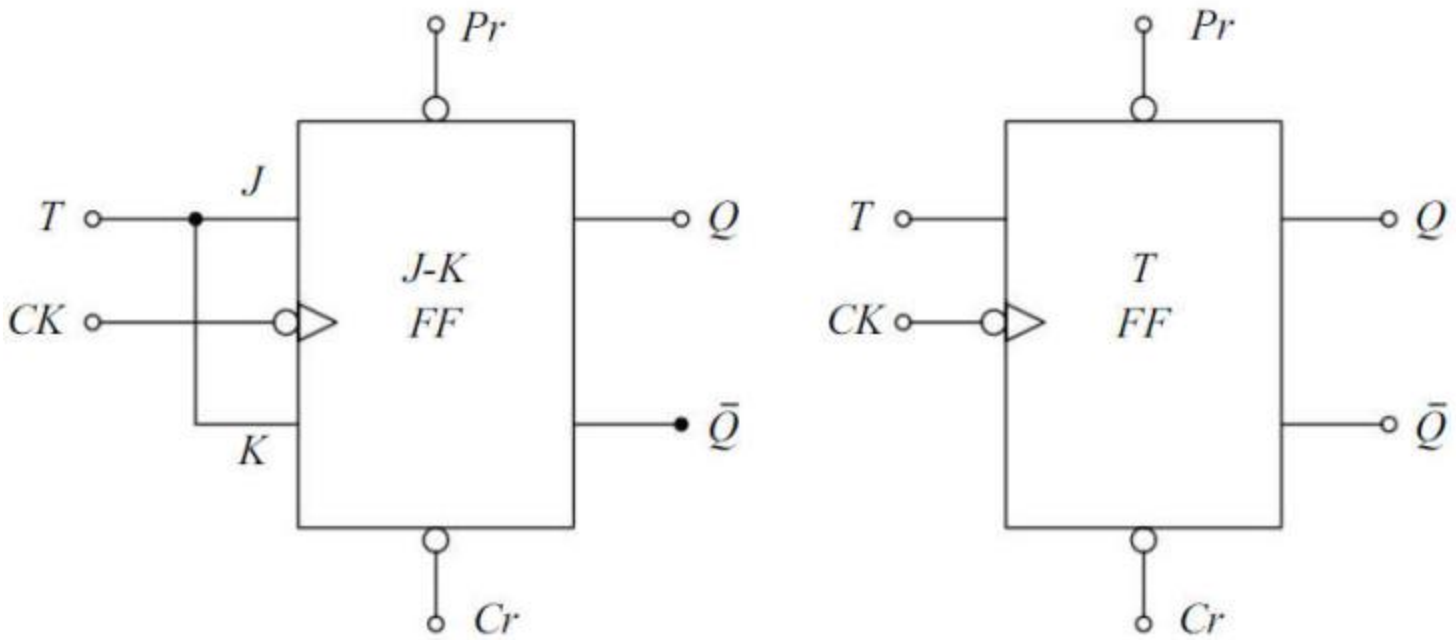


Prest (Pr)
S
CK
R
G_3
G_4
G_1
G_2
Q
$\bar{Q}$
Clear (Cr)

D Flip Flop

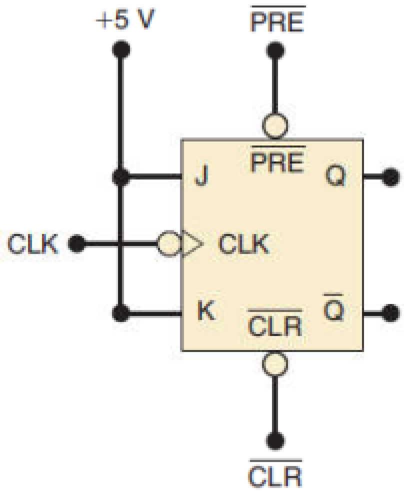


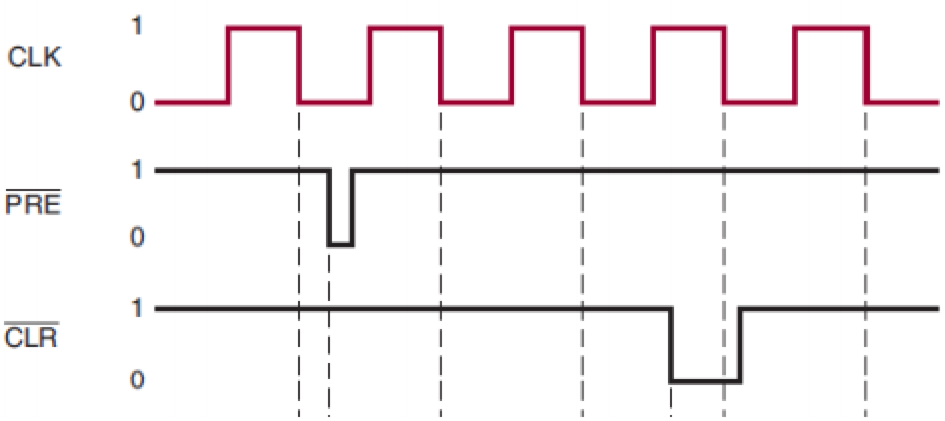
T Flip Flop



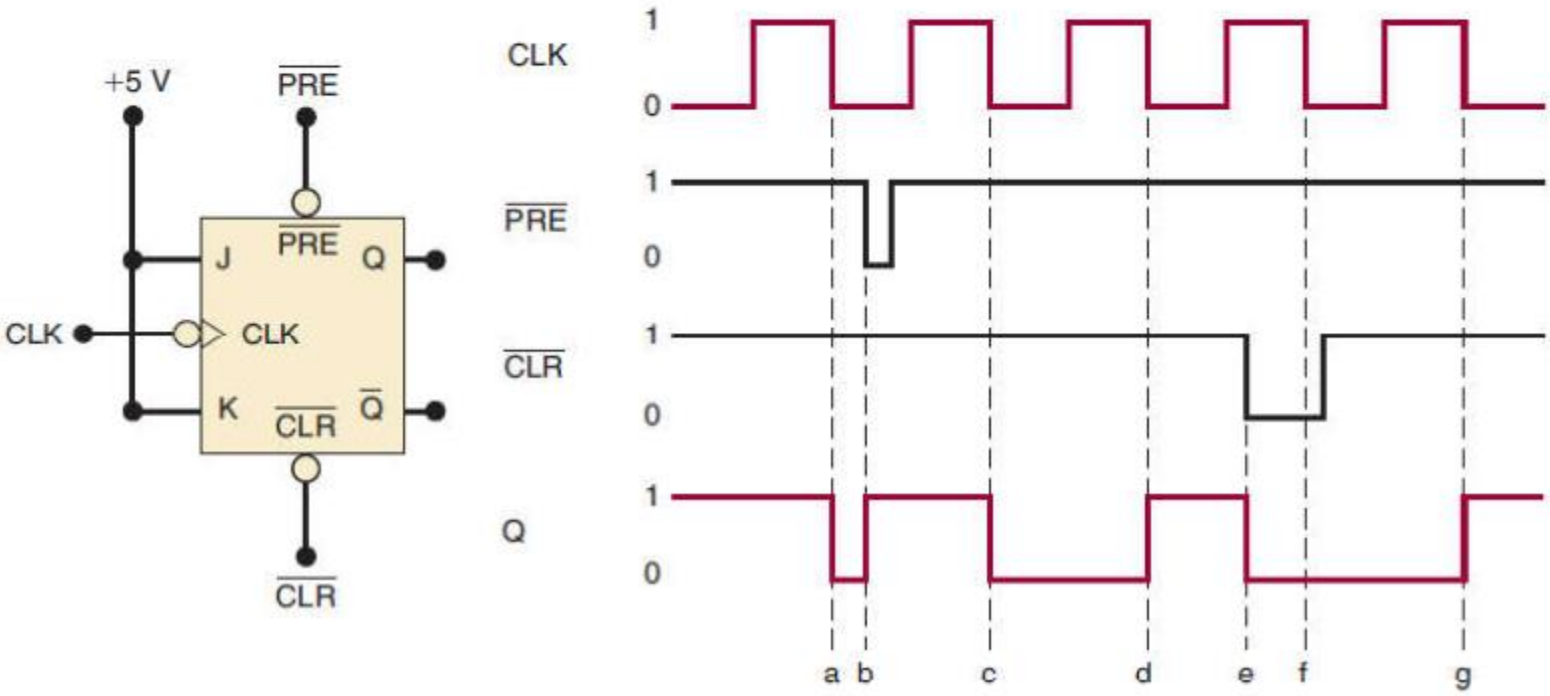
T Flip Flop from JK Flip Flop

Assume Q=1, Initially

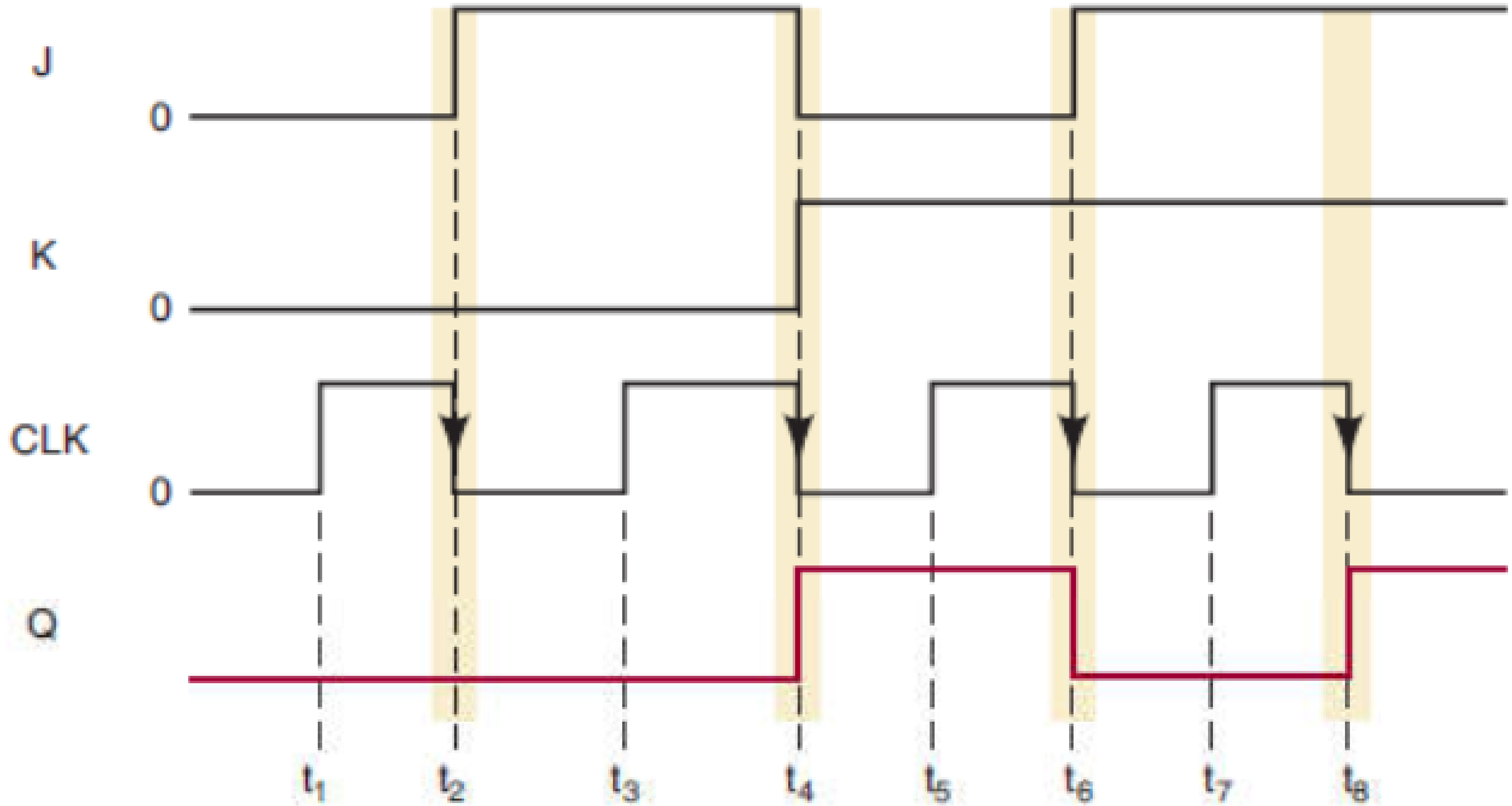




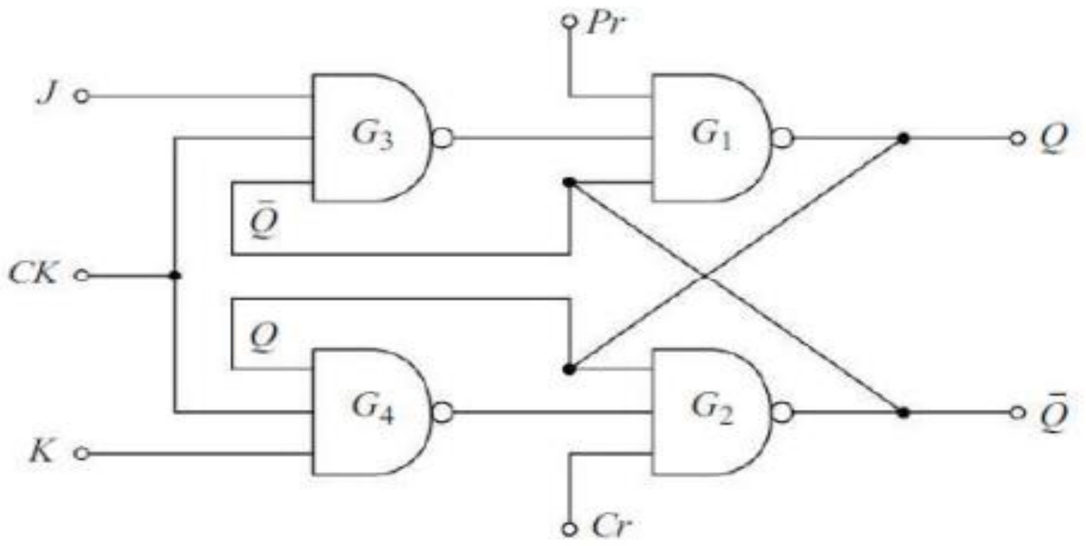
Example

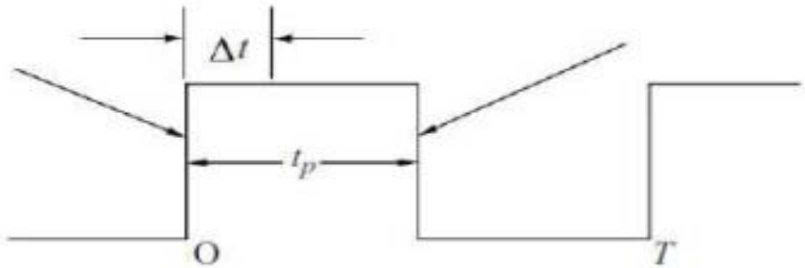


Effect of Propagation Delays



Race Around Condition





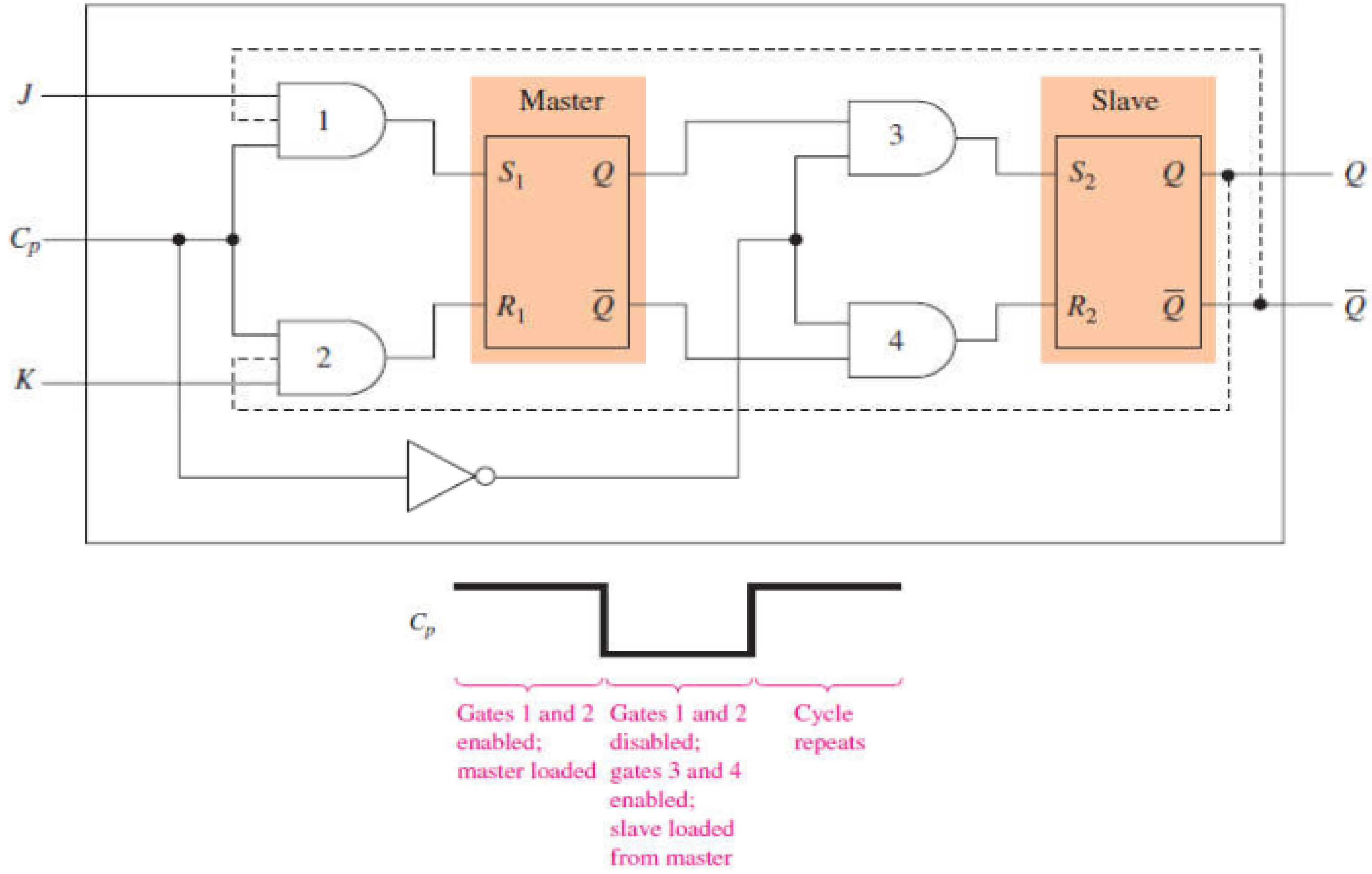
A Clock Pulse

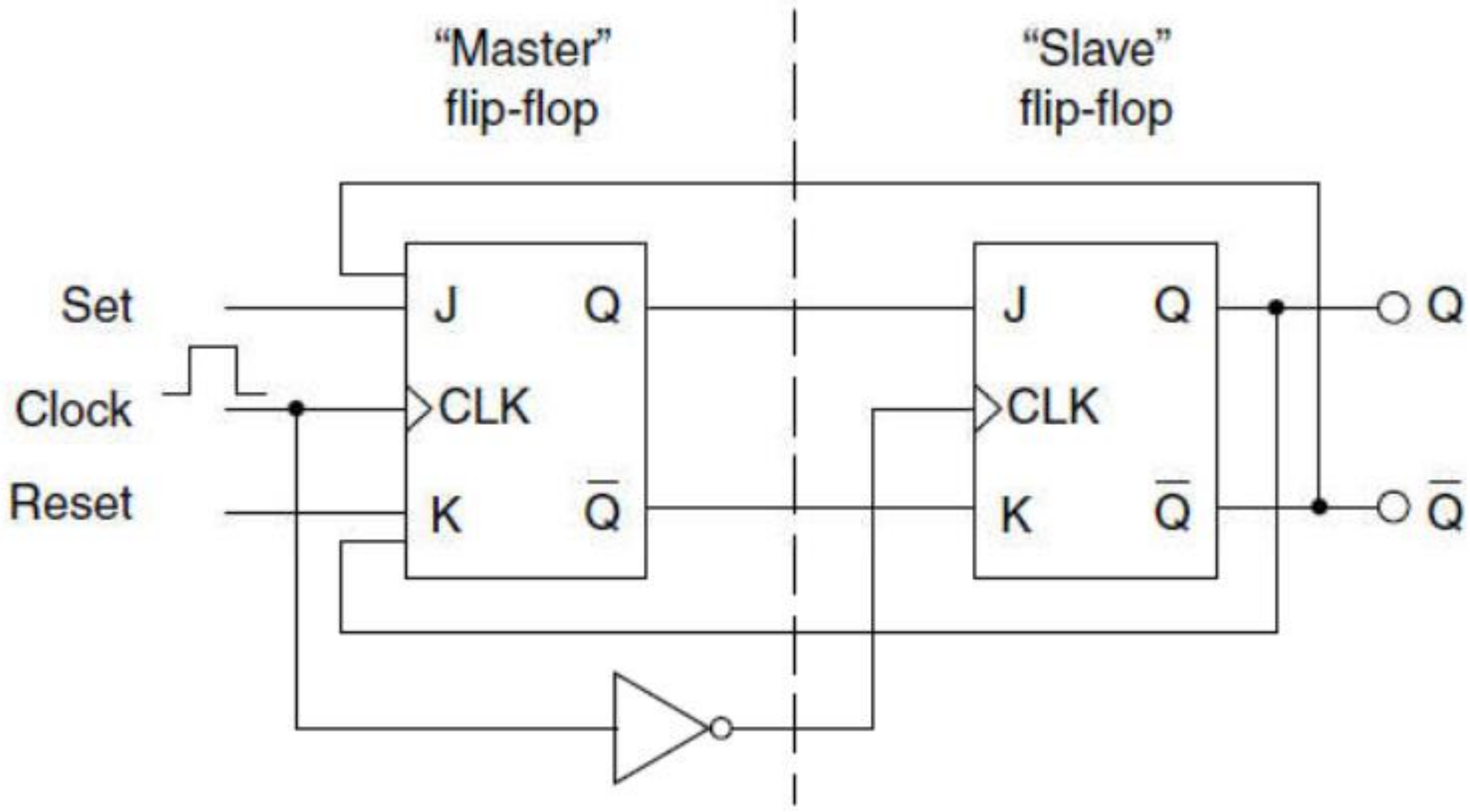
Elimination of Race Around Condition

$$t_p < \Delta t < T$$

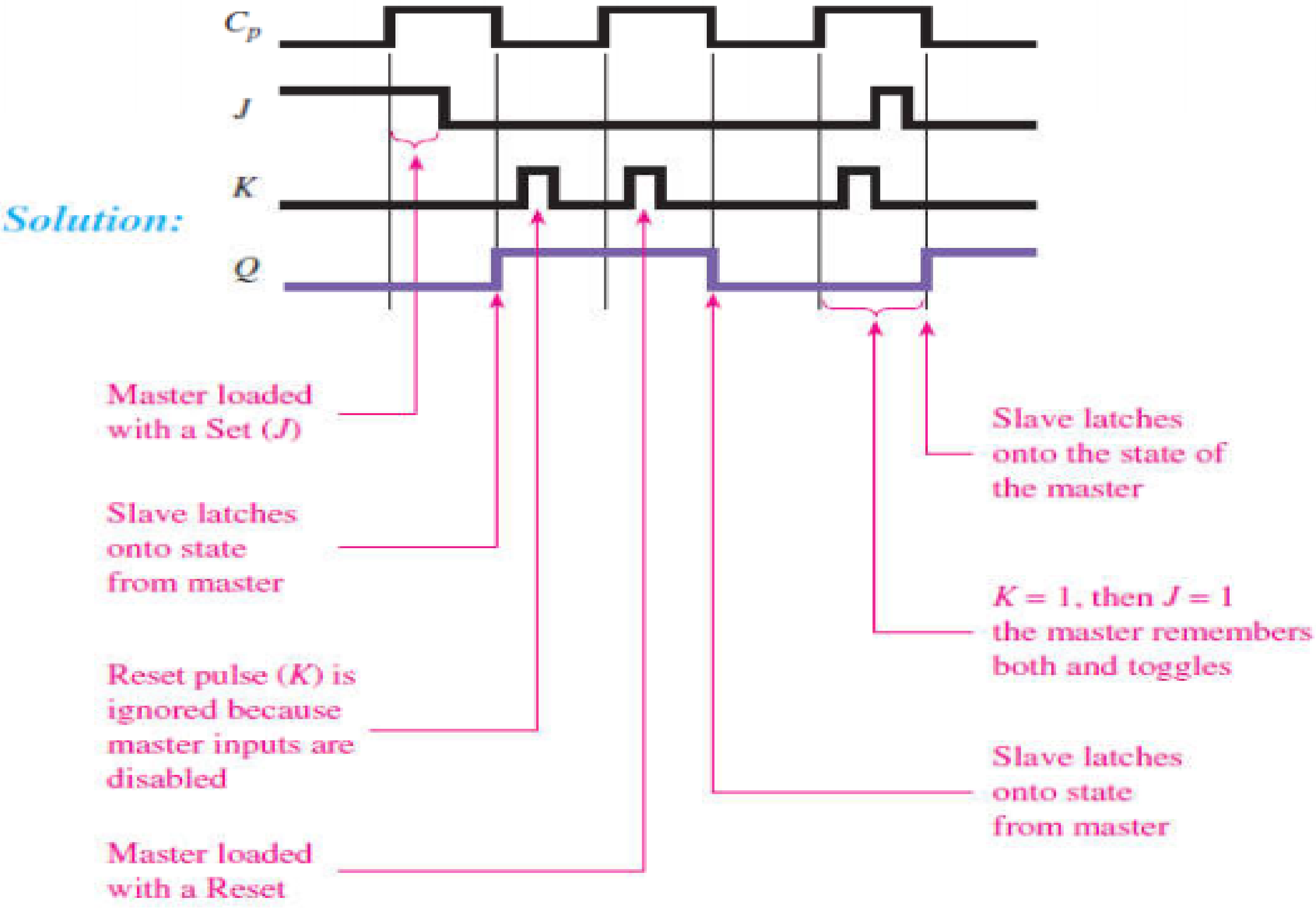
- Pulse width should be less than Propagation Delay
- Use of Master Slave Configuration

Master Slave Arrangement





"Master"
flip-flop
"Slave"
flip-flop
Set
Clock
Reset
J
CLK
K
Q
$\overline{Q}$
J
CLK
K
Q
$\overline{Q}$
Q
$\overline{Q}$



Solution:
C_p
J
K
Q
Master loaded with a Set (J)
Slave latches onto state from master
Reset pulse (K) is ignored because master inputs are disabled
Master loaded with a Reset
Slave latches onto the state of the master
K = 1, then J = 1 the master remembers both and toggles
Slave latches onto state from master

Excitation Tables

S	R	Present state Q_n	Next state Q_{n+1}
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	x
1	1	1	x

Invalid states

Truth table of SR flip flop

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

Excitation table of SR flip flop

D	Present state Q_n	Next state Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

Truth table of D flip flop

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

Excitation table of D flip flop

J	K	Present state Q_n	Next state Q_{n+1}
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

Truth table of JK flip flop

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

Excitation table of JK flip flop

T	Present state Q_n	Next state Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Truth table of T flip flop

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

Excitation table of T flip flop

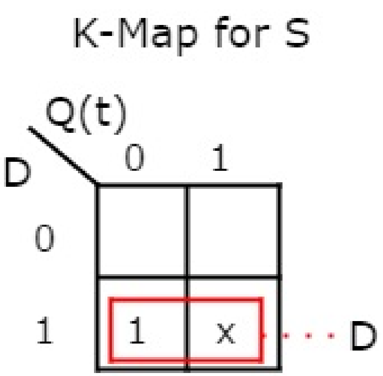
Present State	Next State	S-R FF		J-K FF		T-FF	D-FF
		S_n	R_n	J_n	K_n	T_n	D_n
0	0	0	×	0	×	0	0
0	1	1	0	1	×	1	1
1	0	0	1	×	1	1	0
1	1	×	0	×	0	0	1

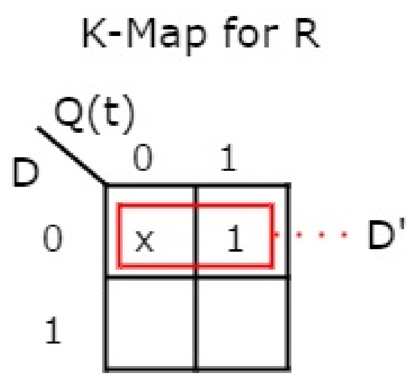
Steps for converting one flip-flop to the other:

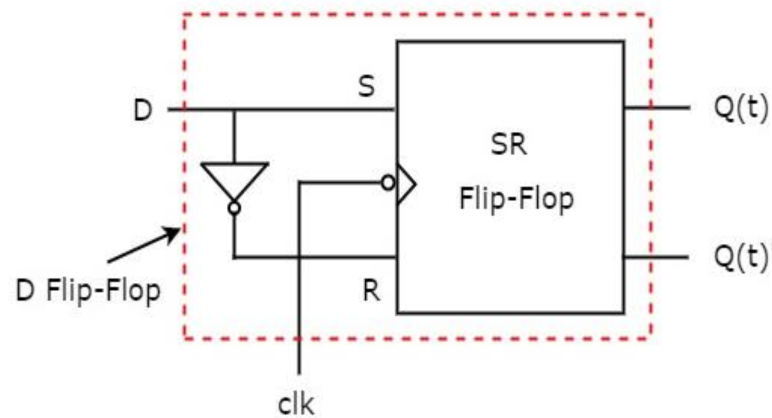
- Consider the **characteristic table** of desired flip-flop.
- Fill the excitation values inputs of given flip-flop for each combination of present state and next state.
- Get the simplified expressions for each excitation input. If necessary, use K-maps for simplifying.
- Draw the circuit diagram of desired flip-flop according to the simplified expressions using given flip-flop and necessary logic gates.

SR flip-flop to D flip-flop conversion

D flip-flop input	Present State	Next State	SR flip-flop inputs	
D	Q t	Q $t+1$	S	R
0	0	0	0	x
0	1	0	0	1
1	0	1	1	0
1	1	1	x	0





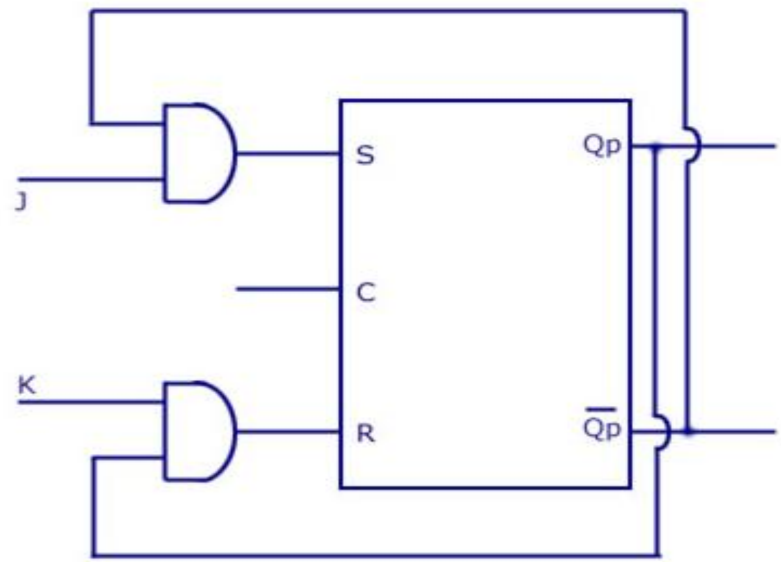


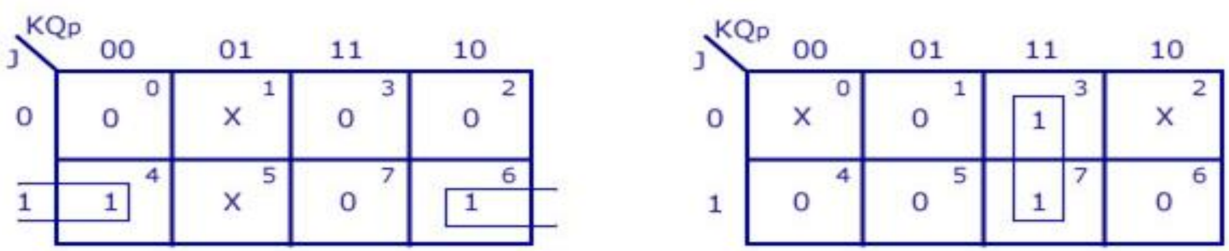
SR to JK flip-flop

Conversion Table

J-K Inputs		Outputs		S-R Inputs	
J	K	Qp	Qp+1	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

Logic Diagram





$S = \overline{J}Q_p$

$R = KQ_p$

K-Map

- JK to SR flip-flop
- JK to D FF
- JK to T FF
- SR to D FF
- SR to T FF
- SR to JK
- D to SR FF
- D to JK FF
- D to T FF
- T to SR FF
- T to JK FF
- T to D FF