UTM UNIVERSITI TEKNOLOGI MALAYSIA

SCHOOL OF ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING

UTM UNIVERSITI TEKNOLOGI MALAYSIA

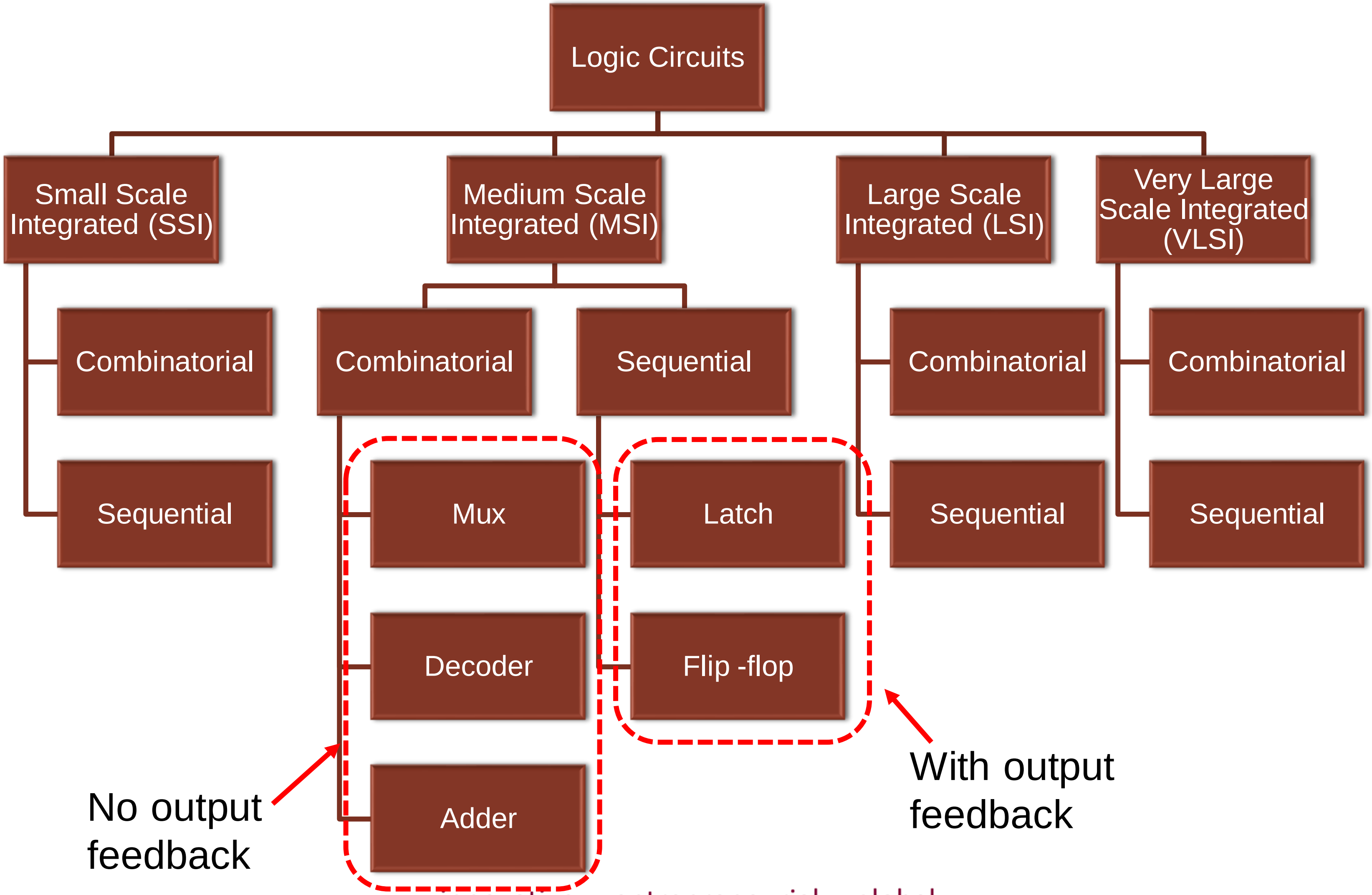
SEEE 1223 DIGITAL ELECTRONICS

CHAPTER 7: LATCHES AND FLIP FLOPS

DR. MOHD SAIFUL AZIMI BIN MAHMUD
P19a-04-03-30
School of Electrical Engineering
Faculty of Engineering
Universiti Teknologi Malaysia
019-7112948
azimi@utm.my

innovative • entrepreneurial • global

COMBINATIONAL VS SEQUENTIAL LOGIC



Logic Circuits

Small Scale Integrated (SSI)

Combinatorial

Sequential

Medium Scale Integrated (MSI)

Combinatorial

Mux

Decoder

Adder

Sequential

Latch

Flip -flop

Large Scale Integrated (LSI)

Combinatorial

Sequential

Very Large Scale Integrated (VLSI)

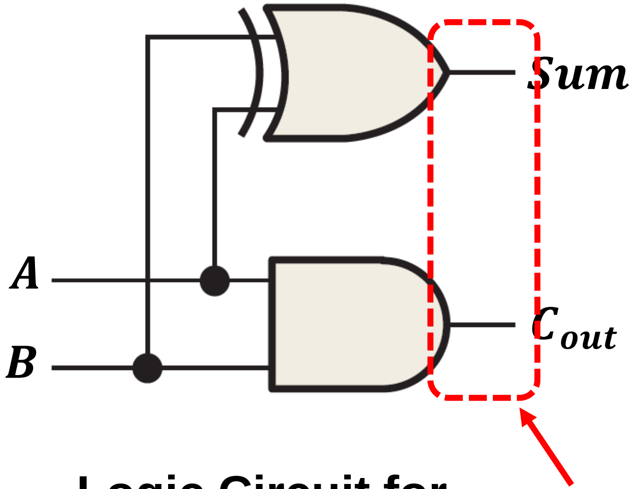
Combinatorial

Sequential

No output feedback

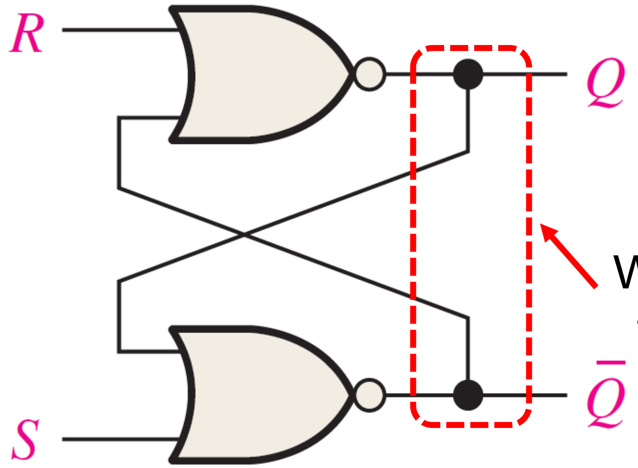
With output feedback

COMBINATIONAL VS SEQUENTIAL LOGIC



Logic Circuit for Half Adder

No output feedback



With output feedback

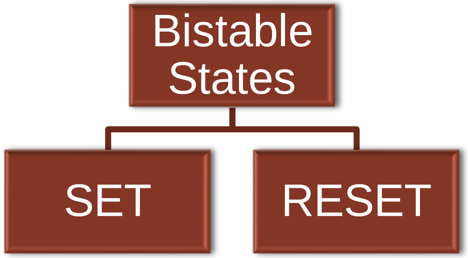
Logic Circuit for Active-HIGH Input *S*-*R* Latch

LATCHES

LATCHES

INTRODUCTION

- **Latch** is a type of **temporary storage device** that has **two stable states (bistable)**.
- It is a basic form of memory, i.e store value of **0** and **1** in a latch.
- Latches are similar to flip-flops because they are **bistable** devices that can reside in either of two states using a feedback arrangement. In which the outputs are connected back to the opposite inputs.
- Latches however, are considered unstable in modern circuits and rarely used.
- Flip-flops are the dominant sequential circuit element and are present in almost all digital system.



Bistable States

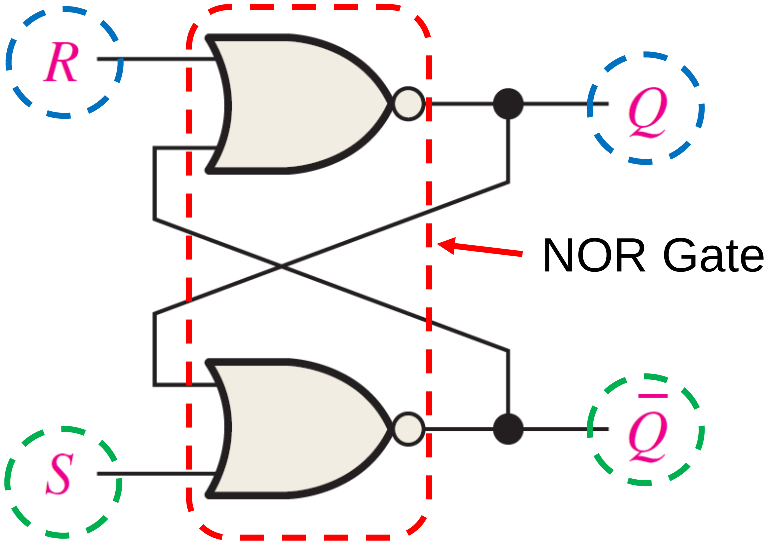
SET

RESET

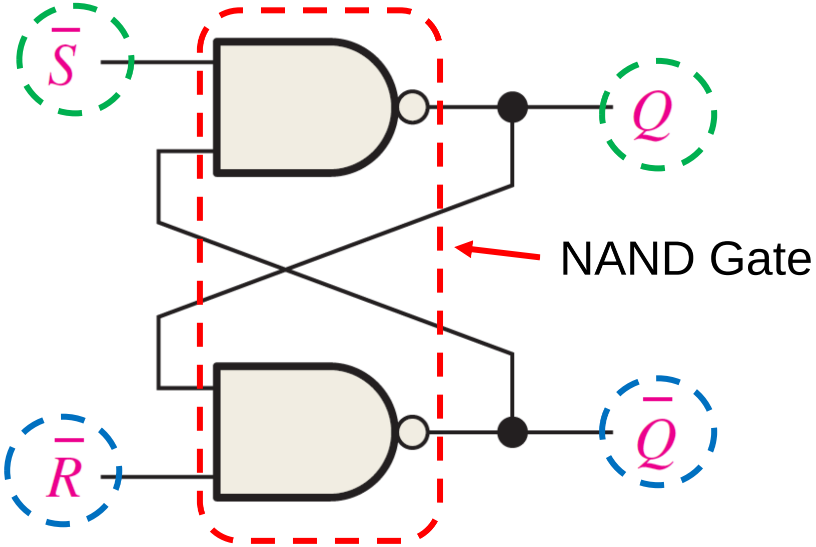
LATCHES

SR LATCH

- **The SR Latch (Set-Reset Latch)** is the most basic type, which can be constructed using NOR or NAND gates.
- An **Active-HIGH input SR Latch** is formed with **two cross-couple NOR gate**.
- An **Active-LOW input $\overline{S}\overline{R}$ Latch** is formed with **two cross-couple NAND gate**.



Active-HIGH Input S-R Latch

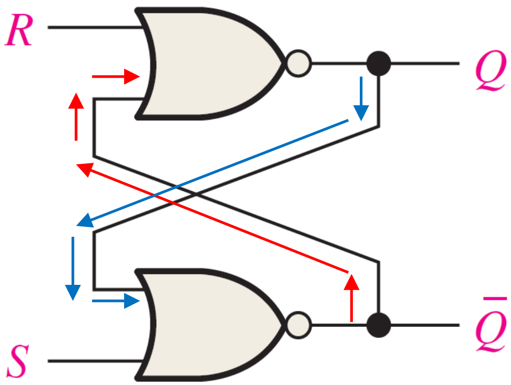


Active-LOW Input $\overline{S}$-$\overline{R}$ Latch

LATCHES

ACTIVE-HIGH SR LATCH

- The Active-High SR Latch has **two inputs S** and **R**, which will let us control the outputs $\boldsymbol{Q}$ and $\boldsymbol{\bar{Q}}$.



- Here Q and $\bar{Q}$ **feed back** into the circuit. They are not the only outputs, they also are the inputs.
- To figure out how Q and $\bar{Q}$ change, we have to look at not only the inputs S and R, but also the current values of Q and $\bar{Q}$ are:

$$\boldsymbol{Q_{next}} = \overline{(\boldsymbol{R} + \overline{\boldsymbol{Q_{current}}})}$$

$$\overline{\boldsymbol{Q_{next}}} = \overline{(\boldsymbol{S} + \boldsymbol{Q}\ \boldsymbol{_{current}})}$$

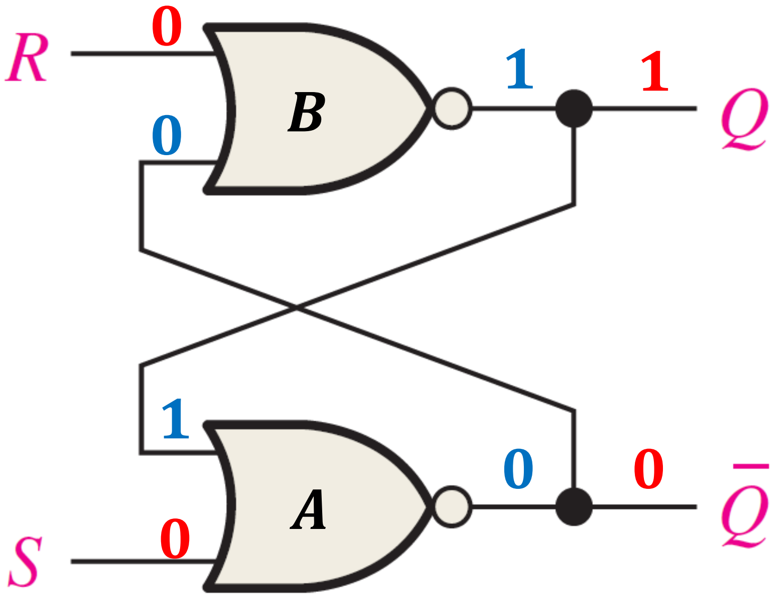
LATCHES

ACTIVE-HIGH SR LATCH: STORING A VALUE: SR = 00

- What if $\boldsymbol{S = 0}$ and $\boldsymbol{R = 0}$?
- The equations on the right reduce to:

$$\boldsymbol{Q_{next} = \overline{(R + \overline{Q_{current}})}}$$
$$\boldsymbol{= \overline{(0 + \overline{Q_{current}})} = Q_{current}}$$

$$\boldsymbol{\overline{Q_{next}} = \overline{(S + Q_{current})}}$$
$$\boldsymbol{= \overline{(0 + Q_{current})} = \overline{Q_{current}}}$$



- So when $SR = 00$, then $Q_{next} = Q_{current}$. Whatever value of $Q_{current}$ has, Q_{next} keeps.
- This is exactly what how the values are **stored** inside the latch.

LATCHES

ACTIVE-HIGH SR LATCH: SETTING THE LATCH: SR = 10

- What if **S =1** and **$R = 0$**?
- Since S =1, $\overline{Q_{next}}$ is 0, regardless of $Q_{current}$:

$$\overline{Q_{next}} = \overline{(S + Q_{\ current})} = \overline{(1 + Q_{current})} = \overline{1} = 0$$

- Then, this new value of $\bar{Q}$ goes into the top NOR gate, along with $R = 0$.

$$Q_{next} = \overline{(R + \overline{Q_{current}})} = \overline{(0 + 0)} = 1$$

- So when $SR = 10$, then $\overline{Q_{next}} = 0$ and $Q_{next} = 1$.
- This is how you **set** the latch to 1. The S input stands for **"SET"**.
- Notice that it can take up to **two steps (two gate delays)** from the time S becomes 1 to the time Q_{next} becomes 1.
- But once Q_{next} becomes 1, the outputs will stop changing. This is a **stable state**.

LATCHES

ACTIVE-HIGH SR LATCH: RESETTING THE LATCH: SR = 01

- What if $\boldsymbol{S = 0}$ and $\boldsymbol{R = 1}$?
- Since R =1, Q_{next} is 0, regardless of $Q_{current}$:

$$\boldsymbol{Q_{next}} = \overline{(\mathbf{1} + \overline{\boldsymbol{Q_{current}}})} = \overline{\mathbf{1}} = \mathbf{0}$$

- Then, this new value of Q goes into the bottom NOR gate, along with $S = 0$.

$$\overline{\boldsymbol{Q_{next}}} = \overline{(\boldsymbol{S} + \boldsymbol{Q}\ \boldsymbol{_{current}})} = \overline{(\mathbf{0} + \mathbf{0})} = \mathbf{1}$$

- So when $SR = 01$, then $Q_{next} = 0$ and $\overline{Q_{next}} = 1$.
- This is how you **reset**, or **clear**, the latch to 0. The R input stands for **“reset”**.
- Again, it can take two gate delays before a change in R propagates to the output $\overline{Q_{next}}$.

LATCHES

ACTIVE-HIGH SR LATCH: WHAT ABOUT: SR = 11

- Both $\boldsymbol{Q_{next}}$ and $\overline{\boldsymbol{Q_{next}}}$ will become **0**.
- This contradicts the assumption that Q and $\bar{Q}$ are always complements.
- Another problem is what happens if we then make $S = 0$ and $R = 0$ together.

$$\boldsymbol{Q_{next}} = \overline{(\mathbf{0} + \mathbf{0})} = \mathbf{1}$$

$$\overline{\boldsymbol{Q_{next}}} = \overline{(\mathbf{0} + \mathbf{0})} = \mathbf{1}$$

- But these new values go back into the NOR gates, and in the next step we get:

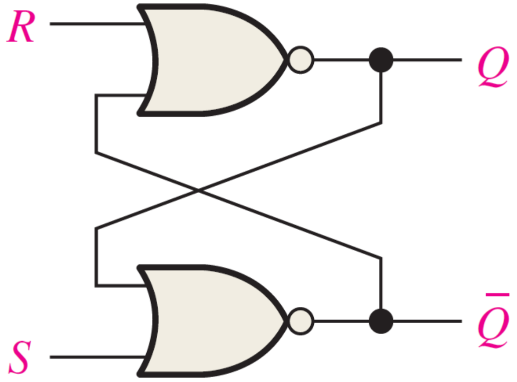
$$\boldsymbol{Q_{next}} = \overline{(\mathbf{0} + \mathbf{1})} = \mathbf{0}$$

$$\overline{\boldsymbol{Q_{next}}} = \overline{(\mathbf{0} + \mathbf{1})} = \mathbf{0}$$

- The circuit enters an **infinite loop**, where Q and $\bar{Q}$ cycle between 0 and 1 forever.
- This is actually the worst case, but the moral is don’t ever set $SR =$ **11!**

LATCHES

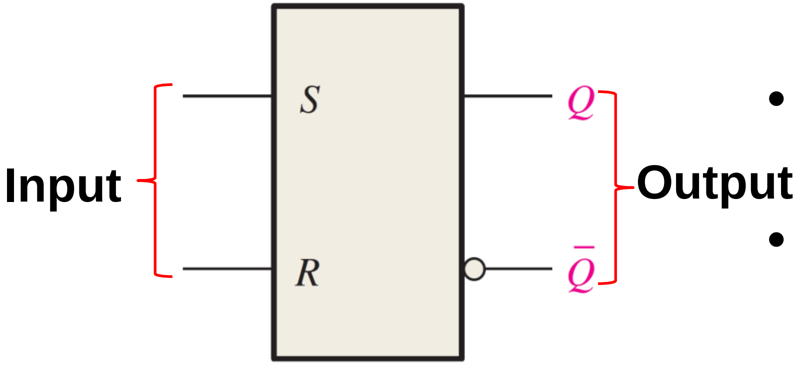
ACTIVE-HIGH SR LATCH



Active-High SR Latch Logic Circuit

Active-High SR Latch Truth Table

Inputs		Outputs		Action
S	R	$Q(next)$	$\overline{Q(next)}$	
0	0	Q	$\bar{Q}$	NO CHANGE
0	1	0	1	RESET
1	0	1	0	SET
1	1	0	0	FORBIDDEN

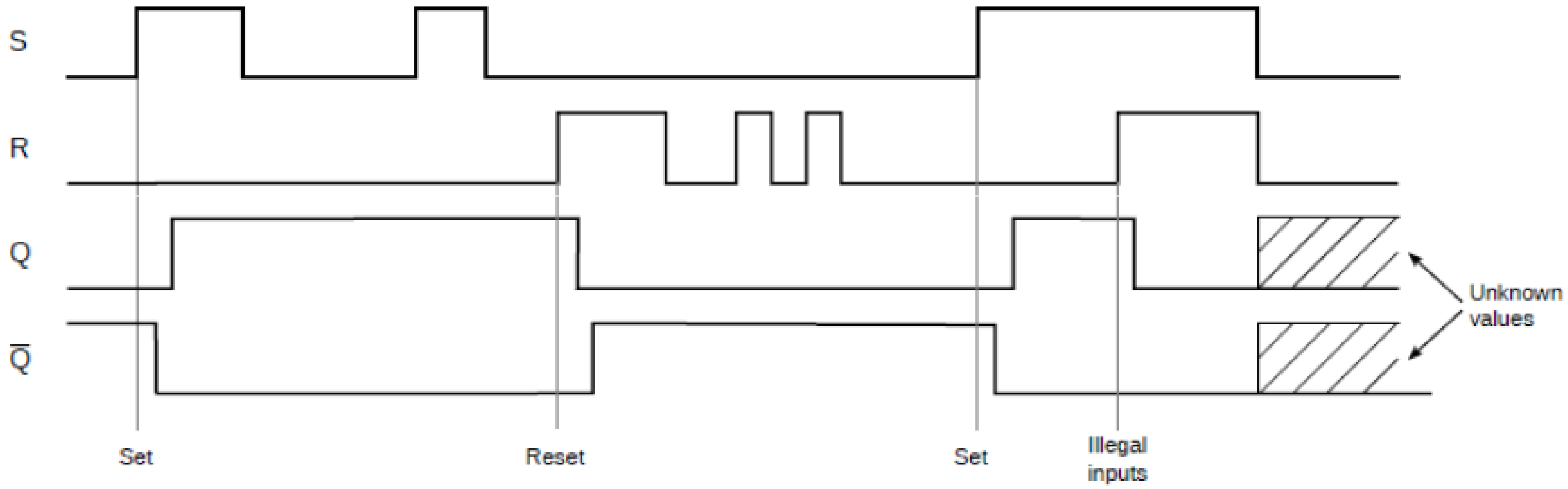


Active-High SR Latch Logic Symbol

- When $S = 0$ and $R = 0$, Q and $\bar{Q}$ **maintains the previous value**.
- When $S = 0$ and $R = 1$, the latch is **RESET** $(Q = 0)$.
- When $S = 1$ and $R = 0$, the latch is **SET** $(Q = 1)$.
- When $S = 1$ and $R = 1$, **both Q and $\bar{Q}$ are 0**, which is **invalid**.

LATCHES

ACTIVE-HIGH SR LATCH



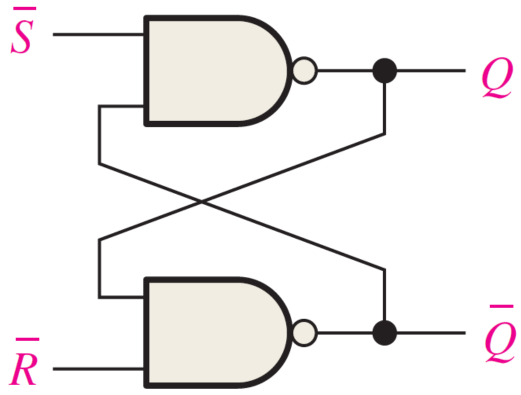
Timing Diagram for SR Latch (with propagation delay)

LATCHES

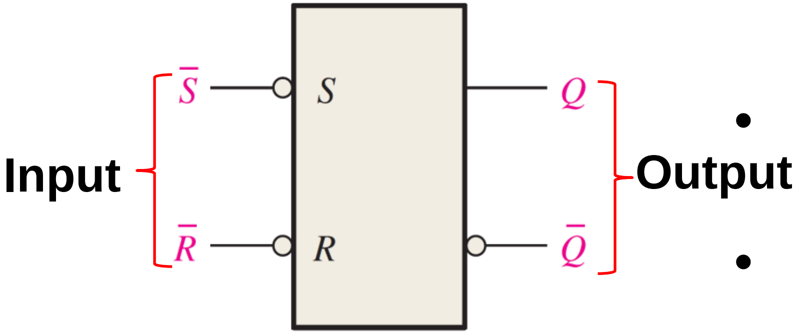
ACTIVE-LOW SR LATCH ($\overline{S}\overline{R}$ LATCH)

- To get an **Active-Low SR Latch ($\overline{S}\overline{R}$ Latch)**, NAND gates are used.

$\overline{S}\overline{R}$ Latch Logic Circuit



$\overline{S}\overline{R}$ Latch Logic Symbol



$\overline{S}\overline{R}$ Latch Truth Table

Inputs		Outputs		Action
$\overline{S}$	$\overline{R}$	$Q(next)$	$\overline{Q(next)}$	
0	0	1	1	FORBIDDEN
0	1	1	0	SET
1	0	0	1	RESET
1	1	Q	$\overline{Q}$	NO CHANGE

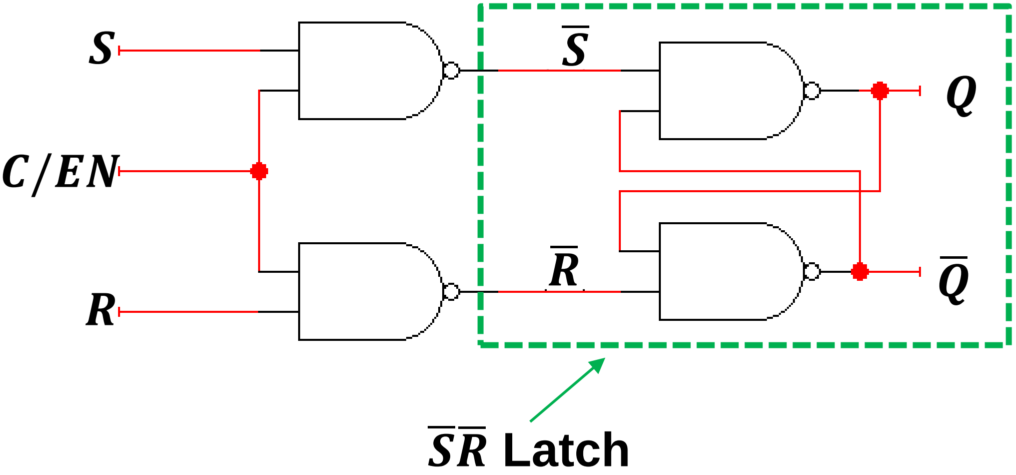
- When $\boldsymbol{S = 0}$ and $\boldsymbol{R = 1}$, the latch is **SET ($\boldsymbol{Q = 1}$)**.
- When $\boldsymbol{S = 1}$ and $\boldsymbol{R = 0}$, the latch is **RESET ($\boldsymbol{Q = 0}$)**.
- When $\boldsymbol{S = 1}$ and $\boldsymbol{R = 1}$, **$\boldsymbol{Q}$ and $\boldsymbol{\overline{Q}}$ maintains the previous value**.
- When $\boldsymbol{S = 0}$ and $\boldsymbol{R = 0}$, **both $\boldsymbol{Q}$ and $\boldsymbol{\overline{Q}}$ are 1**, which is invalid.

LATCHES

GATED SR LATCH

- **Gated SR Latch** is a SR Latch with control input, **C/EN** to enable or disable S and R inputs.

Gated SR Latch Logic Circuit

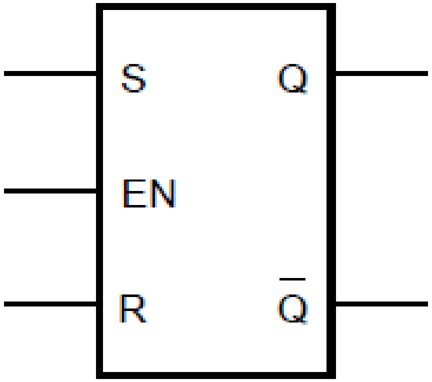


Gated SR Latch Truth Table

EN	S	R	$\bar{S}$	$\bar{R}$	Q_{next}
0	X	X	1	1	NO CHANGE
1	0	0	1	1	NO CHANGE
1	0	1	1	0	0 (RESET)
1	1	0	0	1	1 (SET)
1	1	1	0	0	FORBIDDEN

- The additional NAND gates are simply used to generate the correct inputs for the $\bar{S}\bar{R}$ Latch.
- The control input acts just like an enable.

Gated SR Latch Logic Symbol

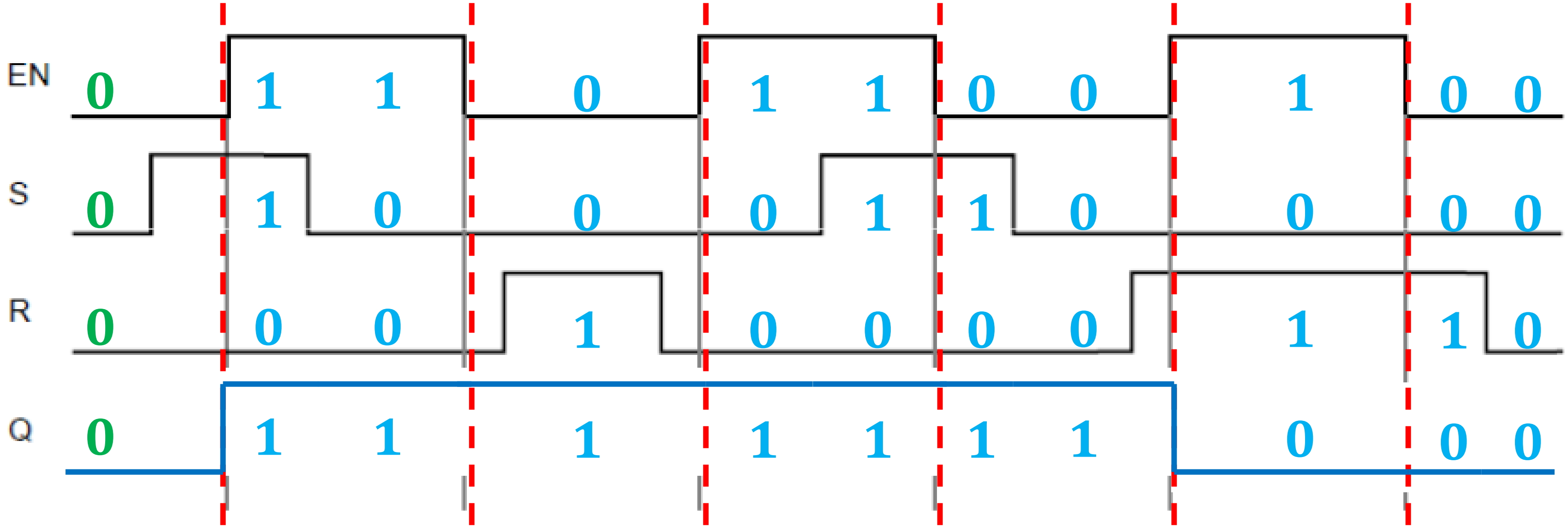


LATCHES

GATED SR LATCH

Example

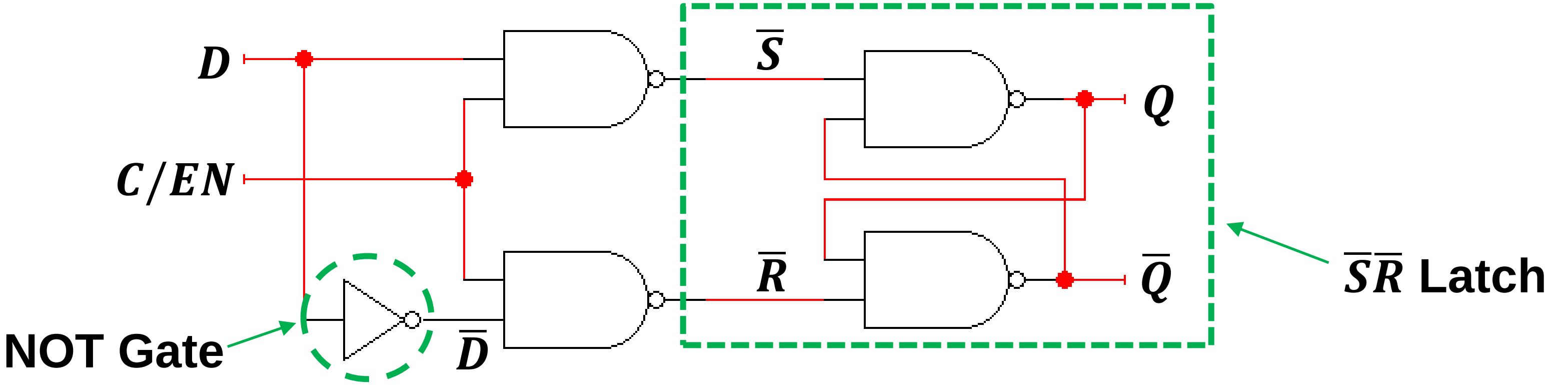
Draw the output Q for the gated SR Latch with Q is initially LOW.



LATCHES

GATED D LATCH

- **Gated D Latch** is based on an $\bar{S}\bar{R}$ Latch. The additional gates generate the $\bar{S}$ and $\bar{R}$ signals, based on inputs **D ("data")** and **EN ("enable")**.
  - When $\boldsymbol{EN = 0}$, $\bar{S}$ and $\bar{R}$ are both 1, so the state Q does not change.
  - When $\boldsymbol{EN = 1}$, the latch output Q will equal the input D.
- No more messing with one input for set and another input for reset!



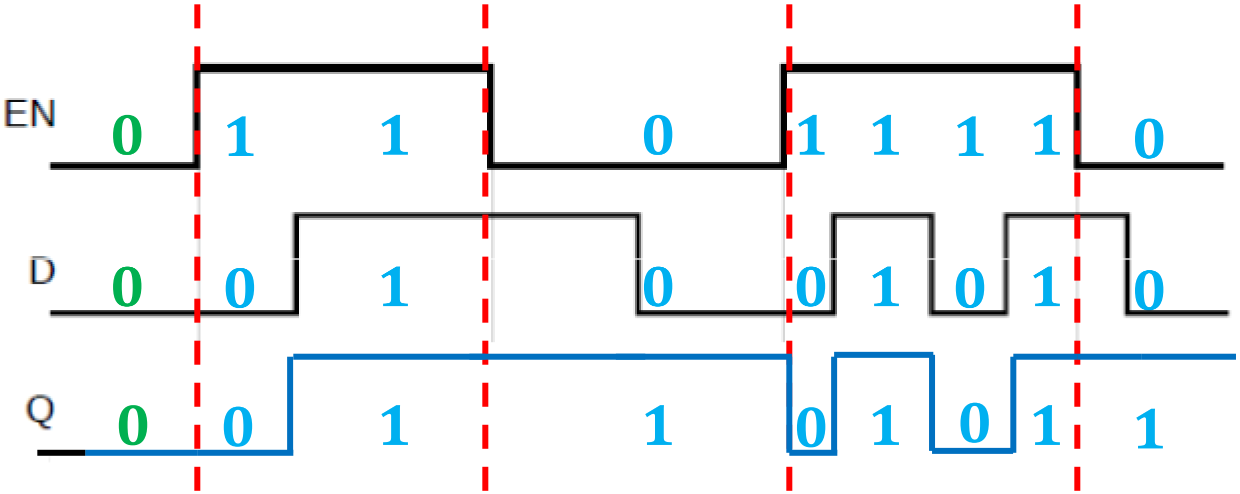
- Also, this latch has no **"bad"** input combinations to avoid. Any of the four possible assignments to C and D are valid.

LATCHES

GATED D LATCH

Example

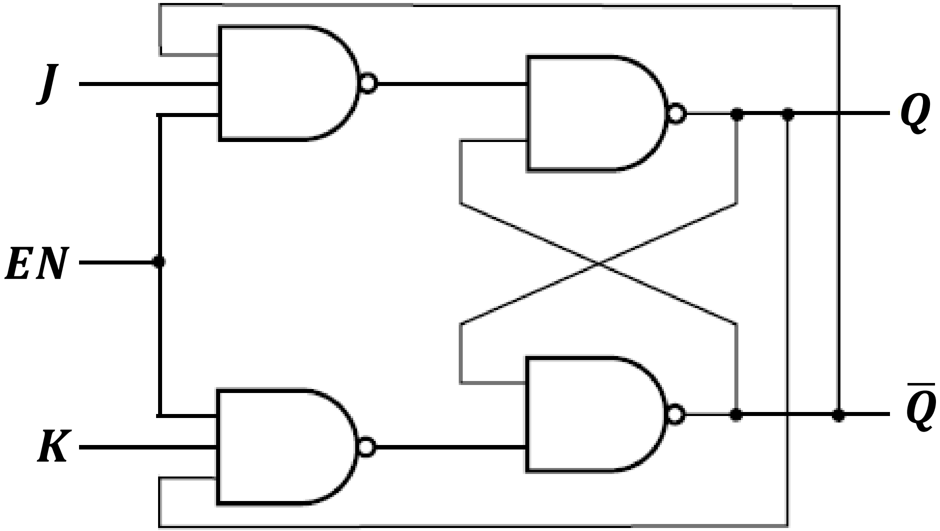
Draw the output Q for the Gated D Latch.



LATCHES

GATED JK LATCH

- **Gated JK Latch** is another way to improve the gated SR Latch.
- The input of **J** and **K** performs exactly the same as **S** and **R**, with the **exception** of the condition of $\boldsymbol{JK = 11}$.
- When J and K are both HIGH ($JK = 11$), the output **toggles** (switch from 0 to 1 or vice versa).

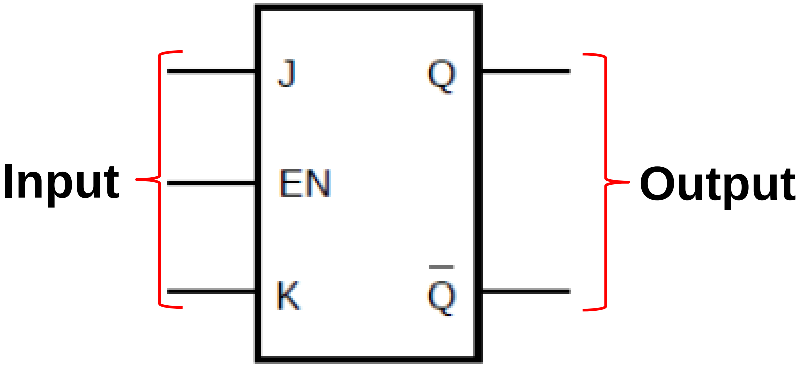


Gated JK Latch Logic Circuit

LATCHES

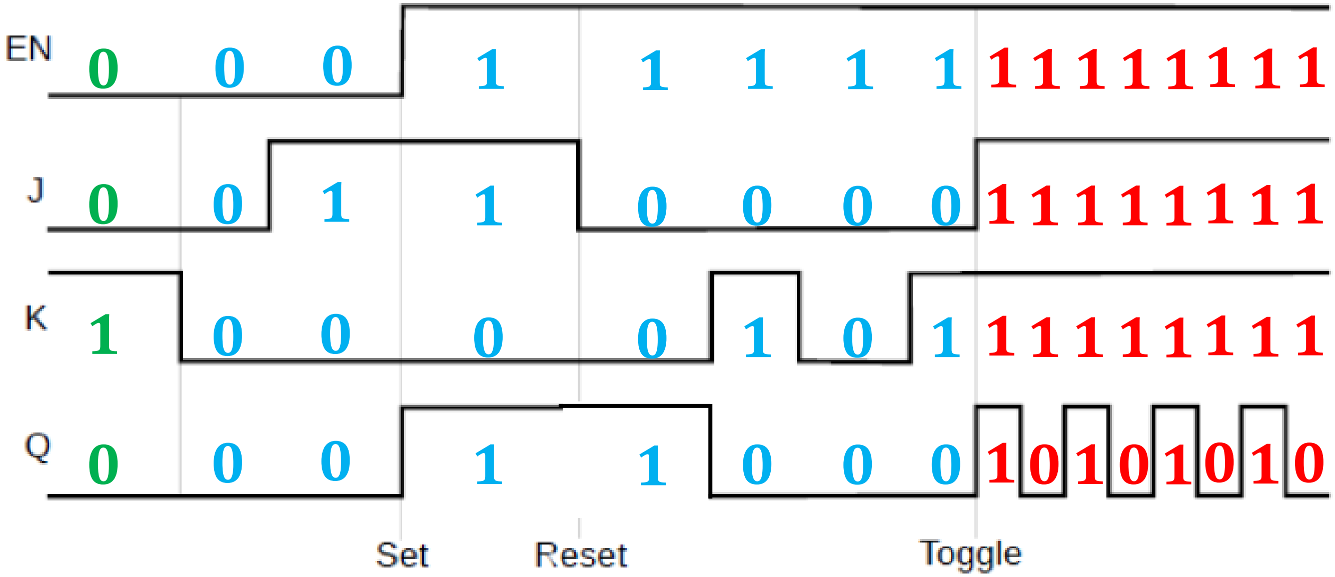
GATED JK LATCH

Gated JK Latch Symbol



Gated JK Latch Truth Table

EN	J	K	Q_{next}	Action
0	X	X	Q	NO CHANGE
1	0	0	Q	NO CHANGE
1	0	1	0	0 (RESET)
1	1	0	1	1 (SET)
1	1	1	$\bar{Q}$	TOGGLE

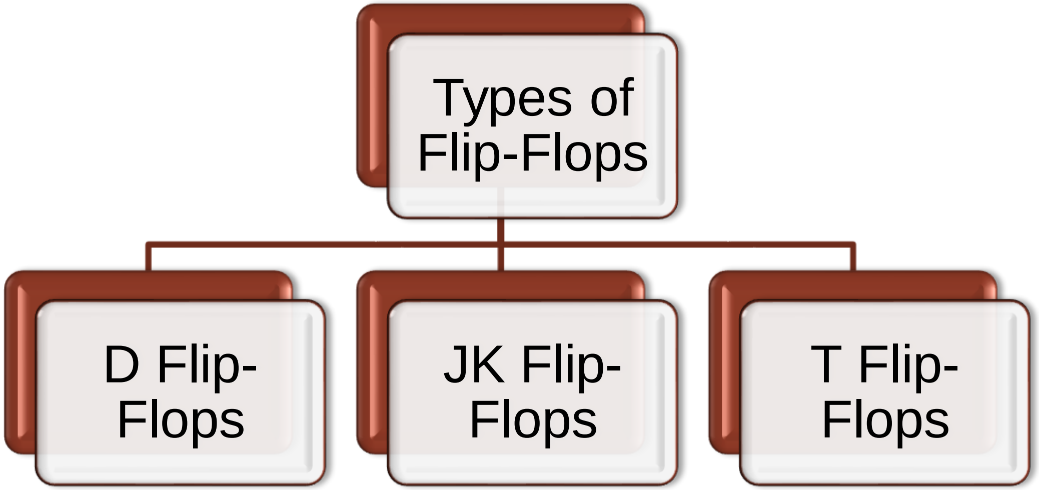


FLIP-FLOPS

FLIP-FLOPS

INTRODUCTION

- **Flip-flops** are synchronous bistable devices.
- The term **synchronous** means that the output changes state only at a specified point on the triggering input called the **clock (CLK)**.
- This **CLK** is designated as a control input, C (in latches); that is, changes in the output occur in synchronization with the **clock**.



Types of Flip-Flops

D Flip-Flops

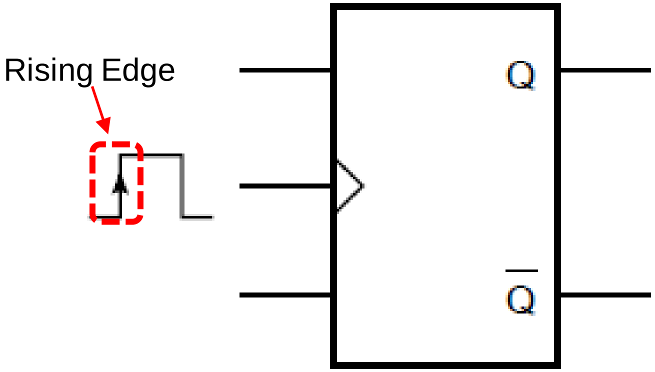
JK Flip-Flops

T Flip-Flops

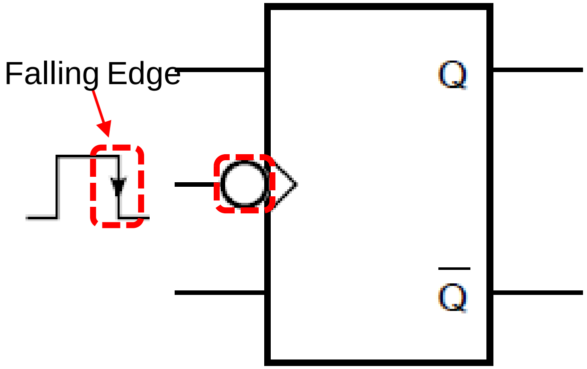
FLIP-FLOPS

EDGE TRIGGER

- An **edge-triggered flip-flops** change state either at the **positive edge** (rising edge) or **negative edge** (falling edge) of the clock pulse.
- It is **sensitive to its inputs only at this transition of the clock**.
- Clock inputs of flip-flops are symbolized by a triangle in logic symbols.
- **Positive edge** triggered no bubble at CLK input.
- **Negative edge** triggered has bubble at CLK input.



Positive edge triggered device



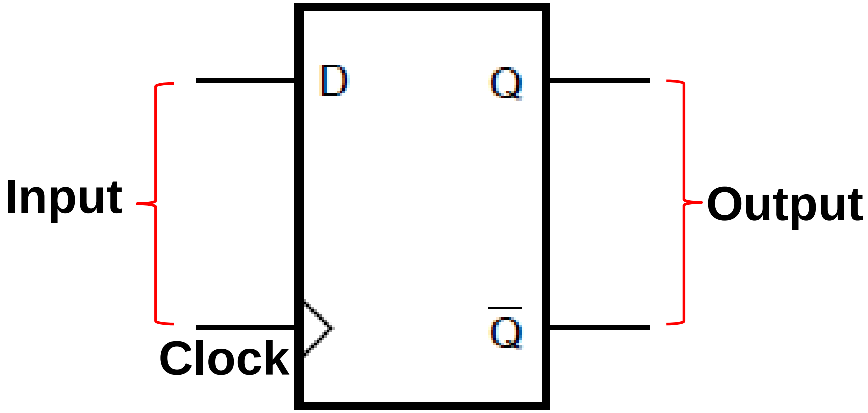
Negative edge triggered device

FLIP-FLOPS

D FLIP-FLOP

- **D flip-flop** is useful when a single data bit **(1 or 0)** is to be stored.
- It works almost the same as the D latch.
- When a clock pulse arrives, the input is transferred to the output.
- The D flip-flop can be either positive edge or negative edge triggered.
- The positive edge triggered D flip-flop, the input only valid or seen during the clock rising edge.

Logic Symbol of Positive edge triggered D flip-flop



Characteristic Table of Positive edge triggered D flip-flop

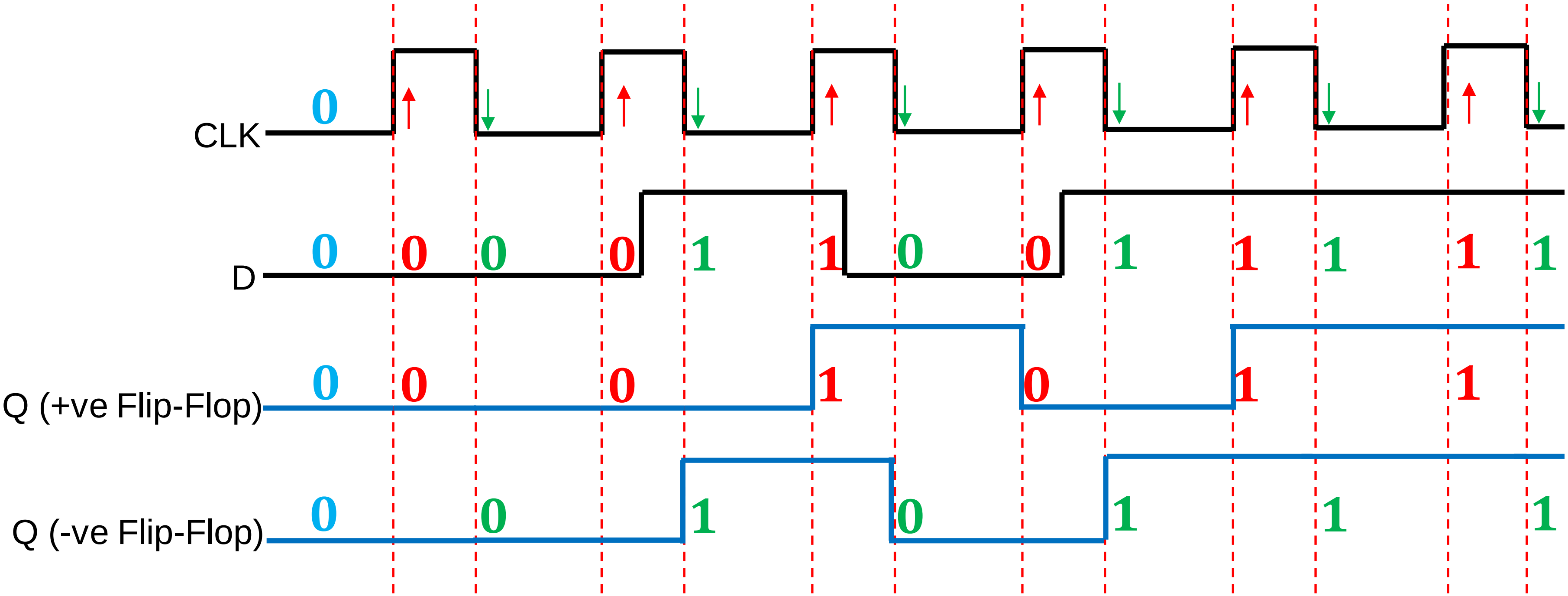
CLK	*D*	Q_{next}	*Action*
0	X	Q	NO CHANGE
1	X	Q	NO CHANGE
↑	0	0	0 (RESET)
↑	1	1	1 (SET)

FLIP-FLOPS

D FLIP-FLOP

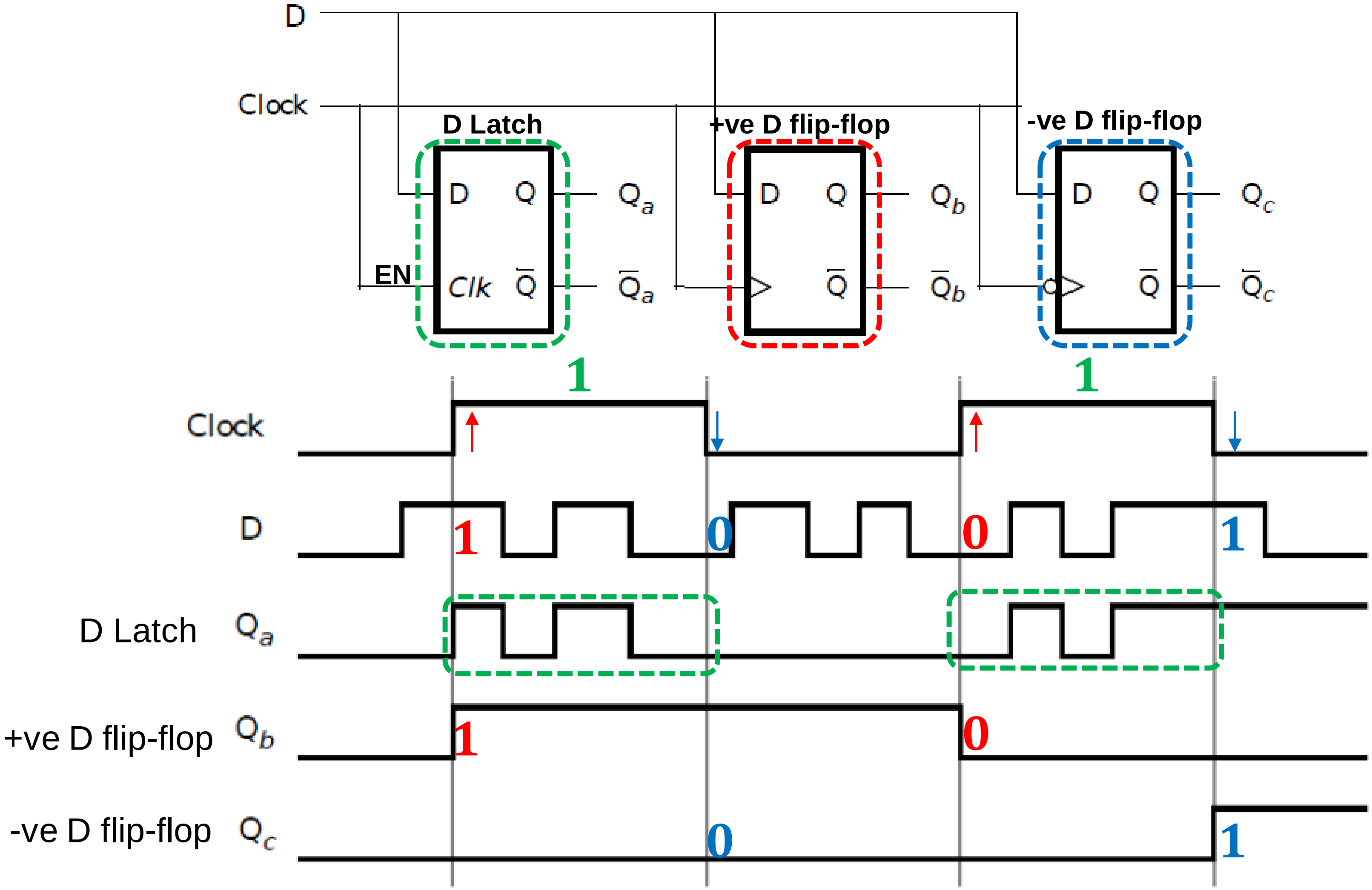
Example

Draw the output waveform of the positive edge triggered and negative edge triggered of the D flip-flop.



FLIP-FLOPS

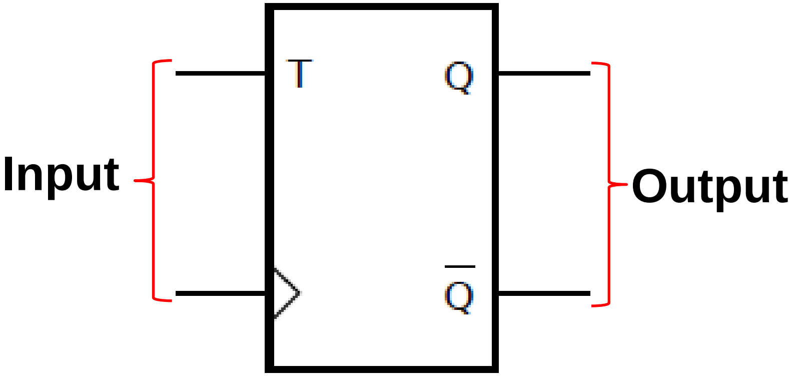
D LATCH VS D FLIP-FLOP



FLIP-FLOPS

T FLIP-FLOP

- **T (toggle) flip-flop** has two inputs: a clock and a T input.
- When T is **0**, clock pulses have no effect on the output.
- When T is **1**, when a clock pulse arrives, the **output toggles**.



Logic Symbol of Positive edge triggered T flip-flop

Characteristic Table of Positive edge triggered T flip-flop

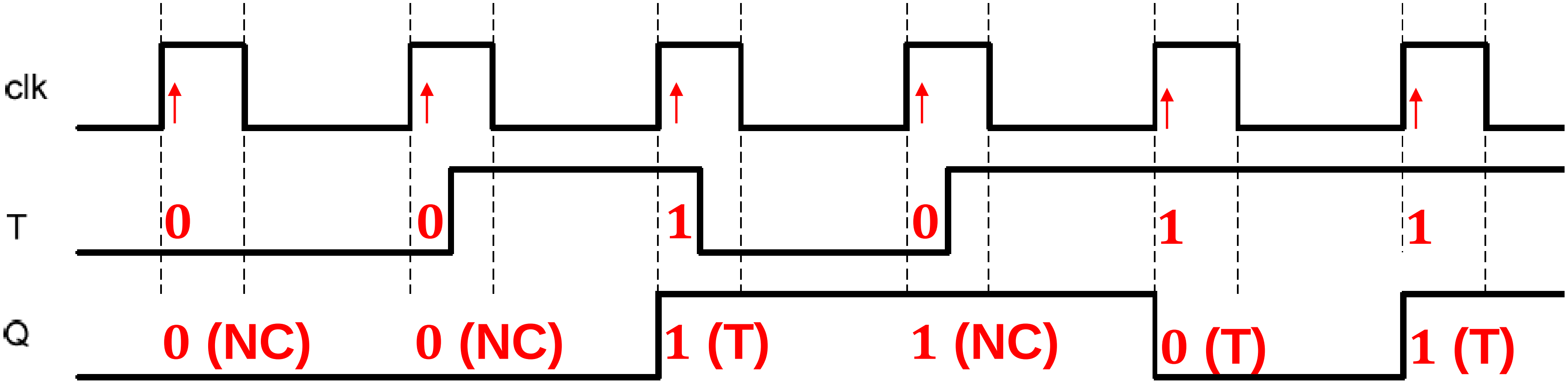
CLK	*T*	Q_{next}	*Action*
0	X	Q	NO CHANGE
1	X	Q	NO CHANGE
↑	0	Q	NO CHANGE
↑	1	$\bar{Q}$	TOGGLE

FLIP-FLOPS

T FLIP-FLOP

Example

Draw the output waveform of the positive edge triggered of the T flip-flop.

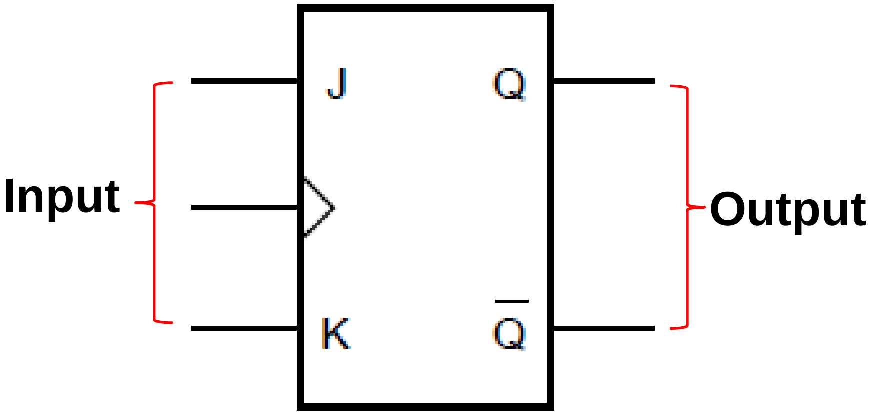


***NC: Not Change, T: Toggle**

FLIP-FLOPS

JK FLIP-FLOP

- **JK flip-flop** is versatile and is widely used type of flip-flop.
- It is identical to JK Latch where the output is toggled or inverted when $JK = 11$. But the input is changes during a clock edge.



Logic Symbol of Positive edge triggered JK flip-flop

Characteristic Table of Positive edge triggered JK flip-flop

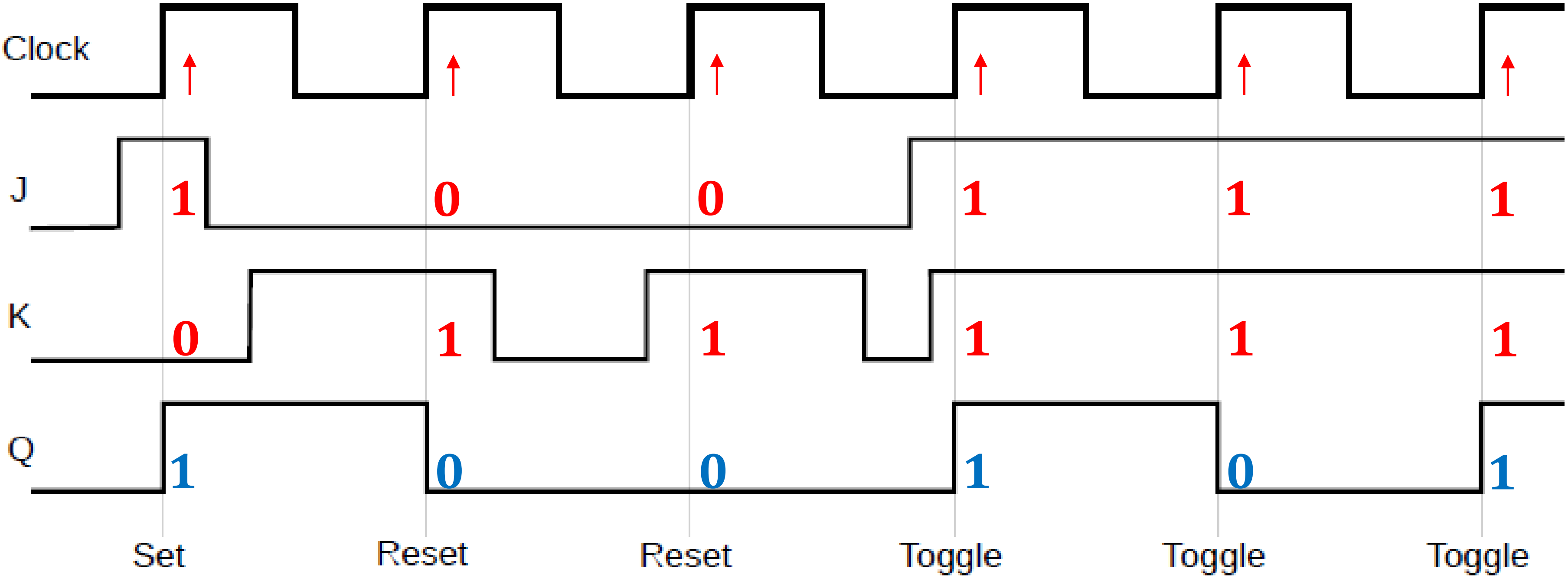
CLK	J	K	Q_{next}	$Action$
0	X	X	Q	NO CHANGE
1	X	X	Q	NO CHANGE
↑	0	0	Q	NO CHANGE
↑	0	1	0	RESET
↑	1	0	1	SET
↑	1	1	$\bar{Q}$	TOGGLE

FLIP-FLOPS

JK FLIP-FLOP

Example

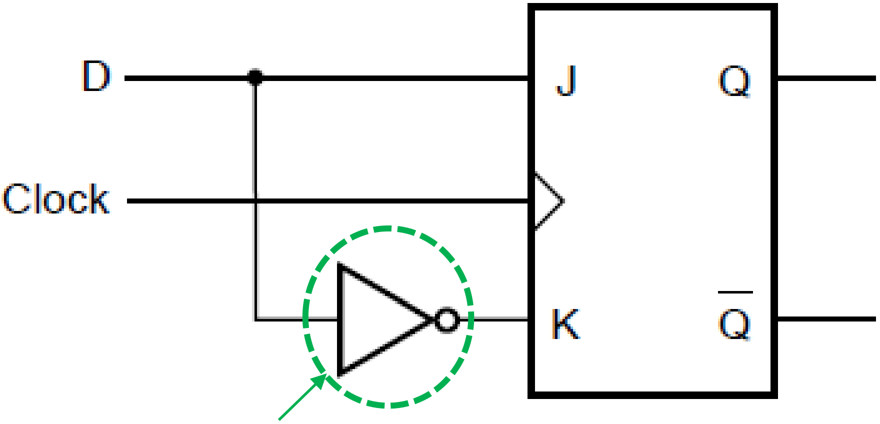
Draw the output waveform of the positive edge triggered of the JK flip-flop.



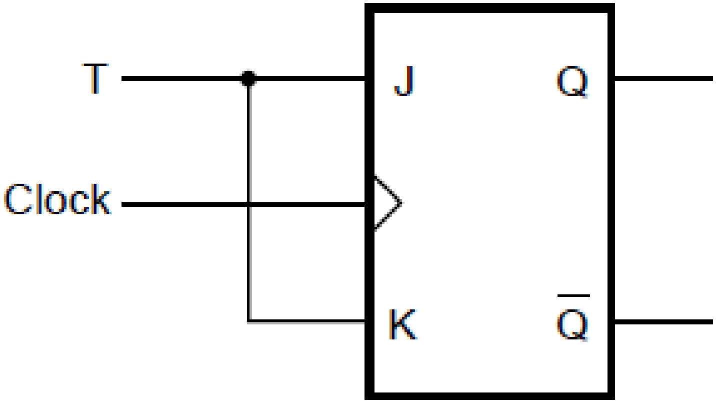
FLIP-FLOPS

JK FLIP-FLOP

- We have seen three types of flip-flops. These flip-flops can be implemented using JK flip-flop.
- To implement a **D flip-flop**, an inverter is placed between J and K inputs.



D flip-flop



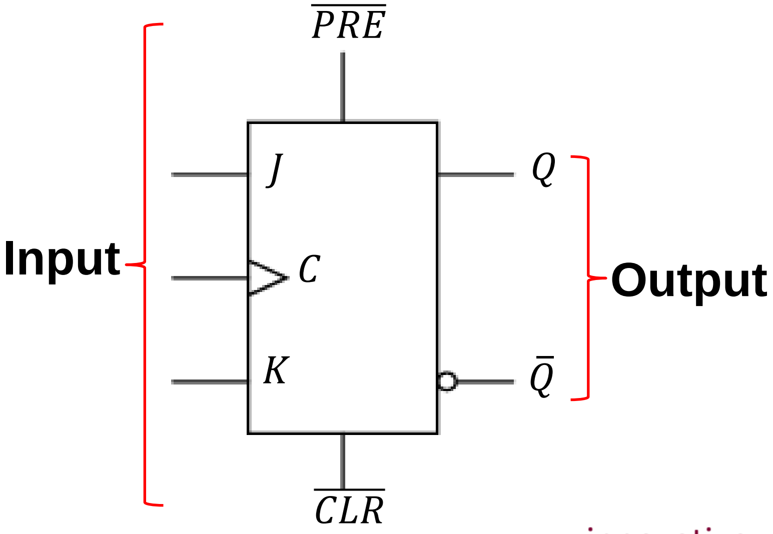
T flip-flop

- To implement a **T flip-flop**, connect both inputs of J and K together.

FLIP-FLOPS

ASYNCHRONOUS FLIP-FLOP

- Previously, we had seen **synchronous flip-flop** such as D, T and JK flip flops.
- **Synchronous flip-flop**: Input transferred on the triggered edge of the clock (data transfer synchronously with the clock).
- **Asynchronous flip-flop**: Input effect flip-flop state (output) independent of the clock.
- The asynchronous flip-flops normally labeled by **preset** $(\overline{PRE})$, direct SET and **clear** $(\overline{CLR})$, direct RESET.

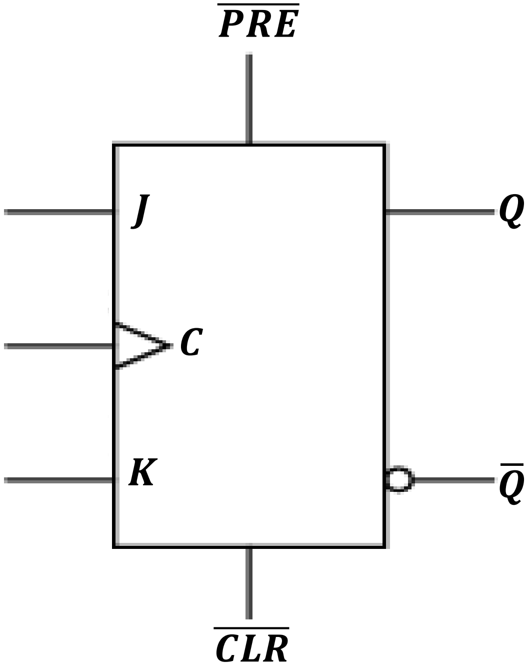


$\overline{PRE}$	$\overline{CLR}$	*FF State*	*Mode*
0	1	*SET*	Asynchronous
1	0	*RESET*	Asynchronous
1	1	*JK*	Synchronous

FLIP-FLOPS

ASYNCHRONOUS FLIP-FLOP

Example of characteristic table for asynchronous JK flip-flop.



$\overline{PRE}$	$\overline{CLR}$	J	K	$Clock$	Q_{next}
1	1	0	0	↑	Q
1	1	0	1	↑	0
1	1	1	0	↑	1
1	1	1	1	↑	$\overline{Q}$
1	1	X	X	0,1	Q
0	1	X	X	X	1
1	0	X	X	X	0
0	0	X	X	X	NA

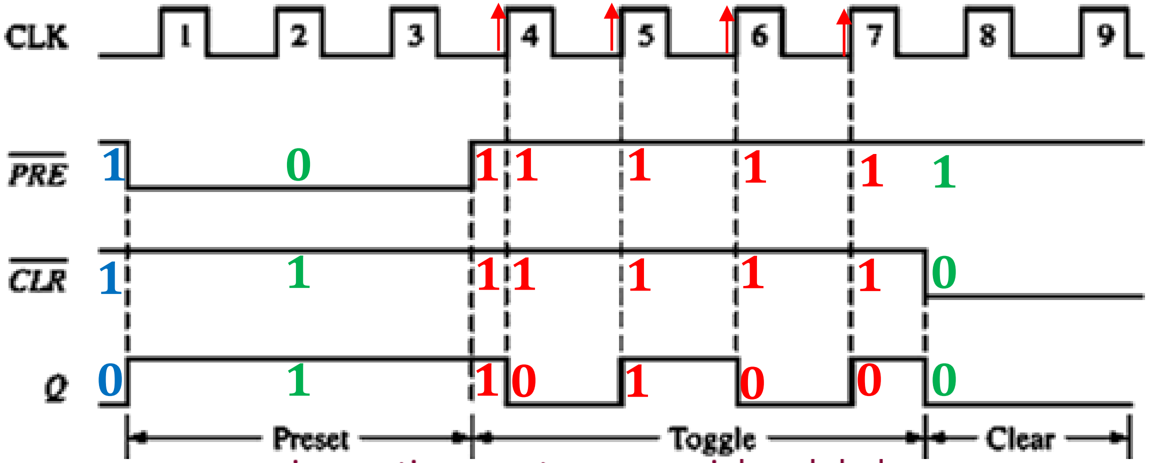
FLIP-FLOPS

ASYNCHRONOUS FLIP-FLOP

Example

Draw the output waveform for asynchronous JK flip-flop given an input timing diagram as below:

CLK pulse	$\overline{PRE}$	$\overline{CLR}$	J	K	FF State	Comment
1, 2, 3	0	1	1	1	SET	JK input - don't care
4,5,6,7	1	1	1	1	TOGGLE	Synchronous mode
8,9	1	0	1	1	RESET	JK input - don't care

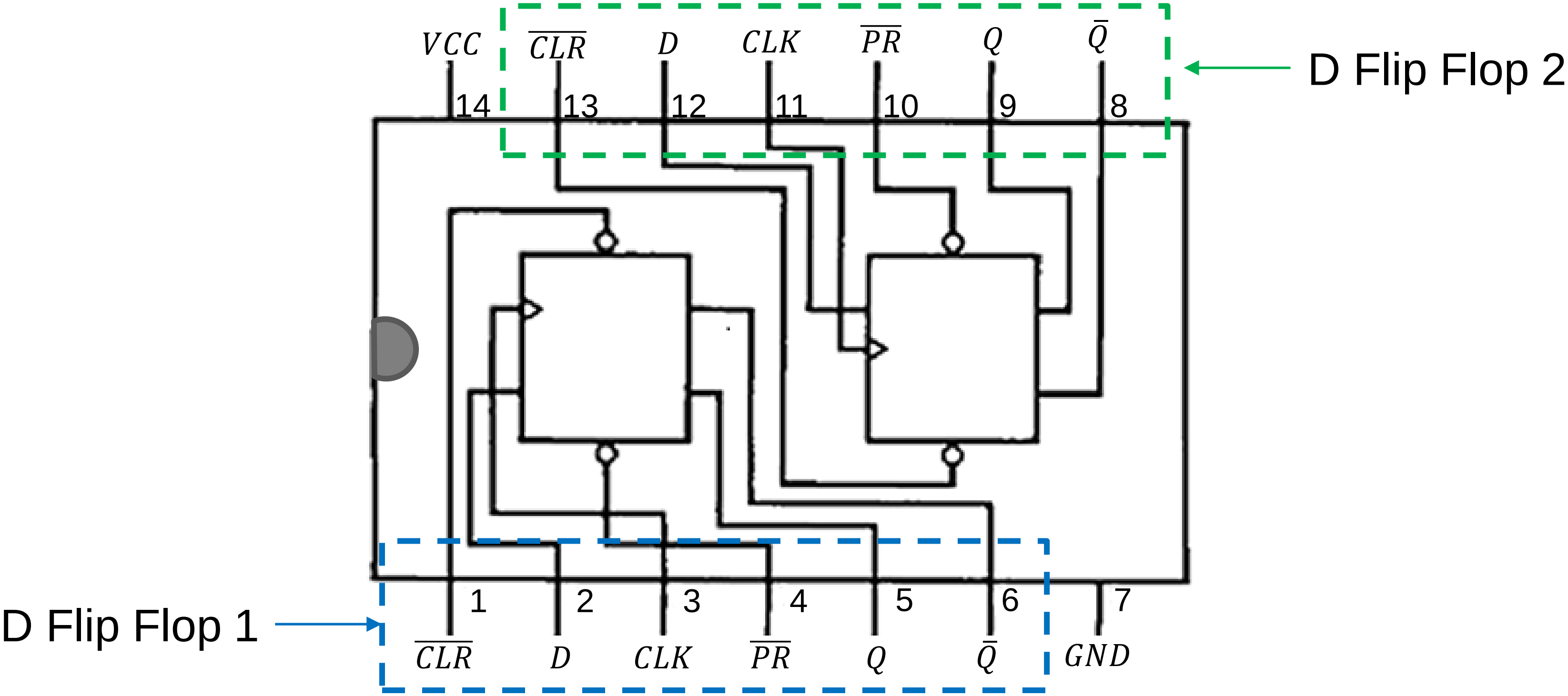


FLIP-FLOPS

FLIP FLOPS IC: 74x74 (DUAL D FLIP FLOP)

- **74x74** contains **two identical D flip-flops** that are independent of each other except sharing VCC and ground.
- The flip-flops are **positive edge triggered** and have active-low asynchronous preset and clear output.

74x74 IC Connection Diagram

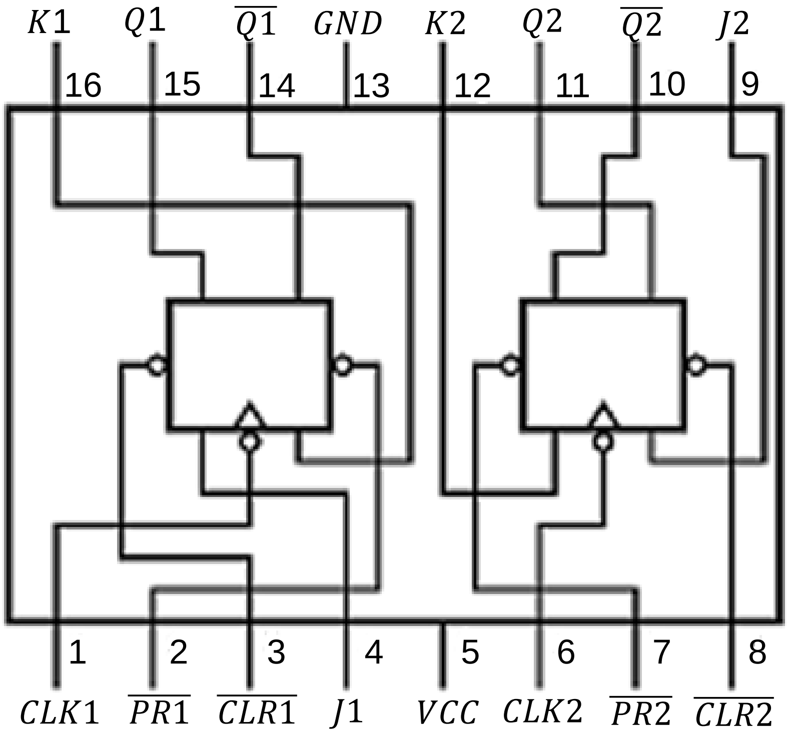


FLIP-FLOPS

FLIP FLOPS IC: 74x76 (DUAL JK FLIP FLOP)

- **74x76** contains **two identical JK flip-flops** that are independent of each other except sharing VCC and ground.
- The flip-flops are **negative edge triggered** and have active-low asynchronous preset and clear output.

74x76 IC Connection Diagram

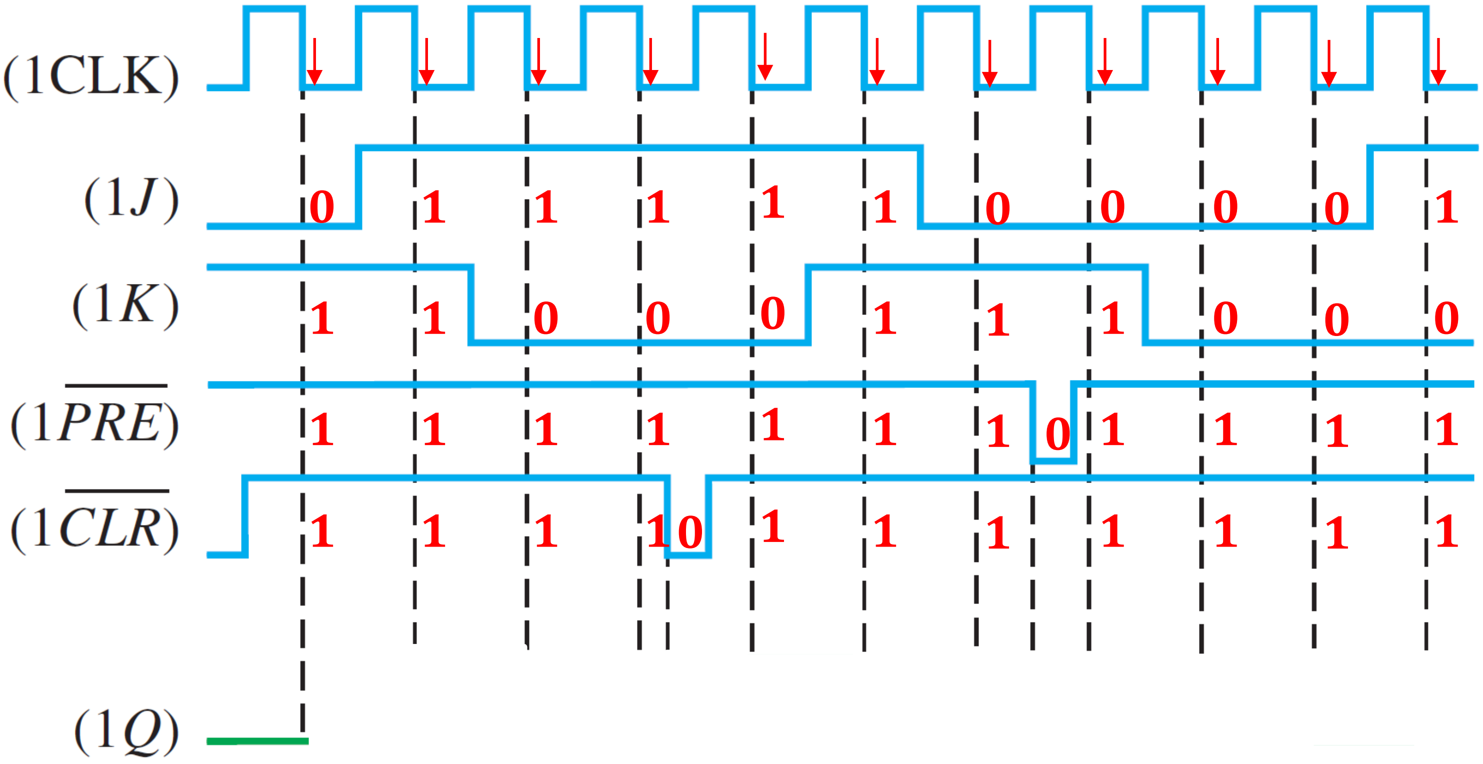


FLIP-FLOPS

FLIP FLOPS IC: 74x76 (DUAL JK FLIP FLOP)

Example

Given input waveforms that are applied to one of the JK flip-flop in 74x76. Determine the 1Q output waveform.


(1CLK)
(1J)
0 1 1 1 1 1 0 0 0 0 1
(1K)
1 1 0 0 0 1 1 1 0 0 0
(1$\overline{PRE}$)
1 1 1 1 1 1 1 0 1 1 1 1
(1$\overline{CLR}$)
1 1 1 1 0 1 1 1 1 1 1 1
(1Q)