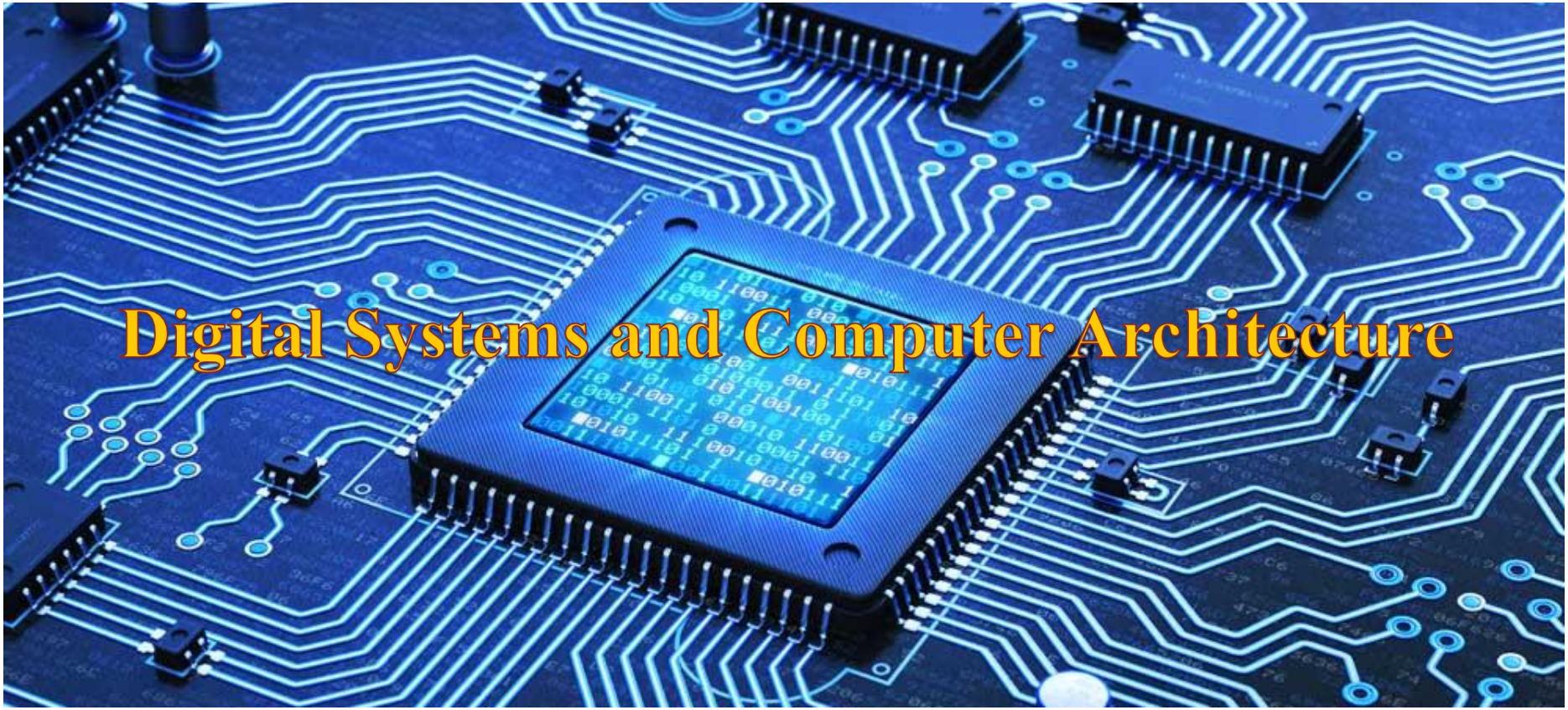


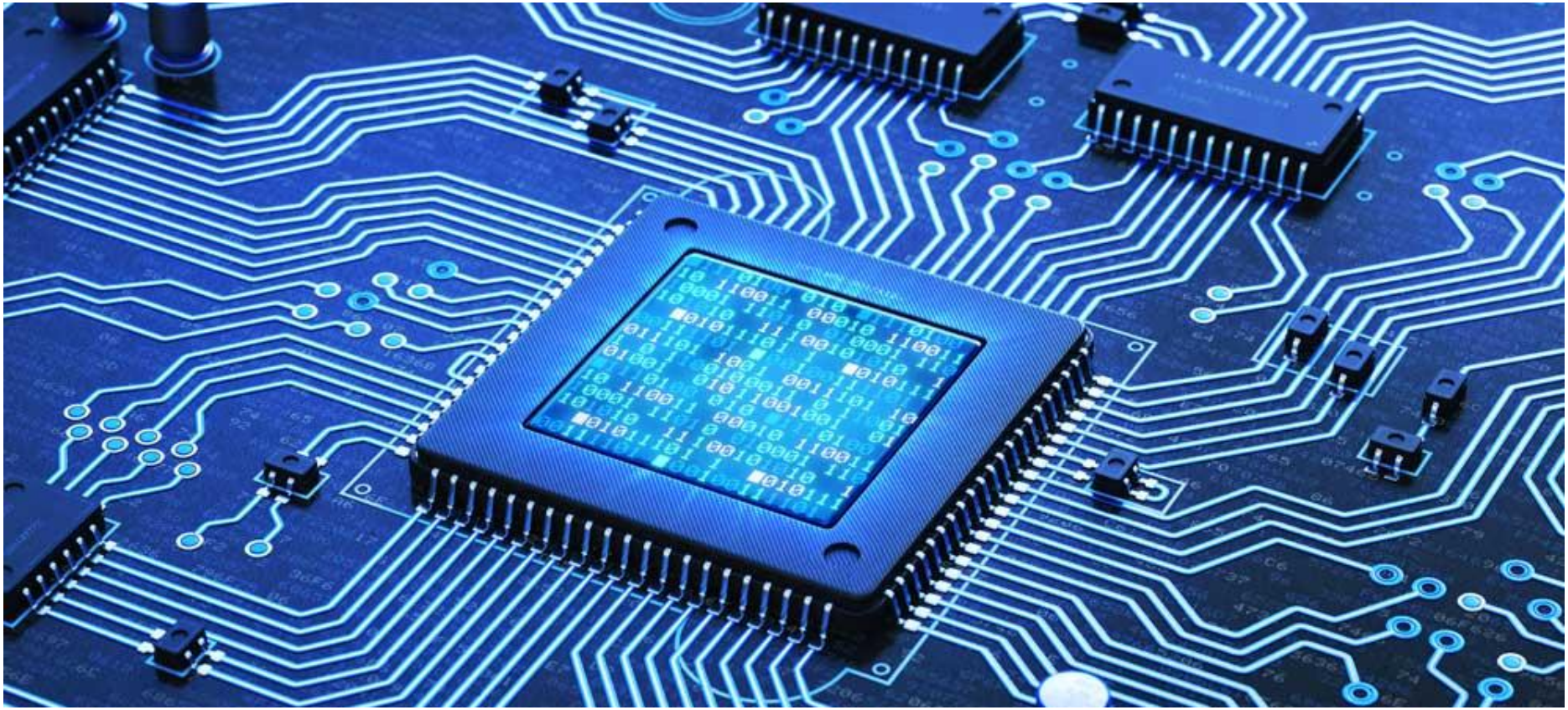
Session 2.10

Module 2

Mouli Sankaran

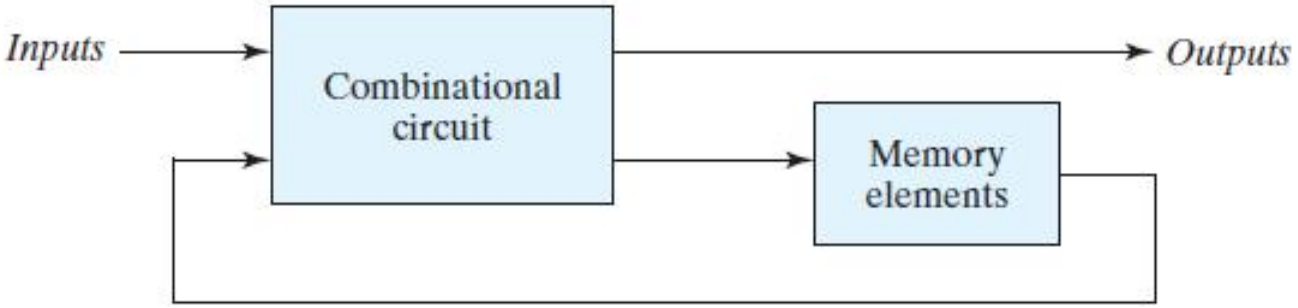
Session 2.10: Focus

- Sequential Circuits
- Classification of Sequential Circuits
  - Asynchronous
  - Synchronous
- Storage Elements
- Level & Edge Triggered
- Latch & Flipflop
- SR Flipflop



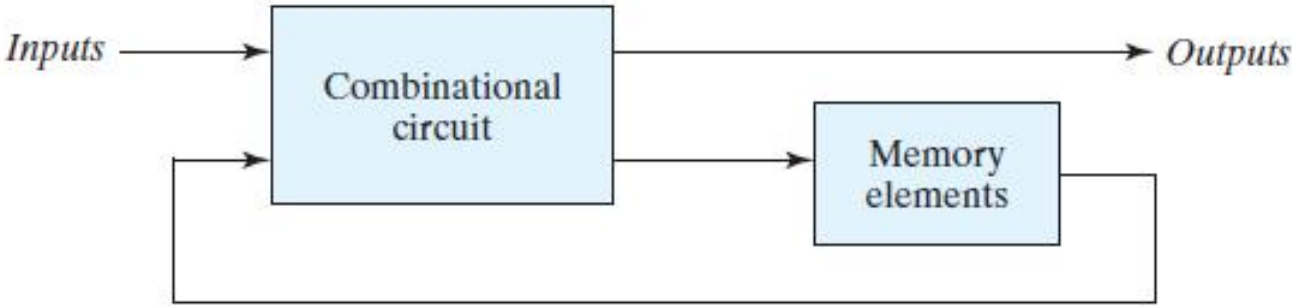
Sequential Circuits

Sequential Circuits



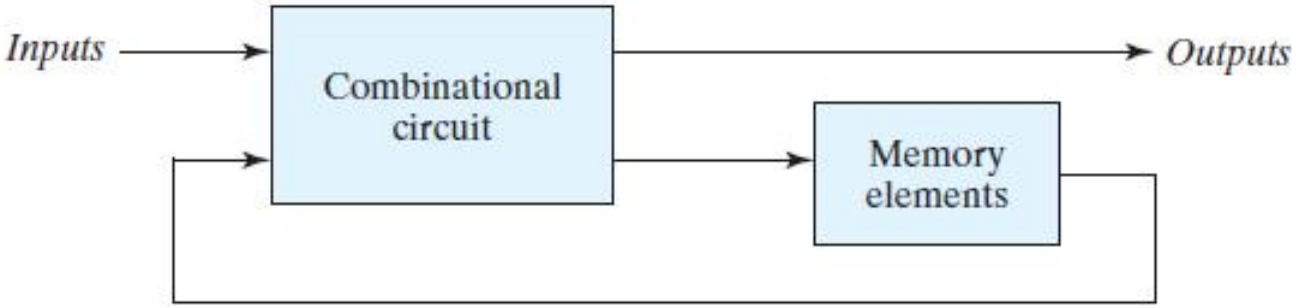
- **Combinational circuits** with **feedback** elements which store the internal **states** (0 or 1) of the circuit are called **Sequential circuits**.
- The **outputs** of **Sequential circuits depend** not only on the **inputs** but also on the **internal stored states** of the circuit
  - Based on the types of storage elements used to store the state, the sequential circuits can be classified

Sequential Circuits … contd.

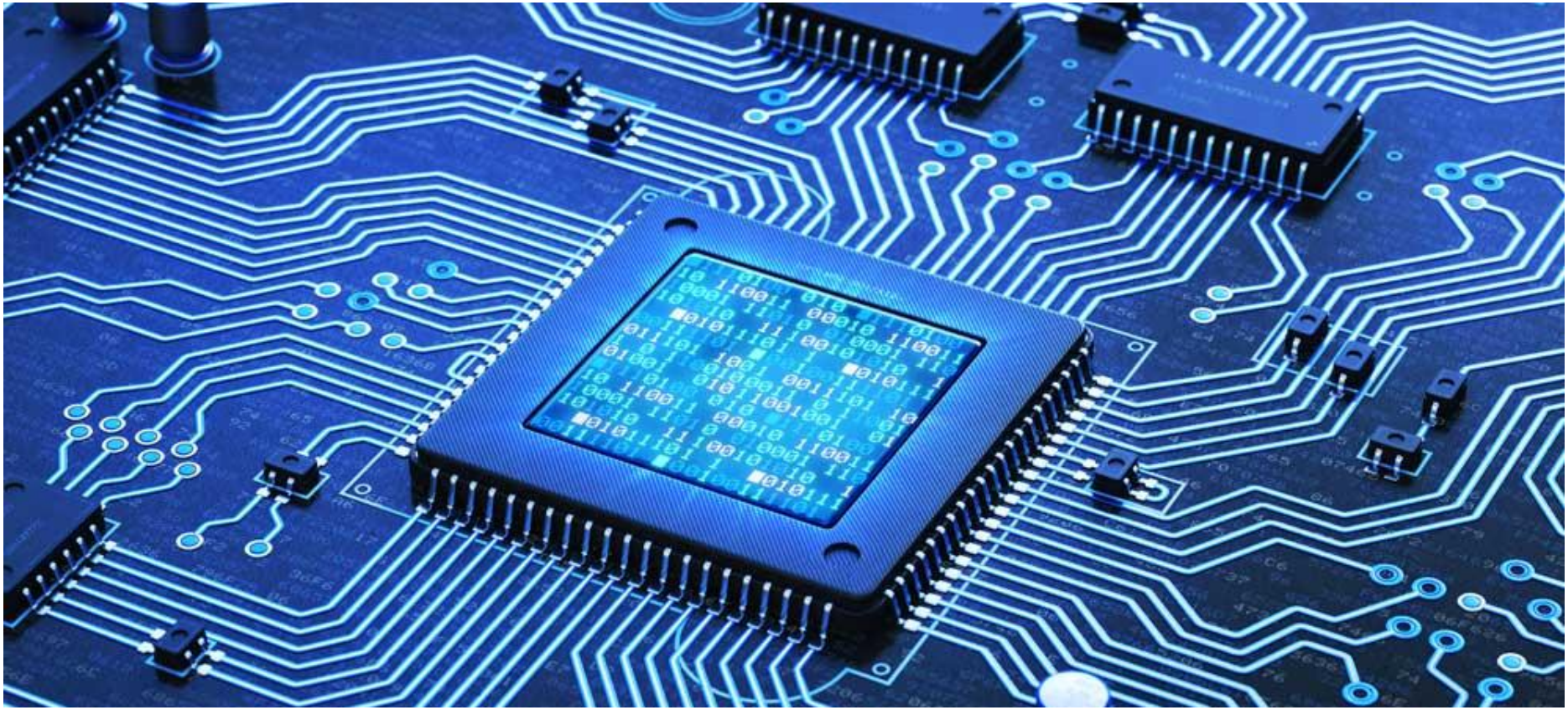


- **Binary information stored** in these elements at any given time **defines** the **state** of **the sequential circuit** at that time.
- The sequential circuit **receives** binary information from **external inputs** that together with the present state of the storage elements,
  - determine the binary **value** of the **outputs**.

Sequential Circuits … contd.



- The next state of the storage elements is also a function of external inputs and the present state.
- Thus, a **sequential circuit** is specified by a time sequence of **inputs**, **outputs**, and **internal states**.
- In contrast, the **outputs** of **combinational logic** depend only on the **present values** of the **inputs**.

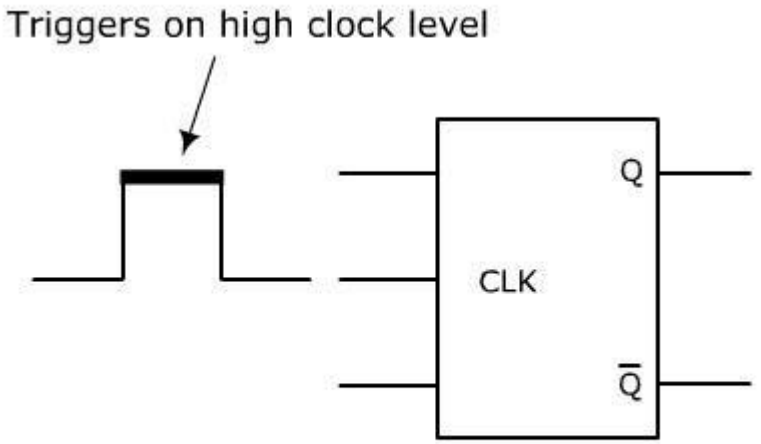


Classification of Sequential Circuits

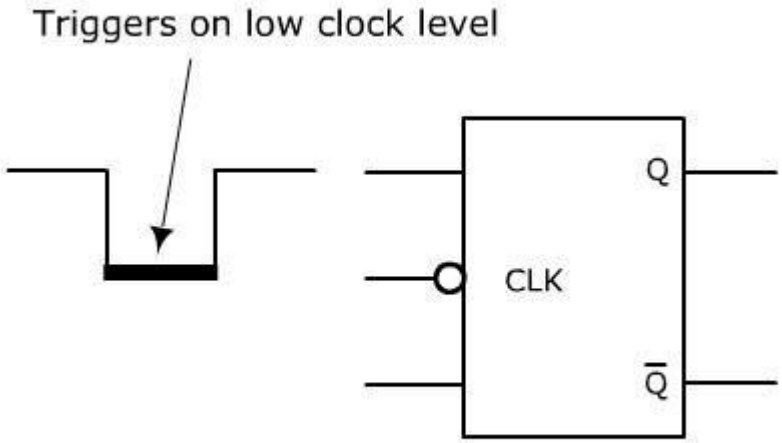
Classification of Sequential Circuits

- There are **two** main **types** of sequential circuits, and their classification is
  - A function of the **timing** of their **signals**
- **Synchronous sequential circuit**
- **Asynchronous sequential circuit**

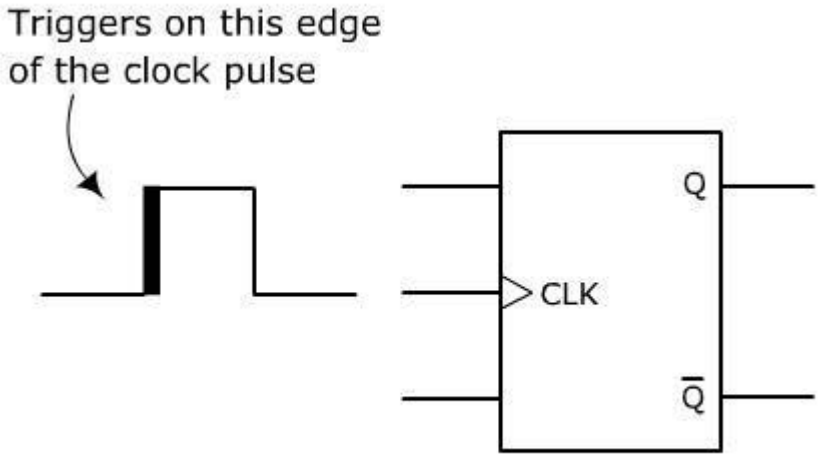
Level and Edge Triggering of Flip-flops



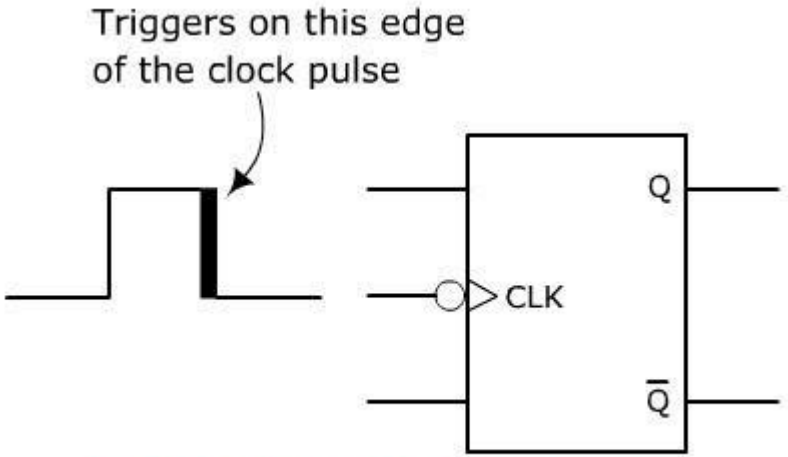
High Level Triggering



Low Level Triggering



Positive Edge Triggering

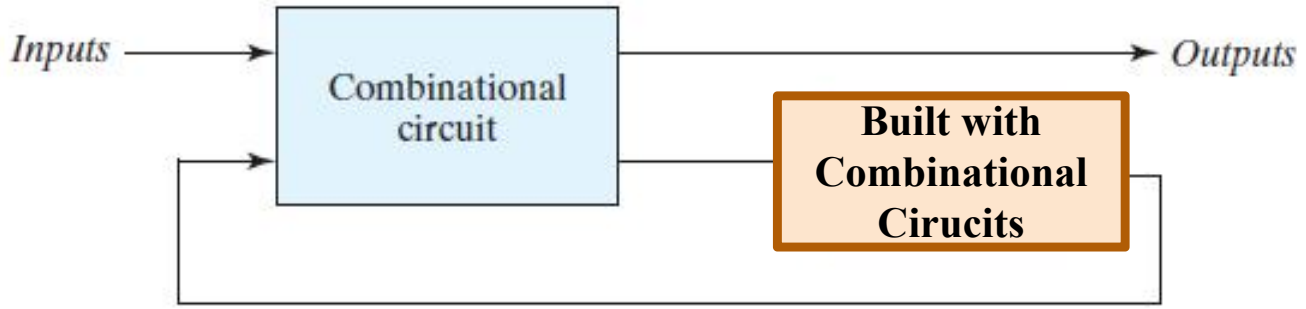


Negative Edge Triggering

Level & Edge Triggered

Edge Triggering	Level Triggering
Edge triggering is based on detecting a sharp edge in the input signal.	Level triggering is based on recognizing a specified signal level.
Edge triggering is frequently used in synchronous circuits, such as counters and flip-flops.	Level triggering is frequently employed in applications that call for continuous monitoring of an input signal, such as data acquisition and control systems.
Edge triggering is particularly helpful in applications that need precise timing.	Level triggering lacks the ability to regulate precise timing.
When edge triggering is used, the output signal is activated when the trigger edge is detected and changes to the opposing state. As long as the input signal is at or above the trigger level.	In level triggering, the output signal will remain in the triggered condition.

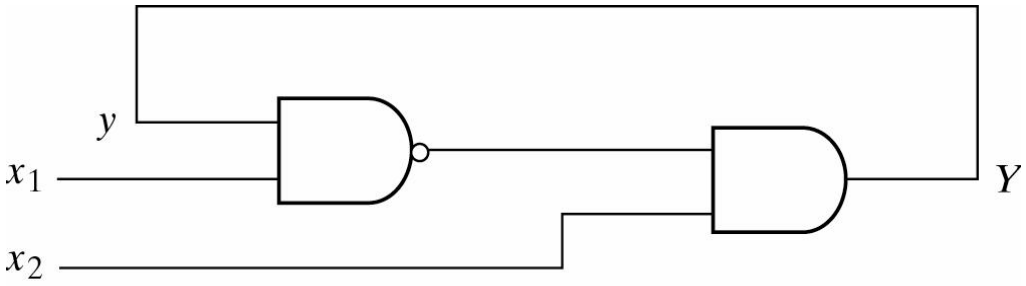
Asynchronous Sequential Circuits



- In Asynchronous Sequential circuits, the **feedback** or **memory** elements are **built with combinational circuits**
- There is a **delay** involved in the **feedback** reaching, as inputs, due to the **propagation delay** of the feedback elements
- There is **no clock signal** here which synchronizes the activities, **leading to unstable state** or **race conditions**
- Due to this, asynchronous circuits are **rarely used**, though it is much **faster than** the **synchronous sequential circuits**

Example: Asynchronous Sequential Circuits

- Combinational circuits with a feedback circuit
- An asynchronous sequential circuit may become unstable and oscillate between **unstable** states because of the presence of feedback.



Let us see what happens the inputs x_1 and x_2 are both are 1s.

We can notice that the output **Y** will alternate between 1 and 0, when both the inputs are 1s.

The excitation function is:

$$Y = (x_1 y)' x_2 = (x_1' + y') x_2 = x_1' x_2 + x_2 y'$$

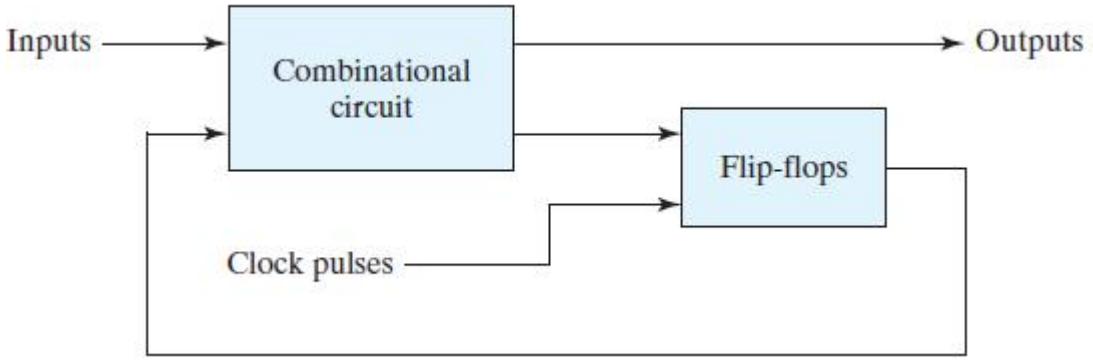
and the transition table for the circuit is:

y \ $x_1 x_2$	00	01	11	10
0	(0)	1	1	(0)
1	0	(1)	0	0

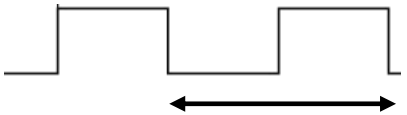
Those values of *Y* that are equal to *y* are circled and represent stable states. When the input $x_1 x_2$ is 11, the state variable alternates between 0 and 1 indefinitely.

Synchronous Sequential Circuits

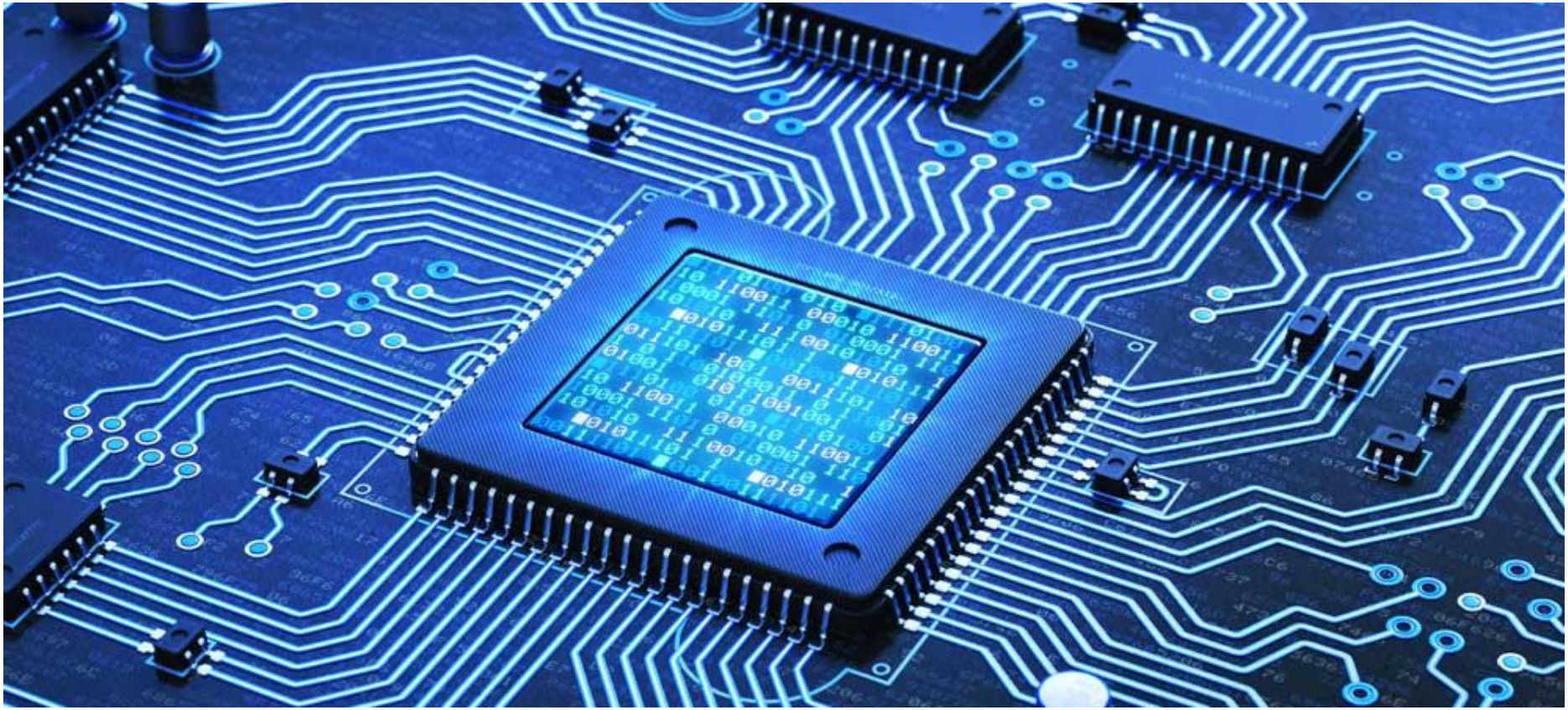
Clock Pulses



- In **synchronous circuits** the **memory** elements are **clocked flip-flops**
- **Transition** of **states** happen **only during** the **clock edges**
- **State** of the **flip-flops** will be "**frozen**" **until** the **next** (correct) transition of the **clock**
- Each **state** is kept **unaltered** in the **flip-flops** (memory elements) for **one period of the clock**
- Let us study about flip-flops now ..



Period



Storage Elements
Latch & Flipflop

Storage Elements – Latches/Flip-flops

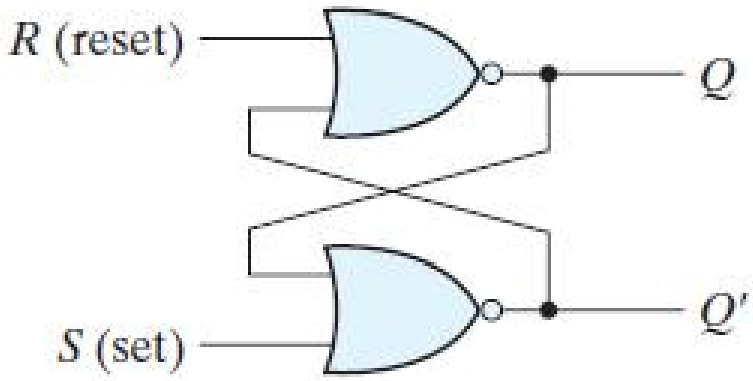
- Storage element in a digital circuit can maintain a binary state indefinitely
  - As long as power is delivered to the circuit
  - And until directed by an input signal to switch states
- Storage elements that operate with **signal levels** (rather than **signal transitions** or **clock edges**) are referred to as **latches**
  - Those **controlled** by a **clock transitions** are **flip-flops**
  - **Latches** are said to be **level sensitive devices**
  - **Flip-flops** are **edge-sensitive devices**

Latch Vs Flipflop

NO	Flip-flop	Latch
1	Flip-flop is a bistable device i.e., it has two stable states that are represented as 0 and 1.	Latch is also a bistable device whose states are also represented as 0 and 1.
2	It checks the inputs but changes the output only at times defined by the clock signal or any other control signal.	It checks the inputs continuously and responds to the changes in inputs immediately.
3	It is a edge triggered device.	It is a level triggered device.
4	Gates like NOR, NOT, AND, NAND are building blocks of flip flops.	These are also made up of gates.
5	They are classified into asynchronous or synchronous flipflops.	There is no such classification in latches.
6	It forms the building blocks of many sequential circuits like counters.	These can be used for the designing of sequential circuits but are not generally preferred.
7	a, Flip-flop always have a clock signal	Latches doesn't have a clock signal
8	Flip-flop can be build from Latches	Latches can be build from gates
9	ex:D Flip-flop, JK Flip-flop	ex:SR Latch, D Latch

SR-Latches with NOR Gates

Logic Diagram



Active HIGH circuit

Function Table

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(forbidden)

Active HIGH here corresponds to the control inputs (**R & S**)
When R = 1, Q is Reset to 0
When S = 1, Q is set to 1

It is forbidden because both **Set (1)** and **Reset (1)** commands should not be given simultaneously

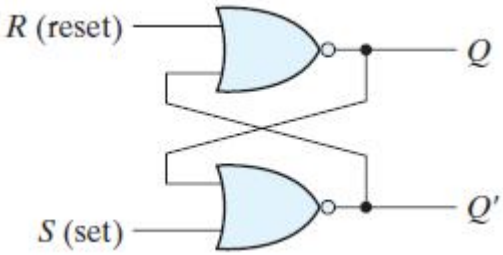
Always **Set** means **setting** a value to **ONE**
Always **Reset** means **resetting** a value to **ZERO**

SR-Latches

- Since **SR-Latches** are l**evel-sensitive** and is **not operated with clocks**, they **cannot be used** in **synchronous** sequential **circuits**
  - Since SR-Latches are the building blocks for Flip-flops
  - We study them here

SR-Latches (NOR) Explained

- When **Q = 0** and **Q' = 1**, it is in the **reset** state
- When **Q = 1** and **Q' = 0,** it is in the **set** state
- Outputs **Q** and **Q'** are expected to be **complement of each other**

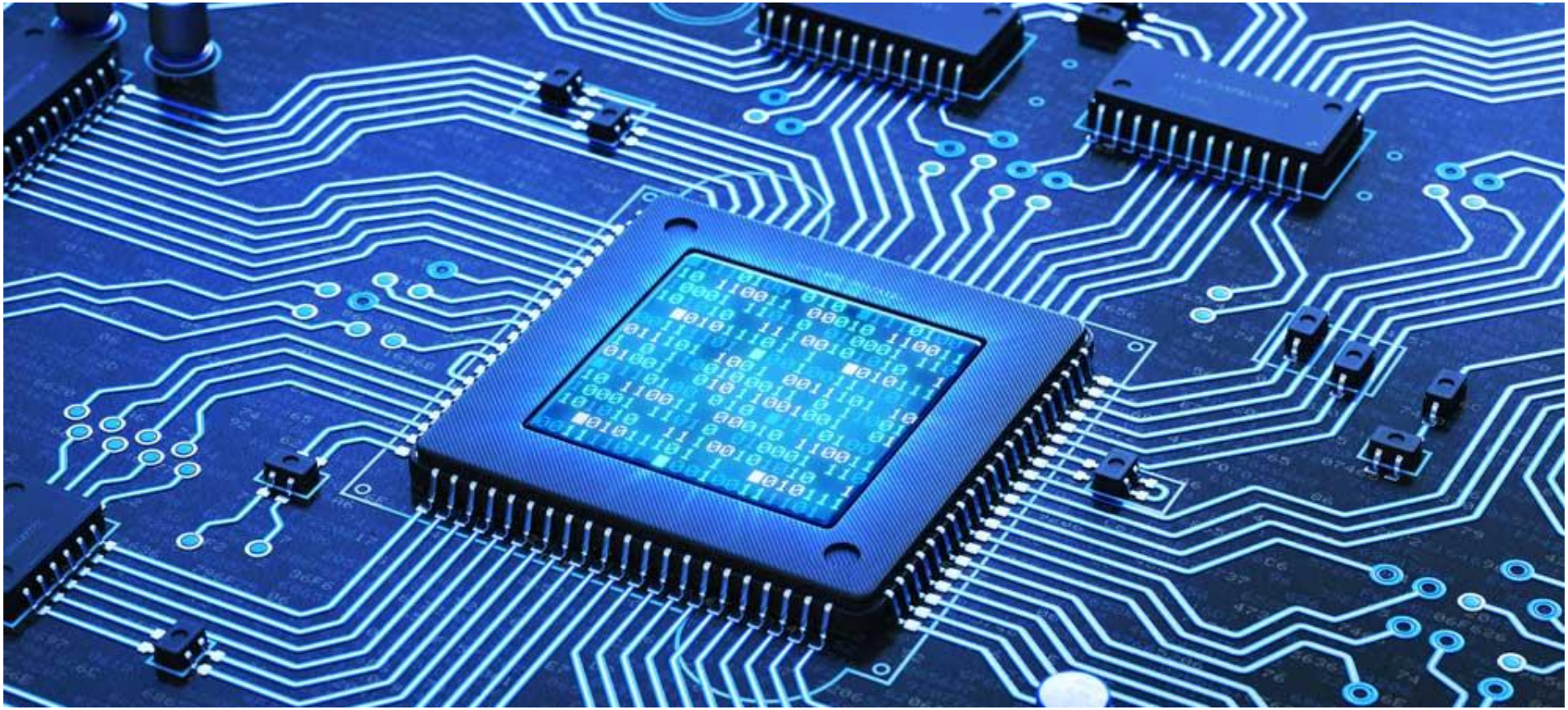


Active HIGH circuit

- However, when **both inputs are equal to 1** at the **same time**, a condition in which **both outputs** are equal to **0** occurs
  - Rather than being mutually complementary
- If **both inputs** are then **switched** to **0 simultaneously**,
  - The device will enter an **unpredictable** or undefined **state** or a **meta-stable state**
- In practical applications, setting **both inputs** to **1** is **forbidden**
- Under normal conditions, **both inputs** of the latch **remain** at **0** unless the state has to be changed

SR-Latches (NOR) Explained … contd.

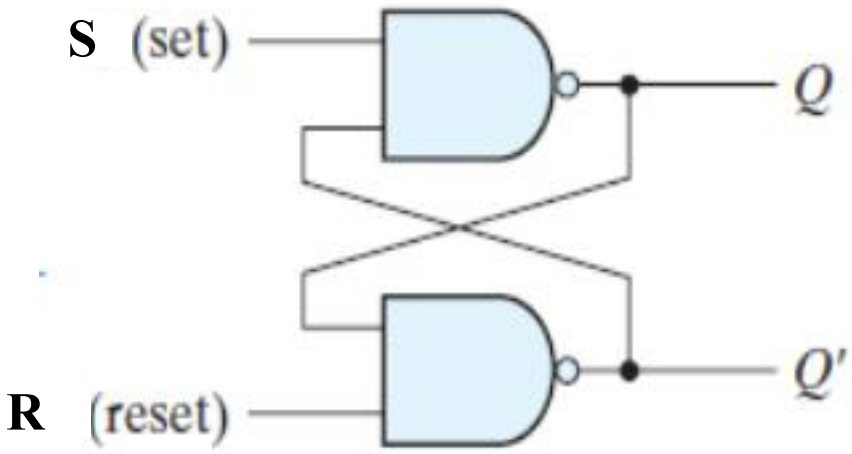
- Thus, when **both inputs S and R** are equal to **0**, the latch can be in **either** in the **set (Q = 1) or** in the **reset** (**Q = 0**) state
  - Depending on **which input (S or R) was most recently a 1**
- In normal operation, having **both inputs S and R** at **1** is **avoided** by making sure that **1's are not applied** to **both** inputs simultaneously
  - Which is a **forbidden state**



Storage Elements
SR Latches – With NAND Gates

SR Flipflop

Logic Diagram



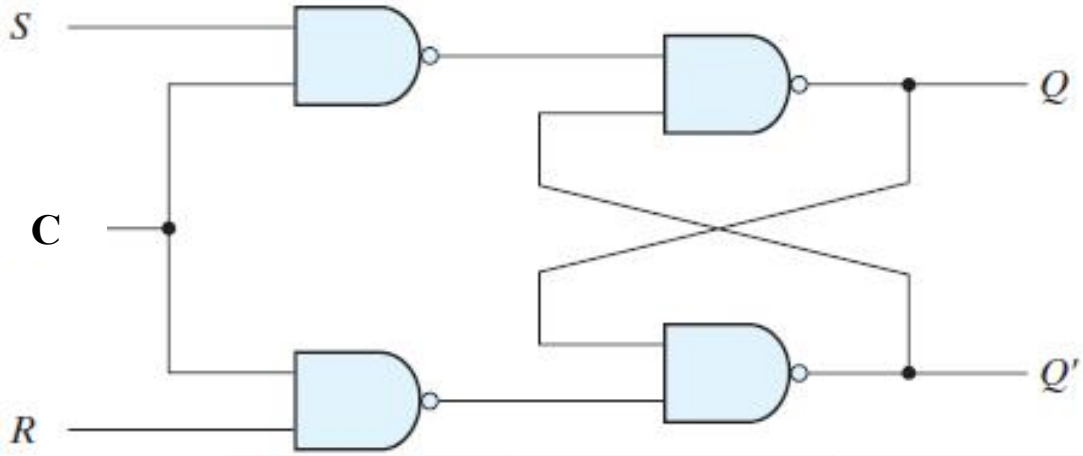
S	R	Q	Q'
0	0	0	1
0	1	0	1
1	0	1	0
1	1	∞	∞

It is forbidden because
If both **Set (1)** and **Reset (1)**
the next state of
Q and Q' are indeterminate

Always **Set** means setting a value to **ONE**
Always **Reset** means resetting a value to **ZERO**

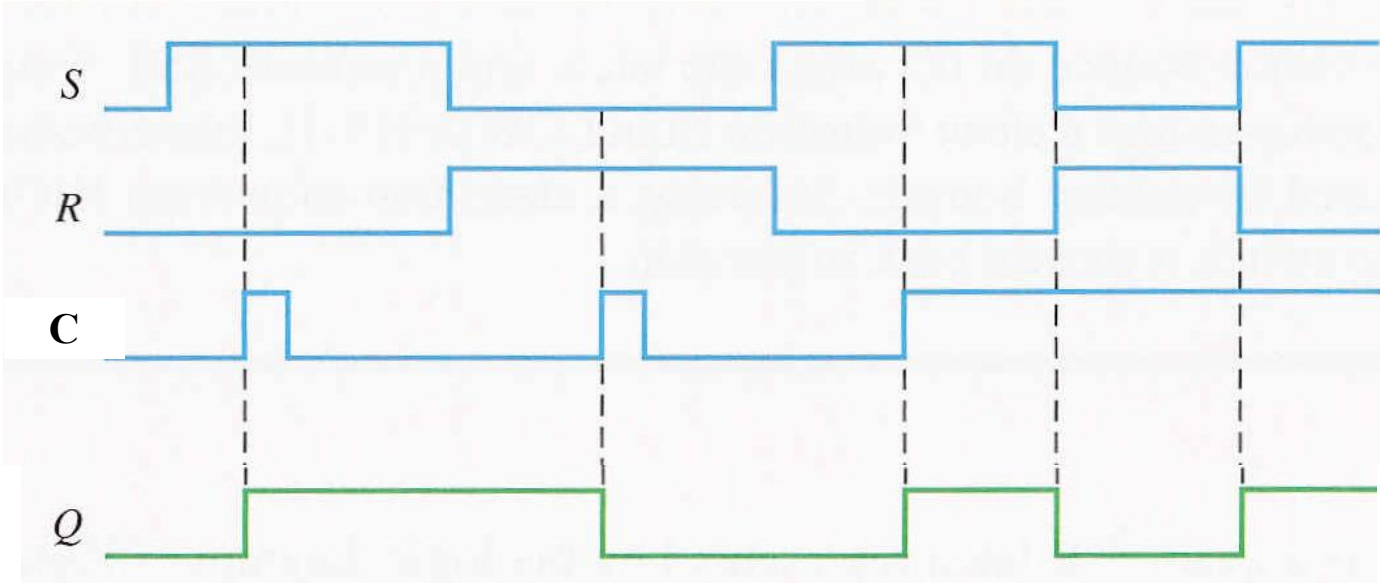
Quiz 1: Draw the output waveform

(Assume Q is initially zero)



C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

ANS:



Session 2.10: Summary

- Sequential Circuits
- Classification of Sequential Circuits
  - Asynchronous
  - Synchronous
- Storage Elements
- Level & Edge Triggered
- Latch & Flipflop
- SR Flipflop