AAB University

Faculty of Computer Sciences

Introduction to Digital Technologies and Circuits

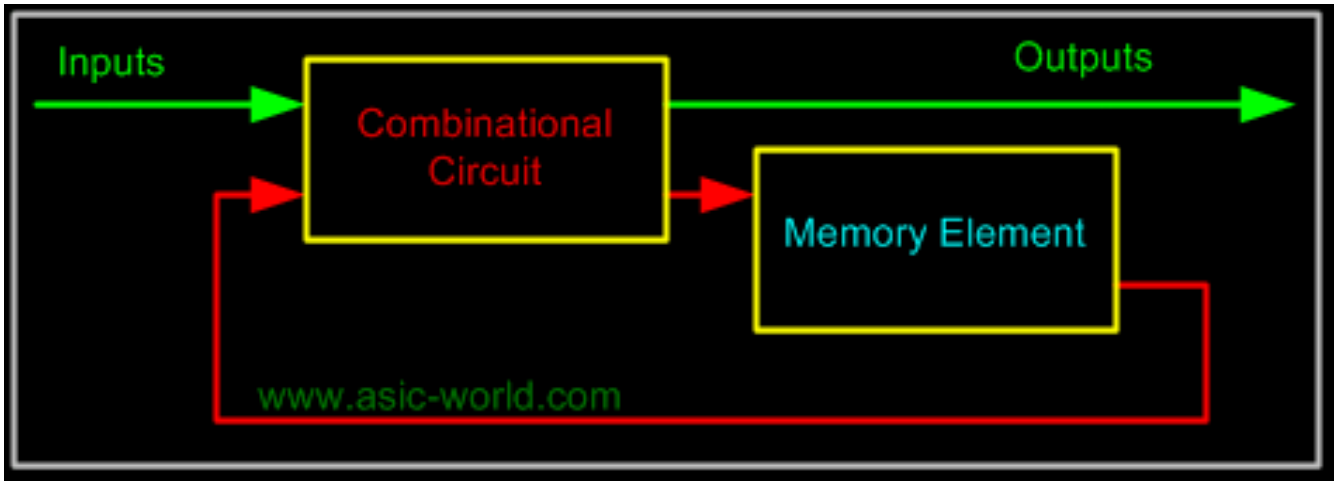
Week 11:

Sequential Logic Circuits

Asst. Prof. Dr. **Mentor Hamiti**
mentor.hamiti@universitetiaab.com

Digital Logic Circuits

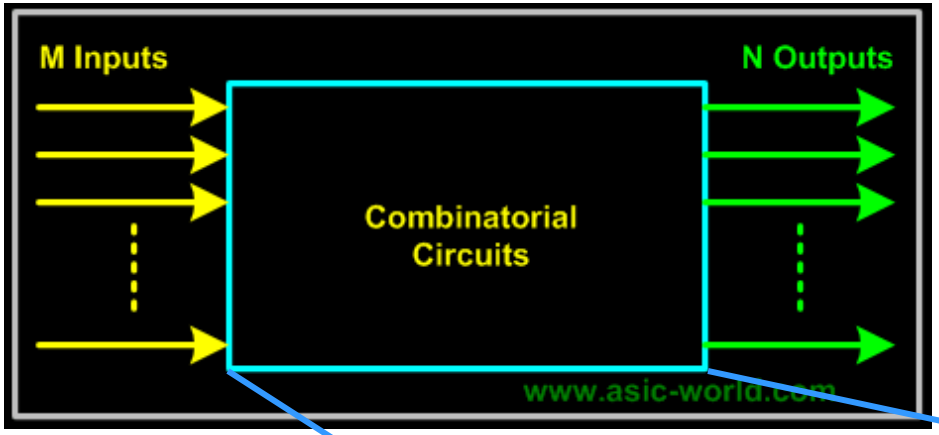
- There are two types of Digital Logic Circuits:
  - Combinational Logic Circuits
  - Sequential Logic Circuits



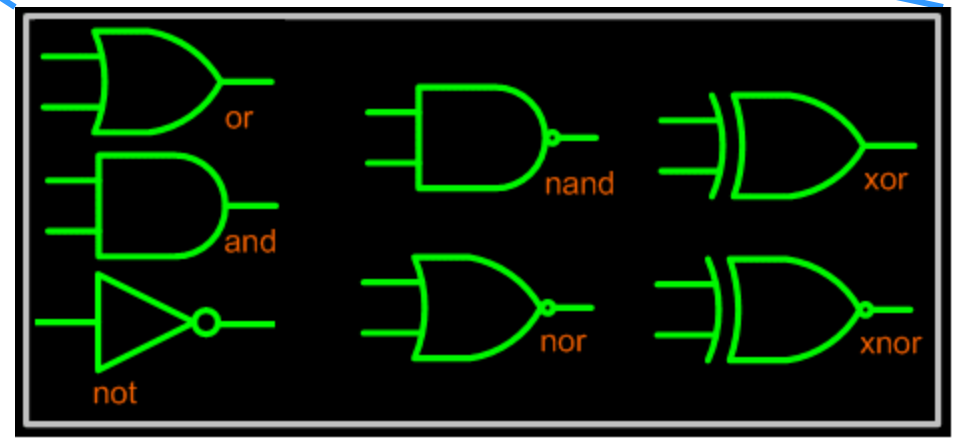
- Combinational logic output depends on the inputs levels, whereas sequential logic output depends on stored levels and also the input levels.

Combinational Logic Circuits

- All Combinational Circuits are composed by simple combinational logic elements - "**Logic Gates**".

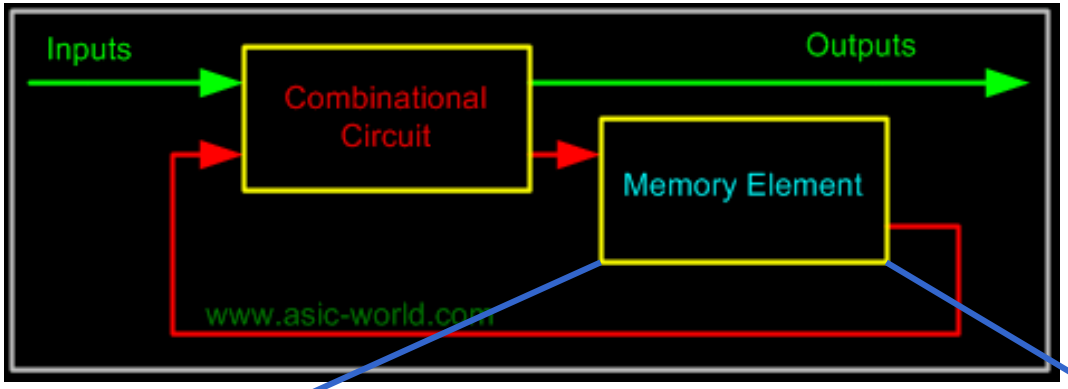


- Logic Gates:
  - Basic Logic Gates
  - Universal Logic Gates
  - Exclusive Logic Gates

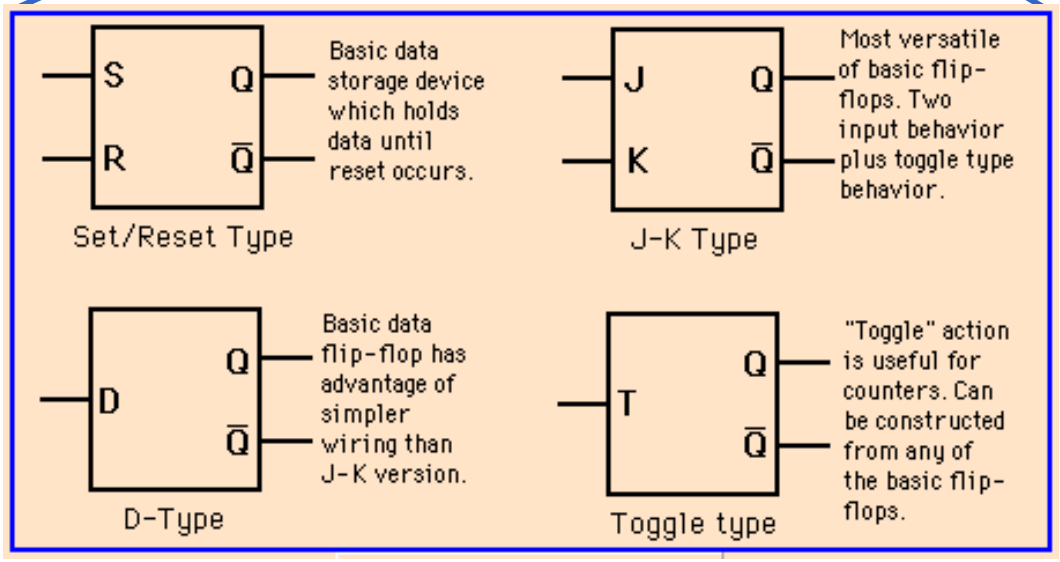


Sequential Logic Circuits

- "**Flip-flop**" is the common name given to two-state devices which offer basic memory for sequential logic operations.



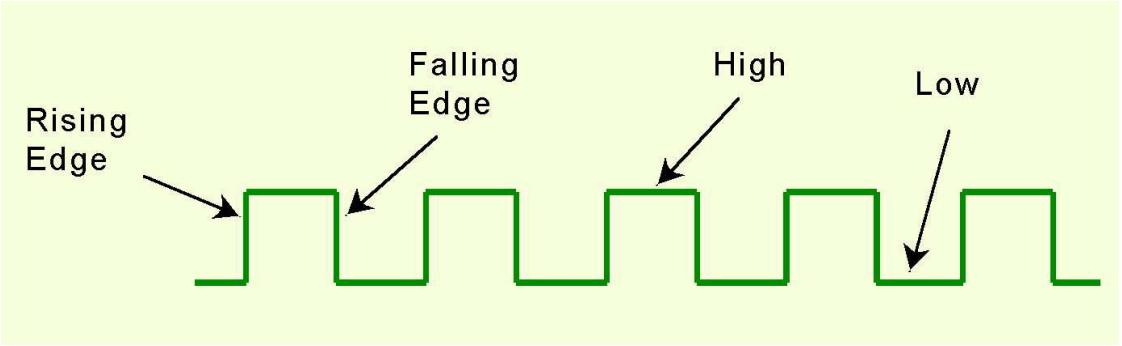
- **S-R**
- **J-K**
- **D**
- **T**



Sequential logic differs from combinational logic in that the output of the logic device is dependent not only on the present inputs to the device, but also on past inputs; i.e., the output of a sequential logic device depends on its present internal state and the present inputs. This implies that a sequential logic device has some kind of memory of at least part of its "history" (i.e., its previous inputs).

Types of Sequential Logic Circuits:

- A **synchronous** sequential circuit uses a clock to order events
  - a clock is a circuit that emits a series of electrical pulses
  - state changes in sequential circuits only occur when the clock ticks



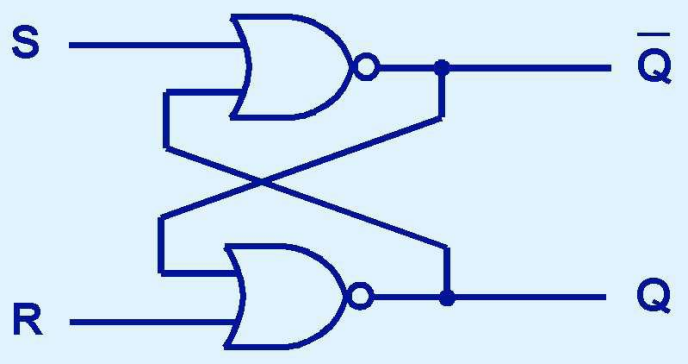
- **Asynchronous** sequential circuits become active the moment any input value changes

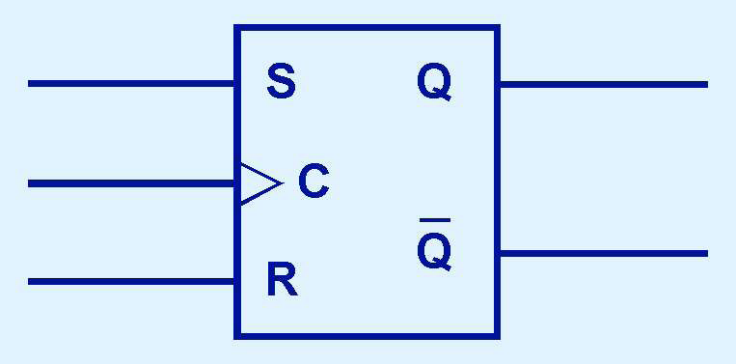
Flip-Flops (Memory Elements)

- Flip-flops can maintain a binary state (i.e. store a bit) indefinitely
- Major differences amongst flip-flops are the number of inputs they possess and the manner in which the inputs affect the current state
- There are four types of flip-flops:
  - **SR**
  - **JK**
  - **D**
  - **T**

SR flip-flop

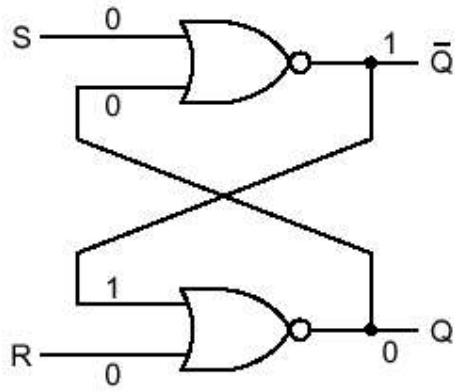
- Also known as the Set-Reset Latch (SR-Latch)
  - one of the most basic memory components
  - consists of a pair of interconnected NOR gates that allows the output to be set to 1 and reset to 0

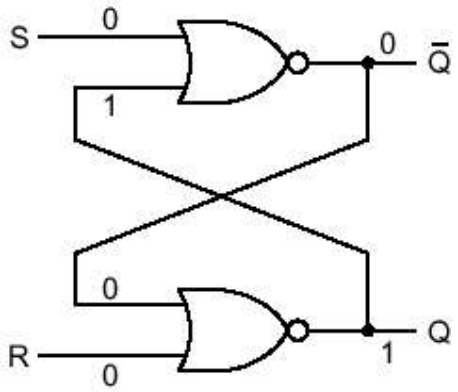




SR flip-flop

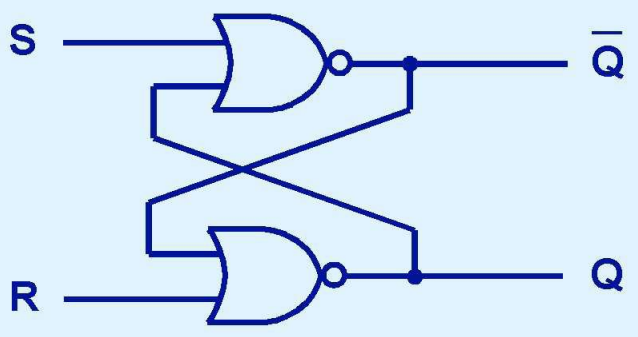
- Operation when S = 0, and R = 0
  - if Q = 1 then it remains so, while Q´ = 0
  - if Q = 0 then it remains so, while Q´ = 1





SR flip-flop

- The behavior of an SR-Latch can be described by a **characteristic table**
  - Q(*t*) means the value of the output at time *t*
  - Q(*t+1*) is the value of the output after the next clock pulse



S	R	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	undefined

SR flip-flop

- The SR-latch has three inputs S, R and its current output, Q

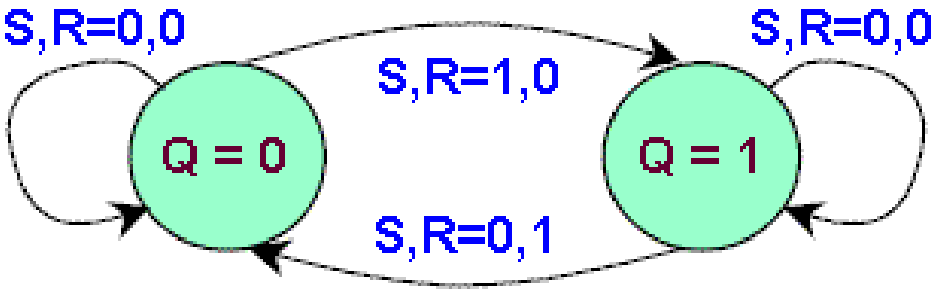
- A **truth table** can be constructed for this circuit

- There are two undefined values:
  - when S = 1, and R = 1 the SR-Latch is undefined (because Q and Q´ both equal 0)

		Present State	Next State
S	R	Q(t)	Q(t+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	undefined
1	1	1	undefined

SR flip-flop

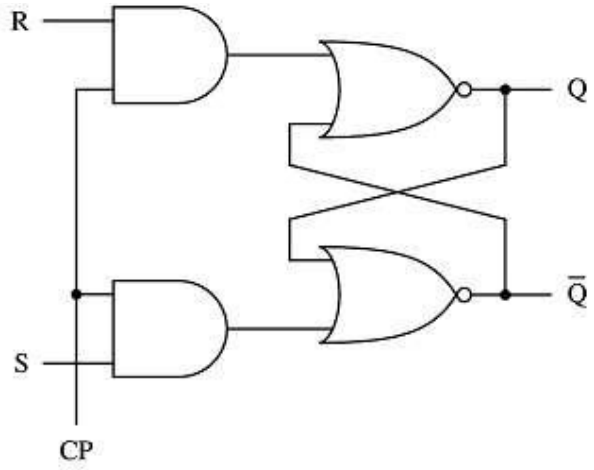
- State Diagram:



S	R	Present State Q(t)	Next State Q(t+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	undefined
1	1	1	undefined

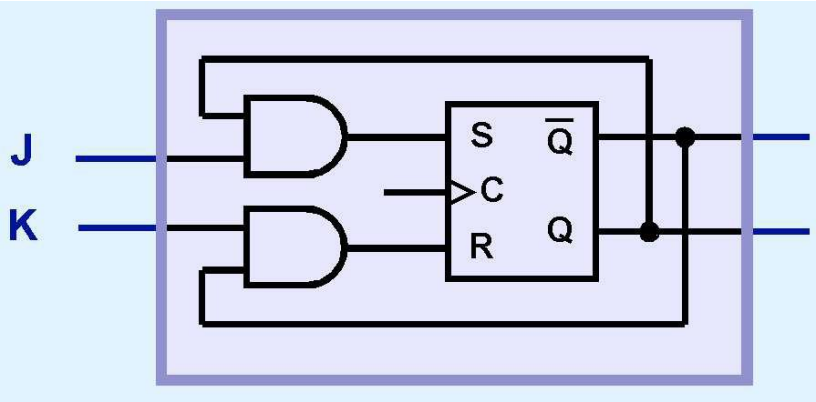
Clocked SR flip-flop

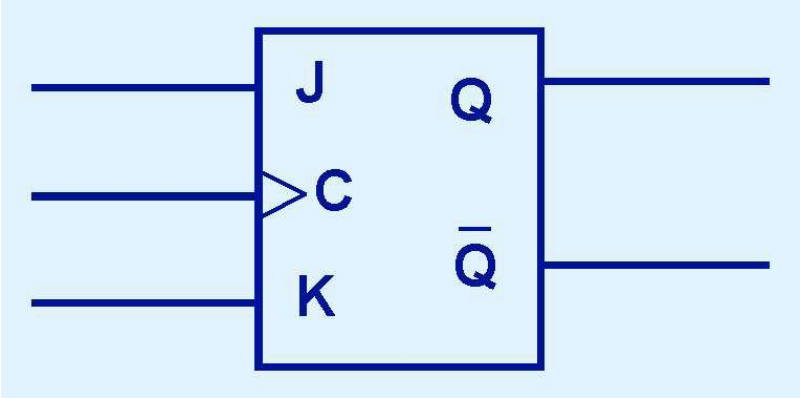
- The inputs S and R are enabled by a clock pulse (**CP**)
  - E.g. when the clock pulse is low, no state changes occur



JK flip-flop

- The SR-Latch can be modified to provide a stable state when both inputs S and R are 1
- This modified circuit is called a JK flip-flop





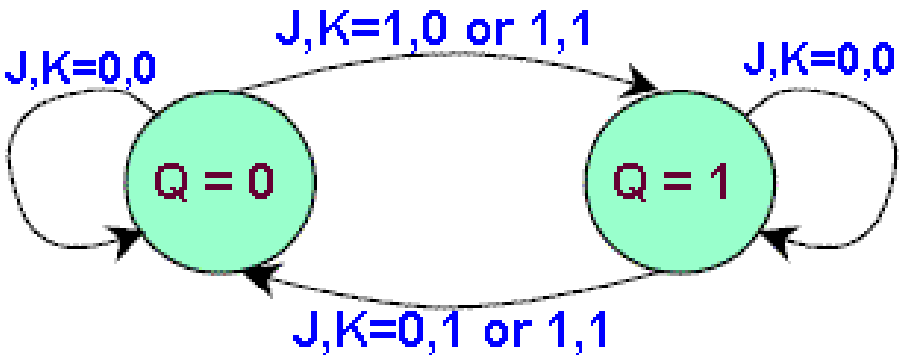
JK flip-flop

- The characteristic truth table for the JK flip-flop shows the flip-flop is stable for all inputs
  - the J input alone performs the **set** function
  - the K input alone performs the **reset** function
  - if J = 1 and K = 1, the output is **toggled** or reversed

J	K	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	$\overline{Q}$(t)

JK flip-flop

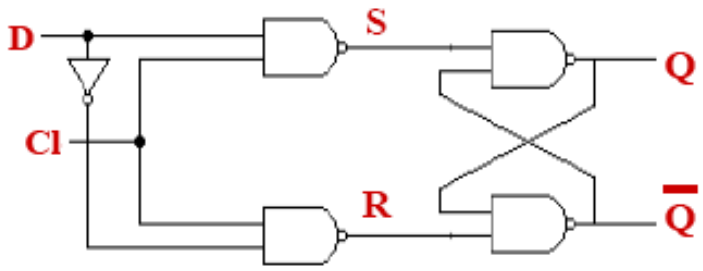
- State Diagram:



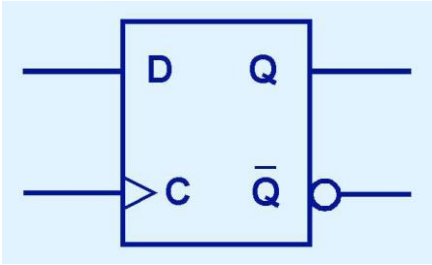
J	K	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	$\overline{Q}$(t)

D flip-flop

- The **D** (data) **flip-flop** is another variant of the SR-Latch



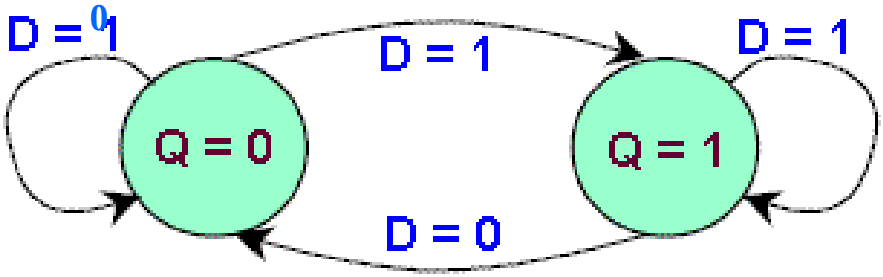
- The output of the circuit only changes when D changes – it remains the same otherwise
  - the circuit stores 1 bit of information and can be used to implement computer memory



D	Q(t+1)
0	0
1	1

D flip-flop

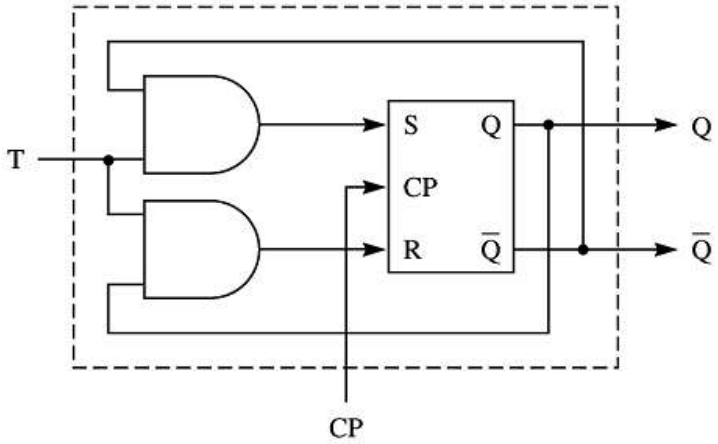
- State Diagram:

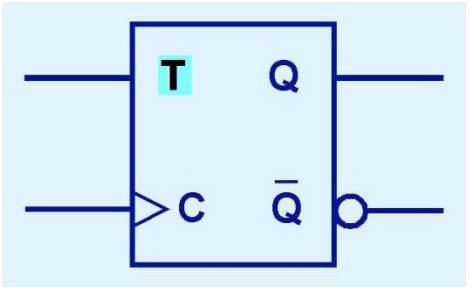


D	Q(t+1)
0	0
1	1

T flip-flop

- The **T** (toggle) **flip-flop** is another variant of the SR-Latch



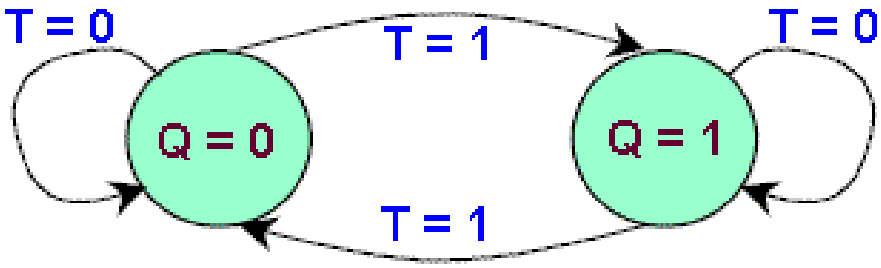


- Characteristic Table for T Flip-Flop:

T(*t*)	Q(*t*)	Q(*t* + 1)	Condition
0	0	0	No change
0	1	1	
1	0	1	Toggle
1	1	0	

T flip-flop

- State Diagram:



T(*t*)	Q(*t*)	Q(*t* + 1)	Condition
0	0	0	No change
0	1	1	
1	0	1	Toggle
1	1	0	

Flip-Flops

- Example 1:
  - Realize the *flip-plop* of type **T** using **D** *flip-flop*!

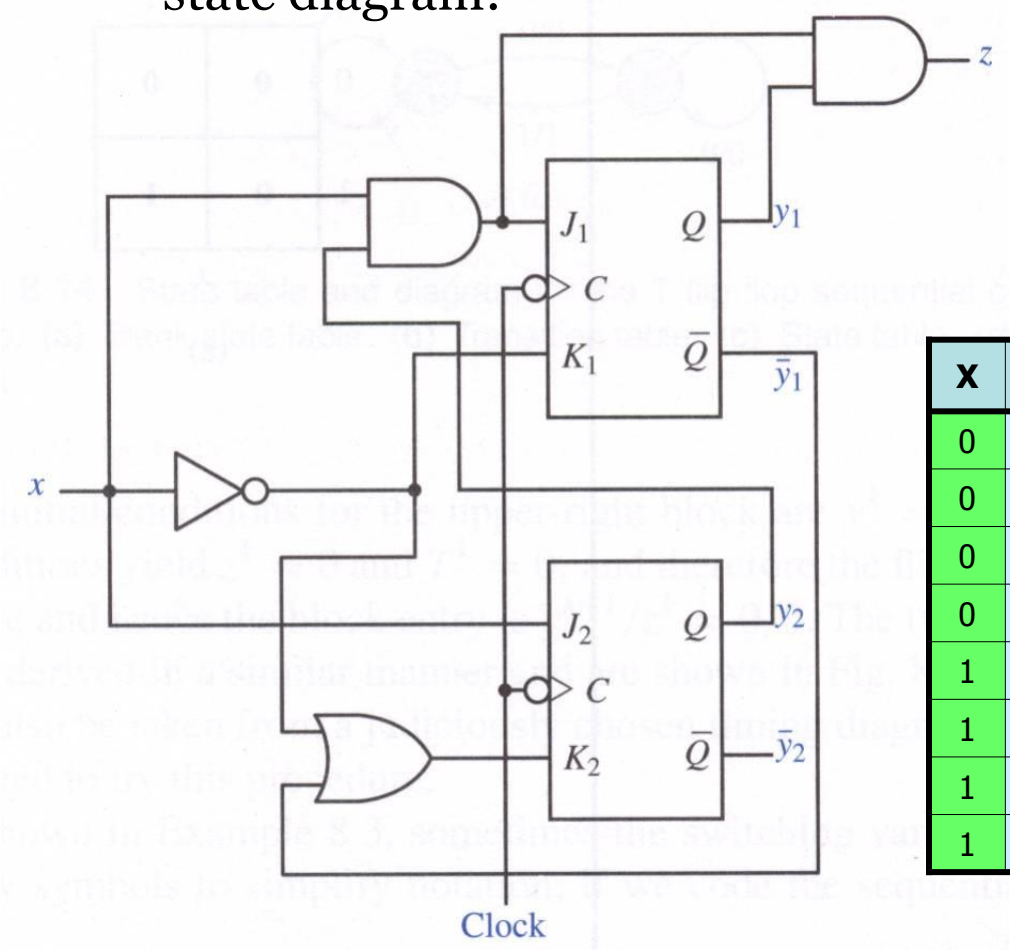
Starting from:

T	Q	Q^+	D
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

- Example 2:
  - Realize the *flip-plop* of type **JK** using **T** *flip-flop!*

Synchronous Sequential Logic Circuit

- Example 3:
  - For the following sequential logic circuit, find the state table and state diagram!



X	Q_1	Q_2	J_1	K_1	J_2	K_2	Z	Q_1^+	Q_2^+
0	0	0							
0	0	1							
0	1	0							
0	1	1							
1	0	0							
1	0	1							
1	1	0							
1	1	1							

- Questions?!

mentor.hamiti@universitetiaab.com