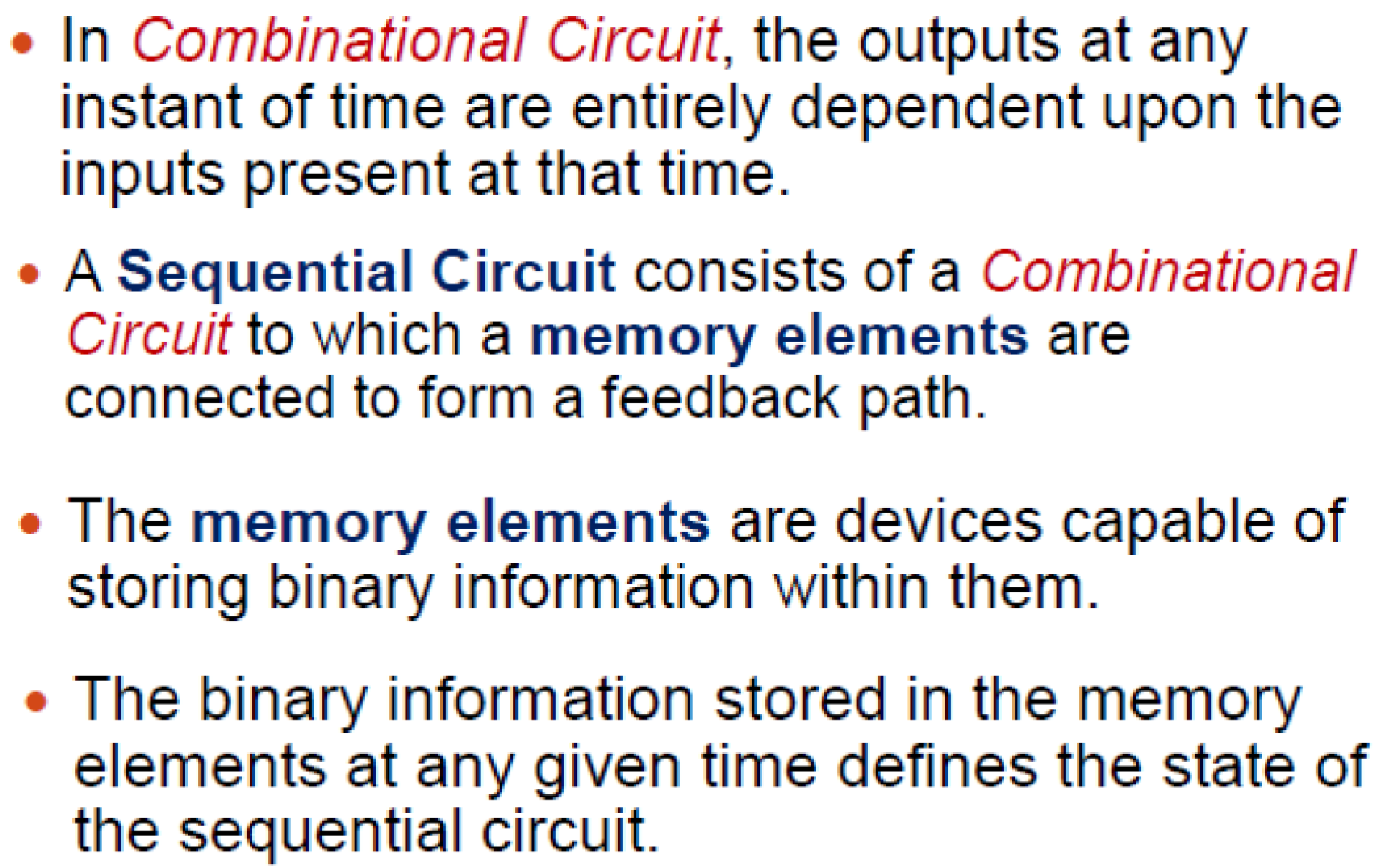
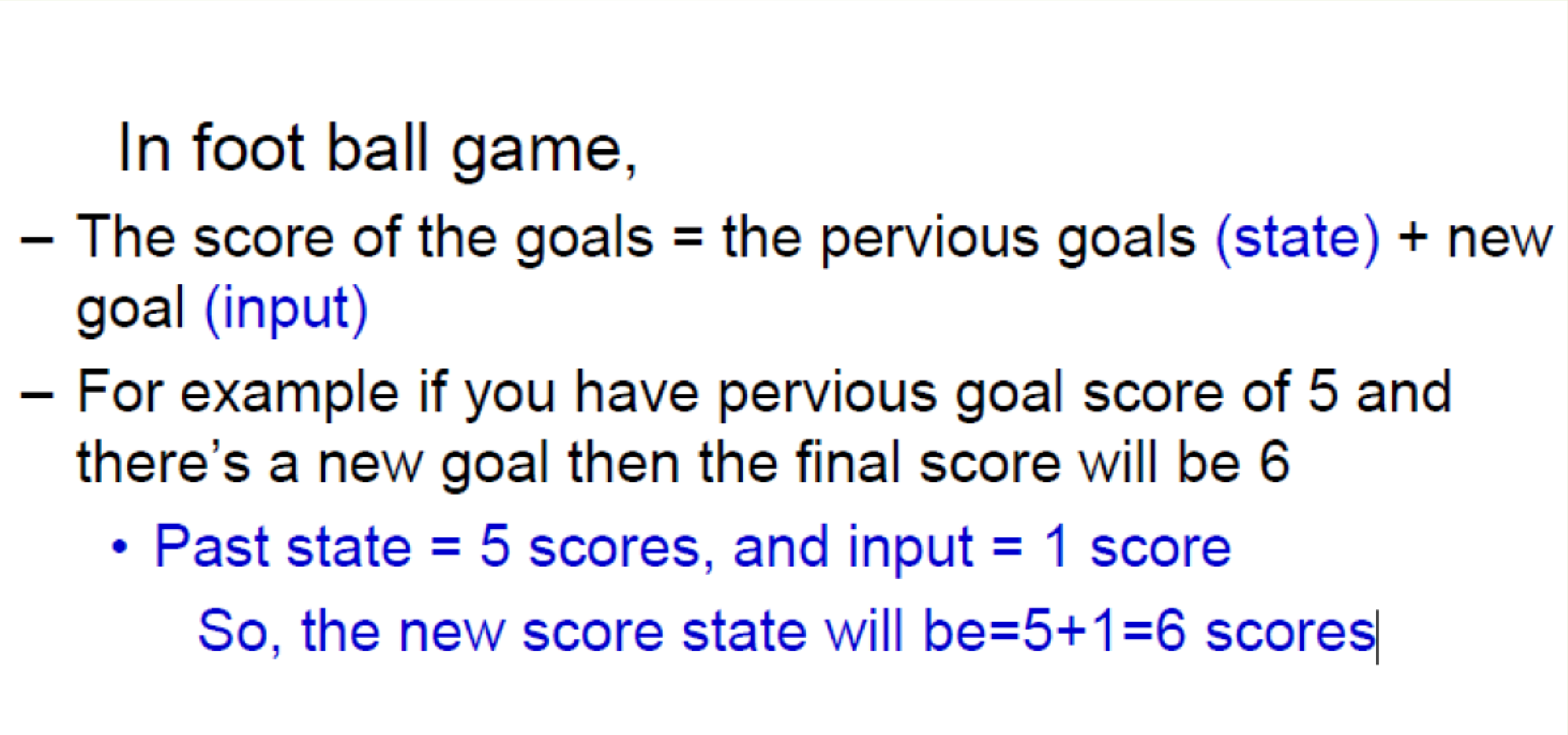
Sequential Logic



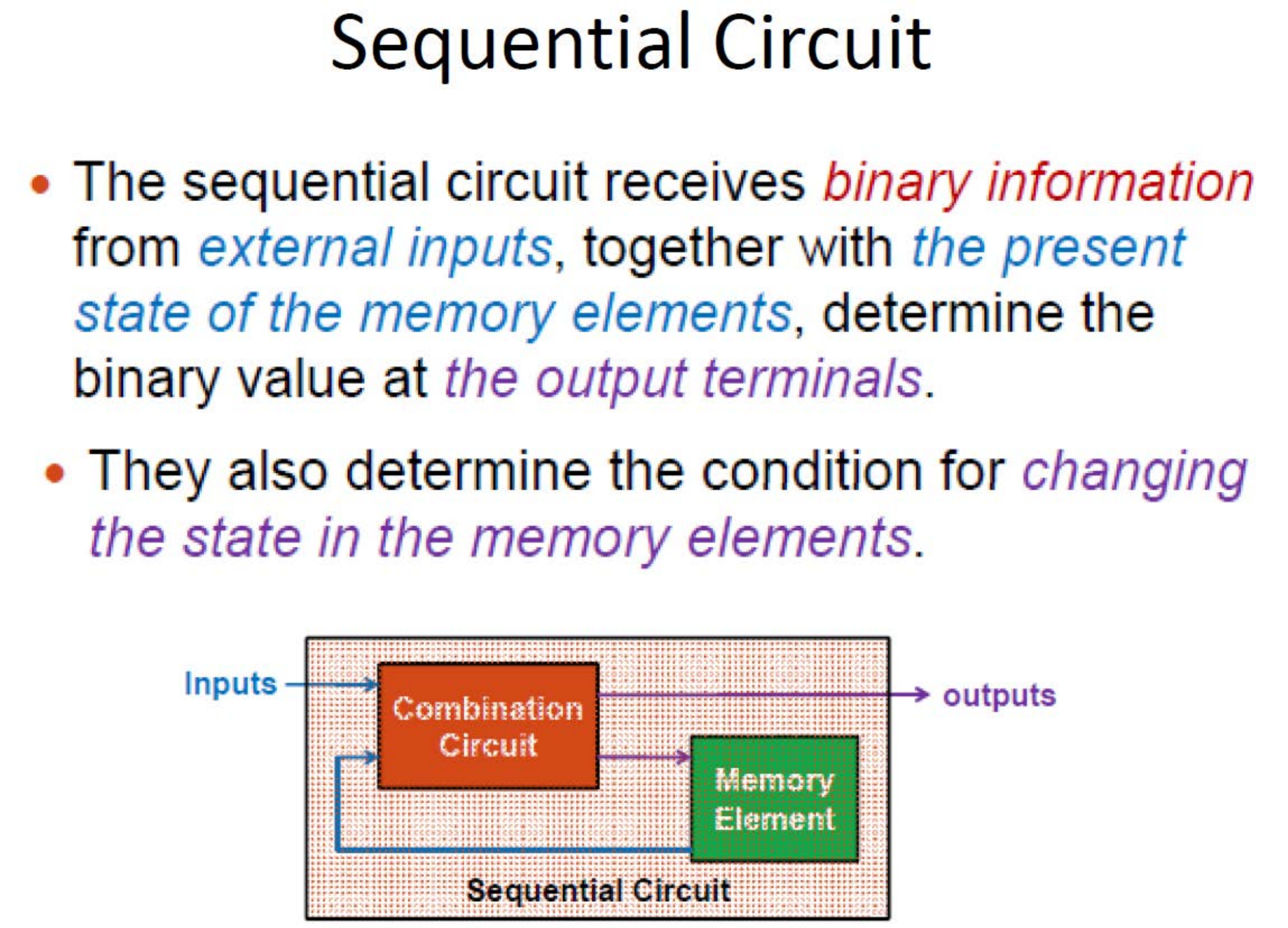
- In *Combinational Circuit*, the outputs at any instant of time are entirely dependent upon the inputs present at that time.
- A **Sequential Circuit** consists of a *Combinational Circuit* to which a **memory elements** are connected to form a feedback path.
- The **memory elements** are devices capable of storing binary information within them.
- The binary information stored in the memory elements at any given time defines the state of the sequential circuit.



In foot ball game,

- The score of the goals = the pervious goals (state) + new goal (input)
- For example if you have pervious goal score of 5 and there’s a new goal then the final score will be 6
  - Past state = 5 scores, and input = 1 score

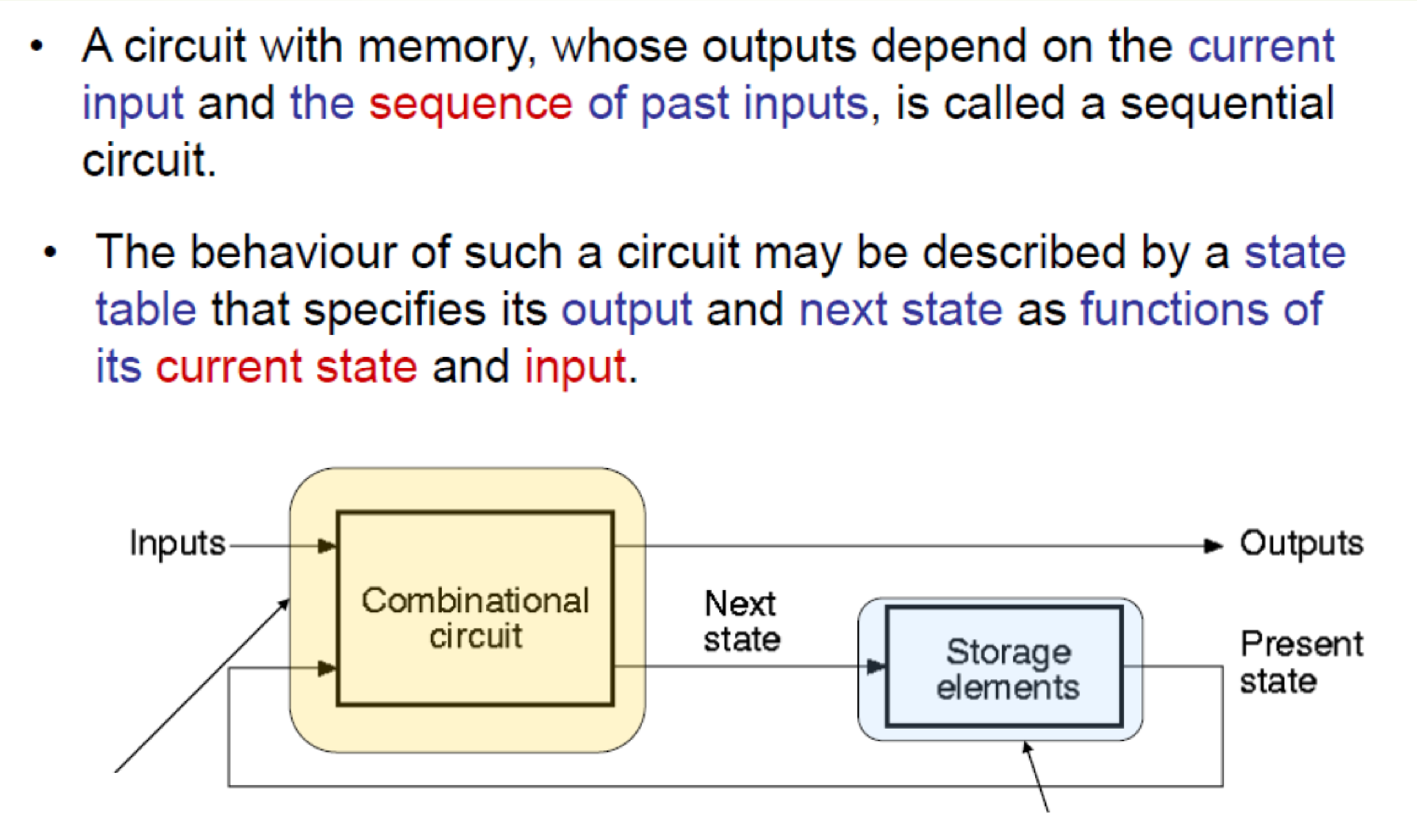
    So, the new score state will be=5+1=6 scores



Sequential Circuit

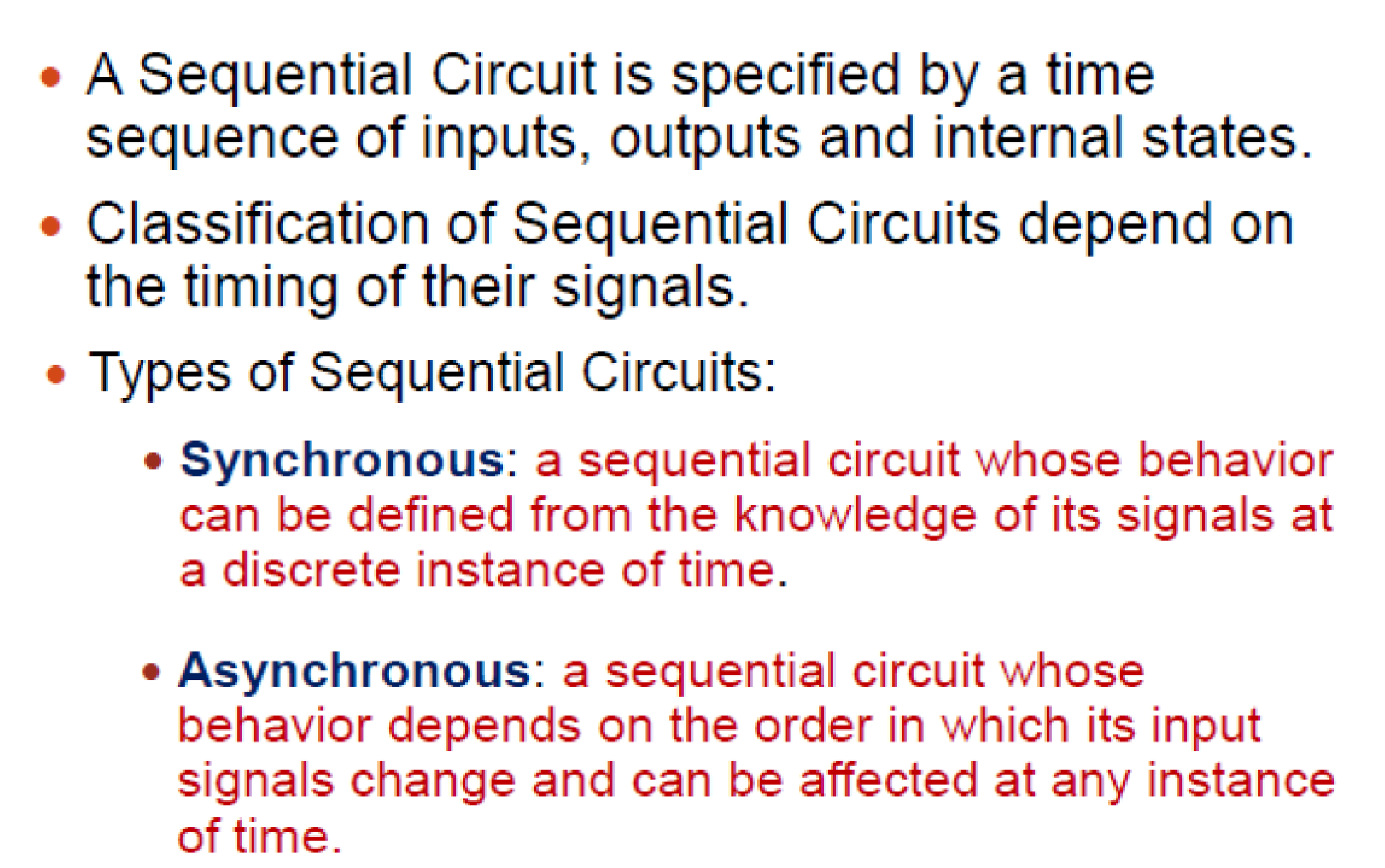
- The sequential circuit receives *binary information* from *external inputs*, together with *the present state of the memory elements*, determine the binary value at *the output terminals*.
- They also determine the condition for *changing the state in the memory elements*.

Inputs
Combination Circuit
outputs
Memory Element
Sequential Circuit

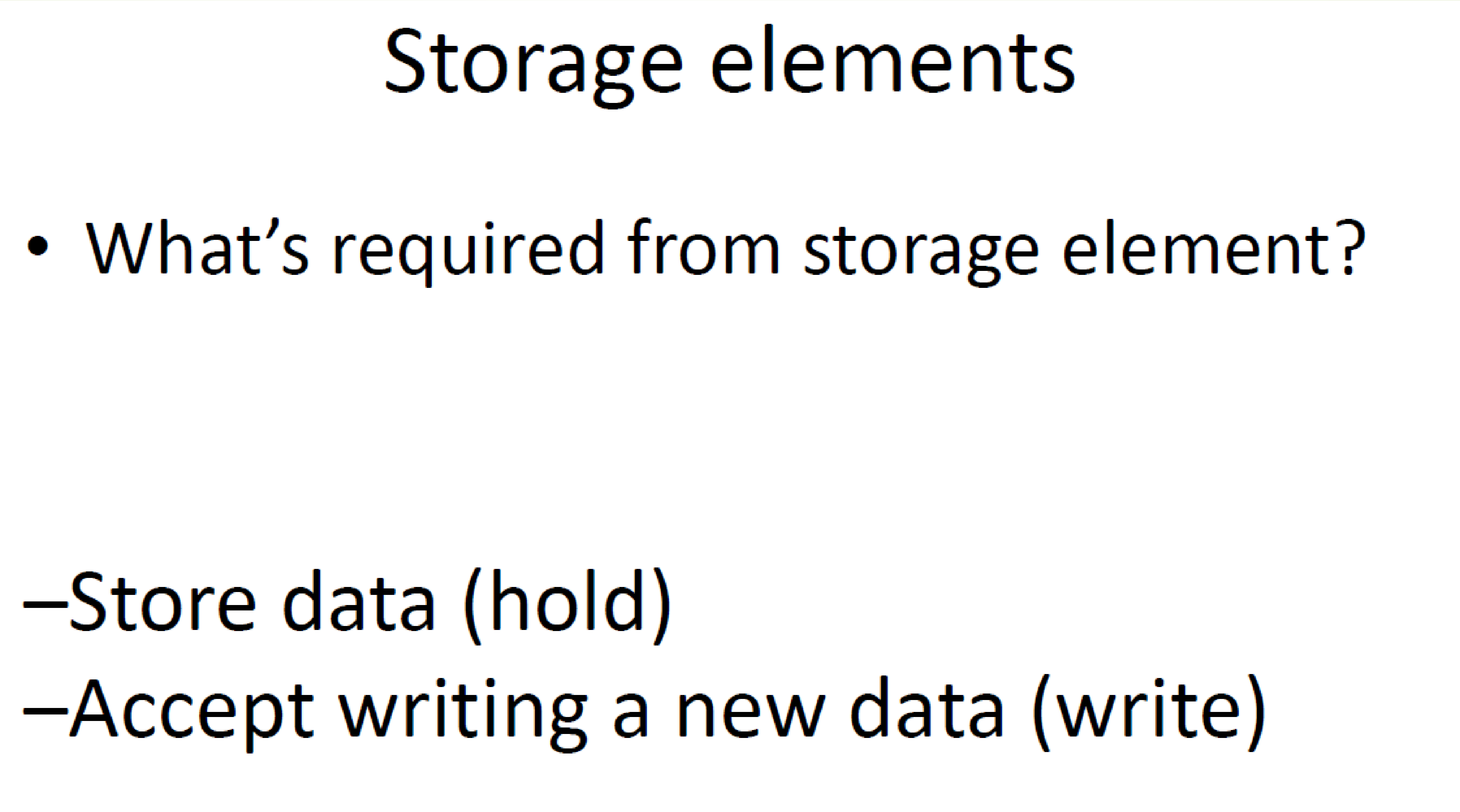


- A circuit with memory, whose outputs depend on the current input and the sequence of past inputs, is called a sequential circuit.
- The behaviour of such a circuit may be described by a state table that specifies its output and next state as functions of its current state and input.

Sequential Circuit



- A Sequential Circuit is specified by a time sequence of inputs, outputs and internal states.
- Classification of Sequential Circuits depend on the timing of their signals.
- Types of Sequential Circuits:
  - **Synchronous**: a sequential circuit whose behavior can be defined from the knowledge of its signals at a discrete instance of time.
  - **Asynchronous**: a sequential circuit whose behavior depends on the order in which its input signals change and can be affected at any instance of time.

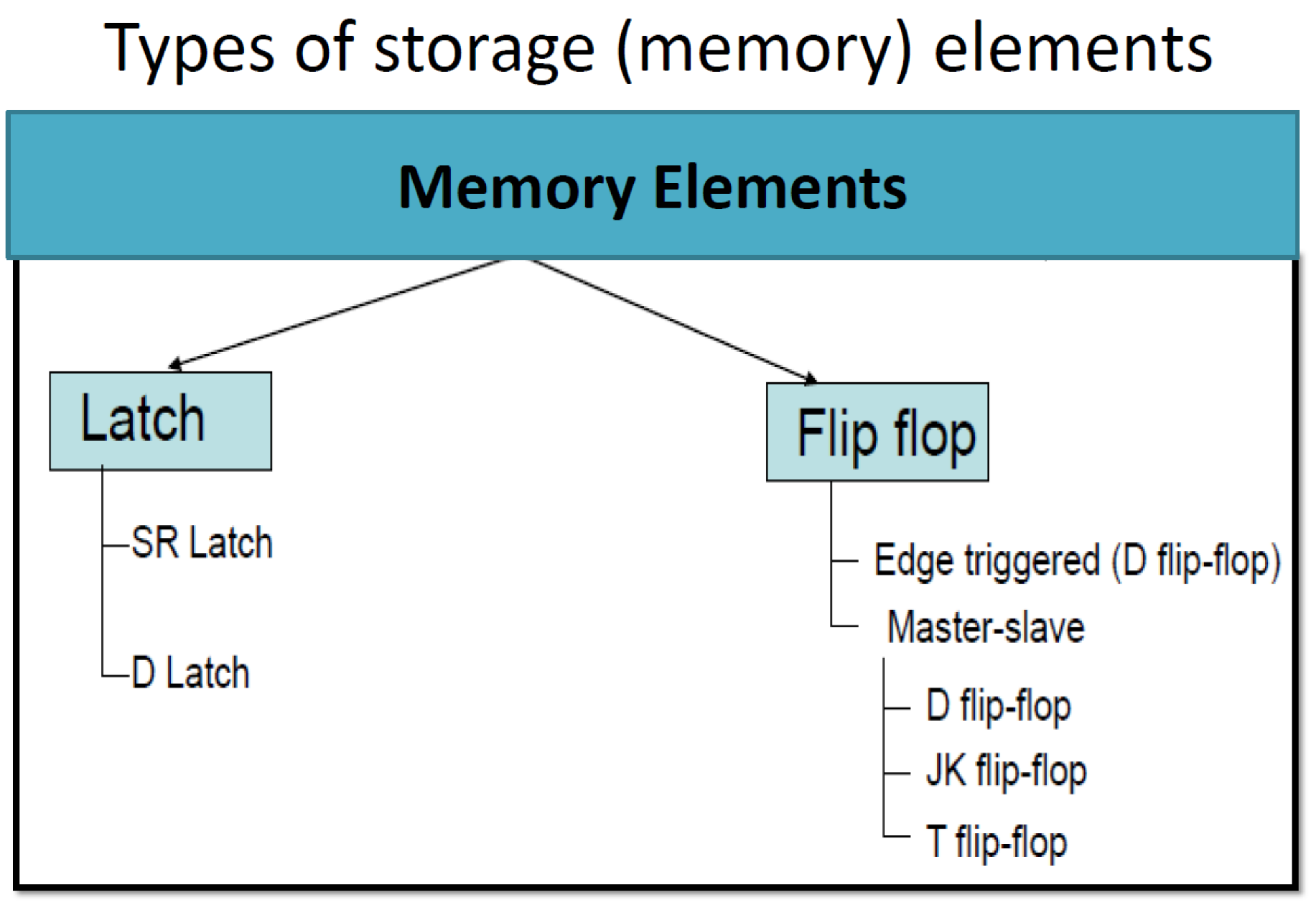


Storage elements

- What’s required from storage element?

–Store data (hold)

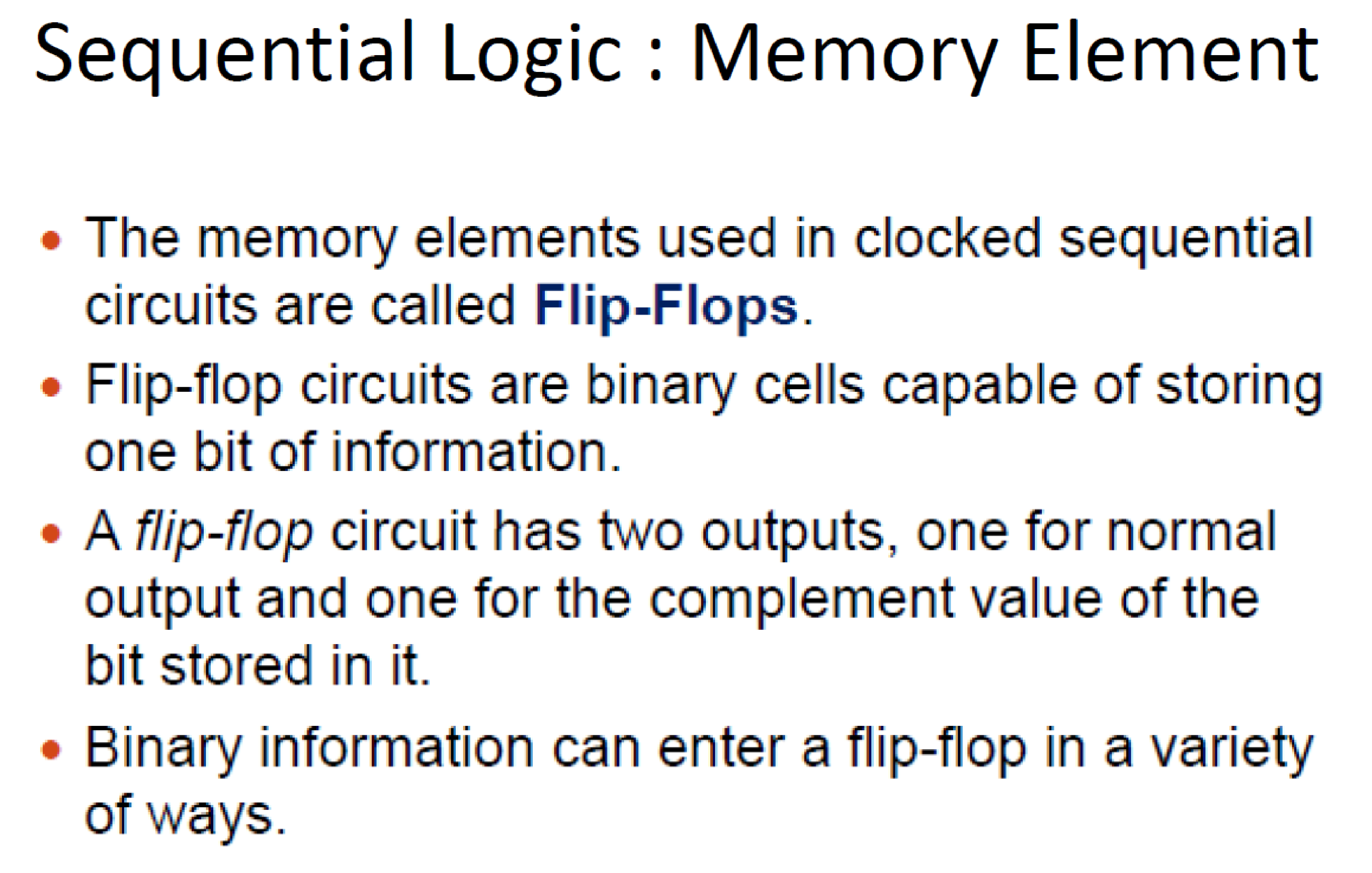
–Accept writing a new data (write)



Types of storage (memory) elements

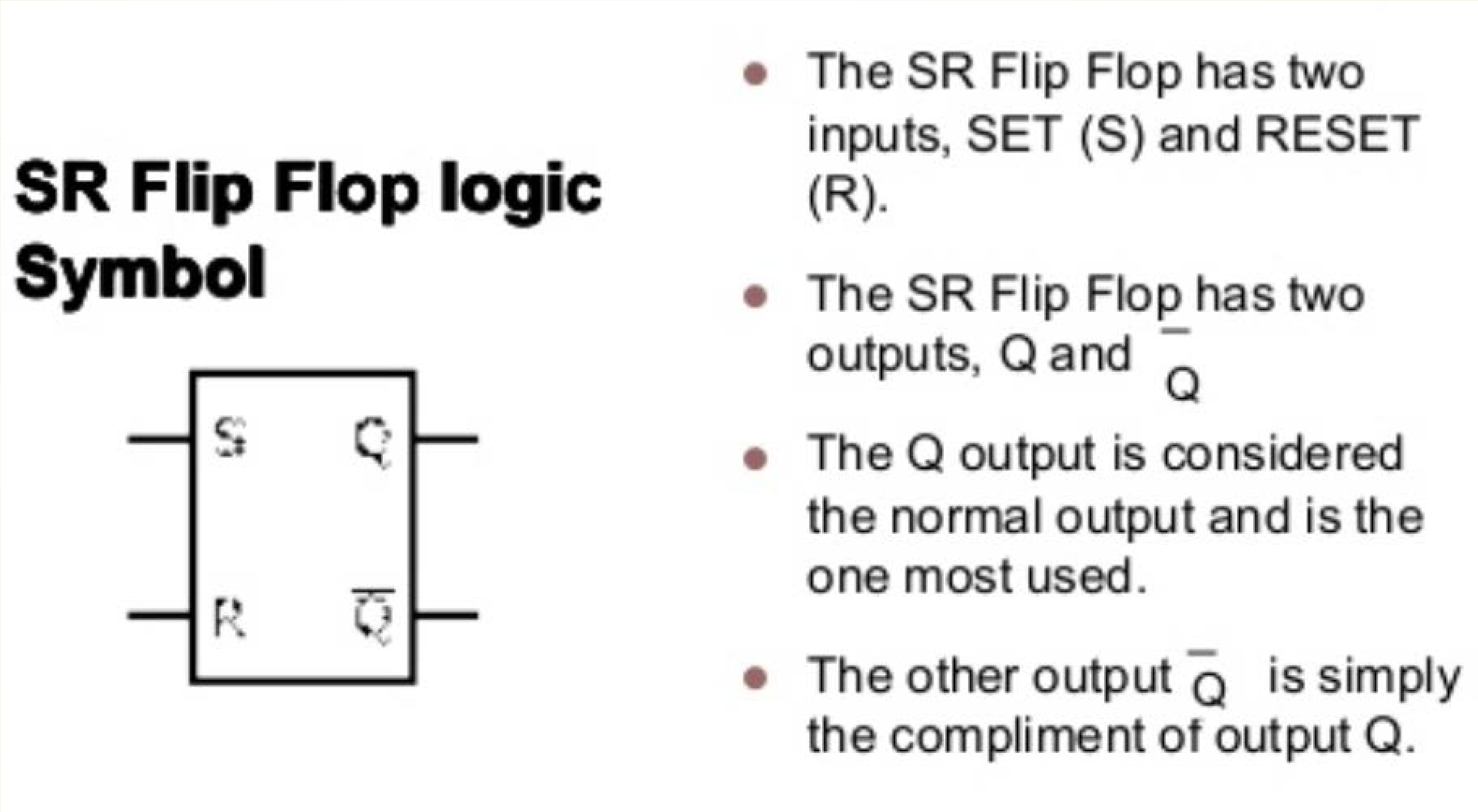
Memory Elements

- Latch
  - SR Latch
  - D Latch
- Flip flop
  - Edge triggered (D flip-flop)
  - Master-slave
    - D flip-flop
    - JK flip-flop
    - T flip-flop



Sequential Logic : Memory Element

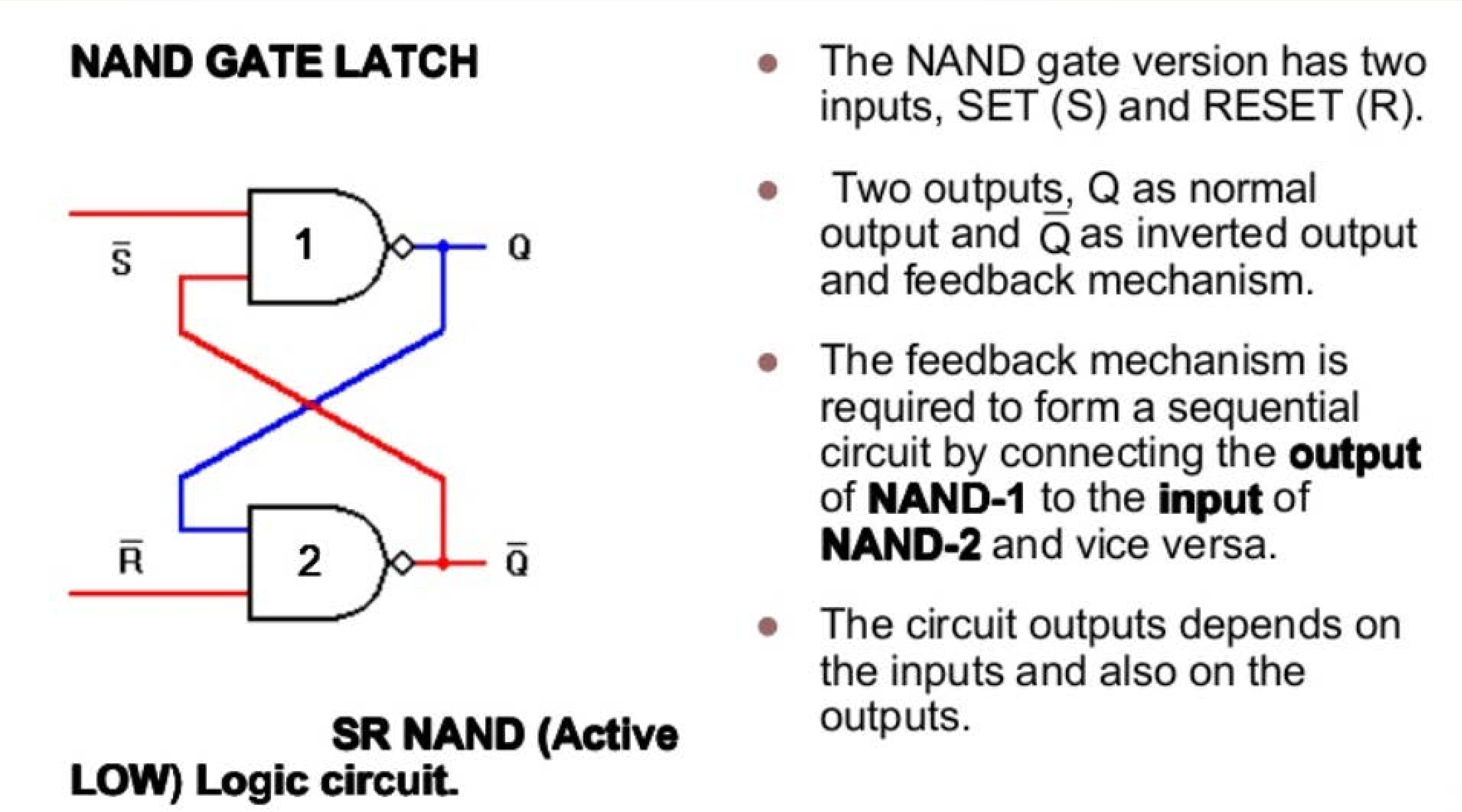
- The memory elements used in clocked sequential circuits are called **Flip-Flops**.
- Flip-flop circuits are binary cells capable of storing one bit of information.
- A *flip-flop* circuit has two outputs, one for normal output and one for the complement value of the bit stored in it.
- Binary information can enter a flip-flop in a variety of ways.



SR Flip Flop logic Symbol

S
Q
R
$\overline{Q}$

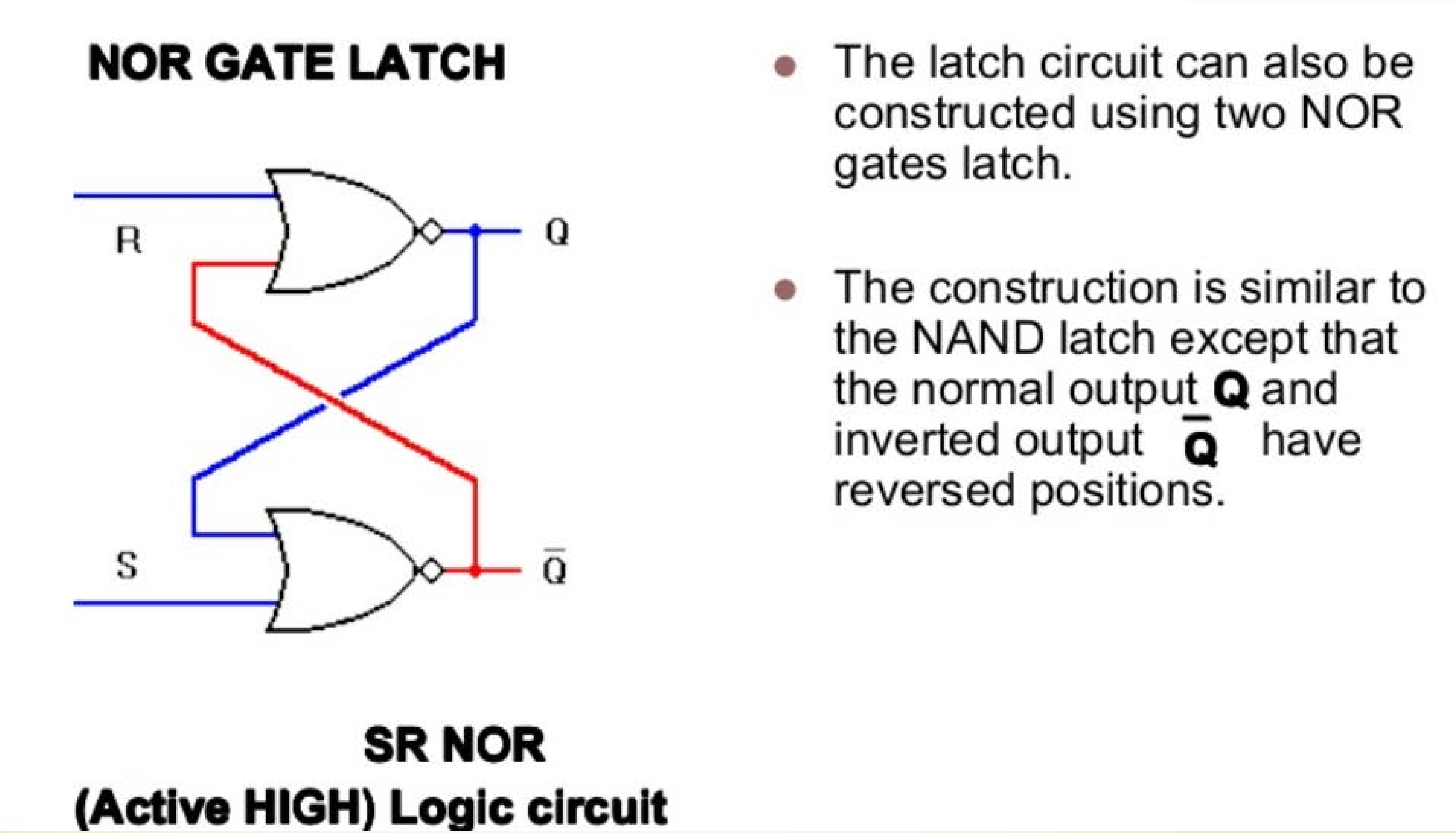
- The SR Flip Flop has two inputs, SET (S) and RESET (R).
- The SR Flip Flop has two outputs, Q and $\overline{Q}$
- The Q output is considered the normal output and is the one most used.
- The other output $\overline{Q}$ is simply the compliment of output Q.



NAND GATE LATCH

SR NAND (Active LOW) Logic circuit.

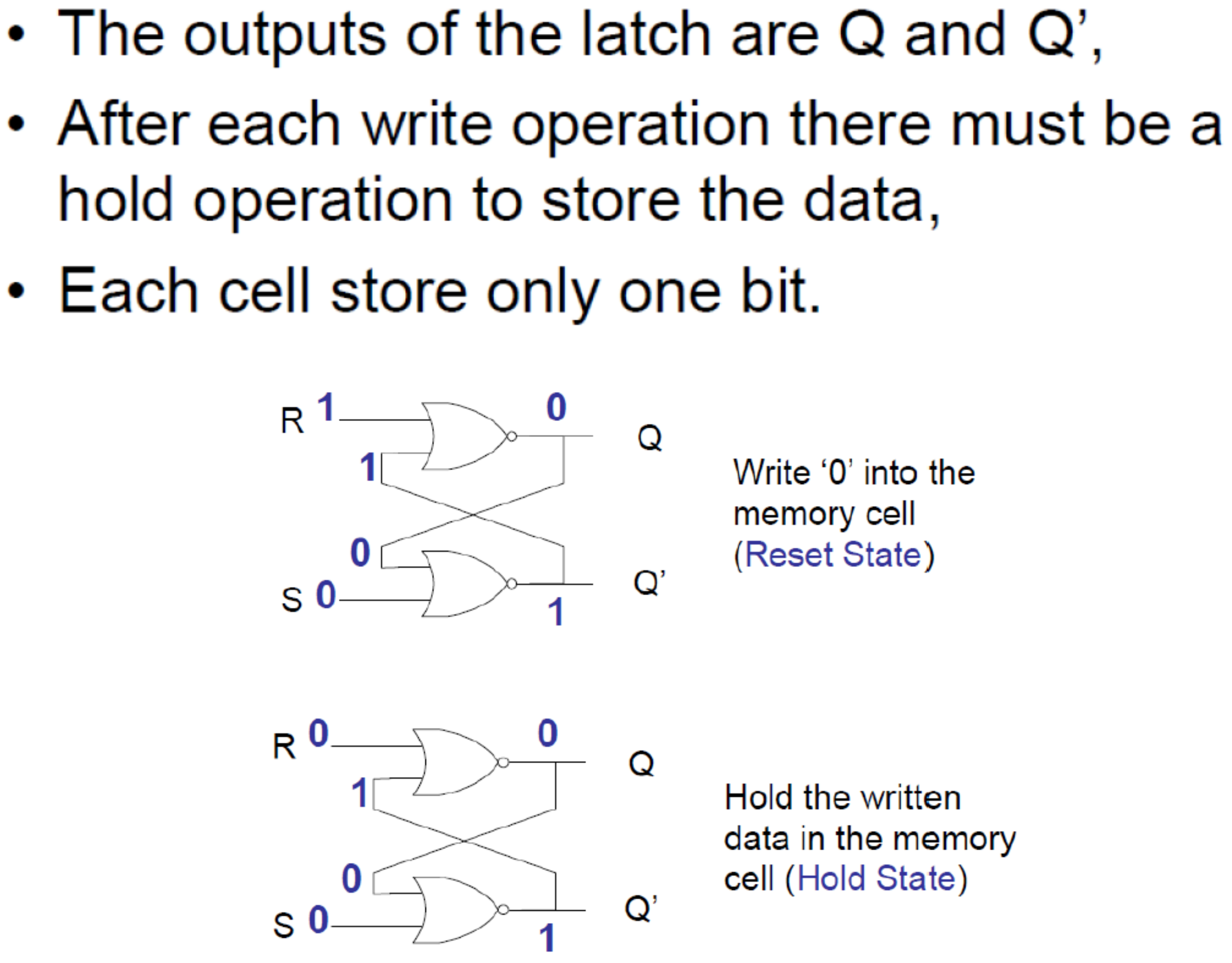
- The NAND gate version has two inputs, SET (S) and RESET (R).
- Two outputs, Q as normal output and $\overline{Q}$ as inverted output and feedback mechanism.
- The feedback mechanism is required to form a sequential circuit by connecting the **output** of **NAND-1** to the **input** of **NAND-2** and vice versa.
- The circuit outputs depends on the inputs and also on the outputs.



NOR GATE LATCH

SR NOR
(Active HIGH) Logic circuit

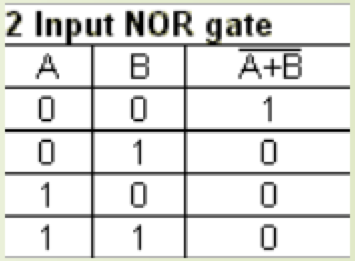
- The latch circuit can also be constructed using two NOR gates latch.
- The construction is similar to the NAND latch except that the normal output **Q** and inverted output $\overline{\mathbf{Q}}$ have reversed positions.



- The outputs of the latch are Q and Q’,
- After each write operation there must be a hold operation to store the data,
- Each cell store only one bit.

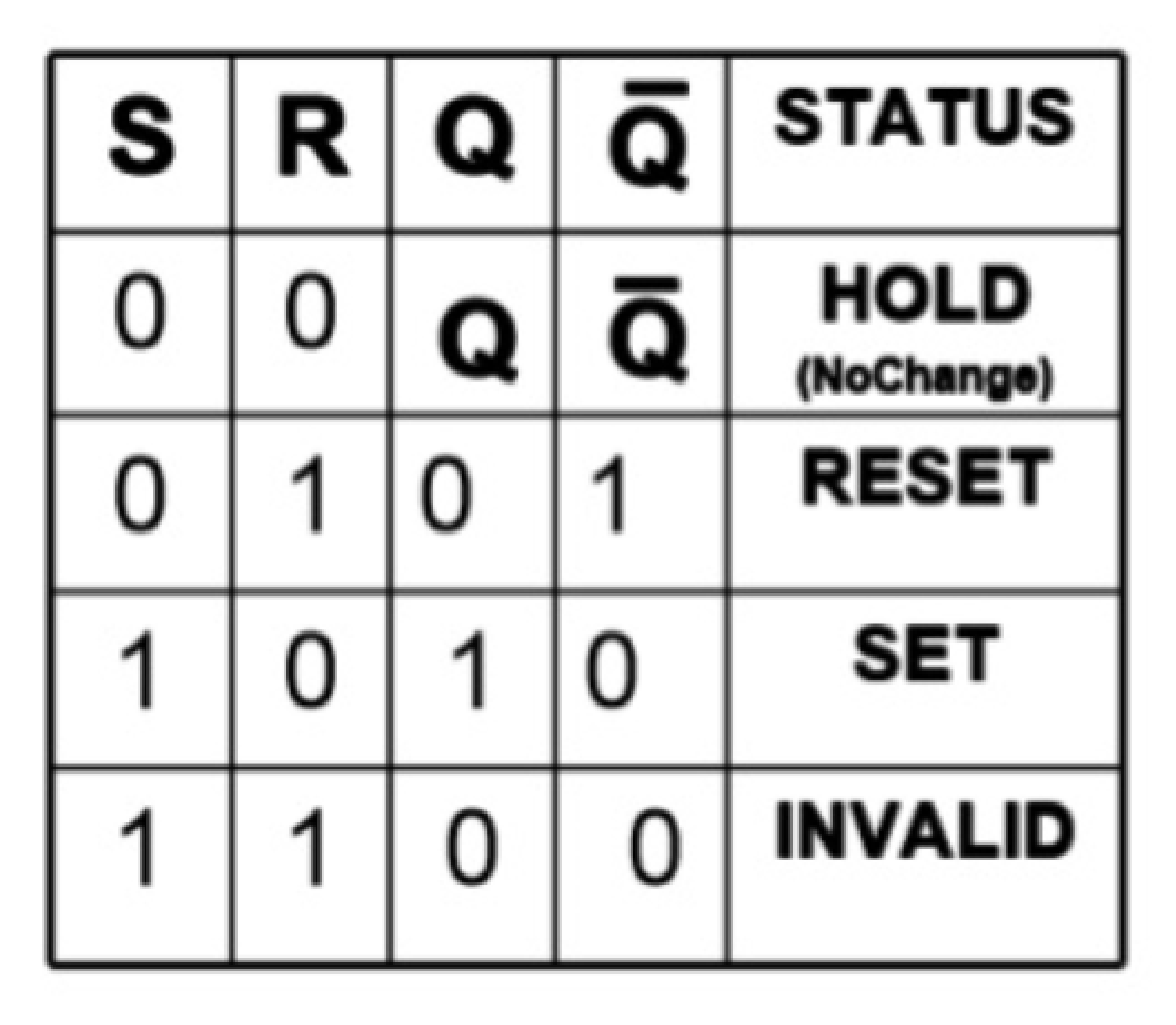
Write ‘0’ into the memory cell (Reset State)

Hold the written data in the memory cell (Hold State)



2 Input NOR gate

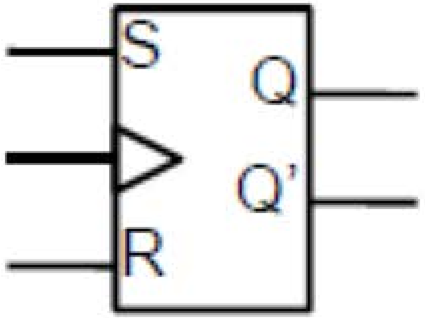
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

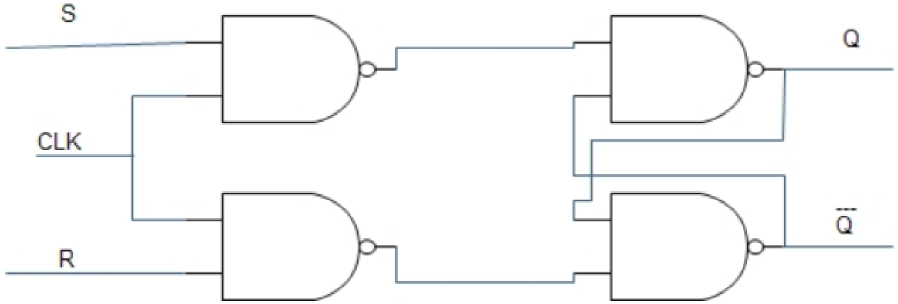


S	R	Q	$\bar{Q}$	STATUS
0	0	Q	$\bar{Q}$	HOLD (NoChange)
0	1	0	1	RESET
1	0	1	0	SET
1	1	0	0	INVALID

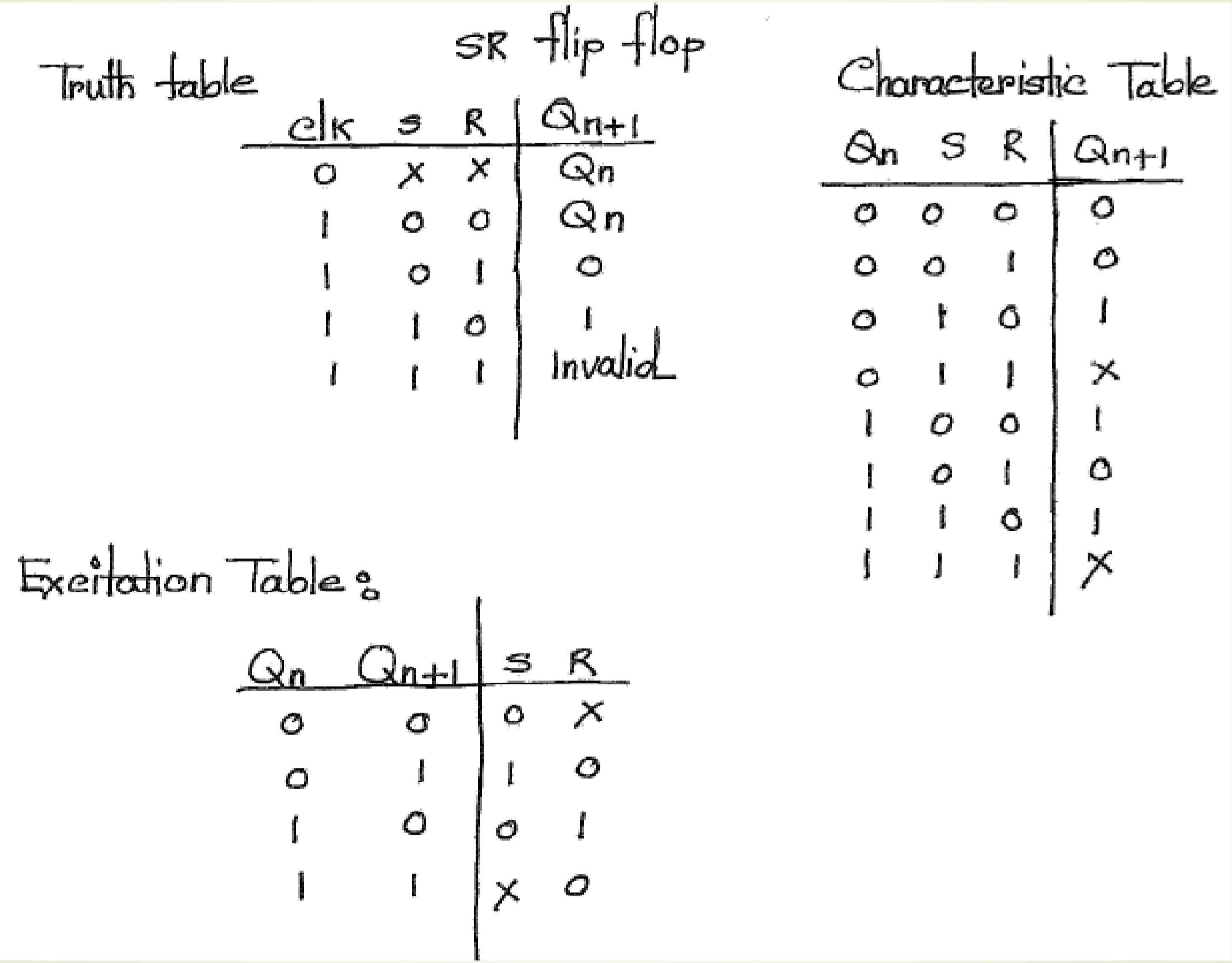
Clocked RS Flip-Flop

- The graphic symbol for Clocked RS flip-flop is





- The Clock Pulse is marked with small triangle.
- The triangle is a symbol for ***dynamic indicator*** and denotes the fact that the flip-flop responds to an input clock transition from a low-level (binary 0) to a high-level (binary 1)



SR flip flop

Truth table

clk	S	R	Q_{n+1}
0	X	X	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	Invalid

Characteristic Table

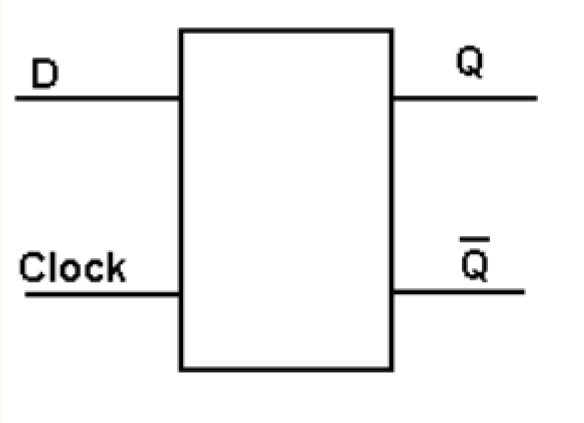
Q_n	S	R	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

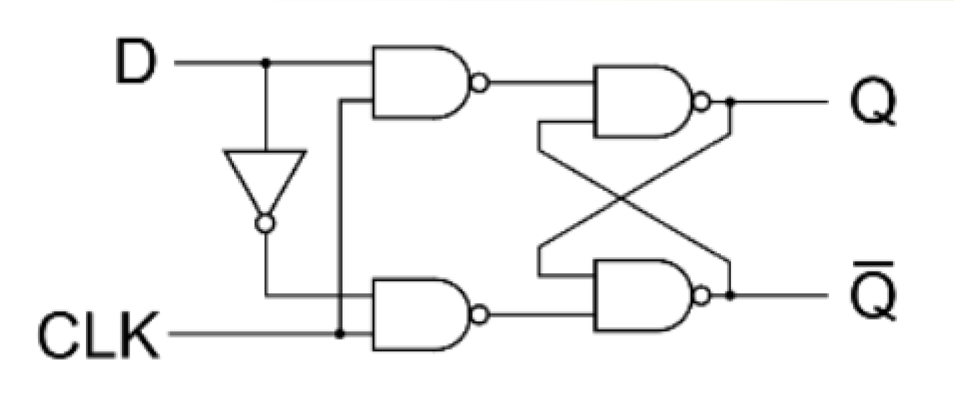
Exc̆itation Table:

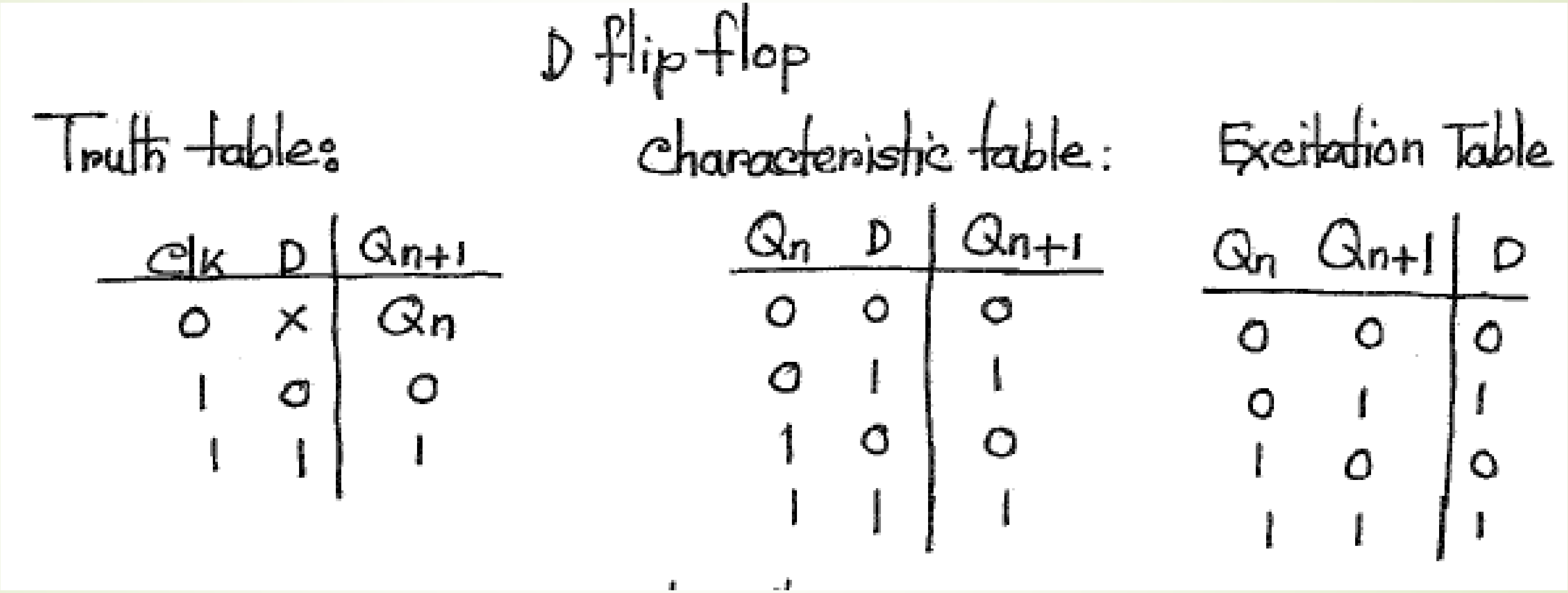
Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

D Flip Flop

A D flip-flop is a modification of the Clocked RS flip-flop.







D flip flop

Truth table:

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

characteristic table:

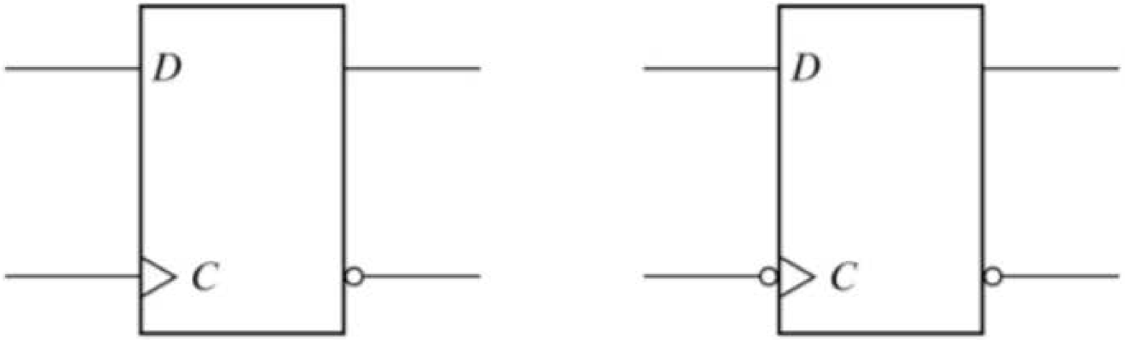
Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

Excitation Table

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

Positive and Negative Edge D Flip-Flop

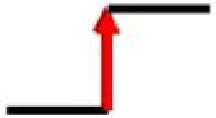
- D flops can be triggered on positive or negative edge
- Bubble before *Clock (C)* input indicates negative edge trigger



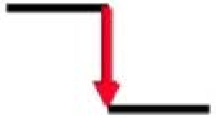
(a) Positive-edge

(a) Negative-edge

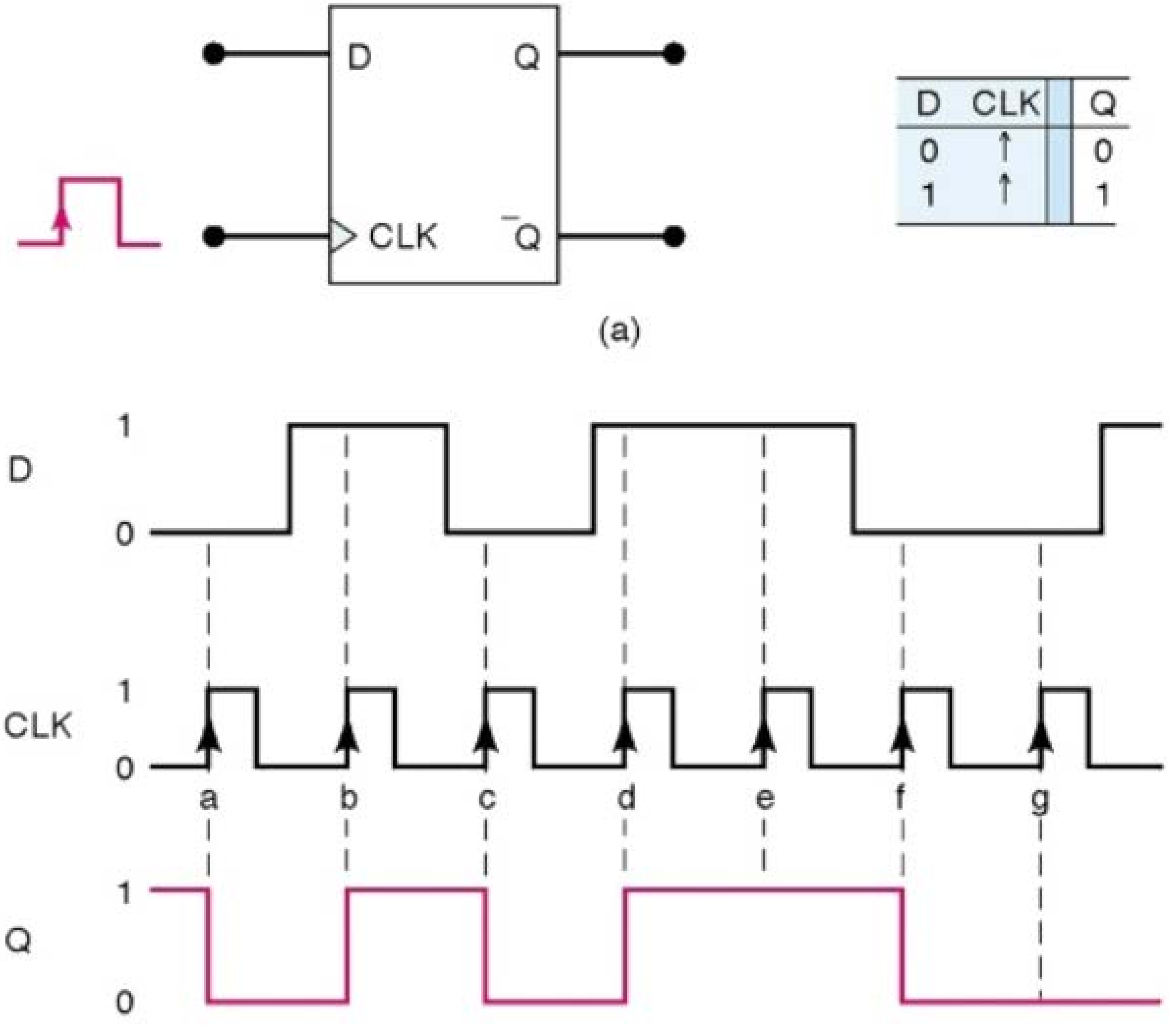
Fig. 5-11 Graphic Symbol for Edge-Triggered *D* Flip-Flop



Lo-Hi edge



Hi-Lo edge



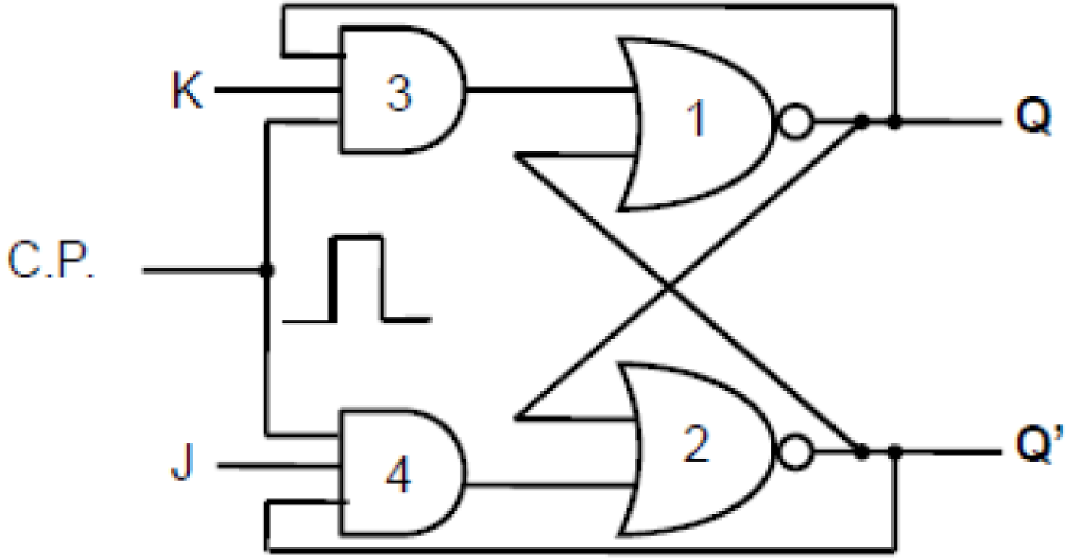
D	CLK	Q
0	↑	0
1	↑	1

(a)

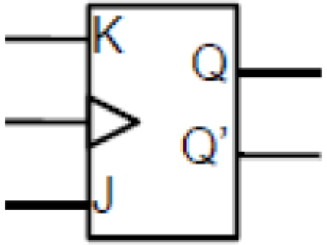
(b)

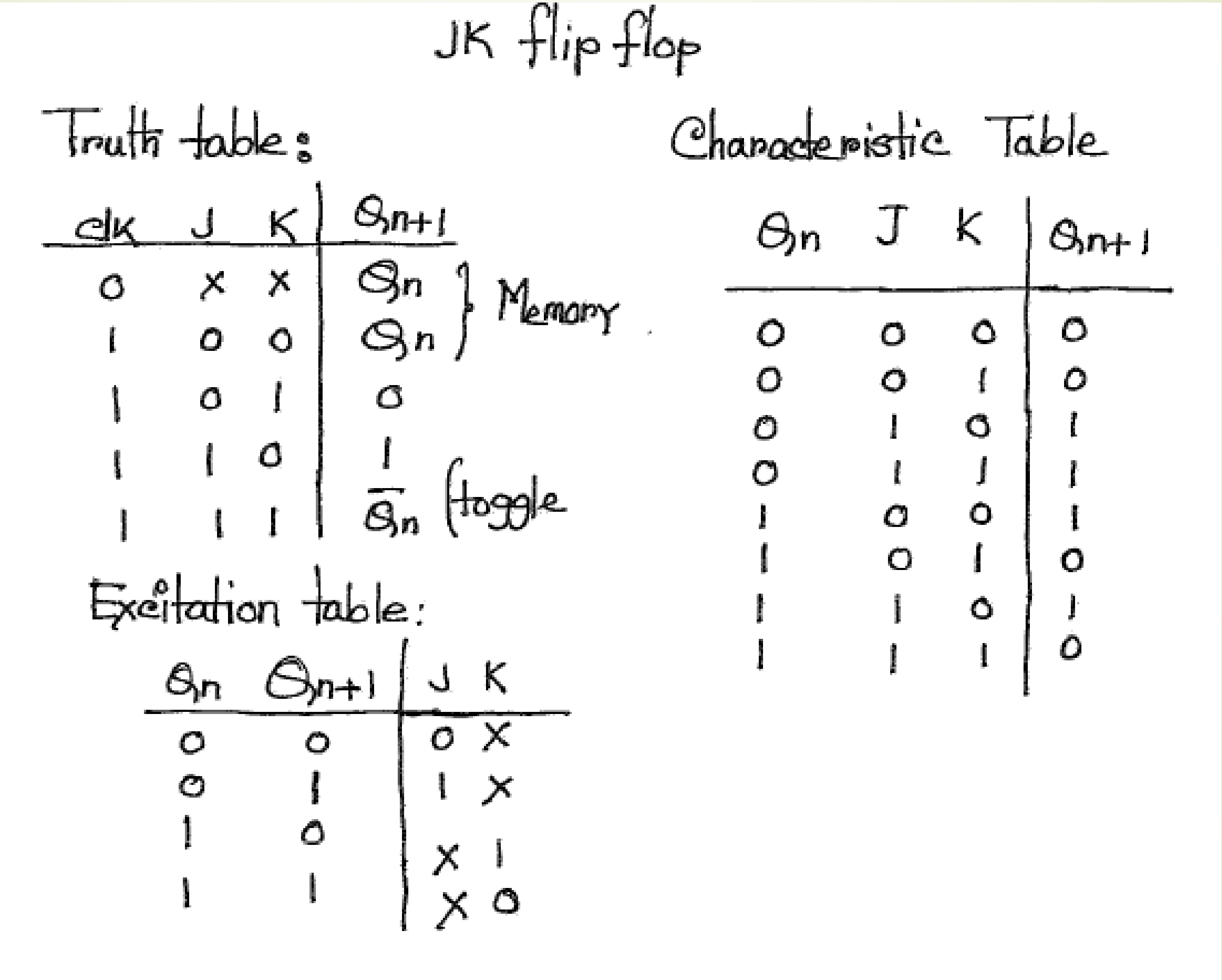
JK Flip Flop

JK Flip-Flop



Graphic Symbol





JK flip flop

Truth table:

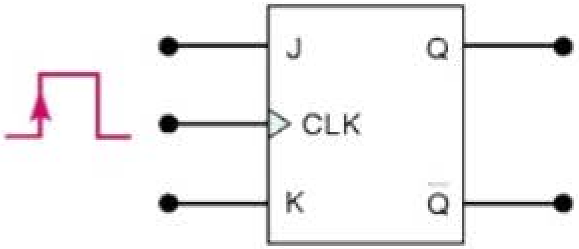
clk	J	K	Q_{n+1}	
0	x	x	Q_n	} Memory
1	0	0	Q_n	
1	0	1	0	
1	1	0	1	
1	1	1	$\bar{Q}_n$	(toggle

Excitation table:

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

Characteristic Table

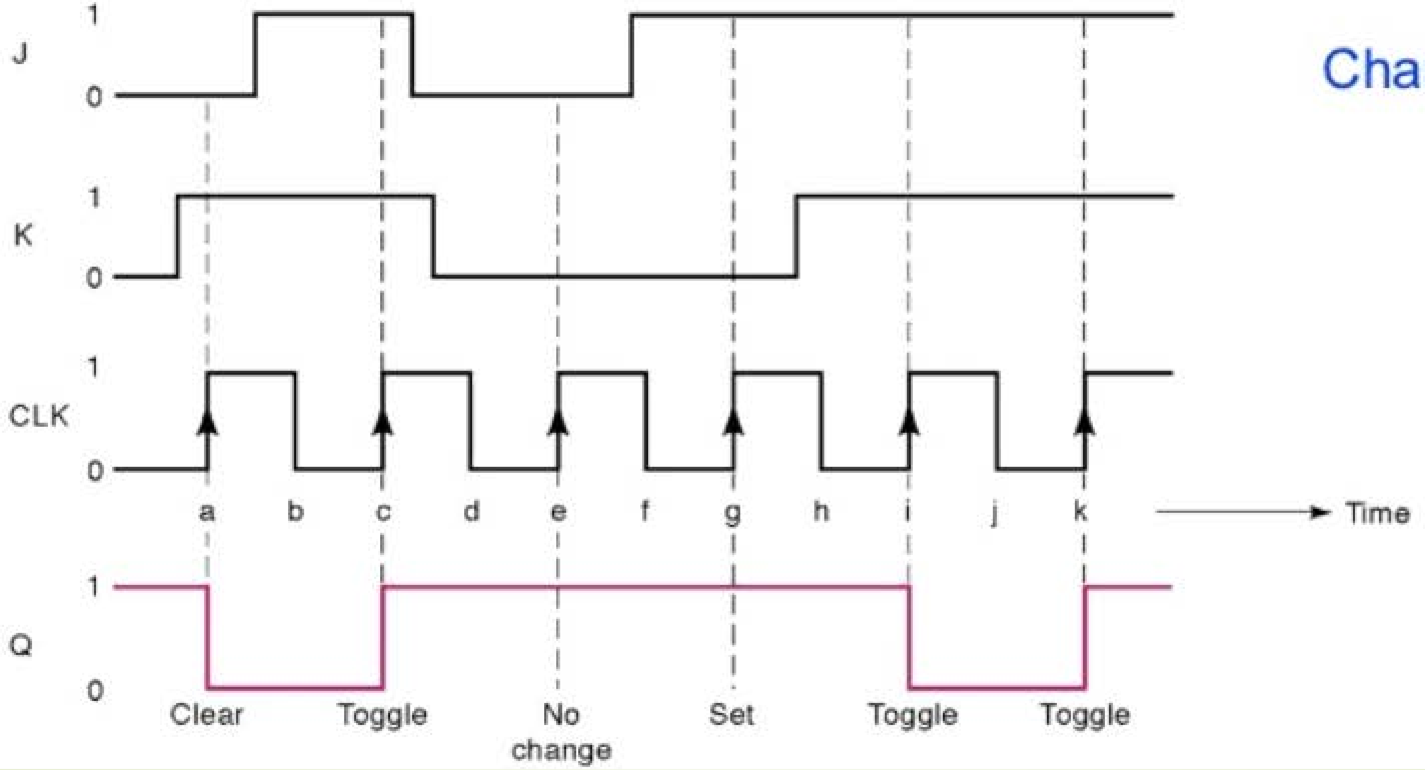
Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

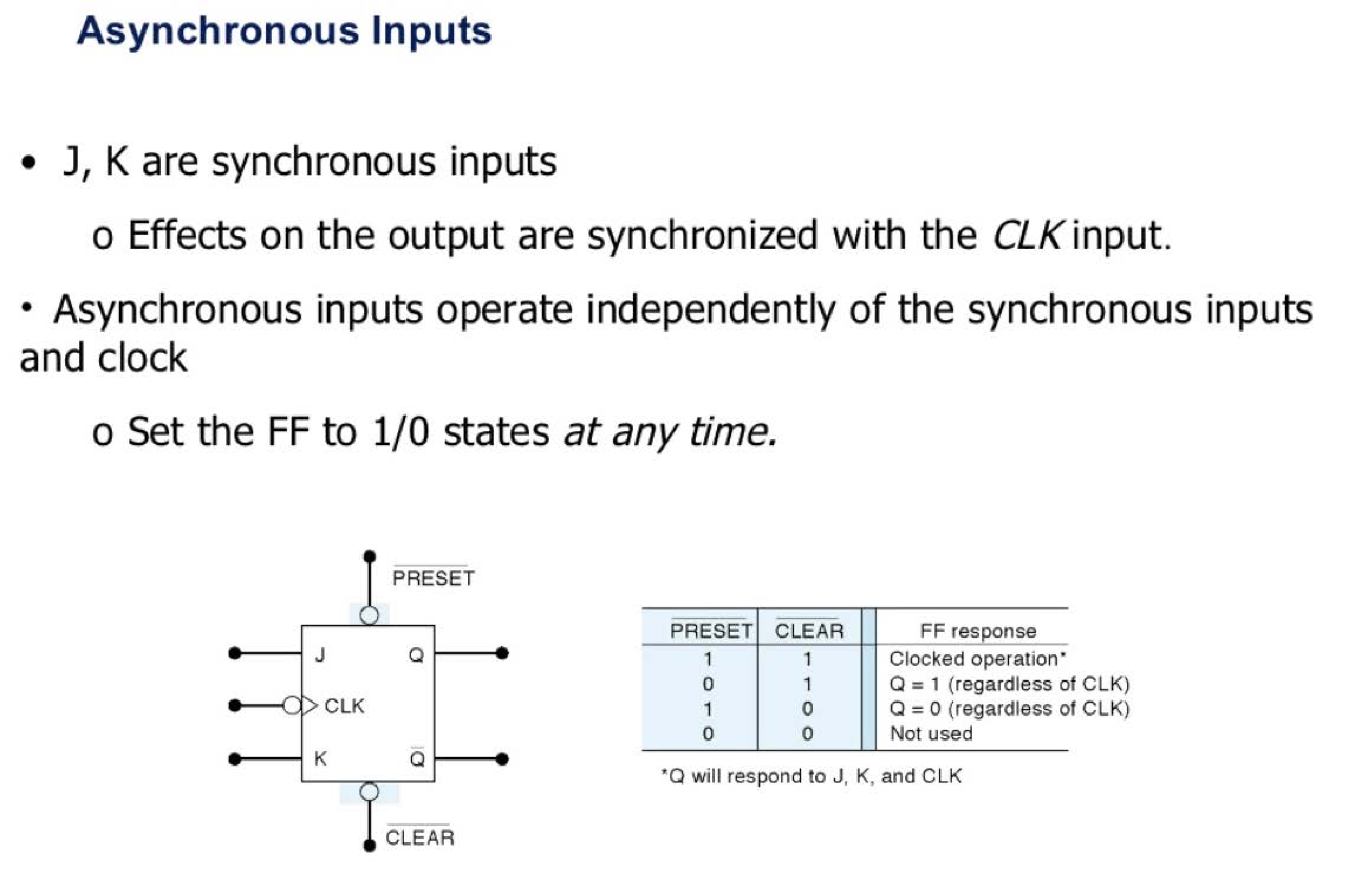


J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q}_0$ (toggles)

(a)

Characteristic Table





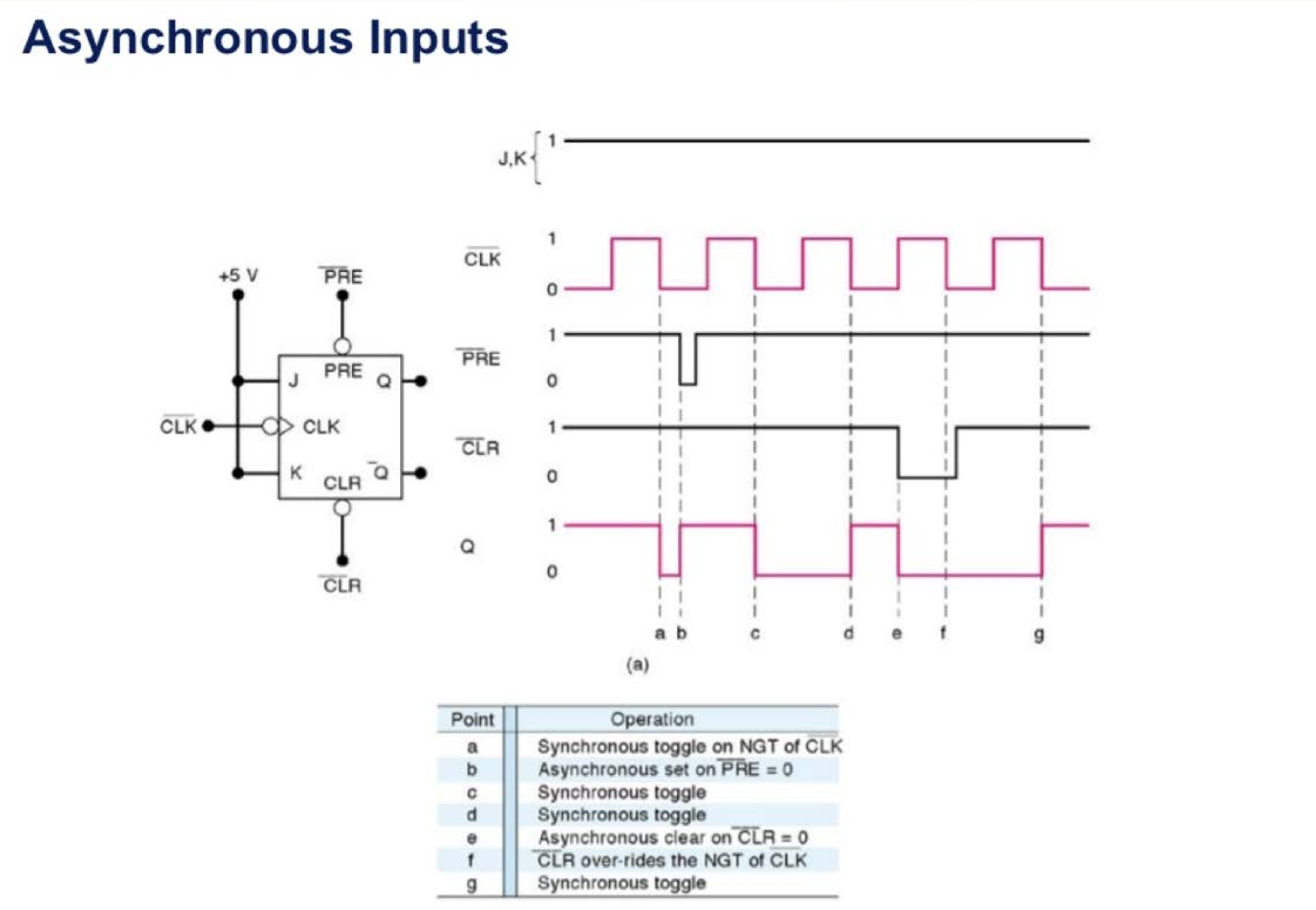
Asynchronous Inputs

- J, K are synchronous inputs
  - o Effects on the output are synchronized with the *CLK* input.
- Asynchronous inputs operate independently of the synchronous inputs and clock
  - o Set the FF to 1/0 states *at any time.*

$\overline{\text{PRESET}}$
J
Q
CLK
K
$\overline{Q}$
$\overline{\text{CLEAR}}$

$\overline{\text{PRESET}}$	$\overline{\text{CLEAR}}$	FF response
1	1	Clocked operation*
0	1	Q = 1 (regardless of CLK)
1	0	Q = 0 (regardless of CLK)
0	0	Not used

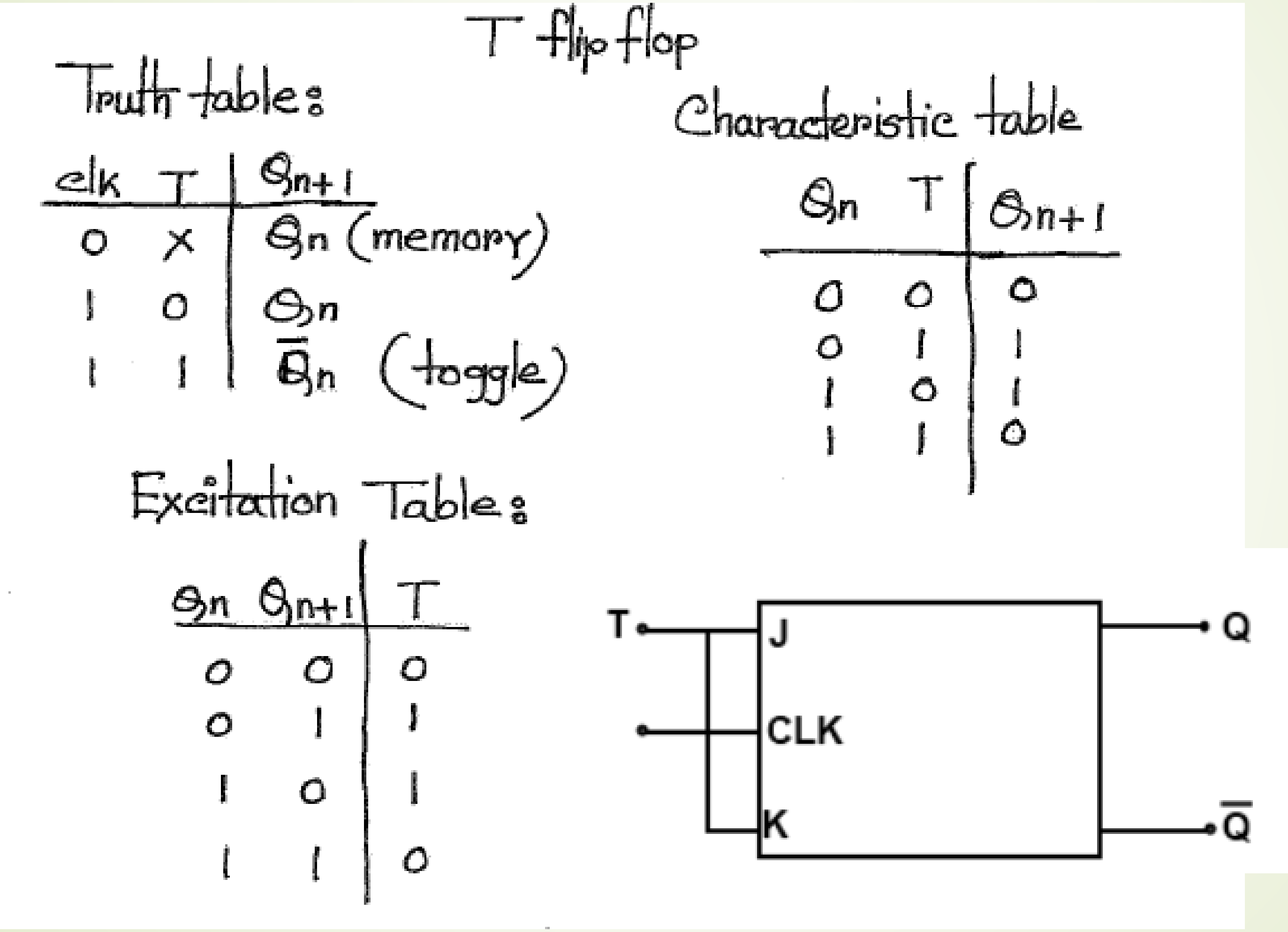
*Q will respond to J, K, and CLK



Asynchronous Inputs

(a)

Point	Operation
a	Synchronous toggle on NGT of $\overline{\text{CLK}}$
b	Asynchronous set on $\overline{\text{PRE}}$ = 0
c	Synchronous toggle
d	Synchronous toggle
e	Asynchronous clear on $\overline{\text{CLR}}$ = 0
f	$\overline{\text{CLR}}$ over-rides the NGT of $\overline{\text{CLK}}$
g	Synchronous toggle



T flip flop

Truth table:

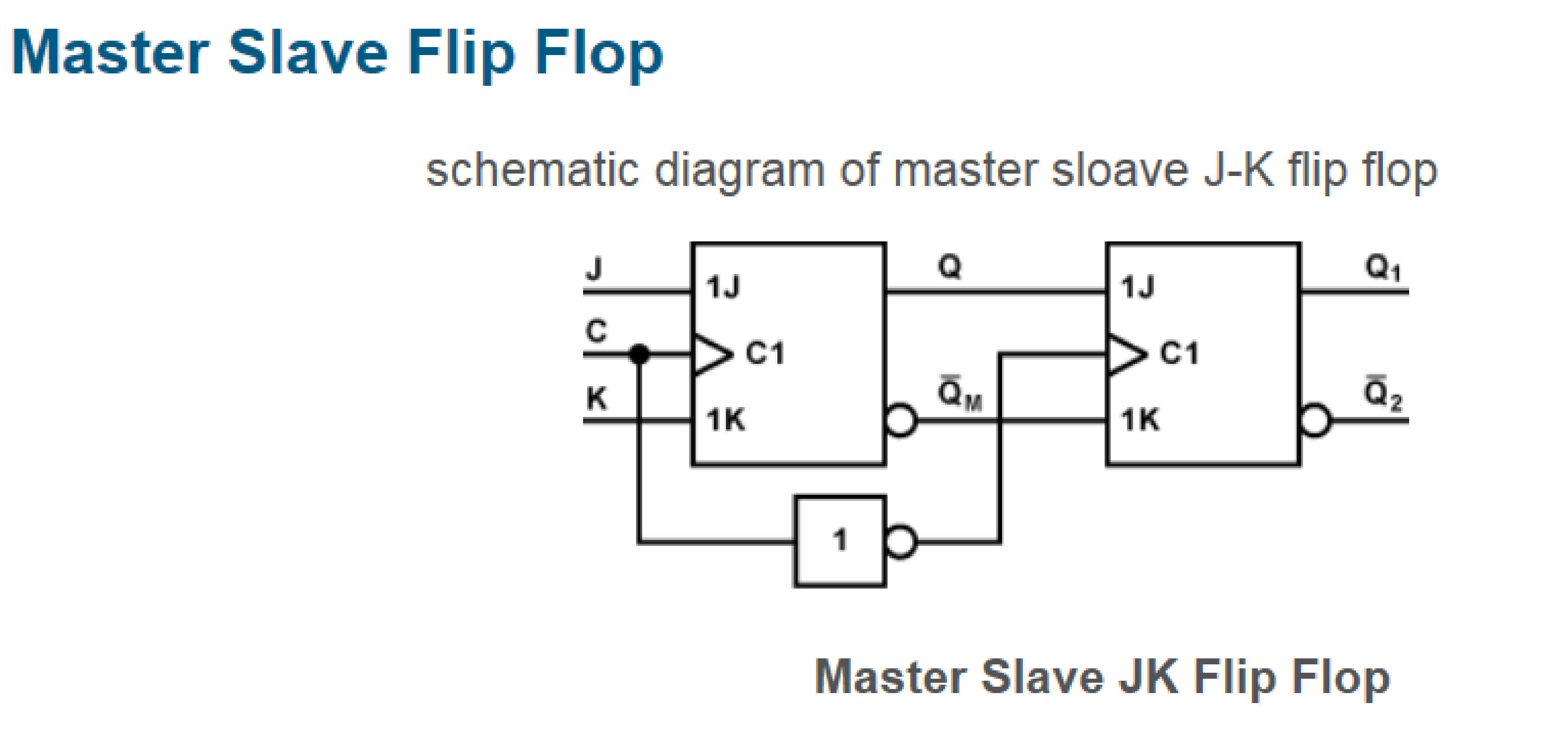
clk	T	Q_{n+1}
0	X	Q_n (memory)
1	0	Q_n
1	1	$\bar{Q}_n$ (toggle)

Characteristic table

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

Excitation Table:

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

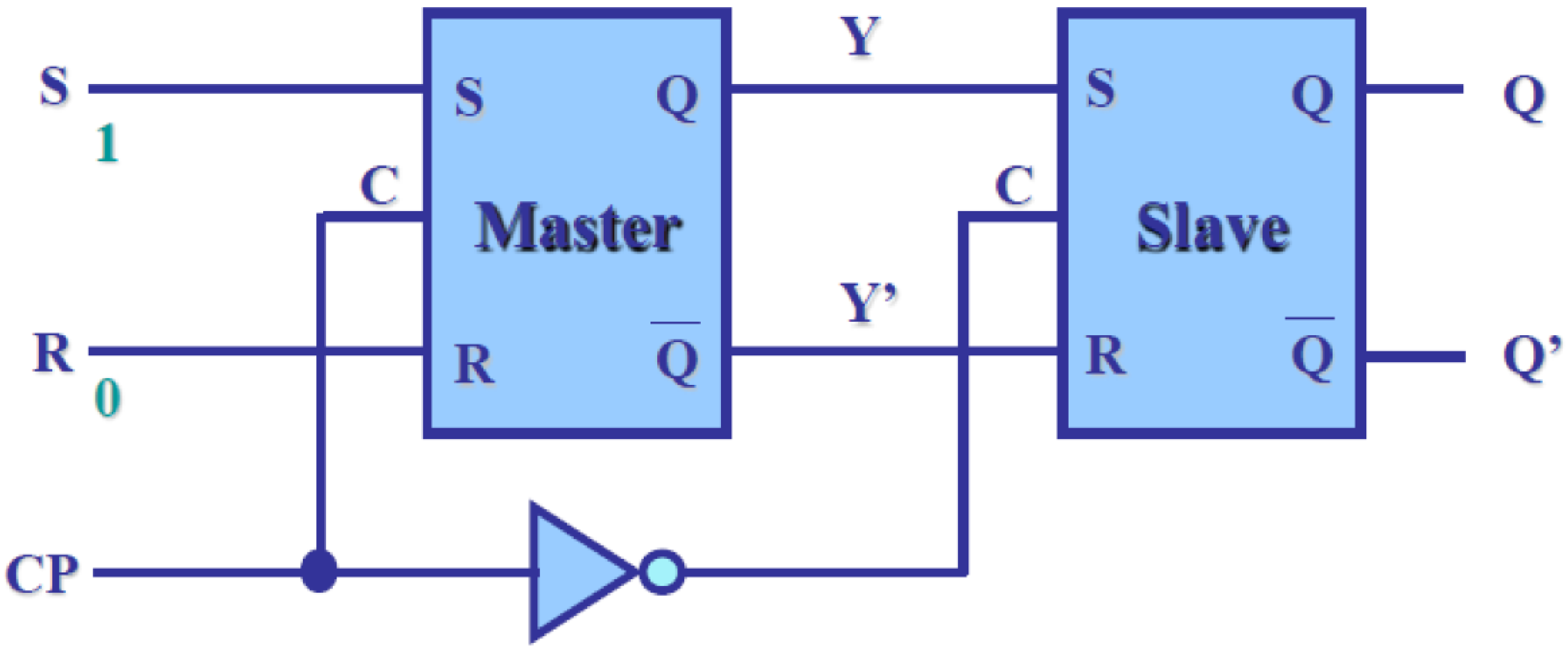


Master Slave Flip Flop

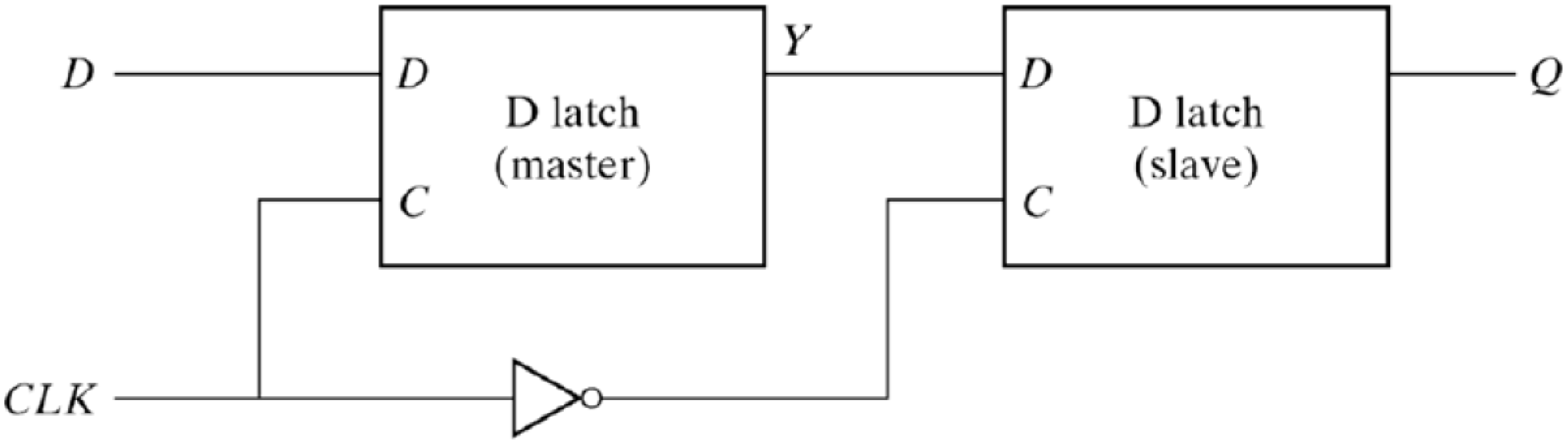
schematic diagram of master sloave J-K flip flop

Master Slave JK Flip Flop

Master slave using SR Latch



Master-Slave using D Flip-Flop



Thanks to Ali Mustafa for the slides