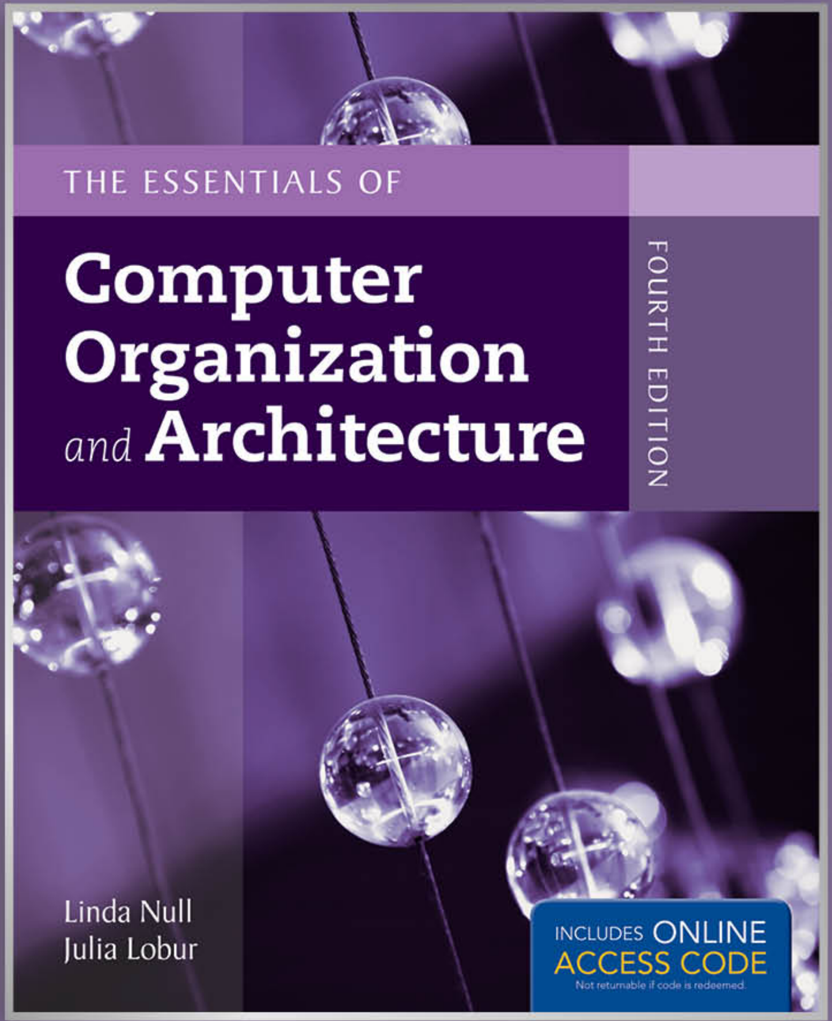
Chapter 3

Boolean Algebra and Digital Logic

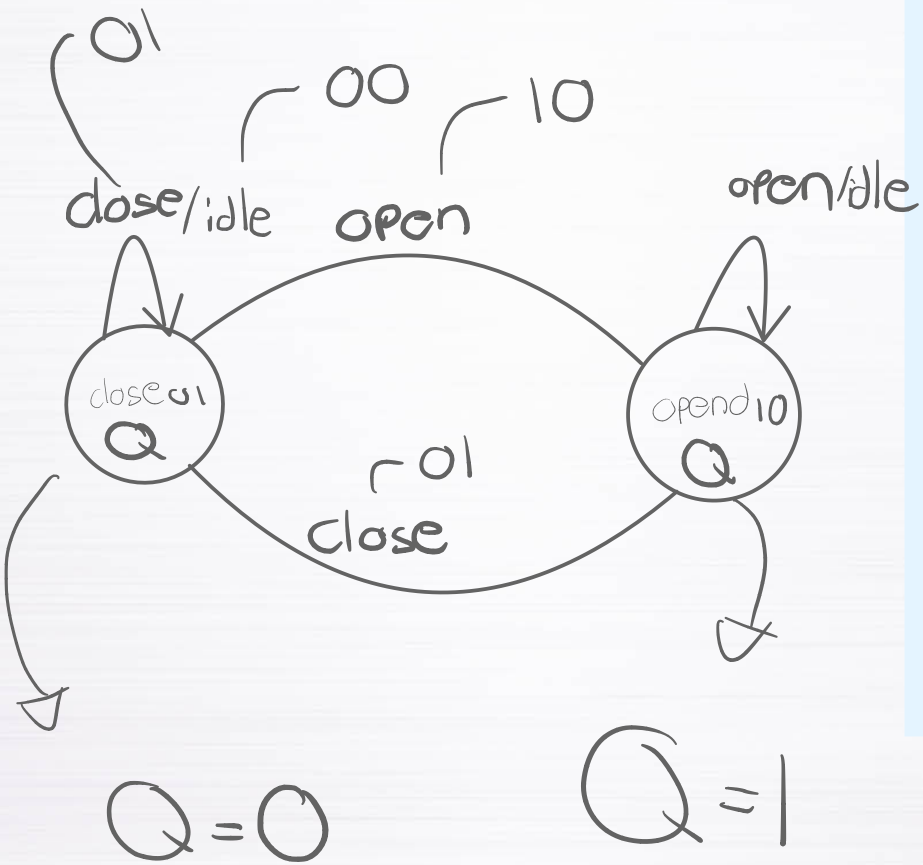


3.6 Sequential Circuits

- **Combinational circuits** are perfect for situations when we require the **immediate application of a Boolean function to a set of inputs**.
- There are other times when we need a circuit to **change its output** with consideration to its **current state** as well as its **inputs**.
  - These circuits have to "**remember**" their current state.
- ***Sequential circuits*** provide this functionality for us.

3.6 Sequential Circuits

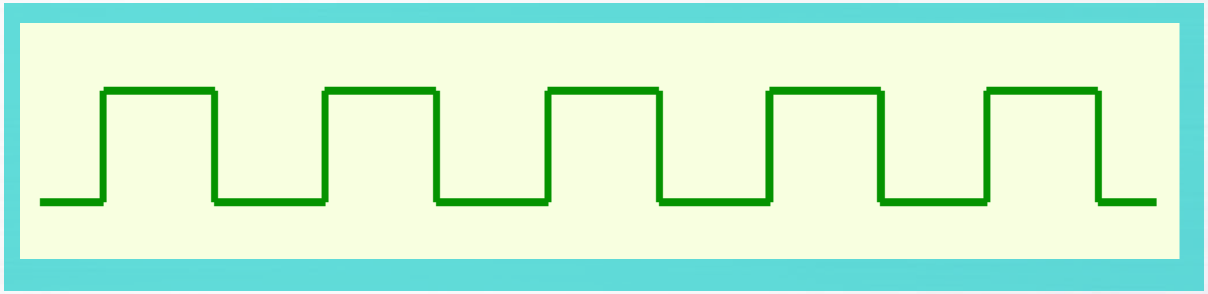
- Truth table of a **boom gate:**



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

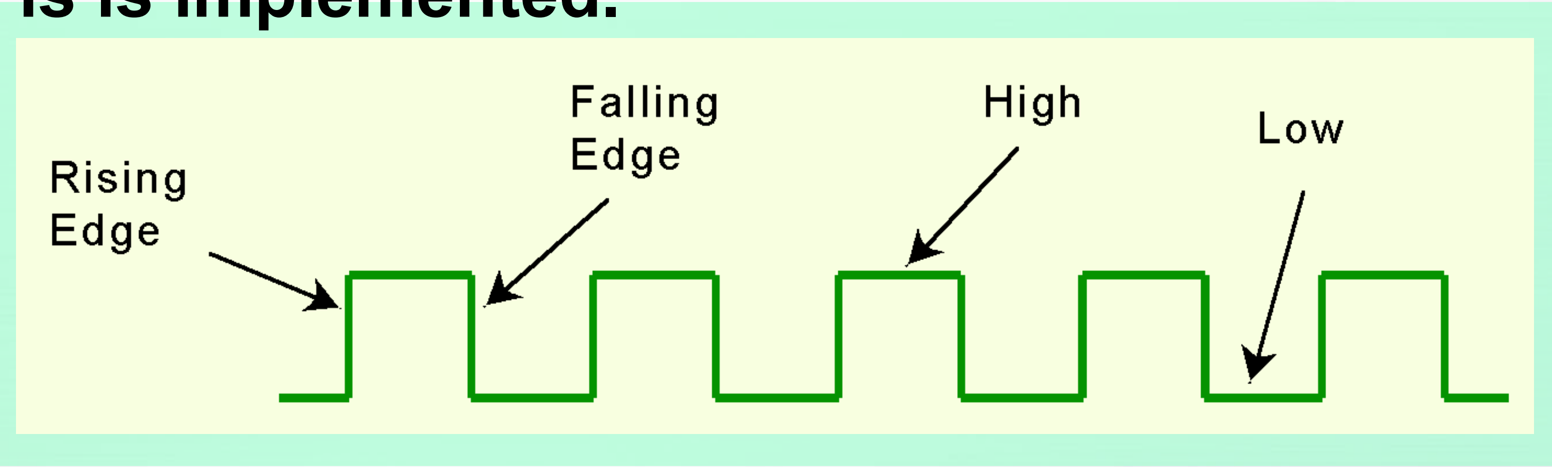
3.6 Sequential Circuits

- **State changes** of sequential circuits are **controlled by clocks.**
  - A "clock" is a special circuit that **sends electrical pulses** through a circuit.
- Clocks **produce electrical waveforms** such as the one shown below.



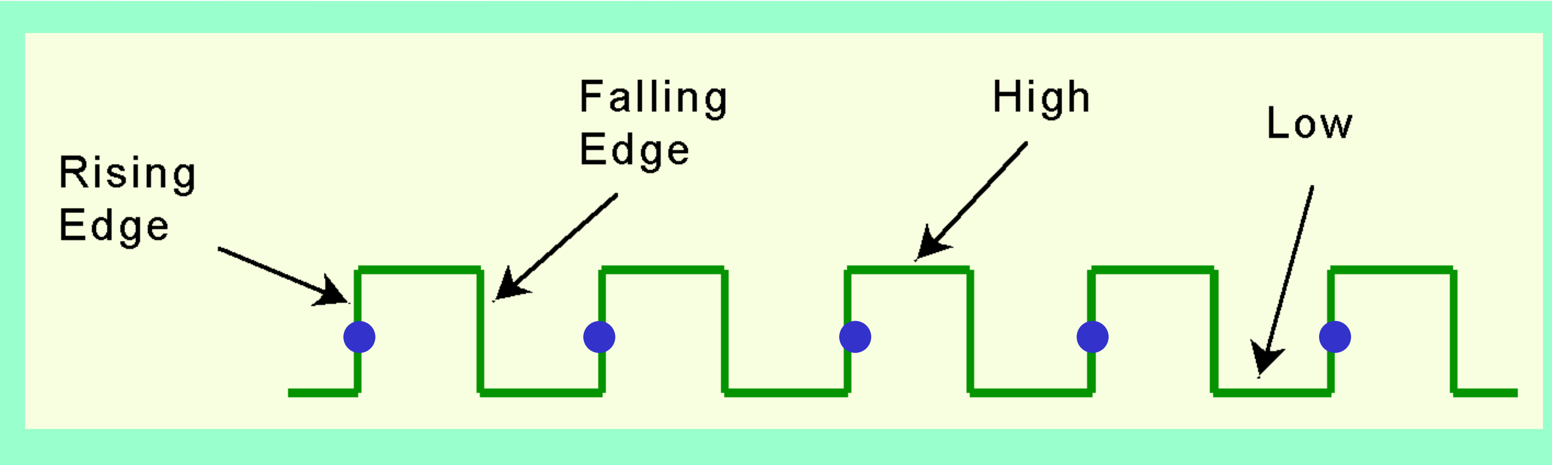
3.6 Sequential Circuits

- **State changes** occur in sequential circuits **only when the clock ticks**.
- Circuits can **change state** on the rising edge, falling edge, or when the clock pulse reaches its highest voltage - **depending on how the circuit is is implemented.**



3.6 Sequential Circuits

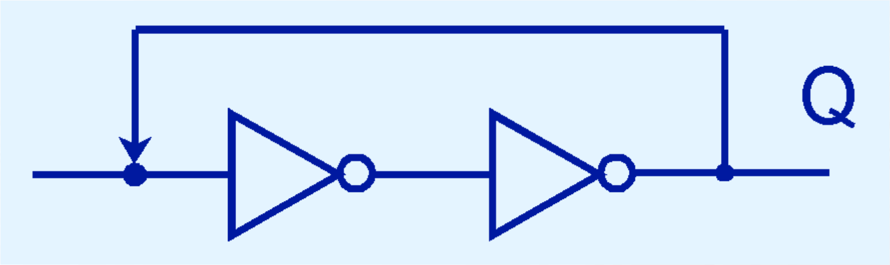
- Circuits that change state on the rising edge, or falling edge of the clock pulse are called ***edge-triggered***.
- ***Level-triggered circuits*** change state when the clock voltage reaches its highest or lowest level.



- In the examples that we consider, **we assume state changes on the rising edge**

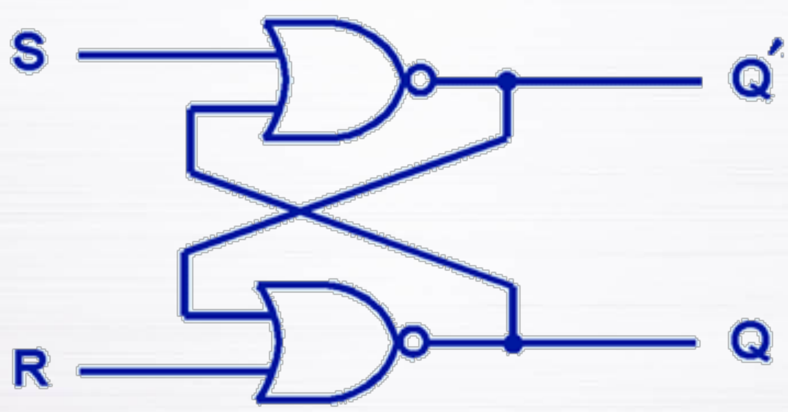
3.6 Sequential Circuits

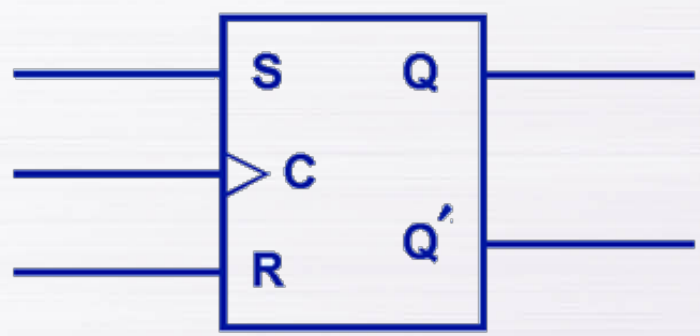
- To retain their state values, **sequential circuits rely on *feedback*.**
- Feedback in digital circuits occurs when an **output is looped back to the input**.
- A simple example of this concept is shown below.
  - If Q is 0 it will always be 0, if it is 1, it will always be 1. Why?



3.6 Sequential Circuits

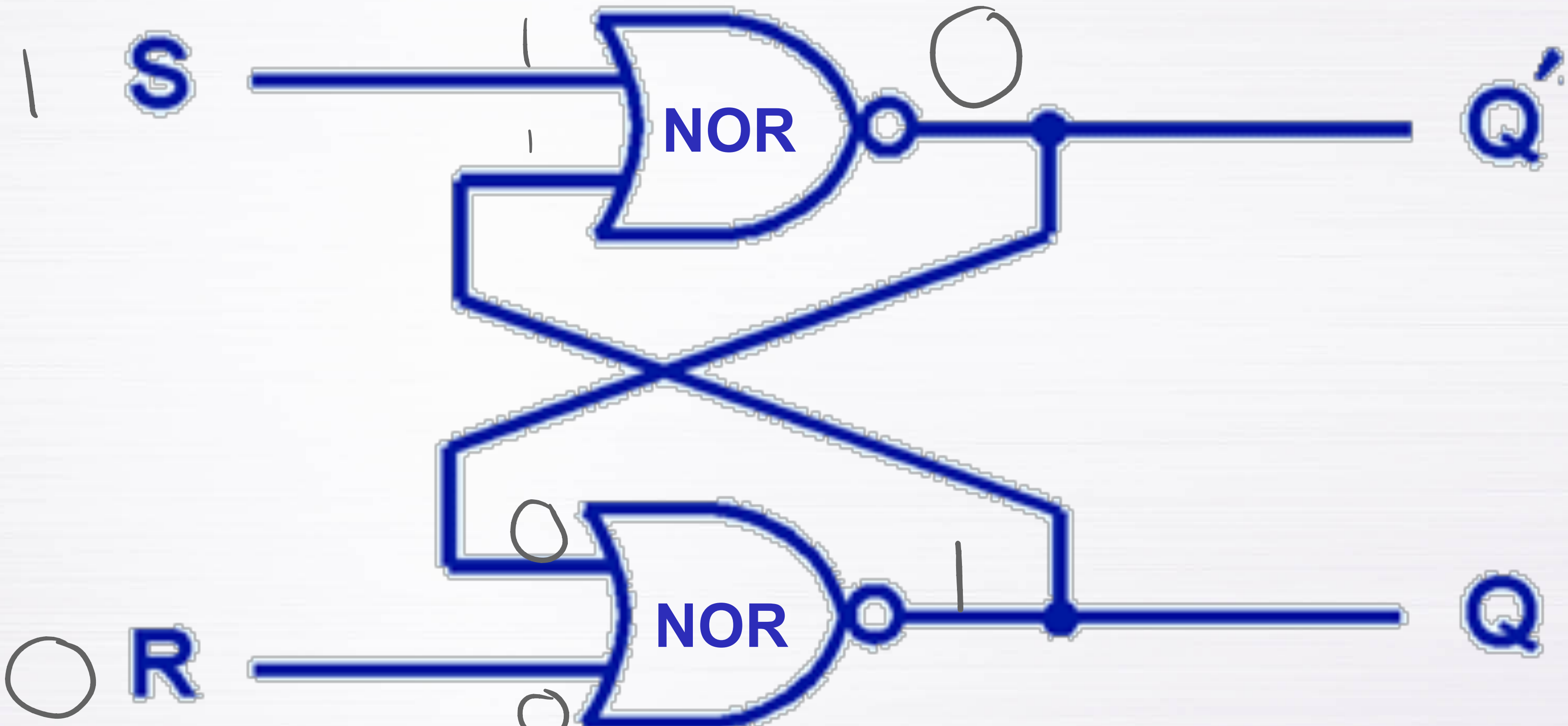
- Feedback illustrated based the most basic sequential logic component, the **SR flip-flop.**
  - **The “SR” stands for set/reset.**
- The internals of an SR flip-flop are shown below, along with its block diagram.
- **State** and **output** of a flip-flop are **synonymous**





3.6 Sequential Circuits

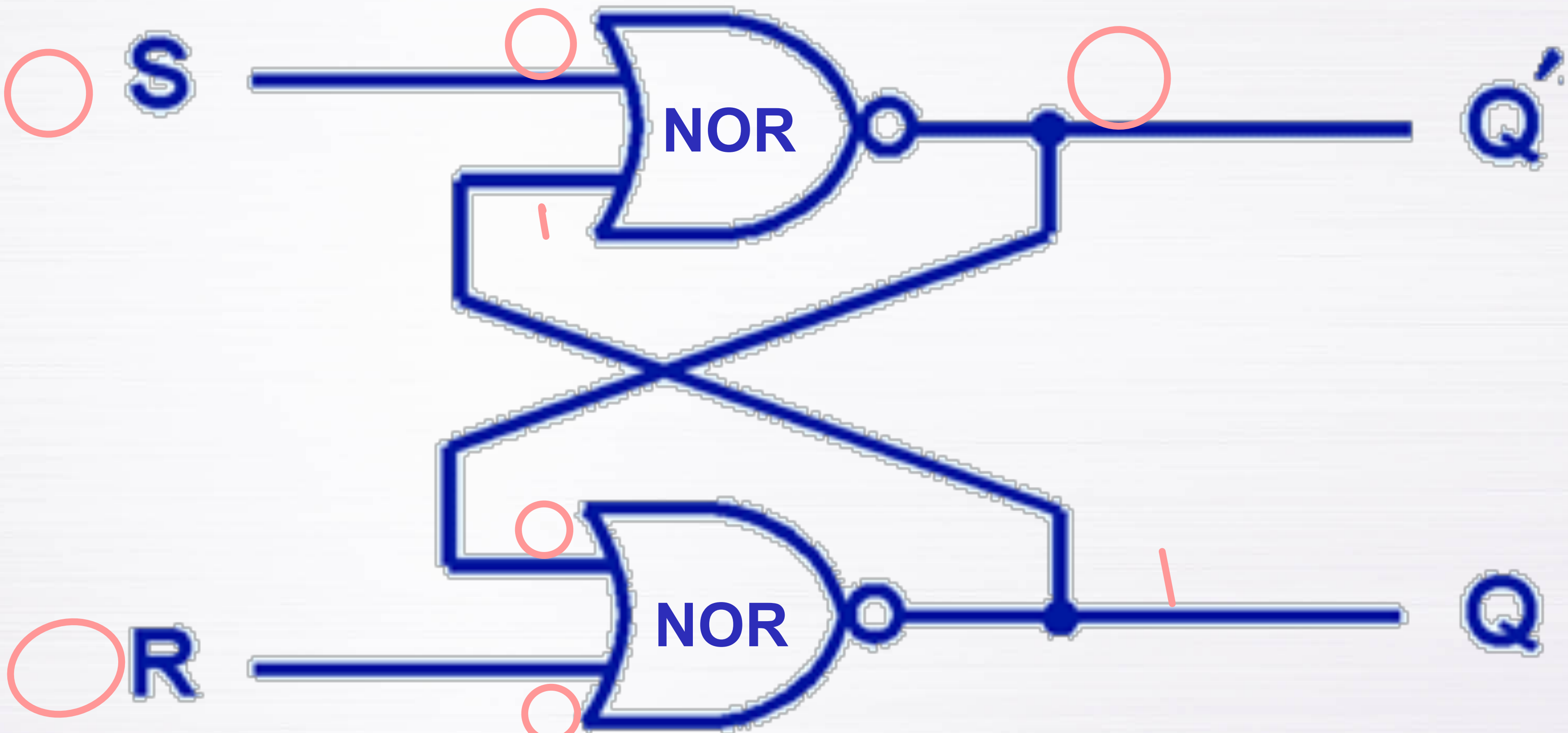
- Un-clocked **SR flip-flop:**



- Remember: NOR(x,y) = (x+y)' = x'y'

3.6 Sequential Circuits

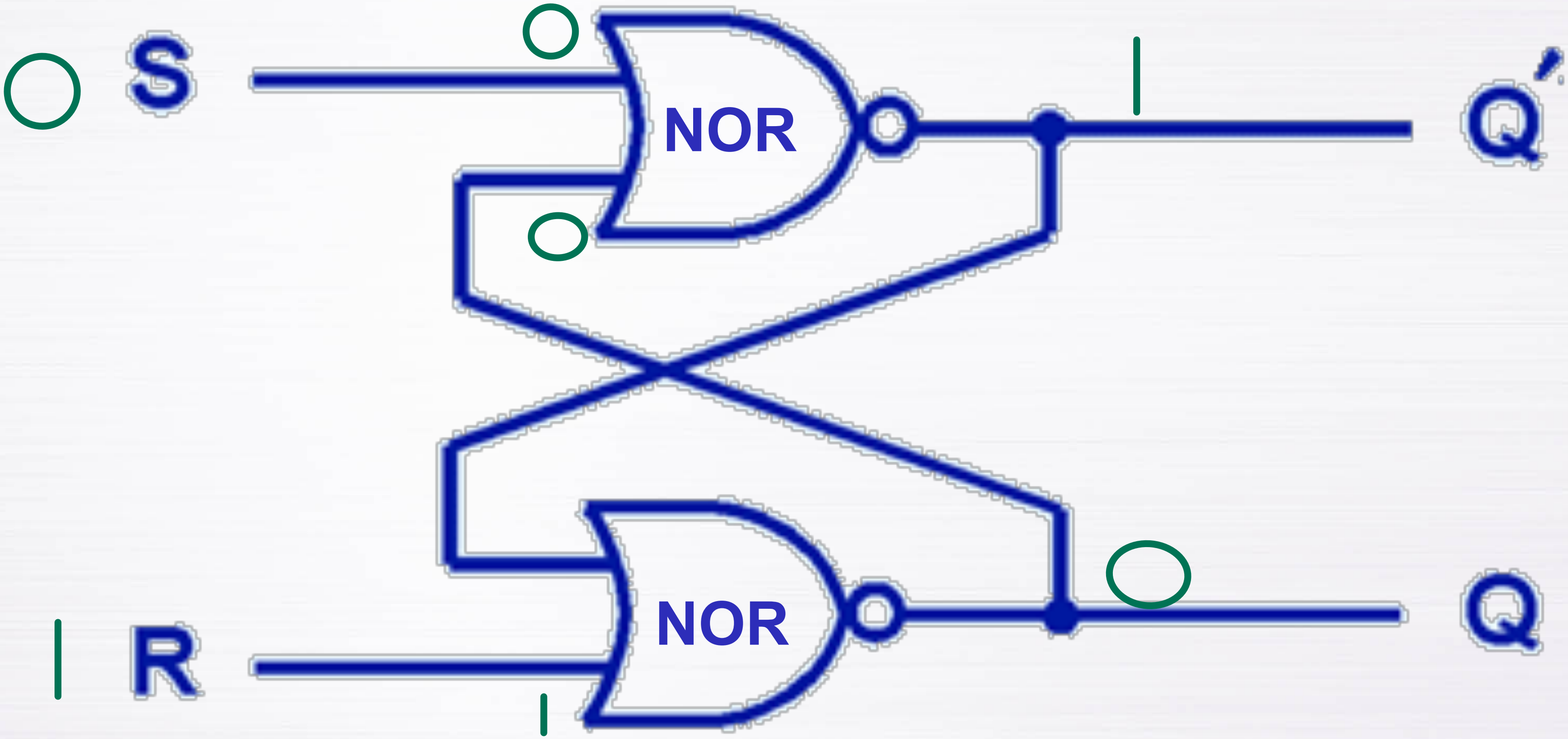
- Un-clocked **SR flip-flop:**



- Remember: NOR(x,y) = (x+y)’ = x’y’

3.6 Sequential Circuits

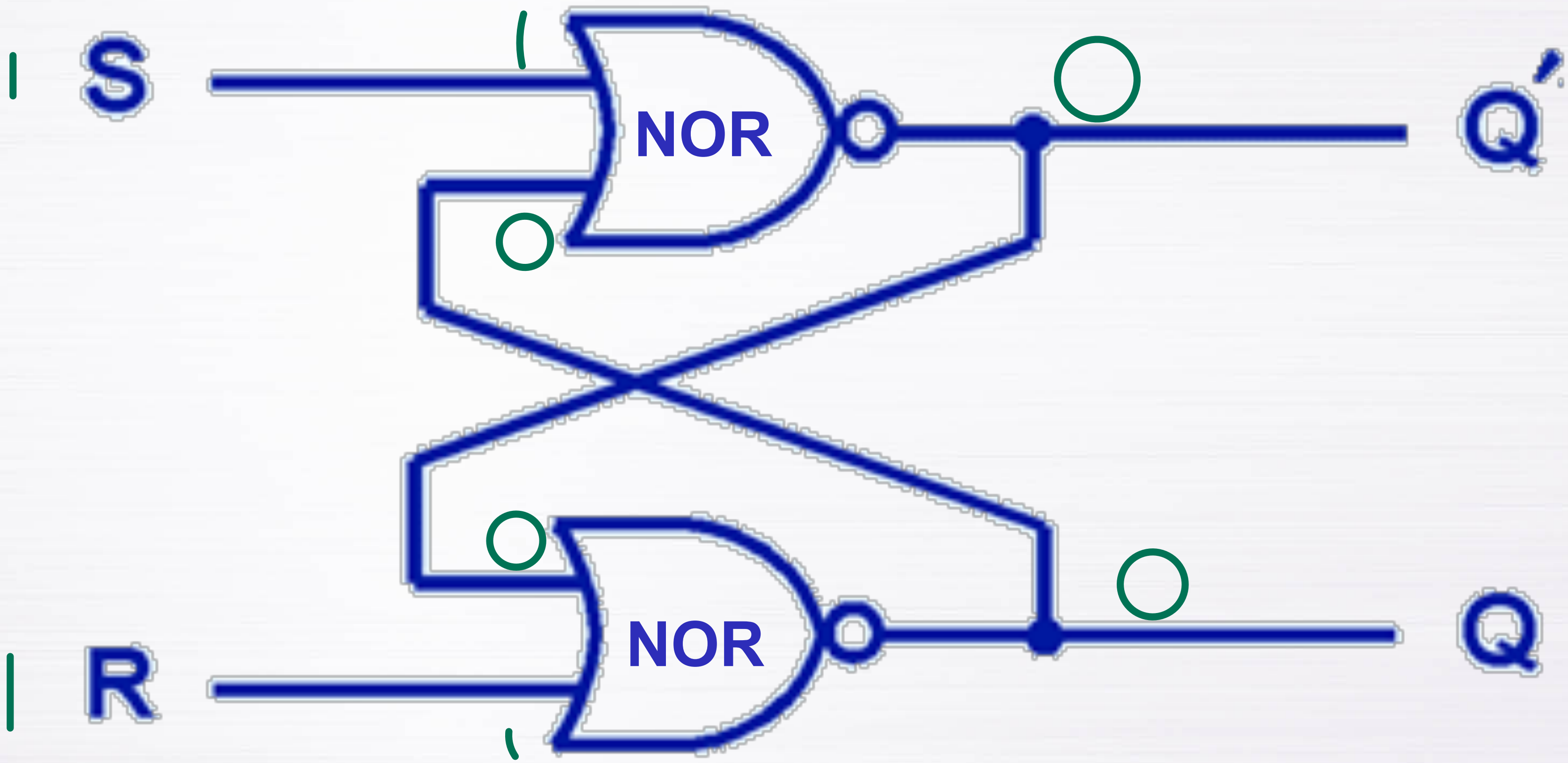
- Un-clocked **SR flip-flop:**



- Remember: NOR(x,y) = (x+y)’ = x’y’

3.6 Sequential Circuits

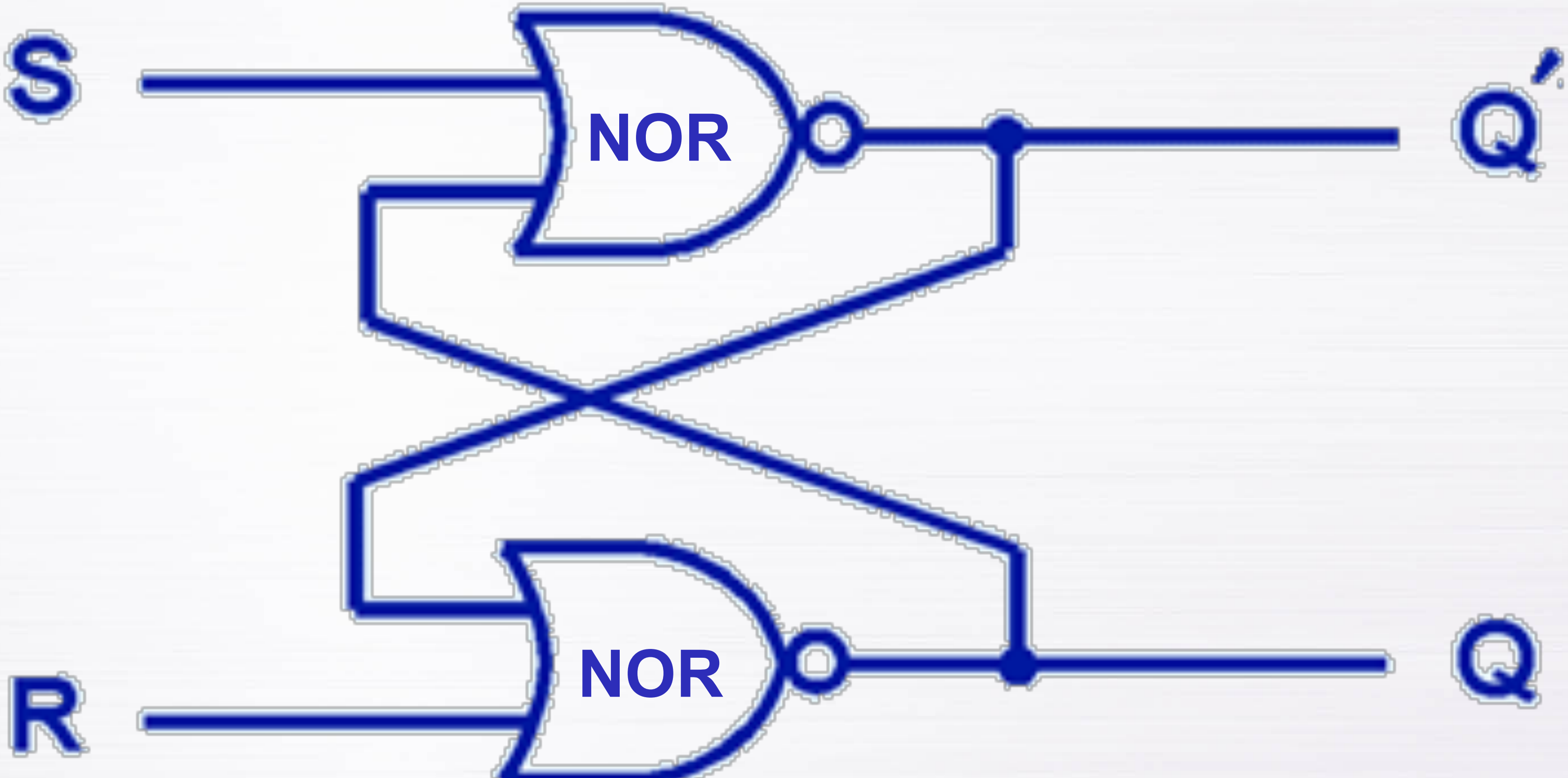
- Un-clocked **SR flip-flop:**



- Remember: NOR(x,y) = (x+y)' = x'y'

3.6 Sequential Circuits

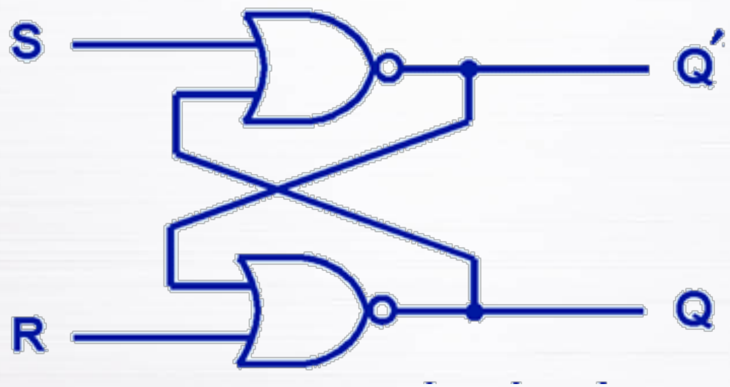
- Un-clocked **SR flip-flop:**



- Remember: NOR(x,y) = (x+y)' = x'y'

3.6 Sequential Circuits

- The **behavior** of an (un-clocked) **SR flip-flop** is described by a **characteristic table**.
- Q(t) means the value of the **output at time t**. Q(t+1) is the value of Q after the next clock tick.



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

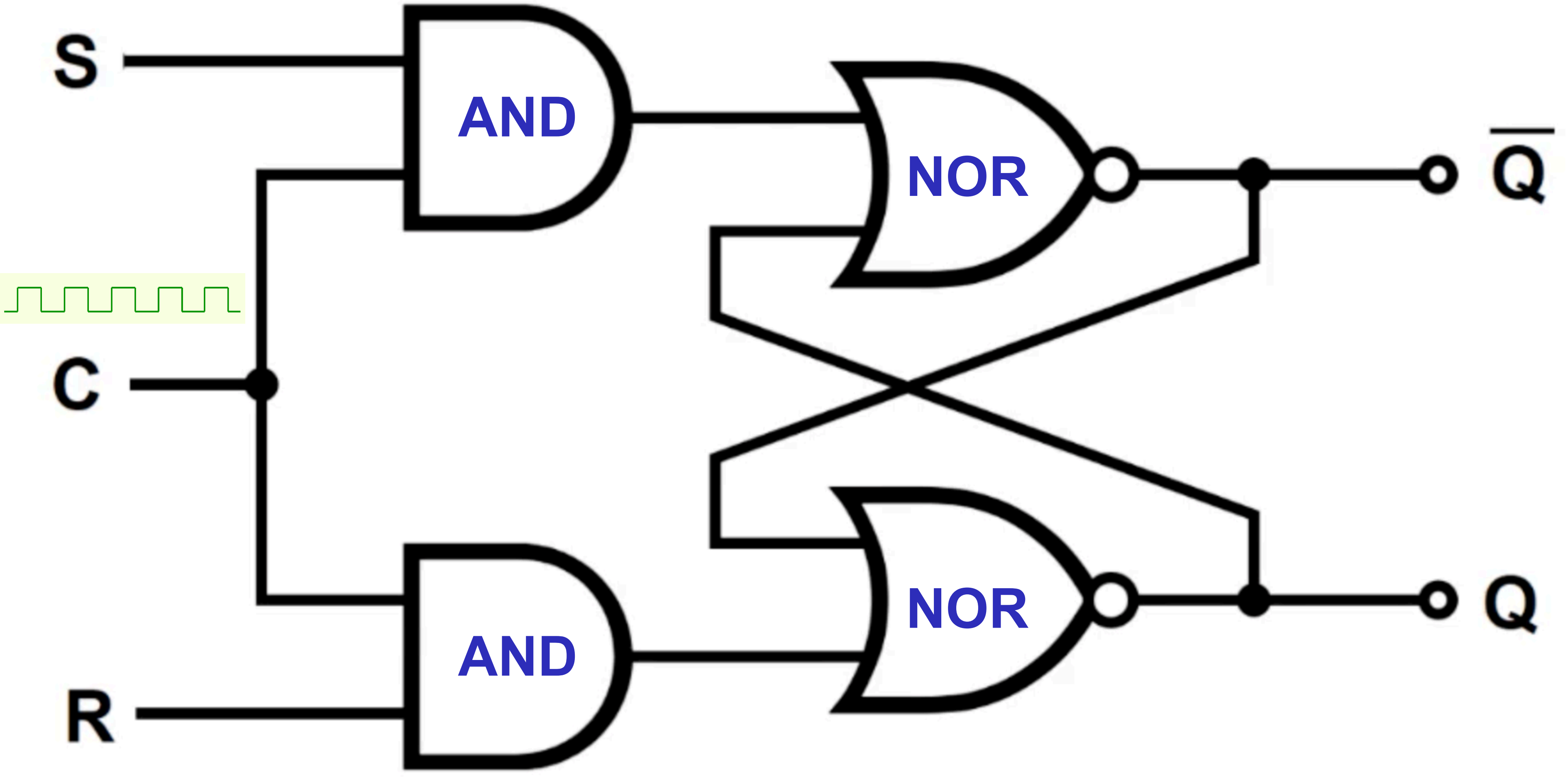
3.6 Sequential Circuits

- The SR flip-flop actually has **three inputs**: S, R, and its current output\state, Q.
- Thus, we can construct a **truth table** for this circuit, as shown at the right.
- **Notice the two undefined values**. When both S and R are 1, the SR flip-flop is unstable.

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

3.6 Sequential Circuits

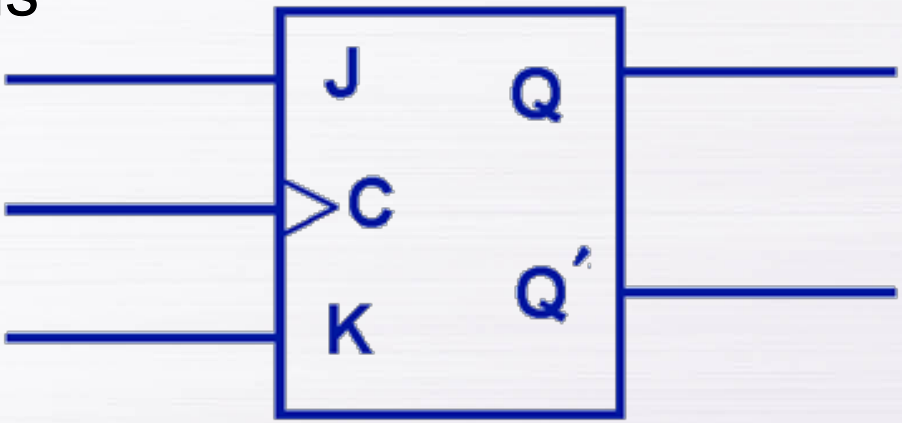
- Clocked **SR flip-flop:**



3.6 Sequential Circuits

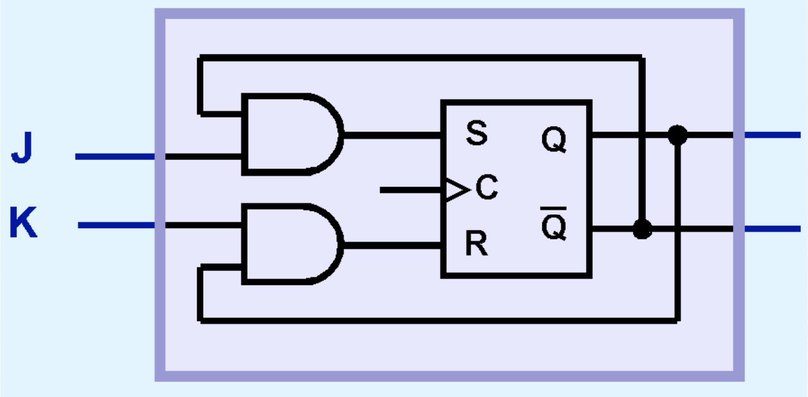
- An **SR flip-flop** with **input 1 1** results in an **unstable** circuit.
- The SR **flip-flop can be modified** to provide a **stable state when both inputs are 1**.

- This modified flip-flop is called a **JK flip-flop**, shown at the right.



3.6 Sequential Circuits

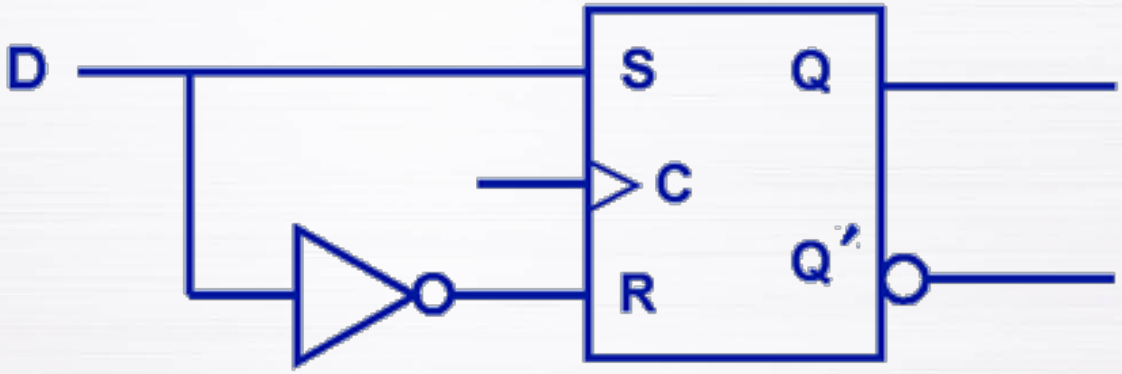
- At the right, we see how an **SR flip-flop** can be **modified to** create a **JK flip-flop**.
- The **characteristic table** indicates that the flip-flop is stable for all inputs.



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)

3.6 Sequential Circuits

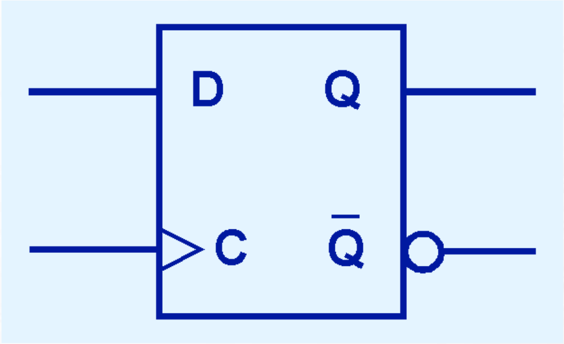
- **Another modification** of the SR flip-flop is the **D flip-flop**, shown below with its characteristic table.
- The **input D at time t** becomes the **output Q at time t+1**
- D flip-flops are also known as **memory cells**



D	Q(t+1)
0	0
1	1

3.6 Sequential Circuits

- The D flip-flop is the fundamental circuit of computer memory.
  - **D flip-flops are usually illustrated using the block diagram shown below.**



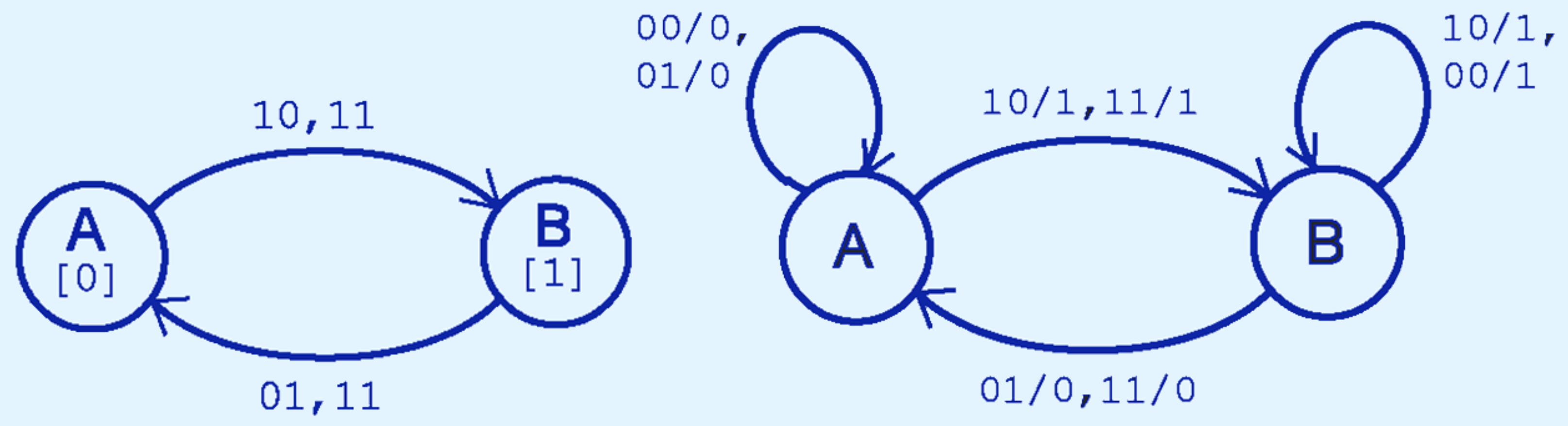
3.6 Sequential Circuits

- The behavior of **sequential circuits** can be expressed using **characteristic tables** or **finite state machines** (FSMs).

- **Moore and Mealy machines** are two types of FSMs that are equivalent.
  - They differ only in how they express the outputs of the machine.
- **Moore machines** place **outputs on each node/state**, while **Mealy machines** present their **outputs on the transitions**.

3.6 Sequential Circuits

- The behavior of a JK flop-flop is depicted by a Moore machine (left) and a Mealy machine (right).



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	$\overline{Q}(t)$

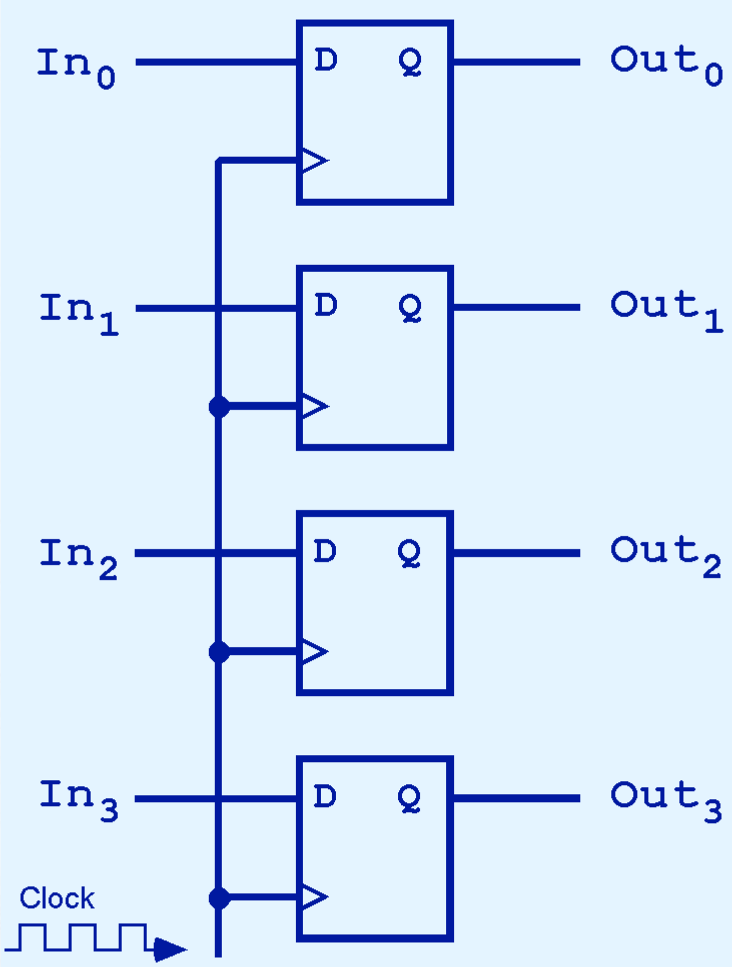
3.6 Sequential Circuits

- **Sequential circuits** are used anytime that we have a **“stateful” application**.
  - A stateful application is one where the **next state of the machine depends on the current state** of the machine and the input.
- **Stateful applications may require both combinational and sequential logic**.
- The following slides provide several **examples** of circuits that fall into this category.

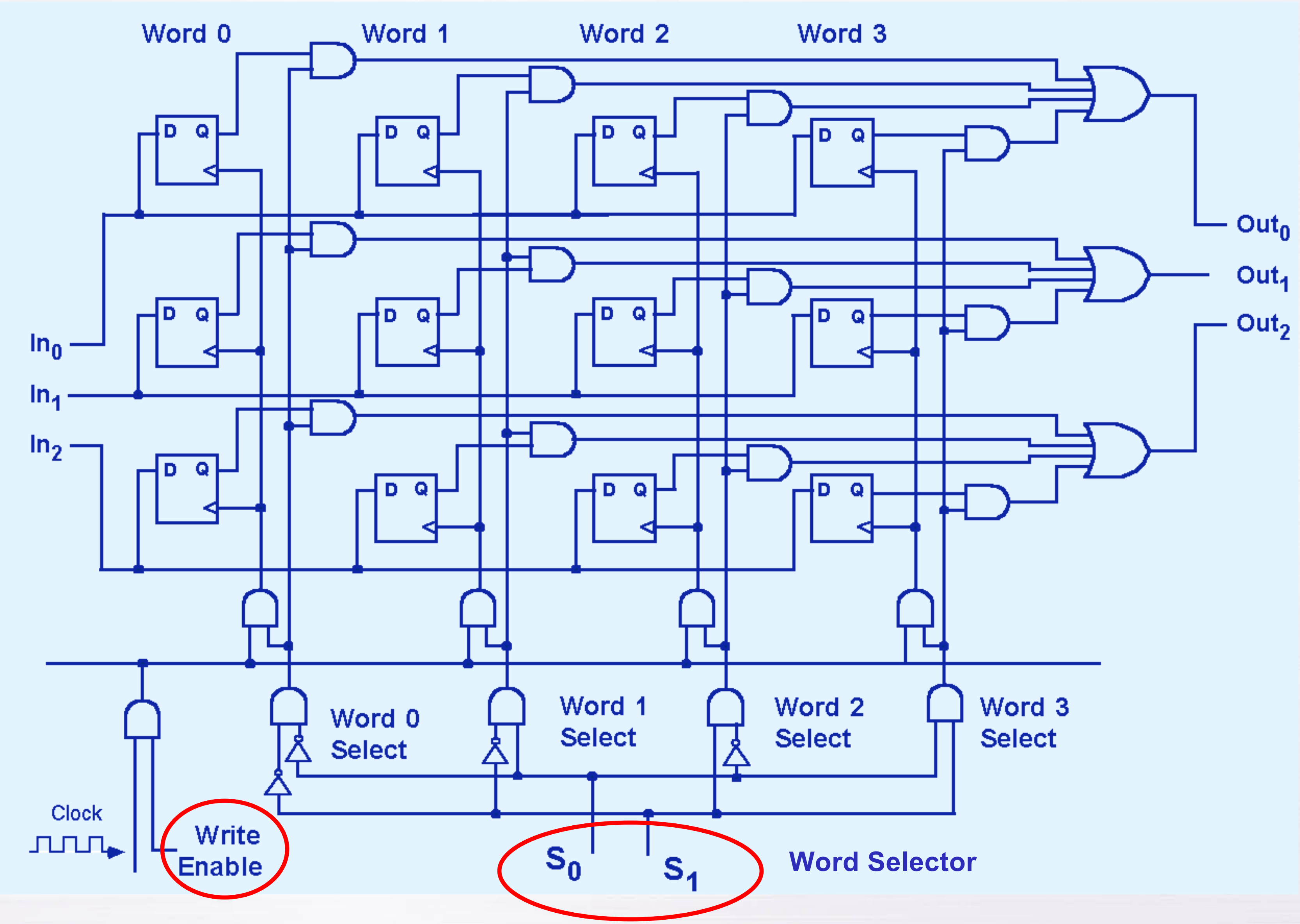
3.6 Sequential Circuits

- This illustration shows a **4-bit register consisting of D flip-flops**. You will usually see its block diagram (below) instead.

A larger memory configuration is shown on the next slide.

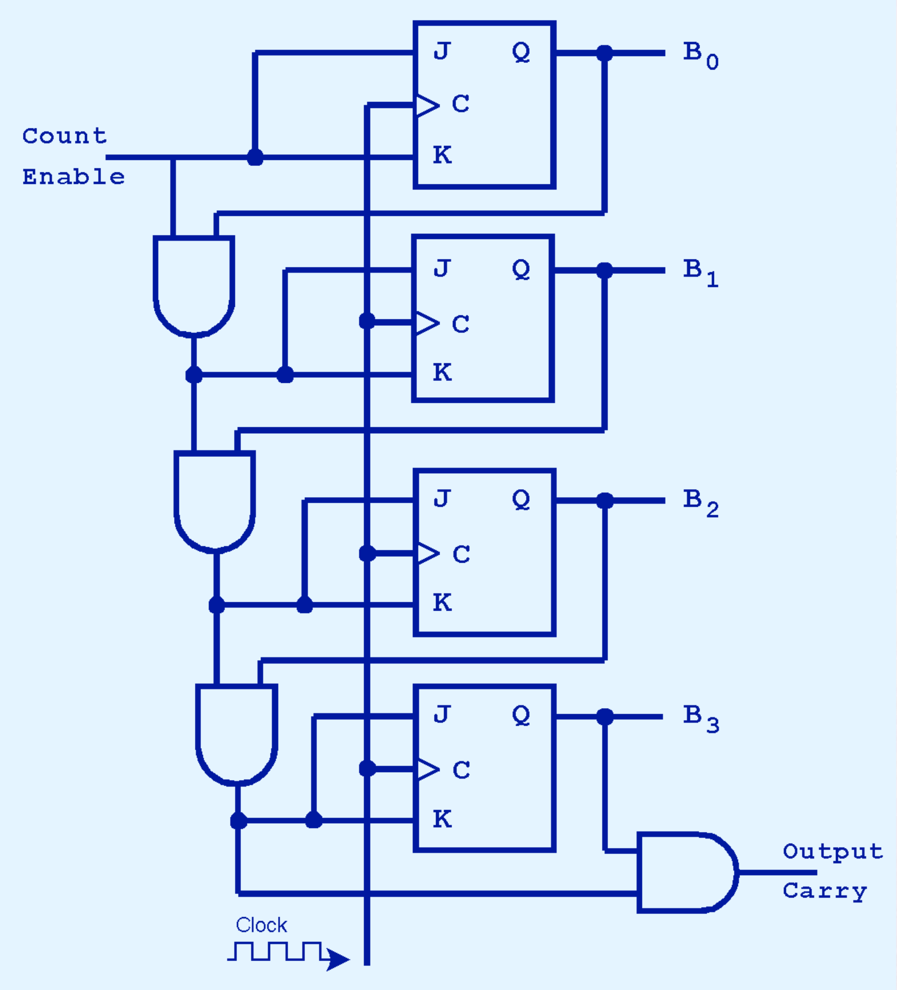


3.6 Sequential Circuits



3.6 Sequential Circuits

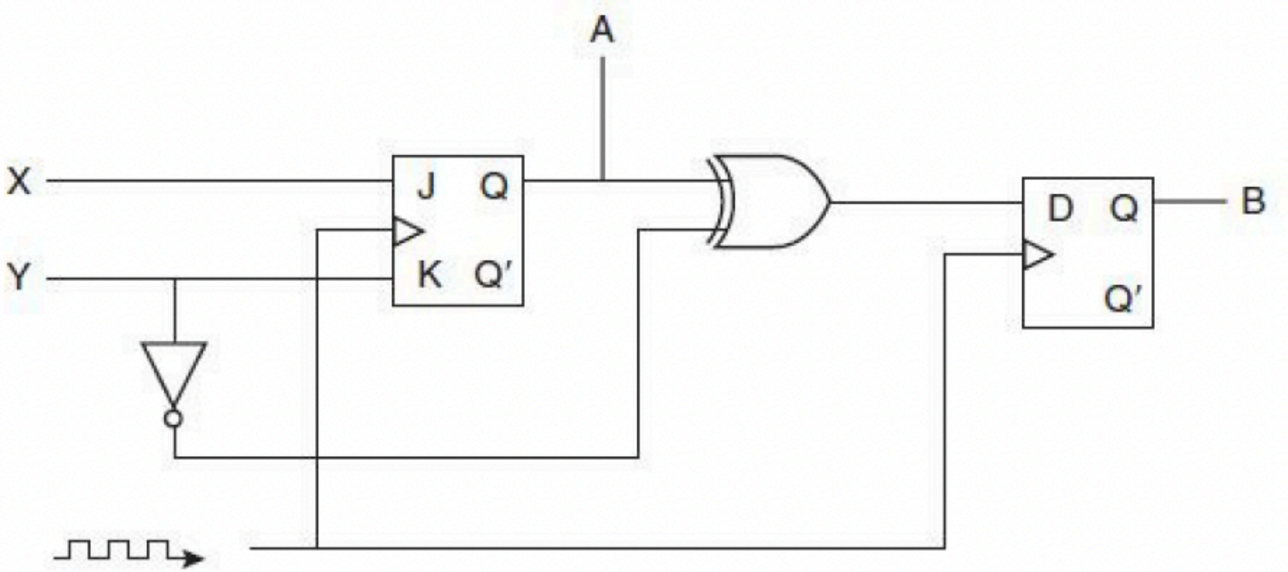
- A **binary counter** is another example of a sequential circuit.
- The **low-order bit is complemented at each clock pulse**.
- **Whenever it changes from 0 to 1, the next bit is complemented**




```
0000        ...
0001        1101
0010        1110
0011        1111
...         0000
```

3.6 Sequential Circuits

Complete the truth table for the following sequential circuit:



			Next State	
X	*Y*	*A*	*A*	*B*
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

The flipflops are edge triggered - hence a signal cannot traverse through a flipflop, exit it and affect the next component during a single upward edge of the clock pulse.

Chapter 3 Summary

- Computers are implementations of Boolean logic.
- Boolean functions are completely described by truth tables.
- Logic gates are small circuits that implement Boolean operators.
- The basic gates are AND, OR, and NOT.
  - The XOR gate is very useful in parity checkers and adders.
- The "universal gates" are NOR, and NAND.

Chapter 3 Summary

- Computer circuits consist of combinational logic circuits and sequential logic circuits.
- Combinational circuits produce outputs (almost) immediately when their inputs change.
- Sequential circuits require clocks to control their changes of state.
- The basic sequential circuit unit is the flip-flop: The behaviors of the SR, JK, and D flip-flops are the most important to know.