

# **CS 303**

# **Logic & Digital System Design**

---

**Ömer Ceylan**

**. Sabancı .  
Universitesi**

# Chapter 5

## Synchronous Sequential Logic Part I

---





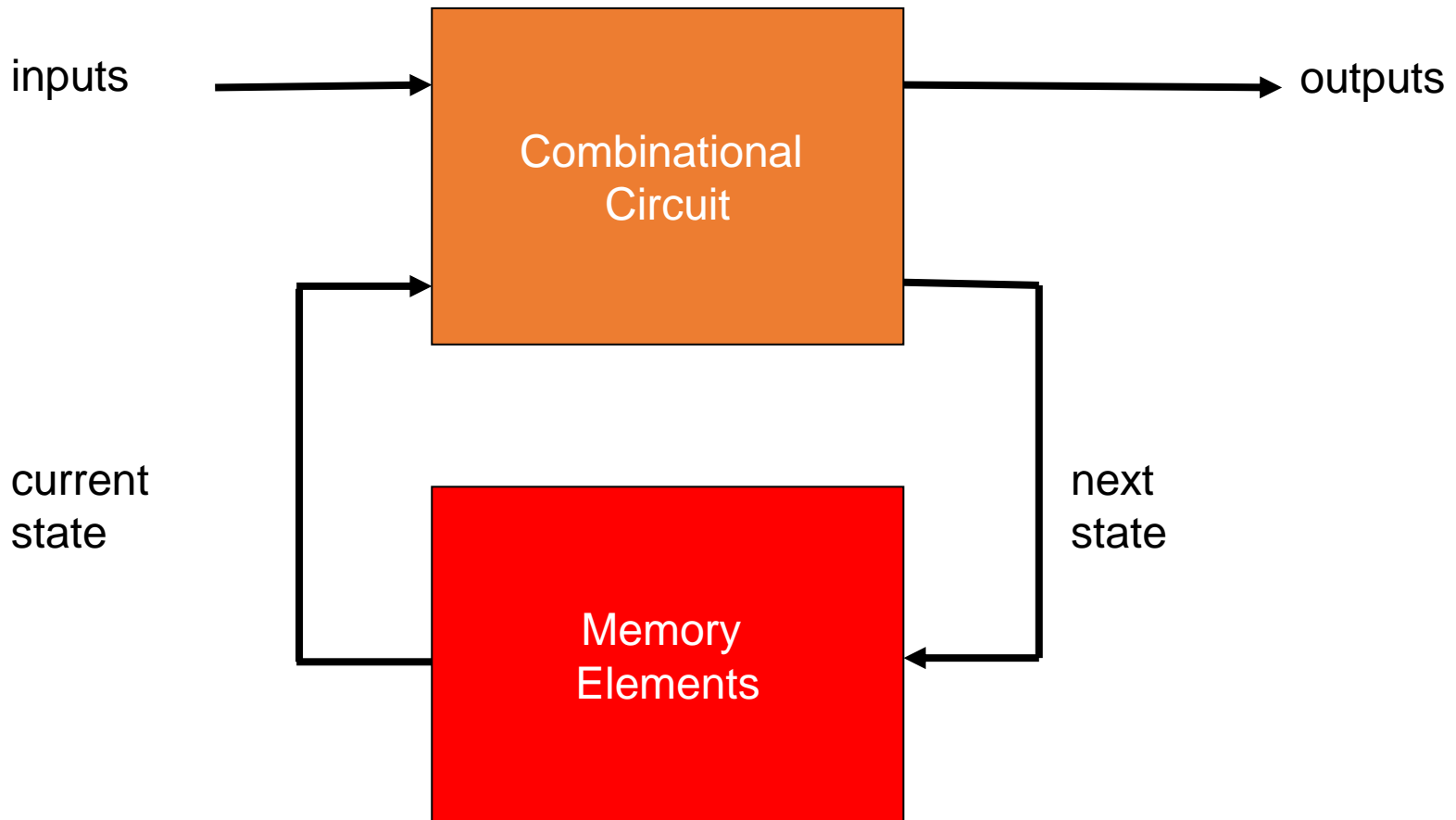
# Sequential Logic

---

- Digital circuits we have learned, so far, have been combinational
  - no memory,
  - outputs are entirely defined by the “current” inputs
- However, many digital systems encountered everyday life are sequential (i.e., they have memory)
  - the memory elements remember past inputs
  - outputs of sequential circuits are not only dependent on the current input but also the state of the memory elements.



# Sequential Circuits Model



Current state is a function of past inputs and **initial state**

# Classification 1/2

---

- Two types of sequential circuits

## 1. Synchronous

- Signals affect the memory elements at discrete instants of time.
- Discrete instants of time requires synchronization.
- Synchronization is usually achieved through the use of a common clock.
- A “clock generator” is a device that generates a periodic train of pulses.



## 1. Synchronous

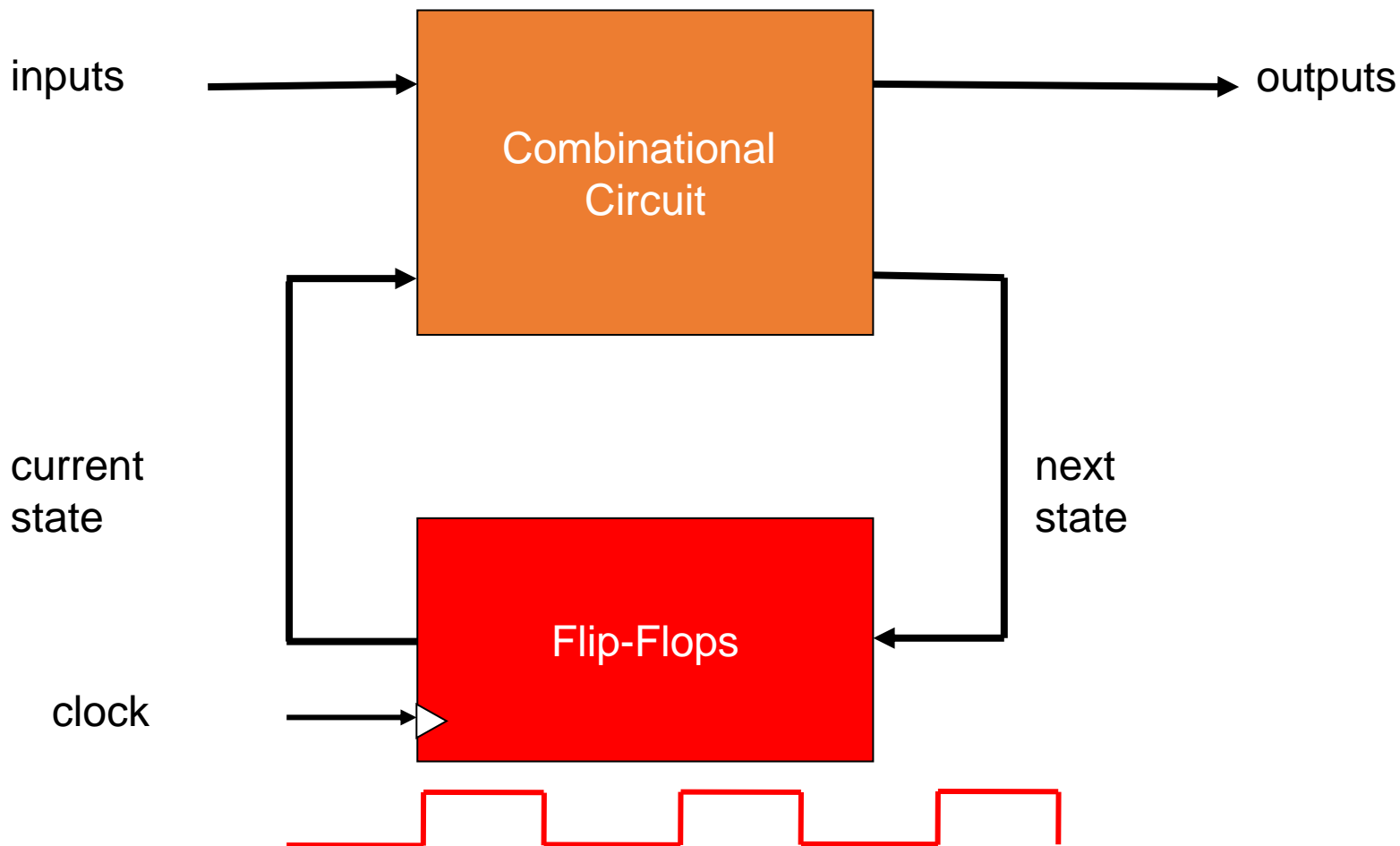
- The state of the memory elements are updated with the arrival of each pulse
- This type of logical circuit is also known as clocked sequential circuits.

## 2. Asynchronous

- No clock
- Behavior of an asynchronous sequential circuits depends upon the input signals at any instant of time and the order in which the inputs change.
- Memory elements in asynchronous circuits are regarded as time-delay elements

# Clocked Sequential Circuits

- Memory elements are **flip-flops** which are logic devices, each of which is capable of storing one bit of information.



# Clocked Sequential Circuits

---

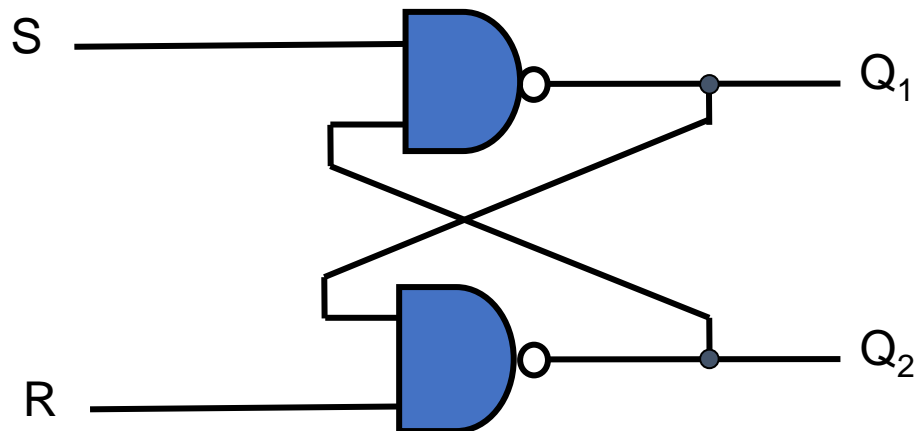
- The outputs of a clocked sequential circuit can come from the combinational circuit, from the outputs of the flip-flops or both.
- The state of the flip-flops can change only during a clock pulse transition
  - i.e., either **low-to-high** or **high-to-low**
  - **clock edge**
- When the clock maintains its value, the flip-flop output does not change
- The transition from one state to the next occurs at the clock edge.



- The most basic types of memory elements are not flip-flops, but latches.
- A latch is a memory device that can maintain a binary state indefinitely.
- Latches are, in fact, asynchronous devices and they usually do not require a clock to operate.
- Therefore, they are not directly used in clocked synchronous sequential circuits.
- They rather be used to construct flip-flops.

# SR-Latches with NAND Gates

- made of cross-coupled NAND (NOR) gates



Also known as S'R'-latch

S	R	Q <sub>1</sub>	Q <sub>2</sub>
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0
0	0	1	1

After S = 1, R = 0

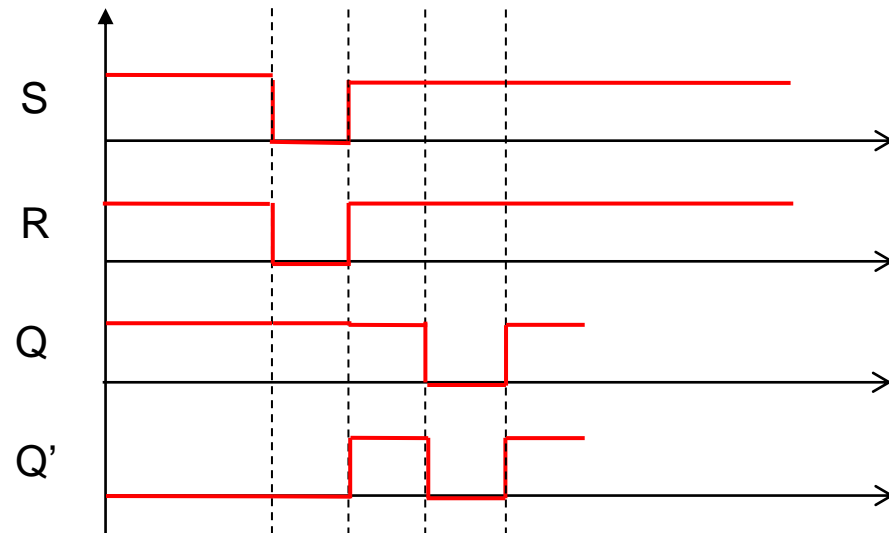
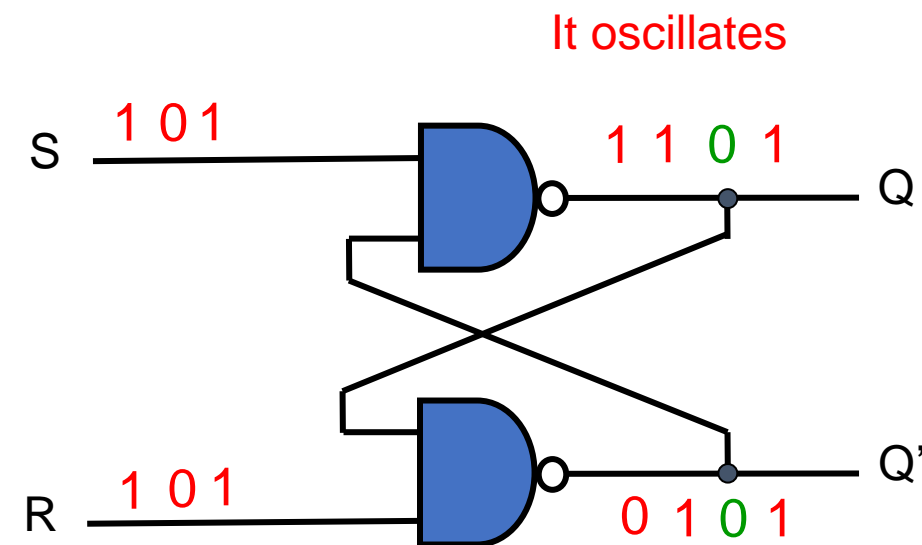
$$Q_2 = Q_1'$$

After S = 0, R = 1

Undefined

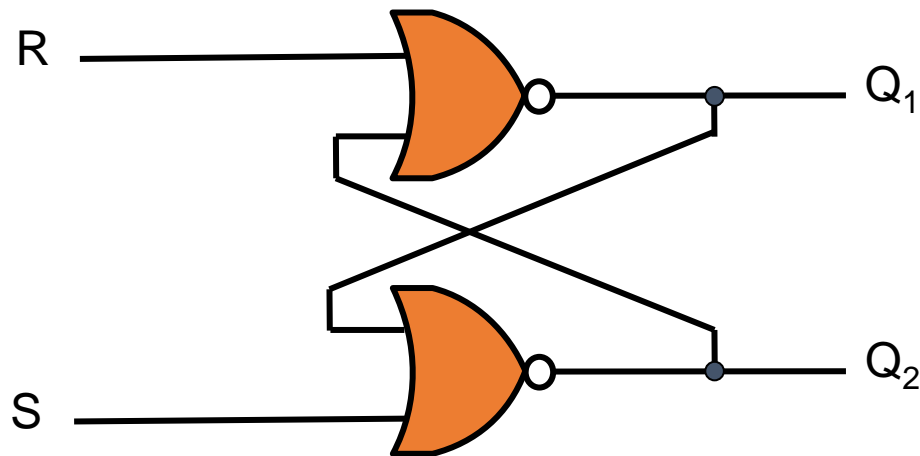
# Undefined State of SR-Latch

- $S = R = 0$  may result in an undefined state
  - the next state is unpredictable when both S and R goes to 1 at the same time after the undefined state.
  - It may oscillate
  - Or the outcome state depend on which of S and R goes to 1 first.



# SR-Latch

- made of cross-coupled NOR (or NAND) gates



S	R	Q <sub>1</sub>	Q <sub>2</sub>
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

After S = 1, R = 0

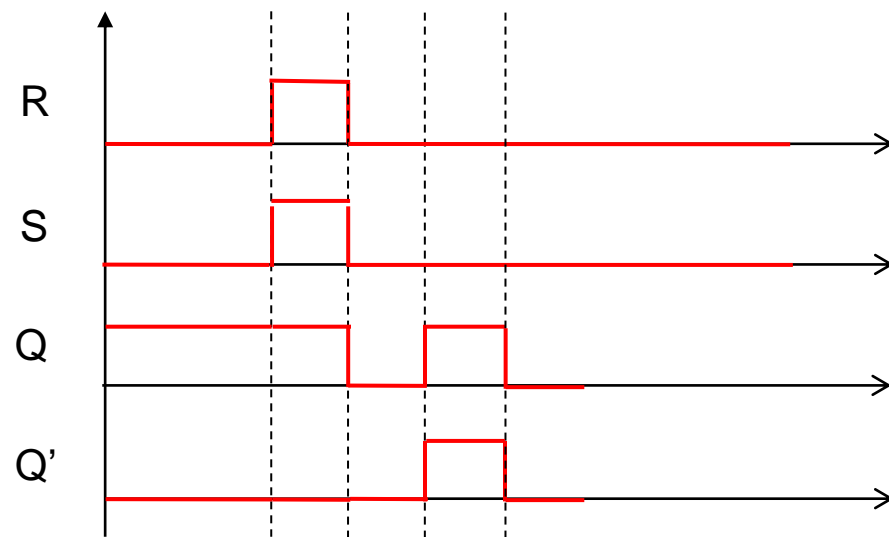
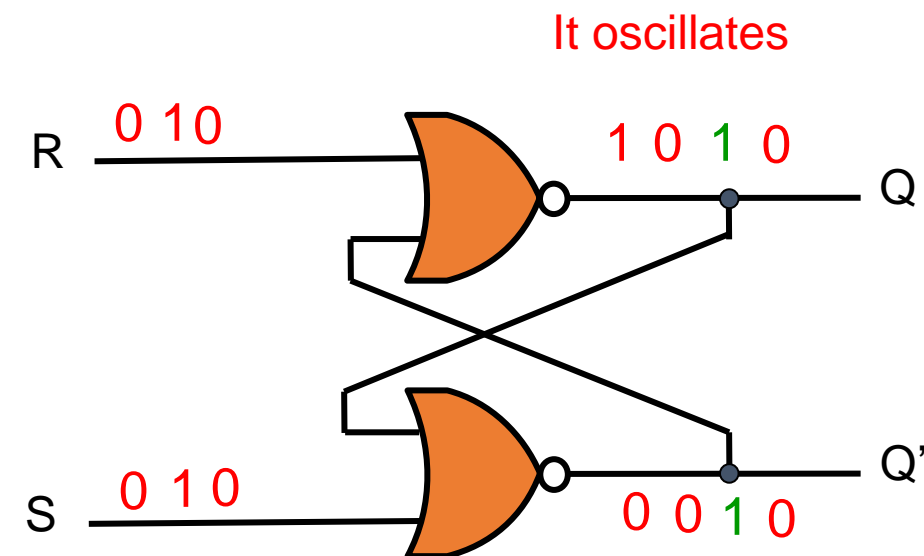
After S = 0, R = 1

Undefined

$$Q_2 = Q_1'$$

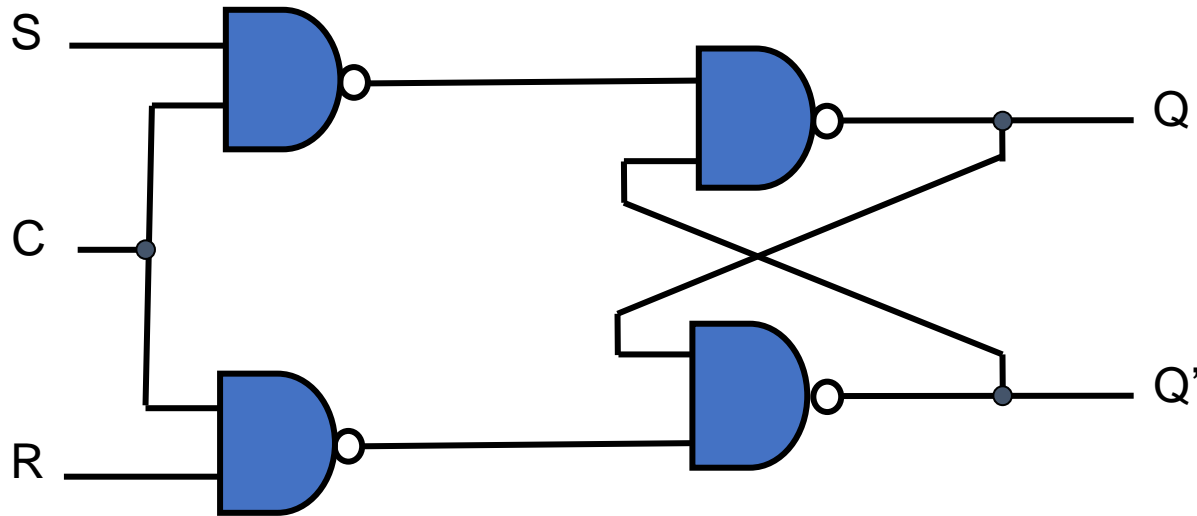
# Undefined State of SR-Latch

- $S = R = 1$  may result in an undefined state
  - the next state is unpredictable when both S and R goes to 0 at the same time after the undefined state.
  - It may oscillate
  - Or the outcome state depend on which of S and R goes to 0 first.



# SR-Latch with Control Input

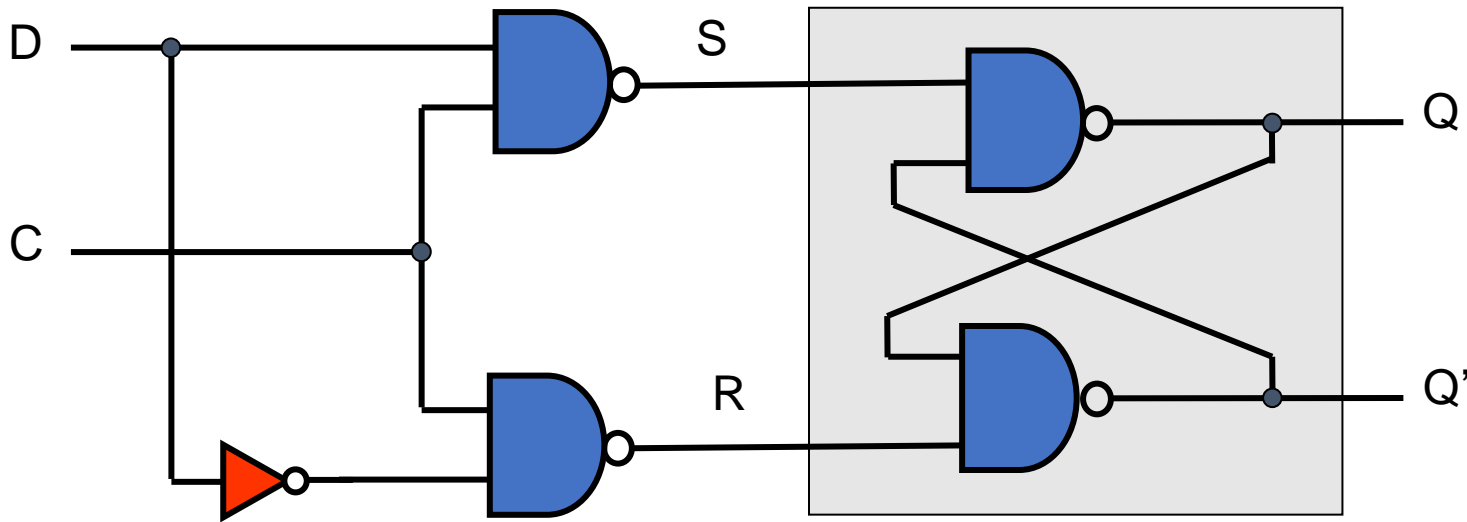
- Control inputs allow the changes at S and R to change the state of the latch.



C	S	R	Q	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	

# D-Latch

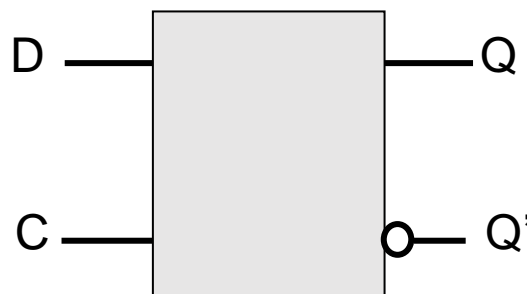
- SR latches are seldom used in practice because the indeterminate state may cause instability
- Remedy: D-latches



This circuit guarantees that the inputs to the S'R'-latch is always complement of each other when  $C = 1$ .

C	D	Next state of Q
0	X	No change
1	0	Q = 0; reset state
1	1	Q = 1; set state

- We say that the D input is sampled when  $C = 1$



D-latch





# D-Latch as a Storage Unit

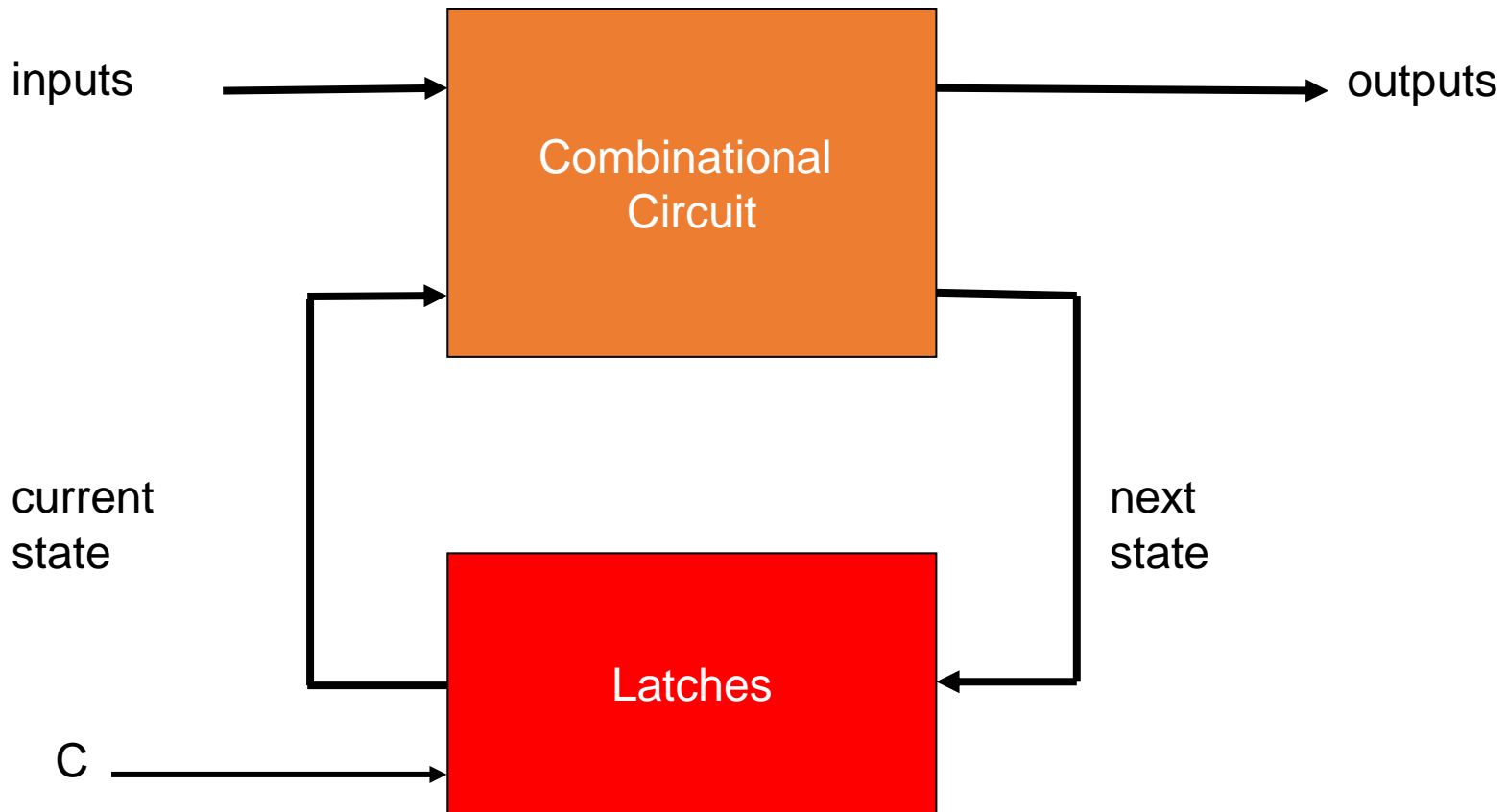
---

- D-latches can be used as temporary storage
  - The input of D-latch is transferred to the Q output when  $C = 1$
  - When  $C = 0$  the binary information is retained.
- We call latches level-sensitive devices.
  - So long as C remains at logic-1 level, any change in data input will change the state and the output of the latch.
  - Level sensitive latches cannot be used as the state elements in clocked sequential circuits
- Memory devices that are sensitive to the rising or falling edge of control input is called flip-flops.



# Need for Flip-Flops 1/2

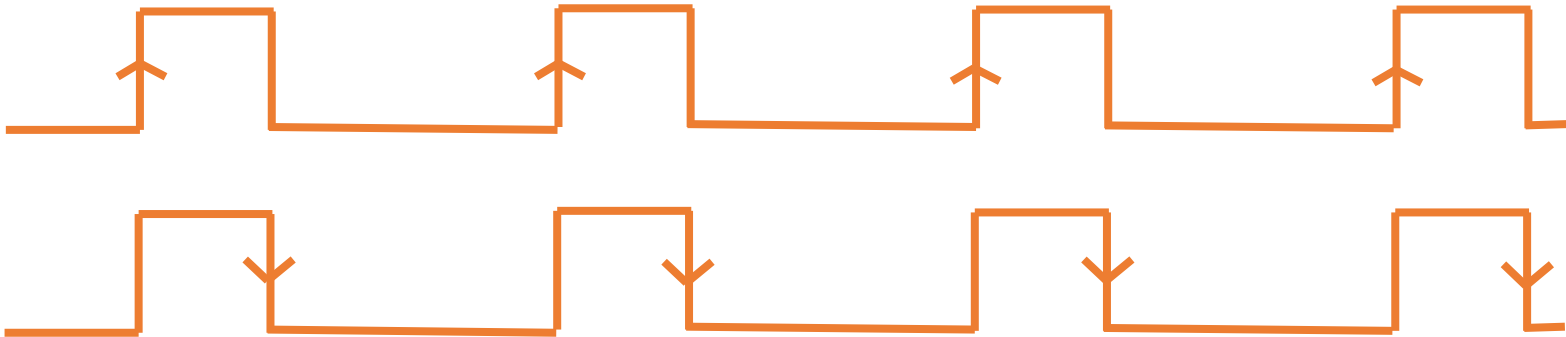
- Outputs may keep changing so long as  $C = 1$





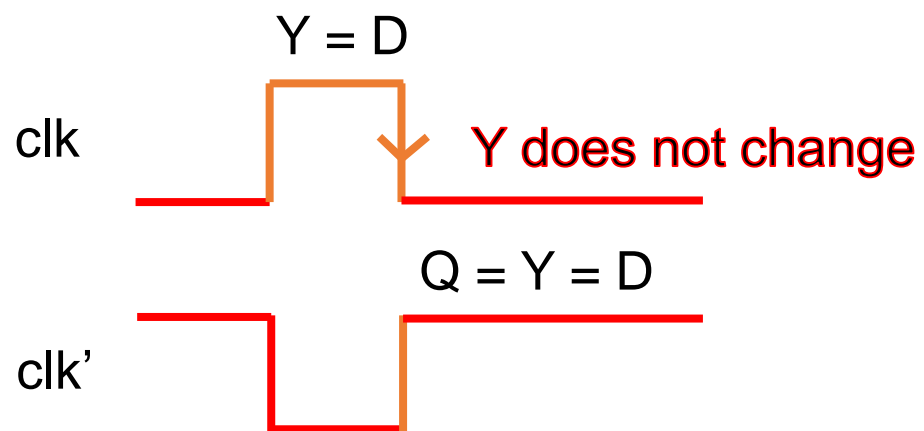
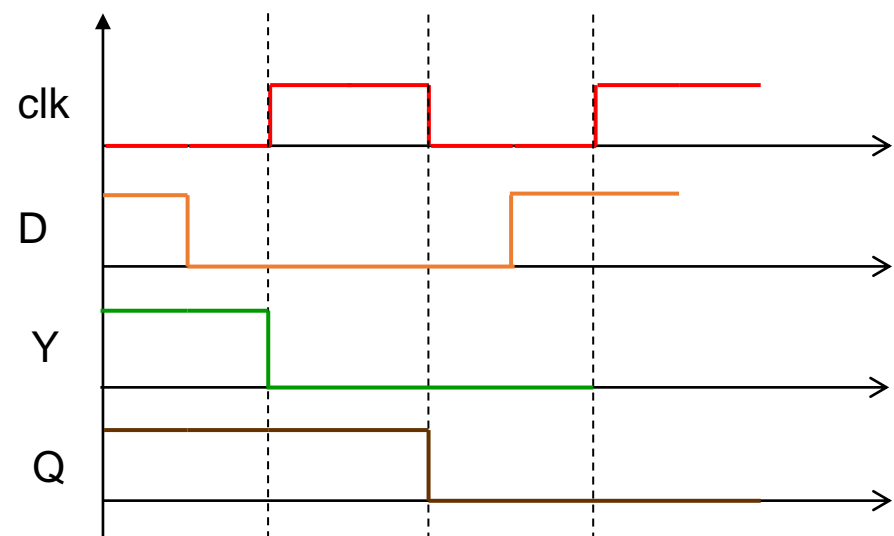
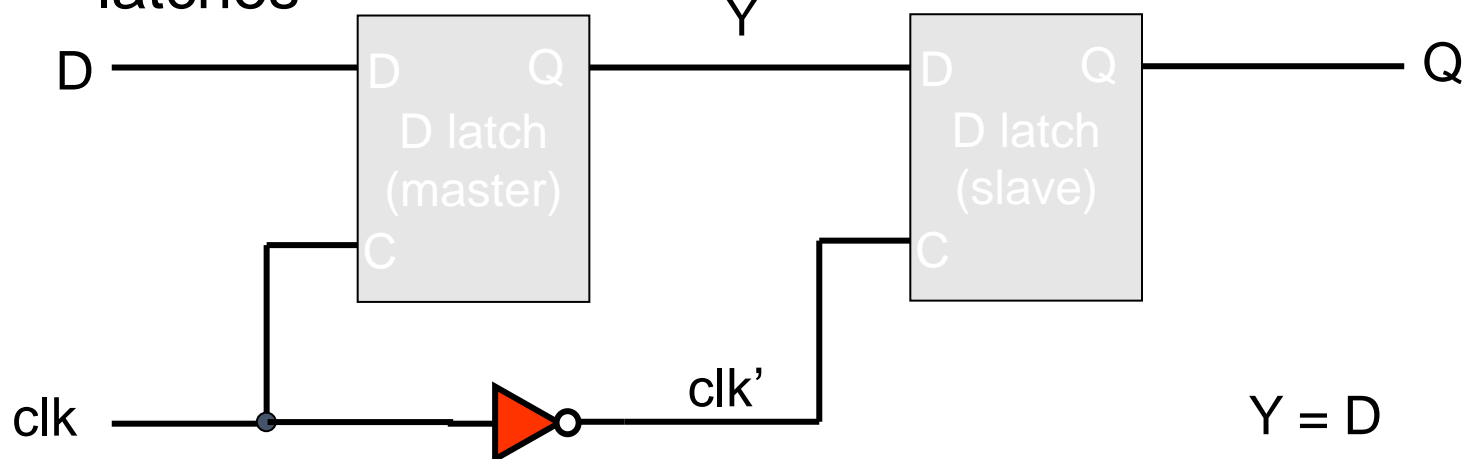
## Need for Flip-Flops 2/2

- Another issue (related to the first one)
  - The states of the memory elements must change synchronously
  - memory elements should respond to the changes in input at certain points in time.
  - This is the very characteristics of synchronous circuits.
  - To this end, we use flip-flops that change states during a signal transition of control input (clock)



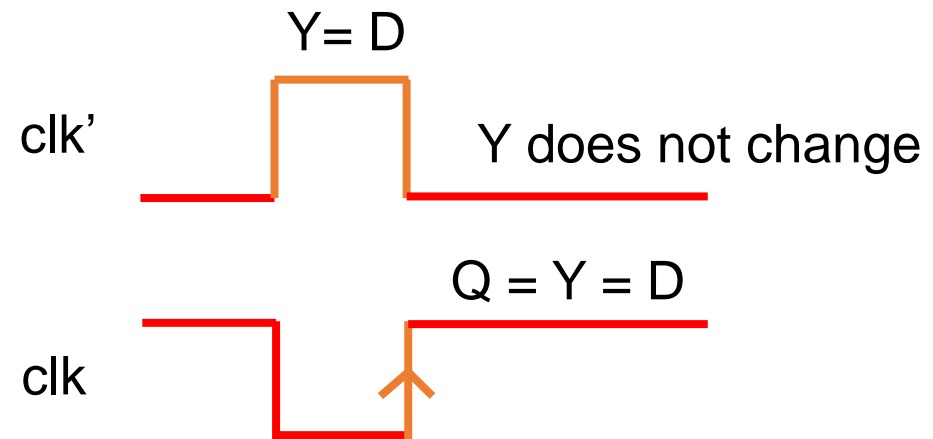
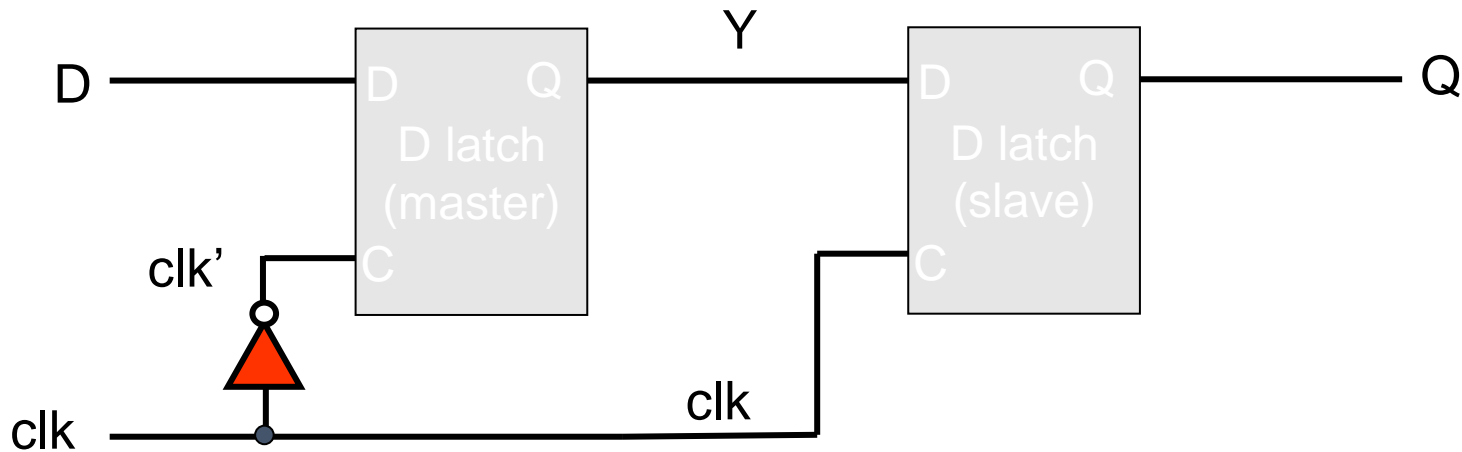
# Edge-Triggered D Flip-Flop

- An edge-triggered D flip-flop can be constructed using two D latches



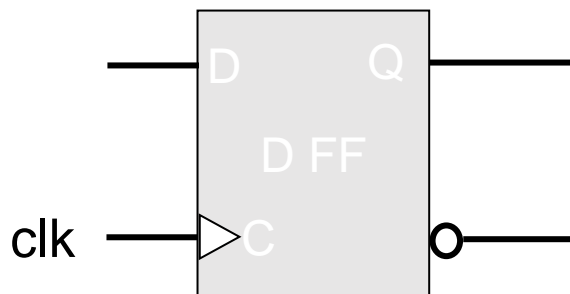
Negative edge-triggered  
D flip-flop

# Positive Edge-Triggered D Flip-Flop

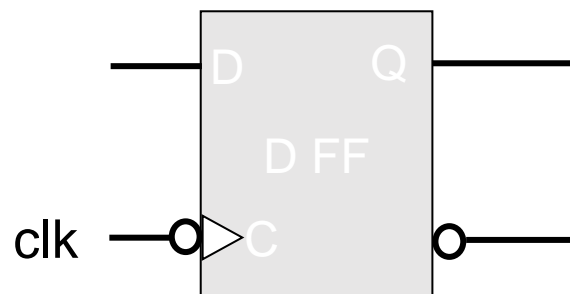


# Symbols for D Flip-Flops

---



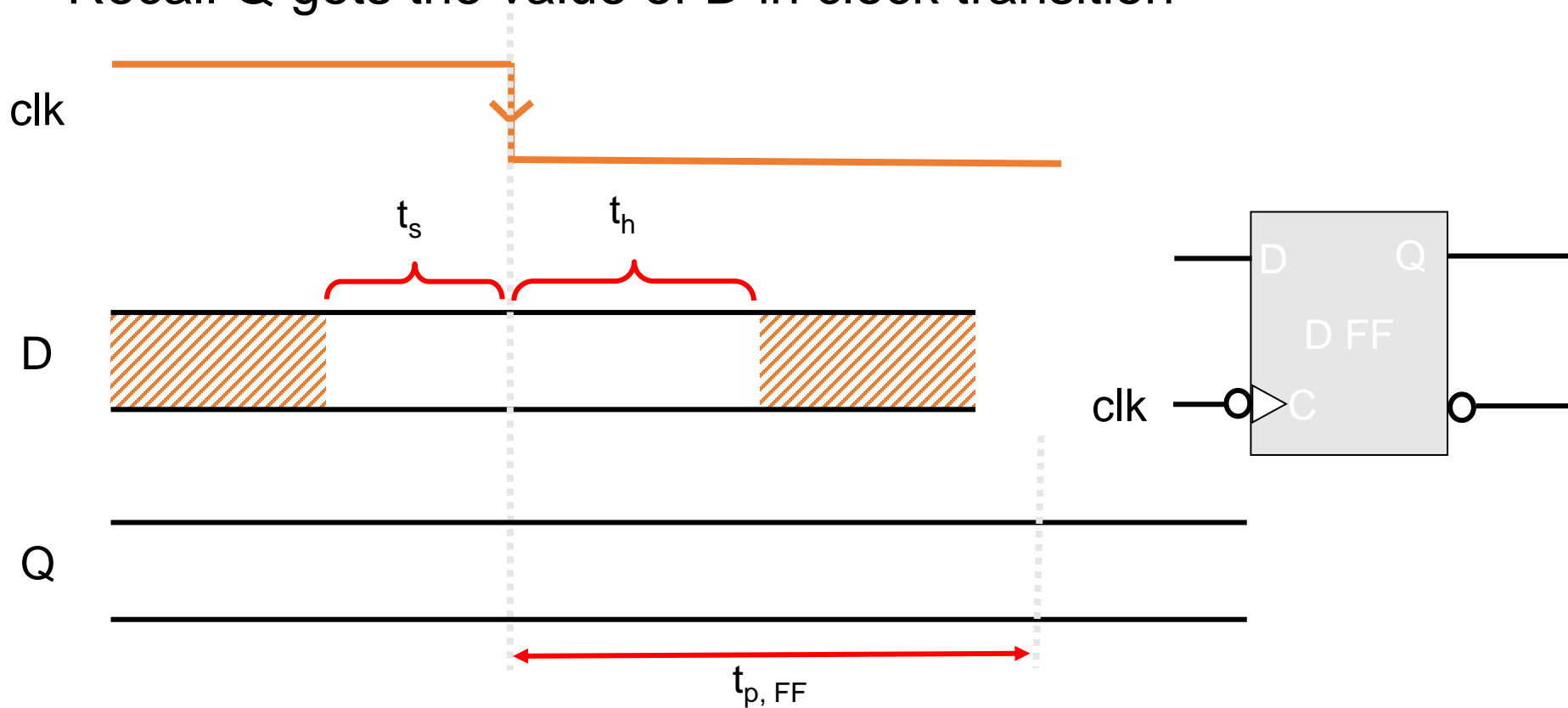
Positive edge-triggered  
D Flip-Flop



Negative edge-triggered  
D Flip-Flop

# Setup & Hold Times 1/2

- Timing parameters are associated with the operation of flip-flops
- Recall Q gets the value of D in clock transition





# Setup & Hold Times 2/2

---

- Setup time,  $t_s$ 
  - The change in the input D must be made before the clock transition.
  - Input D must maintain this new value for a certain minimum amount time.
  - If a change occurs at D less than  $t_s$  second before the clock transition, then the output may not acquire this new value.
  - It may even demonstrate unstable behavior.
- Hold time,  $t_h$ ,
  - Similarly, the value at D must be maintained for a minimum amount of time (i.e.  $t_h$ ) after the clock transition.

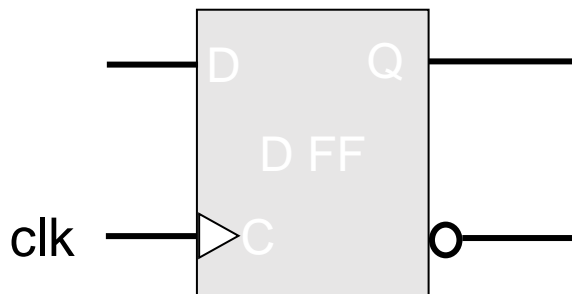


# Propagation Time

---

- Even if setup and hold times are achieved, it takes some time the circuit to propagate the input value to the output.
- This is because of the fact that flip-flops are made of logic gates that have certain propagation times.

# D Flip-Flop



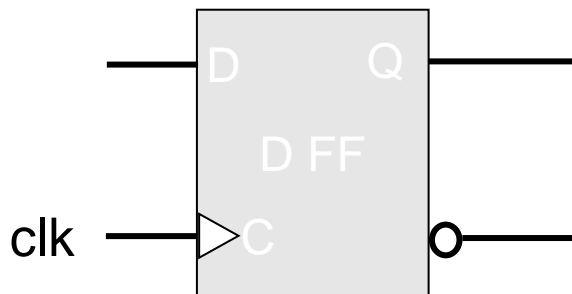
Positive edge-triggered  
D Flip-Flop

- Characteristic equation
  - $Q(t+1) = D$

D	Q(t+1)
0	0
1	1

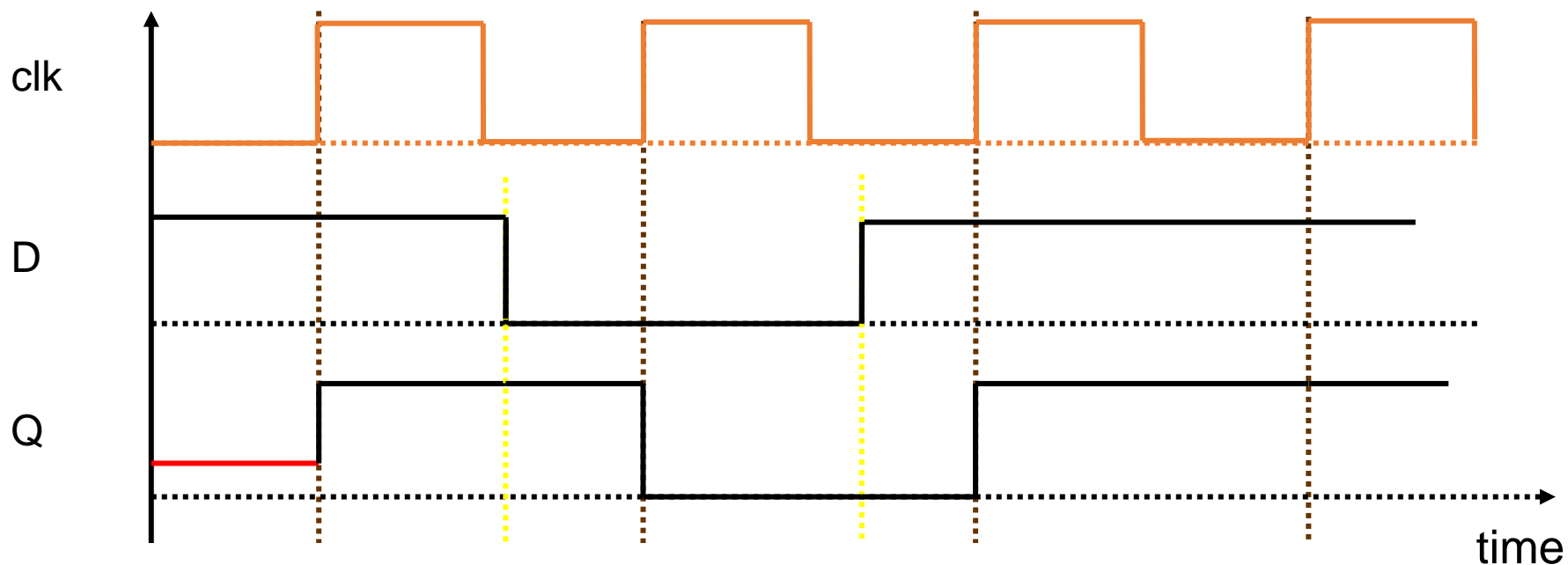
Characteristic Table

# Example



Characteristic equation  
 $Q(t+1) = D$

Positive edge-triggered  
D Flip-Flop



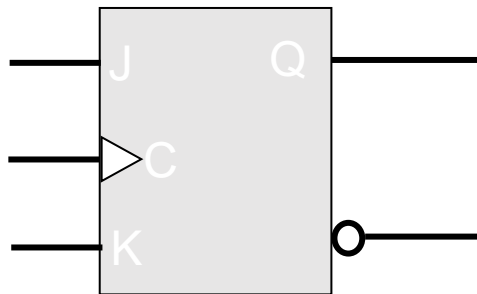


# Other Flip-Flops

---

- D flip-flop is the most common
  - since it requires the fewest number of gates to construct.
- Two other widely used flip-flops
  - JK flip-flops
  - T flip-flops
- JK flip-flops
  - Three FF operations
    1. Set
    2. Reset
    3. Complement

# JK Flip-Flops



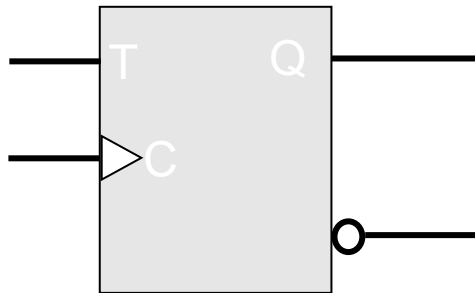
J	K	Q(t+1)	next state
0	0	Q(t)	no change
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Complement

Characteristic Table

- Characteristic equation
  - $Q(t+1) = JQ'(t) + K'Q(t)$

# T (Toggle) Flip-Flop

- Complementing flip-flop

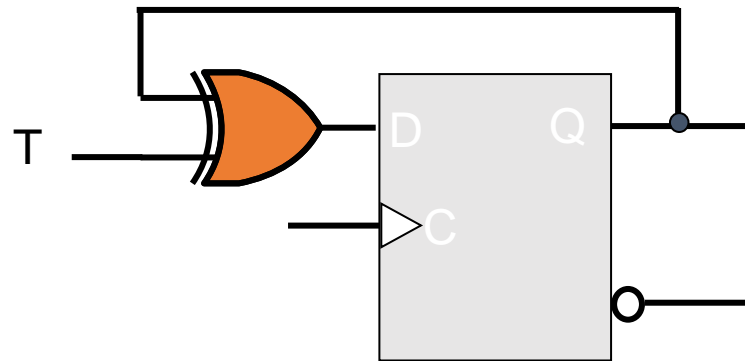
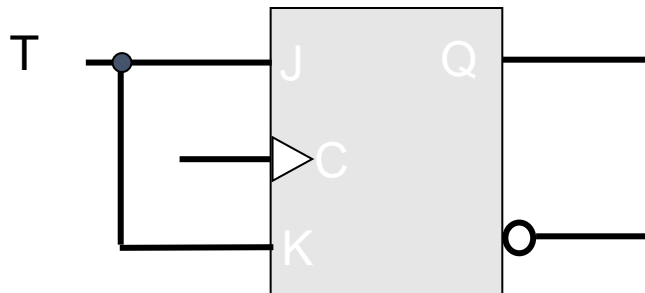


T	Q(t+1)	next state
0	Q(t)	no change
1	Q'(t)	Complement

Characteristic Table

- Characteristic equation

–  $Q(t+1) = T \oplus Q$





# Characteristic Equations

---

- The logical properties of a flip-flop can be expressed algebraically using characteristic equations
- D flip-flop
  - $Q(t+1) = D$
- JK flip-flop
  - $Q(t+1) = JQ' + K'Q$
- T flip-flop
  - $Q(t+1) = T \oplus Q$

# Asynchronous Inputs of Flip-Flops

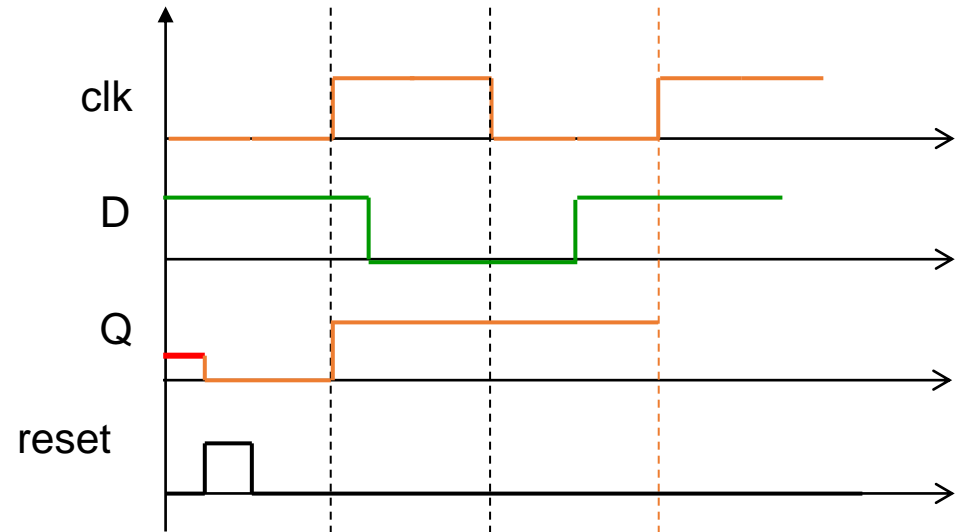
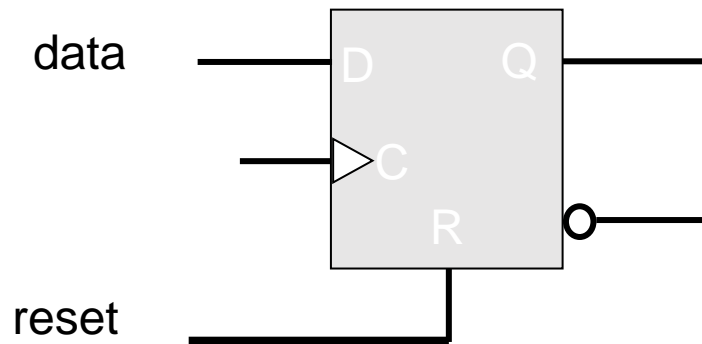
---

- They are used to force the flip-flop to a particular state independent of the clock
  - “Preset” (direct set) set FF state to 1
  - “Clear” (direct reset) set FF state to 0
- They are especially useful at startup.
  - In digital circuits when the power is turned on, the state of flip-flops are unknown.
  - Asynchronous inputs are used to bring all flip-flops to a known “starting” state prior to clock operation.





# Asynchronous Inputs



reset	C	D	Q	Q'	Starting State
1	X	X	0	1	
0	↑	0	0	1	
0	↑	1	1	0	

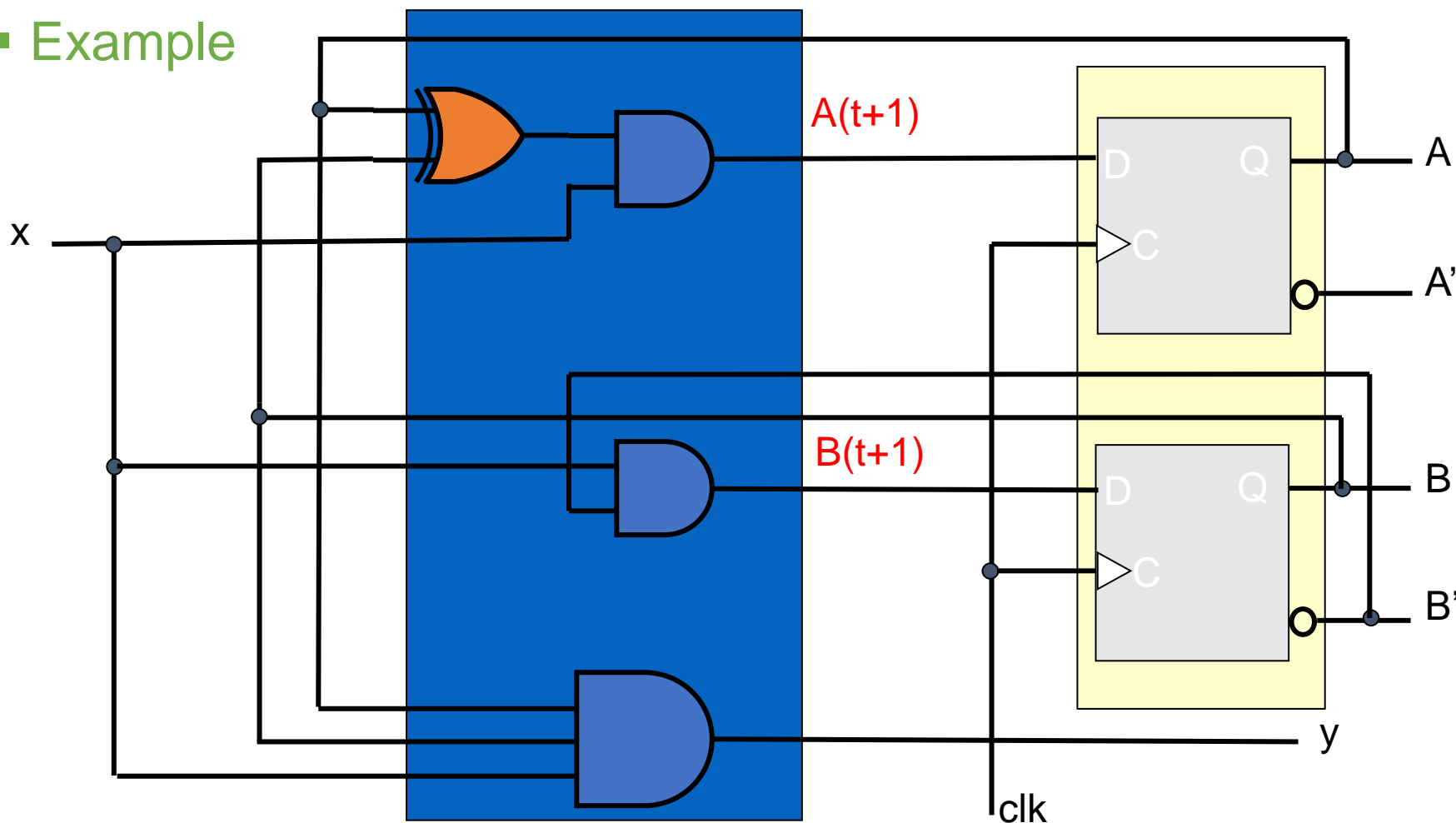
# Analysis of Clocked Sequential Circuits

---

- Goal:
  - to determine the **behavior** of clocked sequential circuits
  - “Behavior” is determined from
    - Inputs
    - Outputs
    - State of the flip-flops
  - We have to obtain
    - Boolean expressions for output and next state
      - output & state equations
    - (state) table
    - (state) diagram

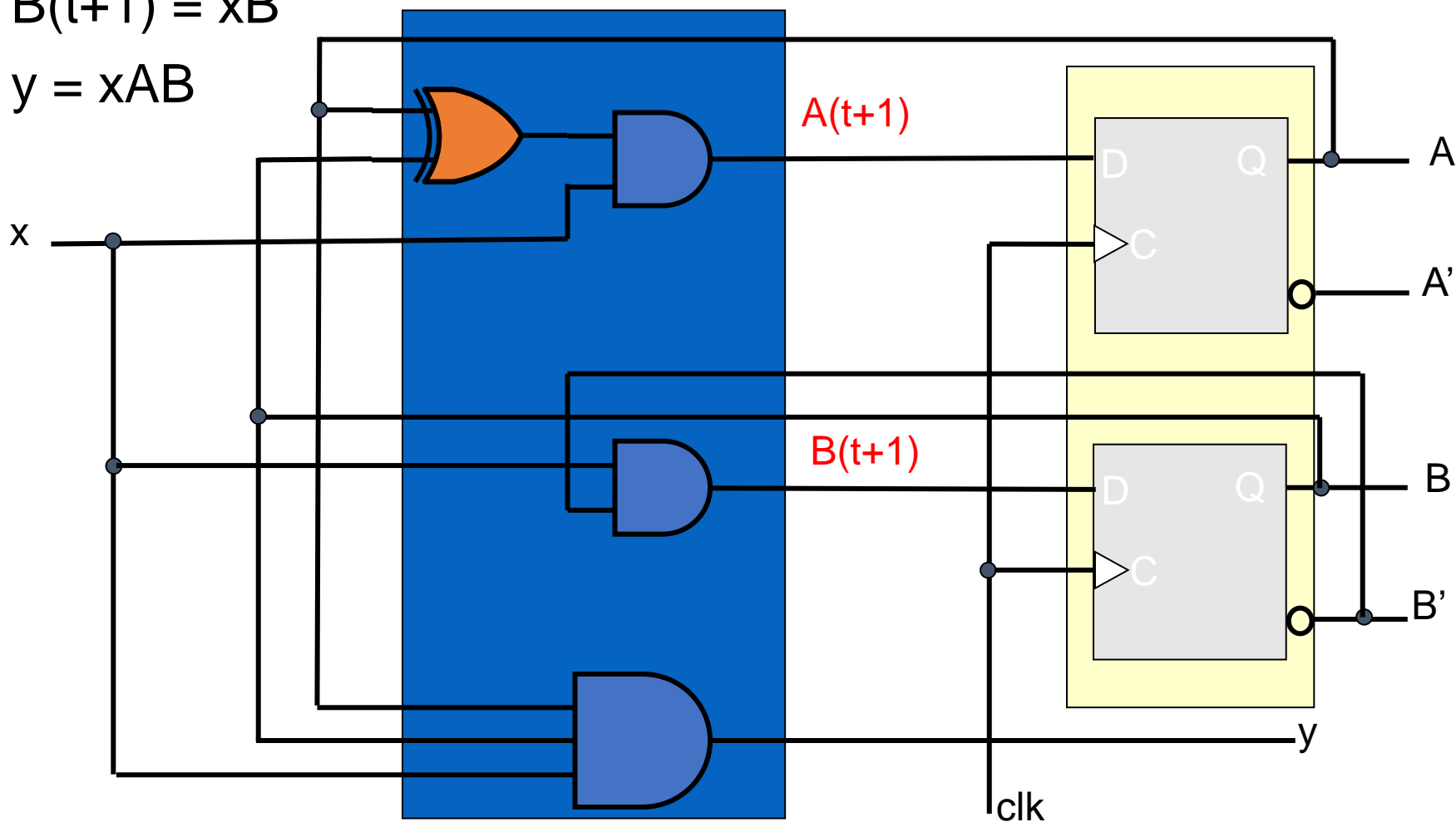
# State Equations

- Also known as “transition equations”
  - specify the next state as a function of the present state and inputs
- Example



# Output and State Equations

- $A(t+1) = x(A \oplus B)$
- $B(t+1) = xB'$
- $y = xAB$





# Flip Flop Input Equations

---

- Flip-Flop input equations
- Same as the state equations in D flip-flops

# Example: State (Transition) Table

$$A(t+1) = x(A \oplus B)$$

$$B(t+1) = xB'$$

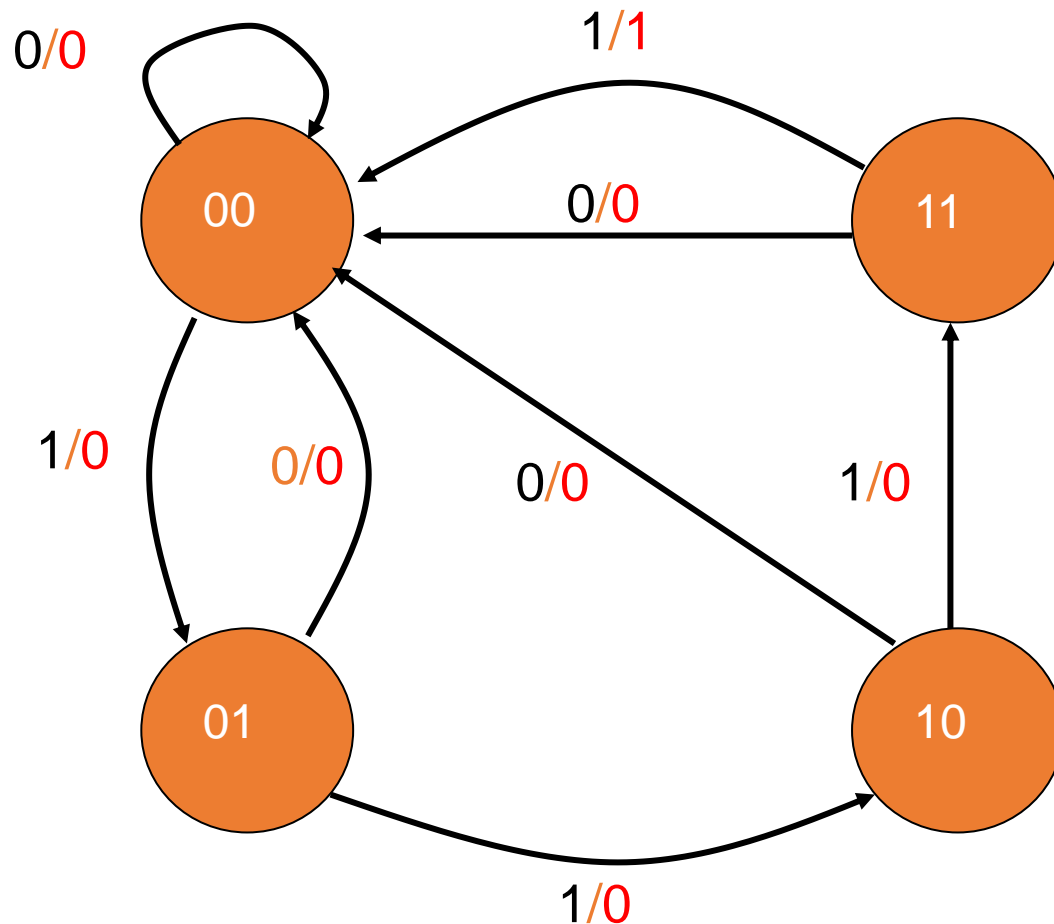
$$y = xAB$$

Present state		input x	Next state		output y
A	B		A	B	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	0	0	1

A sequential circuit with m FFs and n inputs needs  $2^{m+n}$  rows in the transition table



# Example: State Diagram



Present state		input	Next state		output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	0	0	1

What is this circuit doing?

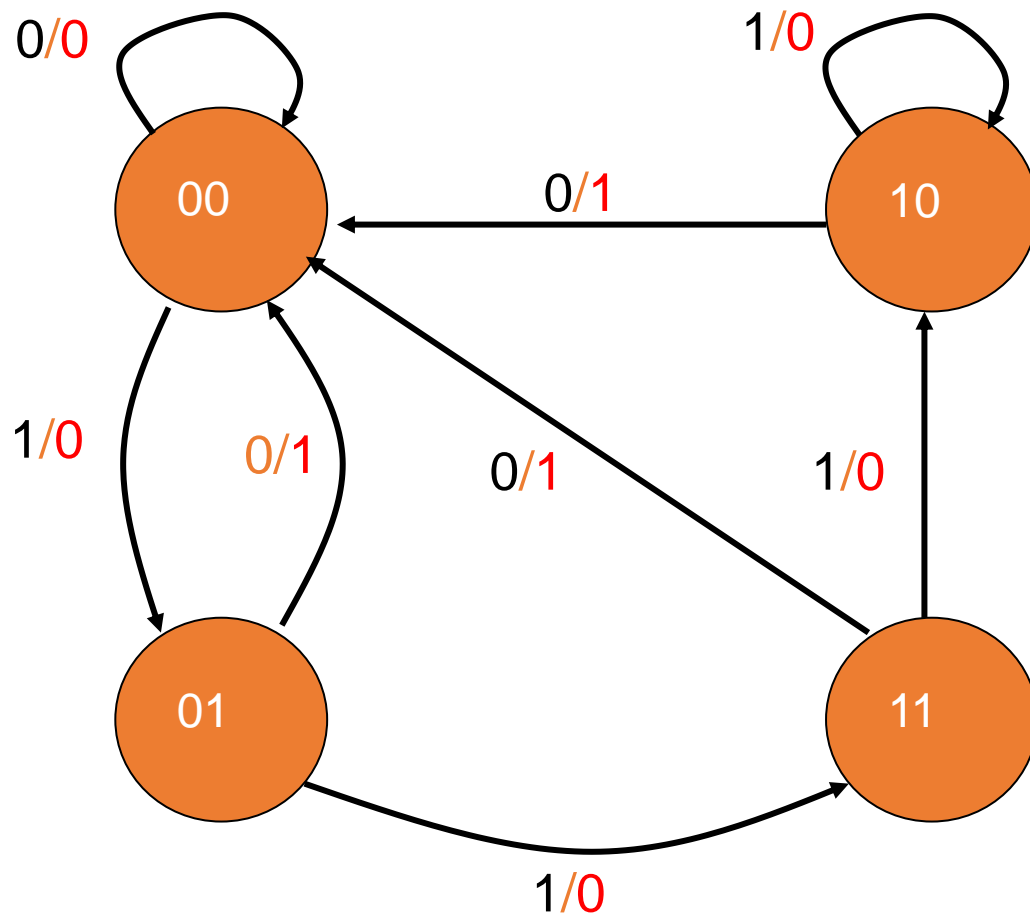
State diagram provides the same information as state table

# Another Example: State Diagram

$$A(t+1) = Ax + Bx$$

$$B(t+1) = xA'$$

$$y = (A+B)x'$$



Present state		input	Next state		output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

What is this circuit doing?

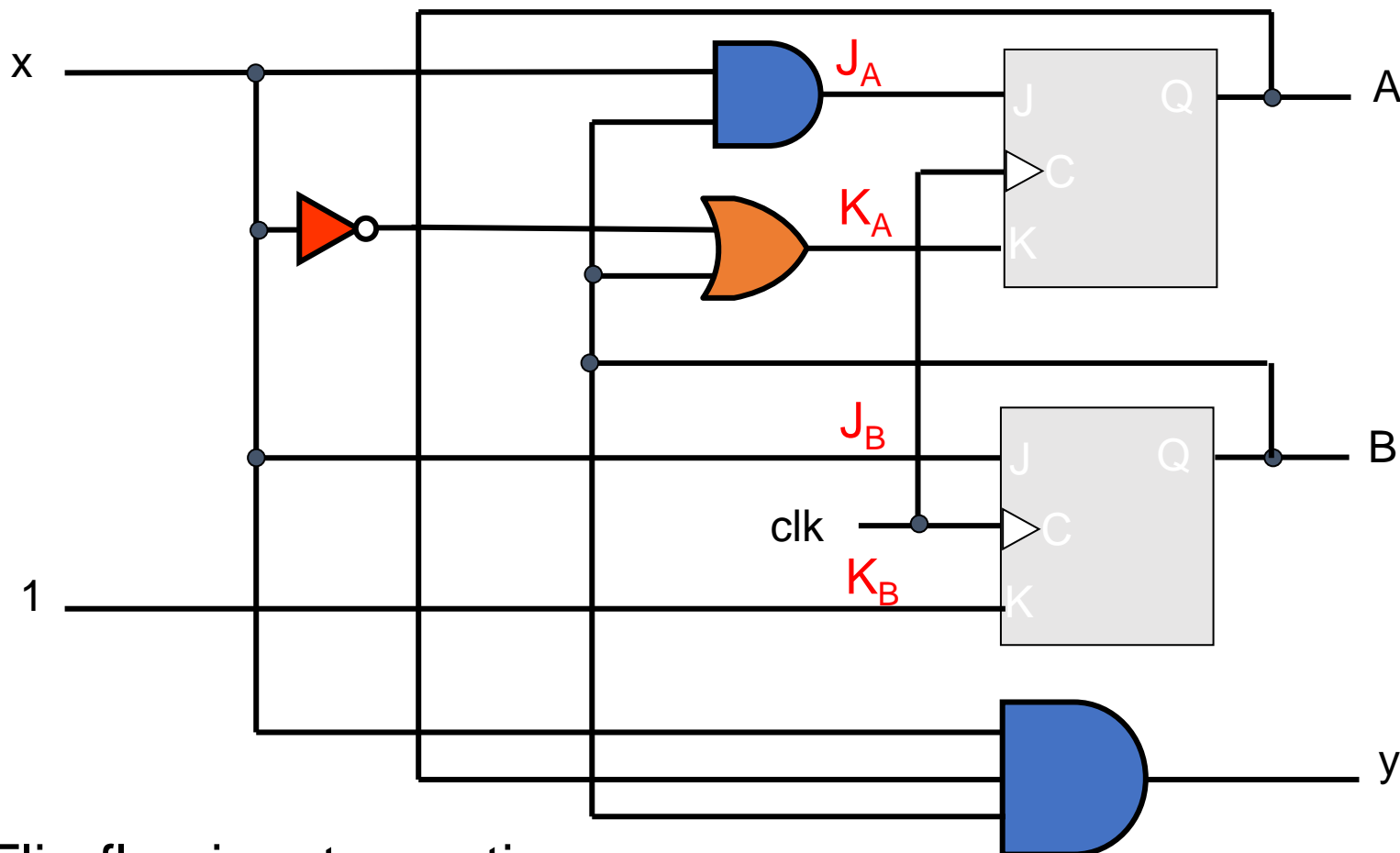


# Analysis with JK Flip-Flops

---

- For a D flip-flop, the state equation is the same as the flip-flop input equation
  - $Q(t+1) = D$
- For JK flip-flops, situation is different
  - Goal is to find state equations
  - Method
    1. Determine flip-flop input equations
    2. Use the corresponding flip-flop characteristic table or characteristic equations to determine the next state values in the state table

# Example: Analysis with JK FFs



## Flip-flop input equations

- $J_A = xB$       and  $K_A = x' + B$
- $J_B = x$       and  $K_B = 1$

# Example: Analysis with JK FFs

- $J_A = Bx$  and  $K_A = x' + B$
- $J_B = x$  and  $K_B = 1$

present State		input	next state		FF inputs			
A	B	x	A	B	$J_A$	$K_A$	$J_B$	$K_B$
0	0	0	0	0	0	1	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	0	0	1	0	1
0	1	1	1	0	1	1	1	1
1	0	0	0	0	0	1	0	1
1	0	1	1	1	0	0	1	1
1	1	0	0	0	0	1	0	1
1	1	1	0	0	1	1	1	1

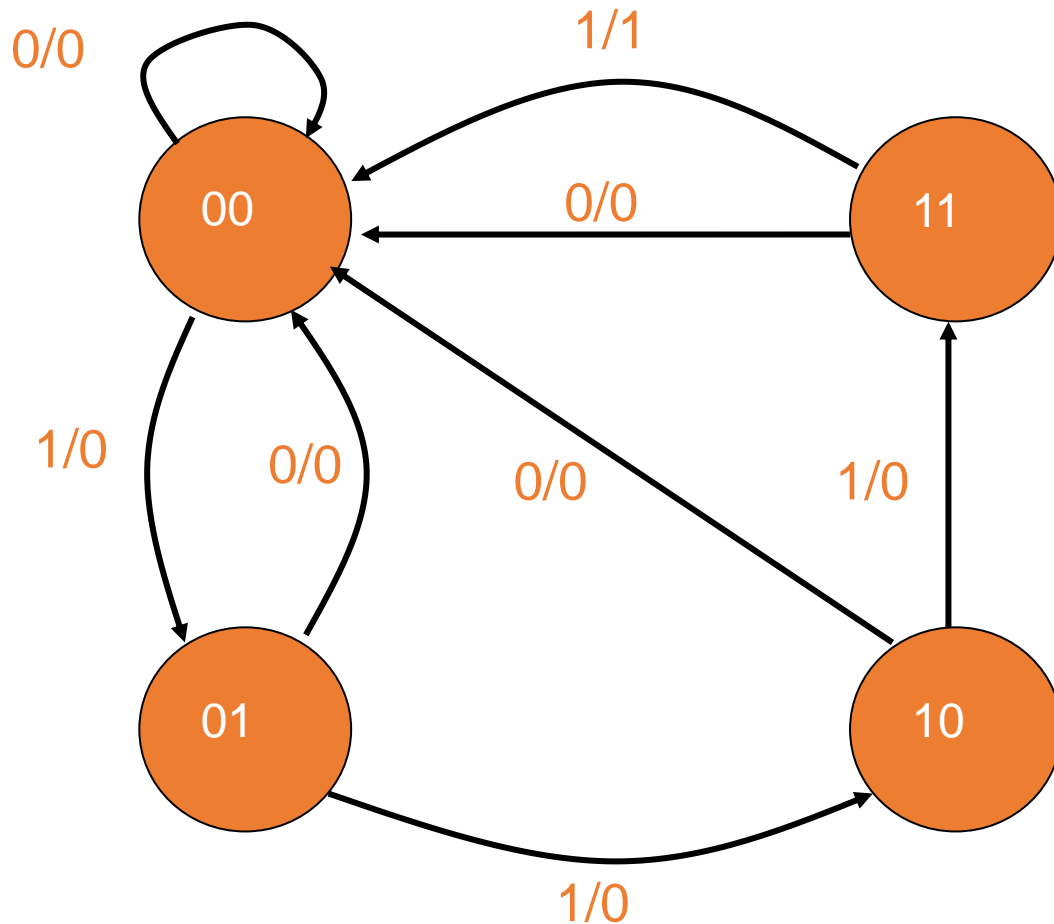
# Example: Analysis with JK FFs

---

- Input equations
  - $J_A = Bx$  and  $K_A = x' + B$
  - $J_B = x$  and  $K_B = 1$
- Characteristic equations
  - $A(t+1) = J_A A' + K'_A A$
  - $B(t+1) = J_B B' + K'_B B$
- State equations
  - $$\begin{aligned} A(t+1) &= J_A A' + K'_A A \\ &= xBA' + (x' + B)'A \\ &= xBA' + xAB' = x(A \oplus B) \end{aligned}$$
  - $$\begin{aligned} B(t+1) &= J_B B' + K'_B B \\ &= xB' \end{aligned}$$



# State Diagram



What is the circuit doing?

$$A(t+1) = x(A \oplus B)$$

$$B(t+1) = xB'$$

$$Y = xAB$$

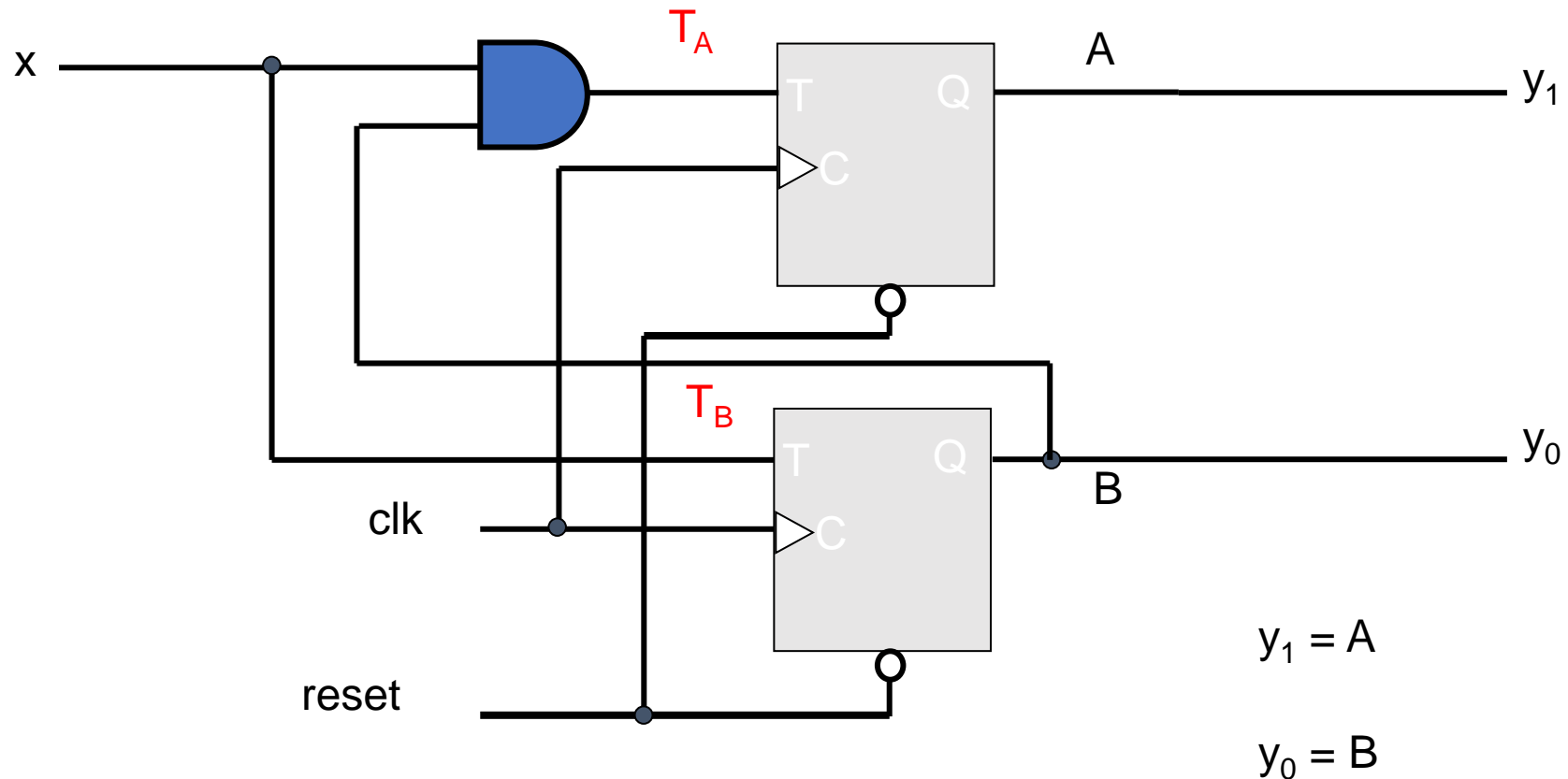
Present state		input	Next state		output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	0	0	1



# Analysis with T Flip-Flops

- Method is the same
- Example

$$T_A = xB$$
$$T_B = x$$



# Example: Analysis with T Flip-Flops

---

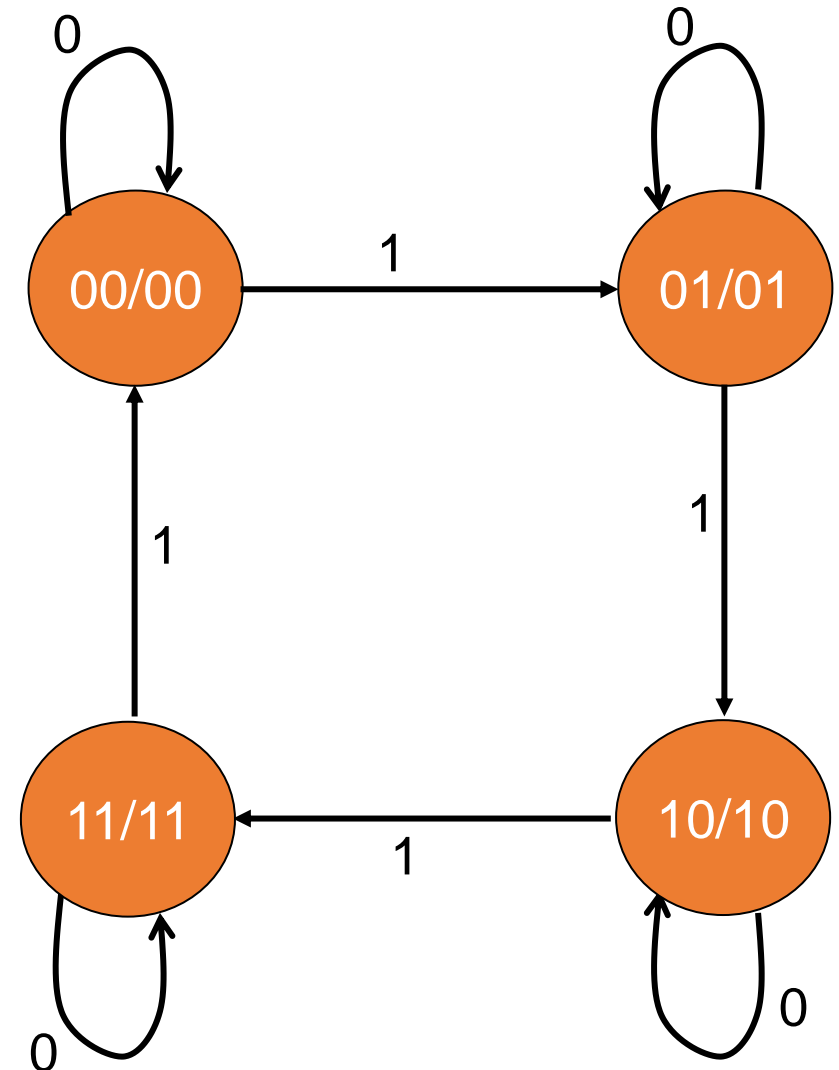
- Input equations
  - $T_A = xB$
  - $T_B = x$
- Characteristic equation
  - $A(t+1) = T_A \oplus A$
  - $B(t+1) = T_B \oplus B$
- Output equations
  - $y_1 = A$
  - $y_0 = B$
- State equations
  - $A(t+1) = xB \oplus A$
  - $B(t+1) = x \oplus B$



# State Table & Diagram

- $A(t+1) = xB \oplus A$
- $B(t+1) = x \oplus B$
- $y_1 = A; y_0 = B$

Present state		input x	Next state		output	
A	B		A	B	$y_1$	$y_0$
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	0	1	0	1
0	1	1	1	0	0	1
1	0	0	1	0	1	0
1	0	1	1	1	1	0
1	1	0	1	1	1	1
1	1	1	0	0	1	1





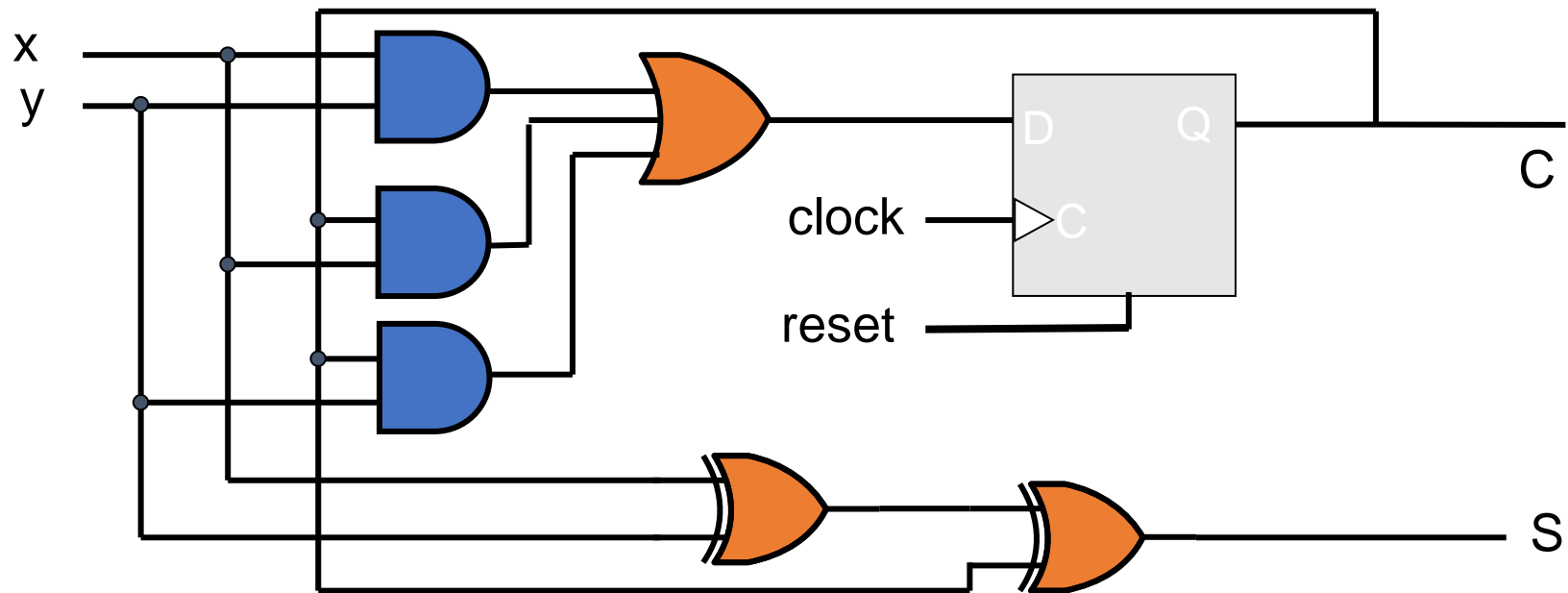


# Mealy and Moore Models

---

- There are two models for sequential circuits
  - Mealy
  - Moore
- They differ in the way the outputs are generated
  - Mealy:
    - output is a function of both present states and inputs
  - Moore
    - output is a function of present state only

# Example: Mealy and Moore Machines



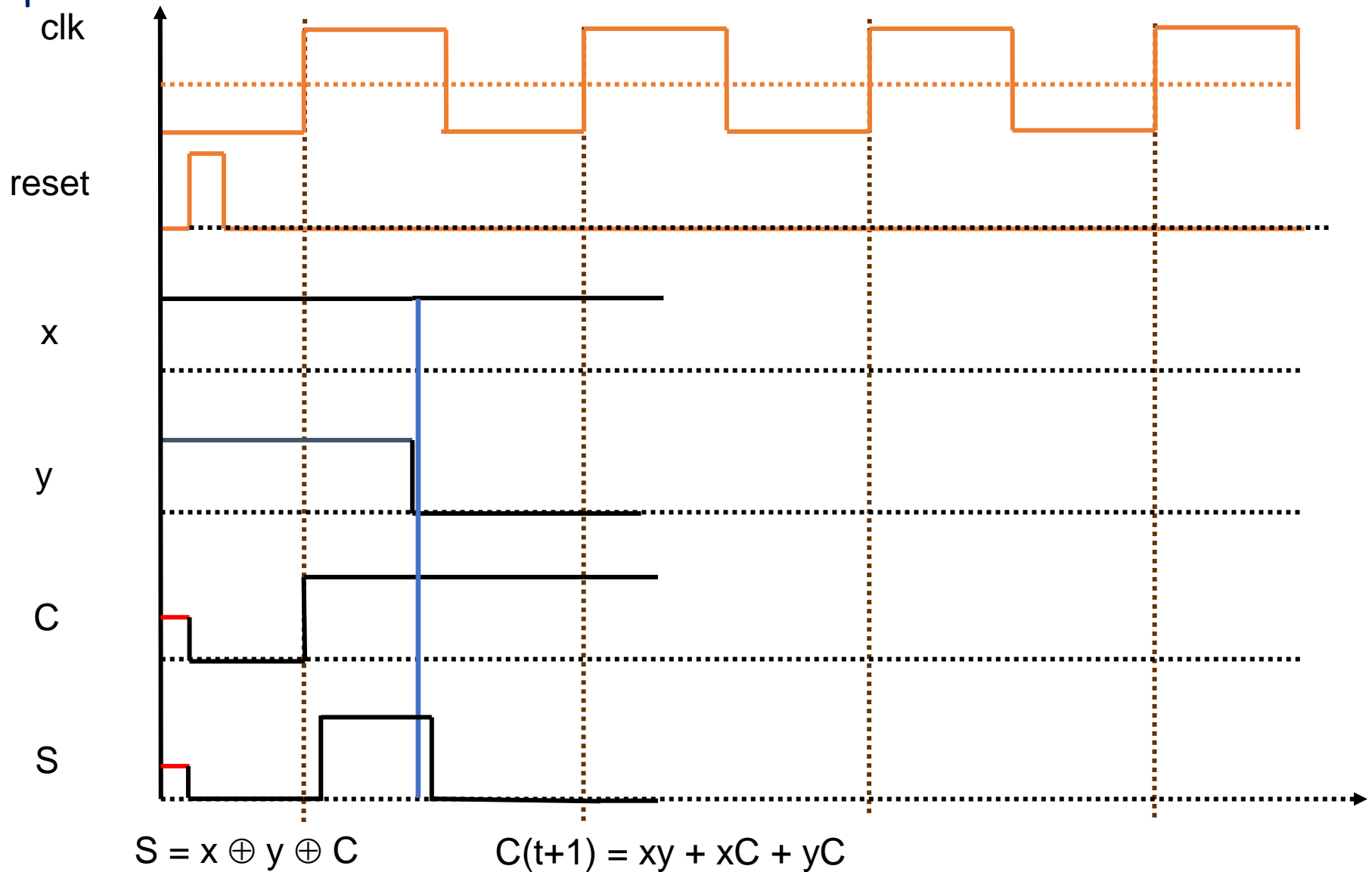
Mealy machine

- External inputs,  $x$  and  $y$ , can be asynchronous
- Thus, Mealy outputs may have momentary (sometimes incorrect) values
- Inputs must be synchronized with clocks
- Outputs must be sampled only during clock edges

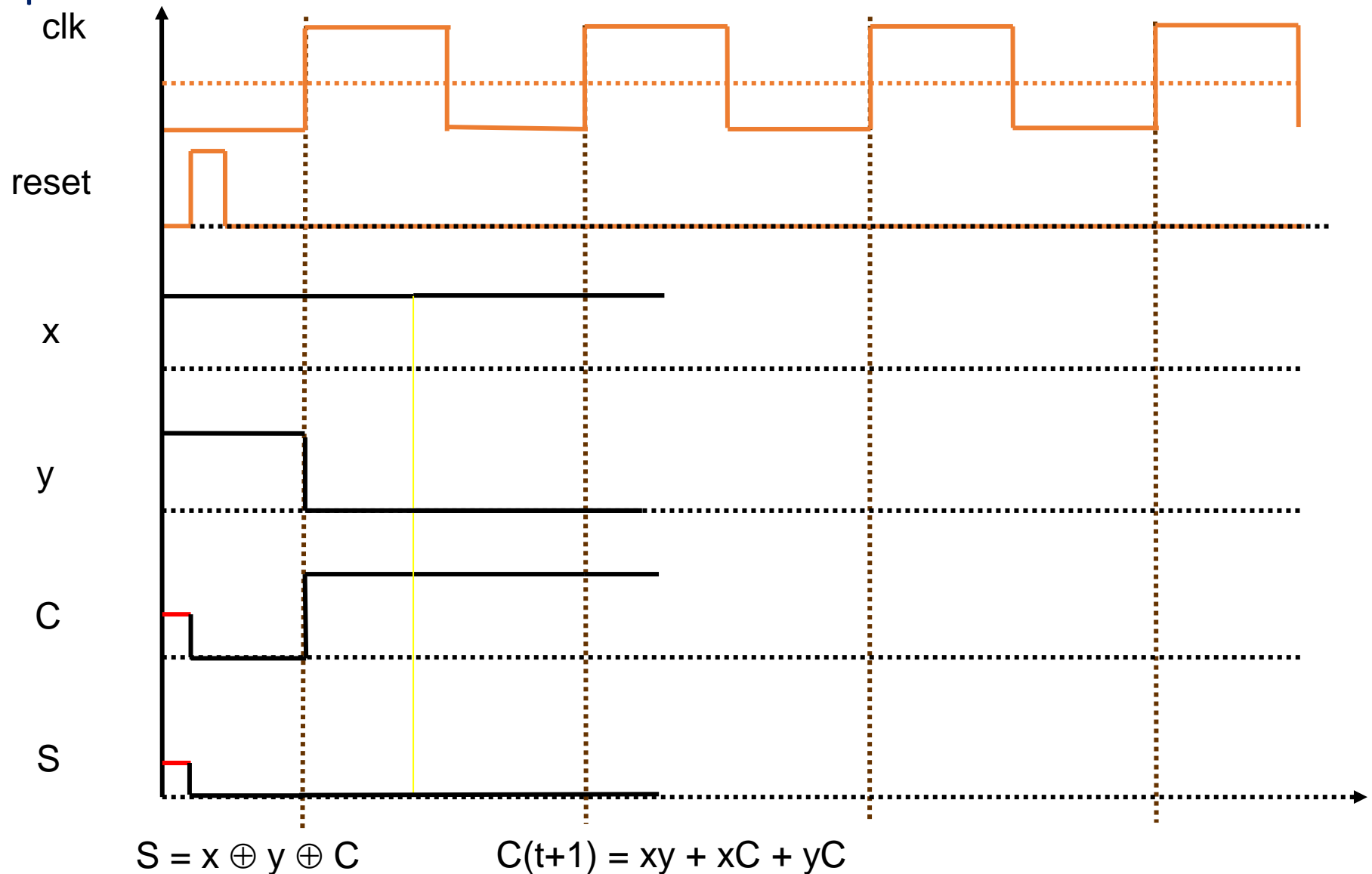


# Timing Diagram

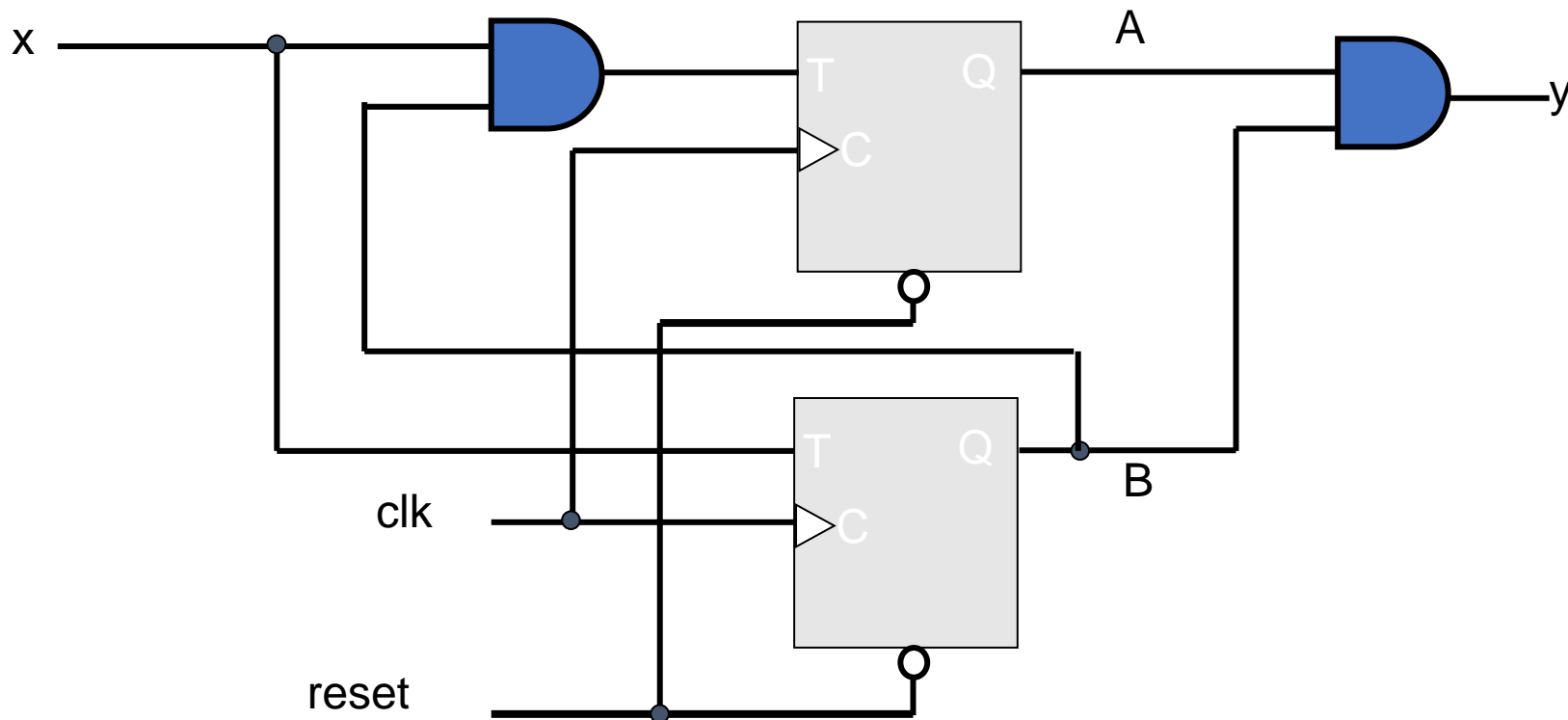
$$(11) + (01) = 100$$



# Timing Diagram



# Example: Moore Machines



- Outputs are already synchronized with clock.
- They change synchronously with the clock edge.