



CSCI 150

Introduction to Digital and Computer System Design

Lecture 4: Sequential Circuit III



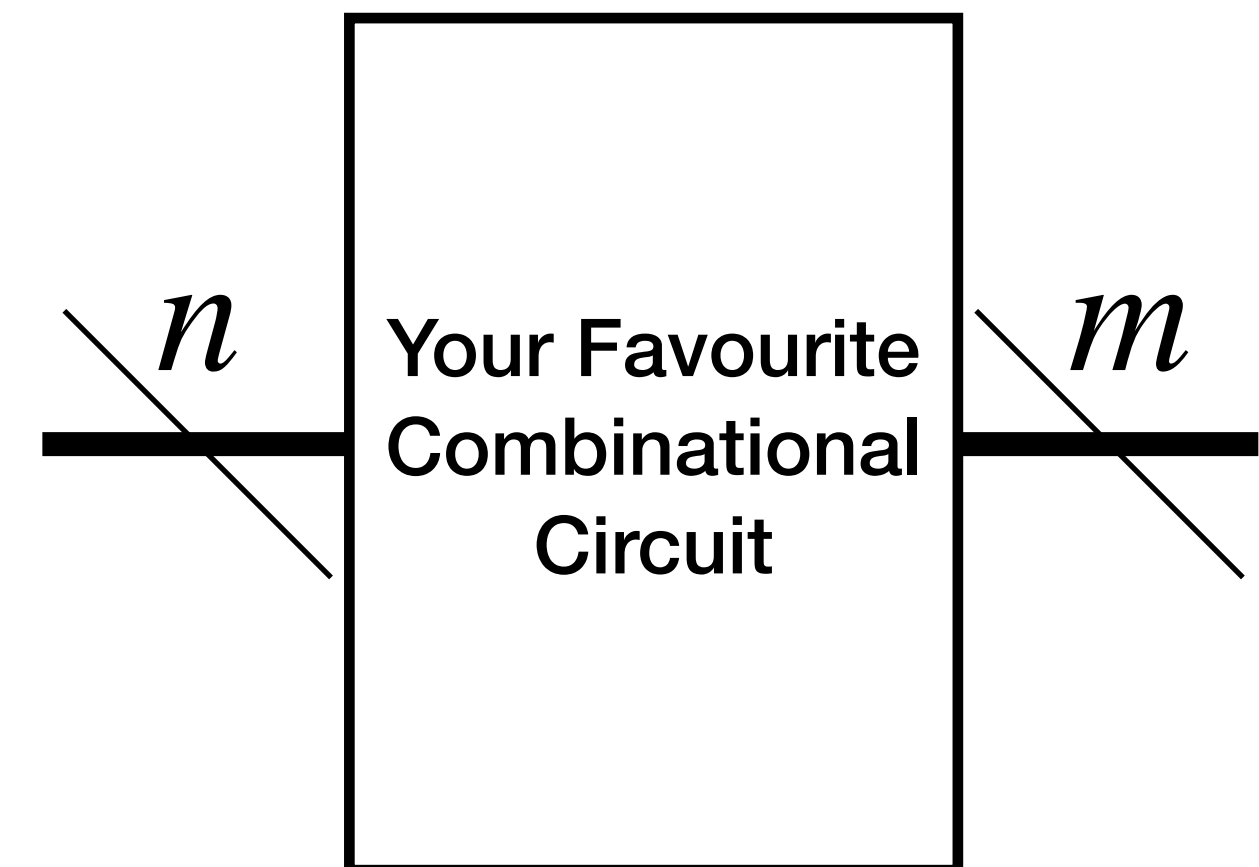
Jetic Gū
2020 Fall Semester (S3)

Overview

- Focus: Basic Information Retaining Blocks
- Architecture: Sequential Circuit
- Textbook v4: Ch5 5.3, 5.4; v5: Ch4 4.2 4.3
- Core Ideas:
 1. Latches and Flip-Flops (with Direct Input)
 2. Sequential Circuit Analysis

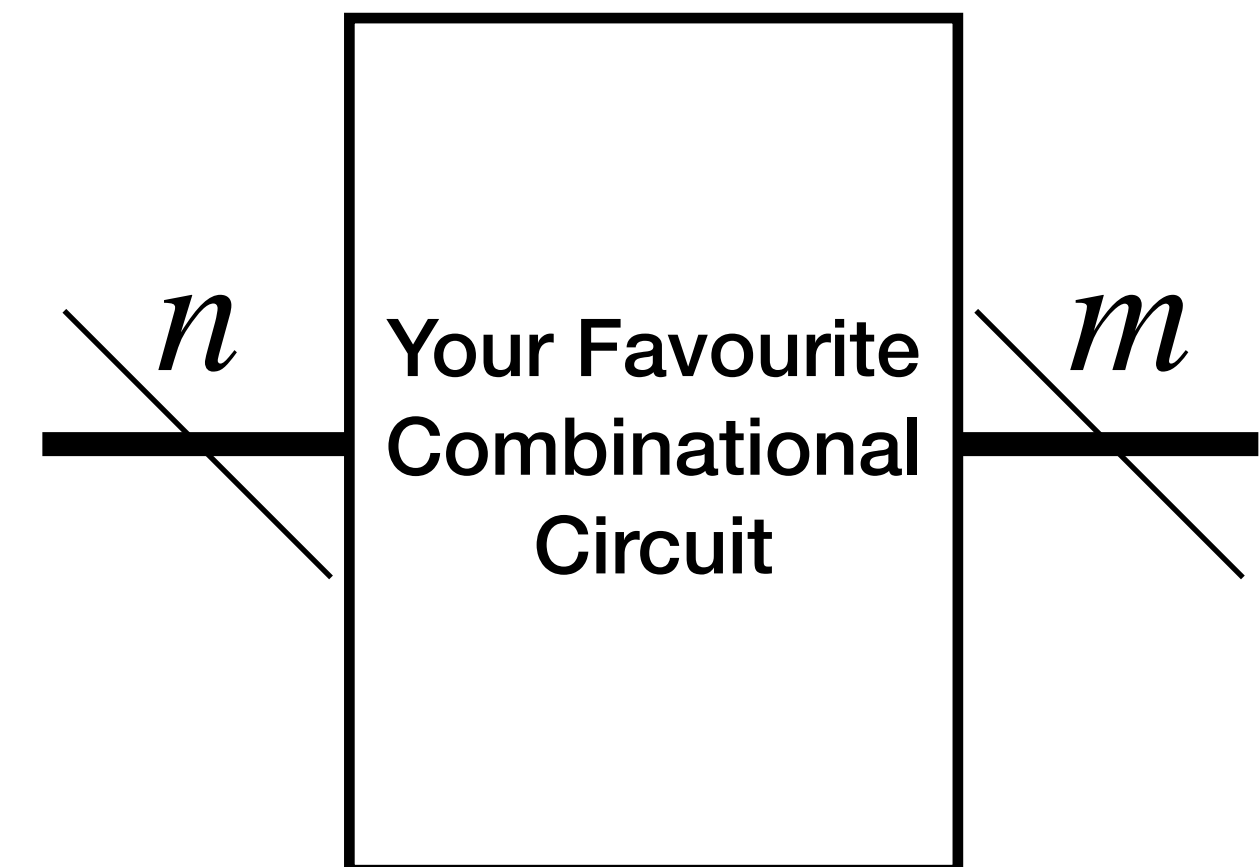
Combinational Logic Circuit Design

- Design Principles
 - Knows: fixed-Length input and output
 - Knows: input/output mapping relations
 - Optimisation: Minimise overall delay



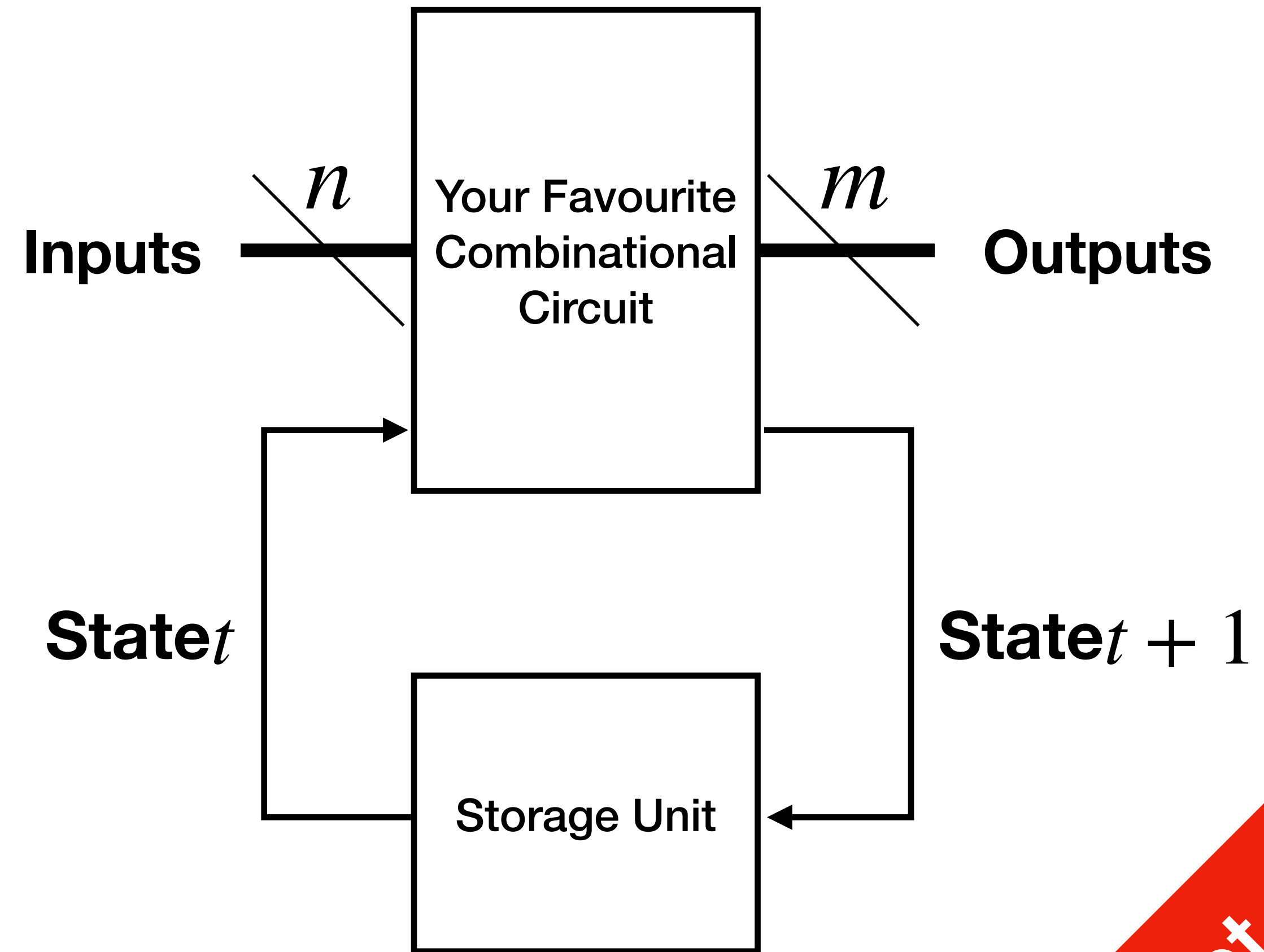
Combinational Logic Circuit Design

- Cannot handle variable length input
- Cannot store information
- Cannot perform multi-step tasks



Definitions

1. **Storage Elements**
circuits that can store binary information
2. **State**
partial results, instructions, etc.
3. **Synchronous Sequential Circuit**
Signals arrive at discrete instants of time,
outputs at next time step
4. **Asynchronous Sequential Circuit**
Signals arrive at any instant of time,
outputs when ready



Definitions

3. Synchronous Sequential Circuit

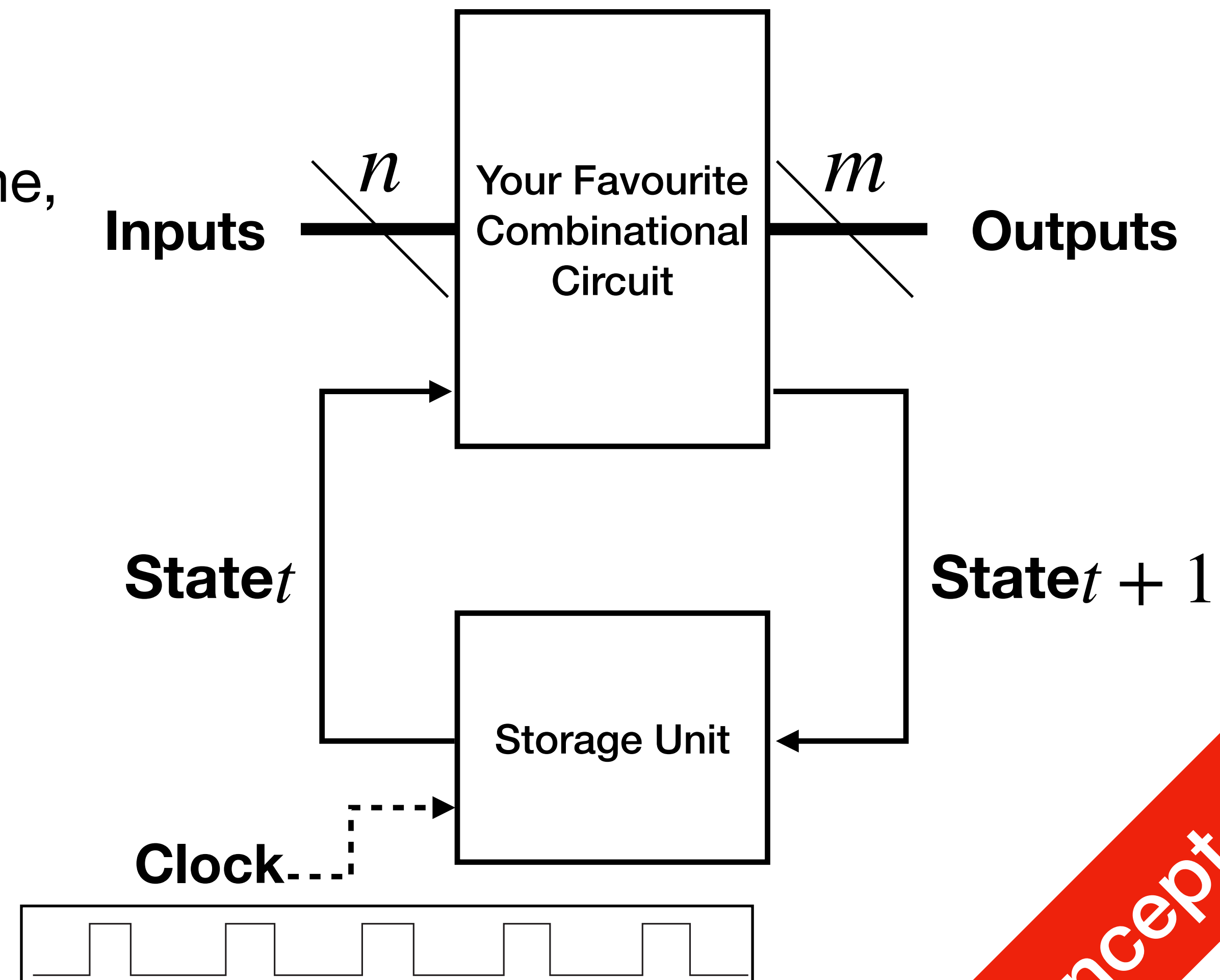
Signals arrive at discrete instants of time, outputs at next time step

- Has Clock

4. Asynchronous Sequential Circuit

Signals arrive at any instant of time, outputs when ready

- May not have Clock

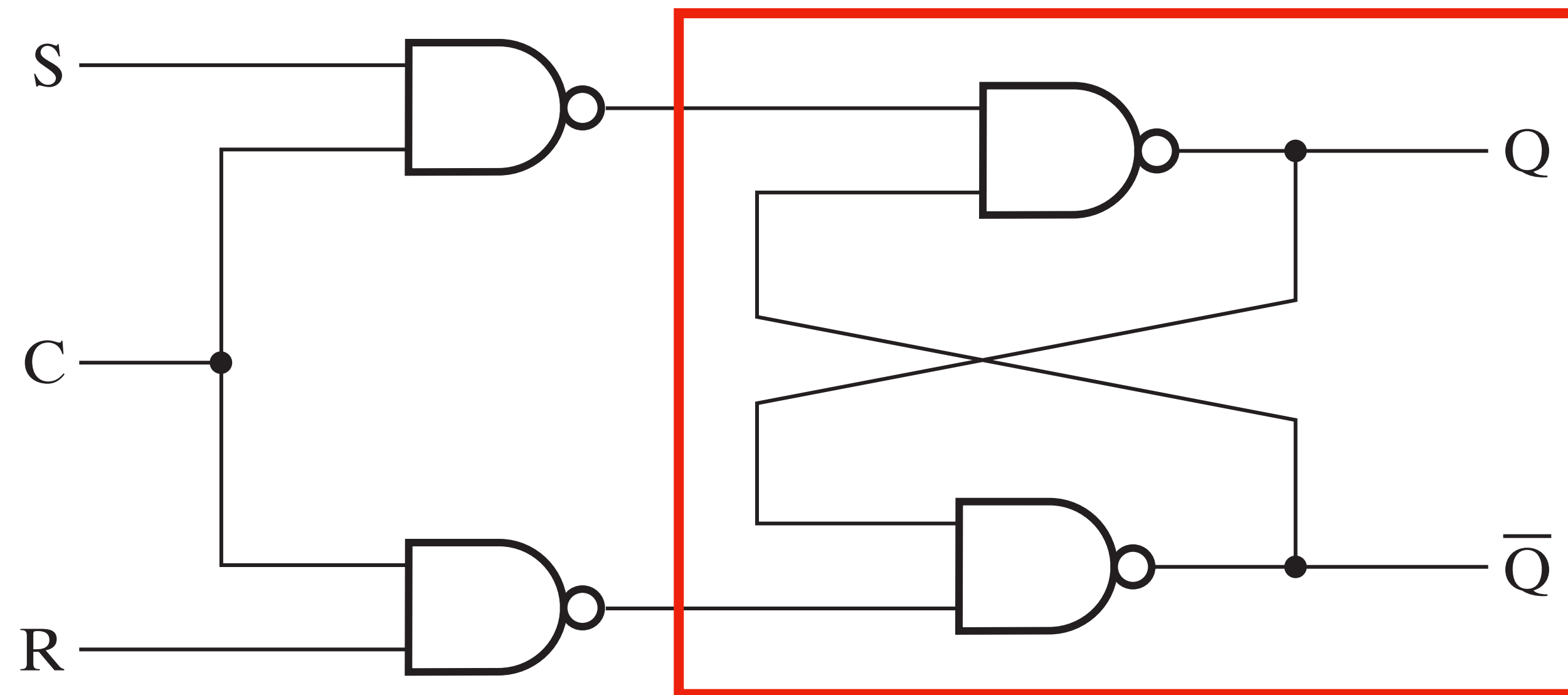


Review:

Latches and Flip-Flops

SR Latches, D Latches, D Flip-Flops

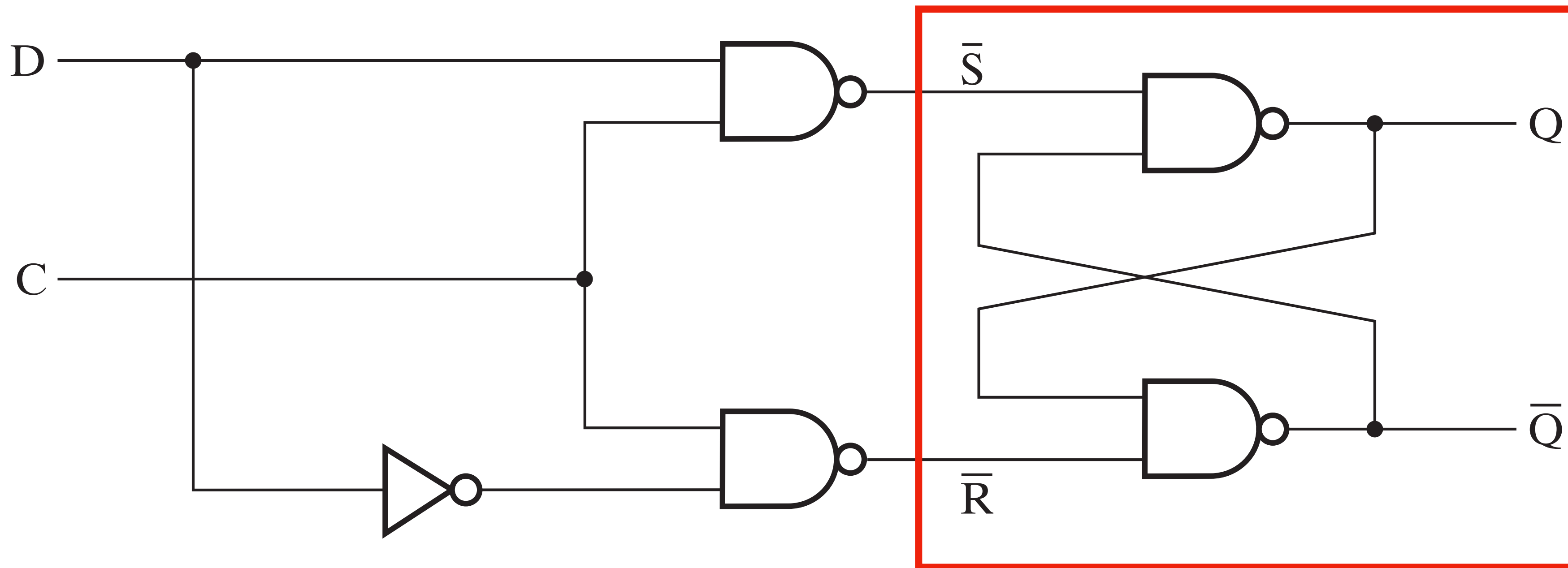
SR Latch with Control Input



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	Undefined

- Implemented using \overline{SR} latches
- C acts as an enabler; otherwise the entire circuit functions as an SR latch

D Latch



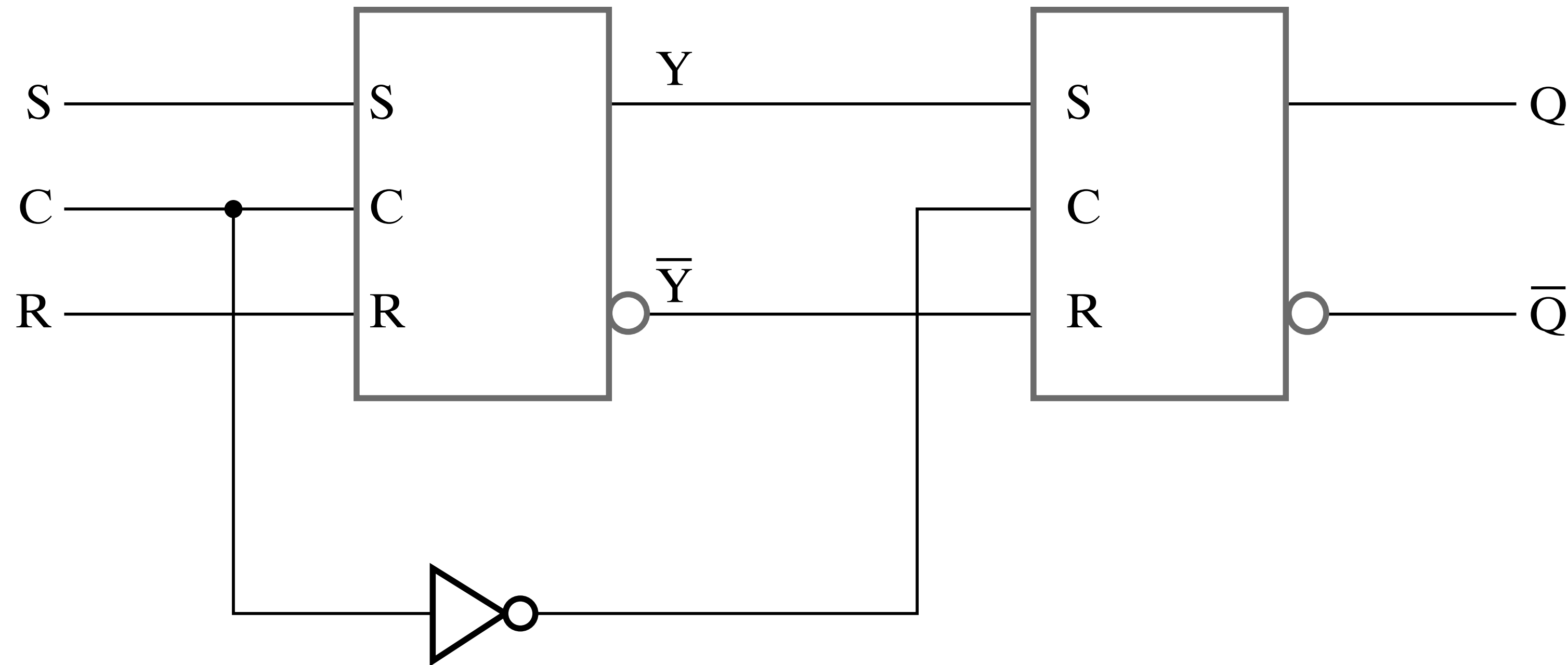
C	D	Next state of Q
0	X	No change
1	0	Q = 0; Reset state
1	1	Q = 1; Set state

- Implemented using \overline{SR} latches
- C : Signals changes to the stored states; D the value to change to

Flip-Flops

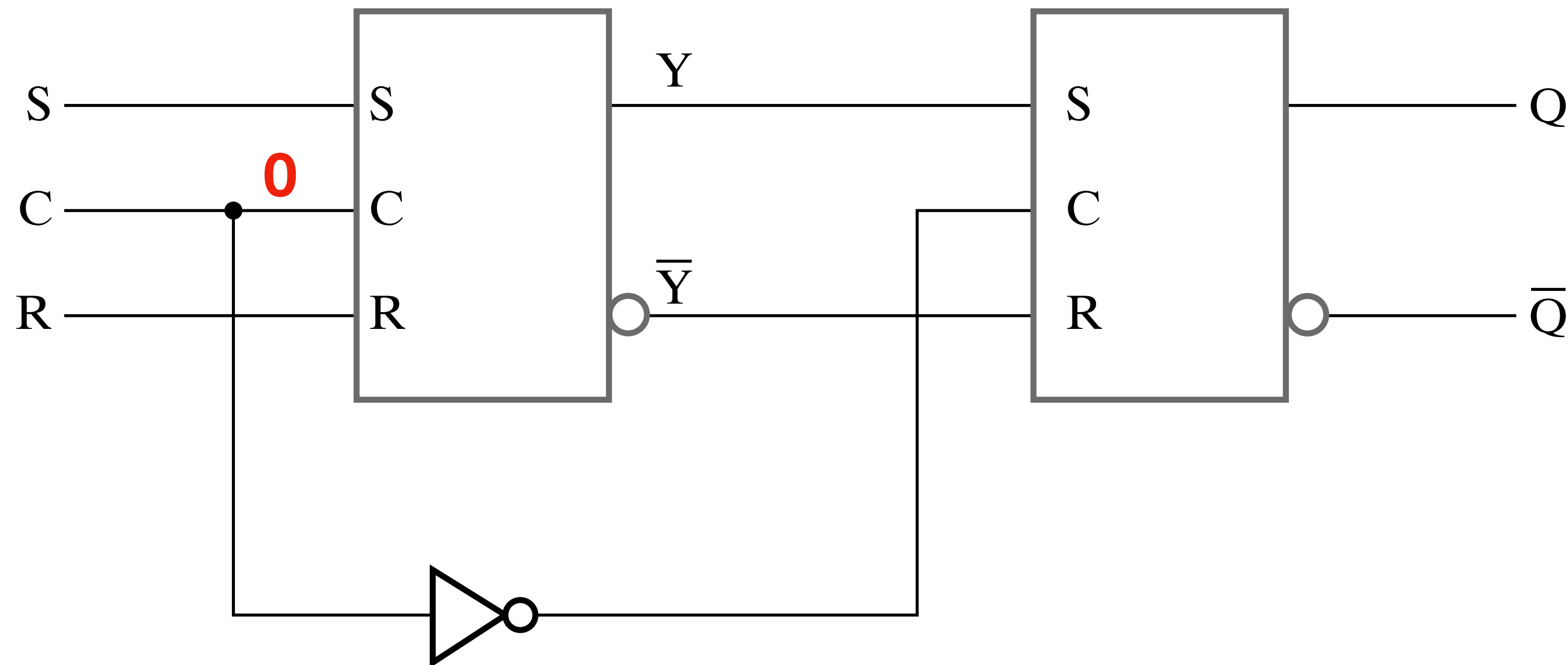
- Latches are **Transparent**
input can be seen from outputs while control pulse is 1
- Flip-Flops are not **Transparent**
Output state changes require changes of control signal

SR Master-Slave Flip-Flop



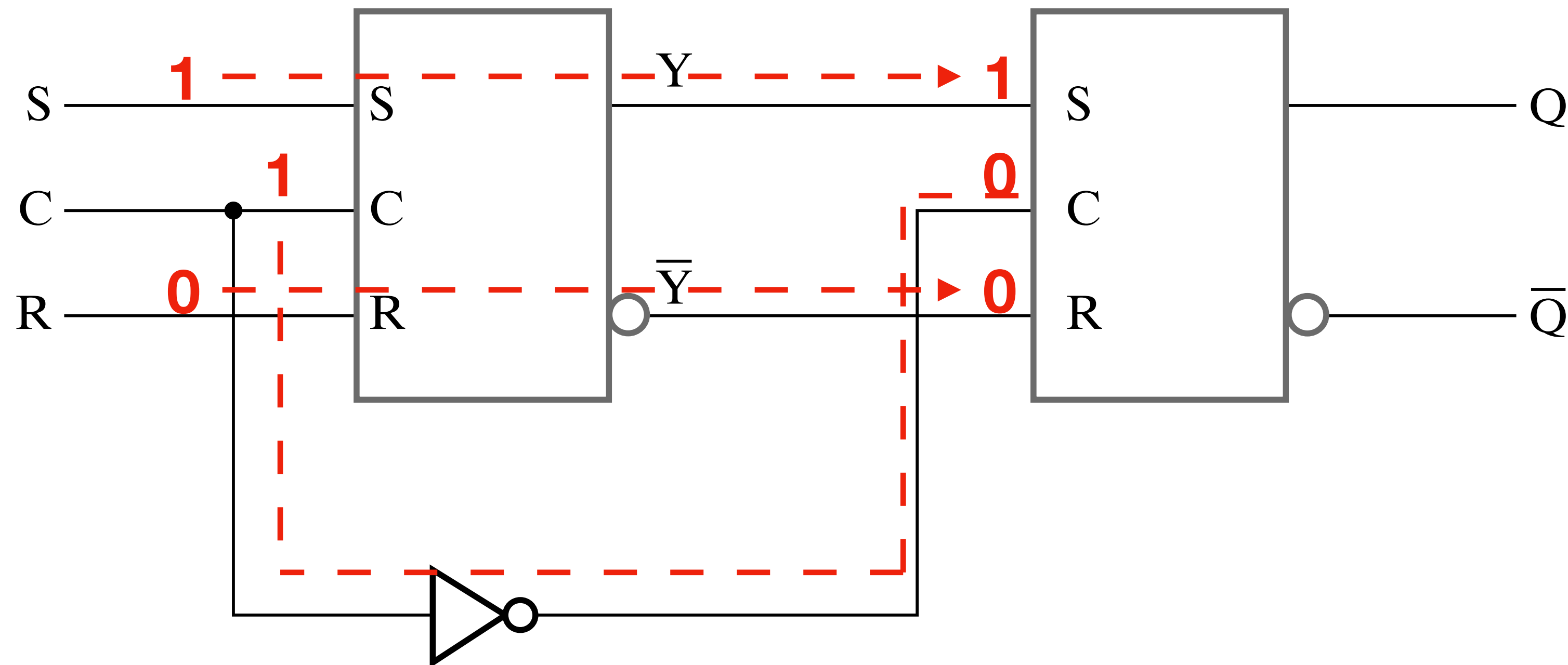
- Constructed using *SR* latches, left Master, right Slave
- Output state changes require $C = 0 \rightarrow C = 1 \rightarrow C = 0$ (Positive Pulse)

SR Master-Slave Flip-Flop



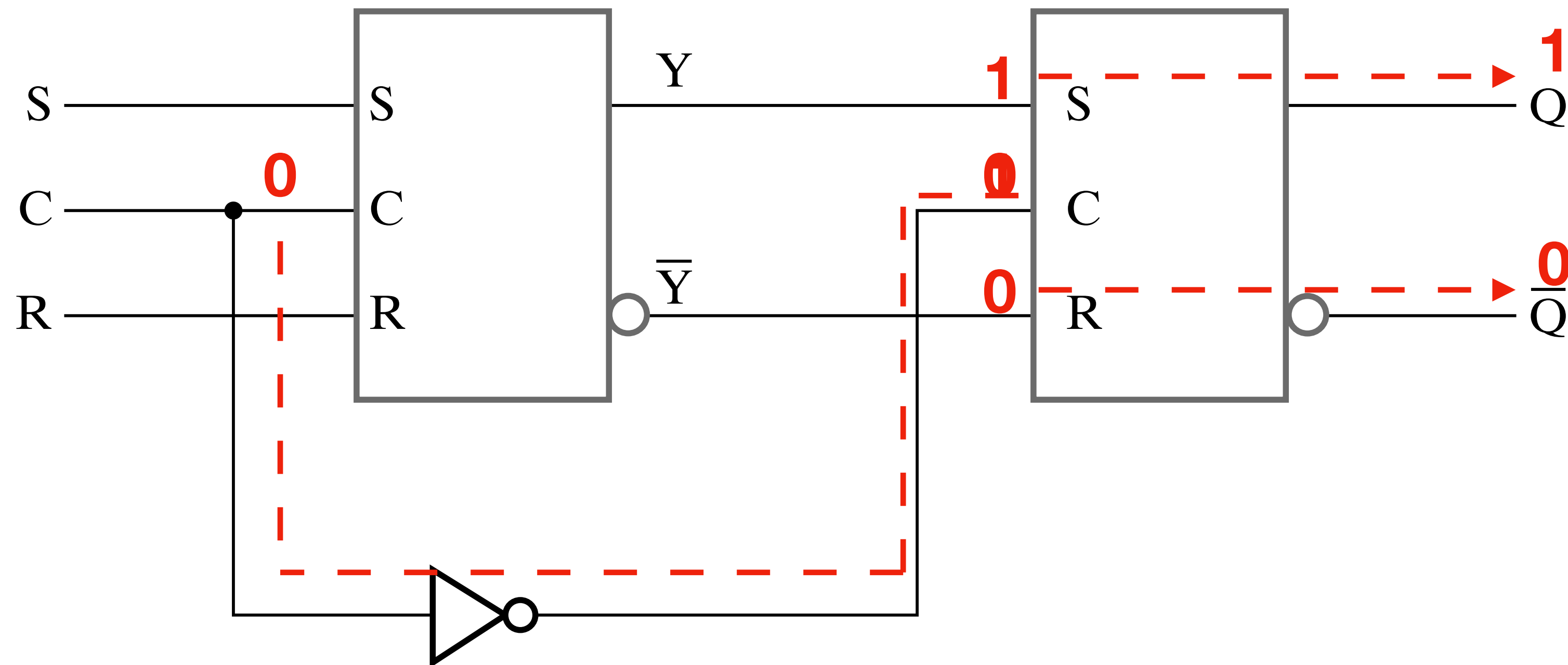
- Constructed using *SR* latches, left Master, right Slave
- Output state changes require $C = 0 \rightarrow C = 1 \rightarrow C = 0$ (Positive Pulse)

SR Master-Slave Flip-Flop



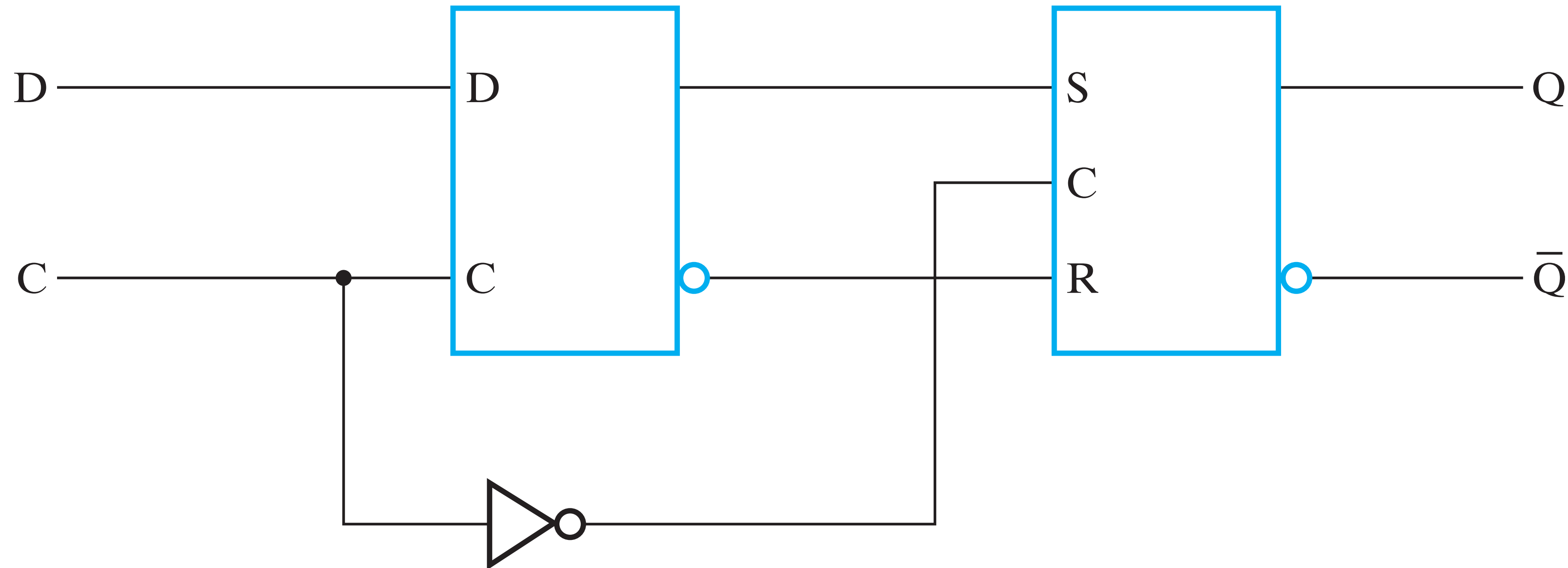
- Constructed using *SR* latches, left Master, right Slave
- Output state changes require $C = 0 \rightarrow C = 1 \rightarrow C = 0$ (Positive Pulse)

SR Master-Slave Flip-Flop



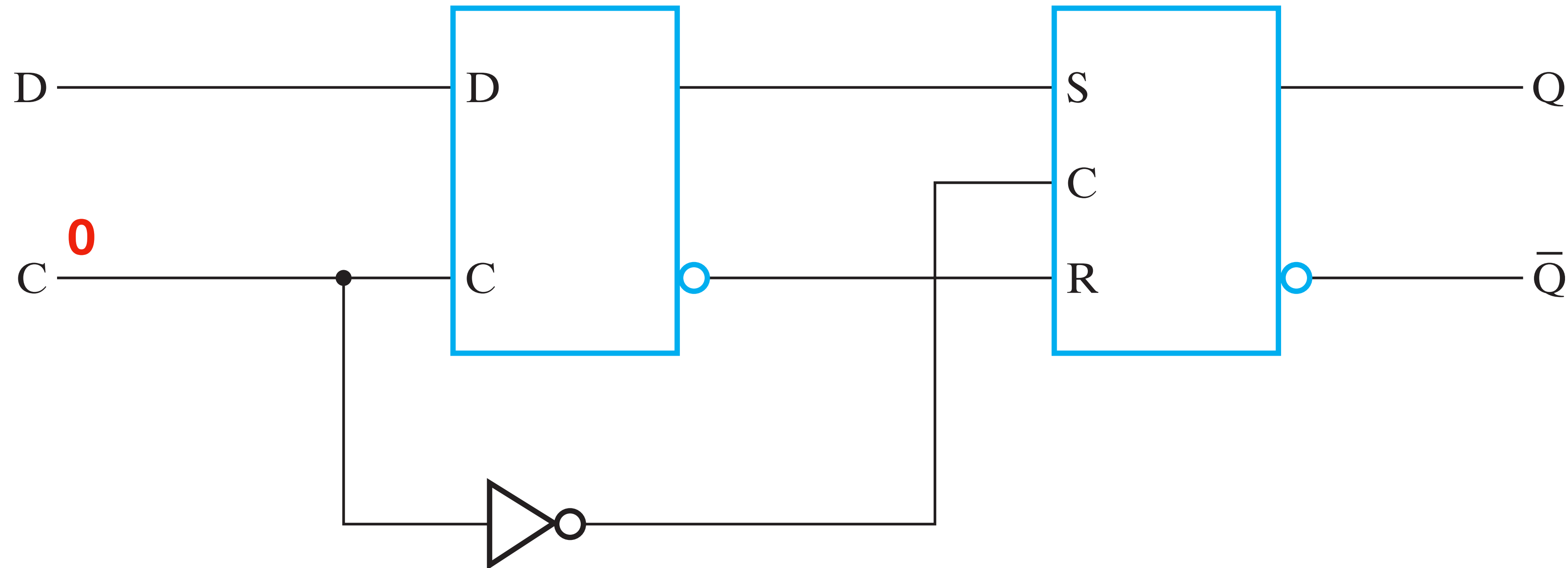
- Constructed using *SR* latches, left Master, right Slave
- Output state changes require $C = 0 \rightarrow C = 1 \rightarrow C = 0$ (Positive Pulse)

D Flip-Flop



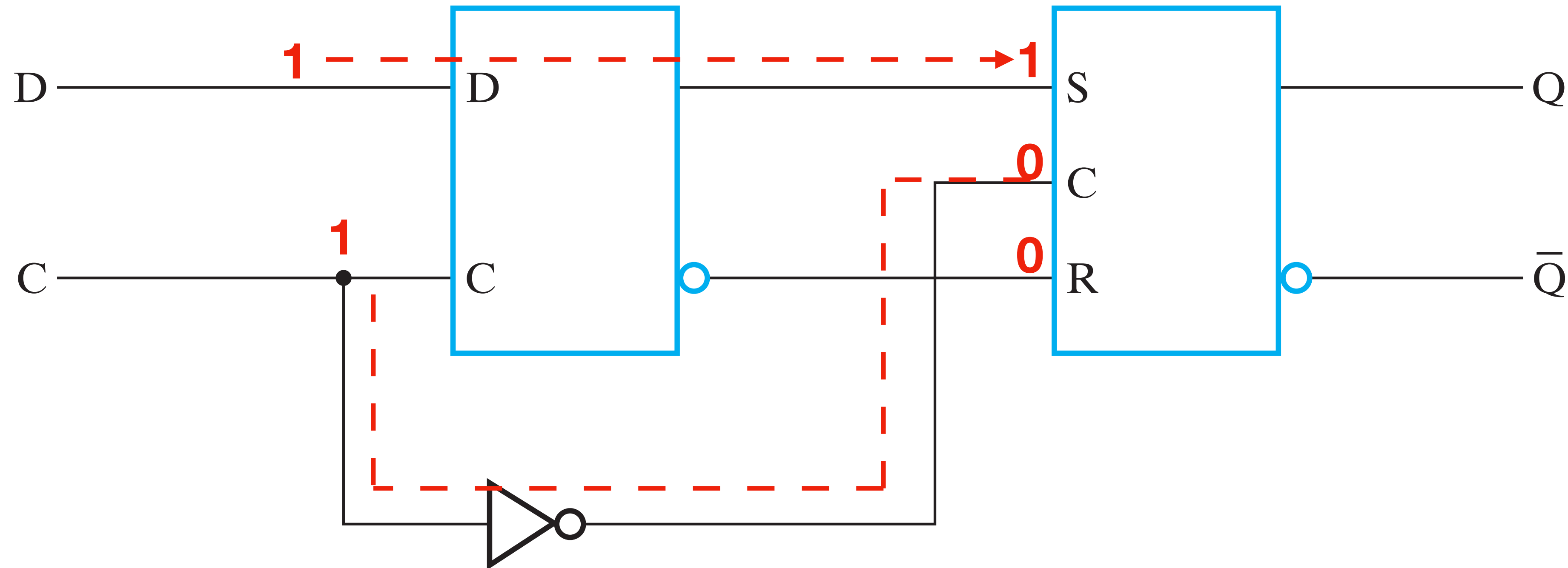
- Replaces *SR* master in *SR* Master-Slave with *D* master Latch
- **Negative Edge Triggered *D*** (Flip-Flop): $C = 1 \rightarrow C = 0$

D Flip-Flop



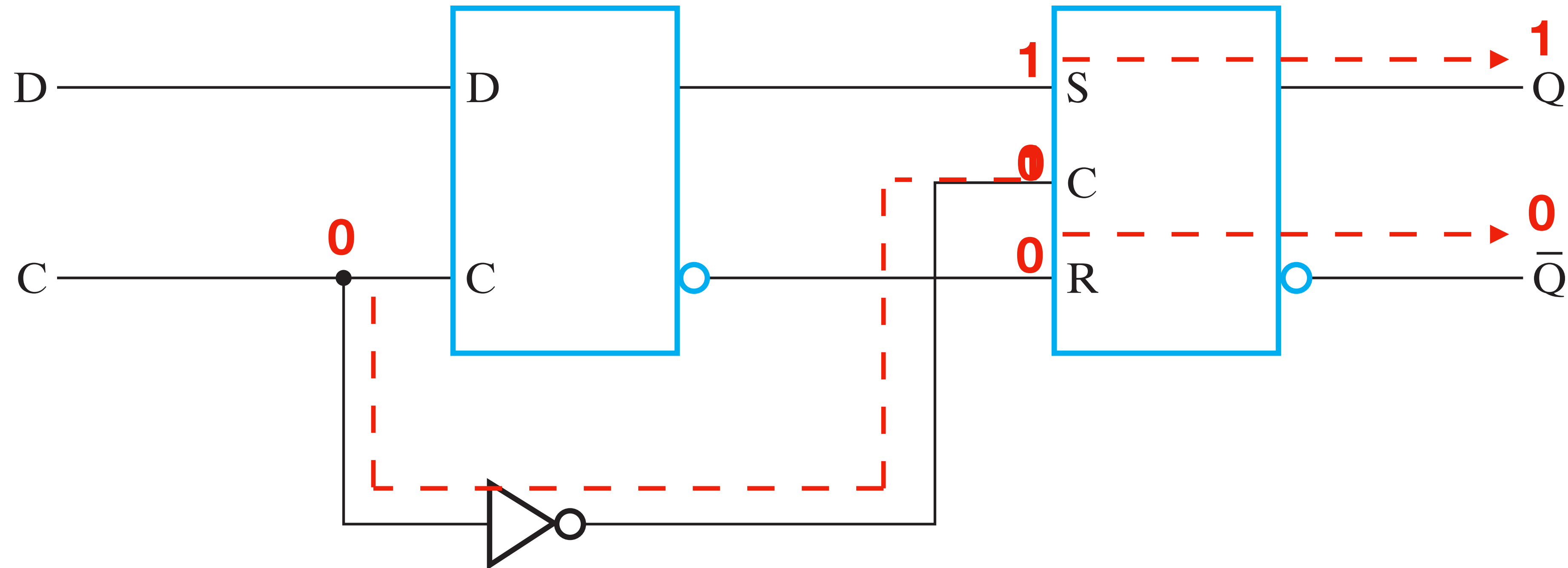
- Replaces *SR* master in *SR* Master-Slave with *D* master Latch
- **Negative Edge Triggered *D*** (Flip-Flop): $C = 1 \rightarrow C = 0$

D Flip-Flop



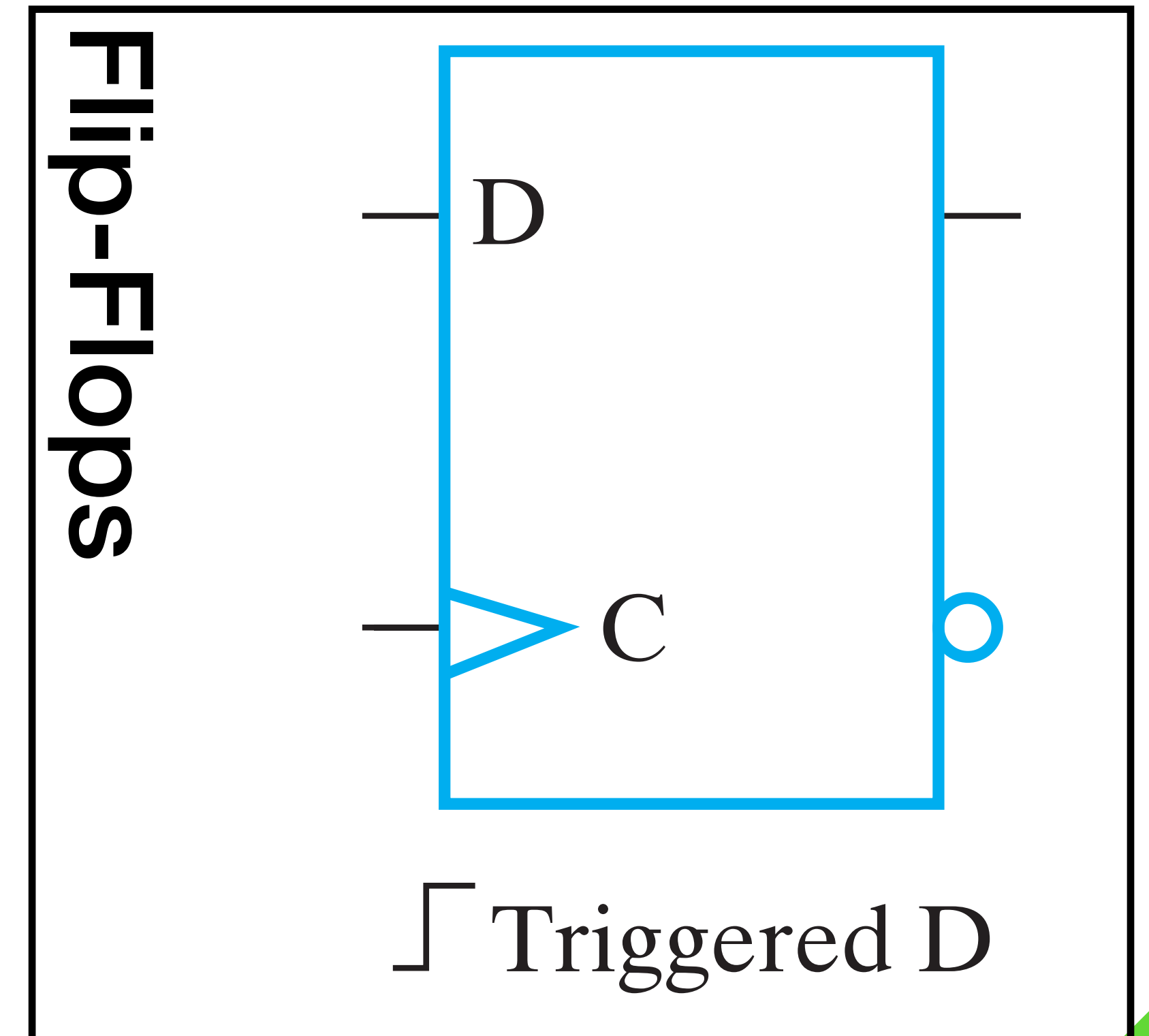
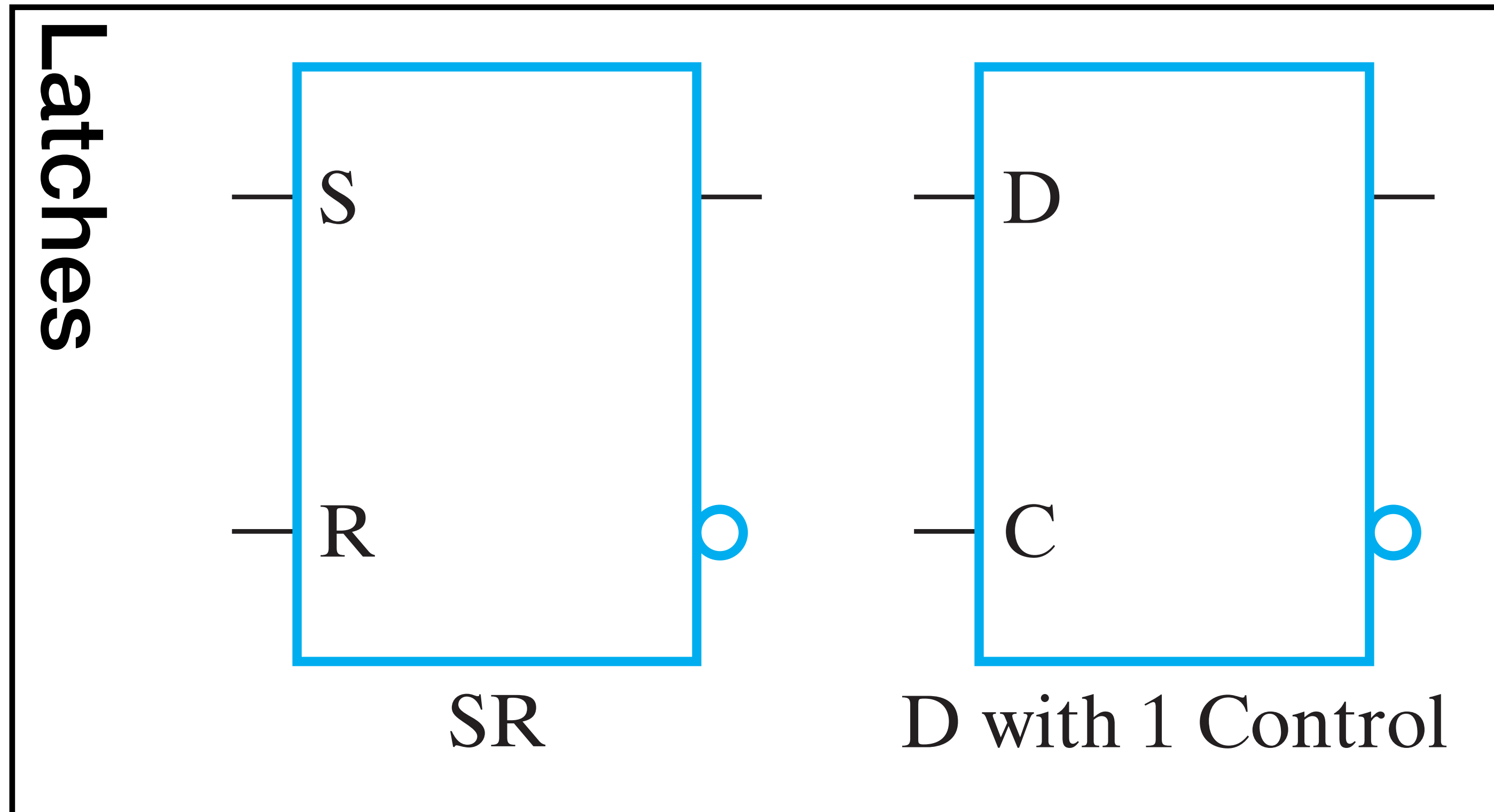
- Replaces *SR* master in *SR* Master-Slave with *D* master Latch
- **Negative Edge Triggered *D*** (Flip-Flop): $C = 1 \rightarrow C = 0$

D Flip-Flop



- Replaces *SR* master in *SR* Master-Slave with *D* master Latch
- **Negative Edge Triggered *D*** (Flip-Flop): $C = 1 \rightarrow C = 0$

Summary



Sequential Circuit Analysis

State Table; State Diagram

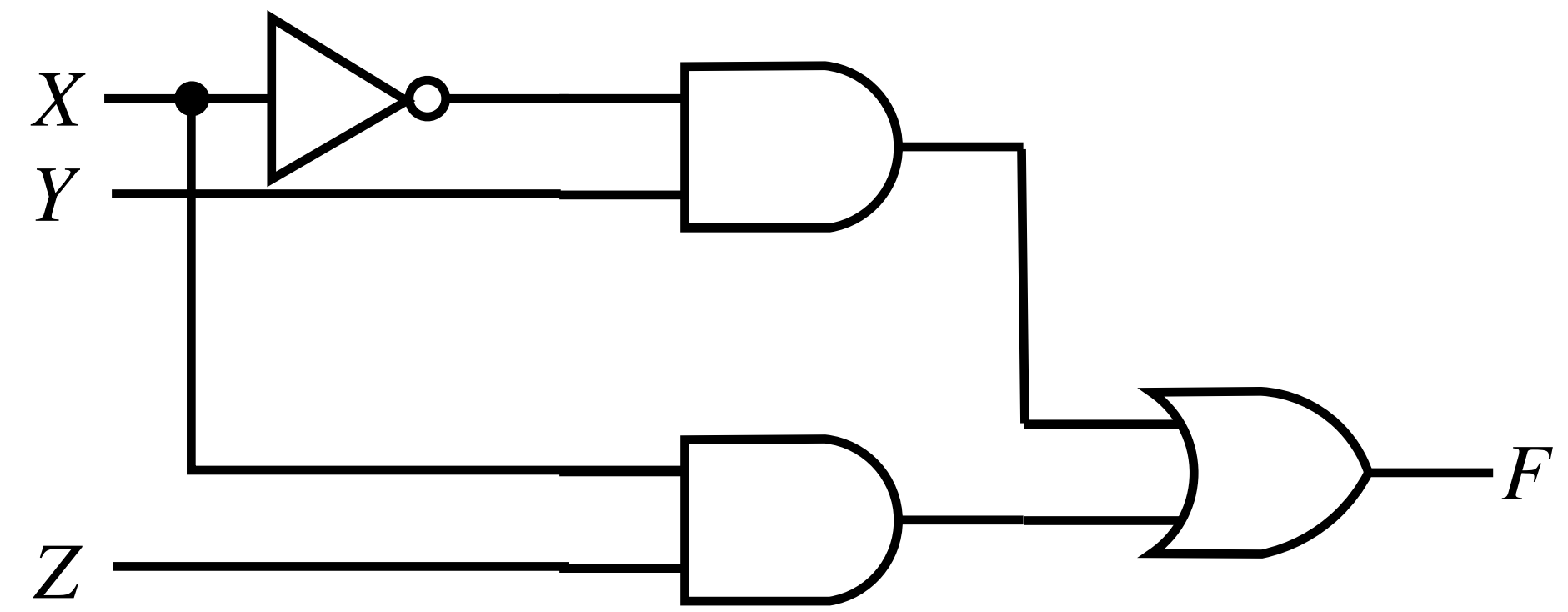
State Table

- Similar to truth table
 - Input, and Current states
 - Output and Next states

State Table

Truth Table

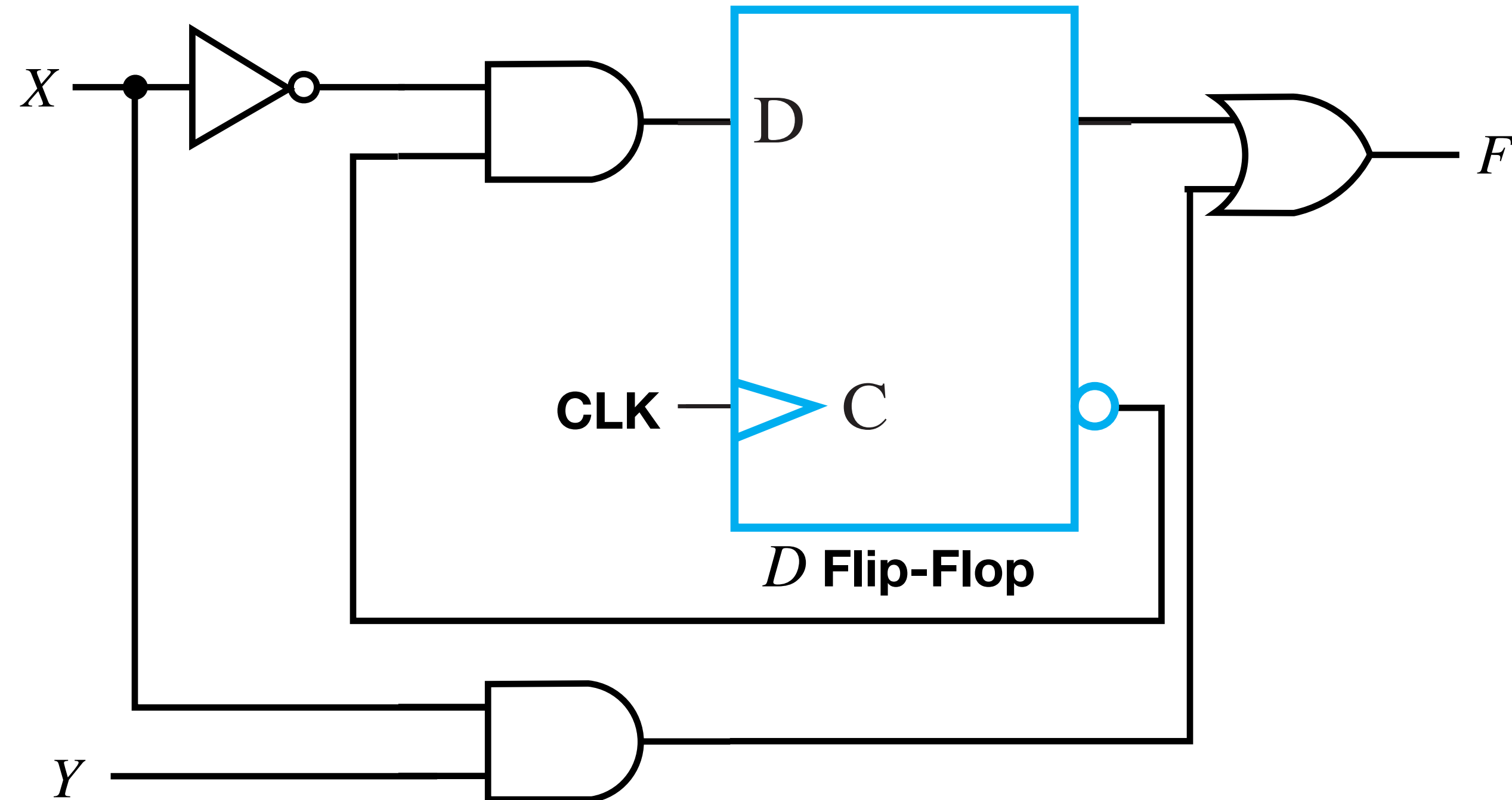
X	Y	Z	F
0	0	0	0
0	1	0	1
1	0	0	0
1	1	0	0
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	1



State Table

State Table

Present State	X	Y	Next State	F
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

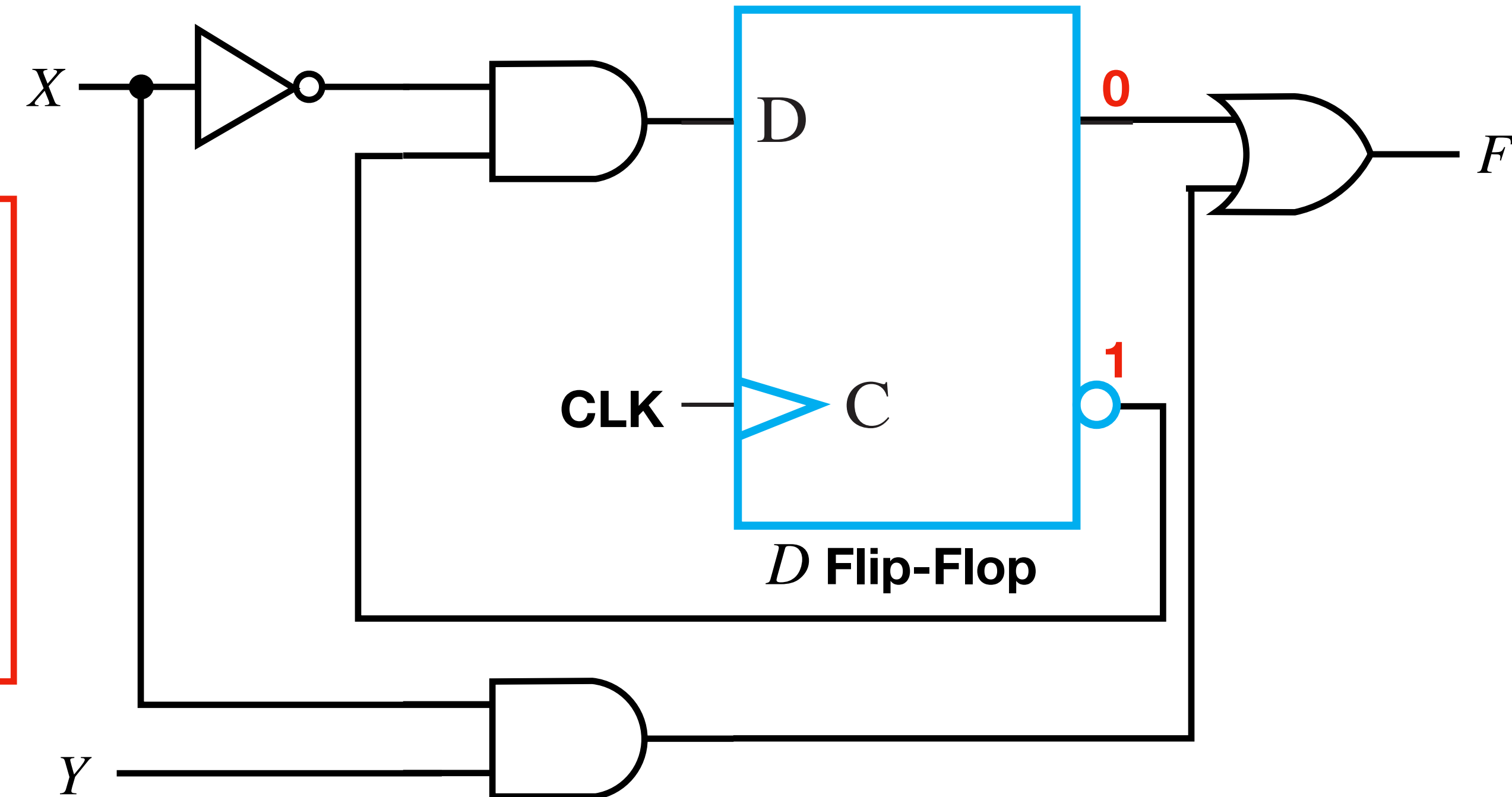


- When filling the state table, consider CLK = 0
- When CLK = 1, next state becomes current state

State Table

State Table

Present State	X	Y	Next State	F
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

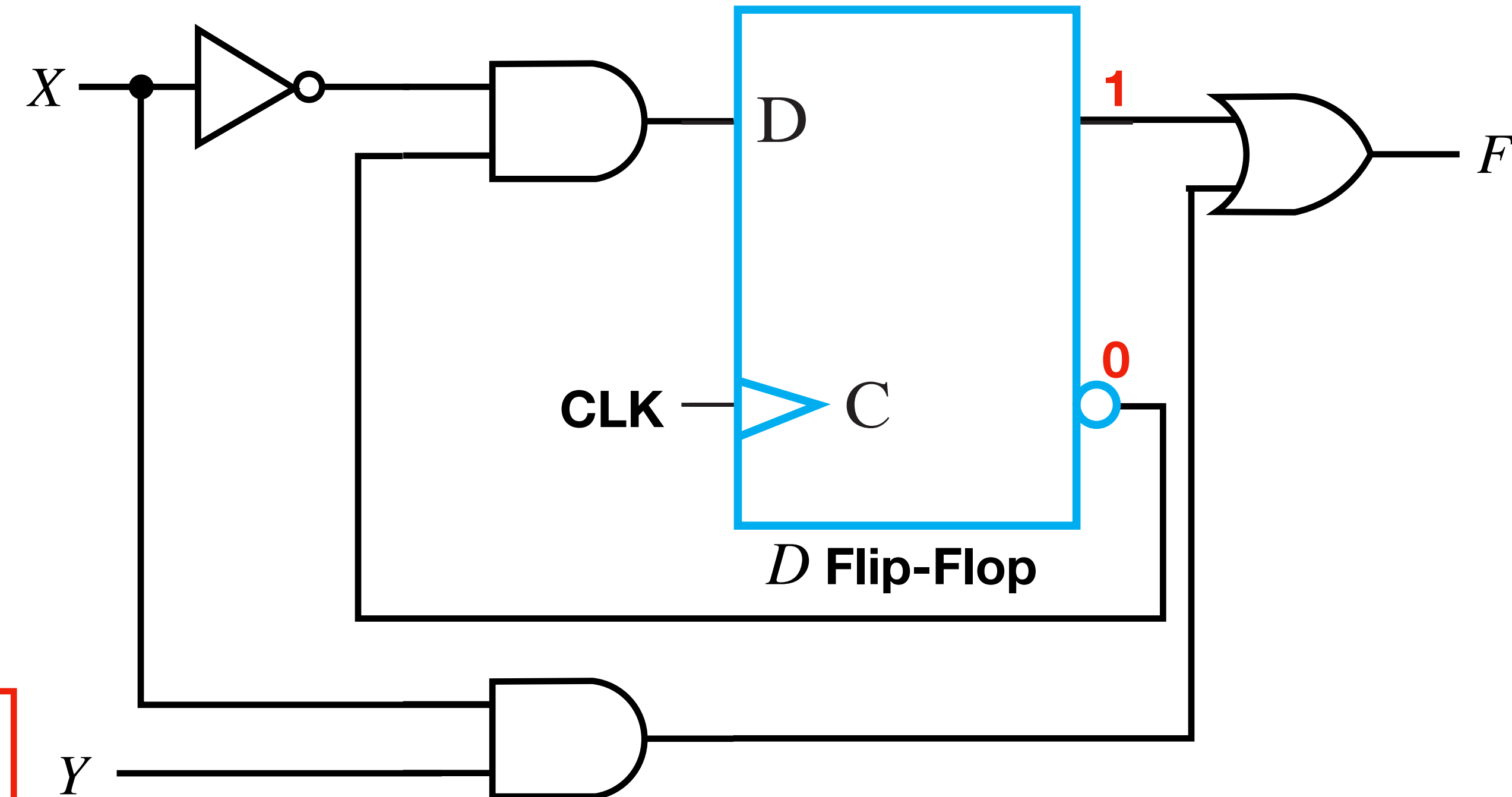


- When filling the state table, consider CLK = 0
- When CLK = 1, next state becomes current state

State Table

State Table

Present State	X	Y	Next State	F
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

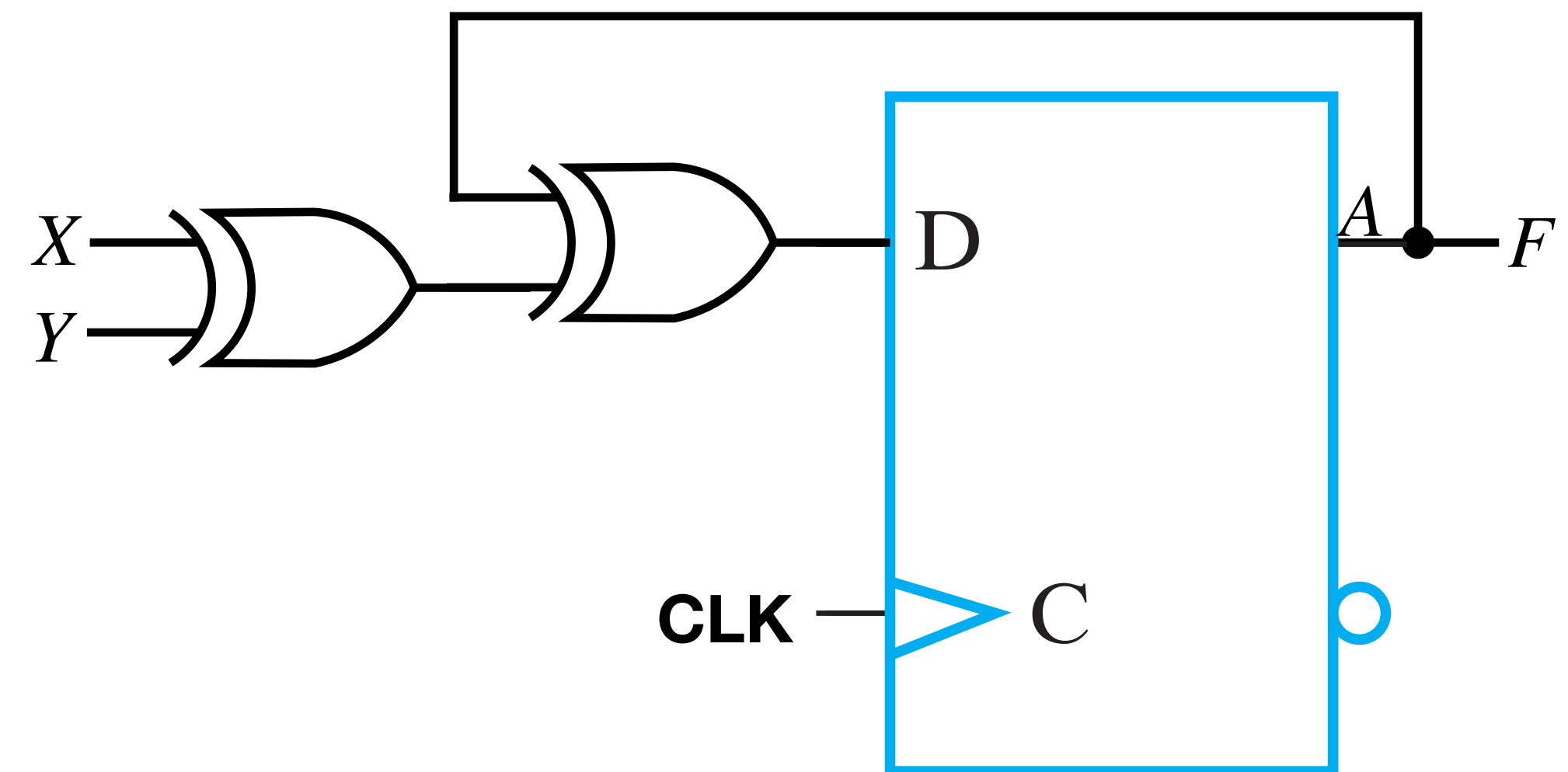


- When filling the state table, consider CLK = 0
- When CLK = 1, next state becomes current state

State Table

State Table

Present State	X	Y	Next State	F
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



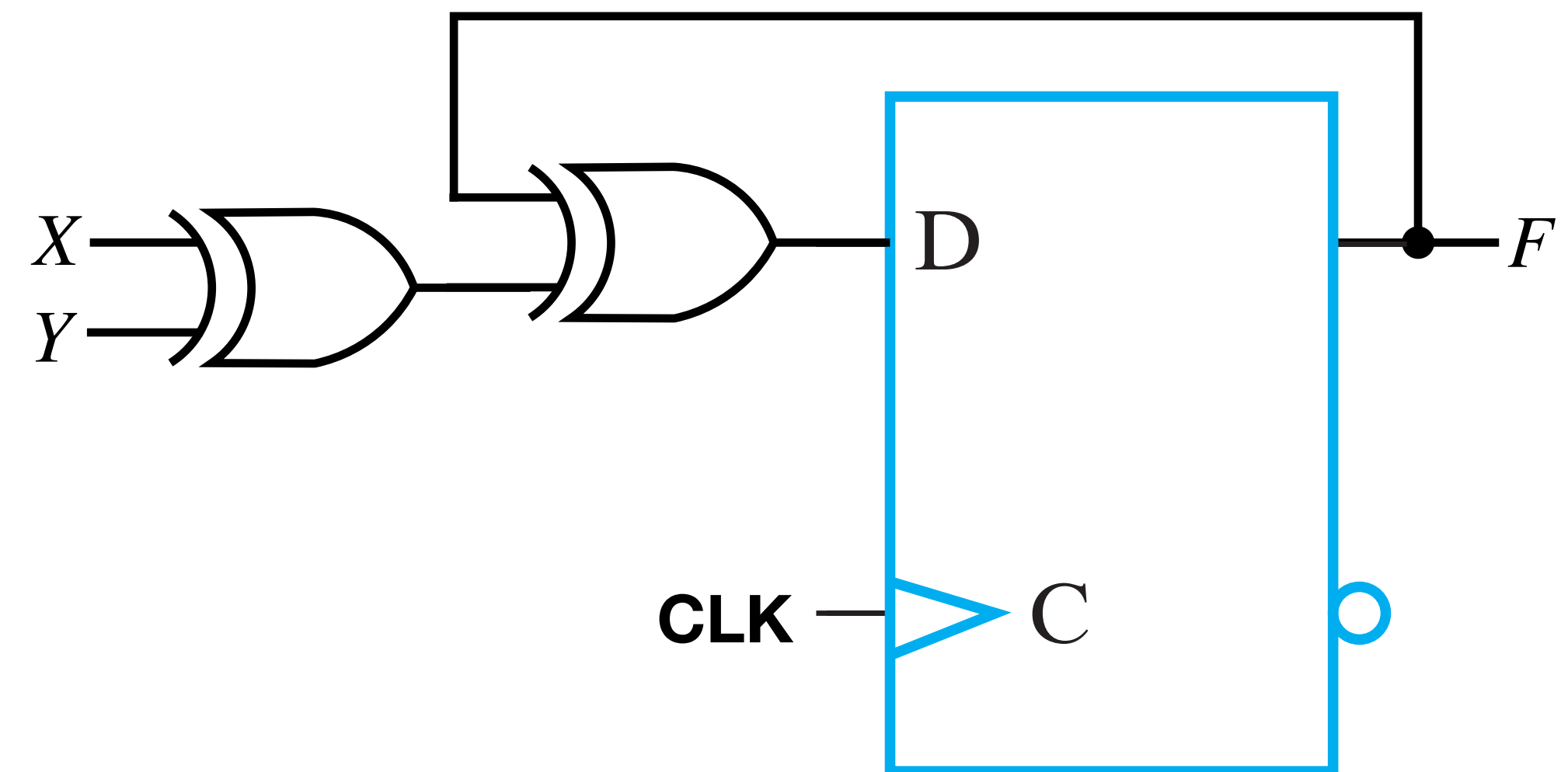
$$D_A = X \oplus Y \oplus A$$

- When filling the state table, consider CLK = 0
- When CLK = 1, next state becomes current state

State Table

State Table

Present State	X	Y	Next State	F
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0		
1	0	1		
1	1	0		
1	1	1		

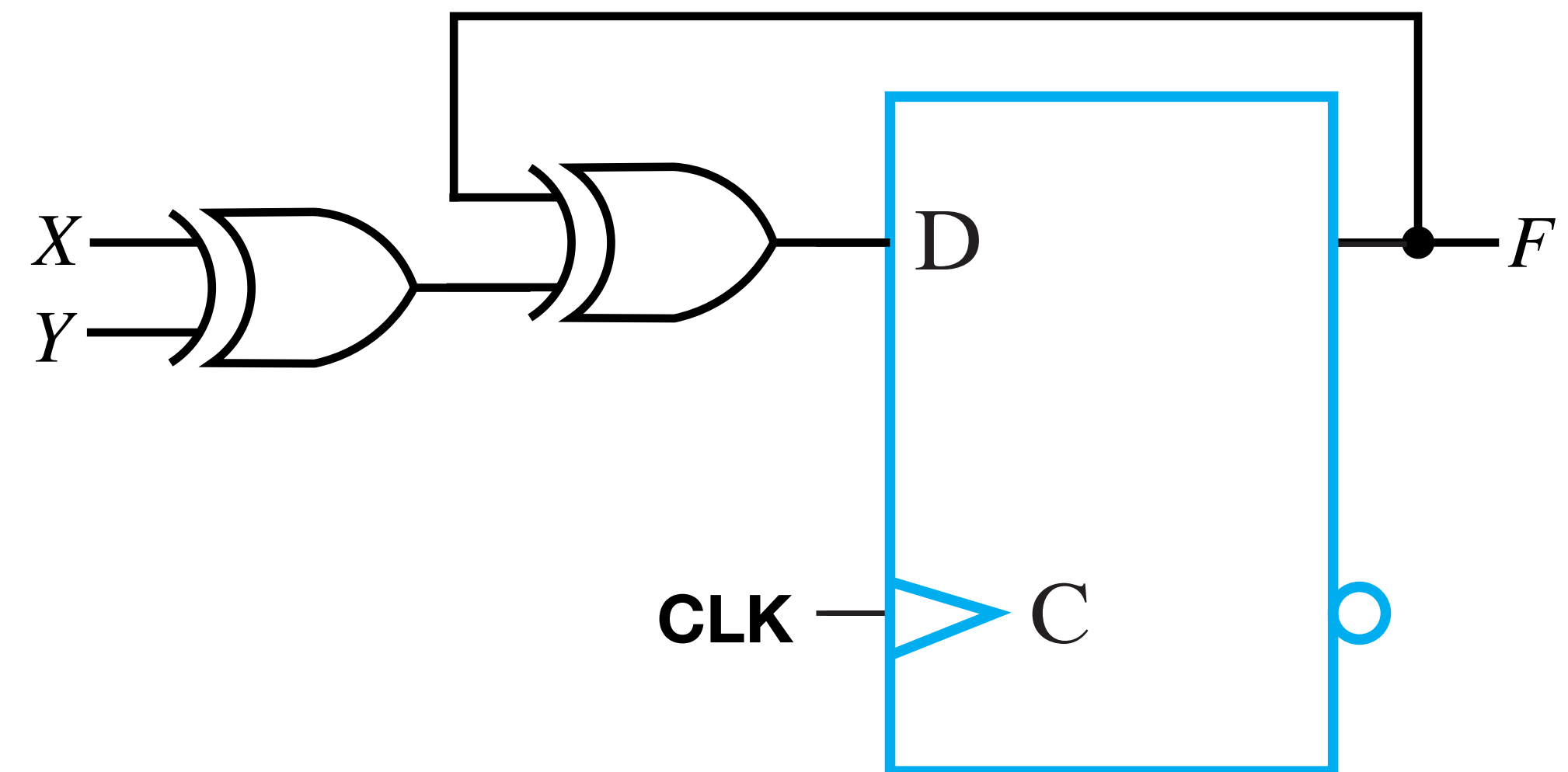


- When filling the state table, consider CLK = 0
- When CLK = 1, next state becomes current state

State Table

State Table

Present State	X	Y	Next State	F
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0		
1	0	1		
1	1	0		
1	1	1		

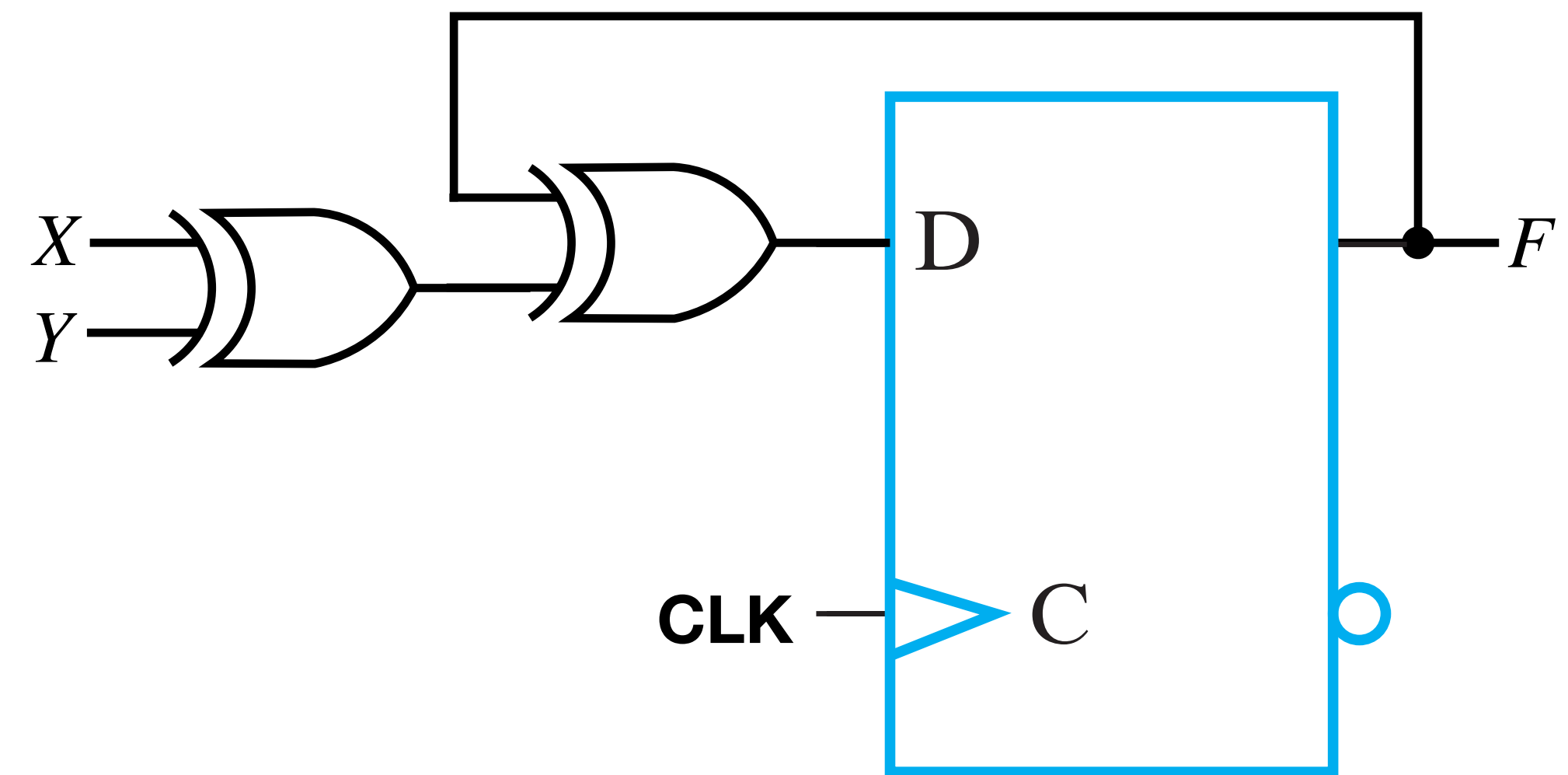


- When filling the state table, consider CLK = 0
- When CLK = 1, next state becomes current state

State Table

State Table

Present State	X	Y	Next State	F
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



- When filling the state table, consider CLK = 0
- When CLK = 1, next state becomes current state

State Table

State Table

Present State		X	Y	Next State		Z
A	B			A	B	
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			

...

$$D_A = \bar{X}A + XY$$

$$D_B = \bar{X}B + XA$$

$$Z = XB$$

- Draw circuit diagram
- Fill the 8 rows

Exercise

State Table

- What happens when there are multiple flip-flops?
- Does it work with Latches?

In Class Exercise 1

- A circuit with one D flip-flop: $D_A = A \oplus X$
- Draw the circuit diagram
- Do the state table

State Table

Present State	X	Next State
0	0	
0	1	
1	1	
1	0	

In Class Exercise 2

- A circuit with 2 D flip-flops: $D_A = A \oplus B$, $D_B = \bar{B} \cdot X$, $F = \bar{A}B$

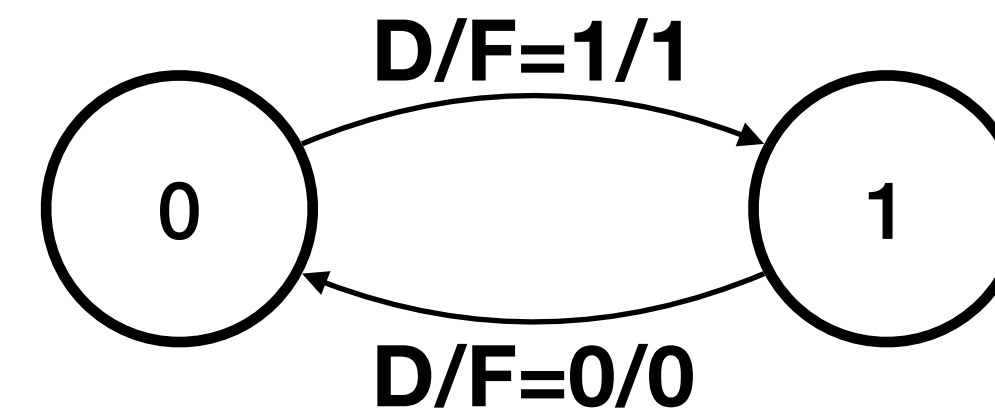
State Table

- Do the state table

Present State		X	Next State		F
A	B		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			

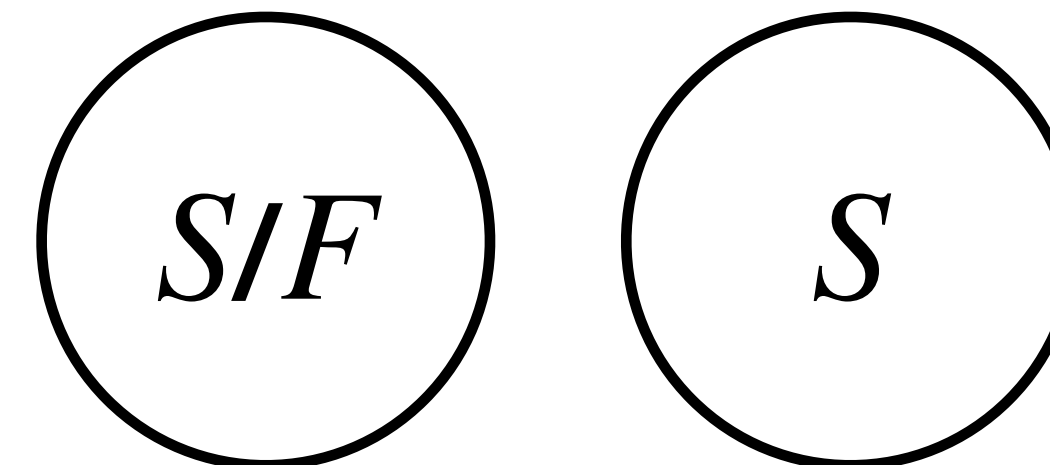
State Diagram

- Similar to state table
- Models state transitions
- A state is represented in a bubble



- Directed links between bubbles: the input used to perform transition
Source: **present** state, Target: **next** state (next **CLK**); (optional output F)

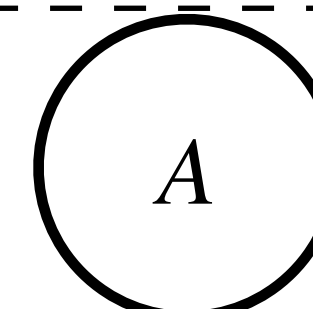
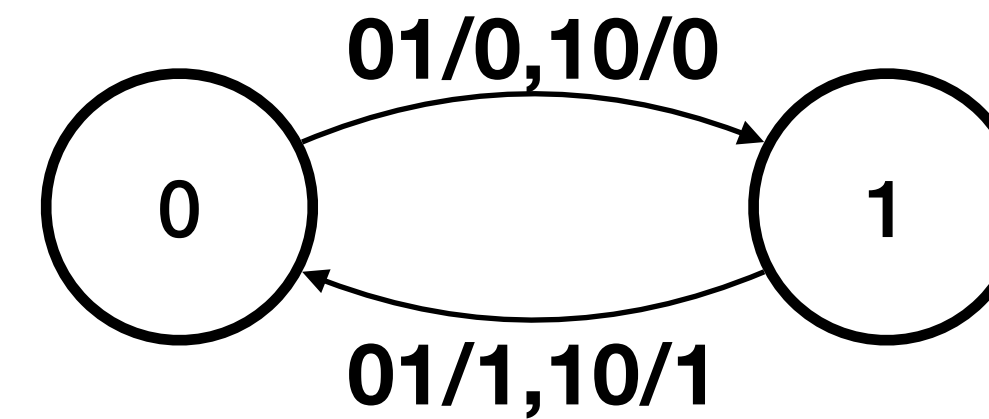
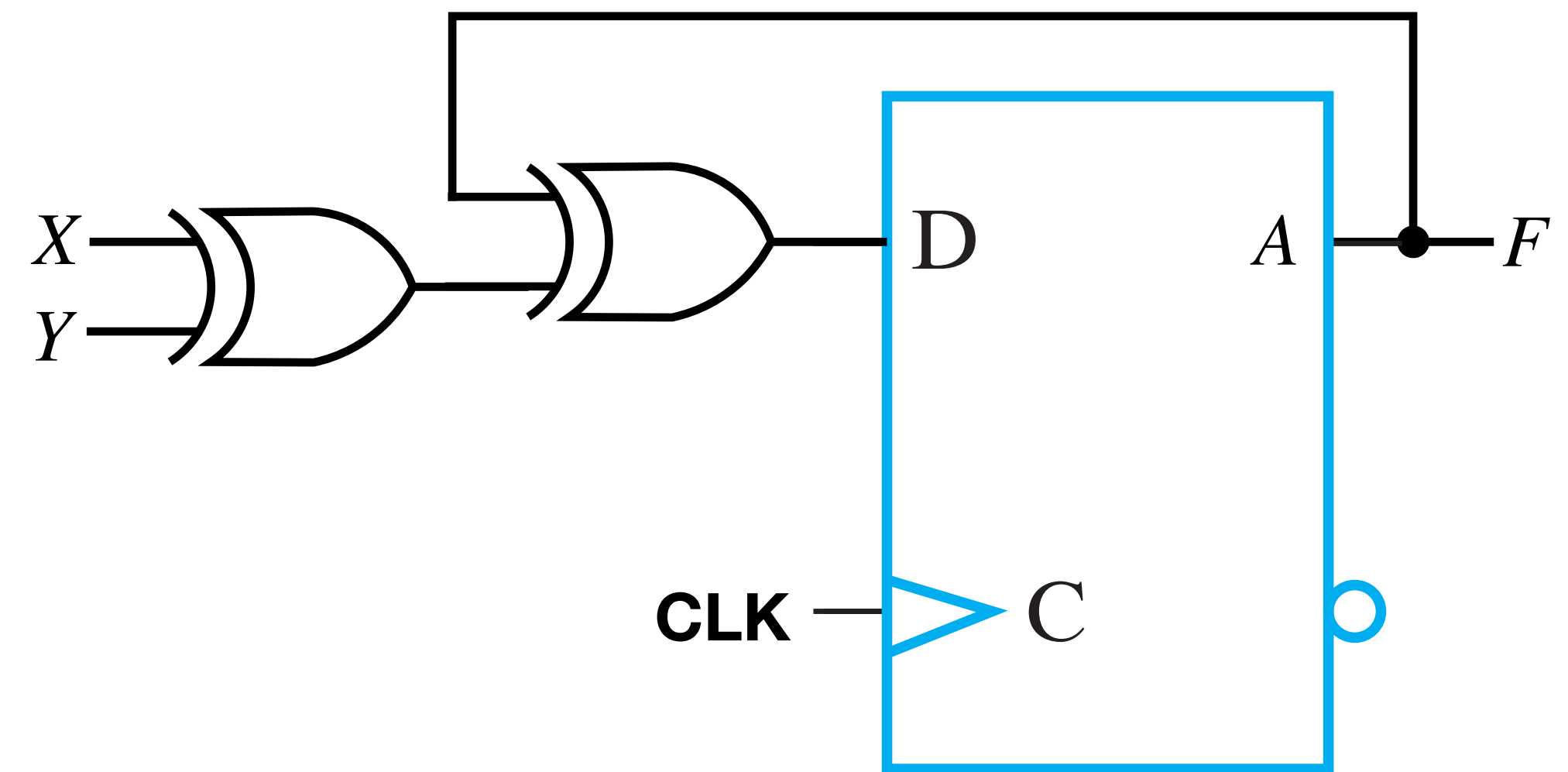
- State bubble with state as S (optional output F)



State Diagram

State Table

Present State	X	Y	Next State	F
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



A single state bubble

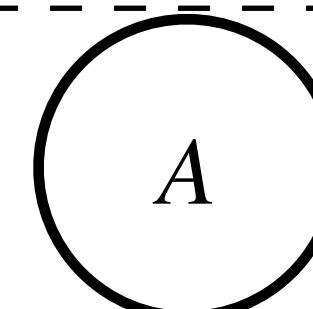
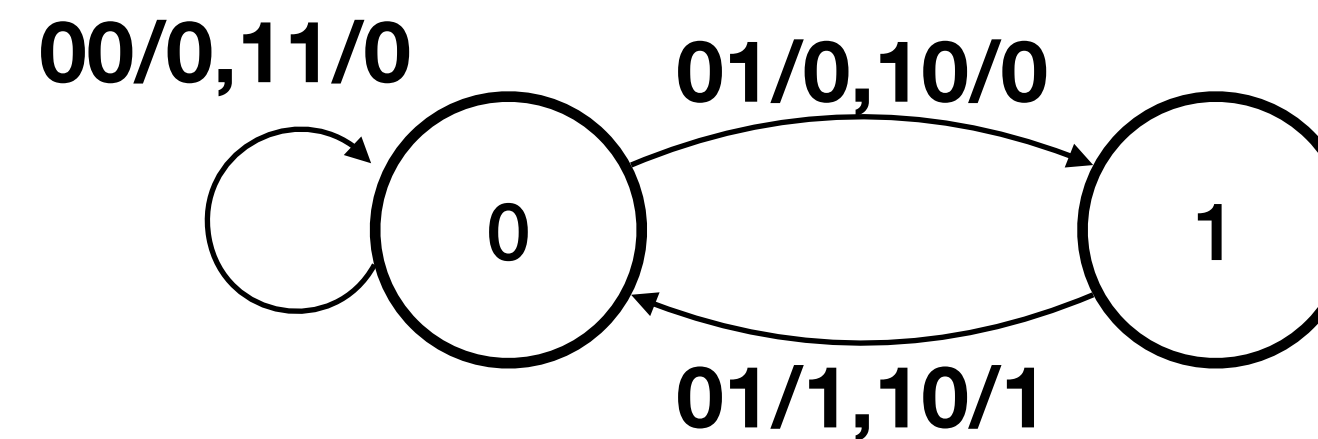
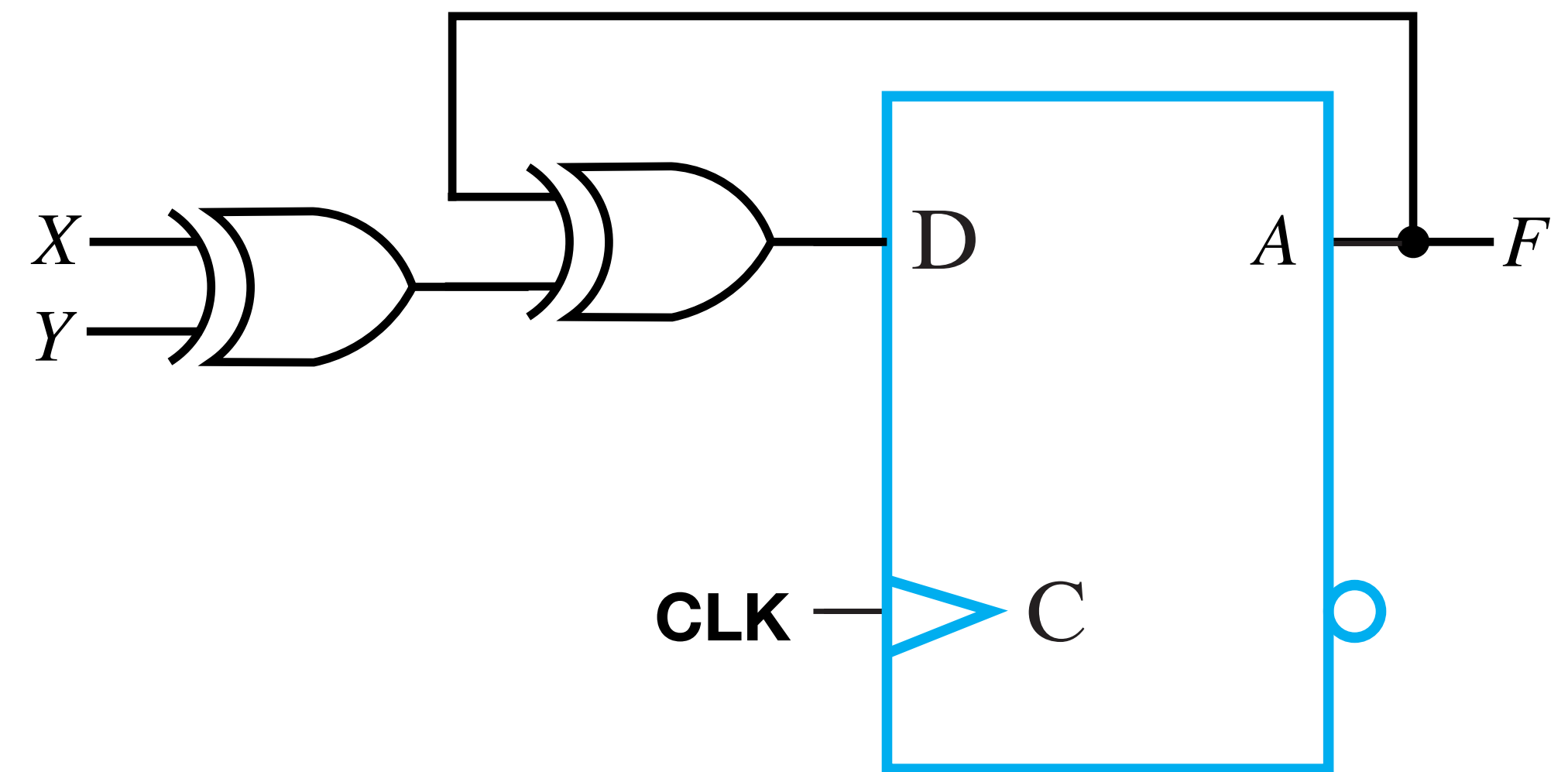


A single link

State Diagram

State Table

Present State	X	Y	Next State	F
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



A single state bubble

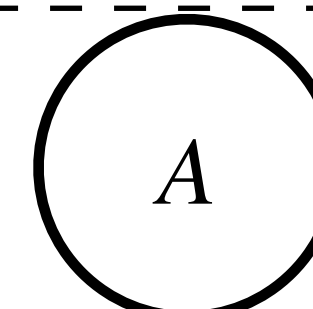
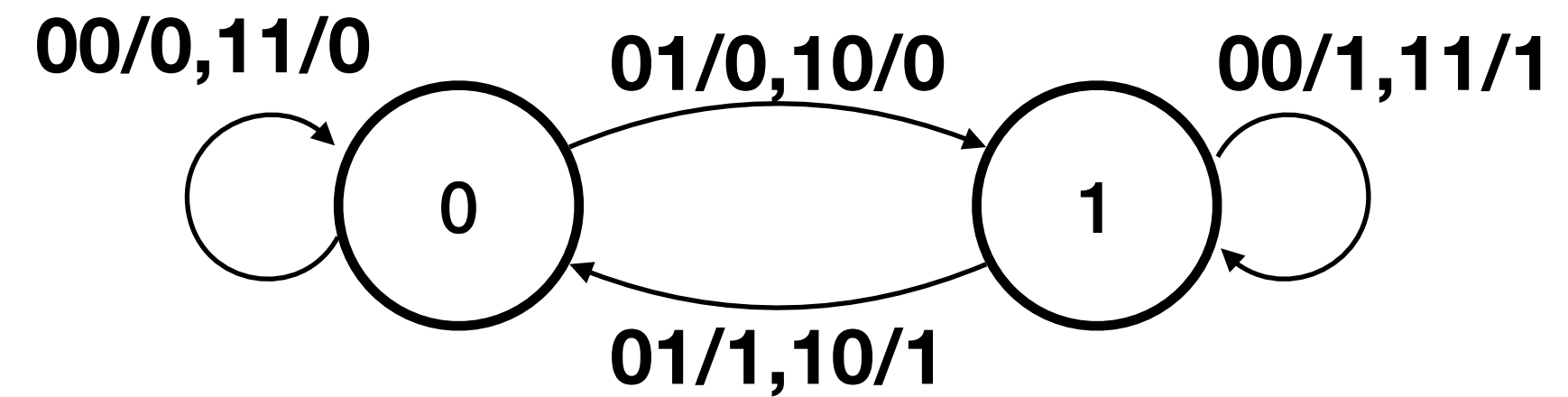
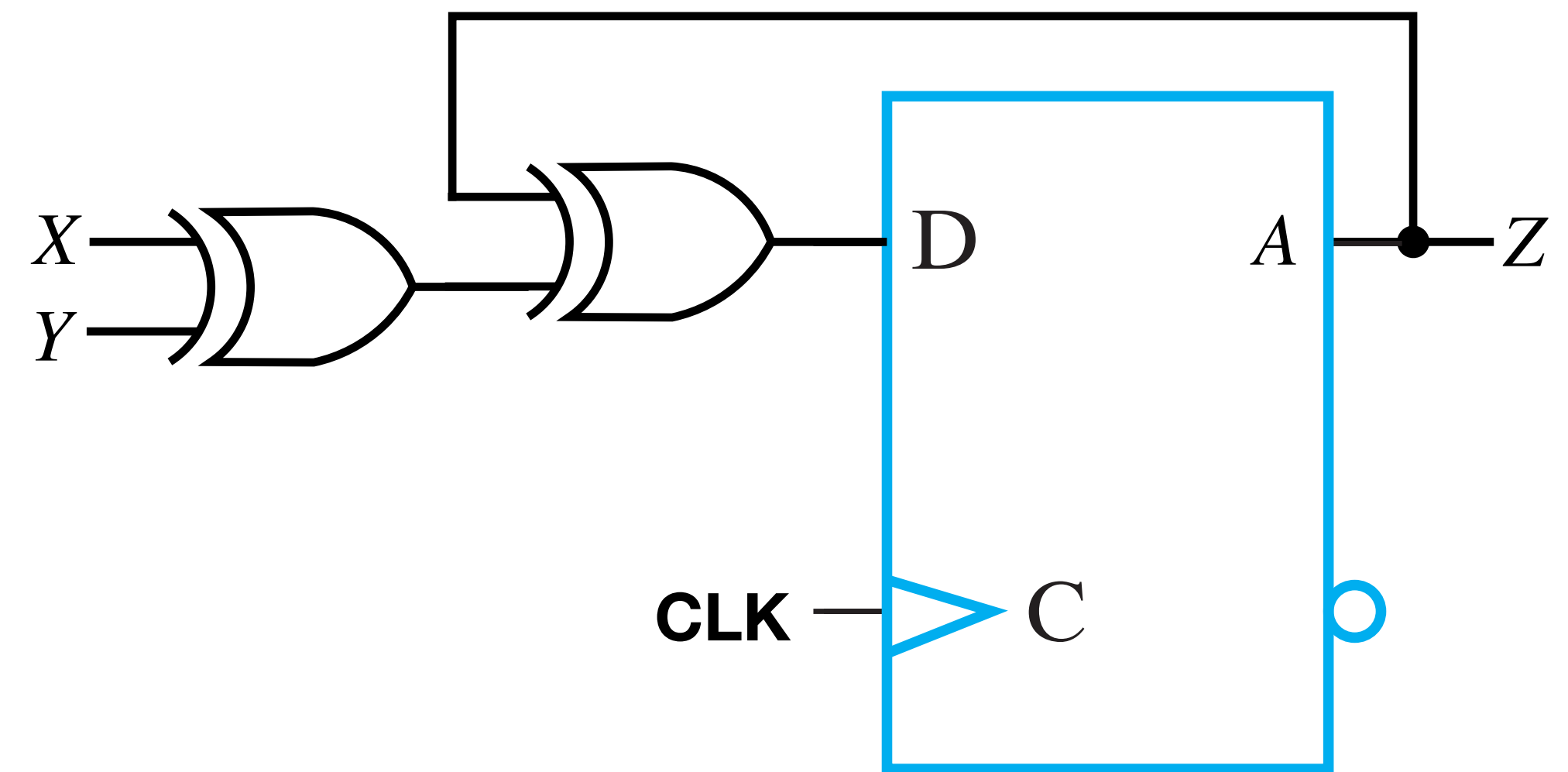


A single link

State Diagram

State Table

Present State	X	Y	Next State	Z
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



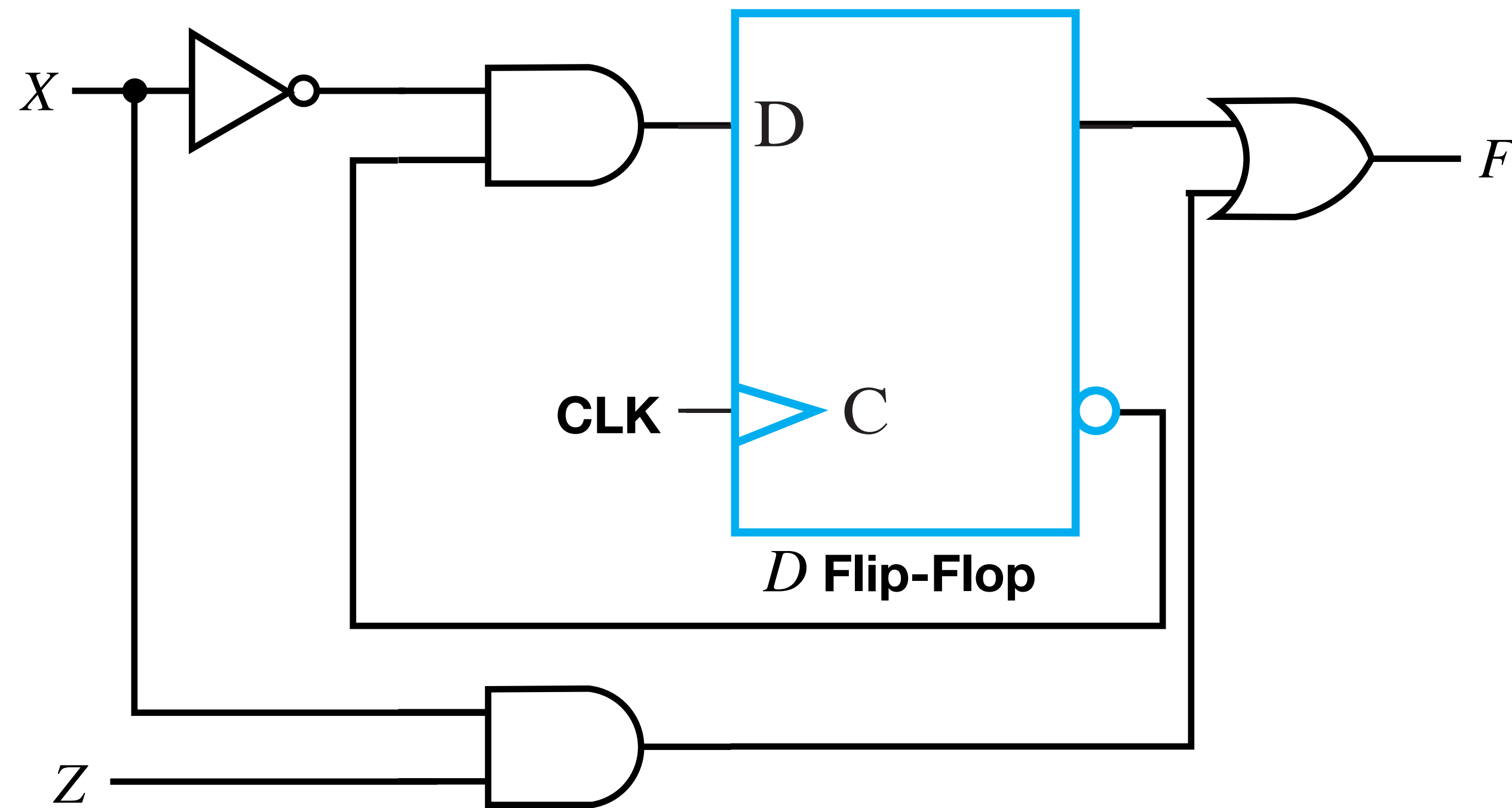
A single state bubble



A single link

Concept

State Diagram



State Diagram

- Draw the state diagram for: $D_A = \bar{X}A + XY$, $D_B = \bar{X}B + XA$, $Z = XB$

In Class Exercise 1

- A circuit with one D flip-flop: $D_A = A \oplus X$
- Draw the state diagram

State Table

Present State	X	Next State
0	0	0
0	1	1
1	0	1
1	1	0

In Class Exercise 2

- A circuit with 2 D flip-flops: $D_A = A \oplus B$, $D_B = \bar{B} \cdot X$, $F = \bar{A}B$

State Table

- Do the state diagram

Present State		X	Next State		F
A	B		A	B	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	1
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	0	0	0

LogicWorks Exercise

- Implement D flip flop using D latch and SR latch
Save it as a component in your library
- Implement circuit $D_S = X \oplus Y \oplus S$, where D_S is a D flip flop
- Implement $D_A = \bar{X}A + XY$, $D_B = \bar{X}B + XA$, $Z = XB$
- Draw the state table and diagram, and verify your table with LogicWorks

