

Lecture 10:

Sequential Networks: Latches

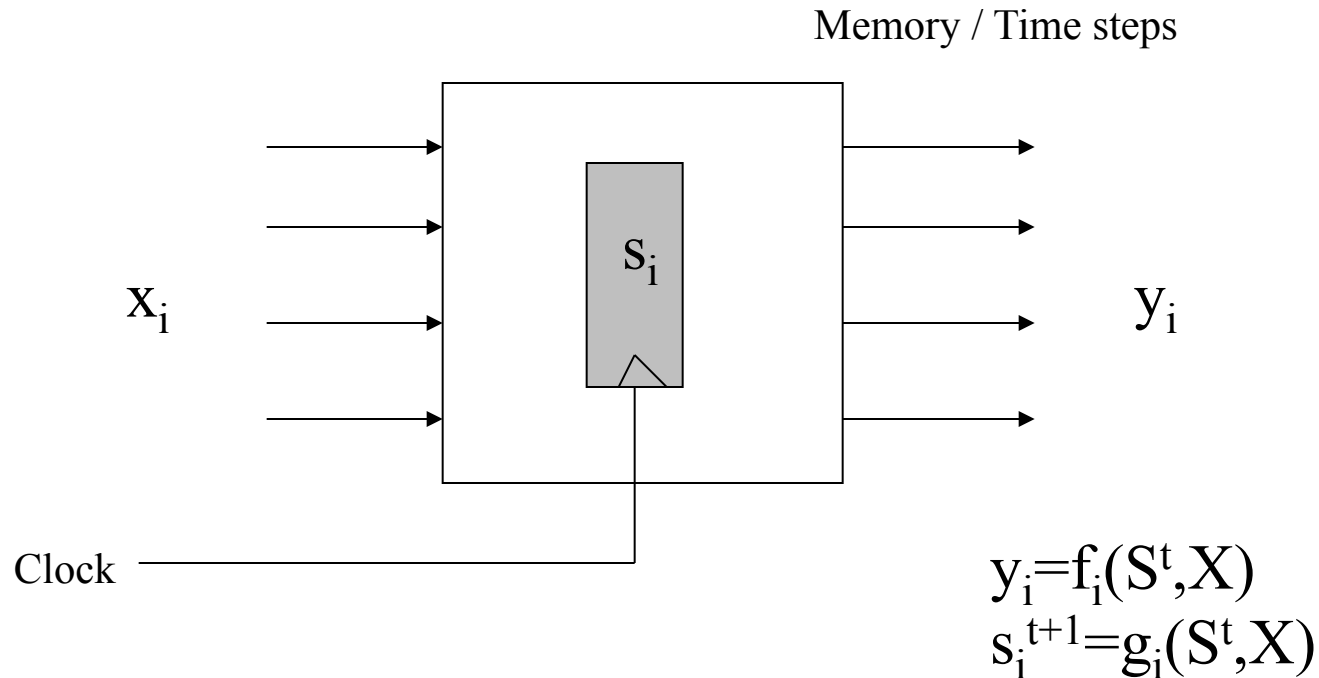
CSE 140: Components and Design Techniques for
Digital Systems

Diba Mirza
Dept. of Computer Science and Engineering
University of California, San Diego

What is a sequential circuit?

“A circuit whose output depends on current inputs and past outputs”

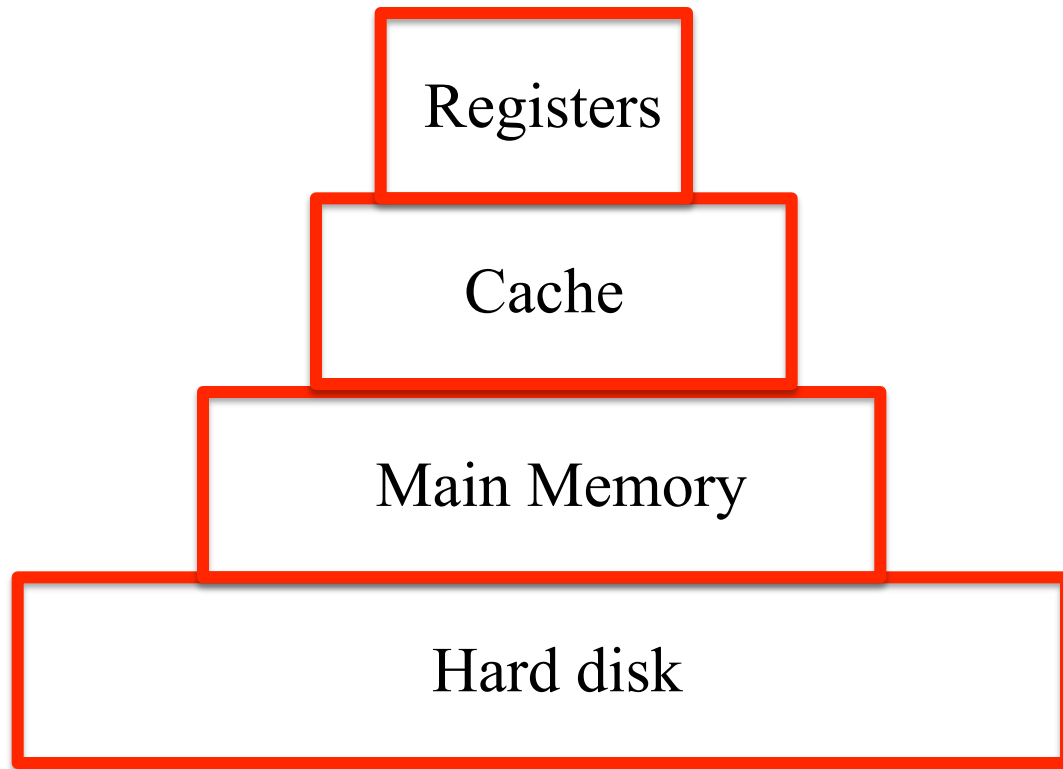
“A circuit with **memory**”



Why do we need circuits with ‘memory’?

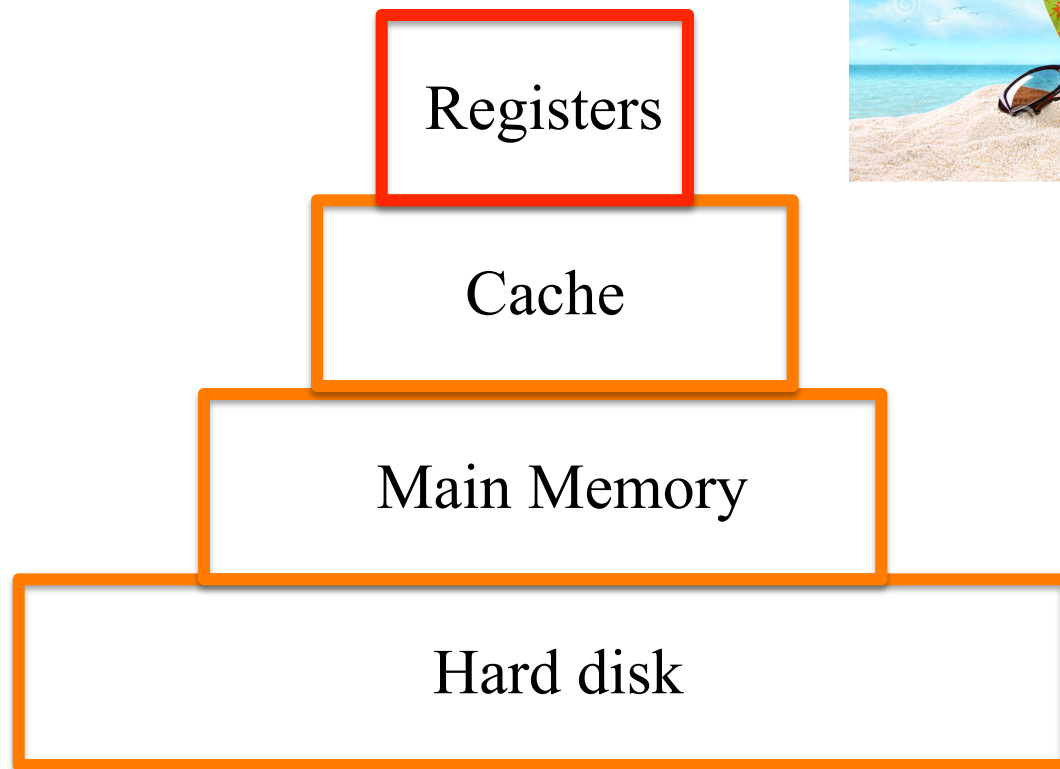
- Circuits with memory can be used to store data
- Complex systems often consist of circuits that perform a sequence of tasks

Memory Hierarchy



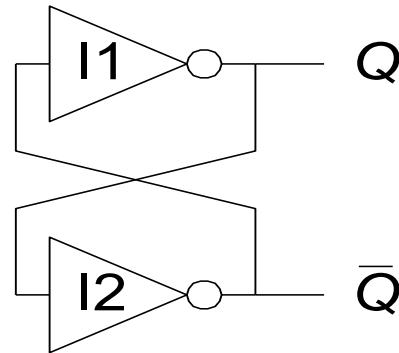
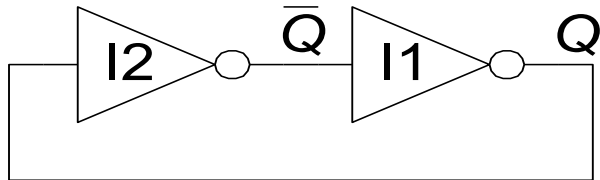
Memory Hierarchy

- What are registers made of?

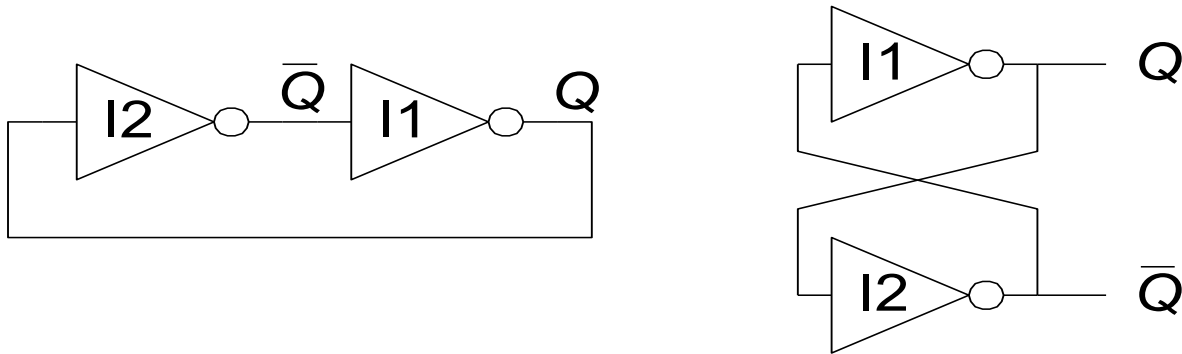


To the flip-flop!

- The most basic memory element:
The capacitive load



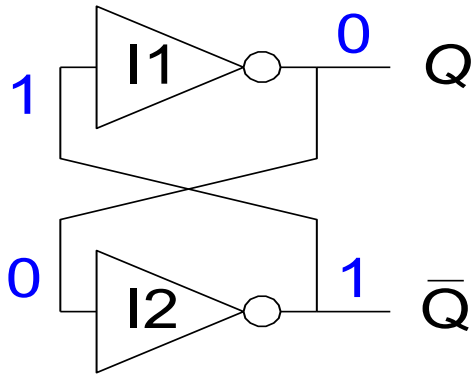
Fundamental Memory Mechanism



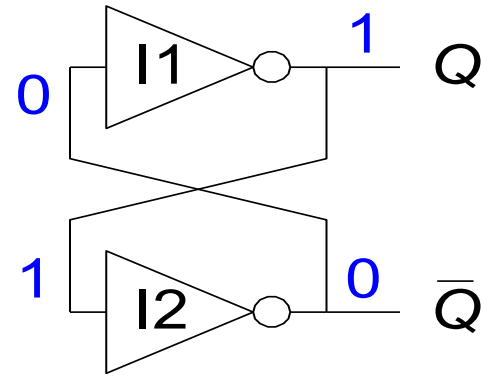
How is this circuit different from a combinational circuit?

- A. Two outputs: Q , Q'
- B. Feedback loop
- C. No inputs

Does this circuit really remember?



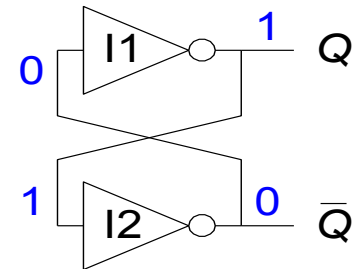
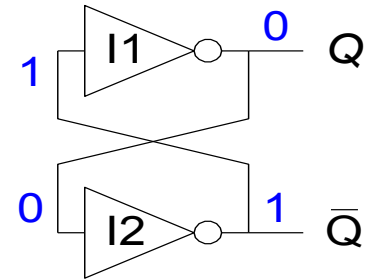
- $Q = 0$: then
 $Q' = 1$ and $Q = 0$



- $Q = 1$: then

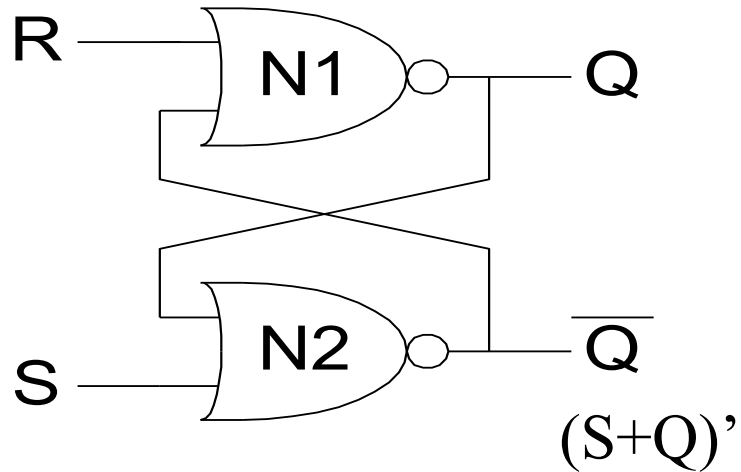
Bi-stable circuit

- Bistable circuit stores 1 bit of state in the state variable, Q (or Q')
- But there are **no inputs to control the state**



SR (Set/Reset) Latch

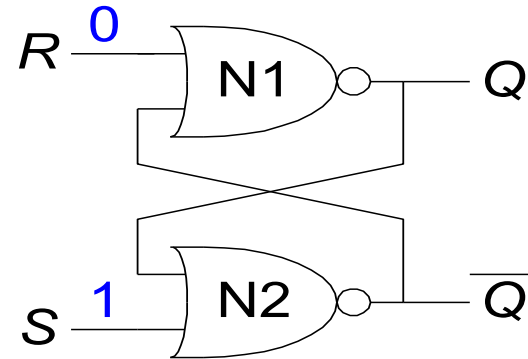
- SR Latch



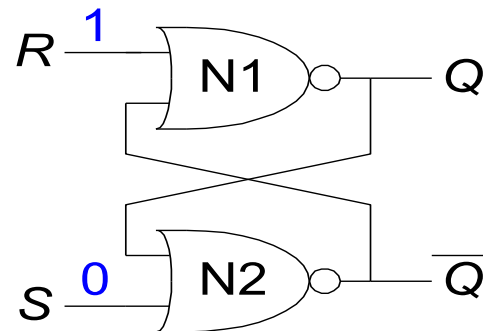
- Consider the four possible cases:
 - $S = 1, R = 0$: set output to '1'
 - $S = 0, R = 1$: (reset) output to '0'
 - $S = 0, R = 0$: store – output should be unchanged
 - $S = 1, R = 1$: Trouble!

SR Latch Analysis

– $S = 1, R = 0$:

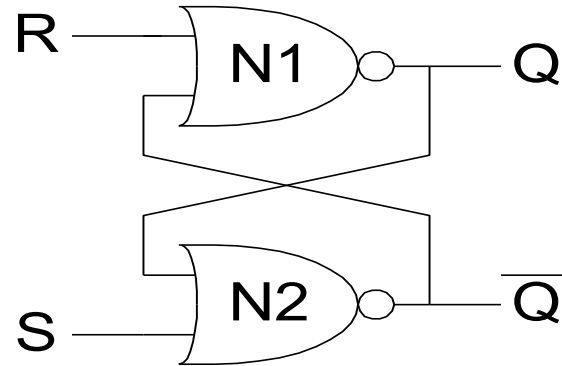


– $S = 0, R = 1$:



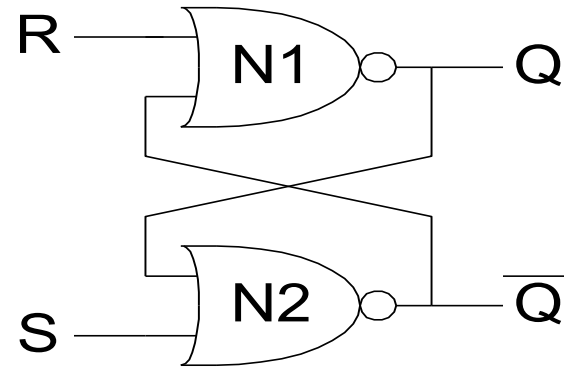
SR Latch Analysis

– $S = 0, R = 0$:



SR Latch Analysis

– $S = 0, R = 0$:

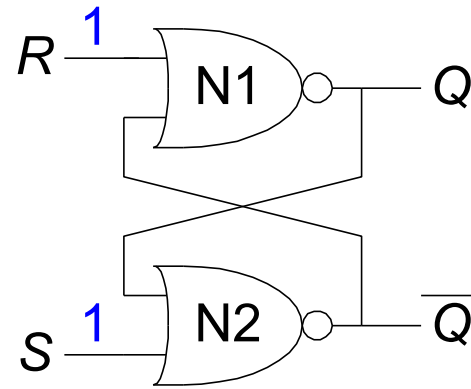


What happens if $Q_{\text{prev}}=0$ and $Q'_{\text{prev}}=0$?

- A. The output Q toggles
- B. The output Q remains 0 and Q' changes to 1
- C. The output Q becomes 1 and Q' remains 0

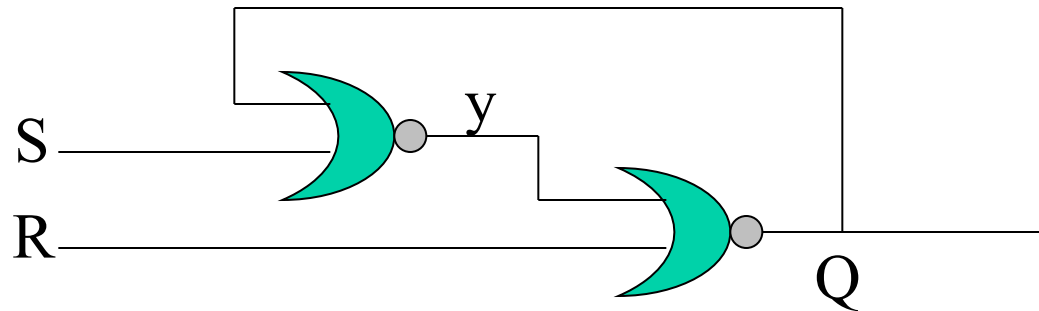
SR Latch Analysis

– $S = 1, R = 1$:



Flip-flop Components

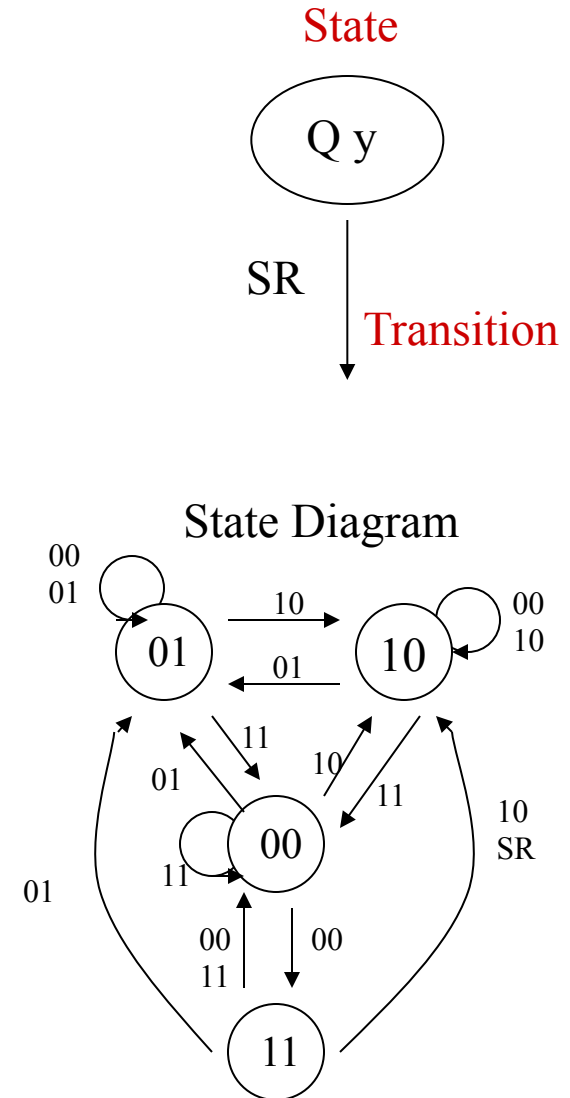
SR latch (Set-Reset)



Inputs: S, R

State: (Q, y)

Id	Q(t)	y(t)	S	R	Q(t ₁)	y(t ₁)	Q(t ₂)	y(t ₂)	Q(t ₃)	y(t ₃)
0	0	0	0	0	1	1	0	0	1	1
1	0	0	0	1	0	1	0	1	0	1
2	0	0	1	0	1	0	1	0	1	0
3	0	0	1	1	0	0	0	0	0	0
4	0	1	0	0	0	1	0	1	0	1
5	0	1	0	1	0	1	0	1	0	1
6	0	1	1	0	0	0	1	0	1	0
7	0	1	1	1	0	0	0	0	0	0
8	1	0	0	0	1	0	1	0	1	0
9	1	0	0	1	0	0	0	1	0	1
10	1	0	1	0	1	0	1	0	1	0
11	1	0	1	1	0	0	0	0	0	0
12	1	1	0	0	0	0	1	1	0	0
13	1	1	0	1	0	0	0	1	0	1
14	1	1	1	0	0	0	1	0	1	0
15	1	1	1	1	0	0	0	0	0	0



$SR=01, (Q,y) = (0,1)$

$SR=10, (Q,y) = (1,0)$

$SR=11, (Q,y) = (0,0)$

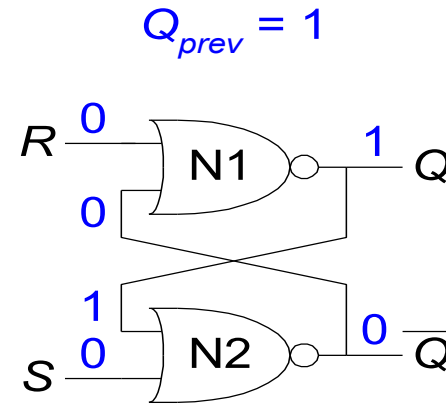
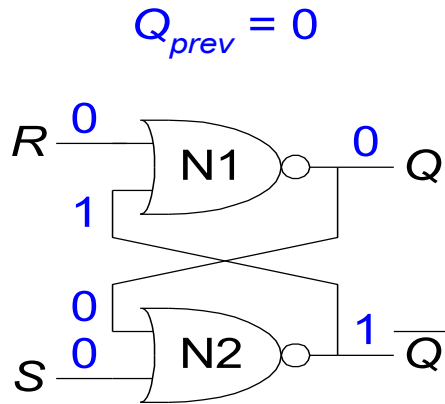
$SR = 00 \Rightarrow$ if $(Q,y) = (0,0)$ or $(1,1)$, the output keeps changing
 $\backslash$

Solutions: 1) $SR \neq (0,0)$, or

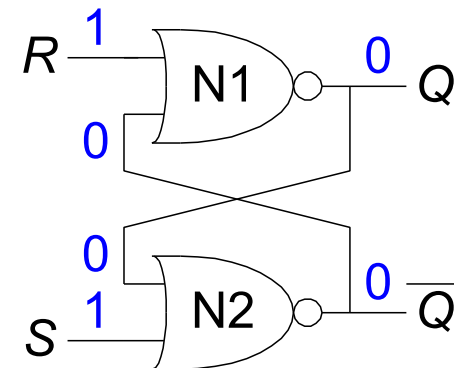
2) $SR = (1,1)$.

SR Latch Analysis

- $S = 0, R = 0$: then $Q = Q_{prev}$ and $\bar{Q} = \bar{Q}_{prev}$ (memory!)



- $S = 1, R = 1$: then $Q = 0$ and $\bar{Q} = 0$ (invalid state: $Q \neq \text{NOT } \bar{Q}$)

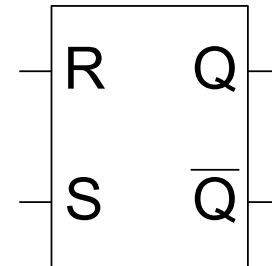


State table

inputs		SR			
		00	01	11	10
PS	Q(t)	0	0	X	1
	1	1	0	X	1

↗

SR Latch Symbol



$Q(t+1)$ NS (next state)

Characteristic Expression

$$Q(t+1) = S(t) + R'(t)Q(t)$$

SR Latch Symbol

- SR stands for Set/Reset Latch
 - Stores one bit of state (Q)
- Control what value is being stored with S , R inputs
 - **Set:** Make the output 1 ($S = 1$, $R = 0$, $Q = \mathbf{1}$)
 - **Reset:** Make the output 0 ($S = 0$, $R = 1$, $Q = \mathbf{0}$)

Must do something to avoid invalid state ($S = R = 1$)