



江西理工大学

Jiangxi University of Science and Technology

信息工程学院

School of information engineering



Dr Ata Jahangir Moshayedi

Clip Lecture series



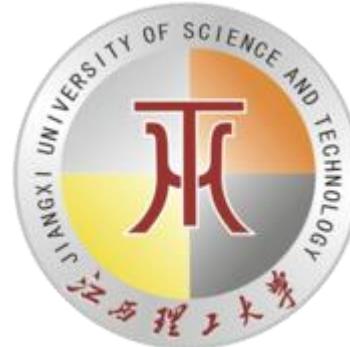
Digital System Design

Spring_2020

Prof Associate , School of information engineering Jiangxi university of science and technology, China

EMAIL: ajm@jxust.edu.cn





Jiangxi University of Science and Technology

Sequential Circuits

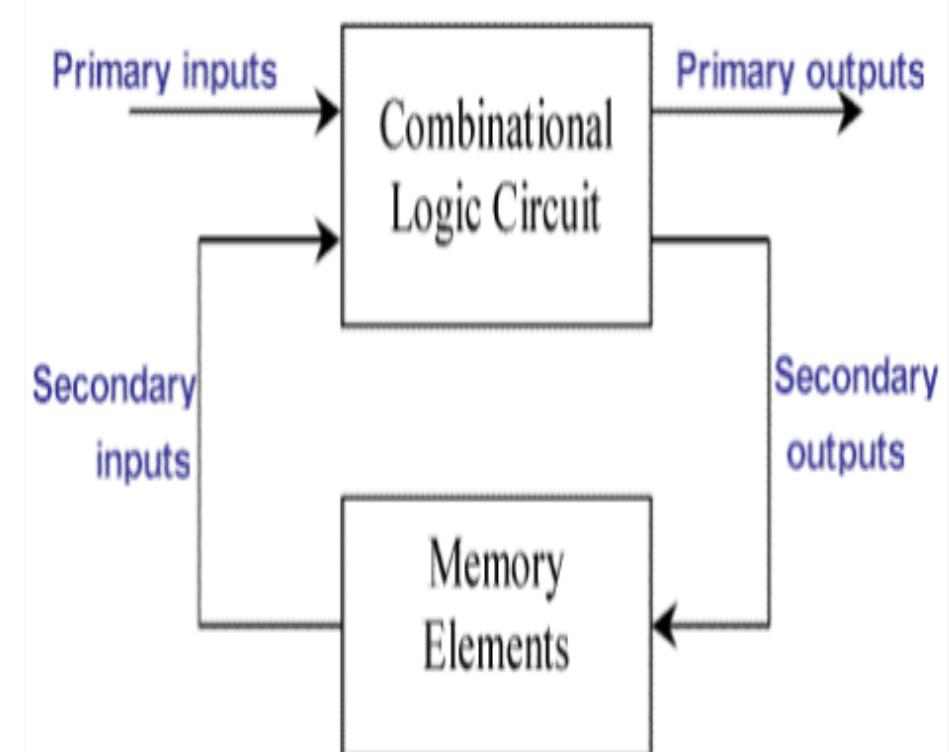
S-R Latch and D Latch





Sequential Logic Circuit

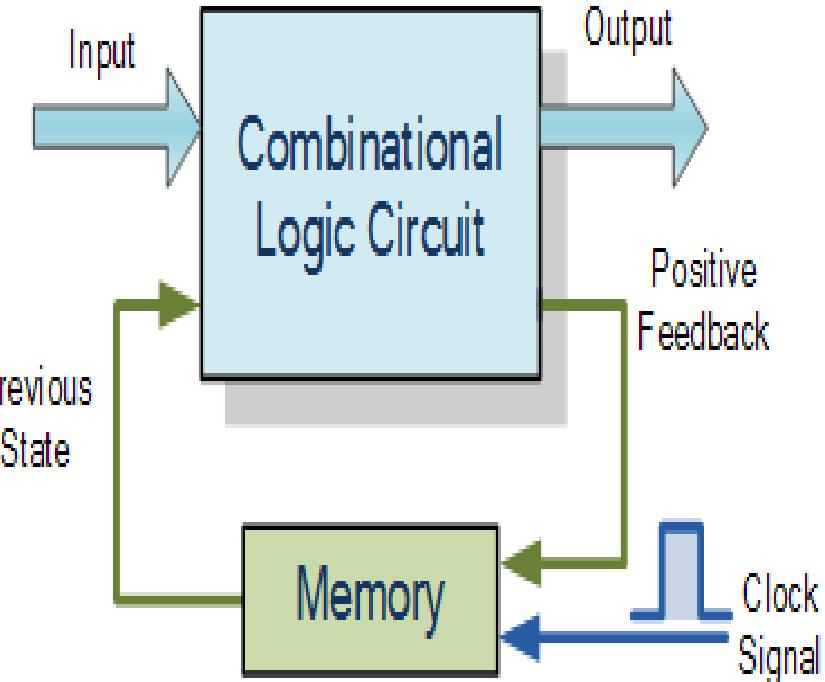
- The output state of a “**sequential logic circuit**” is a function of the following three states, the “**present input**”, the “**past input**” and/or the “**past output**”.
- ***Sequential Logic circuits*** remember these conditions and stay fixed in their current state until the next clock signal changes one of the states, giving sequential logic circuits “**Memory**”.





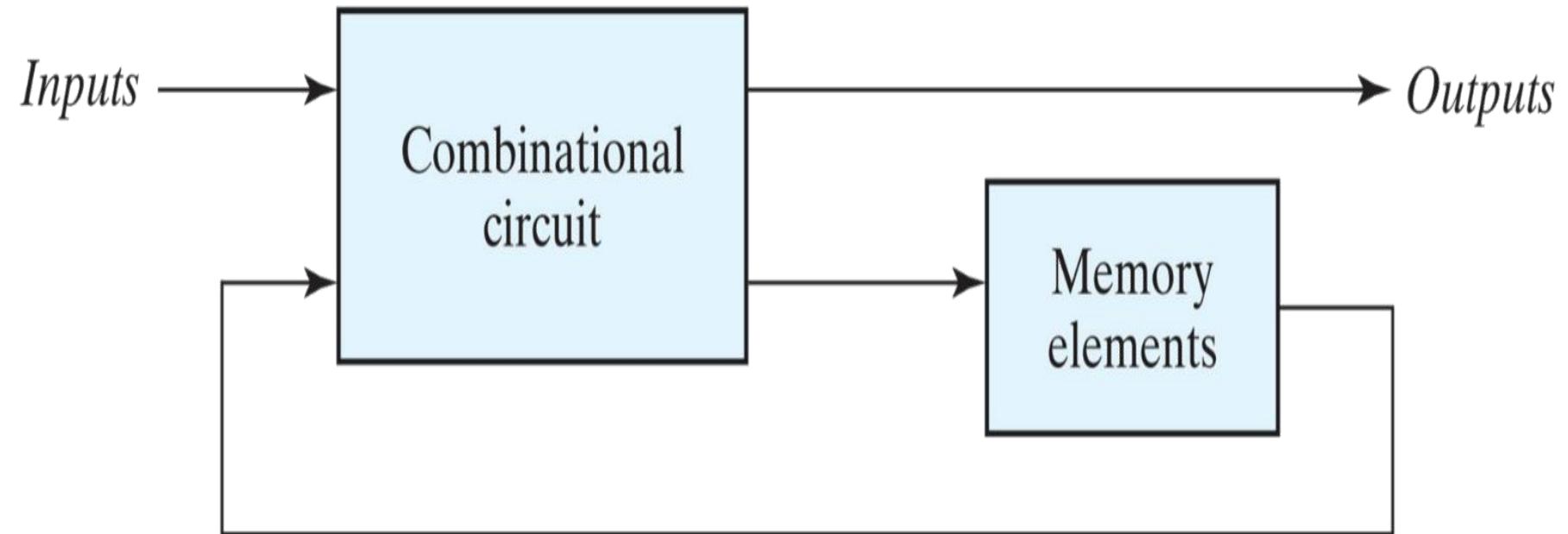
Sequential Logic Circuit

- Simple sequential logic circuits can be constructed from standard Bistable circuits such as: **Flip-flops, Latches and Counters** and which themselves can be made by simply connecting together universal NAND Gates and/or NOR Gates in a particular combinational way to produce the required sequential circuit.

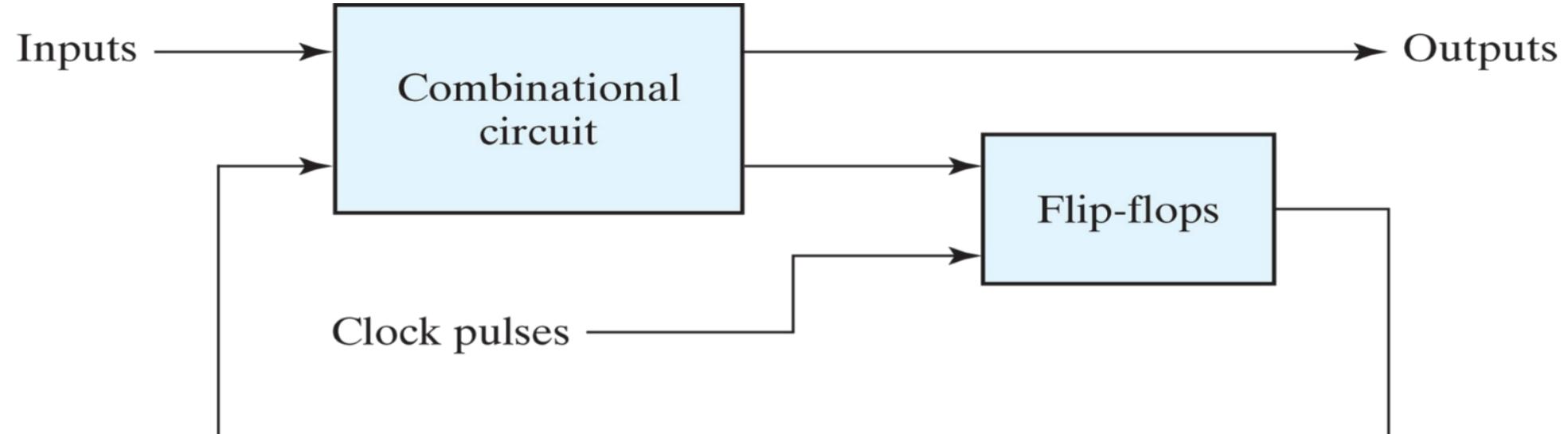




Block Diagram Of Sequential Circuit



Synchronous Clocked Sequential Circuit



(a) Block diagram



(b) Timing diagram of clock pulses



Sequential Logic Circuit

- There are two types of memory elements based on the type of triggering that is suitable to operate it.

- Latches

- Flip-flops

- **Latches operate with enable signal, which is level sensitive.**

Whereas

flip-flops are edge sensitive.

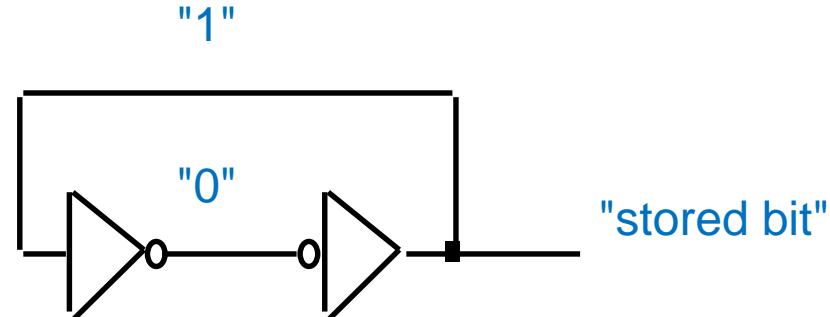
Parameter	Latch	Flip Flop
AREA	LESS	MORE
POWER	LESS	MORE
DESIGN ROBUSTNESS	DESIGN IS NOISY	DESIGN IS ROBUST

How do we store info?

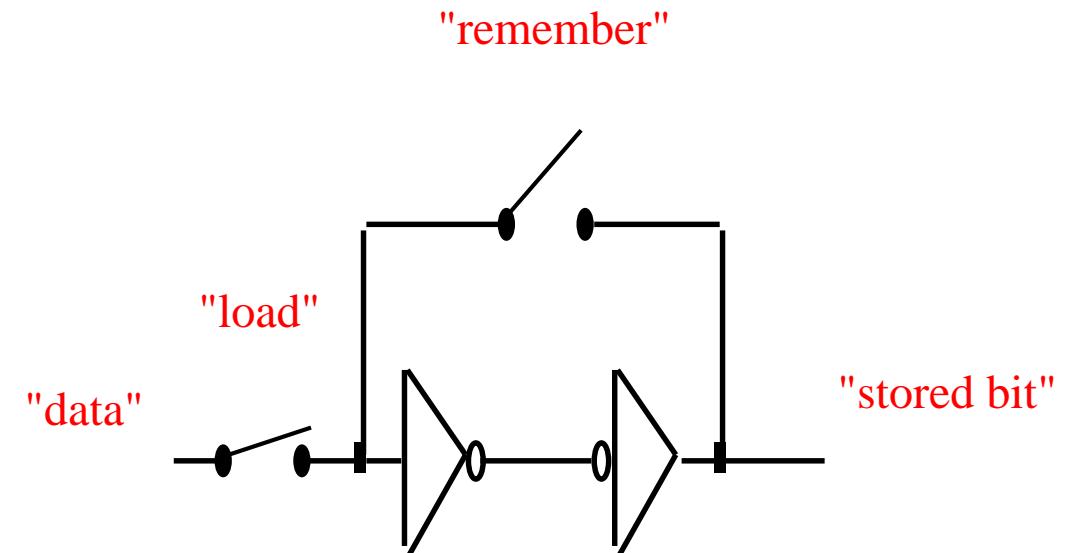
Feedback



- Two inverters can hold a bit
 - As long as power is applied



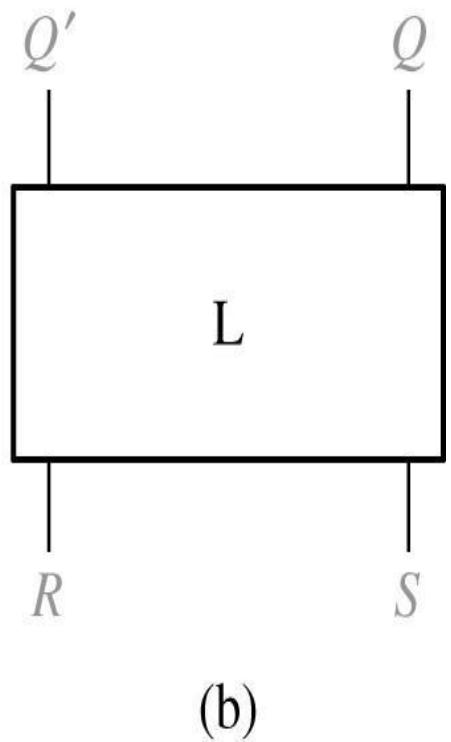
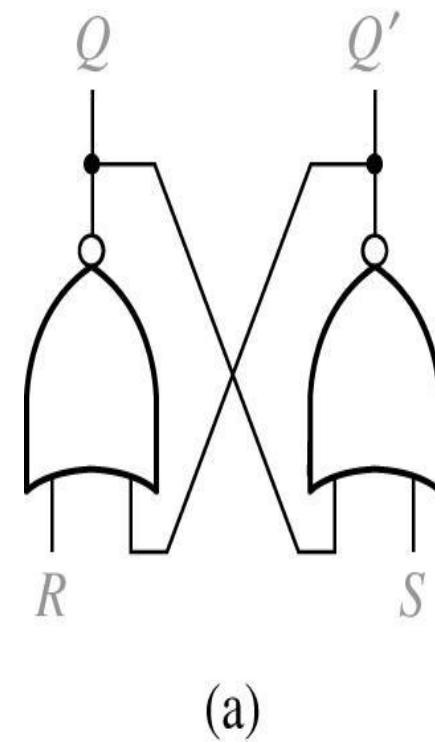
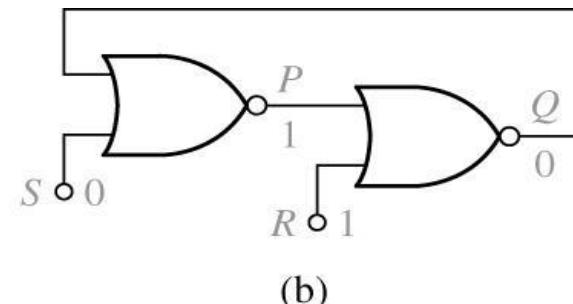
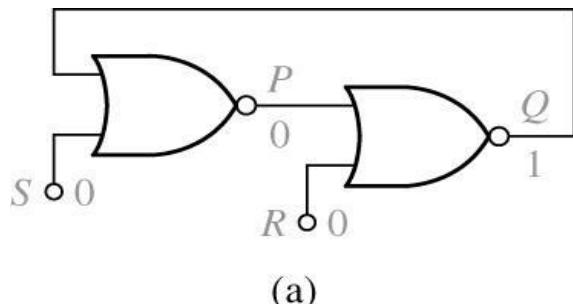
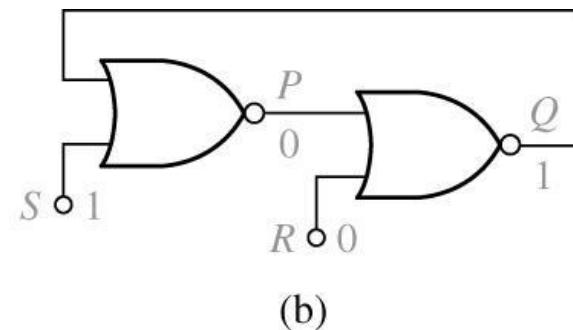
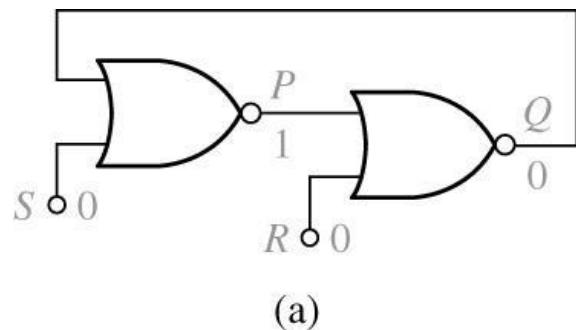
- Storing a new memory
 - Temporarily break the feedback path



S-R Latch



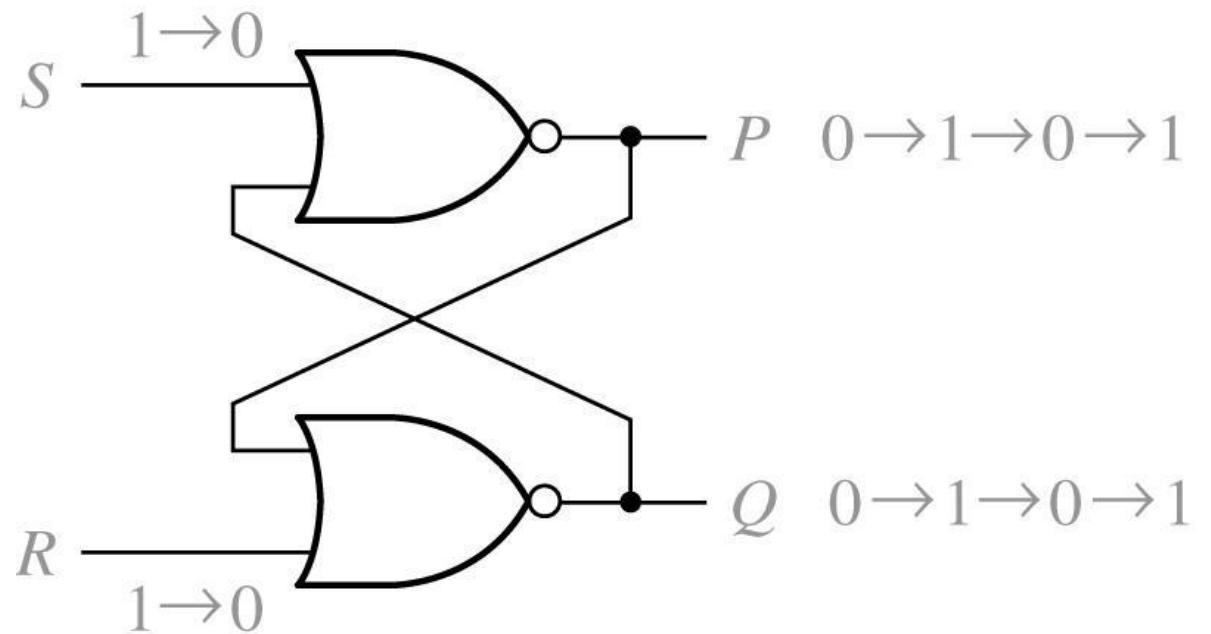
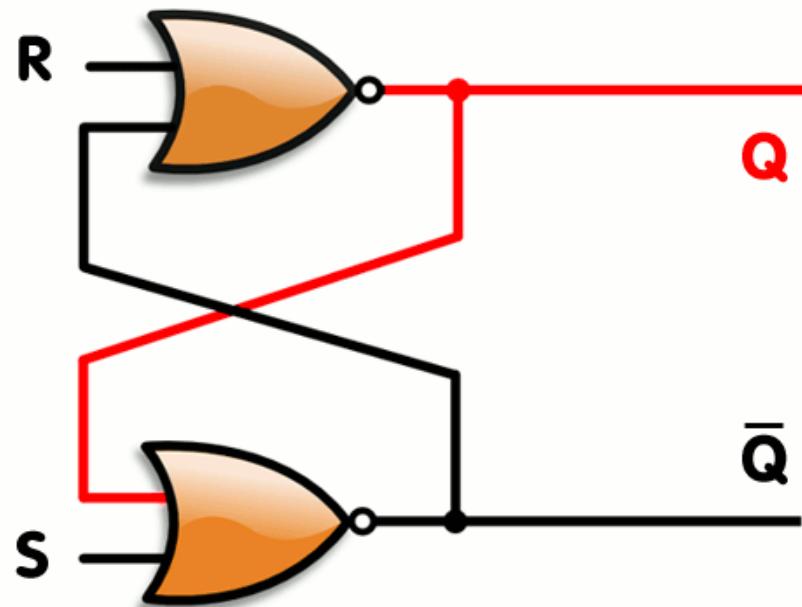
- Set-reset latch
 - Use NOR gate to construct a stable state network



S-R Latch



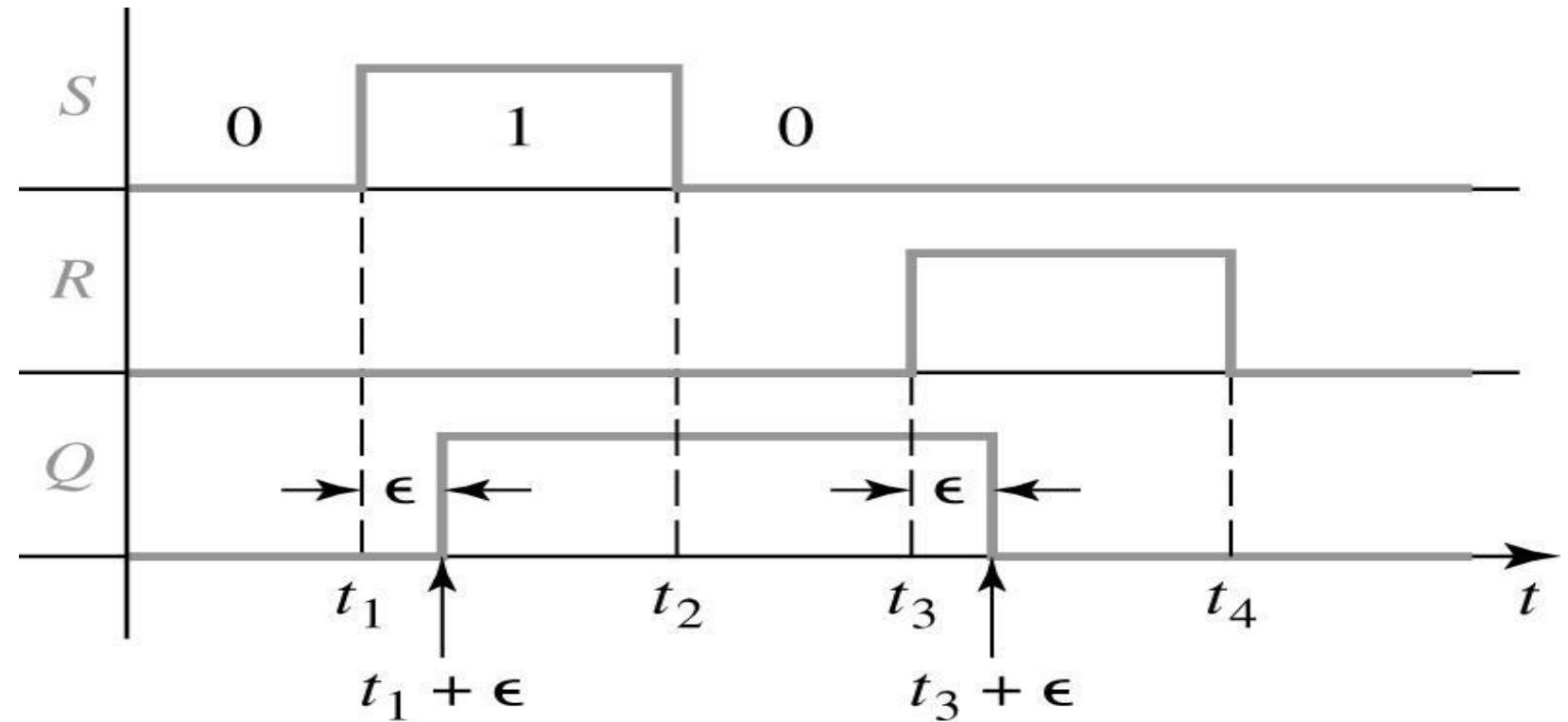
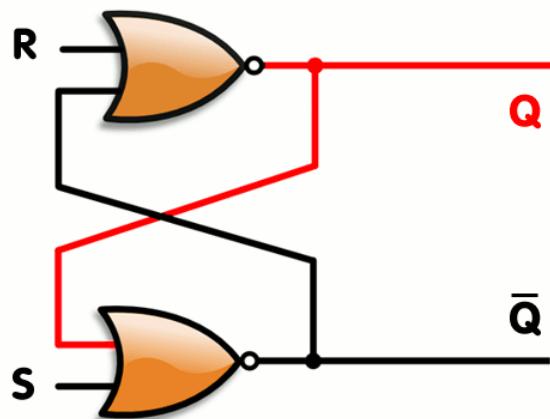
- When $S=R=1$, the S-R latch will not operate properly.
(Is it a stable state, if $S=R=1$?)
 - Q and P are not complementary.
 - If $S=R=1$ changed to $S=R=0$, then the network will oscillate assuming both gates have the same delay. (Critical race occurs)



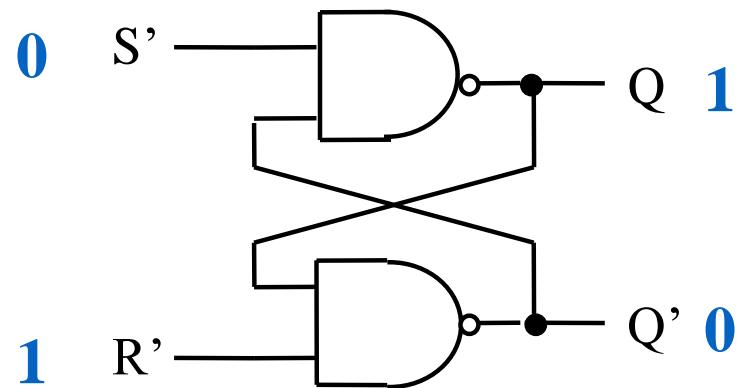
S-R Latch Timing and State



S duration > delay time



SR Latch (NAND version)

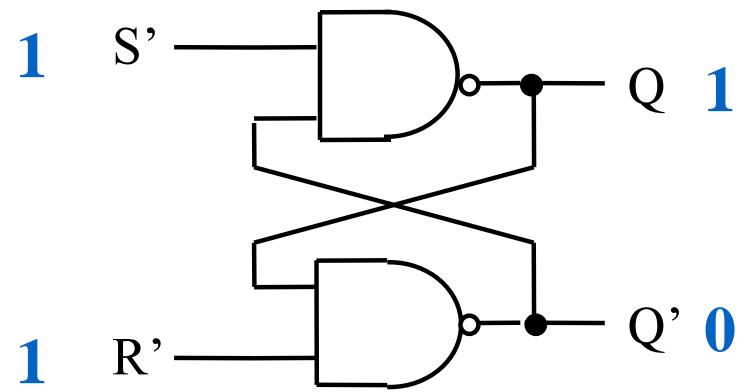


S'	R'	Q	Q'
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	1

1 0 Set

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

SR Latch (NAND version)



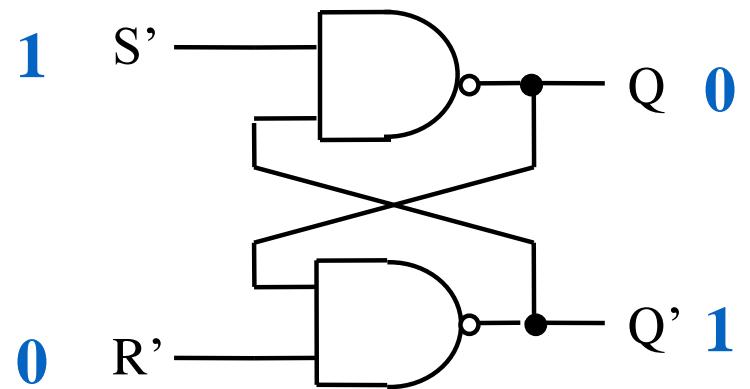
S'	R'	Q	Q'
0	0		
0	1	1	0
1	0	0	1
1	1	1	0

Set

Hold

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

SR Latch (NAND version)

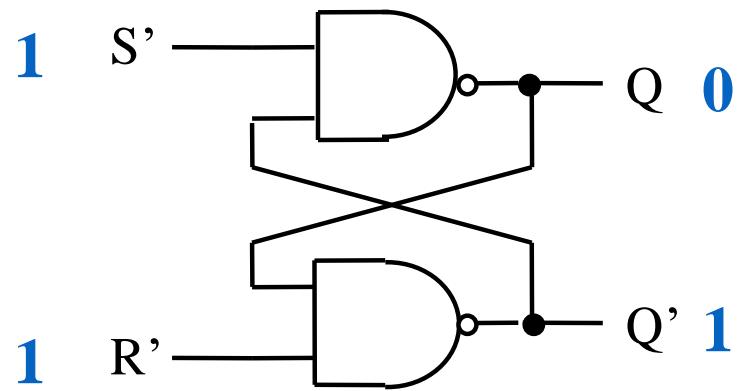


S'	R'	Q	Q'
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	1

1 0 Set
0 1 Reset
1 0 Hold

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

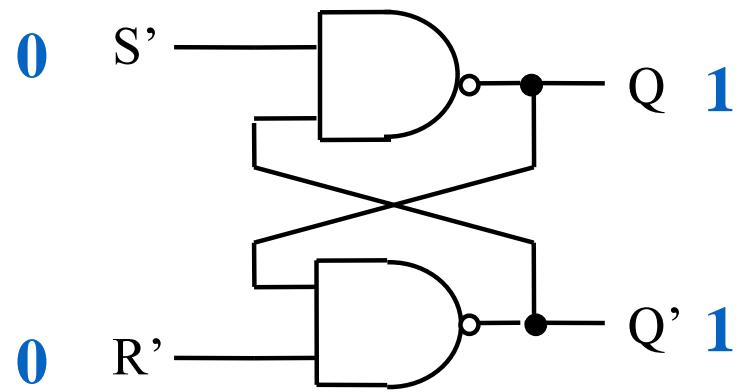
SR Latch (NAND version)



S'	R'	Q	Q'
0	0		
0	1	1	0
1	0	0	1
1	1	1	0
		Set	
		Reset	
		Hold	
		Hold	

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

SR Latch (NAND version)



S'	R'	Q	Q'
0	0	1	1 Disallowed
0	1	1	0 Set
1	0	0	1 Reset
1	1	1	0 Hold
0	1	0	1 Hold

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

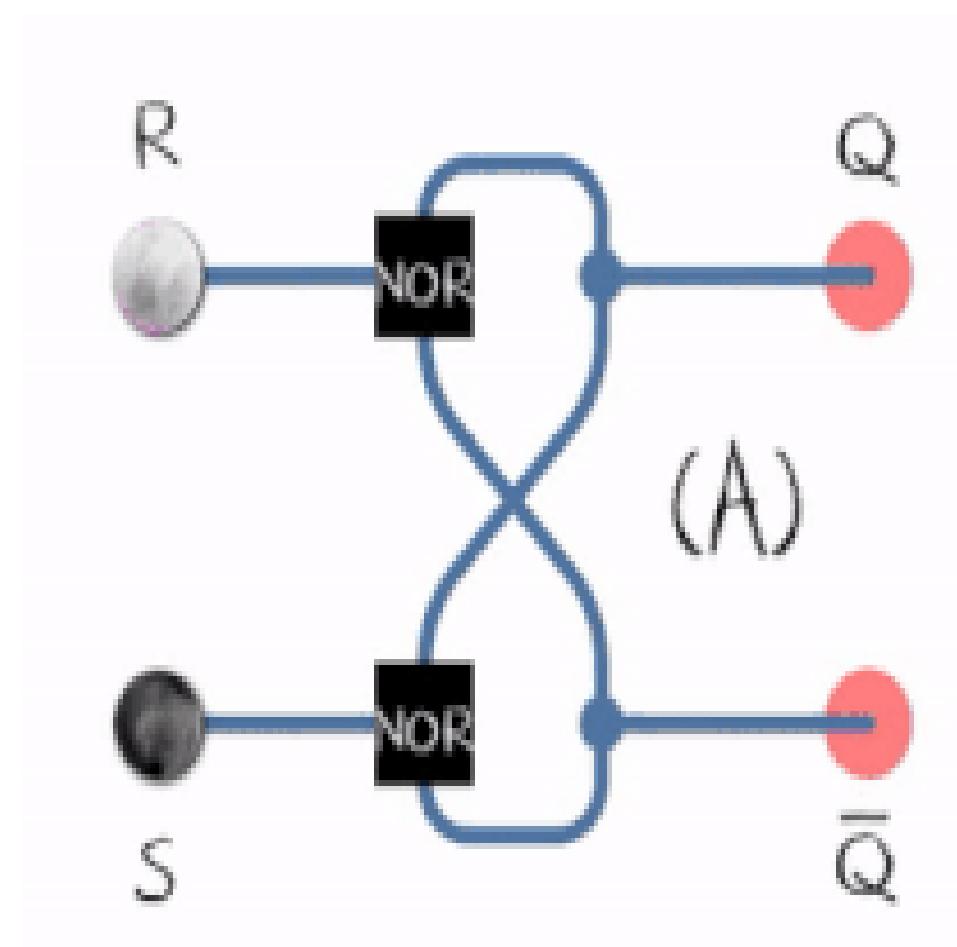
SR Latch

- An animated SR latch. Black and white mean logical '1' and '0', respectively.

- (A) $S = 1, R = 0$: set
- (B) $S = 0, R = 0$: hold
- (C) $S = 0, R = 1$: reset
- (D) $S = 1, R = 1$: not allowed

Transitioning from the restricted combination

(D) to (A) leads to an unstable state.



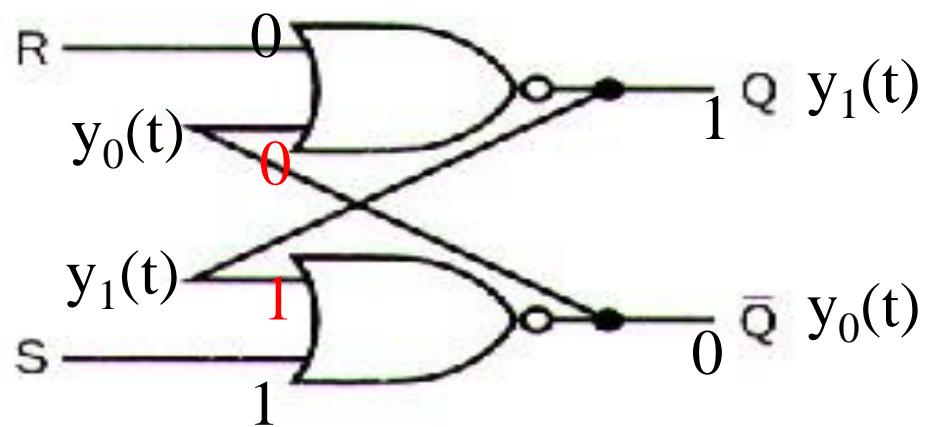
S-R latch behavior

- S-R latch behavior
 - Present state
 - The state of Q output at the time the input signals are applied.
 - Next state
 - The state of Q output after the latch has reacted to the input signals.

$S(t)$	$R(t)$	$Q(t)$	$Q(t + \epsilon)$
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	- } inputs not allowed
1	1	1	

S R latch Analysis

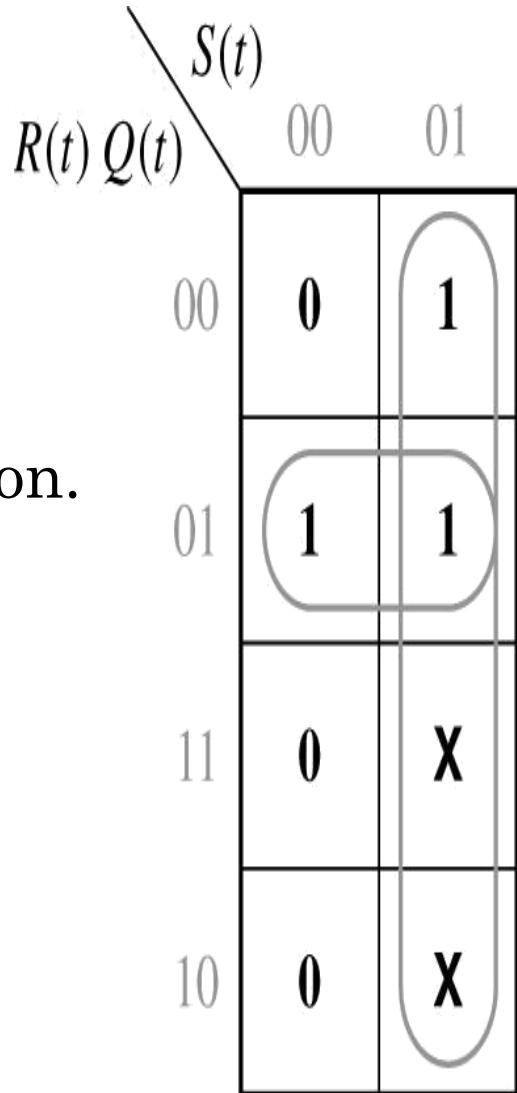
- Total state table
- If next state = present state, stable



		Current input ($S(t), R(t)$)				Next state
		(0, 0)	(0, 1)	(1, 0)	(1, 1)	
Present state	(0, 0)	(1, 1)	(1, 0)	(0, 1)	(0, 0)	$(y_0(t + t_{pd}), y_1(t + t_{pd}))$
	(0, 1)	(0, 1)	(0, 0)	(0, 1)	(0, 0)	
	(1, 0)	(1, 0)	(0, 0)	(0, 0)	(0, 0)	
	(1, 1)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	

K-map for $Q(t+\epsilon)$

- $Q^+ = S + R'Q$ (SR=0)
 - S and R can not be 1 at the same time.
 - Q: present state
 - Q^+ : next state
 - Next state equation or characteristic equation.

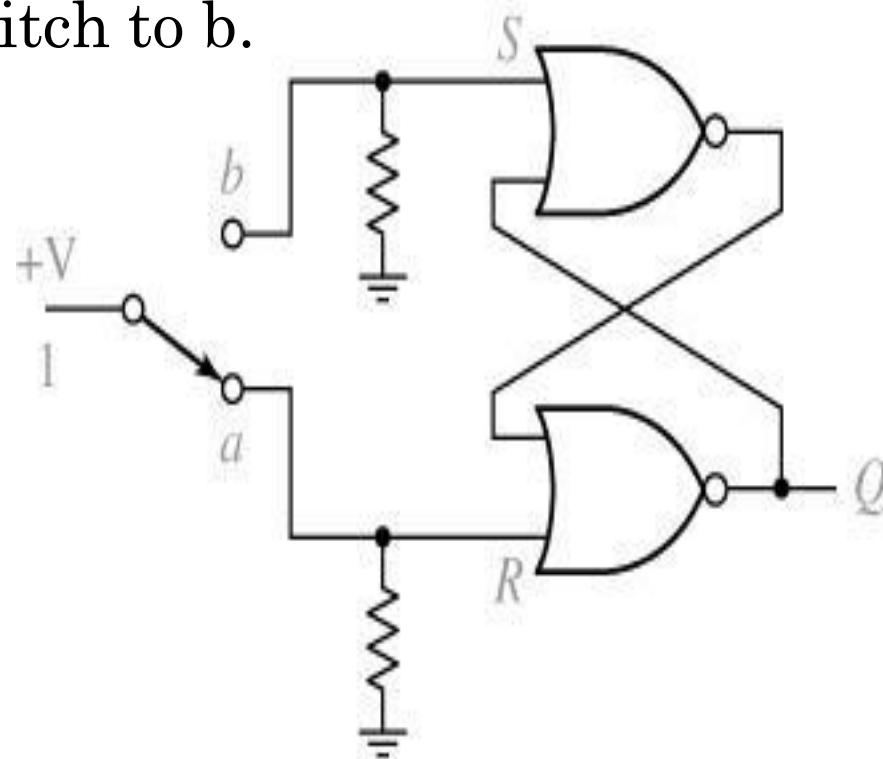


$$Q(t + \epsilon) = S(t) + R'(t) Q(t)$$

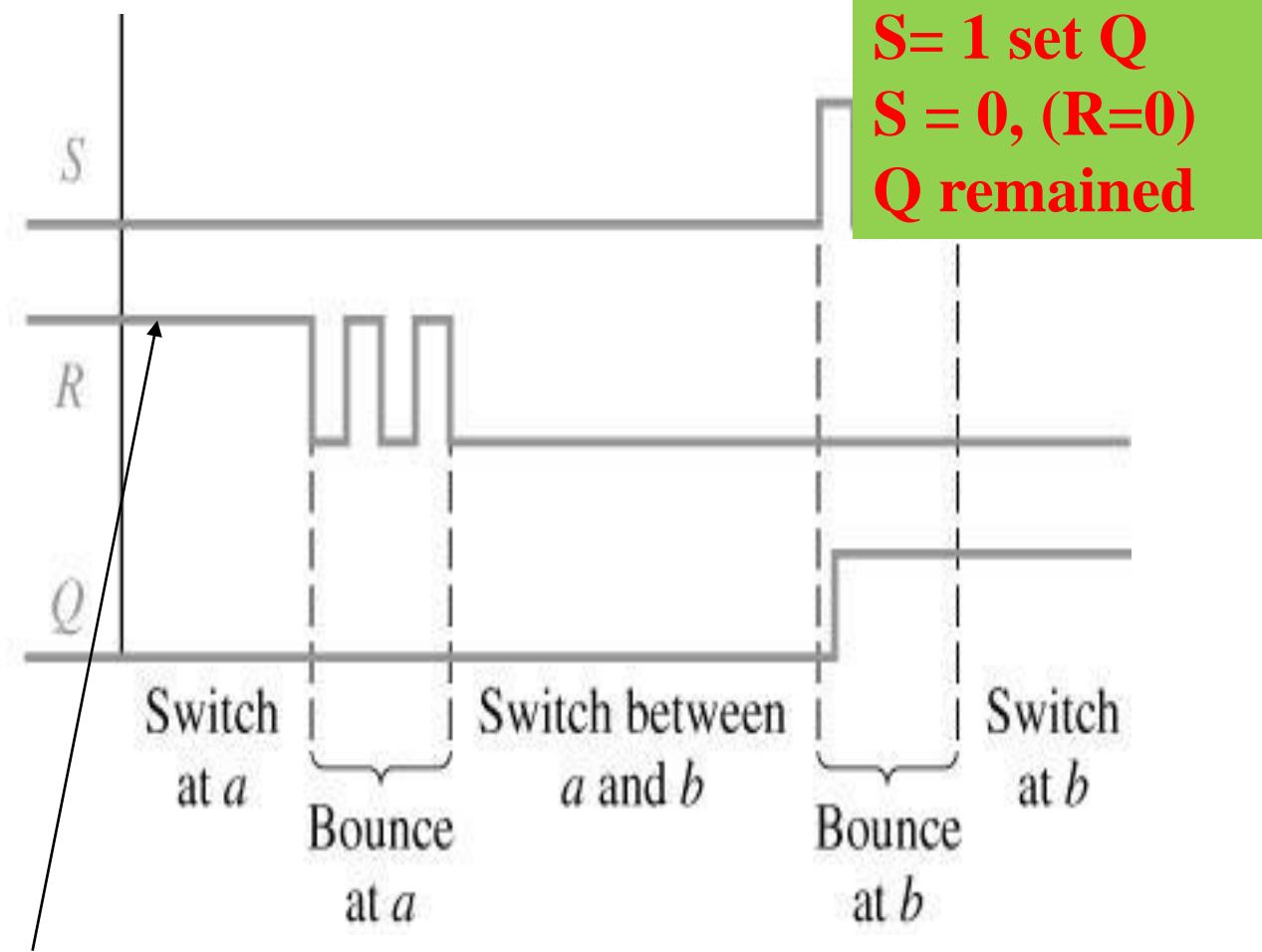
Debouncing Circuit



- Use S-R latch for debouncing.
- Pull-down resistors
- a switch to b.



R = 1 set Q = 0

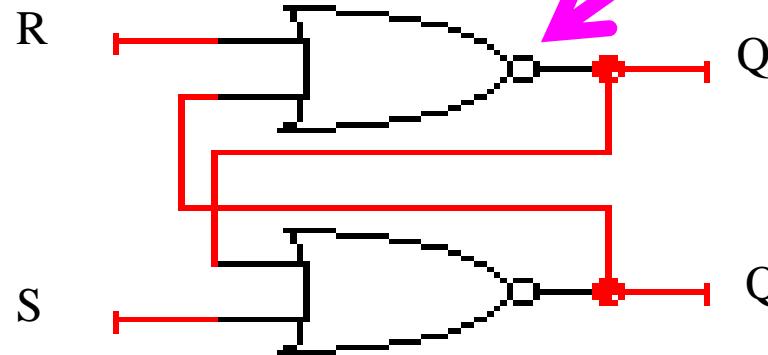


SR latch behavior

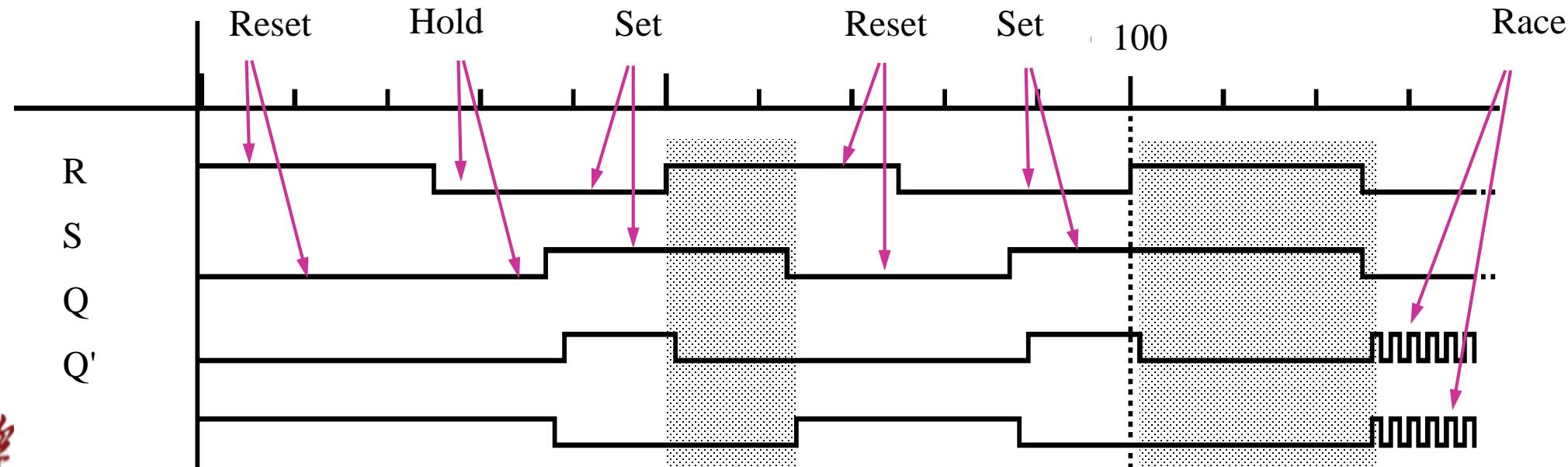
- Truth table and timing

NOR output is 1

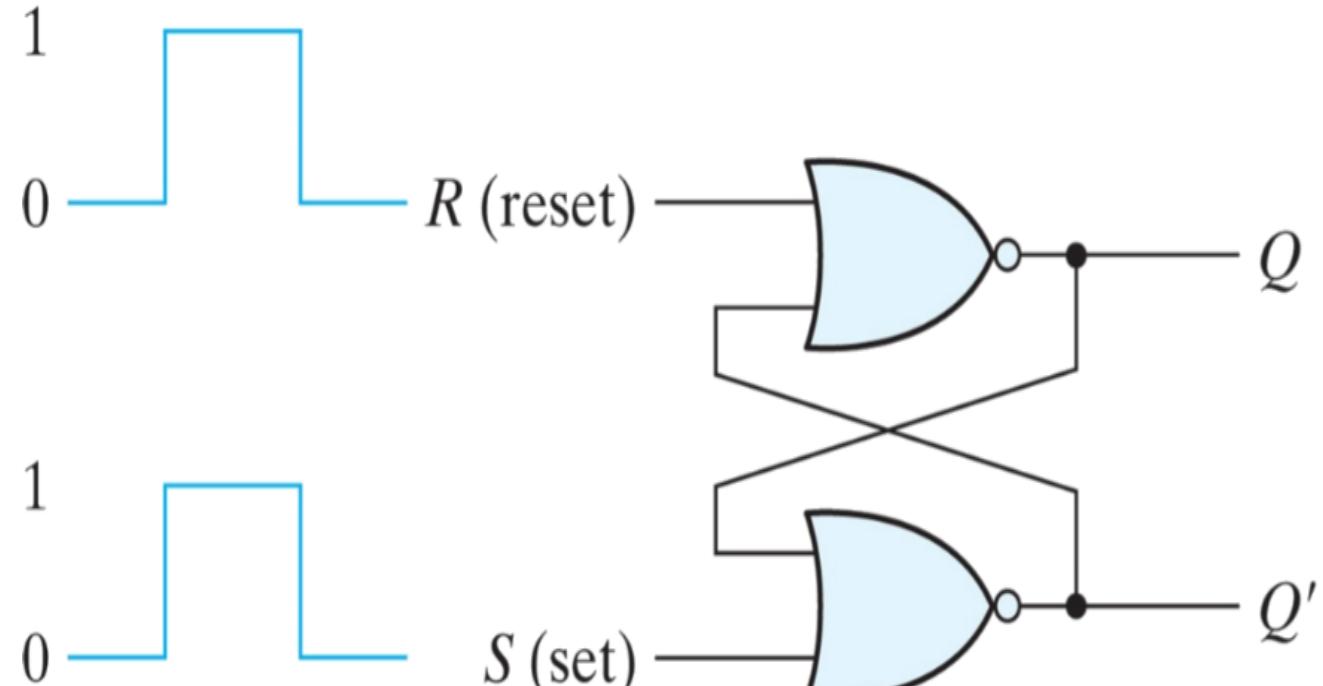
Only when both inputs are 0



S	R	Q
0	0	hold
0	1	0
1	0	1
1	1	disallow



SR latch with NOR gates



(a) Logic diagram

S	R	Q	Q'
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

(b) Function table

Notes: (after $S = 1, R = 0$)
 (after $S = 0, R = 1$)
 (forbidden)

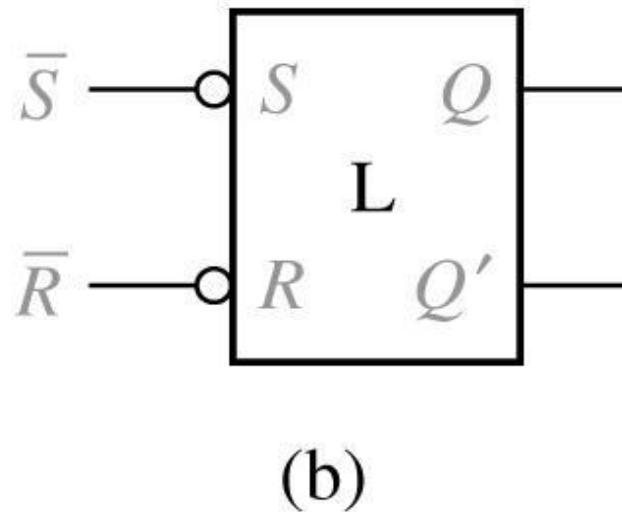
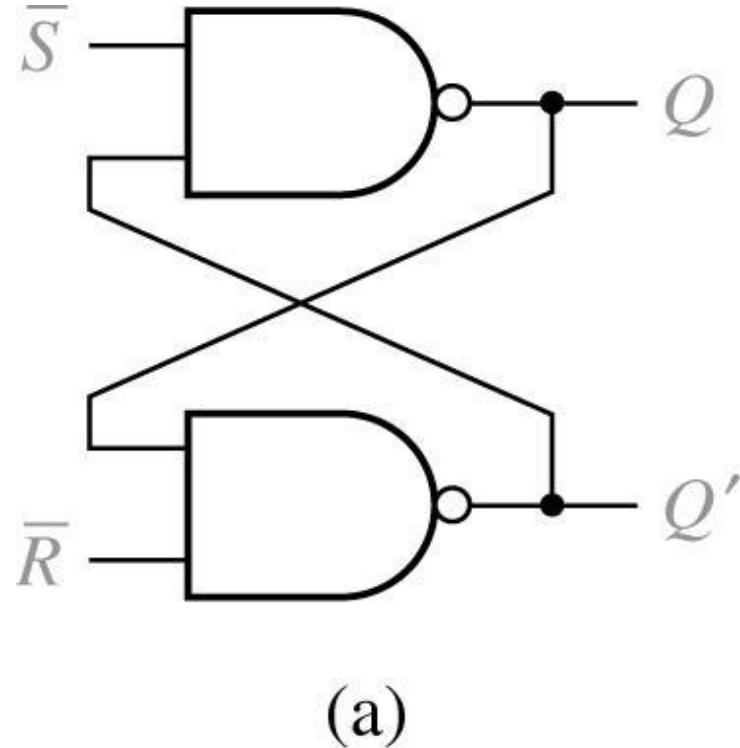


The Basic RS NOR Latch

- For the NOR latch circuit, both inputs should normally be at a logic 0 level. Changing an input to a logic 1 level will force that output to a logic 0. The same logic 0 will also be applied to the second input of the other NOR gate, allowing that output to rise to a logic 1 level. This in turn feeds back to the second input of the original gate, forcing its output to remain at logic 0 even after the external input is removed.
- Applying another logic 1 input to the same gate will have no further effect on this circuit. However, applying a logic 1 to the *other* gate will cause the same reaction in the other direction, thus changing the state of the latch circuit the other way.
- Note that it is forbidden to have both inputs at a logic 1 level at the same time. That state will force both outputs to a logic 0, overriding the feedback latching action. In this condition, whichever input goes to logic 0 first will lose control, while the other input (still at logic 1) controls the resulting state of the latch. If both inputs go to logic 0 simultaneously, the result is a "race" condition, and the final state of the latch cannot be determined ahead of time.

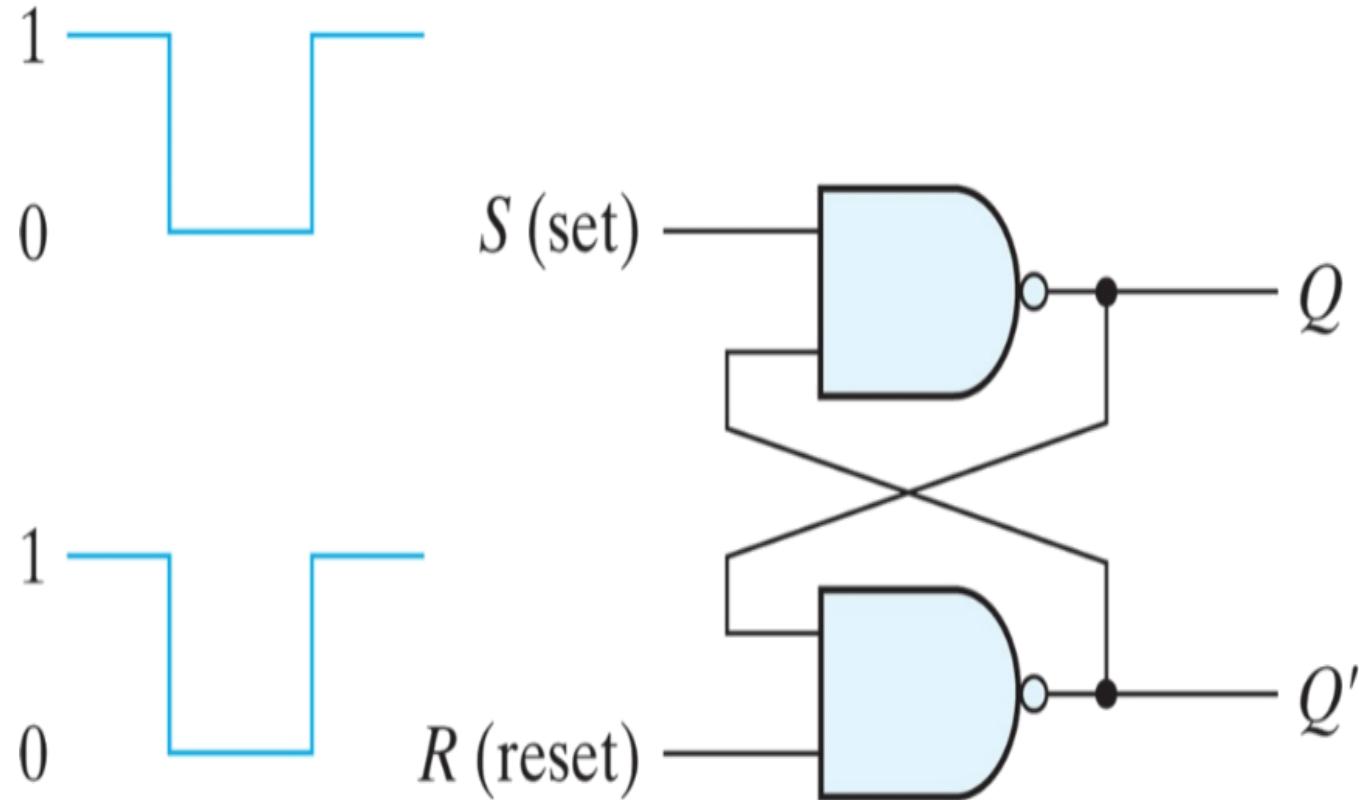
SR latch with NAND gates

- S#-R# Latch, when $S\#=0$ sets $Q = 1$ and $R\#=0$ resets $Q = 0$



S#	R#	Q	Q+
1	1	0	0
1	1	1	1
1	0	0	0
1	0	1	0
0	1	0	1
0	1	1	1
0	0	0	-
0	0	1	-

SR latch with NAND gates



(a) Logic diagram

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)

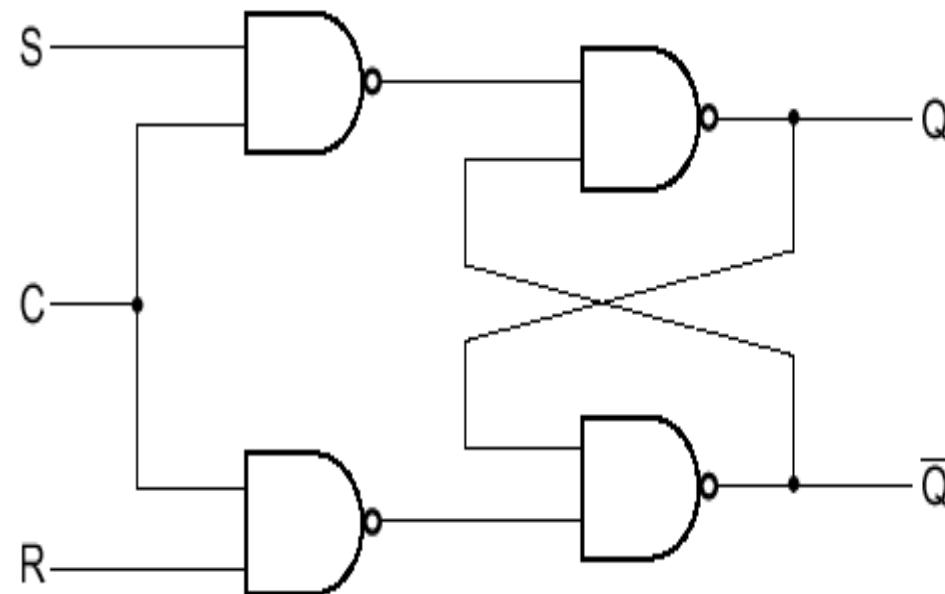
(b) Function table



The Basic RS NAND Latch

- For the NAND latch circuit, both inputs should normally be at a logic 1 level. Changing an input to a logic 0 level will force that output to a logic 1. The same logic 1 will also be applied to the second input of the other NAND gate, allowing that output to fall to a logic 0 level. This in turn feeds back to the second input of the original gate, forcing its output to remain at logic 1.
- Applying another logic 0 input to the same gate will have no further effect on this circuit. However, applying a logic 0 to the *other* gate will cause the same reaction in the other direction, thus changing the state of the latch circuit the other way.
- Note that it is forbidden to have both inputs at a logic 0 level at the same time. That state will force both outputs to a logic 1, overriding the feedback latching action. In this condition, whichever input goes to logic 1 first will lose control, while the other input (still at logic 0) controls the resulting state of the latch. If both inputs go to logic 1 simultaneously, the result is a "race" condition, and the final state of the latch cannot be determined ahead of time.

SR Latch with Clock signal



(a) Logic diagram

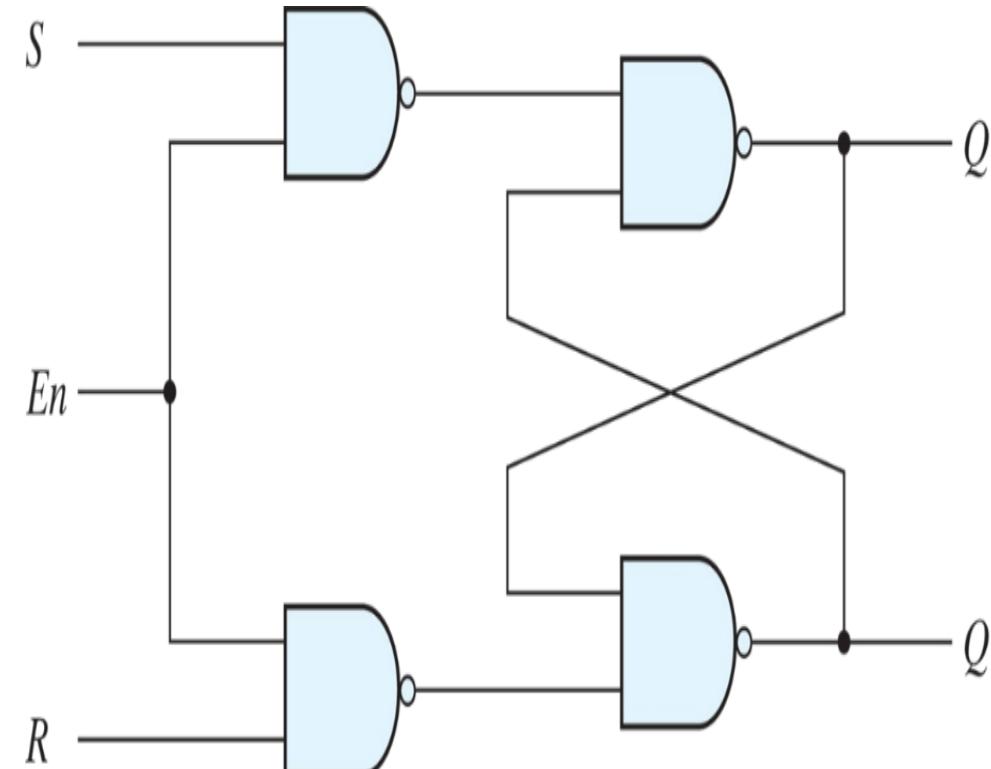
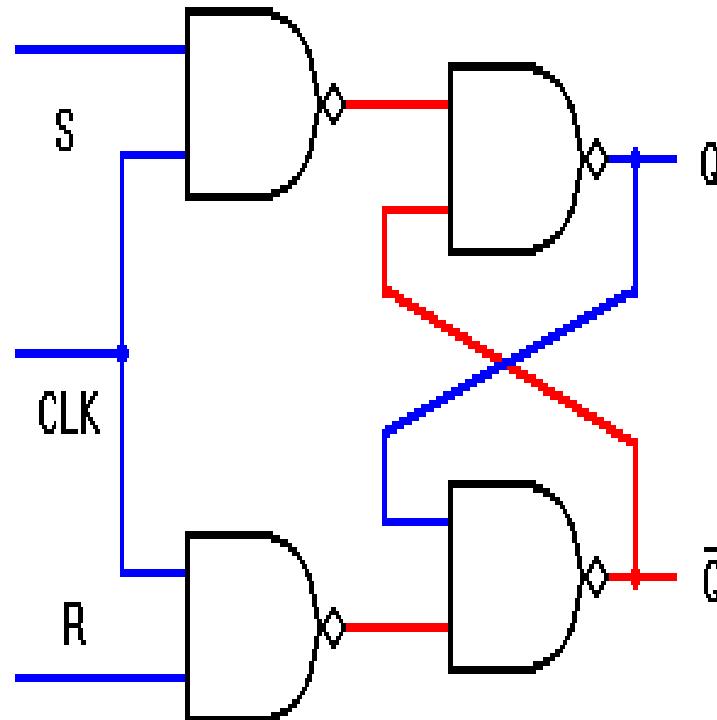
C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	Q = 0; Reset state
1	1	0	Q = 1; Set state
1	1	1	Undefined

(b) Function table

Latch is sensitive to input changes ONLY when $C=1$



SR latch with control input



(a) Logic diagram

(b) Function table

En	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	Indeterminate

Copyright ©2013 Pearson Education, publishing as Prentice Hall



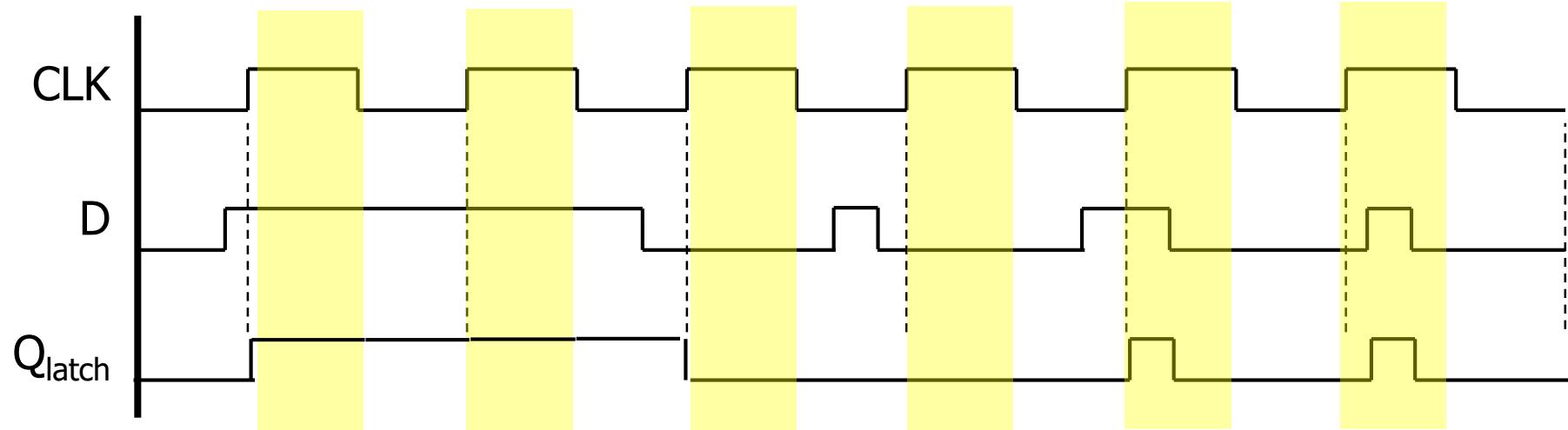
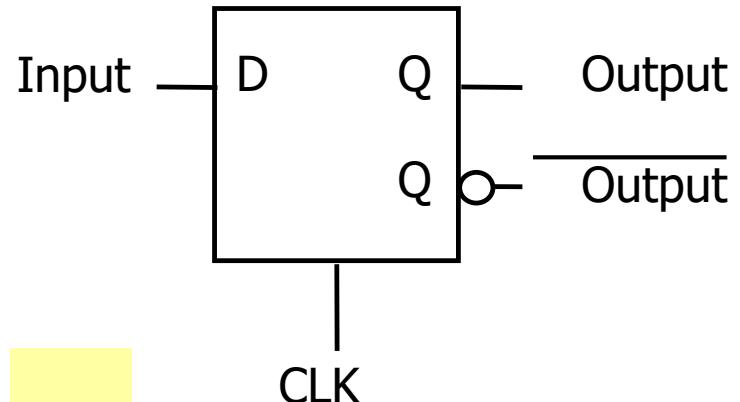
SR latch with control input

- The clocked RS latch circuit is very similar in operation to the basic latch you examined on the previous page. The S and R inputs are normally at logic 0, and must be changed to logic 1 to change the state of the latch. However, with the third input, a new factor has been added. This input is typically designated C or CLK, because it is typically controlled by a clock circuit of some sort, which is used to synchronize several of these latch circuits with each other. The output can only change state while the CLK input is a logic 1. When CLK is a logic 0, the S and R inputs will have no effect.
- The same rule about not activating both the S and R inputs simultaneously holds true: if both are logic 1 when the clock is also logic 1, the latching action is bypassed and both outputs will go to logic 1. The difference in this case is that if the CLK input drops to logic 0 first, there is no question or doubt -- a true race condition will exist, and you cannot tell which way the outputs will come to rest. The example circuit on this page reflects this uncertainty.
- For correct operation, the selected R or S input should be brought to logic 1, then the CLK input should be made logic 1 and then logic 0 again. Finally, the selected input should be returned to logic 0.



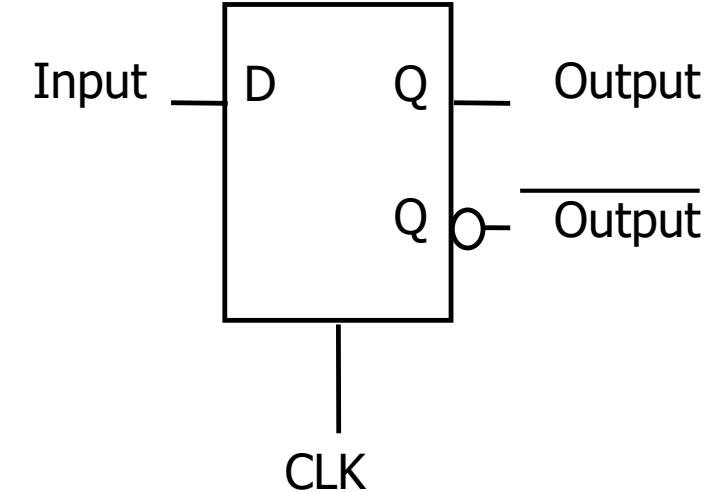
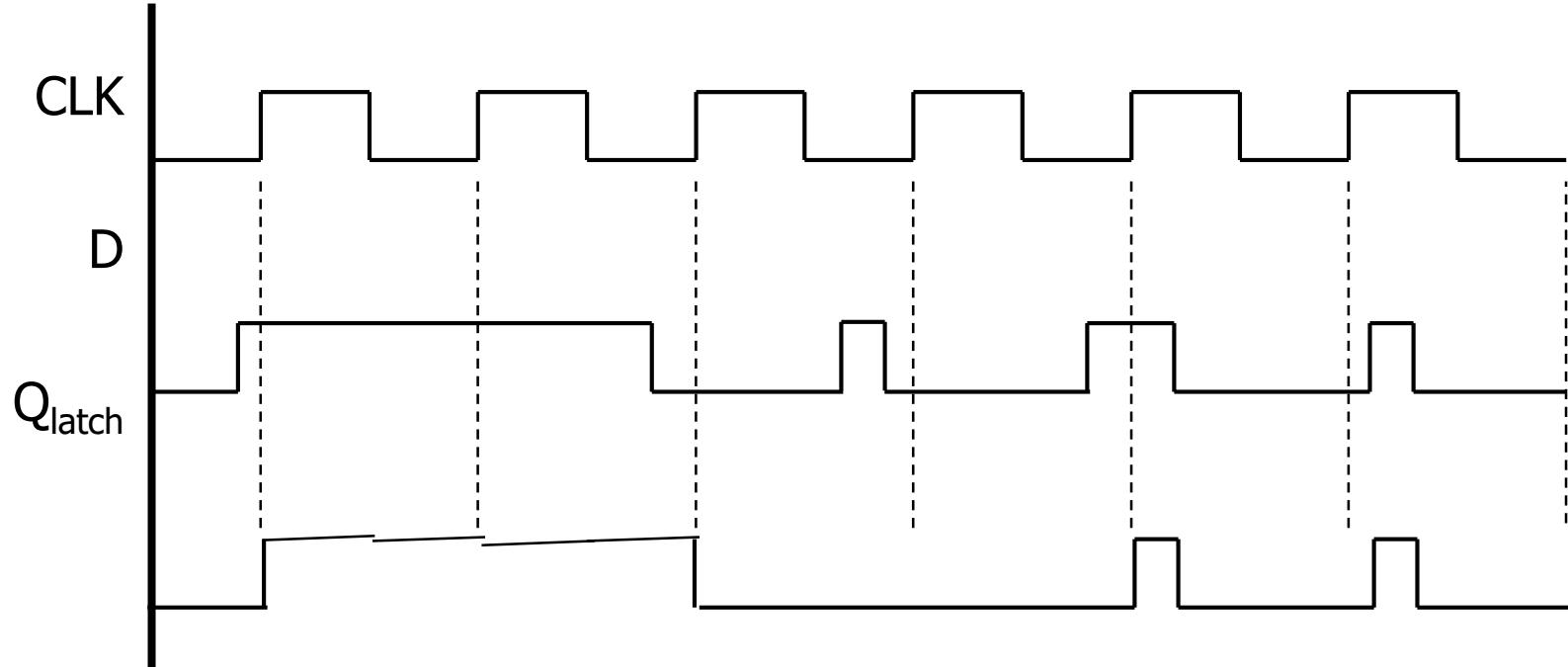
The D latch: store it and look it up

- Output depends on clock
 - Clock high: Input passes to output
 - Clock low: Latch holds its output
- Latches are level sensitive and “transparent”



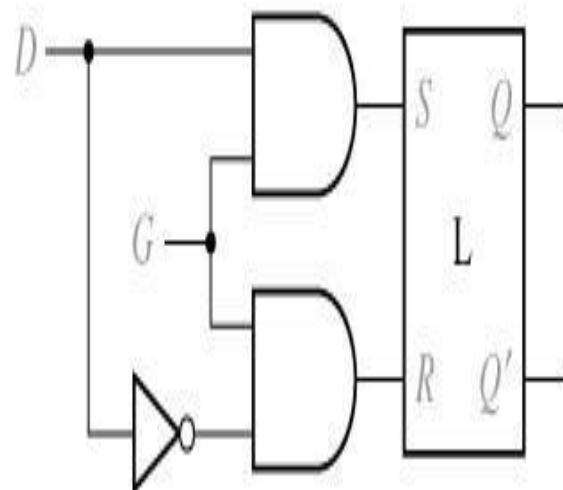


The D latch

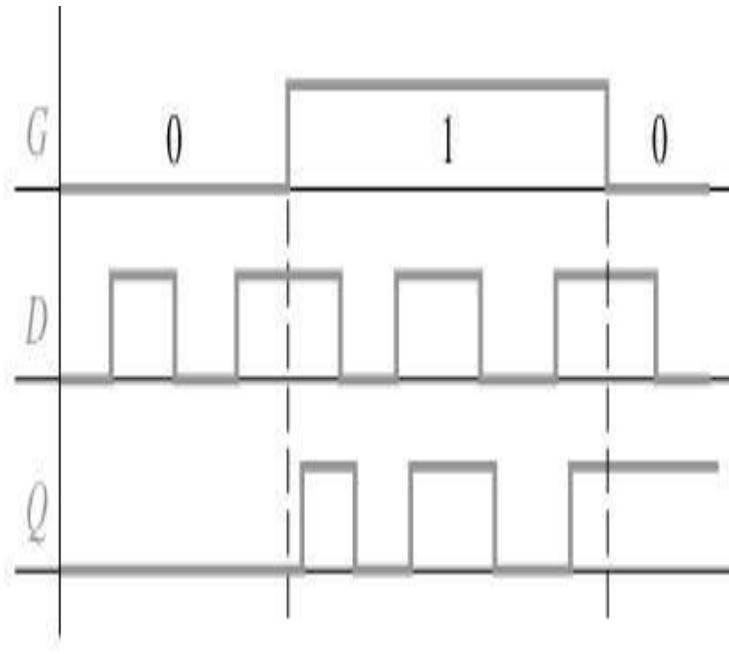


Gated D Latch

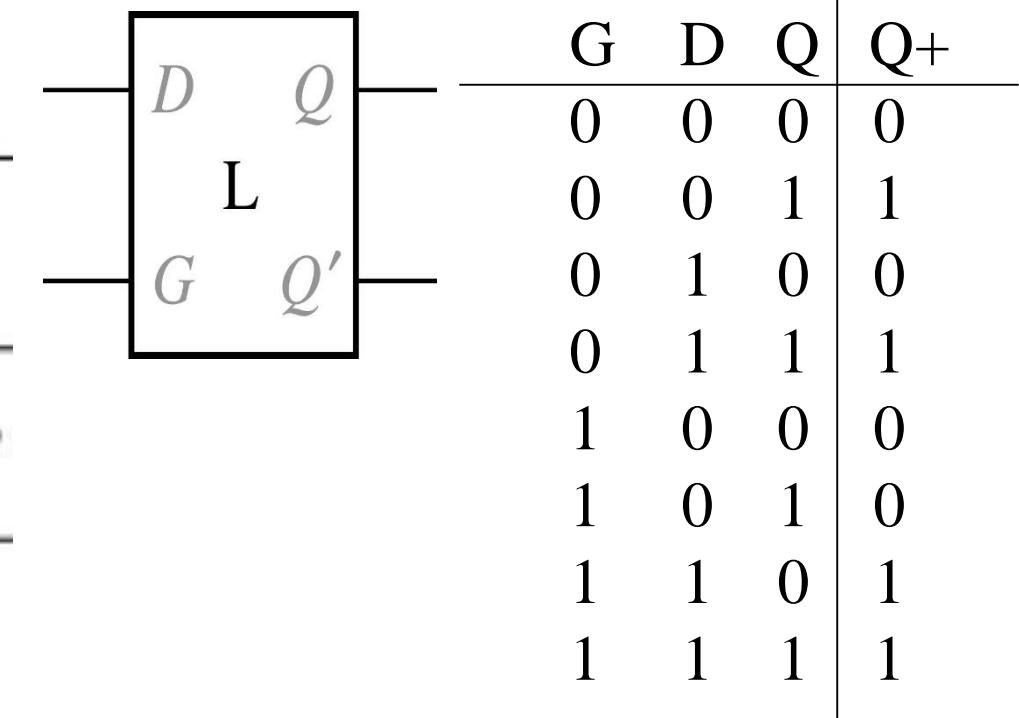
- Gate input G
 - Transparent latch (when $G=1$, $Q = D$)



(a)

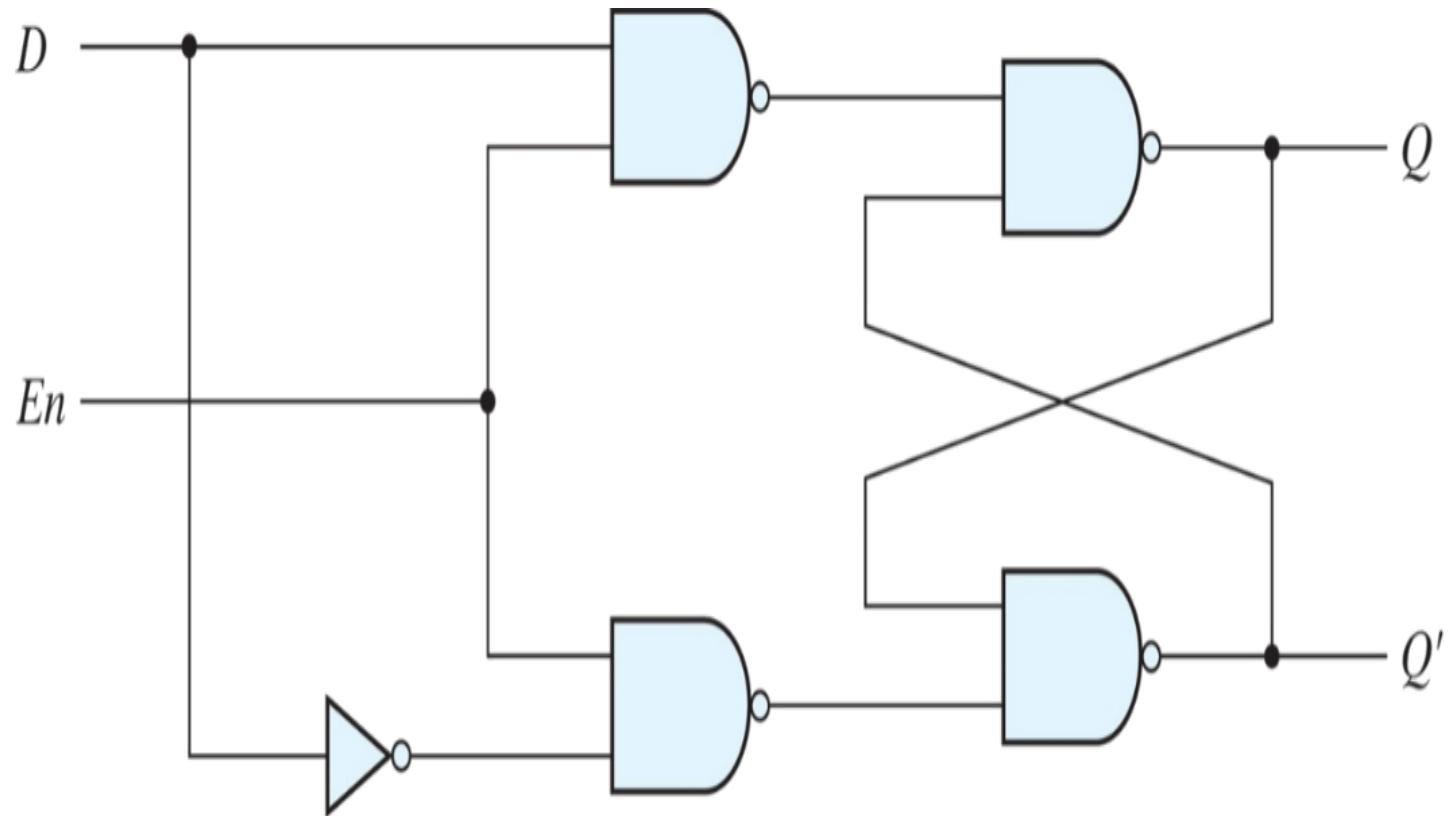


(b)





D latch



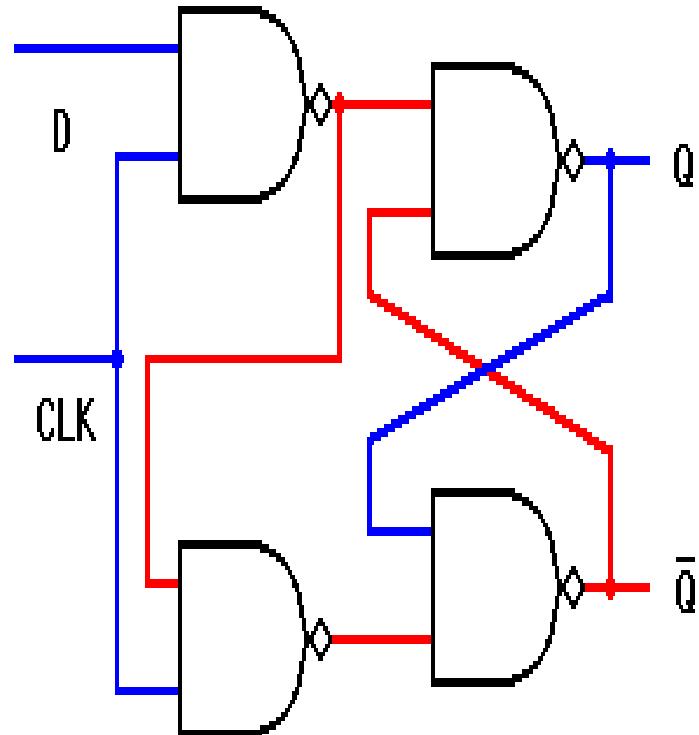
(a) Logic diagram

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

(b) Function table

D latch

- One very useful variation on the RS latch circuit is the Data latch, or D latch as it is generally called. As shown in the logic diagram below, the D latch is constructed by using the inverted S input as the R input signal.
- The single remaining input is designated "D" to distinguish its operation from other types of latches. It makes no difference that the R input signal is effectively clocked twice, since the CLK signal will either allow the signals to pass both gates or it will not.





D latch

- In the D latch, when the CLK input is logic 1, the Q output will always reflect the logic level present at the D input, no matter how that changes. When the CLK input falls to logic 0, the last state of the D input is trapped and held in the latch, for use by whatever other circuits may need this signal.
- Because the single D input is also inverted to provide the signal to reset the latch, this latch circuit cannot experience a "race" condition caused by all inputs being at logic 1 simultaneously. Therefore the D latch circuit can be safely used in any circuit.

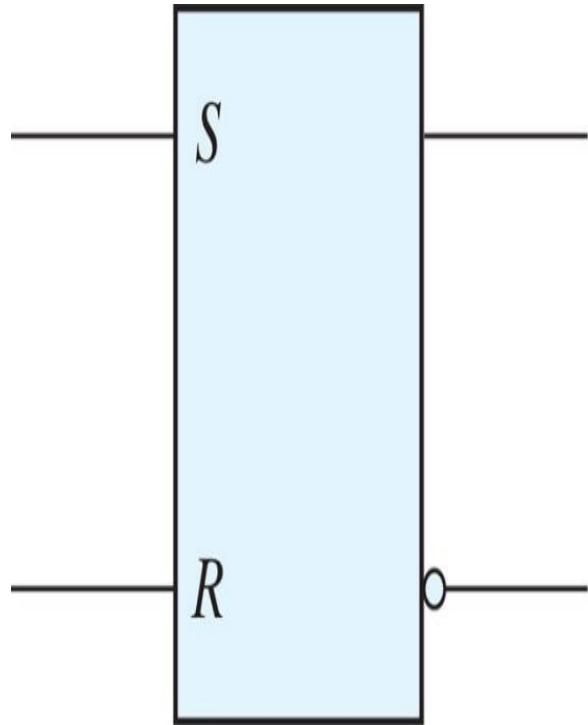


D latch

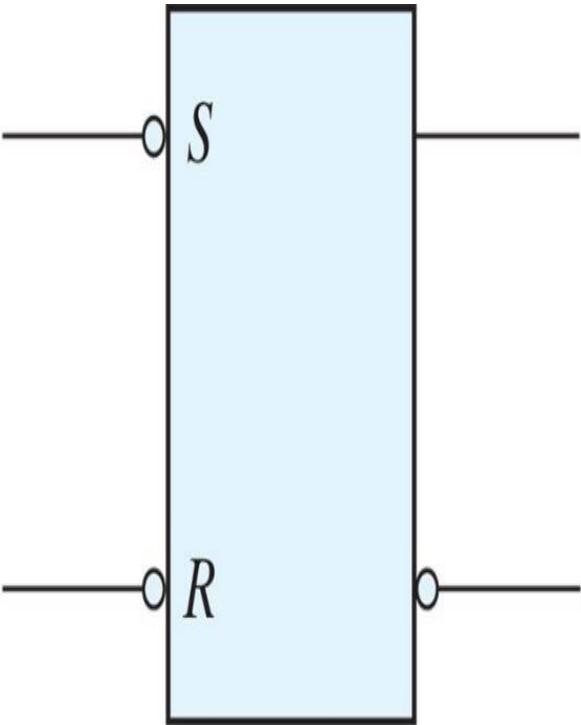
- Although the D latch does not *have* to be made edge triggered for safe operation, there are some applications where an edge-triggered D flip-flop is desirable.
- This can be accomplished by using a D latch circuit as the master section of an RS flip-flop, as shown on the next page. Both types are useful, so both are made commercially available.



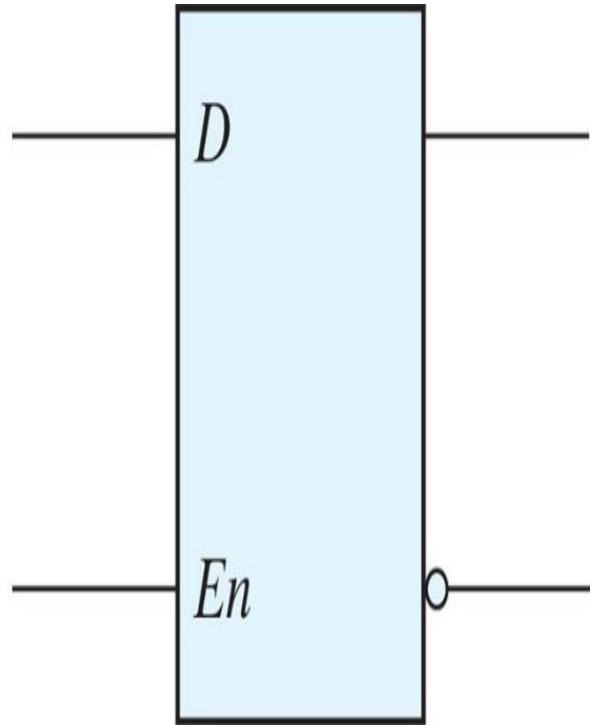
Graphic symbols for latches



SR



$\bar{S}\bar{R}$



D

Clock response in latch and flip-flop



(a) Response to positive level



(b) Positive-edge response



(c) Negative-edge response



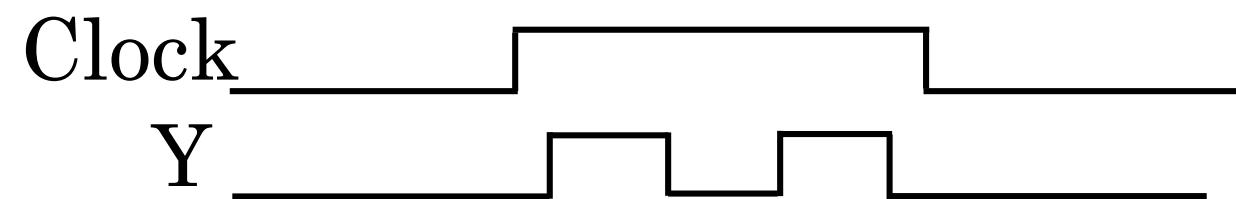
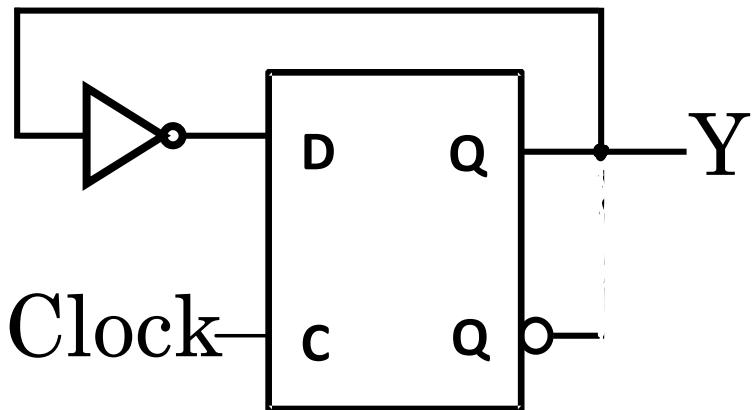
The Latch Timing Problem

- In a sequential circuit, paths may exist through combinational logic:
 - From one storage element to another
 - From a storage element back to the same storage element
- The combinational logic between a latch output and a latch input may be as simple as an interconnect
- For a clocked D-latch, the output Q depends on the input D whenever the clock input C has value 1

The Latch Timing Problem(continued)



- Consider the following circuit:



- As long as $C = 1$, the value of Y continues to change!
- The changes are based on the delay present on the loop through the connection from Y back to Y .
- This behavior is clearly unacceptable.
- Desired behavior: Y changes only once per clock pulse

The Latch Timing Problem (continued)



- A solution to the latch timing problem is to break the closed path from Y to Y within the storage element
- The commonly-used, path-breaking solutions replace the clocked D-latch with:
 - a master-slave flip-flop
 - an edge-triggered flip-flop

Reference

1. Mano book
2. ee.hawaii.edu/~sasaki/EE361/Fall06/Lab/7disp.html
3. https://www.tutorialspoint.com/computer_logical_organization/combinational_circuits.htm
4. https://www.electronics-tutorials.ws/sequential/seq_1.html

