Fetch decode execute cycle : class on 9th October

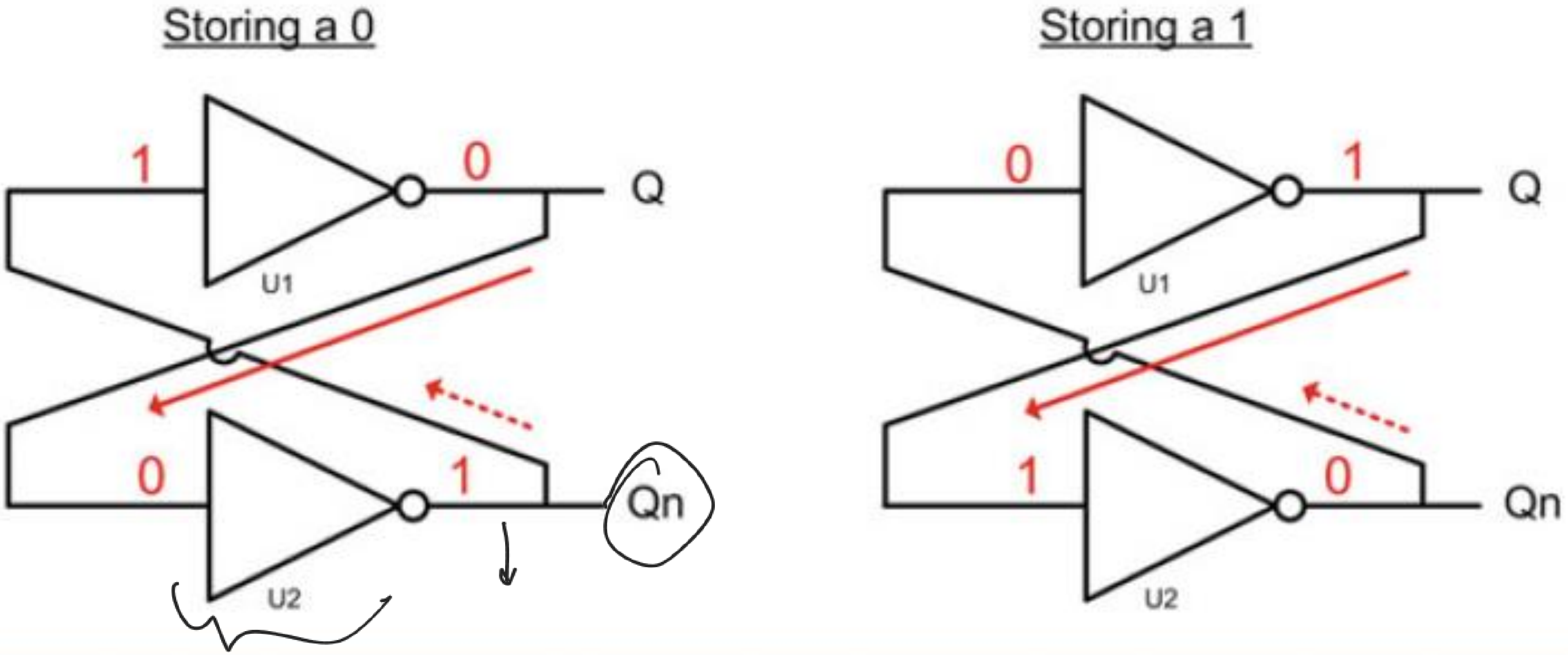
https://www.youtube.com/watch?v=Z5JC9Ve1sfI

"Introduction to Logic Circuits and Logic Design with Verilog", 3rd Ed, Springer (2024)

↑

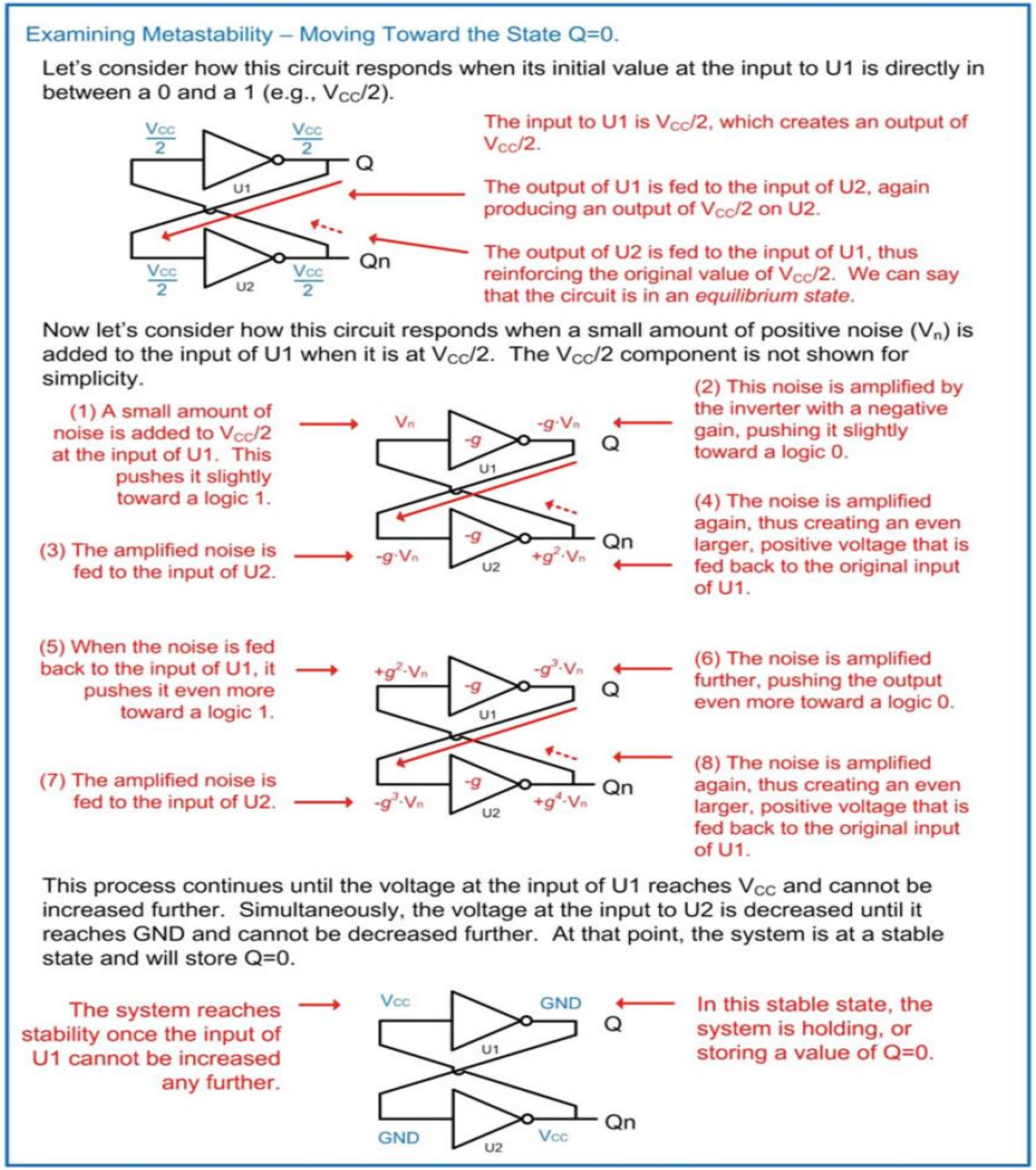
Slides acknowledgment: textbook by Brock J. LaMeres and http://www.youtube.com/@ComputerScienceLessons

Storage using a Cross-Coupled Inverter Pair



Storing a 0

Storing a 1



Examining Metastability – Moving Toward the State Q=0.

Let's consider how this circuit responds when its initial value at the input to U1 is directly in between a 0 and a 1 (e.g., $V_{CC}/2$).

$\frac{V_{CC}}{2}$
$\frac{V_{CC}}{2}$
Q
U1
Qn
$\frac{V_{CC}}{2}$
U2
$\frac{V_{CC}}{2}$

The input to U1 is $V_{CC}/2$, which creates an output of $V_{CC}/2$.

The output of U1 is fed to the input of U2, again producing an output of $V_{CC}/2$ on U2.

The output of U2 is fed to the input of U1, thus reinforcing the original value of $V_{CC}/2$. We can say that the circuit is in an *equilibrium state*.

Now let's consider how this circuit responds when a small amount of positive noise (V_n) is added to the input of U1 when it is at $V_{CC}/2$. The $V_{CC}/2$ component is not shown for simplicity.

(1) A small amount of noise is added to $V_{CC}/2$ at the input of U1. This pushes it slightly toward a logic 1.

(2) This noise is amplified by the inverter with a negative gain, pushing it slightly toward a logic 0.

(3) The amplified noise is fed to the input of U2.

(4) The noise is amplified again, thus creating an even larger, positive voltage that is fed back to the original input of U1.

(5) When the noise is fed back to the input of U1, it pushes it even more toward a logic 1.

(6) The noise is amplified further, pushing the output even more toward a logic 0.

(7) The amplified noise is fed to the input of U2.

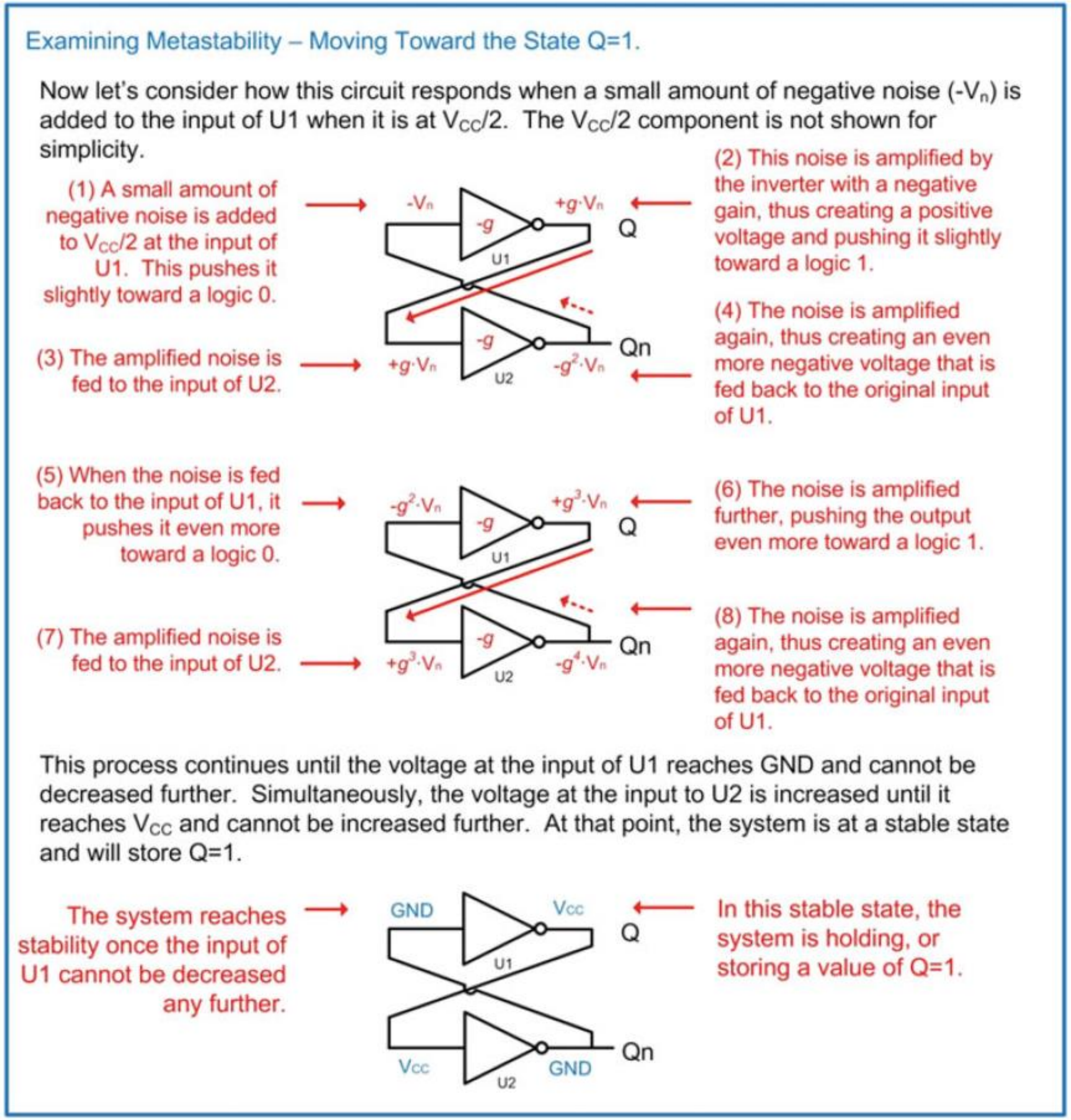
(8) The noise is amplified again, thus creating an even larger, positive voltage that is fed back to the original input of U1.

This process continues until the voltage at the input of U1 reaches V_{CC} and cannot be increased further. Simultaneously, the voltage at the input to U2 is decreased until it reaches GND and cannot be decreased further. At that point, the system is at a stable state and will store Q=0.

The system reaches stability once the input of U1 cannot be increased any further.

In this stable state, the system is holding, or storing a value of Q=0.

Ignore (not covered in class)



Examining Metastability – Moving Toward the State Q=1.

Now let's consider how this circuit responds when a small amount of negative noise ($-V_n$) is added to the input of U1 when it is at $V_{CC}/2$. The $V_{CC}/2$ component is not shown for simplicity.

(1) A small amount of negative noise is added to $V_{CC}/2$ at the input of U1. This pushes it slightly toward a logic 0.

(2) This noise is amplified by the inverter with a negative gain, thus creating a positive voltage and pushing it slightly toward a logic 1.

(3) The amplified noise is fed to the input of U2.

(4) The noise is amplified again, thus creating an even more negative voltage that is fed back to the original input of U1.

(5) When the noise is fed back to the input of U1, it pushes it even more toward a logic 0.

(6) The noise is amplified further, pushing the output even more toward a logic 1.

(7) The amplified noise is fed to the input of U2.

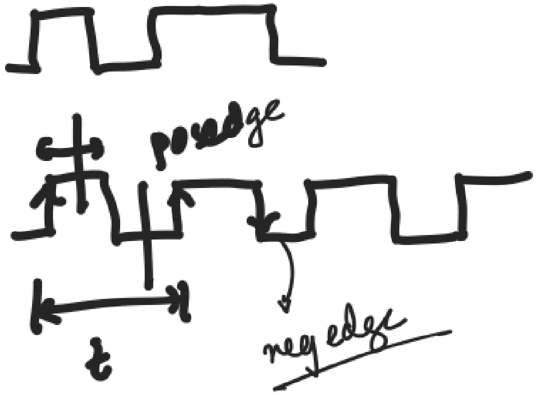
(8) The noise is amplified again, thus creating an even more negative voltage that is fed back to the original input of U1.

This process continues until the voltage at the input of U1 reaches GND and cannot be decreased further. Simultaneously, the voltage at the input to U2 is increased until it reaches V_{CC} and cannot be increased further. At that point, the system is at a stable state and will store Q=1.

The system reaches stability once the input of U1 cannot be decreased any further.

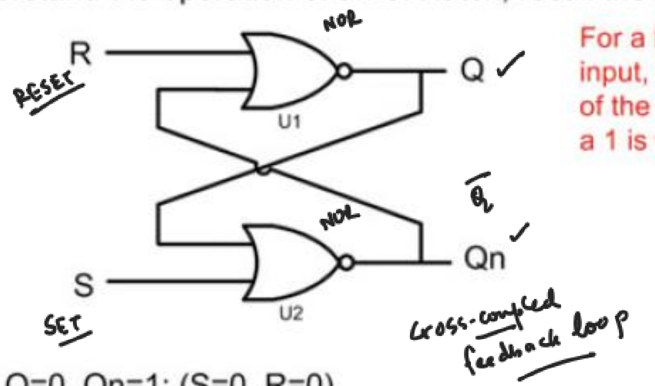
In this stable state, the system is holding, or storing a value of Q=1.

← Ignore (not covered in class).



SR Latch Behavior – Store State (S=0, R=0)

To understand the operation of an SR latch, recall the truth table for a NOR gate:

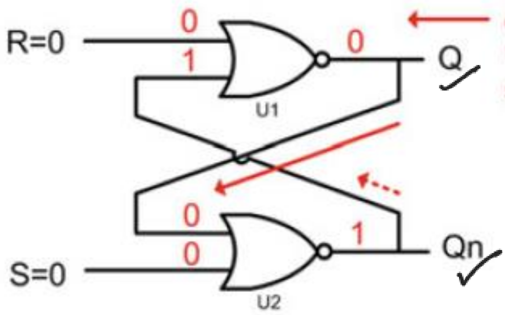


For a NOR gate, anytime there is a 1 on an input, the output is a 0 regardless of the value of the other input. The only time the output is a 1 is when both inputs are both 0's.

NOR

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

Storing Q=0, Qn=1: (S=0, R=0)

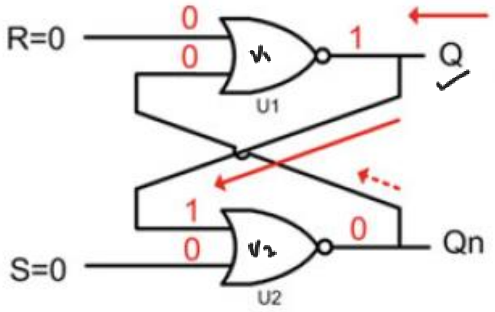


If Q starts at a 0, it will be fed back to U2 creating an output of Qn=1. This 1 will be fed back to the input of U1 creating an output of Q=0, thus reinforcing the initial state and storing Q=0, Qn=1.

NOR

A	B	Out	
0	0	1	(U2)
0	1	0	(U1)
1	0	0	
1	1	0	

Storing Q=1, Qn=0: (S=0, R=0)



If Q starts at a 1, it will be fed back to U2 creating an output of Qn=0. This 0 will be fed back to the input of U1 creating an output of Q=1, thus reinforcing the initial state and storing Q=1, Qn=0.

NOR

A	B	Out	
0	0	1	(U1)
0	1	0	
1	0	0	(U2)
1	1	0	

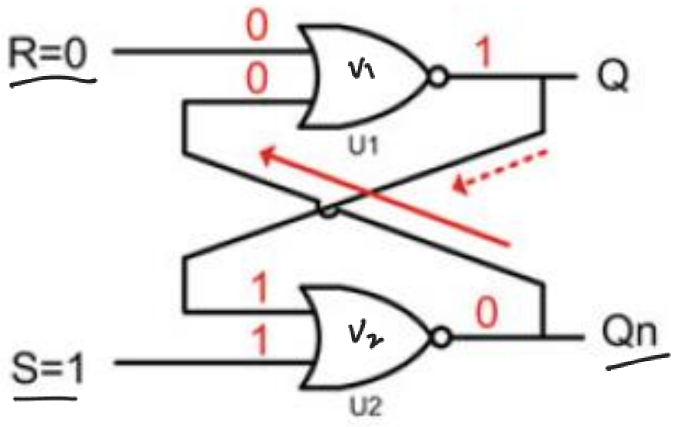
SR Latch Behavior – Set (S=1, R=0) and Reset (S=0, R=1) States

Setting Q=1: (S=1, R=0)

If S=1, it will force an output on U2 of Qn=0. This will be fed back to U1 creating an output of Q=1. This is fed back to U2 reinforcing the original output of Qn=0. This state will have outputs of Q=1, Qn=0.

NOR

A	B	Out	
0	0	1	(U1)
0	1	0	
1	0	0	
1	1	0	(U2)



S:1, R=0

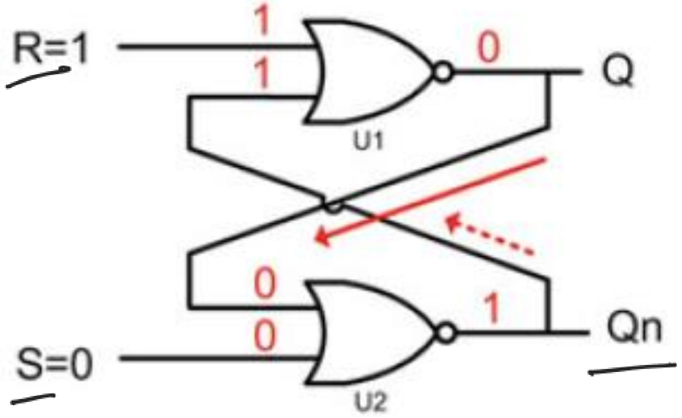
R	S	
0	0	hold
0	1	Q→1
1	0	Q→0

Resetting Q=0: (S=0, R=1)

If R=1, it will force an output on U1 of Q=0. This will be fed back to U2 creating an output of Qn=1. This is fed back to U1 reinforcing the original output of Q=0. This state will have outputs of Q=0, Qn=1.

NOR

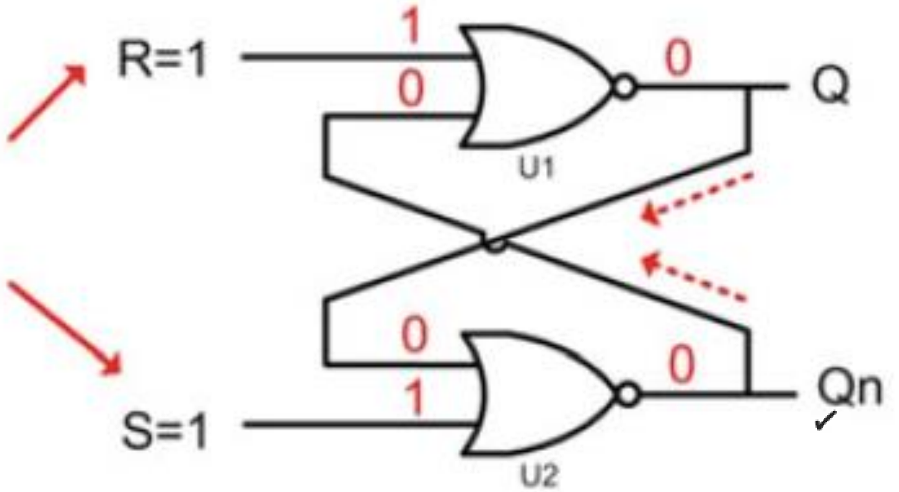
A	B	Out	
0	0	1	(U2)
0	1	0	
1	0	0	
1	1	0	(U1)



SR Latch Behavior – Don’t Use State (S=1, R=1)

S=1, R=1

When both S=1 and R=1, it forces the outputs of both U1 and U2 to 0. These 0's are fed back to the U2 and U1 but have no impact on the outputs. This input condition results in Q=0 and Qn=0.



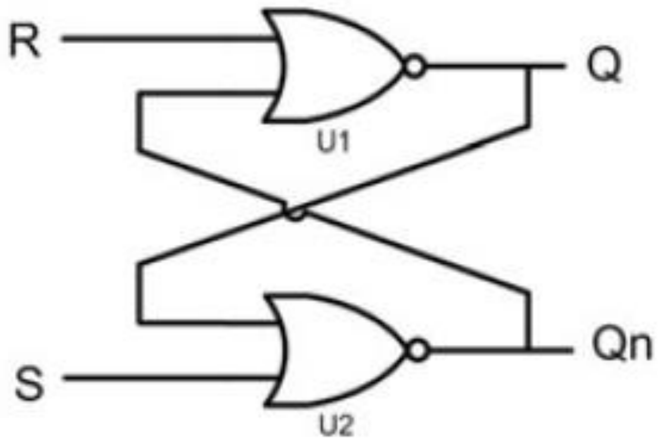
NOR

A	B	Out	
0	0	1	
0	1	0	(U2)
1	0	0	(U1)
1	1	0	

The problem with this state is that if the inputs are changed to the store state (S=0, R=0), the outputs will go metastable and then ultimately go to one of the two stable states (Q=0 or Q=1). The problem is that the final state is random and unknown.

SR Latch Truth Table

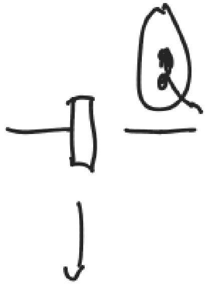
The following is the final truth table for the SR Latch.



S=1

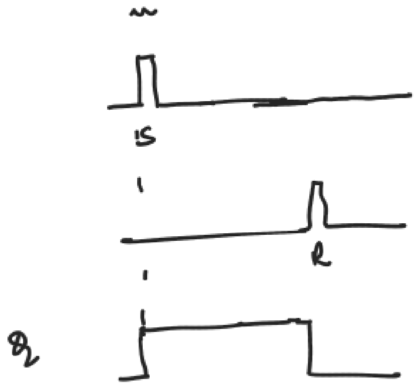
S	R	Q	Qn	
0	0	Last Q	Last Qn	Hold or Store
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Don't Use

electrical

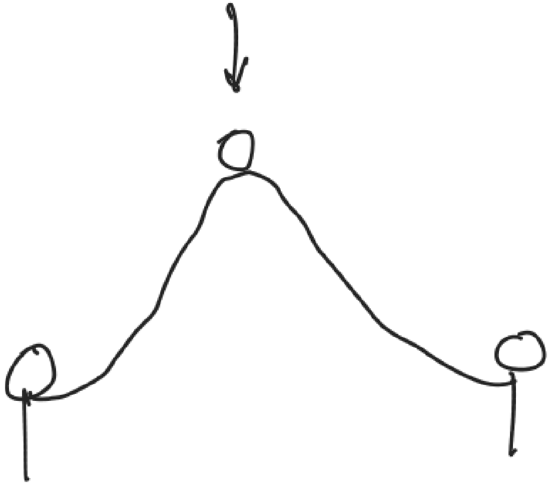


mechanical

bouncing



bistable



S'R' Latch Behavior – Store State (S'=1, R'=1)

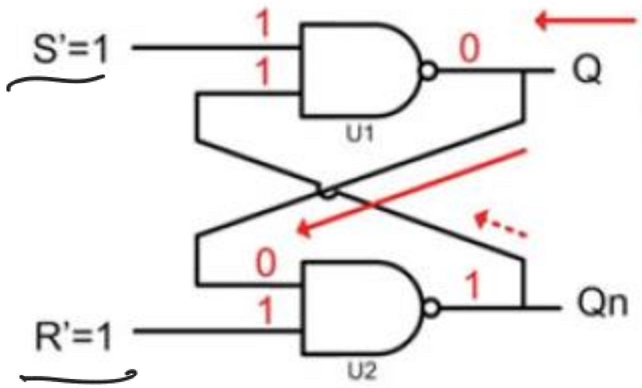
To understand the operation of an SR latch, recall the truth table for a NAND gate:

For a NAND gate, anytime there is a 0 on an input, the output is a 1 regardless of the value of the other input. The only time the output is a 0 is when both inputs are both 1's.

NAND

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

Storing Q=0, Qn=1: (S'=1, R'=1)

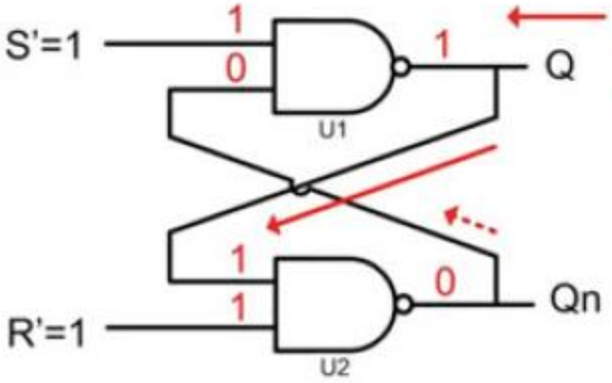


If Q starts at a 0, it will be fed back to U2 creating an output of Qn=1. This 1 will be fed back to the input of U1 creating an output of Q=0, thus reinforcing the initial state and storing Q=0, Qn=1.

NAND

A	B	Out	
0	0	1	
0	1	1	(U2)
1	0	1	
1	1	0	(U1)

Storing Q=1, Qn=0: (S'=1, R'=1)



If Q starts at a 1, it will be fed back to U2 creating an output of Qn=0. This 0 will be fed back to the input of U1 creating an output of Q=1, thus reinforcing the initial state and storing Q=1, Qn=0.

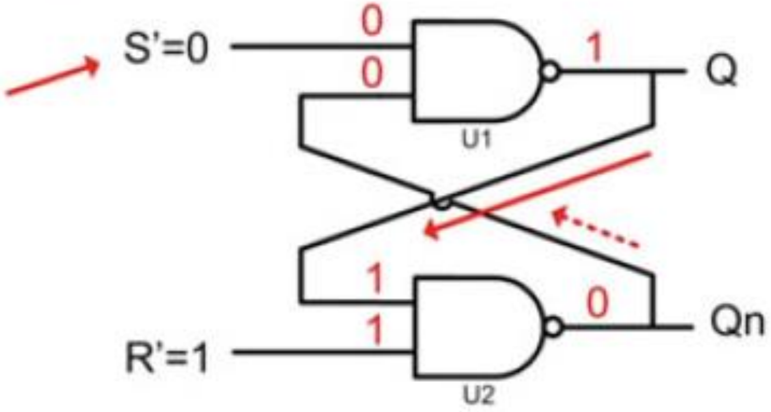
NAND

A	B	Out	
0	0	1	
0	1	1	
1	0	1	(U1)
1	1	0	(U2)

S'R' Latch Behavior – Set (S'=0, R'=1) and Reset (S'=1, R'=0) States

Setting Q=1: (S'=0, R'=1)

If S'=0, it will force an output on U1 of Q1=1. This will be fed back to U2 creating an output of Qn=0. This is fed back to U1 reinforcing the original output of Q=1. This state will have outputs of Q=1, Qn=0.

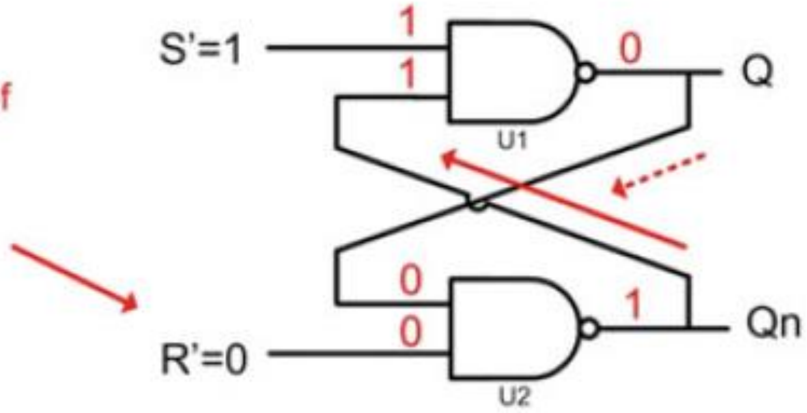


NAND

A	B	Out	
0	0	1	(U1)
0	1	1	
1	0	1	
1	1	0	(U2)

Resetting Q=0: (S'=1, R'=0)

If R'=0, it will force an output on U2 of Qn=1. This will be fed back to U1 creating an output of Q=0. This is fed back to U2 reinforcing the original output of Qn=1. This state will have outputs of Q=0, Qn=1.



NAND

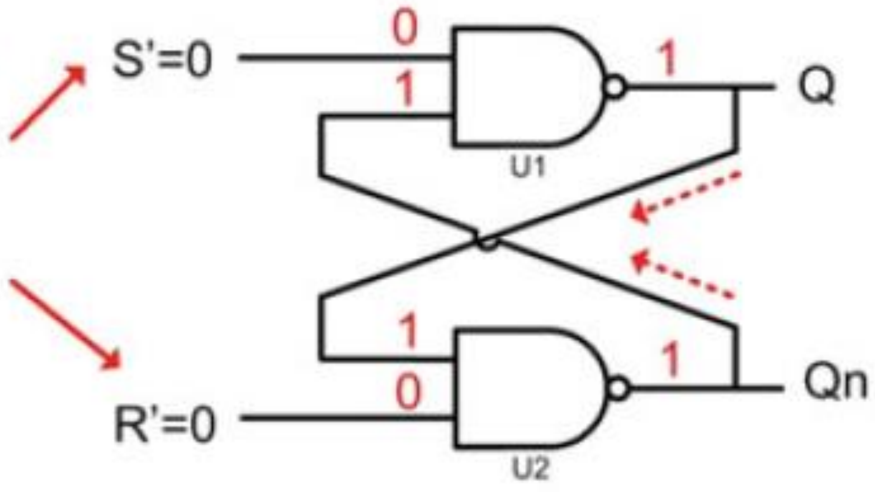
A	B	Out	
0	0	1	(U2)
0	1	1	
1	0	1	
1	1	0	(U1)

S'R' Latch Behavior – Don't Use State (S'=0, R'=0)

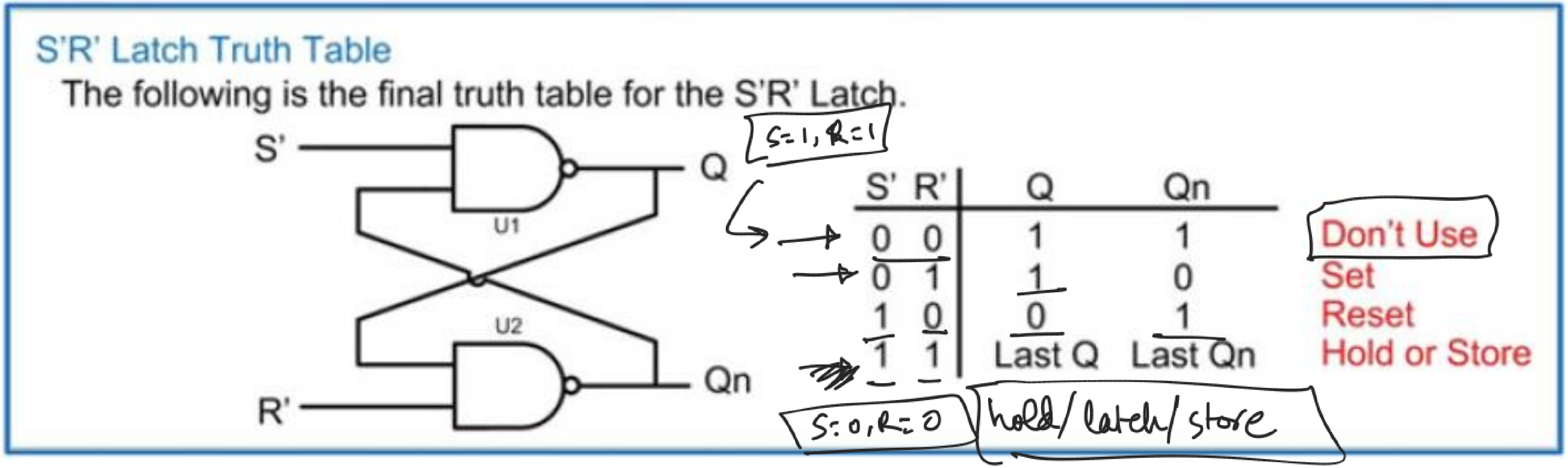
S'=0, R'=0

When both S'=0 and R'=0, it forces the outputs of both U1 and U2 to 1. These 1's are fed back to the U2 and U1 but have no impact on the outputs. This input condition results in Q=1 and Qn=1.

	A	B	Out	
	0	0	1	
NAND	0	1	1	(U1)
	1	0	1	(U2)
	1	1	0	



Again, the problem with this state is that if the inputs are changed to the store state (S'=1, R'=1), the outputs will go metastable and then ultimately go to one of the two stable states (Q=0 or Q=1). The final state is random and unknown.



S'R' Latch Truth Table

The following is the final truth table for the S'R' Latch.

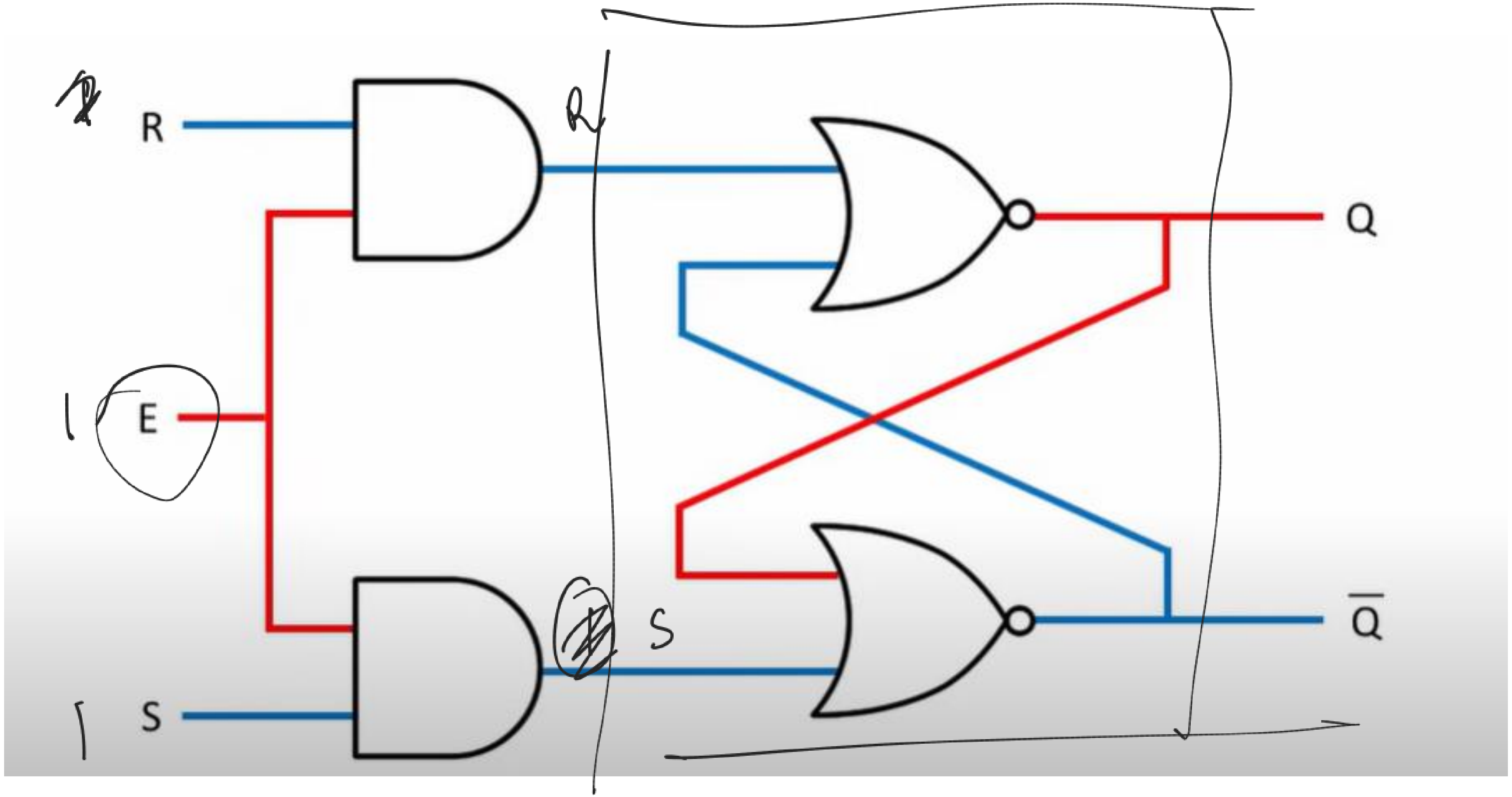
S=1, R=1

S'	R'	Q	Qn	
0	0	1	1	Don't Use
0	1	1	0	Set
1	0	0	1	Reset
1	1	Last Q	Last Qn	Hold or Store

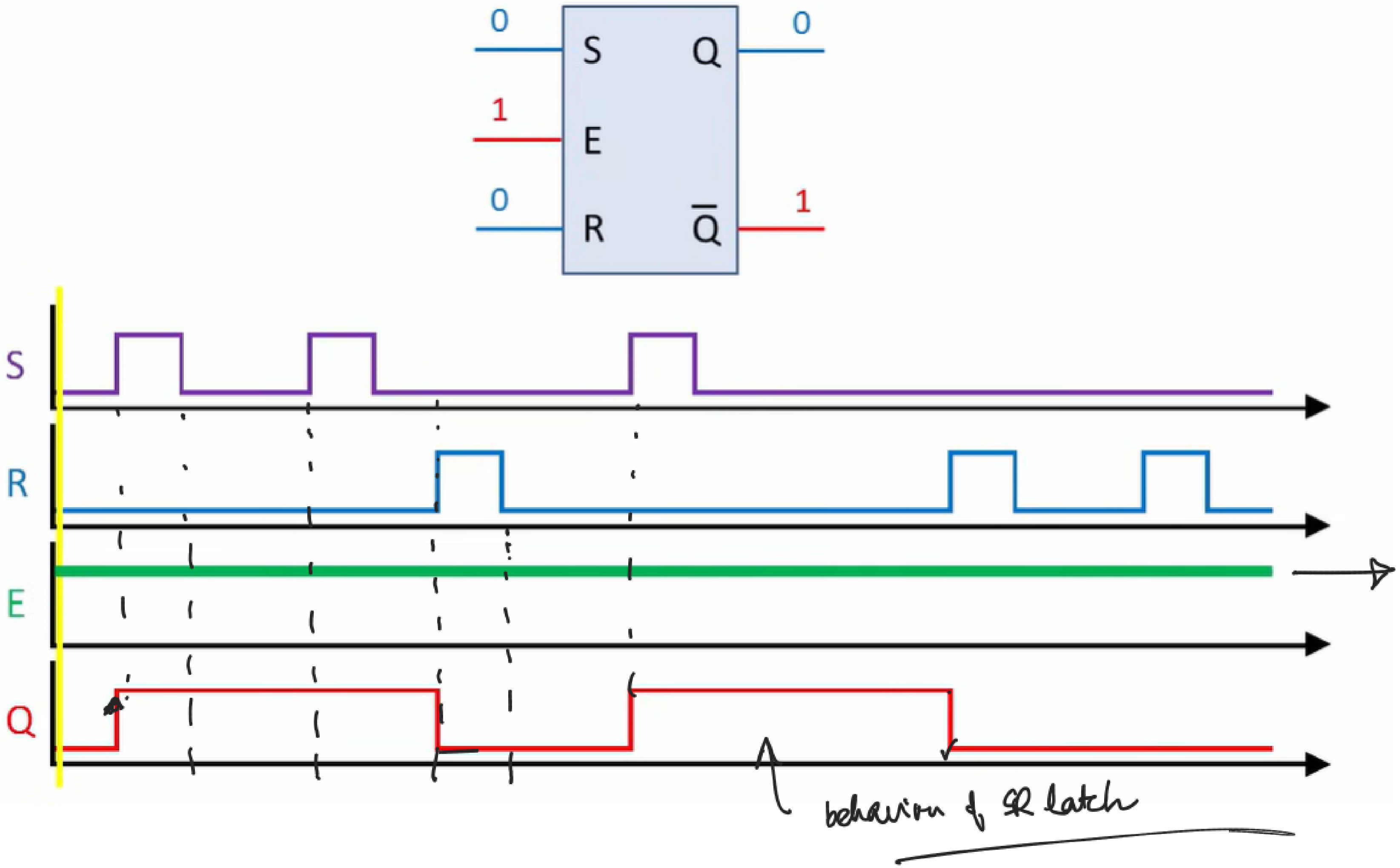
S=0, R=0

hold/ latch/ store

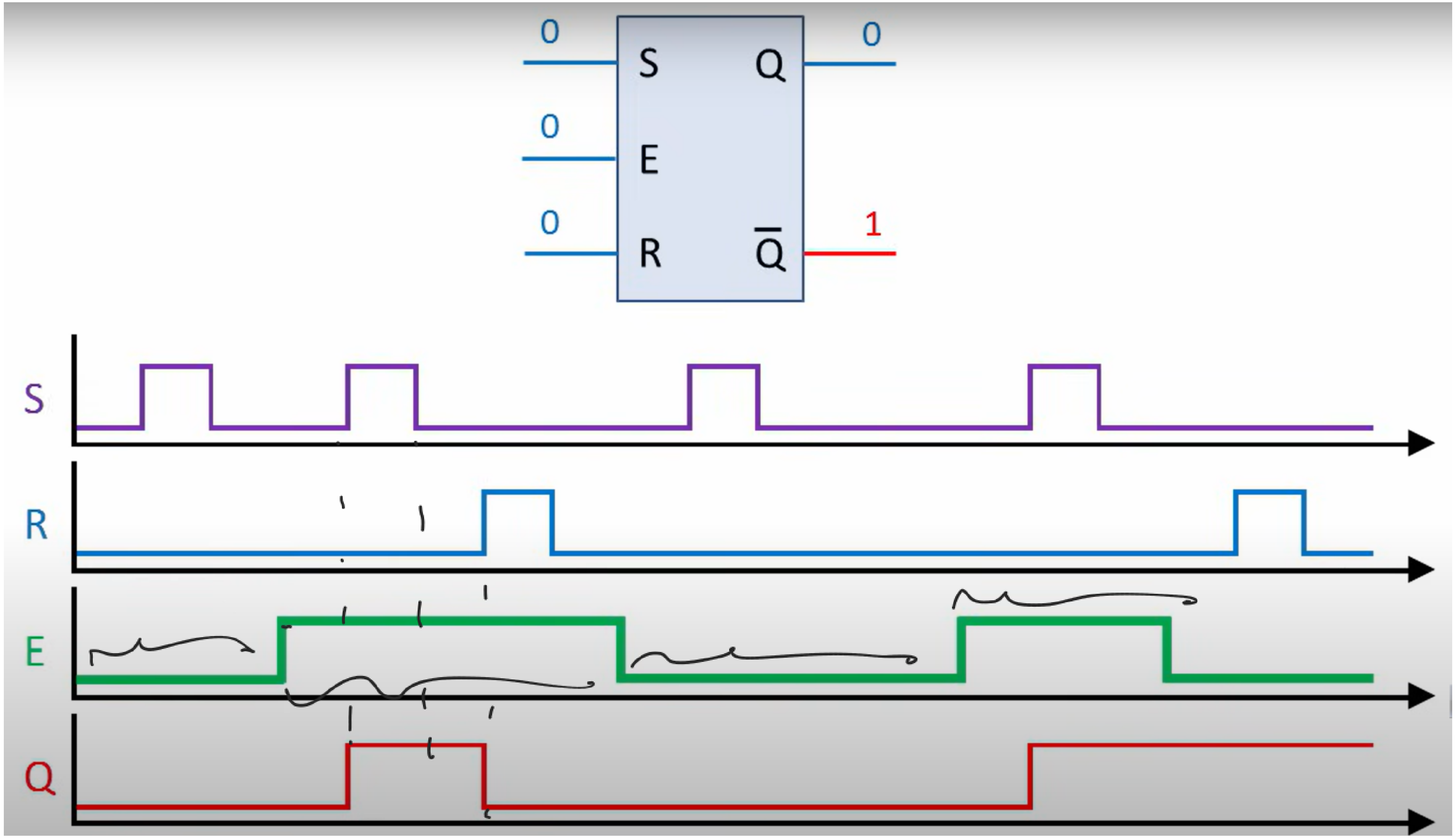
SR Latch with Enable



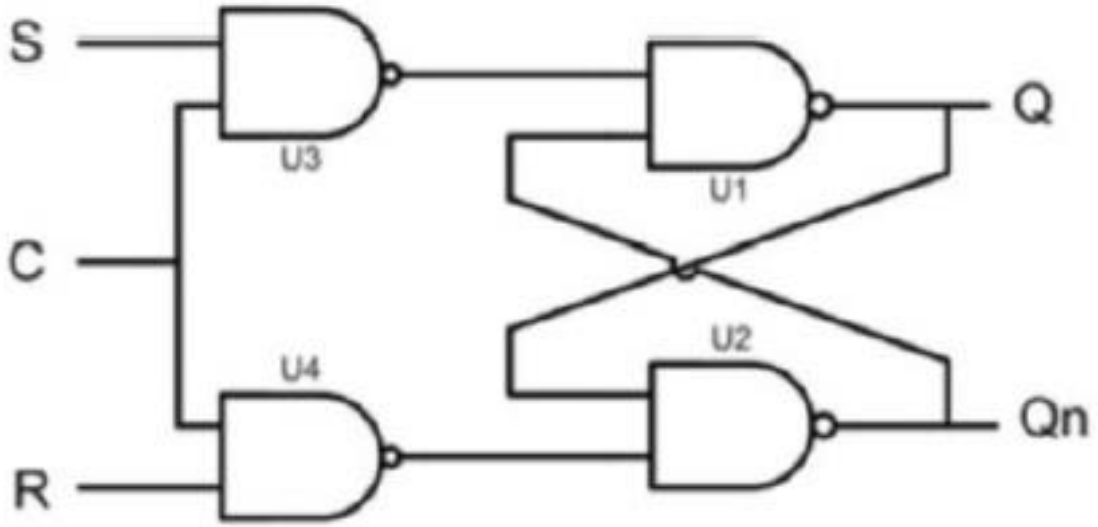
SR Latch with Enable: Timing Diagram



SR Latch with Enable/Control: Timing Diagram



SR Latch with Enable



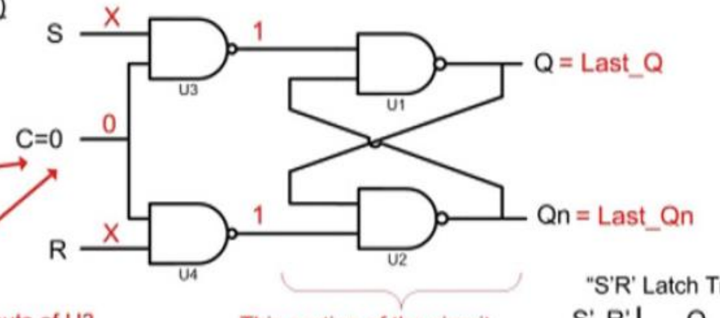
~~C~~ E	S	R	Q	Qn	
0	X	X	Last Q	Last Qn	Store
1	0	0	Last Q	Last Qn	Don't Use
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	1	1	Don't Use

SR Latch with Enable Behavior

Store State (C=0)

NAND

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0



When C=0, the outputs of U3 and U4 are 1's, regardless of the values of S and R.

This portion of the circuit is an S'R' Latch. With inputs of S'=1 and R'=1, the S'R' Latch is put into "store" mode.

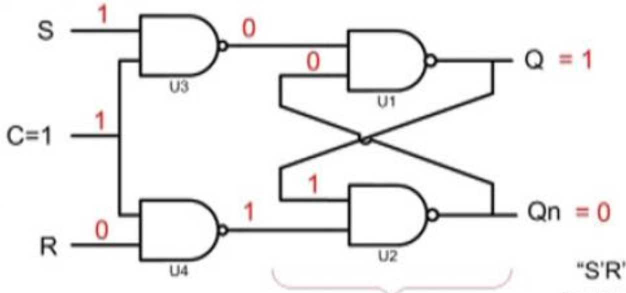
"S'R' Latch Truth Table"

S'	R'	Q	Qn
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Last Q	Last Qn

Setting Q=1: (S=1, R=0, C=1)

NAND

A	B	Out	
0	0	1	
0	1	1	
1	0	1	(U4)
1	1	0	(U3)



"S'R' Latch" →

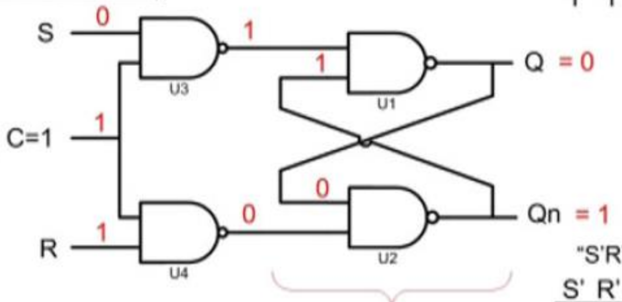
"S'R' Latch Truth Table"

S'	R'	Q	Qn
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Last Q	Last Qn

Resetting Q=0: (S=0, R=1, C=1)

NAND

A	B	Out	
0	0	1	
0	1	1	(U3)
1	0	1	
1	1	0	(U4)

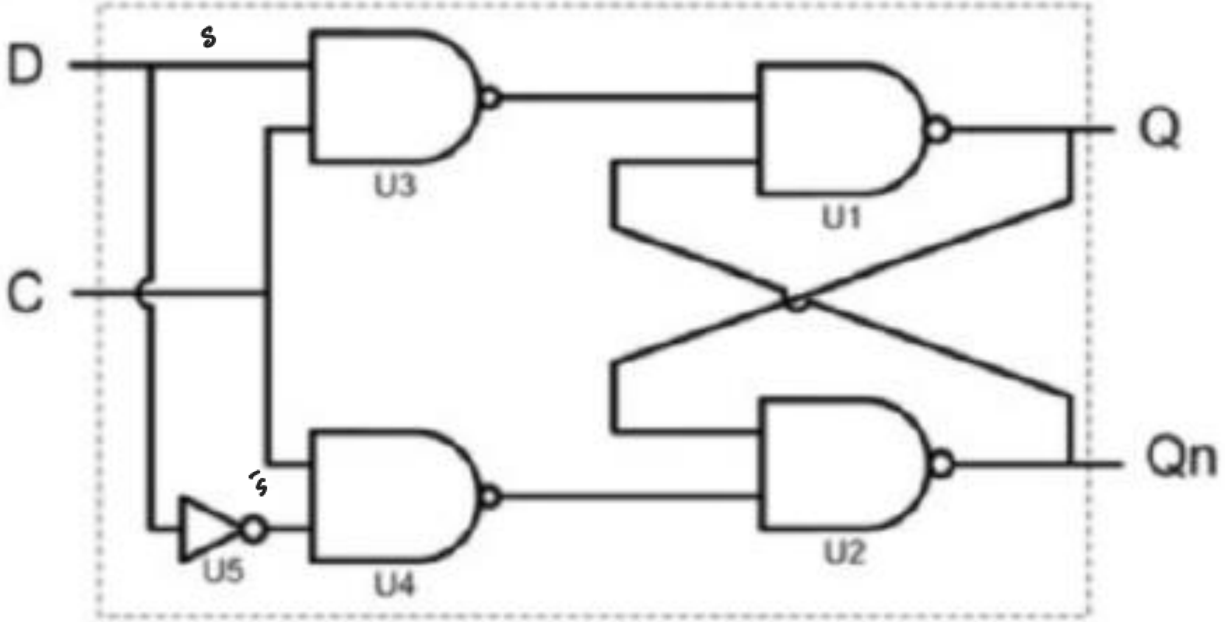


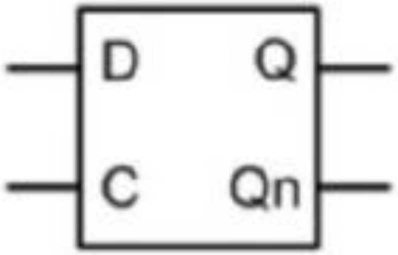
"S'R' Latch" →

"S'R' Latch Truth Table"

S'	R'	Q	Qn
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Last Q	Last Qn

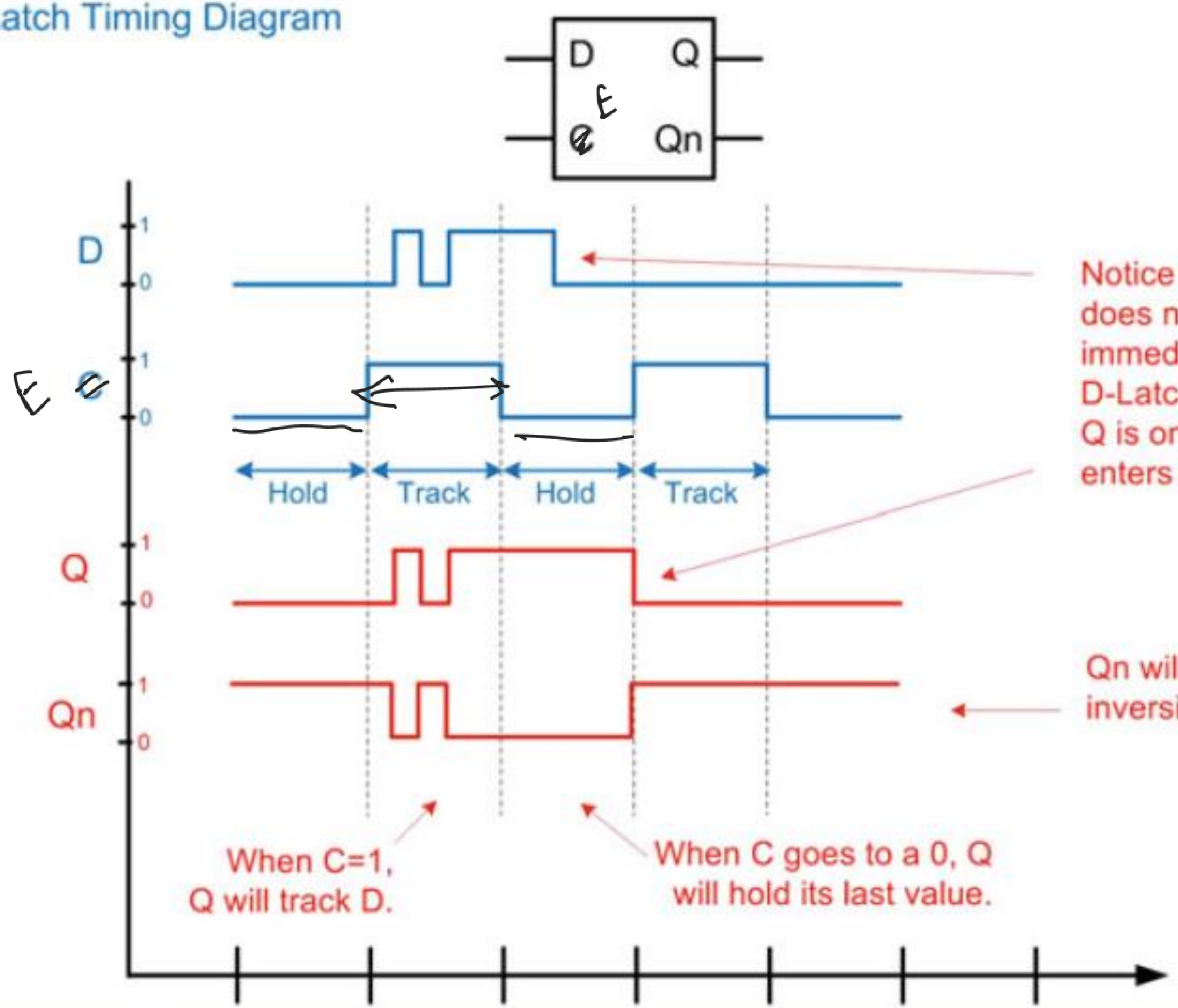
D-Latch Schematic, Symbol and Truth Table





C	D	Q	Qn	
0	X	Last Q	Last Qn	Store
1	0	0	1	Track
1	1	1	0	Track

D-Latch Timing Diagram



Notice this transition does not impact Q immediately because the D-Latch is in hold mode. Q is only updated once it enters track mode.

Qn will always be the inversion of Q.

When C=1, Q will track D.

When C goes to a 0, Q will hold its last value.