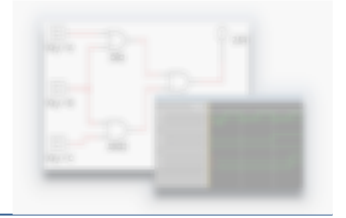


ECE119 – Ψηφιακή Σχεδίαση

Διδάσκοντες Εργαστηρίου: Δ. Καραμπερόπουλος
Δ. Γαρυφάλλου

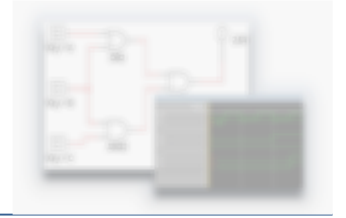
➤ Lab 8: Latches and Sequential Logic Circuits

Περιεχόμενα Εργαστηριακού Μαθήματος



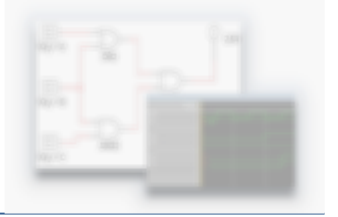
- Εισαγωγή
- Lab 1: Multisim Circuit Simulation and Basic Gates
- Lab 2: Truth Tables and Basic Logic Gates
- Lab 3: Logic Gates Explored and Boolean Algebra
- Lab 4: Karnaugh Maps
- Lab 5: Binary Conversion and Adders
- Lab 6: Encoders and Decoders
- Lab 7: Multiplexers and Demultiplexers
- **Lab 8: Latches and Sequential Logic Circuits**
- Lab 9: Flip-Flops
- Lab 10: Sequential Circuits - FSM

Latches (Μανδαλωτές) and Sequential Logic Circuits



- In all previous labs, we were dealing with **combinational logic circuits**.
- A combinational logic circuit is one such that all outputs may be determined from the current inputs.
- In this lab, we will introduce **sequential logic circuits** through the application of latches.
- A sequential logic circuit is one such that outputs depend on the previous inputs in addition to the current inputs.
- We will also look at how clock signals can be implemented into a circuit.

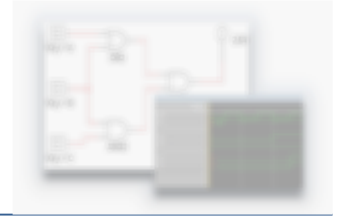
Learning Objectives



In this lab, students will:

- Understand the difference between **synchronous** and **asynchronous** sequential circuits.
- Test and compare circuits for **D latches** using both logic gates and latches.
- Confirm the characteristic table of a **gated SR latch**.
- Observe and articulate the difference between D latches and SR latches

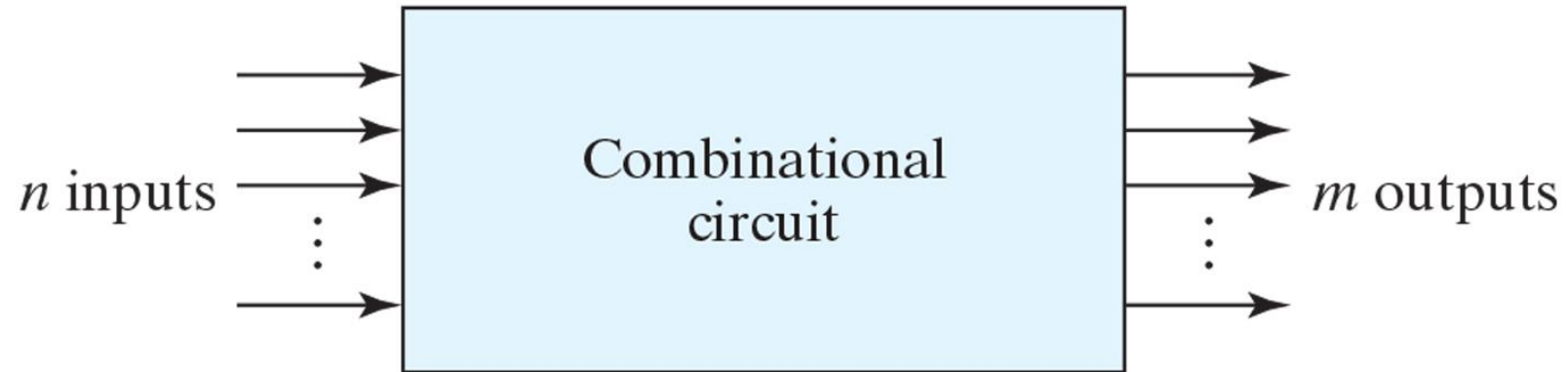
Expected Deliverables



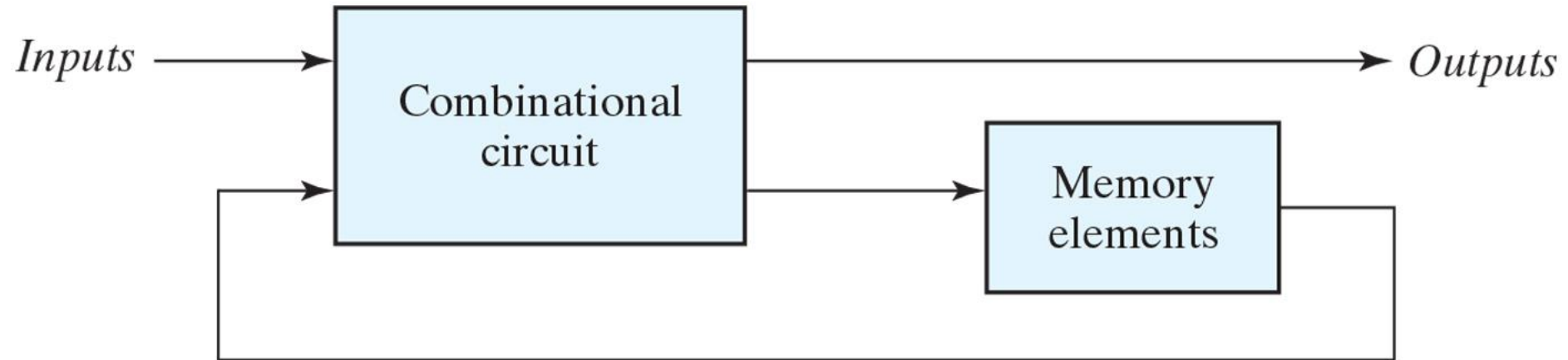
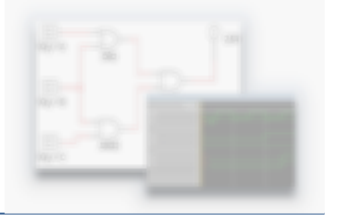
In this lab, you will collect the following deliverables:

- Latch features table
- Latch comparisons
- Observations of probe behavior
- Conclusion questions

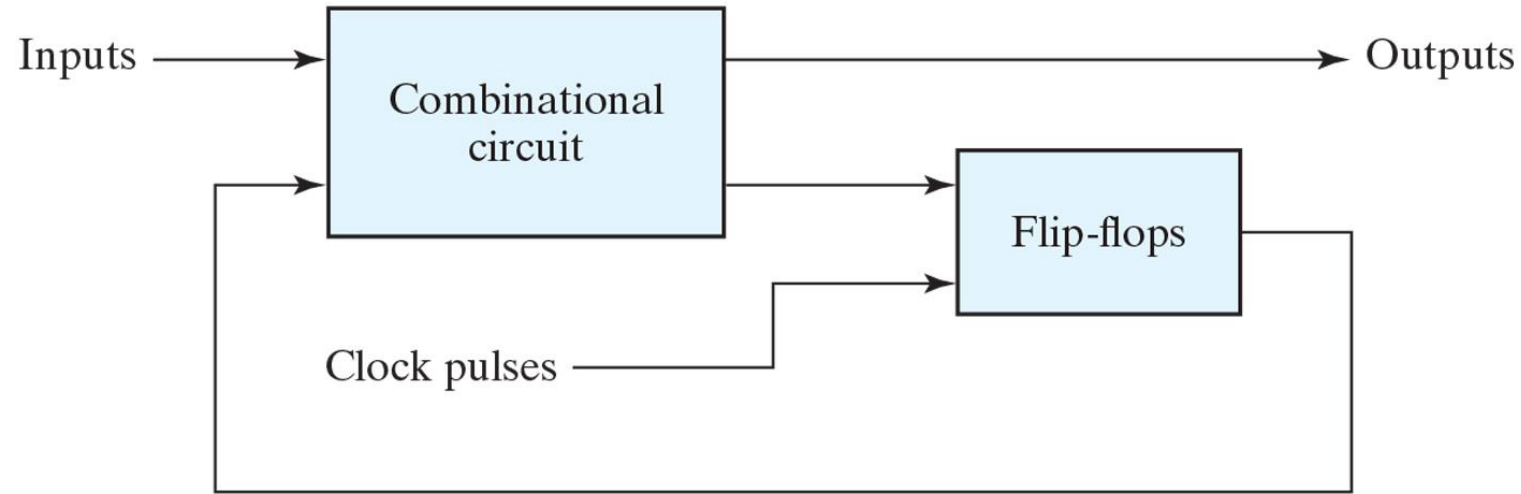
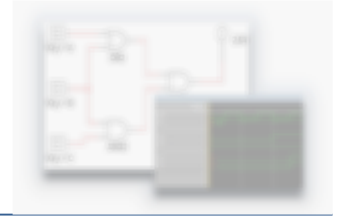
Συνδυαστικό κύκλωμα



Ακολουθιακό κύκλωμα



Σύγχρονο Ακολουθιακό κύκλωμα με ρολόι

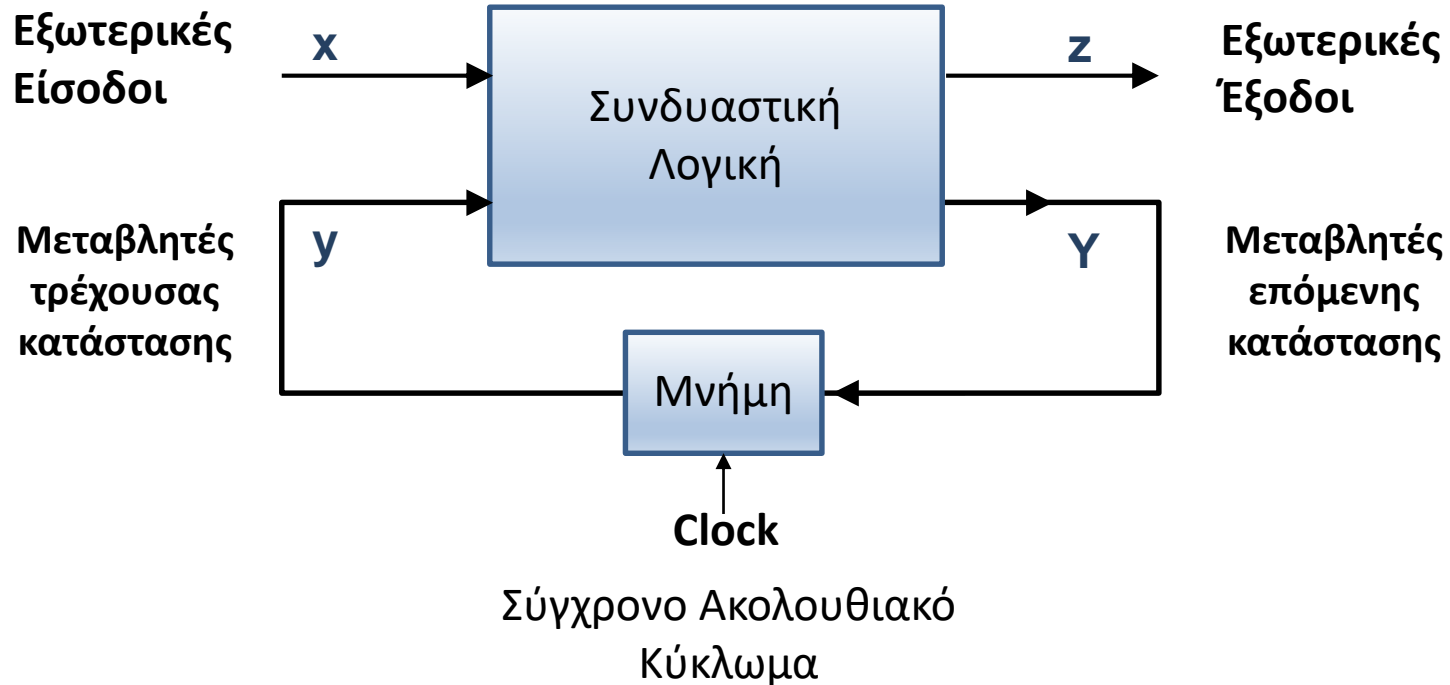
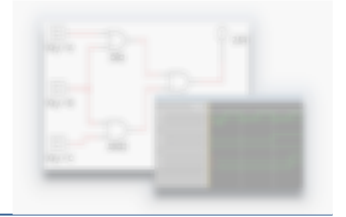


(a) Block diagram



(b) Timing diagram of clock pulses

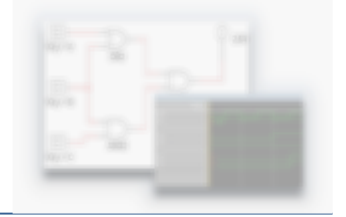
Σύγχρονο Ακολουθιακό κύκλωμα με ρολόι



Εξισώσεις Εξόδων

$$z = f(x, y)$$
$$Y = g(x, y)$$

- Στα ακολουθιακά κυκλώματα η έξοδος z δεν είναι συνάρτηση μόνο της εισόδου x αλλά και της παρούσης κατάστασης y του κυκλώματος πριν την εφαρμογή του clock. Το ίδιο συμβαίνει μια με τις μεταβλητές επόμενης κατάστασης.
- Τα ακολουθιακά κυκλώματα «θυμούνται προηγούμενες τιμές» μέσω της σύνδεσης της ανάδρασης.



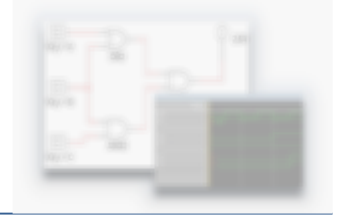
Κύριες κατηγορίες των ακολουθιακών κυκλωμάτων

- **Ασύγχρονα:** Αλλάζουν κατάσταση σύμφωνα με τις αλλαγές των εισόδων τους. Απαιτούνται ειδικές τεχνικές σχεδιασμού.
- **Σύγχρονα:** Τα σήματα ανάδρασης διακόπτονται από καταχωρητές που σκανδαλίζονται από παλμούς ρολογιού.
- Συνεπώς η κατάστασή του κυκλώματος αλλάζει σύμφωνα με τους **παλμούς του ρολογιού**.
- Η κατάσταση του κυκλώματος ορίζεται από το περιεχόμενο των στοιχείων της μνήμης.
- Ένα σύγχρονο ακολουθιακό κύκλωμα πρέπει εξ'ορισμού να χρησιμοποιεί σήματα συγχρονισμού.
- Τα σήματα συγχρονισμού επηρεάζουν τα στοιχεία μνήμης του κυκλώματος, ώστε αυτά να **αλλάζουν κατάσταση σε διακριτές χρονικές στιγμές**
- Ο συγχρονισμός επιτυγχάνεται μέσω μιας «γεννήτριας κύριου-ρολογιού» η οποία τροφοδοτεί το σύστημα με μία περιοδική σειρά “παλμών ρολογιού”
- Τα στοιχεία μνήμης μπορούν να αλλάξουν κατάσταση μόνο κατά την έλευση παλμών ρολογιού



Το Latch (Μανδαλωτής) και το Flip-Flop ως στοιχεία μνήμης

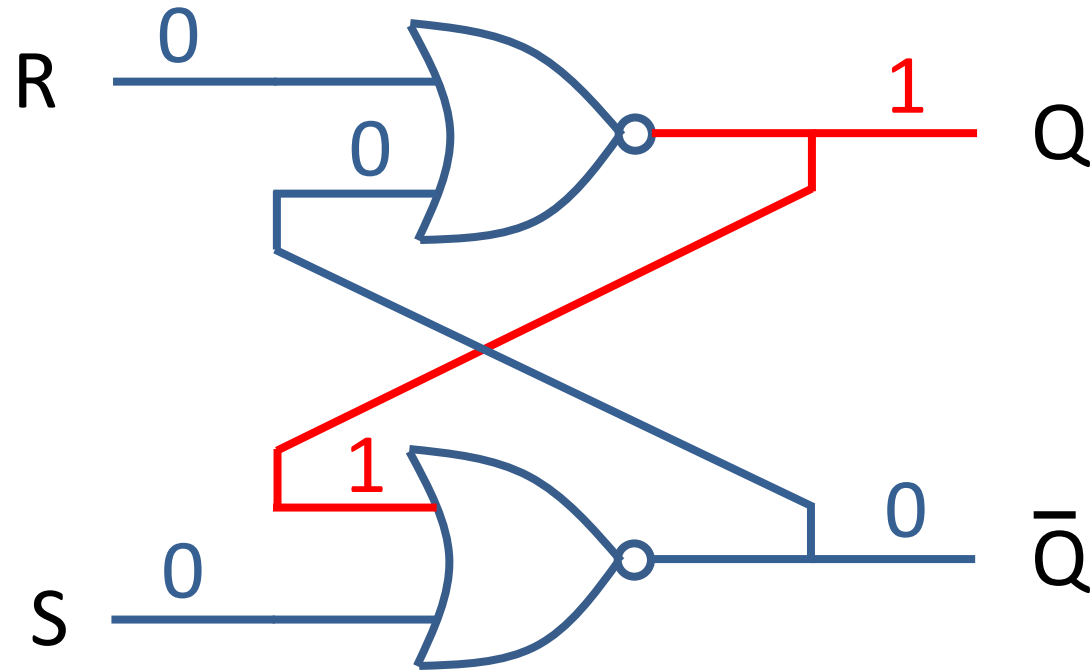
- Βασικά στοιχεία των ψηφιακών κυκλωμάτων είναι οι πύλες. Το χαρακτηριστικό των πυλών είναι ότι η έξοδός τους σε κάποια χρονική στιγμή εξαρτάται αποκλειστικά από την είσοδό τους την συγκεκριμένη χρονική στιγμή και όχι από προηγούμενες καταστάσεις τους. Δηλαδή οι πύλες δεν έχουν μνήμη.
- Αντίθετα, **τα Latches και τα flip-flops είναι τα βασικά στοιχεία μνήμης** τα οποία μπορούν να αποθηκεύσουν μία δυαδική πληροφορία. Η πληροφορία αυτή που είναι το “1” ή το “0” παραμένει σταθερή μέχρις ότου το flip-flop να ξαναδιεγερθεί.
- Ένα κύκλωμα Latch ή flip-flop μπορεί να διατηρηθεί σε μία δυαδική κατάσταση επ’ αόριστον, έως ότου κάποιο σήμα προκαλέσει αλλαγή κατάστασης.



Το Latch (Μανδαλωτής) και το Flip-Flop ως στοιχεία μνήμης

- Τα στοιχεία μνήμης που ενεργοποιούνται από την παρουσία συγκεκριμένων επιπέδων (συγκεκριμένων τιμών 0 ή 1) του σήματος ρολογιού ονομάζονται **μανδαλωτές (latches)**, ενώ εκείνα που ενεργοποιούνται από μεταβάσεις τιμών του σήματος ρολογιού (clock transitions) ονομάζονται **flip-flop**
- Τα latches είναι διατάξεις ευαίσθητες στο επίπεδο σήματος ρολογιού.
- Τα flip-flop είναι διατάξεις ευαίσθητες στις ακμές του σήματος ρολογιού ή ακμοπυροδότητες διατάξεις.
- Το latch είναι το βασικό κύκλωμα κατασκευής ενός flip-flop.

SR Latch

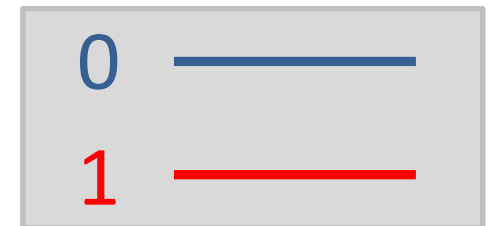




SR Latch



- Στοιχείο μνήμης 1-bit
- Κατάσταση θέσης (set state):
 - $R = 0$
 - $S = 1$

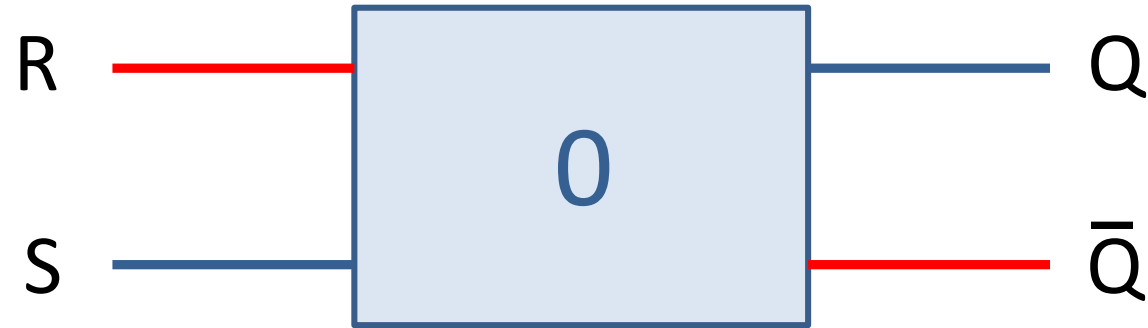


SR Latch



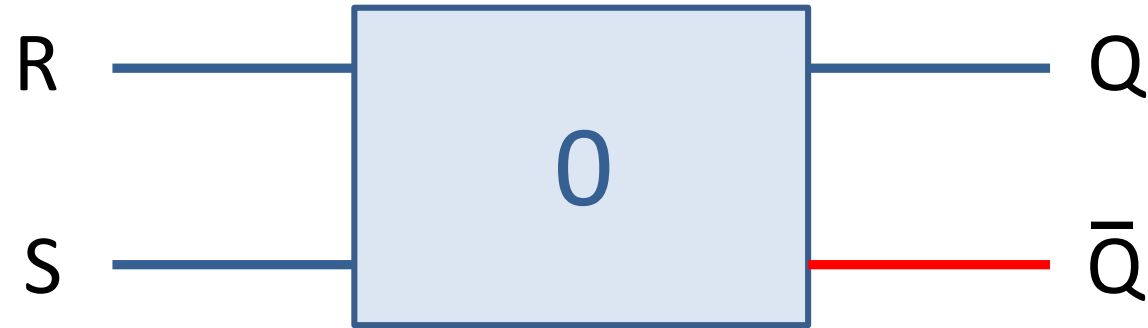
- Κατάσταση αναμονής (μνήμη):
 - $R = 0$
 - $S = 0$

SR Latch



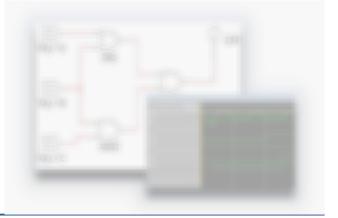
- Κατάσταση επαναφοράς ή μηδενισμού (reset state):
 - $R = 1$
 - $S = 0$

SR Latch

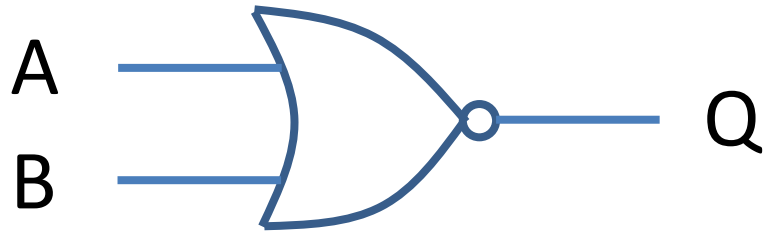


- Κατάσταση αναμονής (μνήμη):
 - $R = 0$
 - $S = 0$

SR Latch

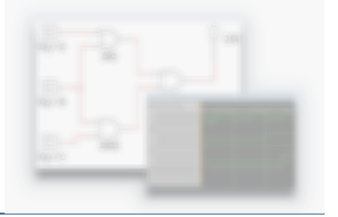


NOR



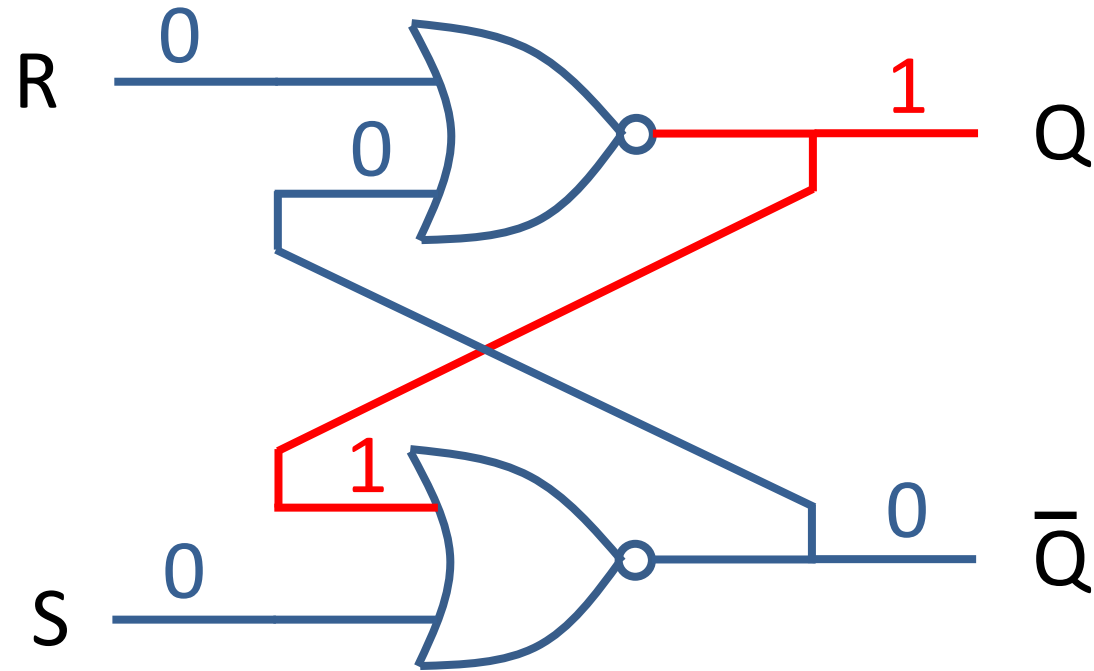
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

SR Latch

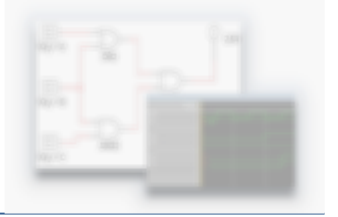


➤ Normal state

S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0



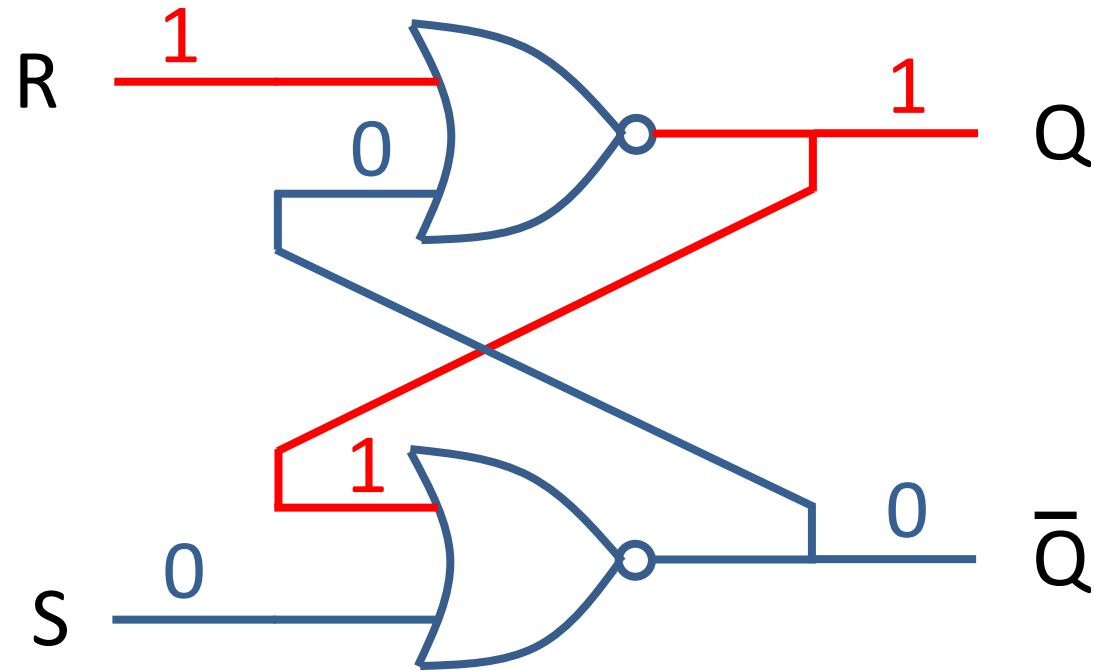
SR Latch



➤ Reset state (1/3)



S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0



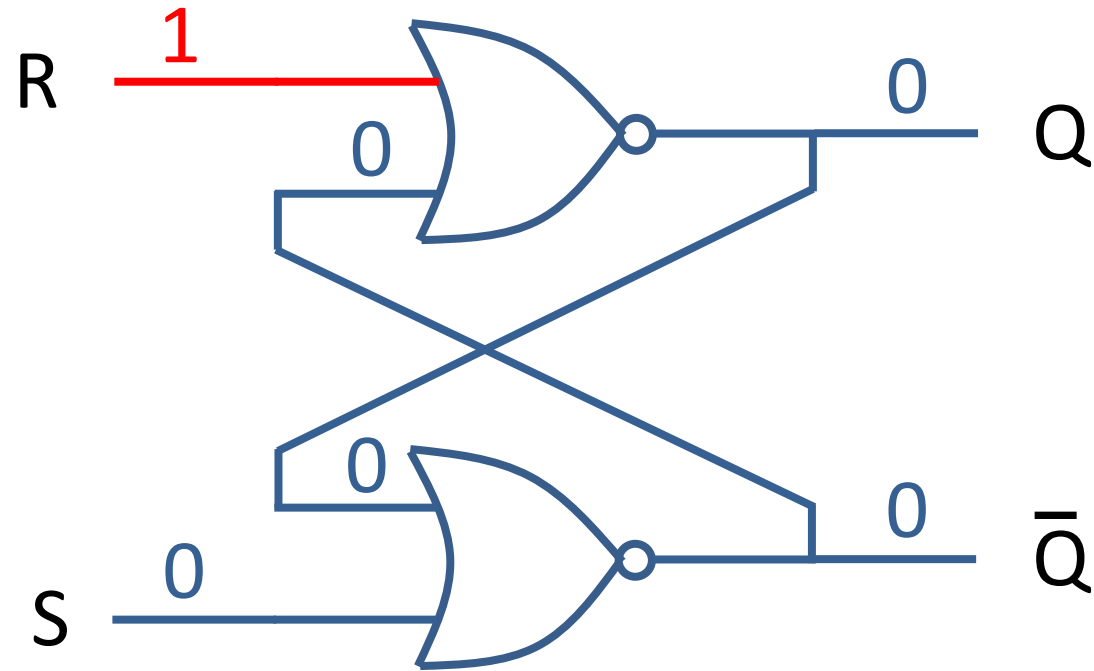
SR Latch



➤ Reset state (2/3)



S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0



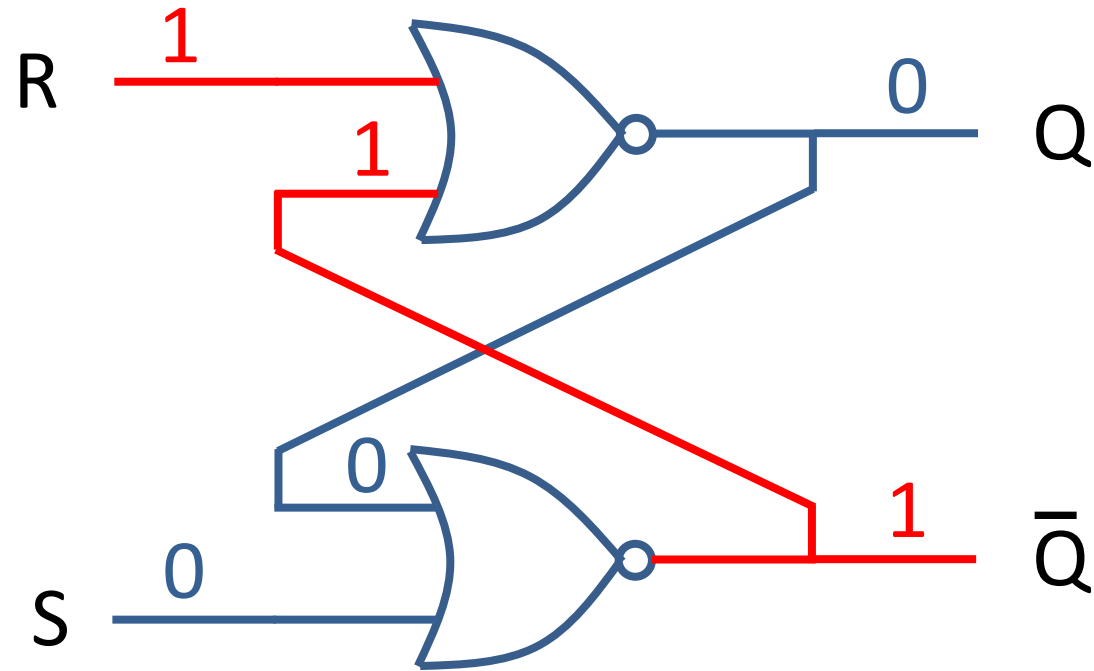
SR Latch



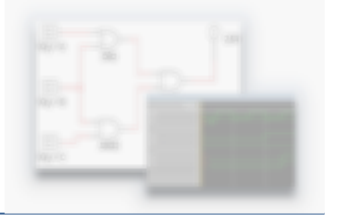
> Reset state (3/3)



S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0

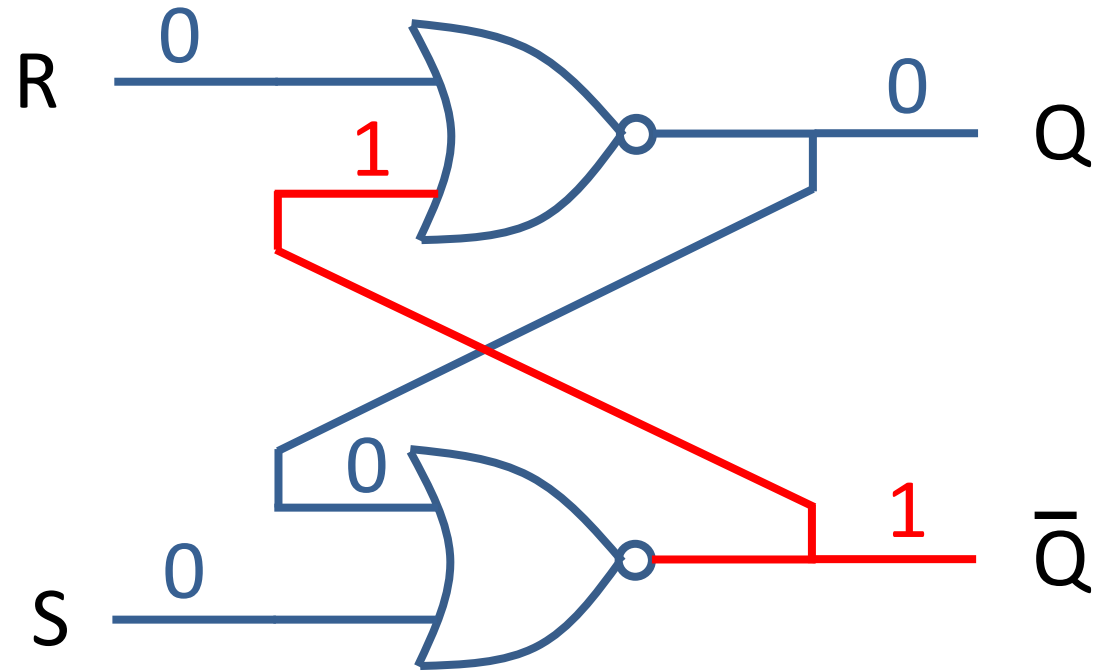


SR Latch



➤ Normal state

S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0

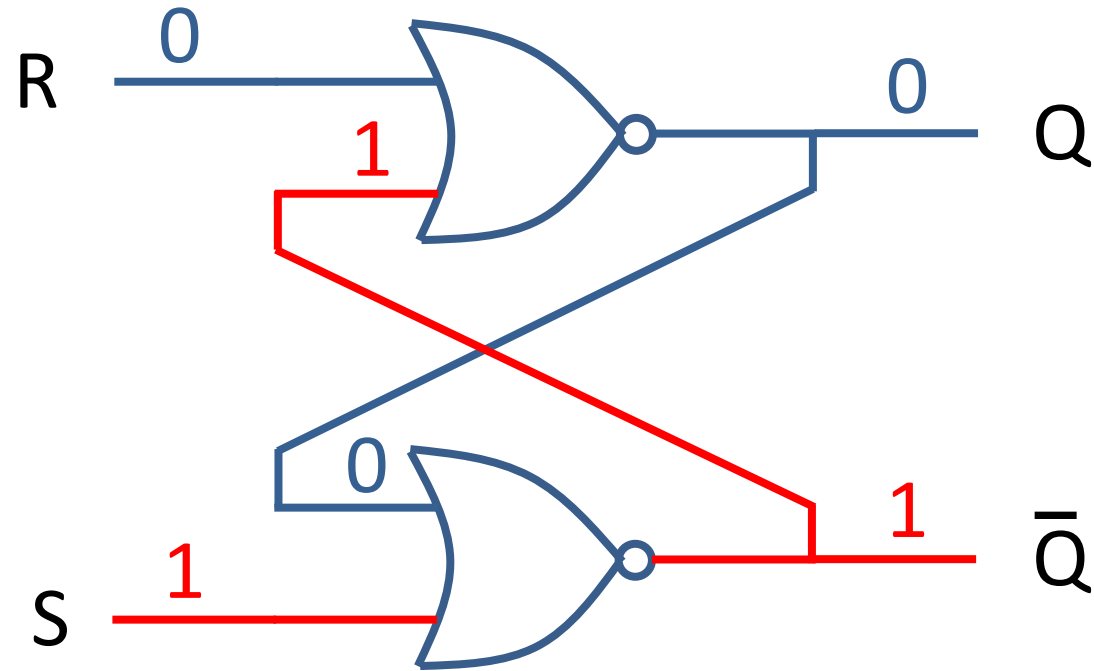


SR Latch



➤ Set state (1/3)

S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0

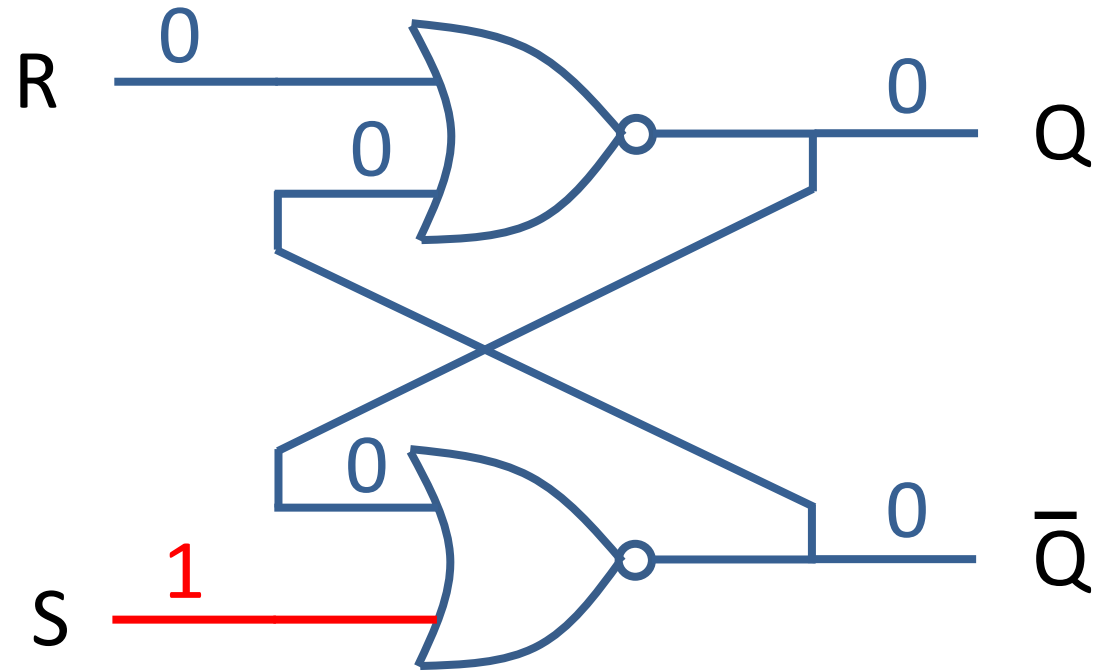


SR Latch

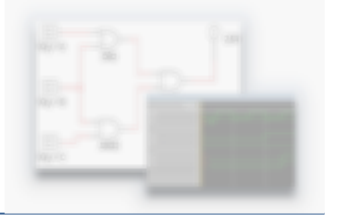


➤ Set state (2/3)

S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0

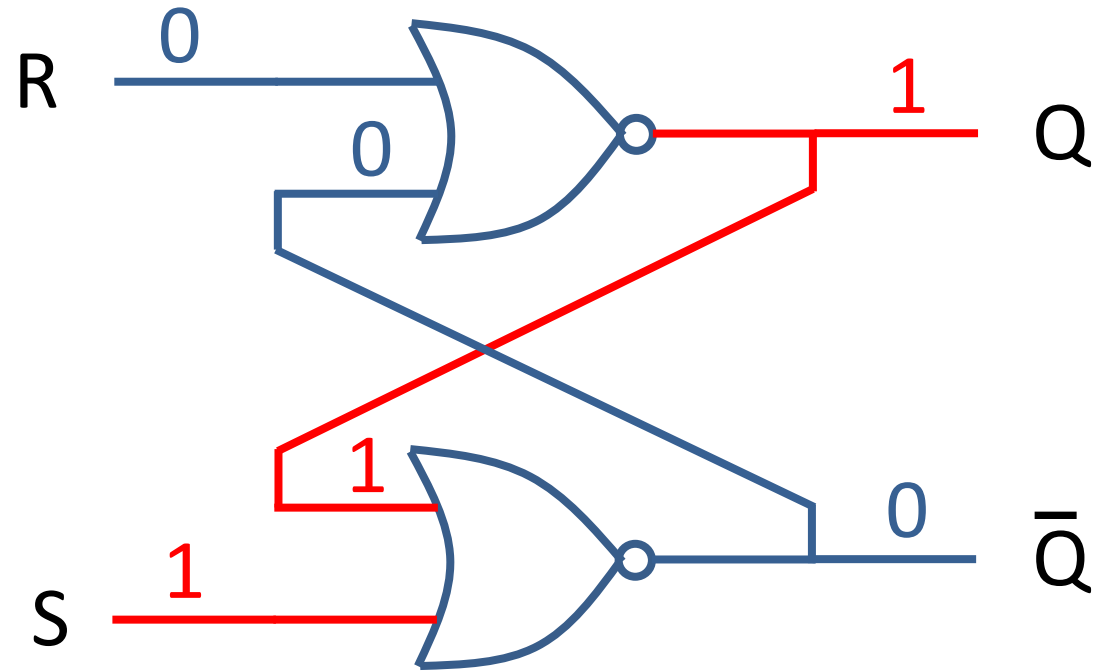


SR Latch

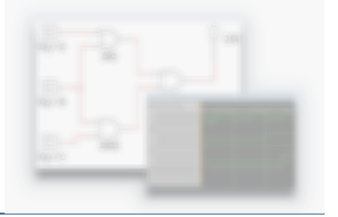


> Set state (3/3)

S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0



SR Latch

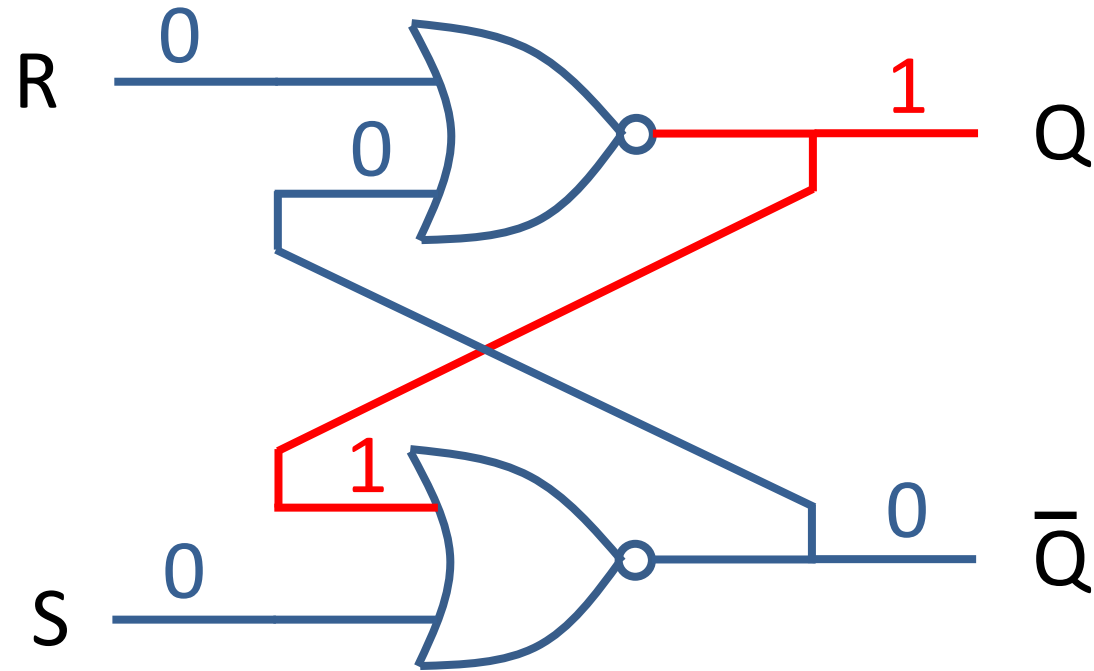


> Normal state

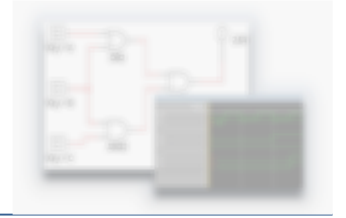
→

S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0

←

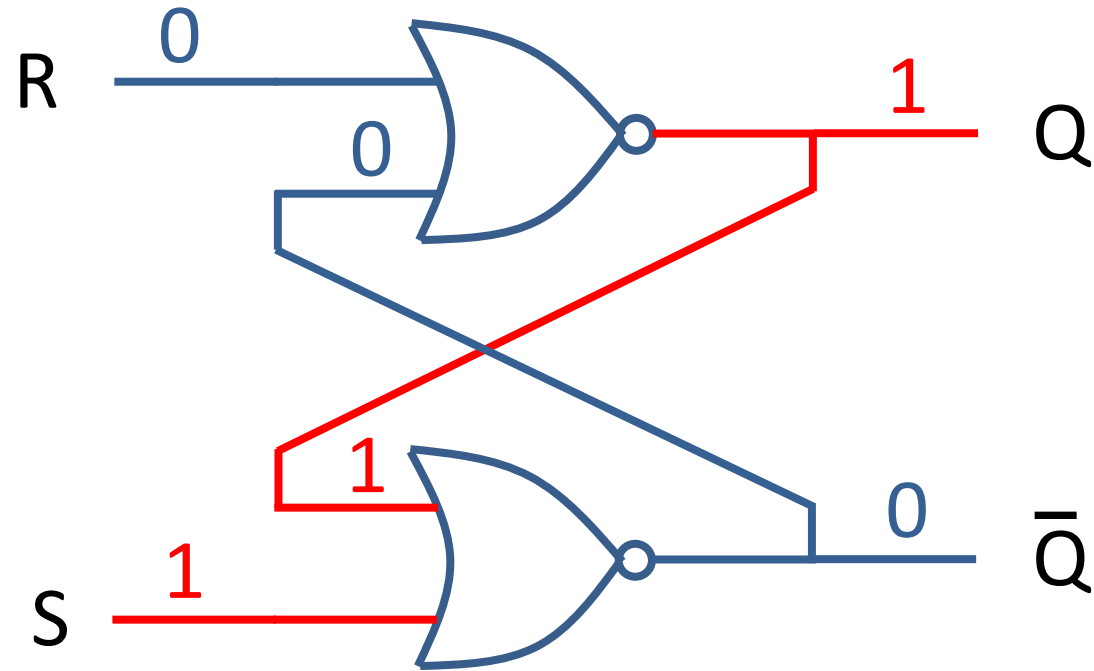


SR Latch



➤ **Set** state (no change)

S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0



SR Latch

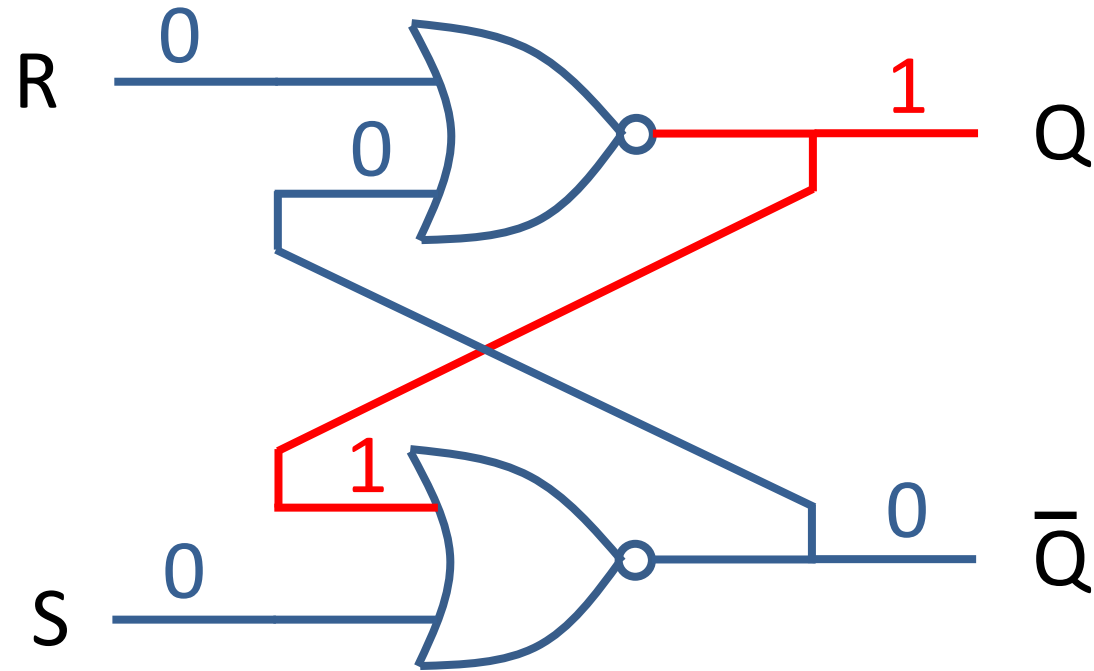


> Normal state

→

S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0

←



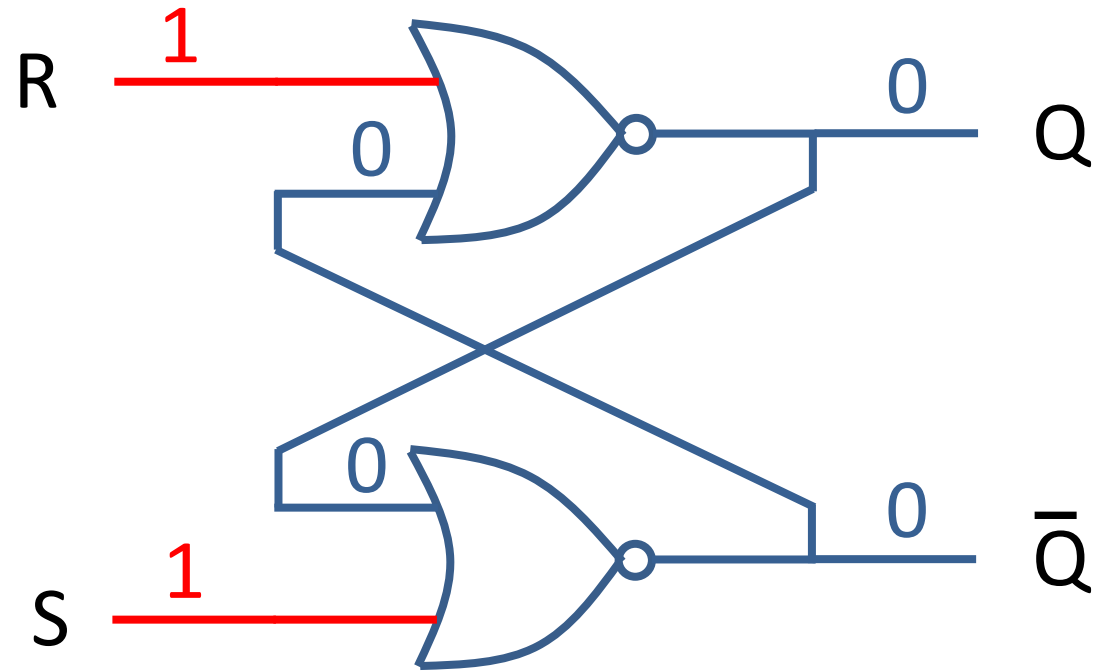
SR Latch



➤ Invalid state

S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0

Invalid



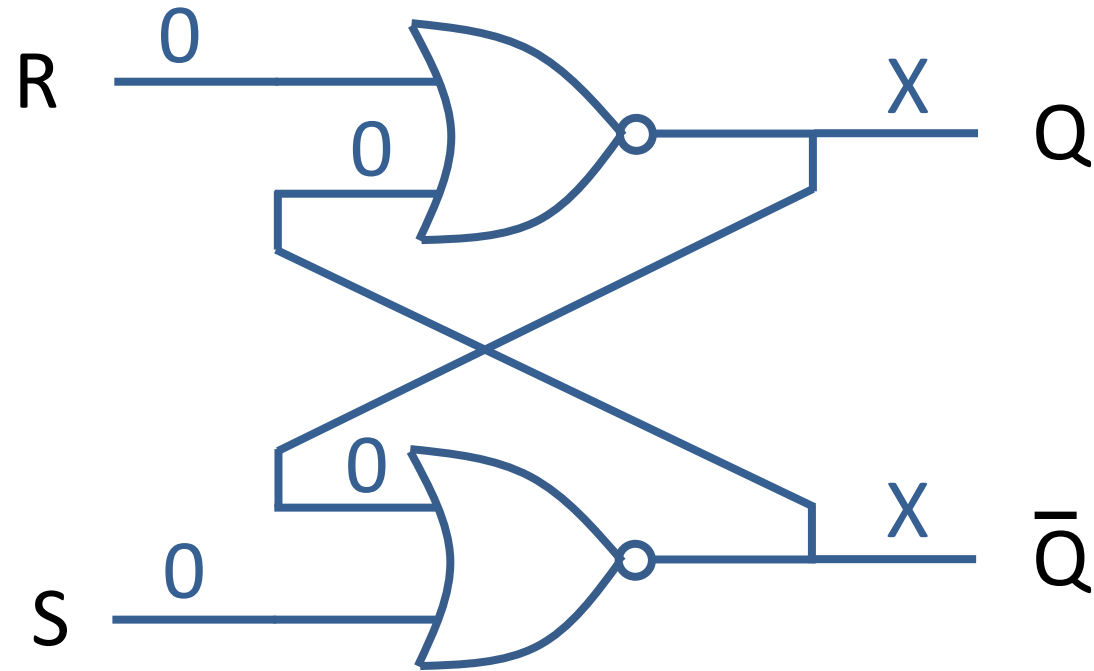
SR Latch



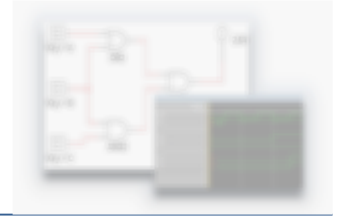
➤ Εάν μετά $R = S = 0$ ταυτόχρονα: Δεν μπορούμε να ξέρουμε ποια θα είναι η έξοδος! (Race)

S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0

Invalid



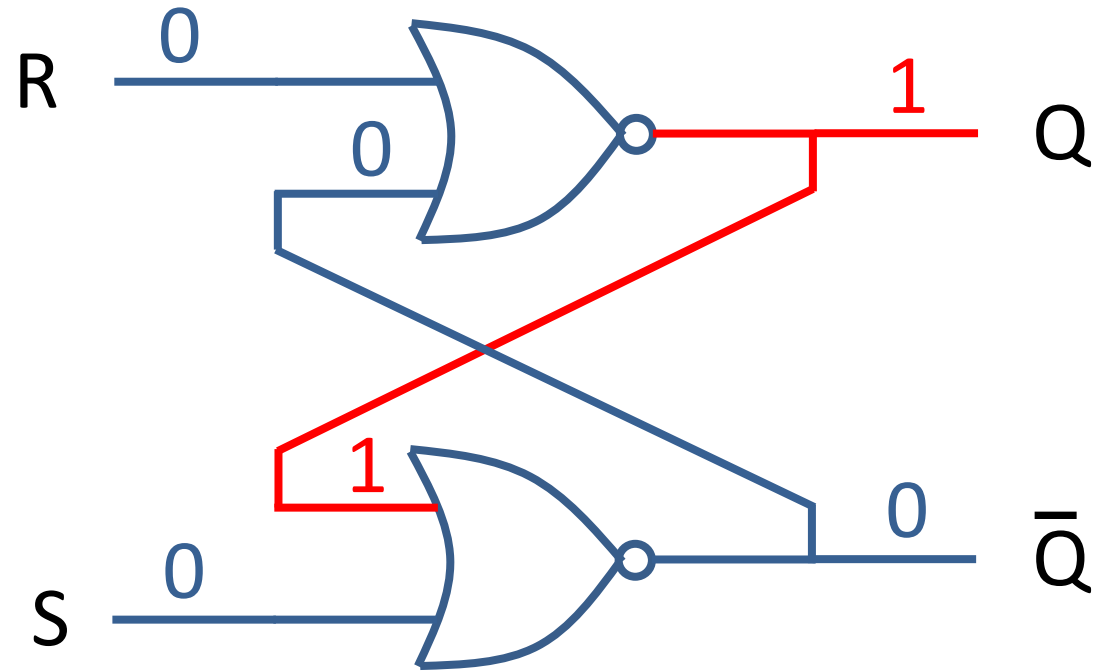
SR Latch



> Level Sensitive Active High SR latch

➔

S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0



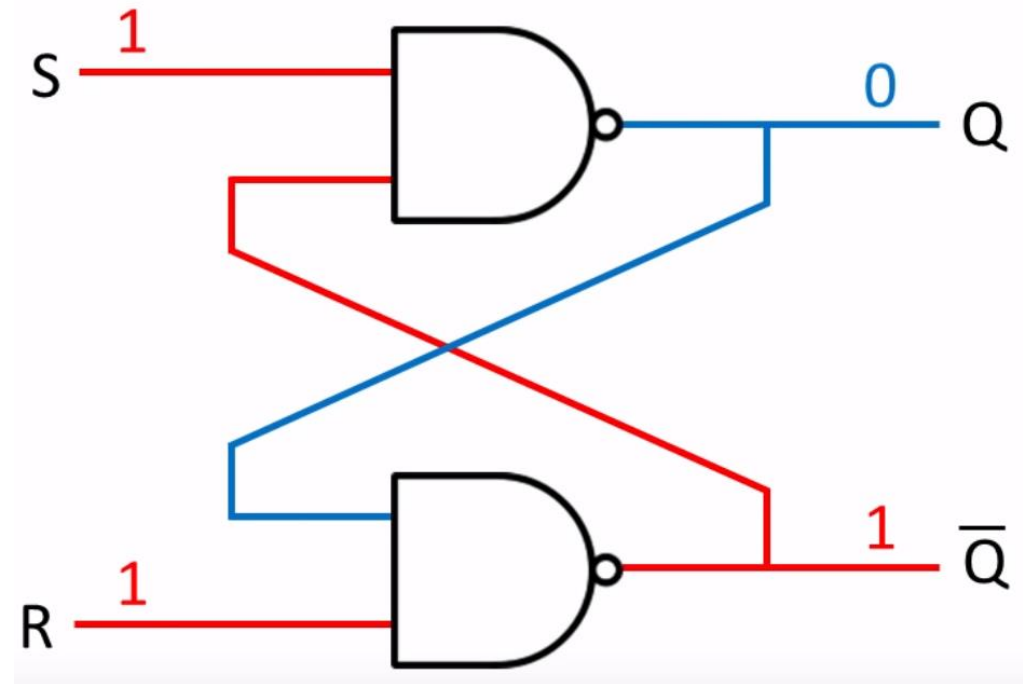
SR Latch



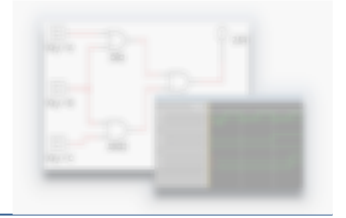
> Level Sensitive Active Low SR latch

S	R	Q	\bar{Q}
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0	1
		1	0

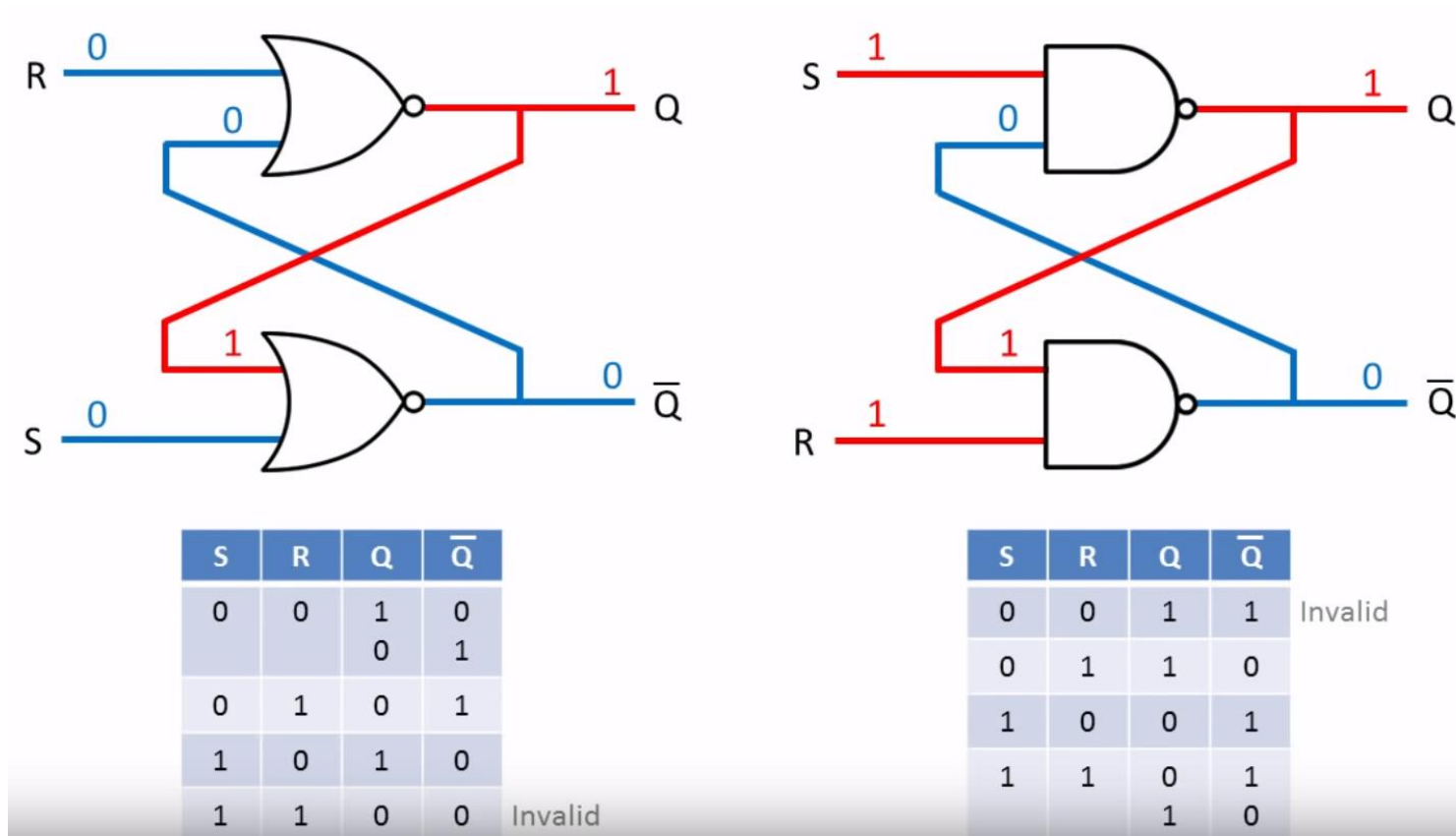
Invalid



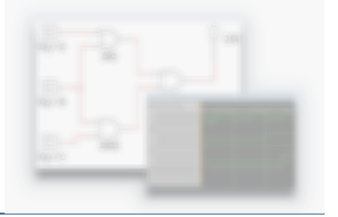
SR Latch



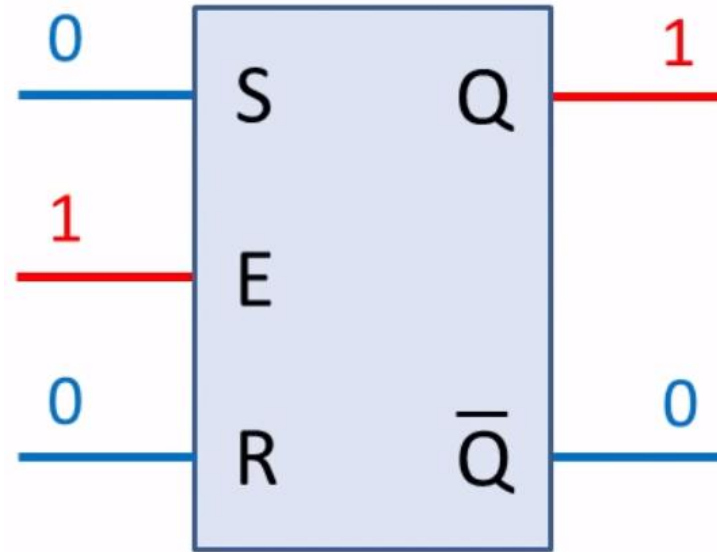
- Εκτελούν την ίδια λειτουργία, αλλά ελέγχονται με διαφορετικό τρόπο.



Gated SR Latch



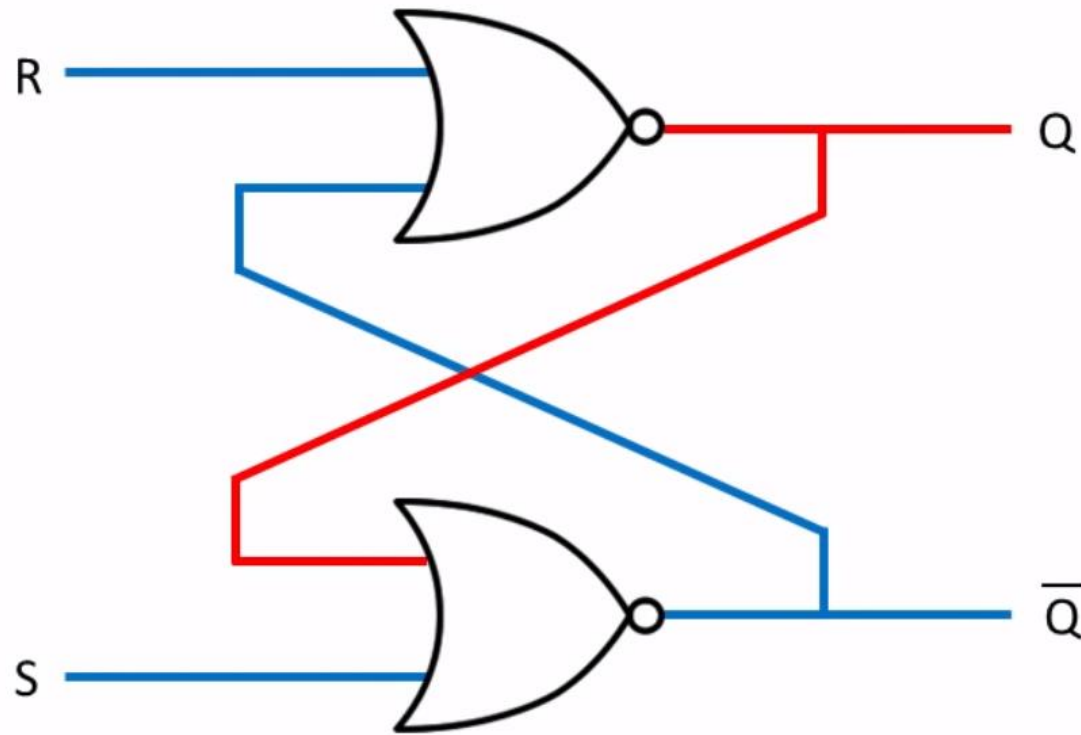
- Μπορεί να αλλάξει κατάσταση μόνο όταν είναι enable.



Gated SR Latch

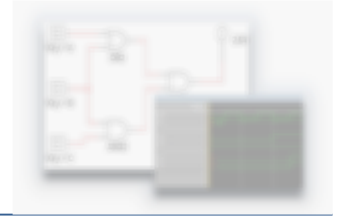


➤ Ungated Active High SR latch

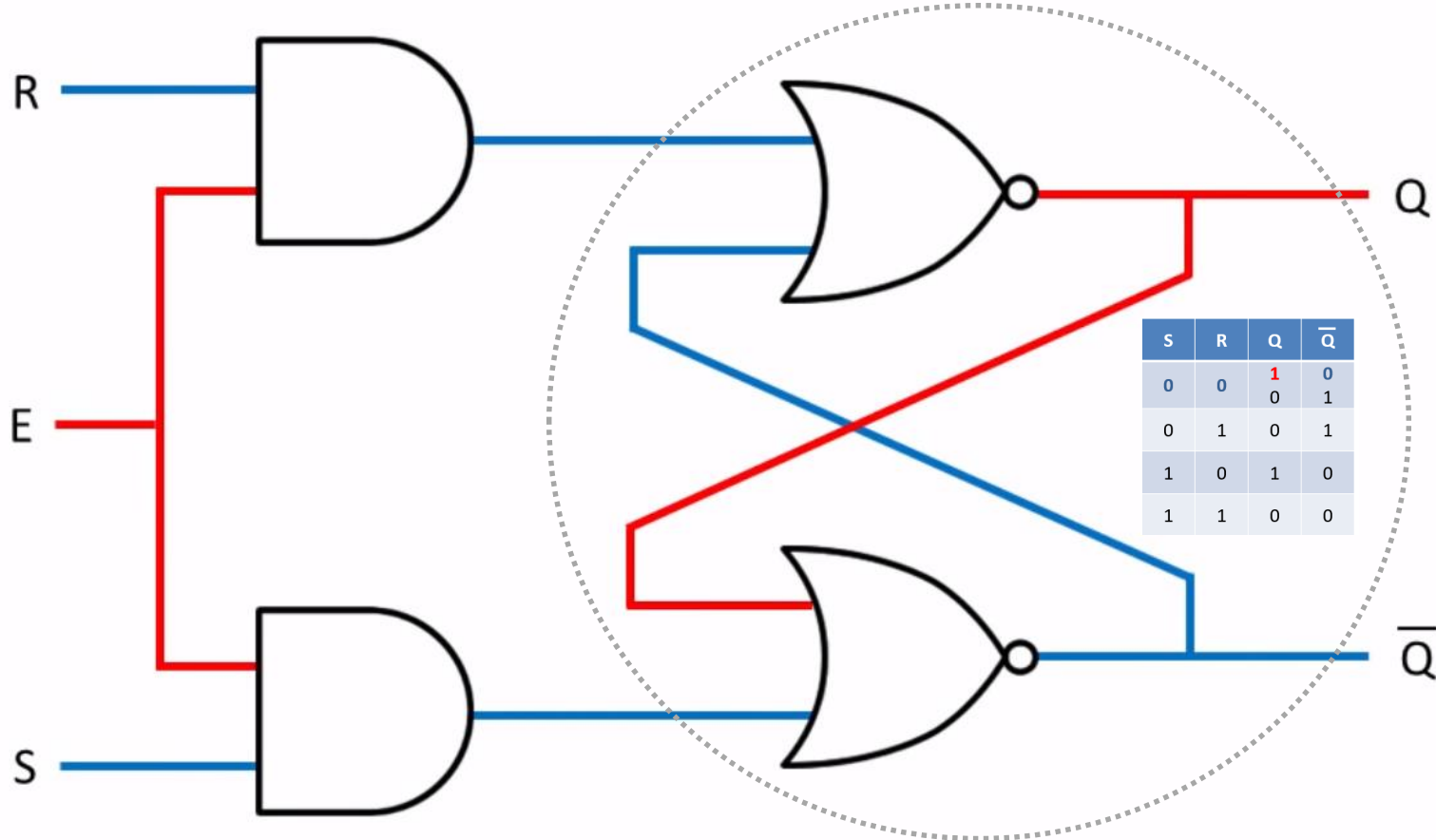


S	R	Q	\bar{Q}
0	0	1	0
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0

Gated SR Latch

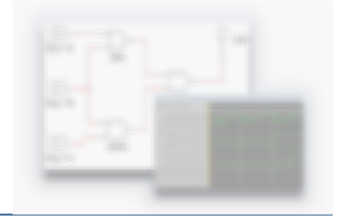


➤ **Gated Active High** SR latch. Η είσοδος E ενεργοποιεί ή όχι το latch

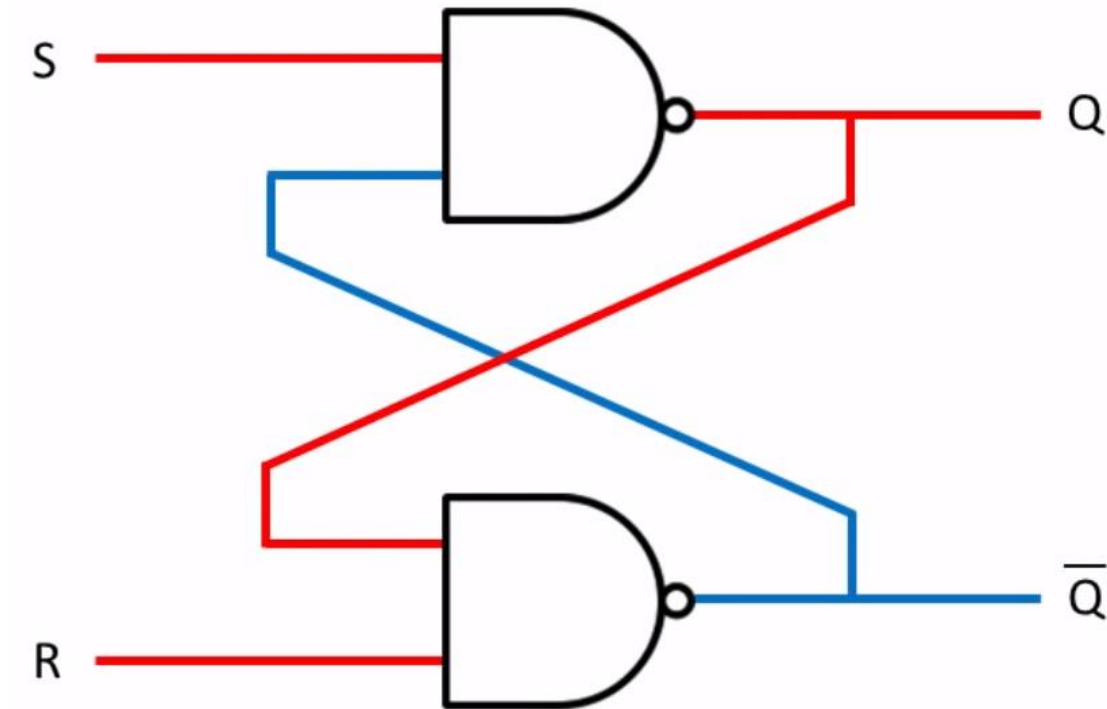


E	S	R	Q(t+1)
0	x	x	Q(t)
1	0	0	Q(t)
1	0	1	0
1	1	0	1
1	1	1	x

Gated SR Latch



➤ Ungated Active Low SR latch

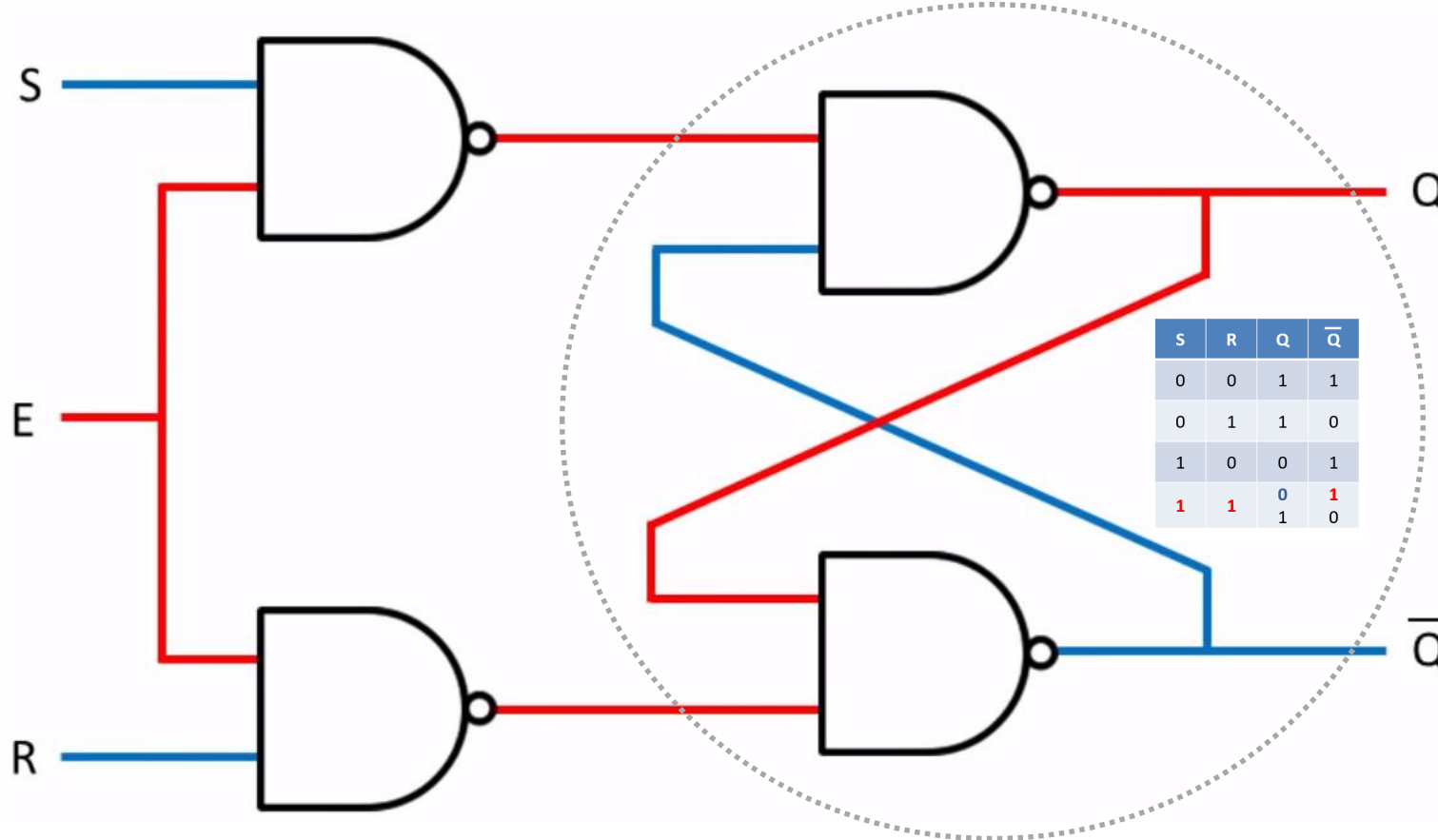


S	R	Q	\bar{Q}
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0 1	1 0

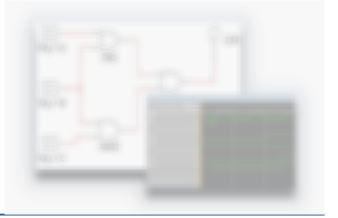
Gated SR Latch



➤ **Gated Active High** SR latch. Η είσοδος E ενεργοποιεί ή όχι το latch

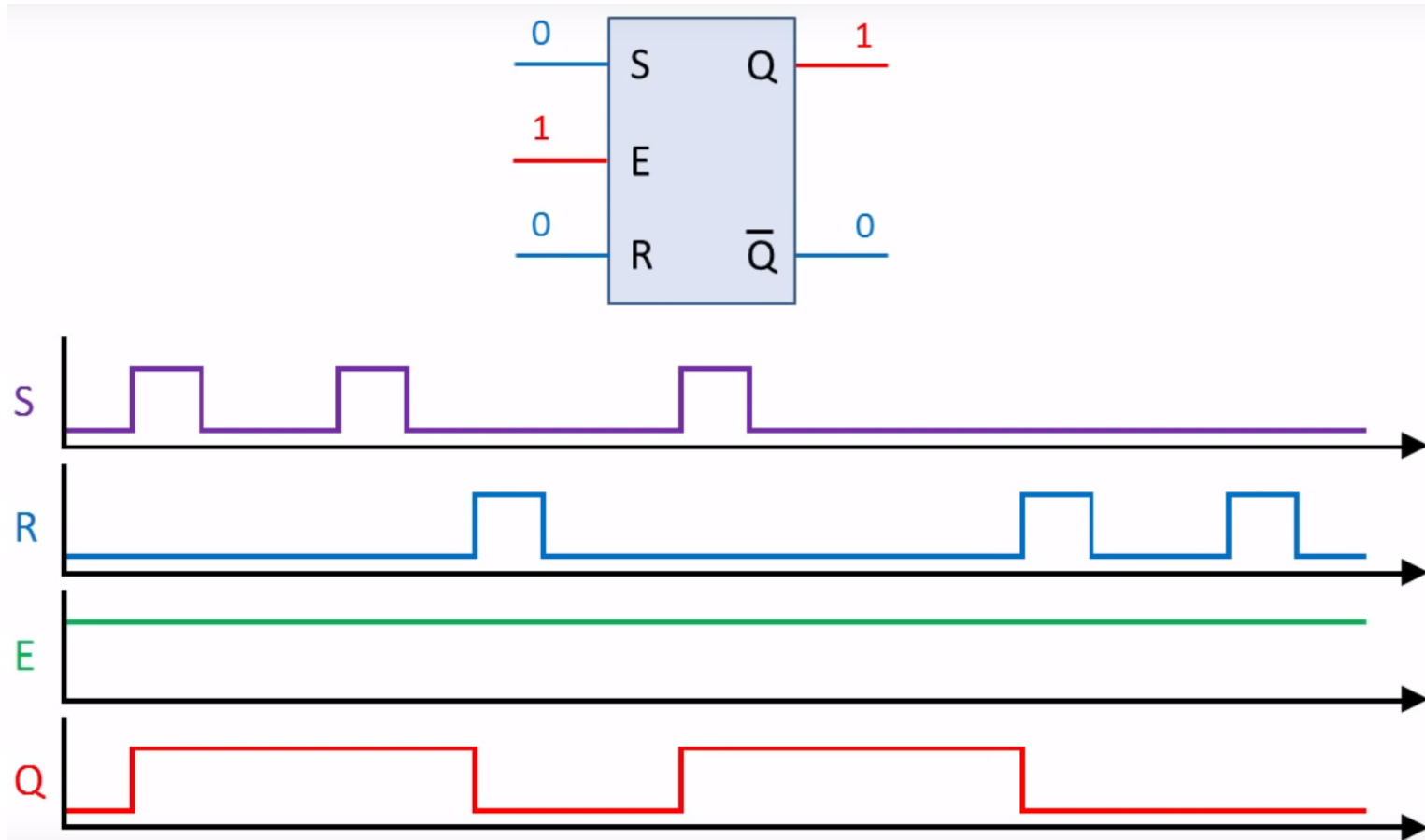


E	S	R	Q(t+1)
0	x	x	Q(t)
1	0	0	Q(t)
1	0	1	0
1	1	0	1
1	1	1	x



Gated SR Latch

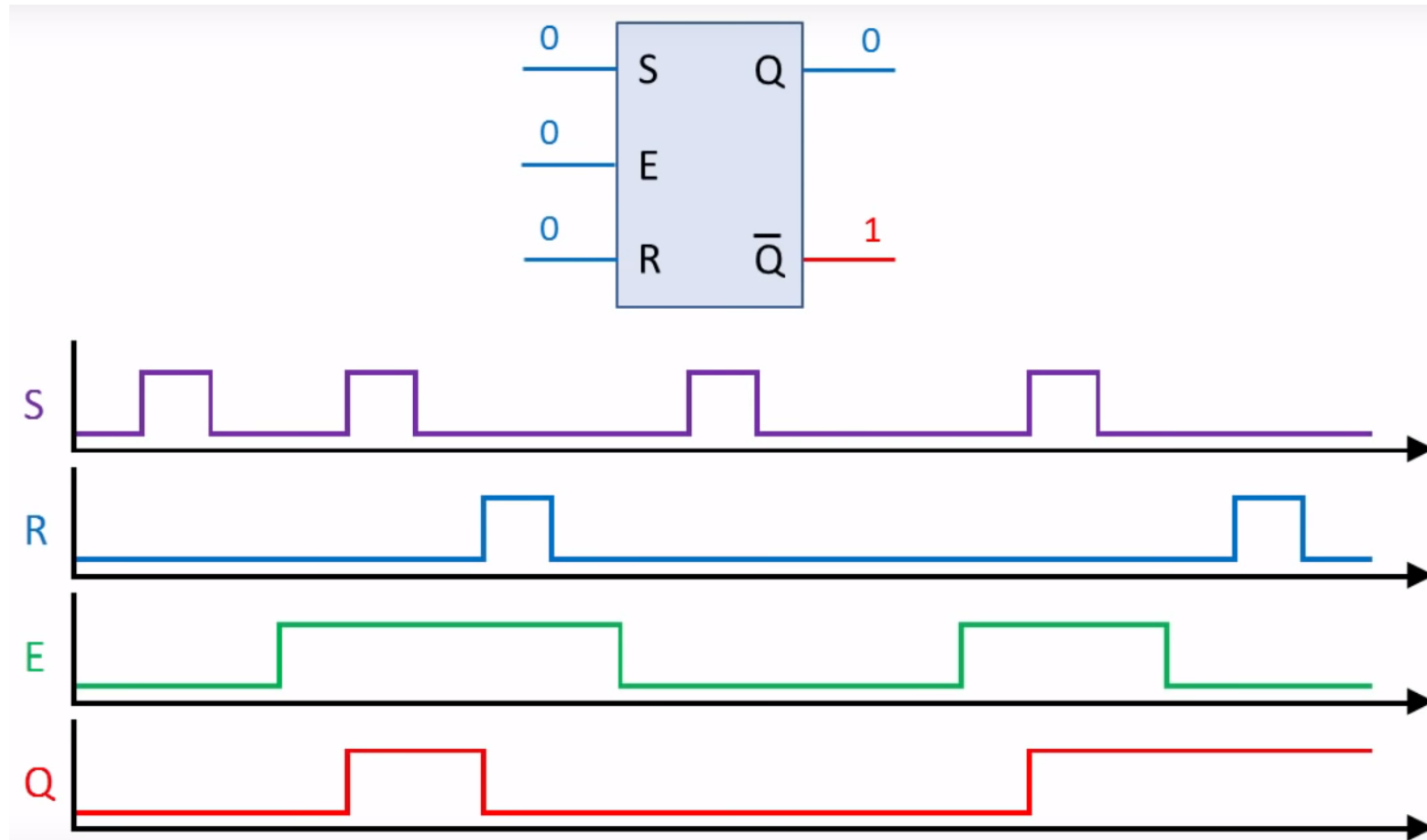
- Παράδειγμα 1: Το latch είναι πάντα enable.



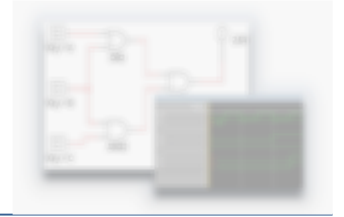


Gated SR Latch

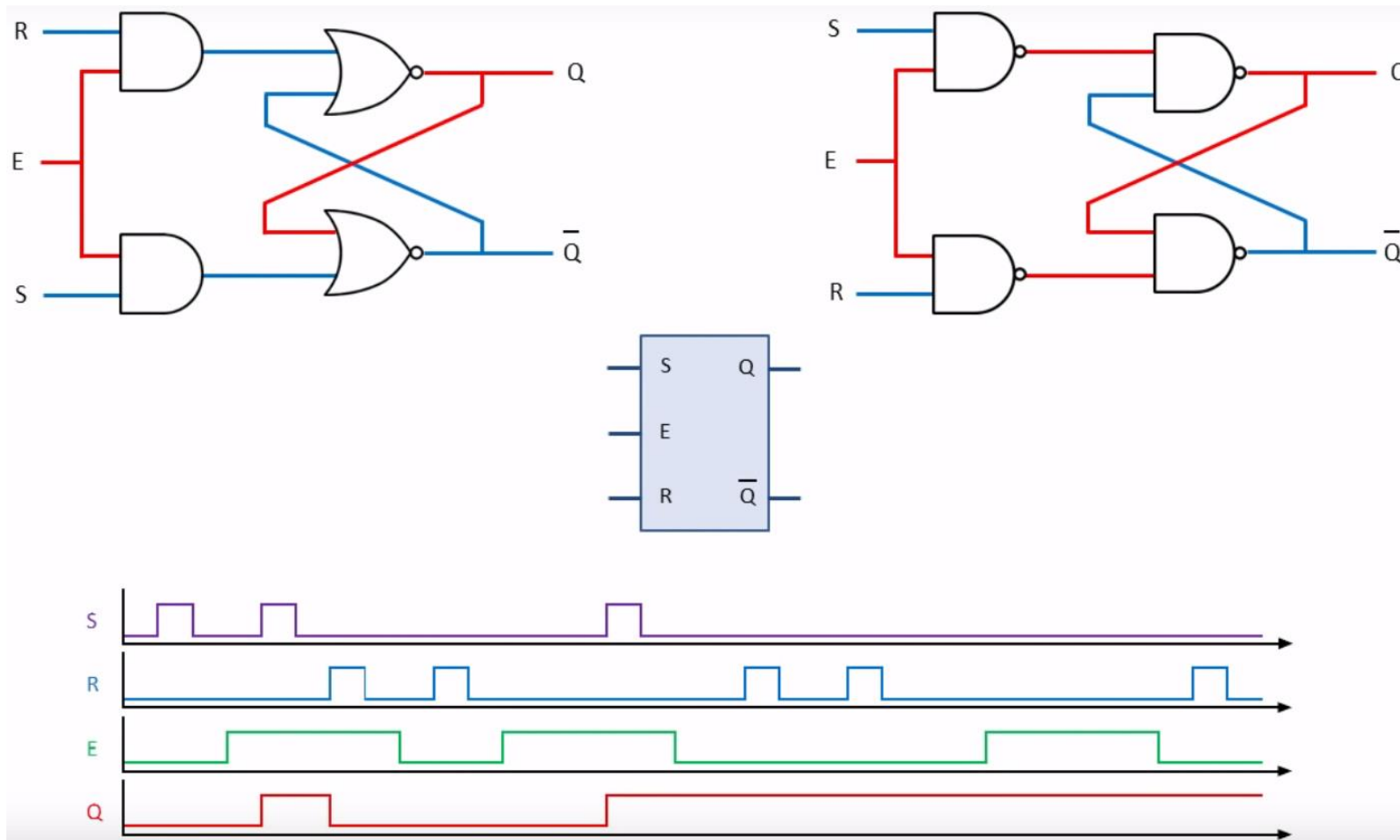
- Παράδειγμα 2: Το σήμα enable μεταβάλλεται.



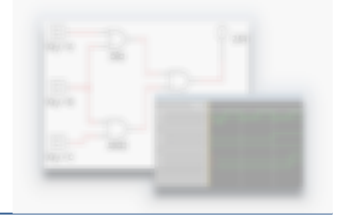
Gated SR Latch



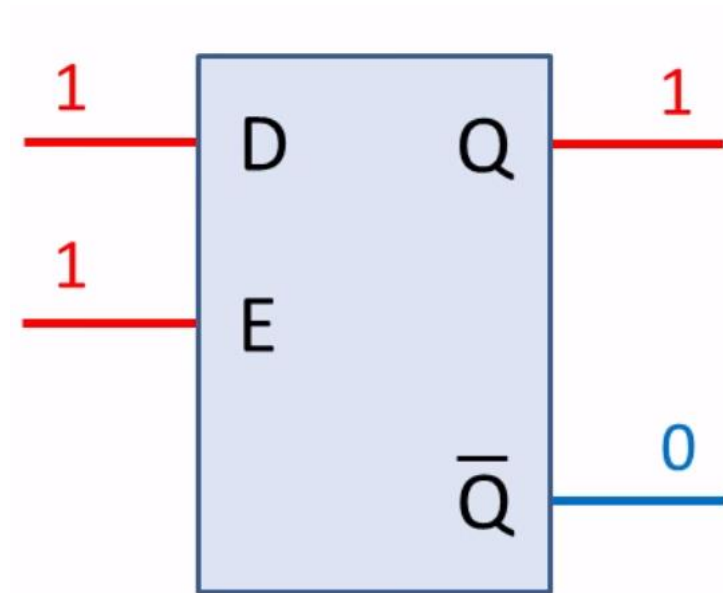
➤ Εκτελούν την ίδια λειτουργία.



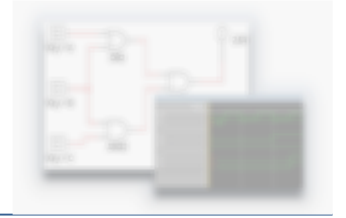
Gated D Latch



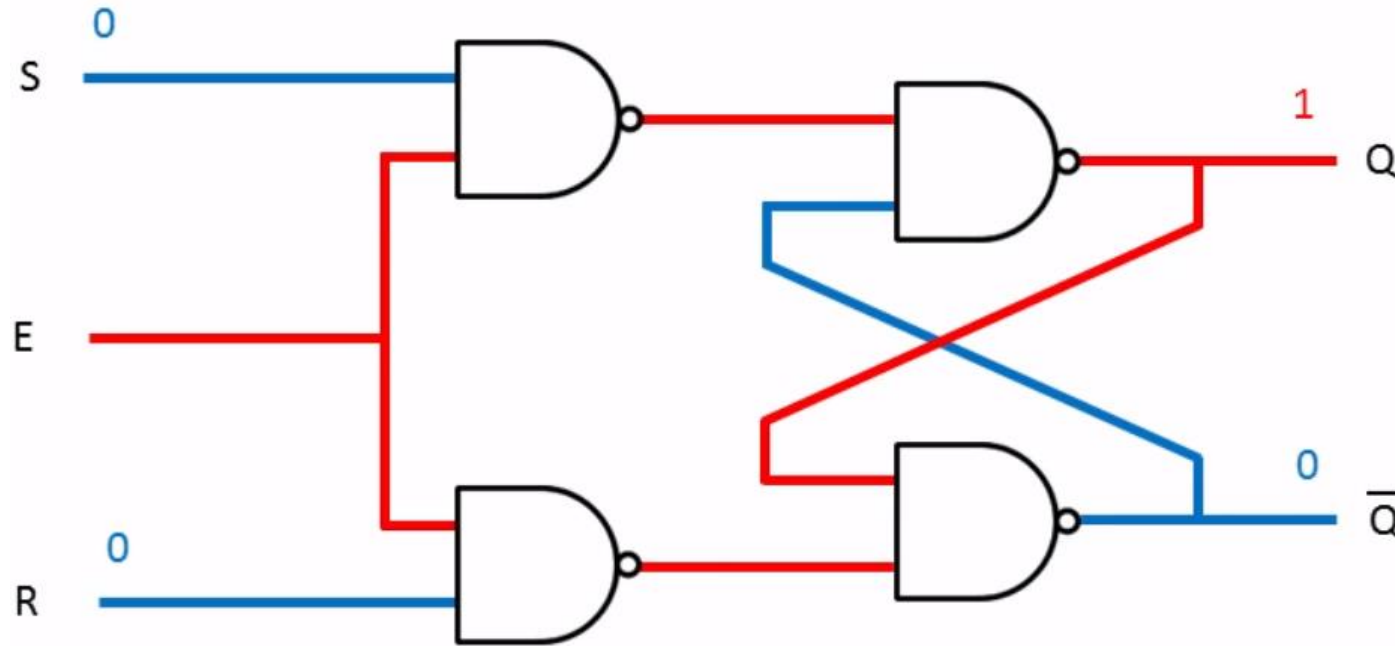
- 1-bit memory device.



Gated D Latch

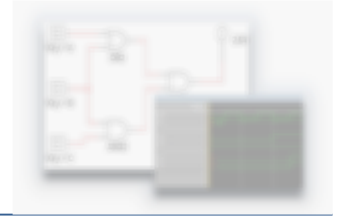


- Gated SR latch with NAND is an Active high latch

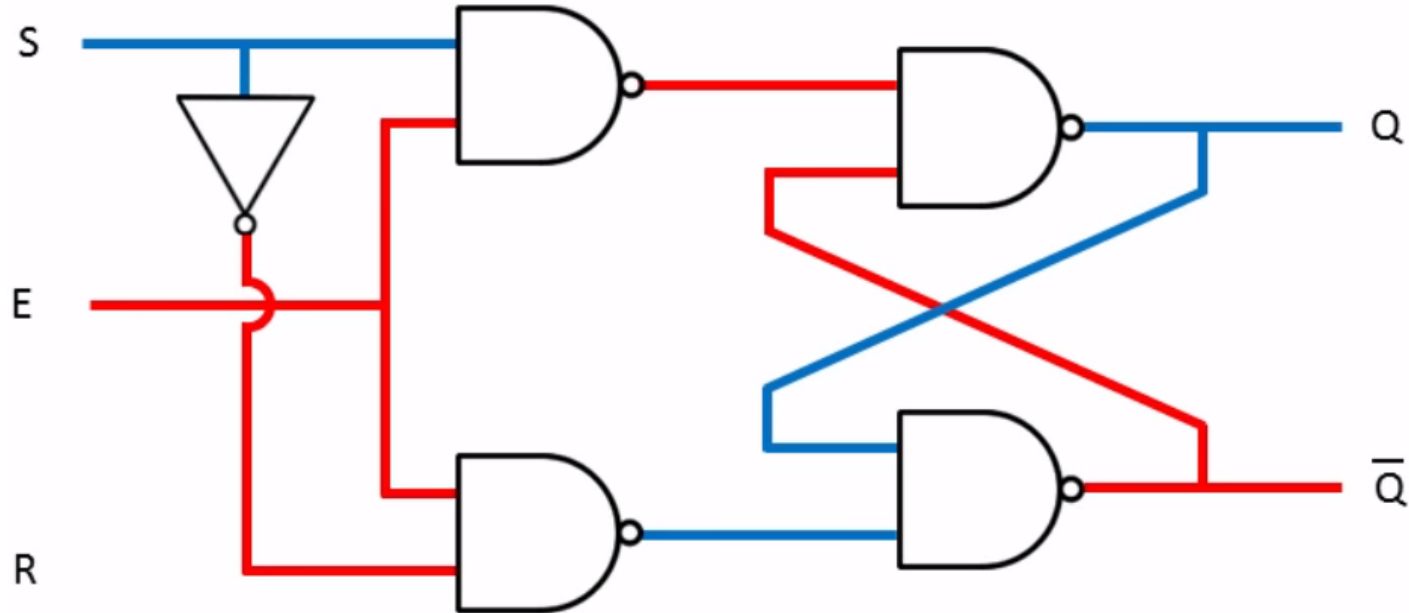


E	S	R	Q(t+1)
0	x	x	Q(t)
1	0	0	Q(t)
1	0	1	0
1	1	0	1
1	1	1	x

Gated D Latch



- Οι είσοδοι S και R είναι πάντα αντίθετες μεταξύ τους.

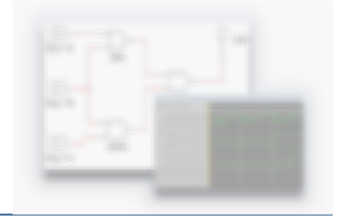


E	S	X	Q(t+1)
0	x	x	Q(t)
1	0	0	Q(t)
1	0	1	0
1	1	0	1
1	1	1	x

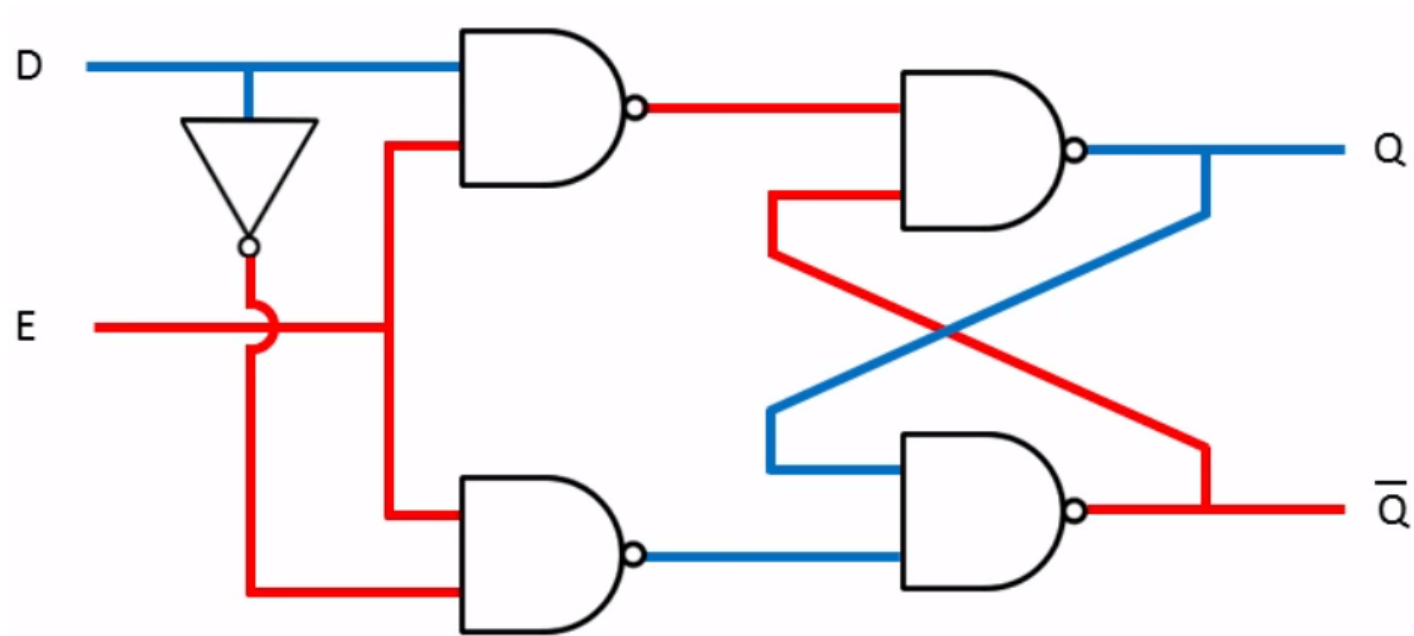


E	S		Q(t+1)
0	x		Q(t)
1	0		0
1	1		1

Gated D Latch

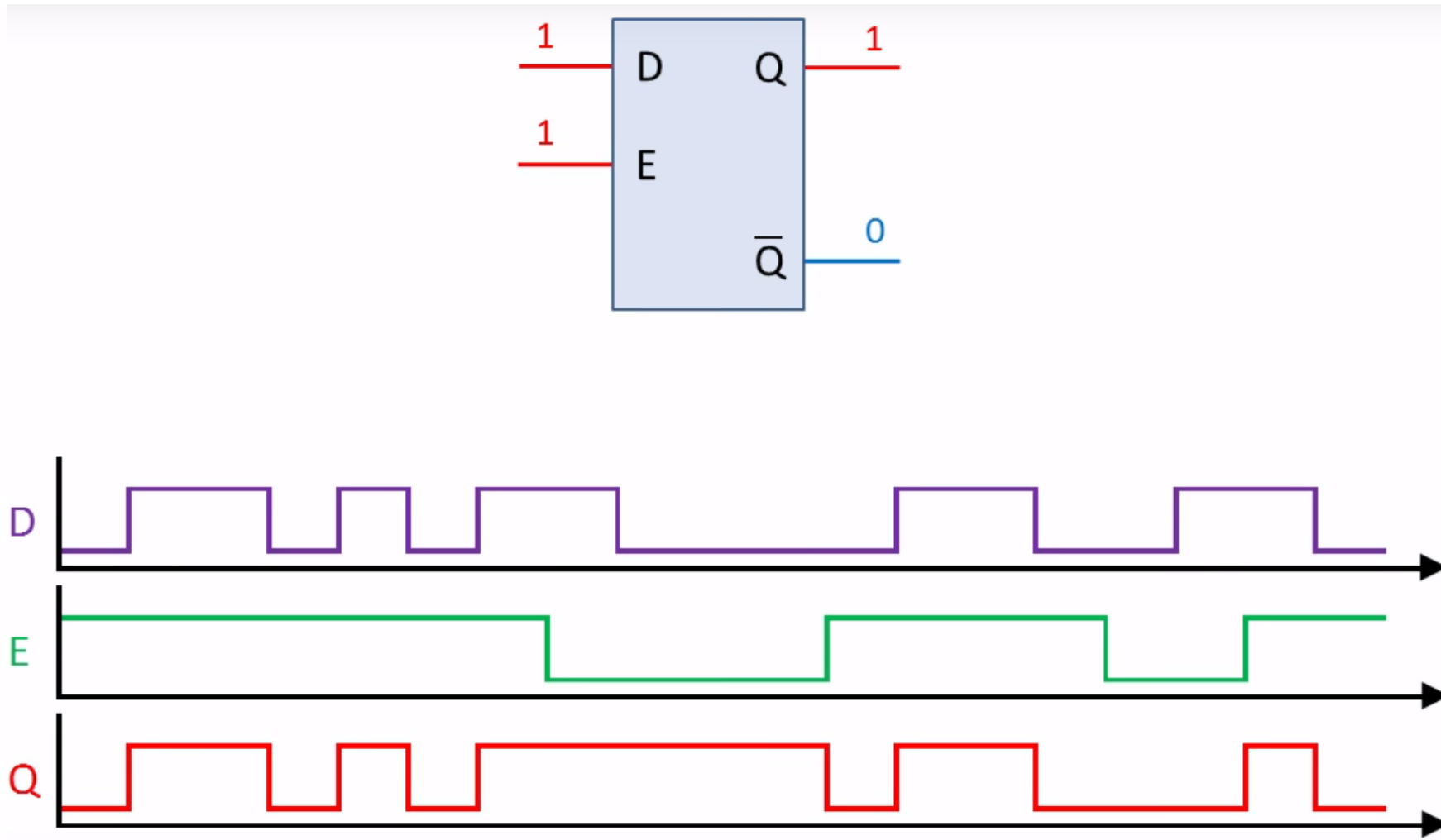
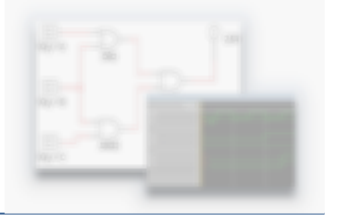


- Αλλάζουμε το όνομα της εισόδου από S σε **D**.



E	D	Q(t+1)
0	x	Q(t)
1	0	0
1	1	1

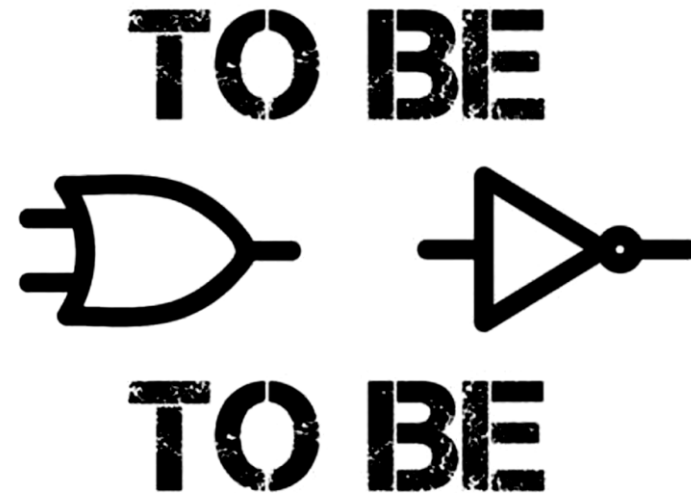
Gated D Latch



Ευχαριστώ για την προσοχή σας!



➤ Ερωτήσεις / Απορίες ;



Επικοινωνία: ece119.uth@gmail.com