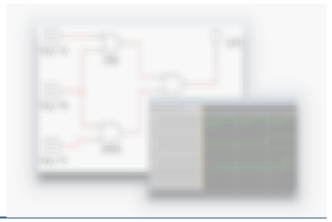


ECE119 – Ψηφιακή Σχεδίαση

Διδάσκοντες Εργαστηρίου: Δ. Καραμπερόπουλος
Α. Φεύγας
Μ. Πανταζή

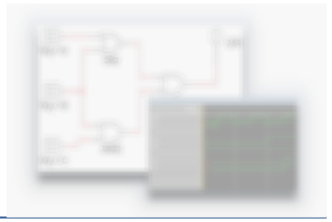
- Lab 8: Latches and Sequential Logic Circuits

Περιεχόμενα Εργαστηριακού Μαθήματος



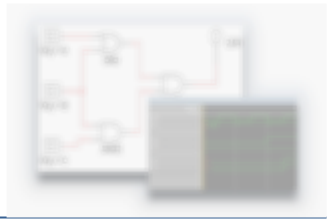
- Εισαγωγή
- Lab 1: Multisim Circuit Simulation and Basic Gates
- Lab 2: Truth Tables and Basic Logic Gates
- Lab 3: Logic Gates Explored and Boolean Algebra
- Lab 4: Karnaugh Maps
- Lab 5: Binary Conversion and Adders
- Lab 6: Encoders and Decoders
- Lab 7: Multiplexers and Demultiplexers
- **Lab 8: Latches and Sequential Logic Circuits**
- Lab 9: Flip-Flops
- Lab 10: Sequential Circuits – FSM (1ο)
- Lab 11: Sequential Circuits – FSM (2ο)

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 whose outputs depend on the current inputs and the stored state of the circuit, which reflects its previous behavior.
- We will also examine how enable and clock signals control when sequential circuits can change their stored state.

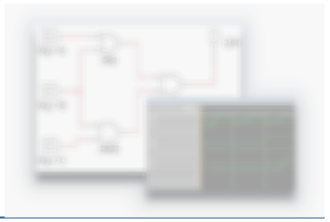
Learning Objectives



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

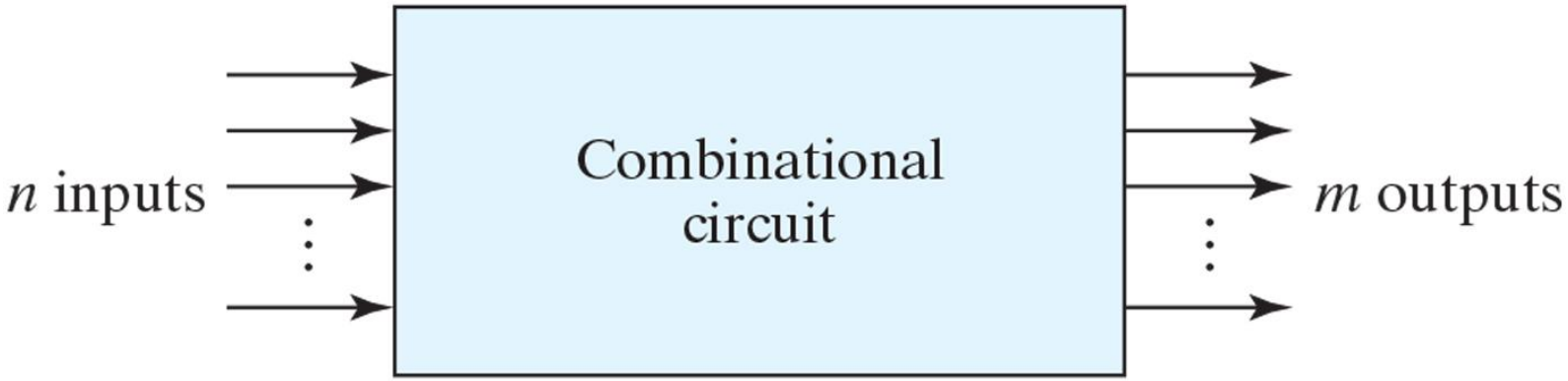
Expected Deliverables



In this lab, you will collect the following deliverables:

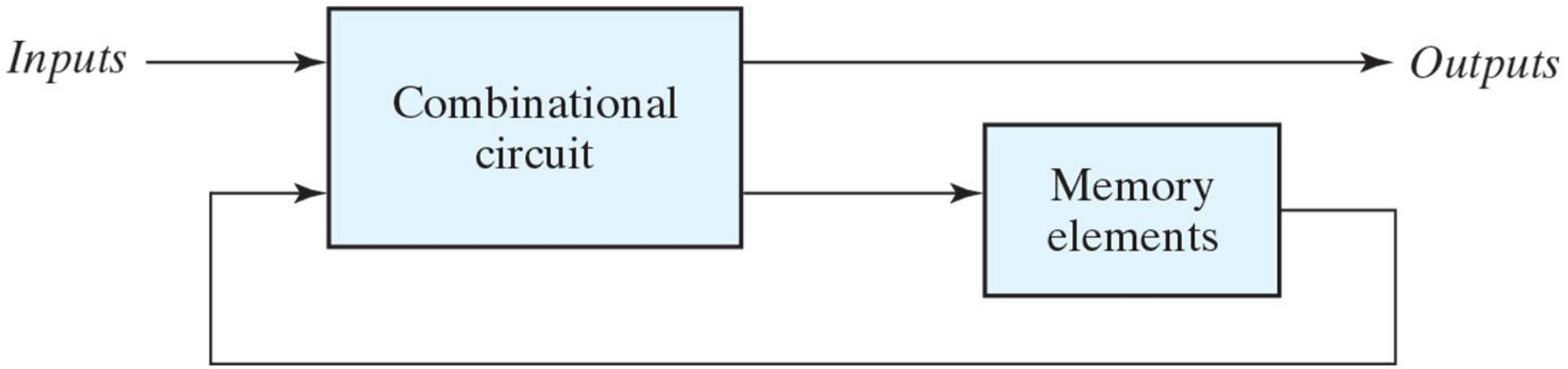
- Latch features table
- Latch comparisons
- Observations of probe behavior
- Conclusion questions

Συνδυαστικό κύκλωμα

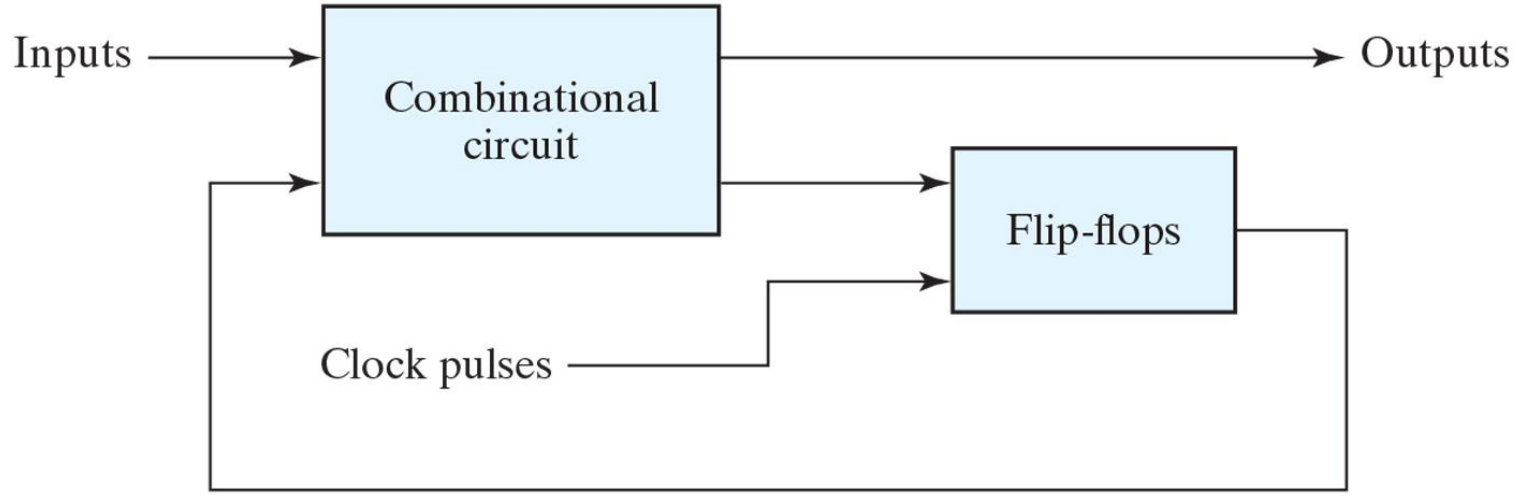


Ακολουθιακό κύκλωμα





Σύγχρονο Ακολουθιακό κύκλωμα με ρολόι

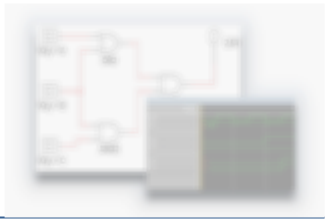


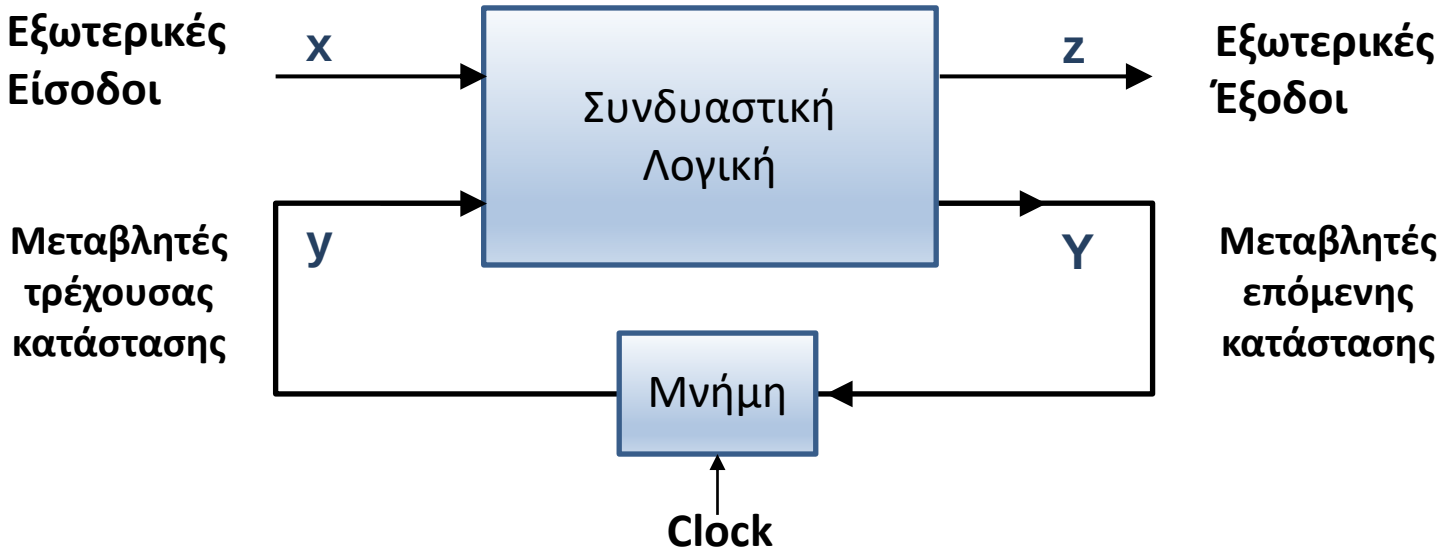
(a) Block diagram



(b) Timing diagram of clock pulses

Σύγχρονο Ακολουθιακό κύκλωμα με ρολόι



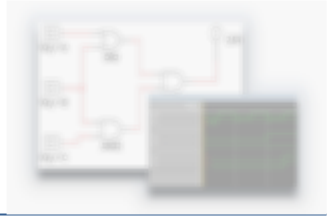


Σύγχρονο Ακολουθιακό Κύκλωμα

Εξισώσεις Εξόδων
z = f (x, y)
Y = g (x, y)

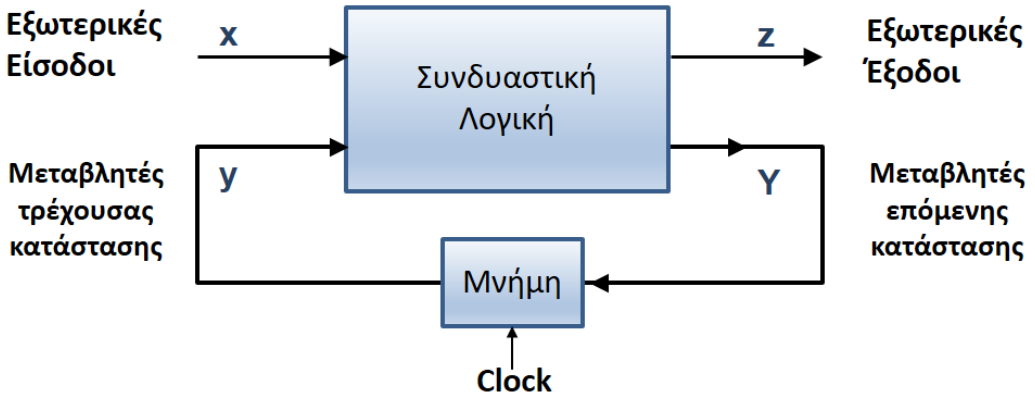
- Στα ακολουθιακά κυκλώματα η έξοδος **z** μπορεί να εξαρτάται τόσο από τις τρέχουσες εισόδους **x** όσο και από την παρούσα κατάσταση **y**. Η επόμενη κατάσταση **Y** υπολογίζεται επίσης από τις εισόδους και την παρούσα κατάσταση και αποθηκεύεται στα στοιχεία μνήμης κατά το κατάλληλο γεγονός συγχρονισμού.
- Τα ακολουθιακά κυκλώματα «θυμούνται προηγούμενες τιμές» μέσω της σύνδεσης της ανάδρασης.

Κύριες κατηγορίες των ακολουθιακών κυκλωμάτων



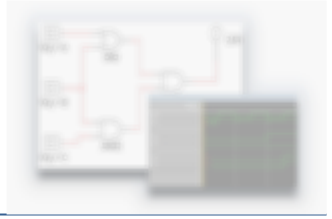
- **Ασύγχρονα**: Αλλάζουν κατάσταση σύμφωνα με τις αλλαγές των εισόδων τους. Απαιτούνται ειδικές τεχνικές σχεδιασμού.

- **Σύγχρονα**: Τα σήματα ανάδρασης διακόπτονται από καταχωρητές (Μνήμη) που σκανδαλίζονται από παλμούς ρολογιού (Clock).
- Συνεπώς η κατάστασή του κυκλώματος αλλάζει σύμφωνα με τους **παλμούς του ρολογιού**.
- Η κατάσταση του κυκλώματος ορίζεται από το περιεχόμενο των στοιχείων της μνήμης (y).

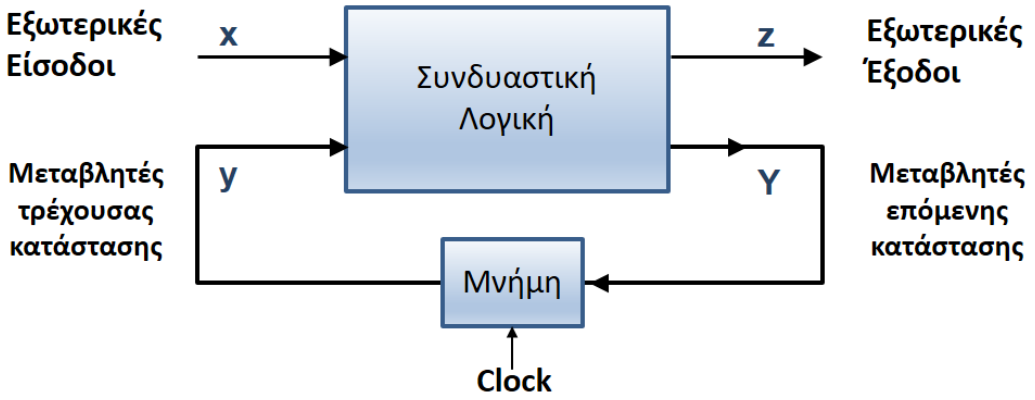


Εξισώσεις Εξόδων
z = f (x, y)
Y = g (x, y)

Κύριες κατηγορίες των ακολουθιακών κυκλωμάτων

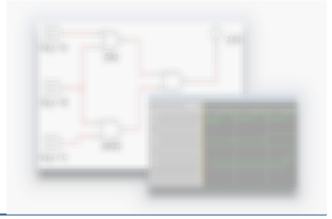


- Ένα σύγχρονο ακολουθιακό κύκλωμα πρέπει εξ΄ορισμού να χρησιμοποιεί σήματα συγχρονισμού (Clock).
- Τα σήματα συγχρονισμού επηρεάζουν τα στοιχεία μνήμης του κυκλώματος, ώστε αυτά να **αλλάζουν κατάσταση σε διακριτές χρονικές στιγμές.**
- Ο συγχρονισμός επιτυγχάνεται μέσω μιας «γεννήτριας κύριου-ρολογιού» η οποία τροφοδοτεί το σύστημα με μία περιοδική σειρά “παλμών ρολογιού”.

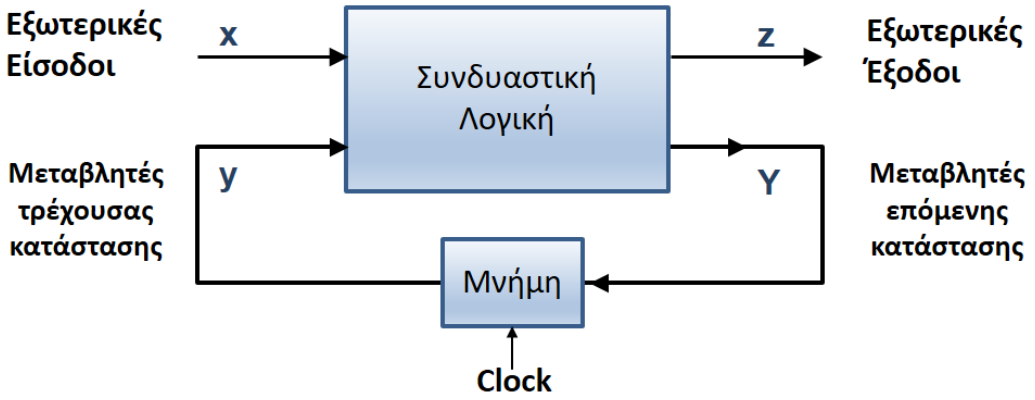


Εξισώσεις Εξόδων
z = f (x, y)
Y = g (x, y)

Κύριες κατηγορίες των ακολουθιακών κυκλωμάτων

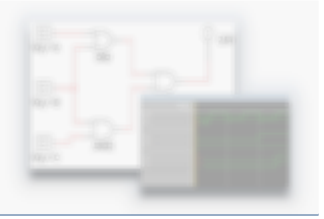


- Στα σύγχρονα ακολουθιακά κυκλώματα οι αλλαγές της αποθηκευμένης κατάστασης ελέγχονται από σήματα συγχρονισμού. Στα **flip-flops** πραγματοποιούνται σε συγκεκριμένες ακμές του clock, ενώ στα **latches** μπορούν να πραγματοποιούνται όσο το σήμα ενεργοποίησης βρίσκεται στην ενεργή στάθμη του.



Εξισώσεις Εξόδων
z = f (x, y)
Y = g (x, y)

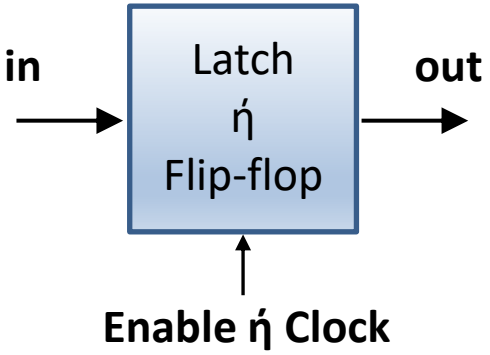
Το Latch (Μανδαλωτής) και το Flip-Flop ως στοιχεία μνήμης



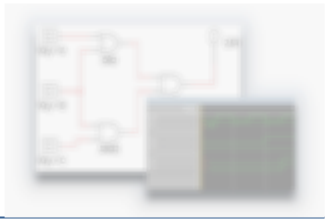
- Βασικά στοιχεία των ψηφιακών κυκλωμάτων είναι οι πύλες. Το χαρακτηριστικό των πυλών είναι ότι η έξοδος τους σε κάποια χρονική στιγμή εξαρτάται αποκλειστικά από την είσοδό τους την συγκεκριμένη χρονική στιγμή και όχι από προηγούμενες καταστάσεις τους. Δηλαδή οι πύλες δεν έχουν μνήμη.

- 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 Latch/flip-flop να ξαναδιεγερθεί.

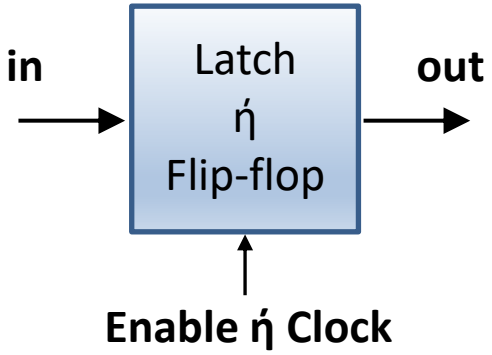
- Ένα κύκλωμα Latch ή flip-flop μπορεί να διατηρηθεί σε μία δυαδική κατάσταση επ’ αόριστον, έως ότου κάποιο σήμα (Enable ή Clock) προκαλέσει αλλαγή κατάστασης.



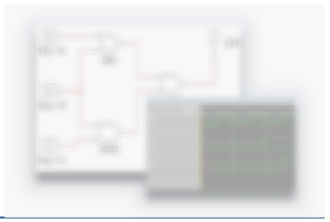
Το Latch (Μανδαλωτής) και το Flip-Flop ως στοιχεία μνήμης

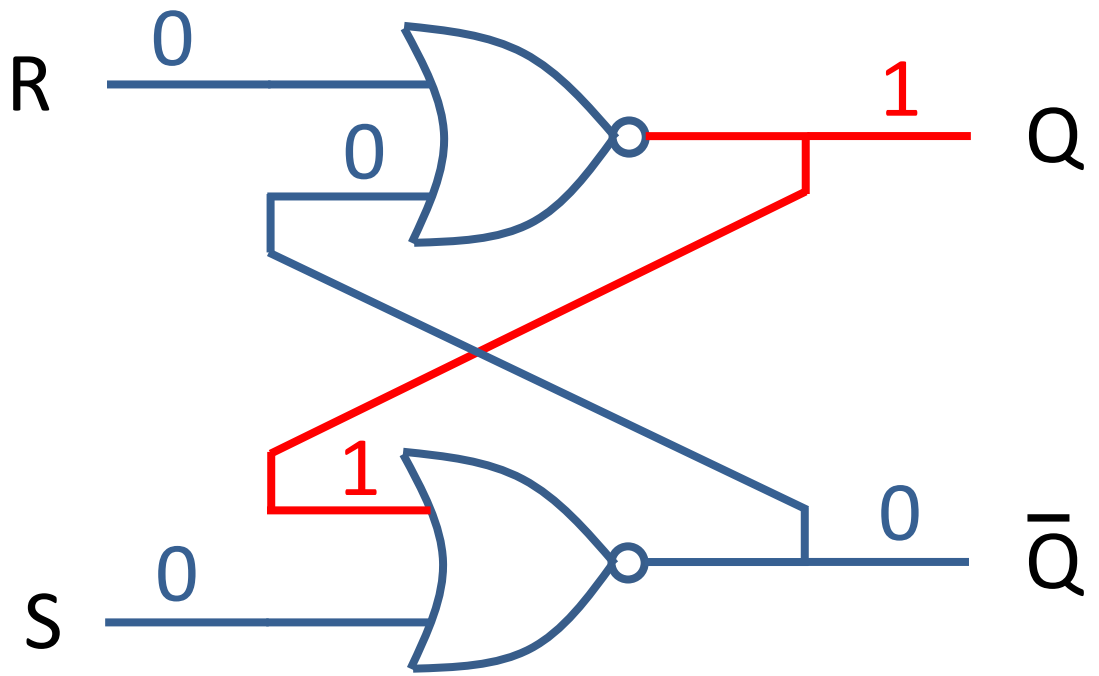


- Τα στοιχεία μνήμης που ενεργοποιούνται από την παρουσία συγκεκριμένων επιπέδων (συγκεκριμένων τιμών 0 ή 1) του σήματος ρολογιού ονομάζονται **μανδαλωτές (latches)**, ενώ εκείνα που ενεργοποιούνται από μεταβάσεις τιμών του σήματος ρολογιού (clock transitions) ονομάζονται **flip-flop**
- **Τα latches είναι διατάξεις ευαίσθητες στο επίπεδο σήματος ρολογιού**.
- **Τα flip-flop είναι διατάξεις ευαίσθητες στις ακμές του σήματος ρολογιού** ή ακμοπυροδότητες διατάξεις.
- Τα latches αποτελούν βασικά δομικά στοιχεία για την κατασκευή ορισμένων τύπων flip-flops, όπως τα master–slave flip-flops



SR Latch

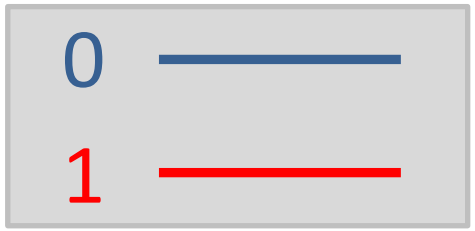




SR Latch



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

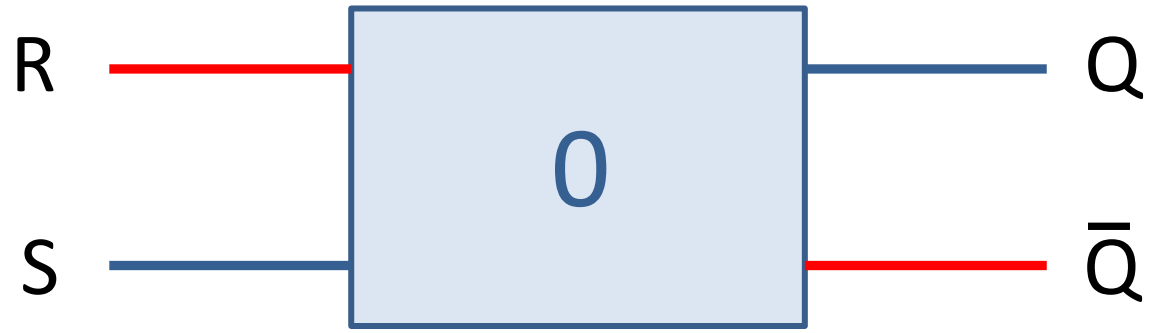


SR Latch



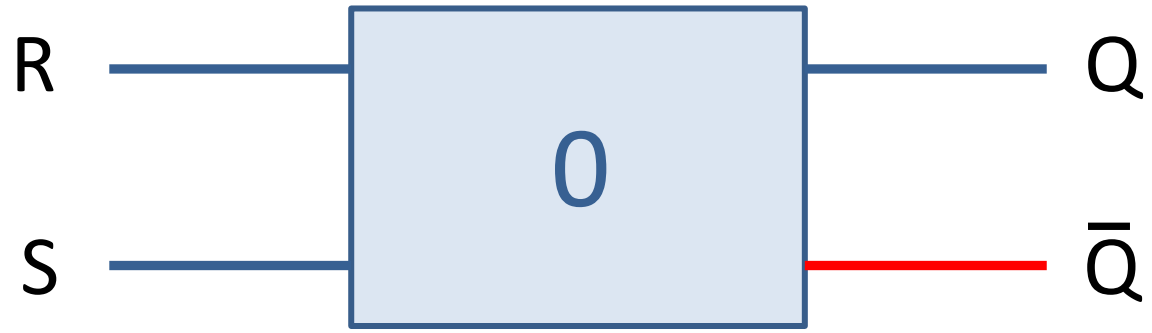
- Κατάσταση αναμονής (μνήμη):
  - R = 0
  - S = 0

SR Latch



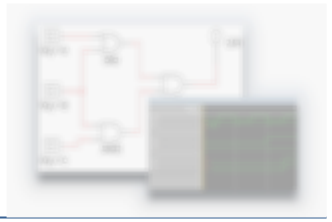
- Κατάσταση επαναφοράς ή μηδενισμού (reset state):
  - R = 1
  - S = 0

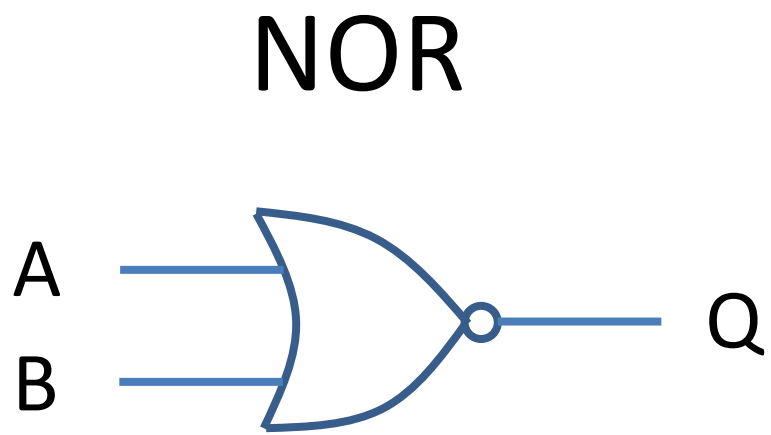
SR Latch



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

SR Latch



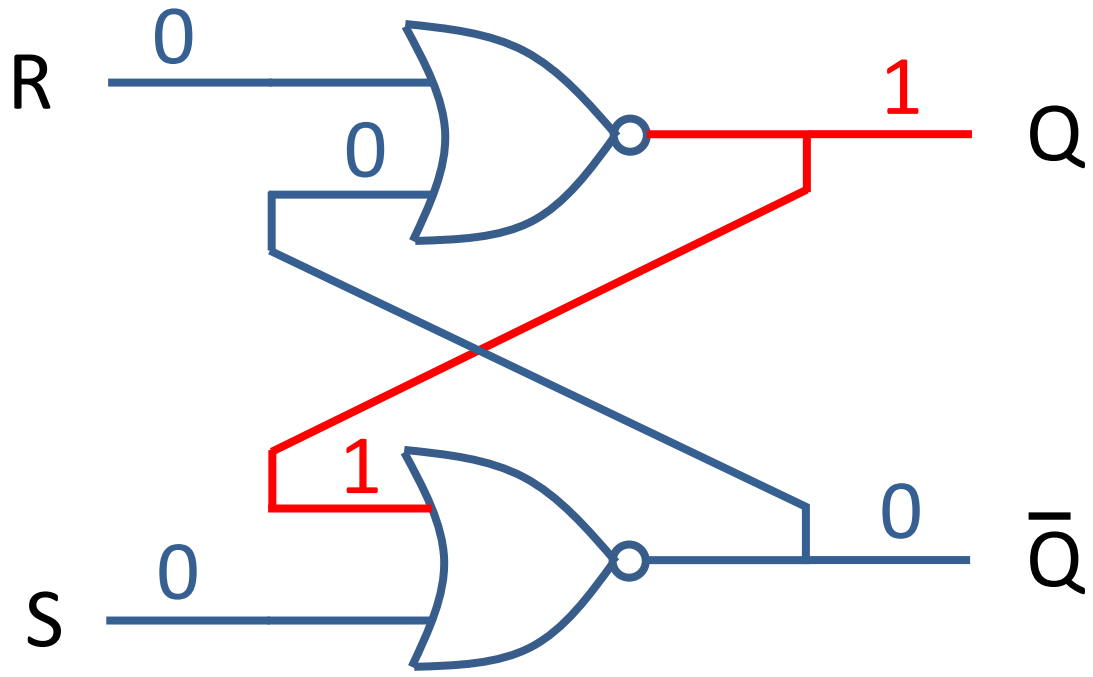


A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

SR Latch

- **Normal** state

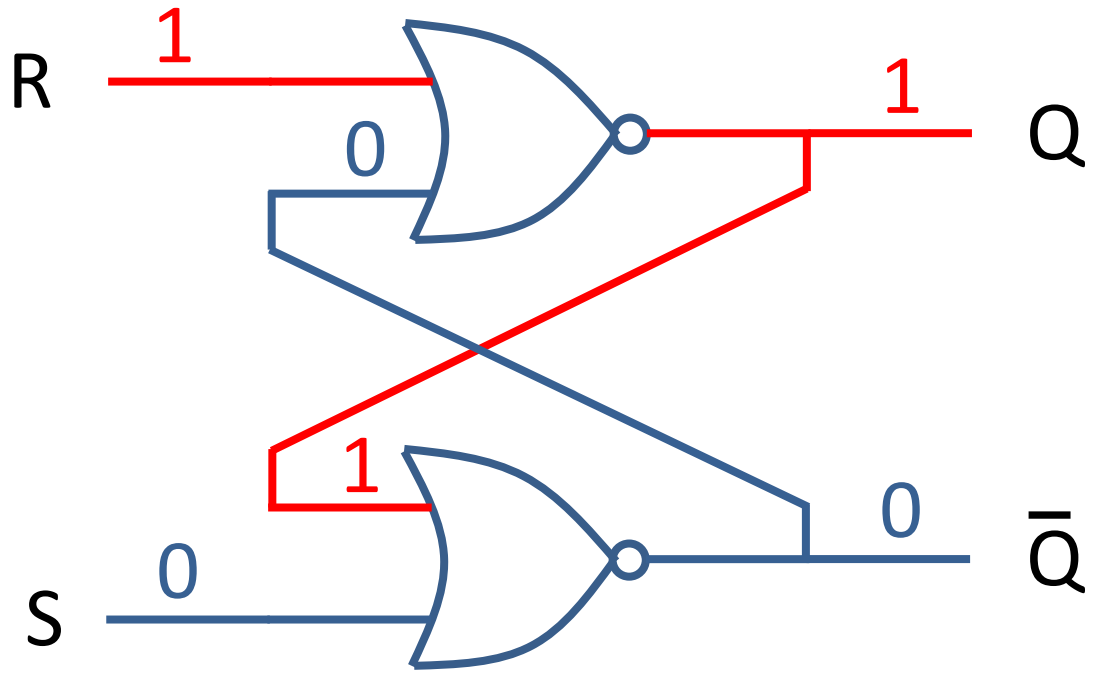
S	R	Q	$\overline{Q}$
0	0	1 0	0 1
0	1	0	1
1	0	1	0
1	1	0	0



SR Latch

- **Reset** state (1/3)

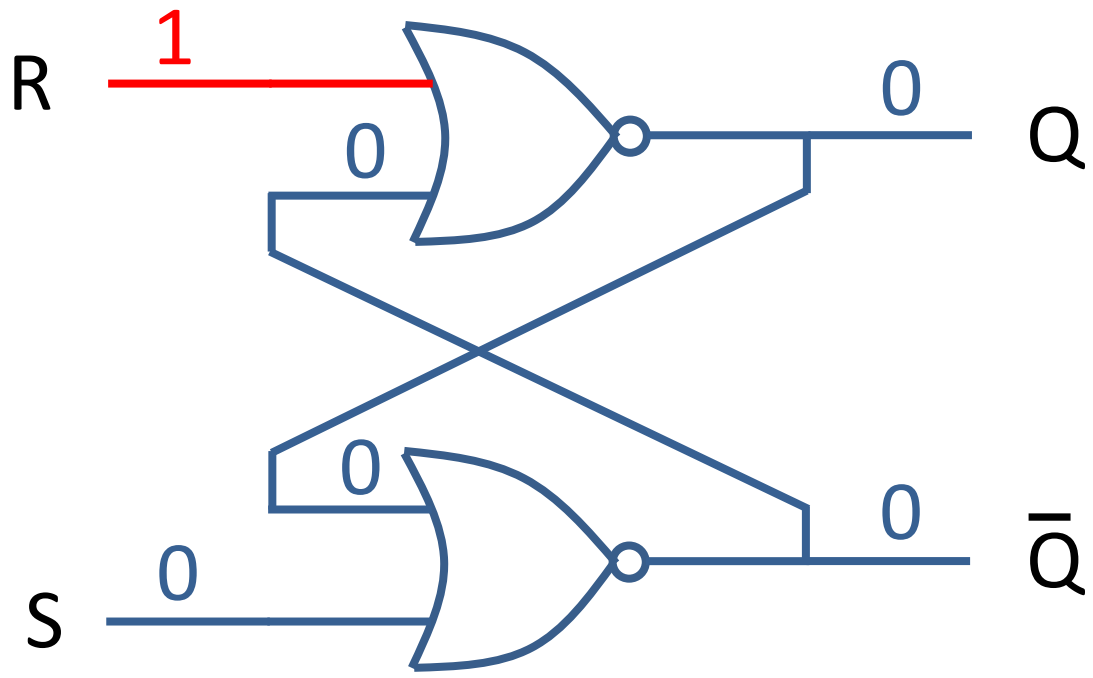
S	R	Q	$\overline{Q}$
0	0	1 0	0 1
0	**1**	**0**	**1**
1	0	1	0
1	1	0	0



SR Latch

- **Reset** state (2/3)

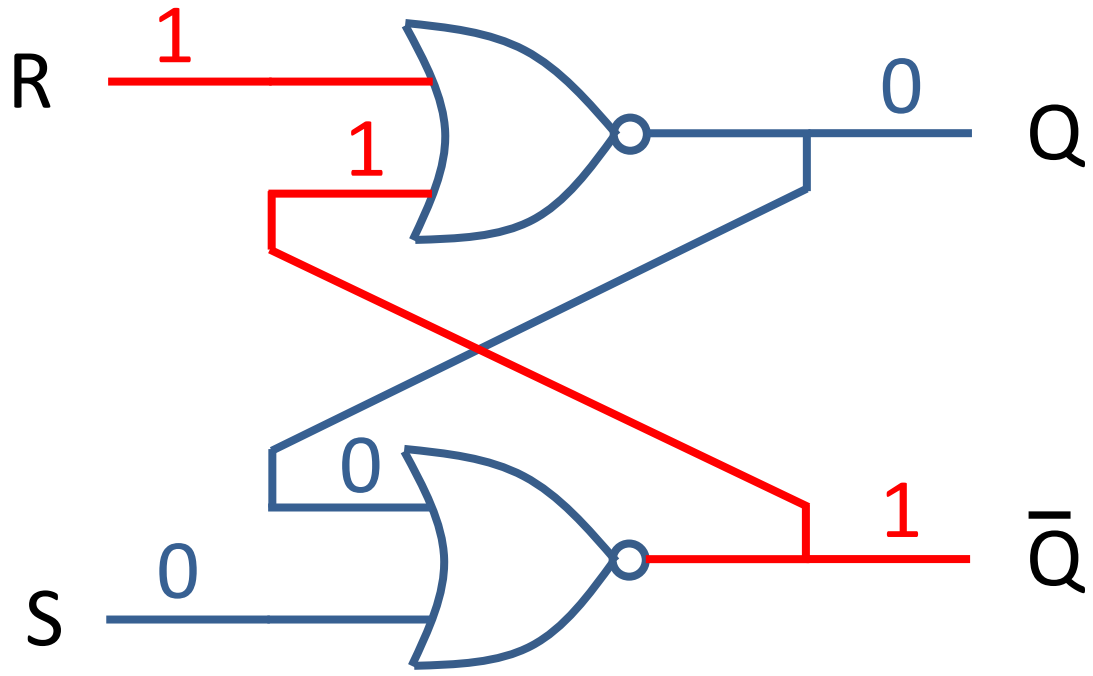
S	R	Q	$\overline{Q}$
0	0	1 0	0 1
0	**1**	**0**	**1**
1	0	1	0
1	1	0	0



SR Latch

- **Reset** state (3/3)

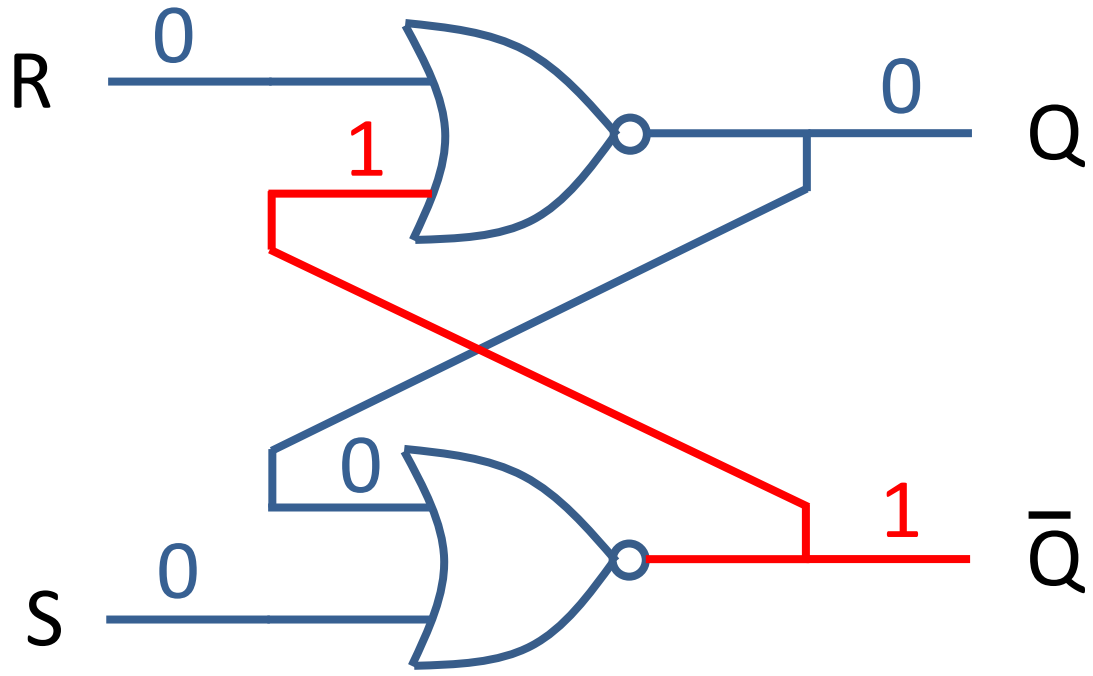
S	R	Q	$\overline{Q}$
0	0	1 0	0 1
0	**1**	**0**	**1**
1	0	1	0
1	1	0	0



SR Latch

- **Normal** state

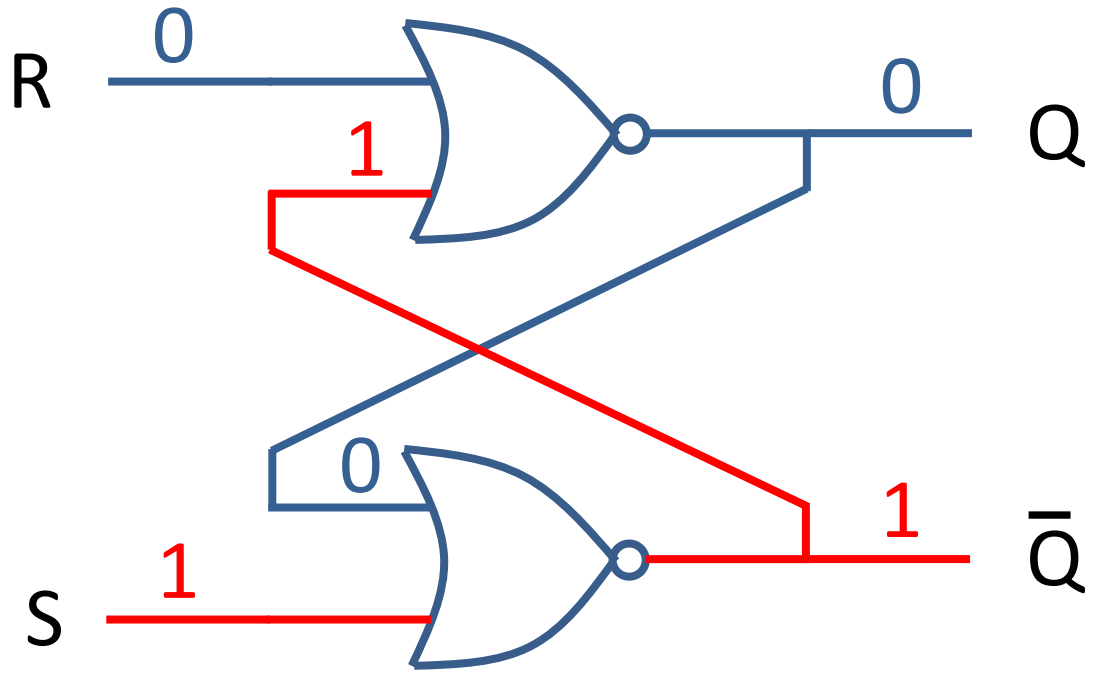
S	R	Q	$\overline{Q}$
0	**0**	1 **0**	0 **1**
0	1	0	1
1	0	1	0
1	1	0	0



SR Latch

- **Set** state (1/3)

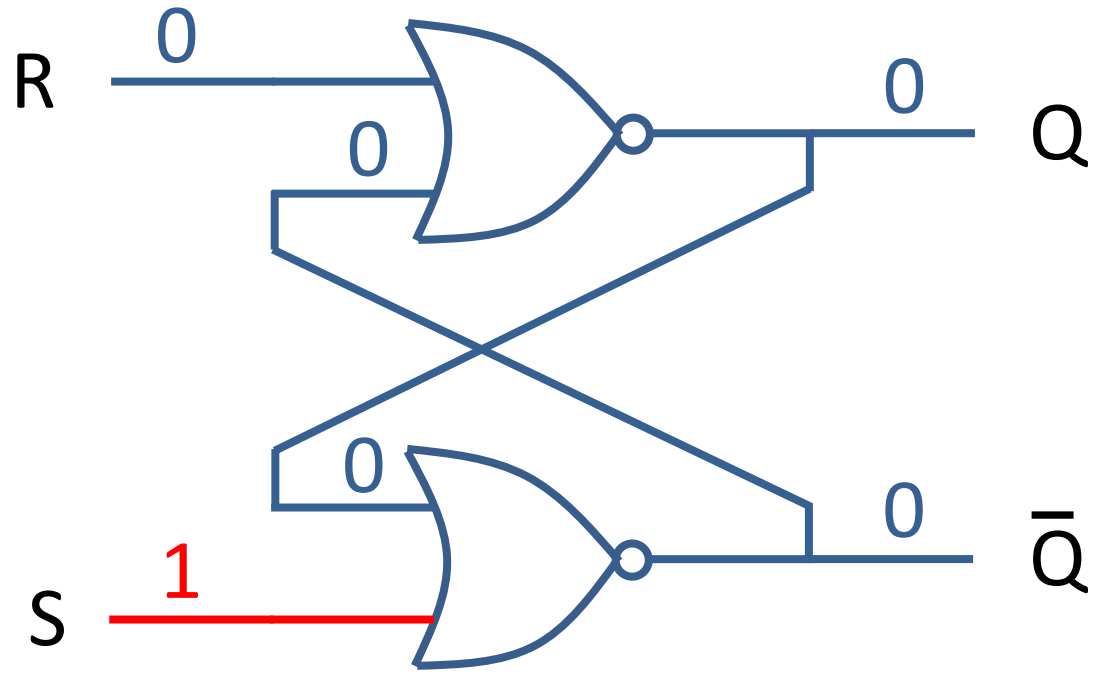
S	R	Q	$\overline{Q}$
0	0	1 0	0 1
0	1	0	1
1	**0**	**1**	**0**
1	1	0	0



SR Latch

- **Set** state (2/3)

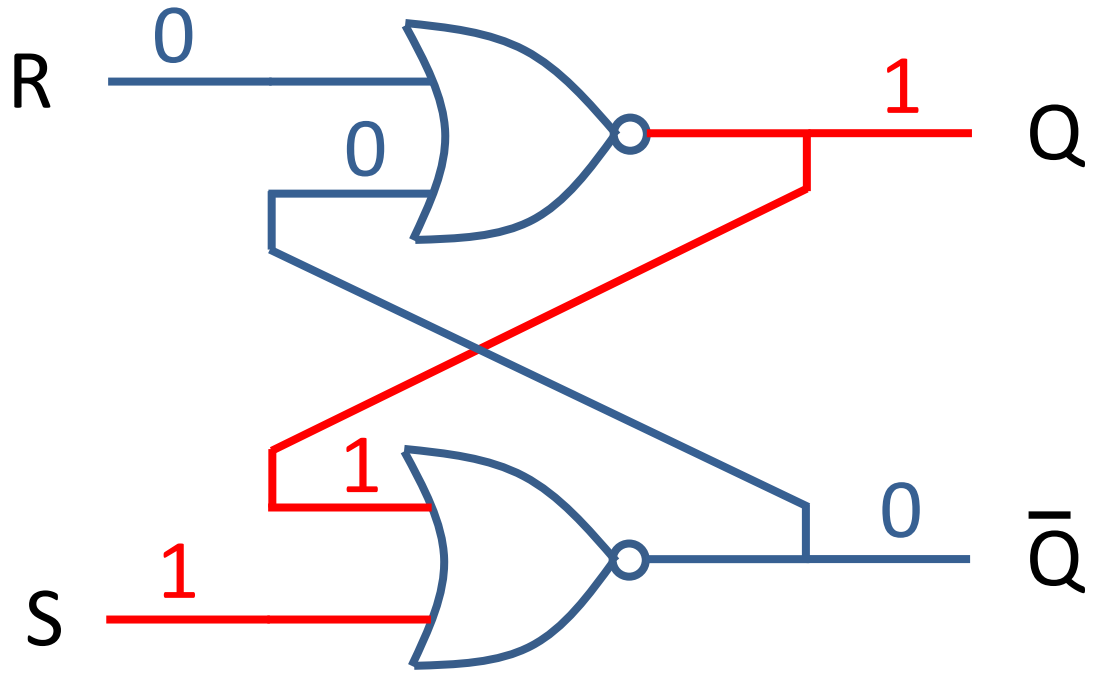
S	R	Q	$\overline{Q}$
0	0	1 0	0 1
0	1	0	1
1	**0**	**1**	**0**
1	1	0	0



SR Latch

- **Set** state (3/3)

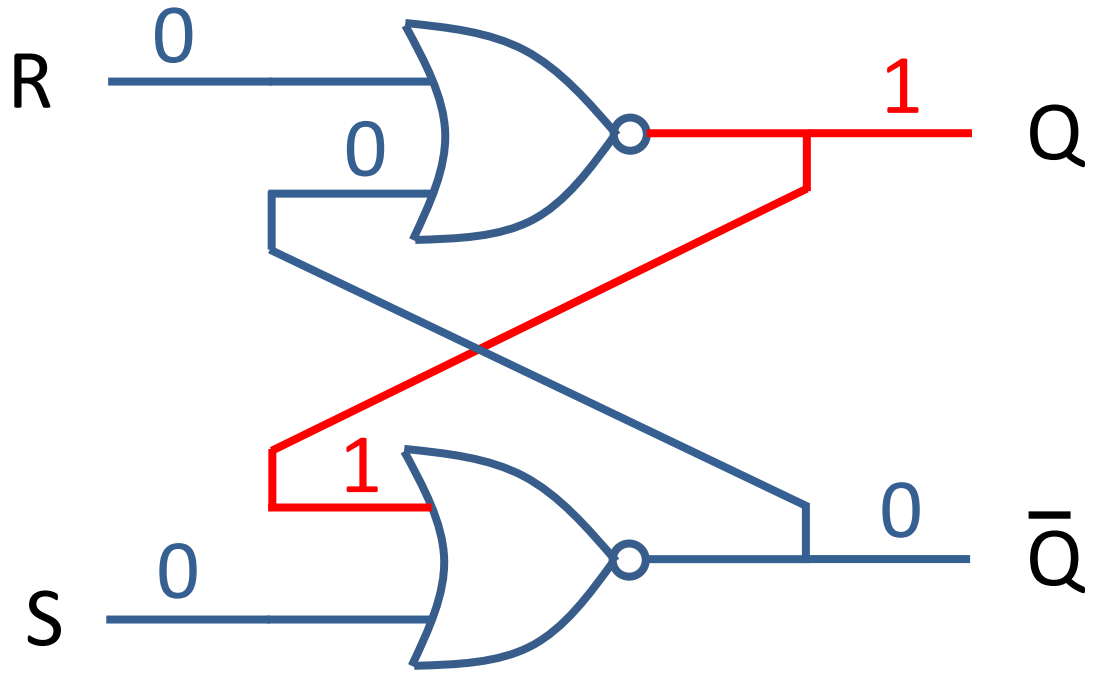
S	R	Q	$\overline{Q}$
0	0	1 0	0 1
0	1	0	1
1	**0**	**1**	**0**
1	1	0	0



SR Latch

- **Normal** state

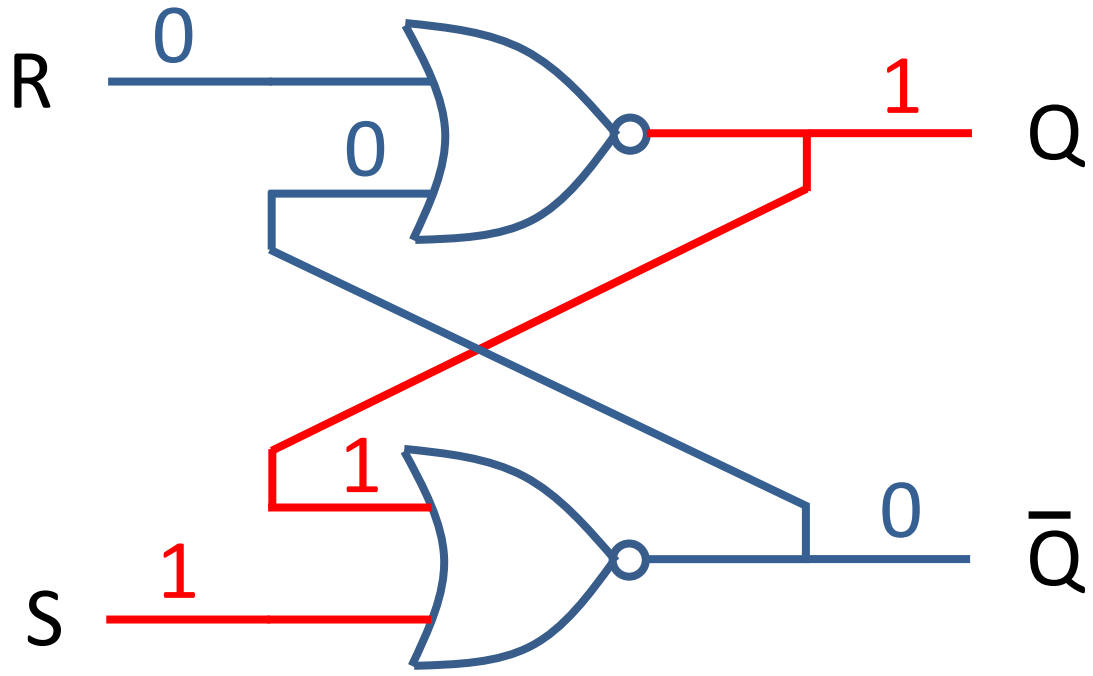
S	R	Q	$\overline{Q}$
0	0	1 0	0 1
0	1	0	1
1	0	1	0
1	1	0	0



SR Latch

- **Set** state (no change)

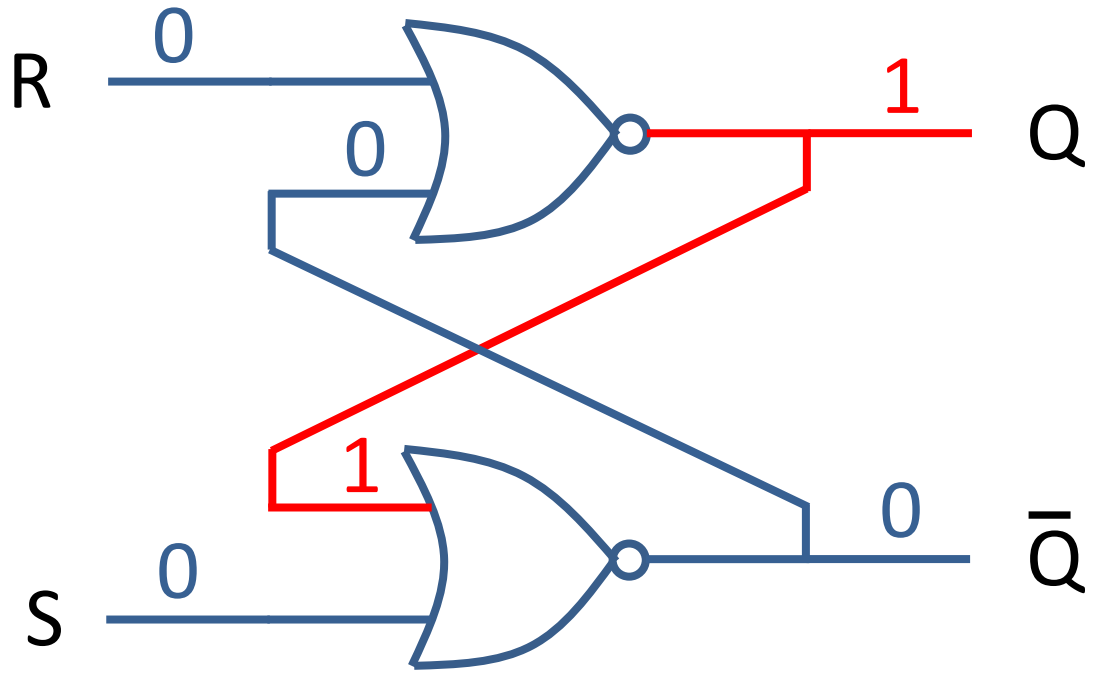
S	R	Q	$\overline{Q}$
0	0	1 0	0 1
0	1	0	1
1	**0**	**1**	**0**
1	1	0	0



SR Latch

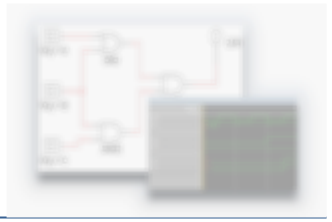
- **Normal** state

S	R	Q	$\overline{Q}$
0	0	1 0	0 1
0	1	0	1
1	0	1	0
1	1	0	0



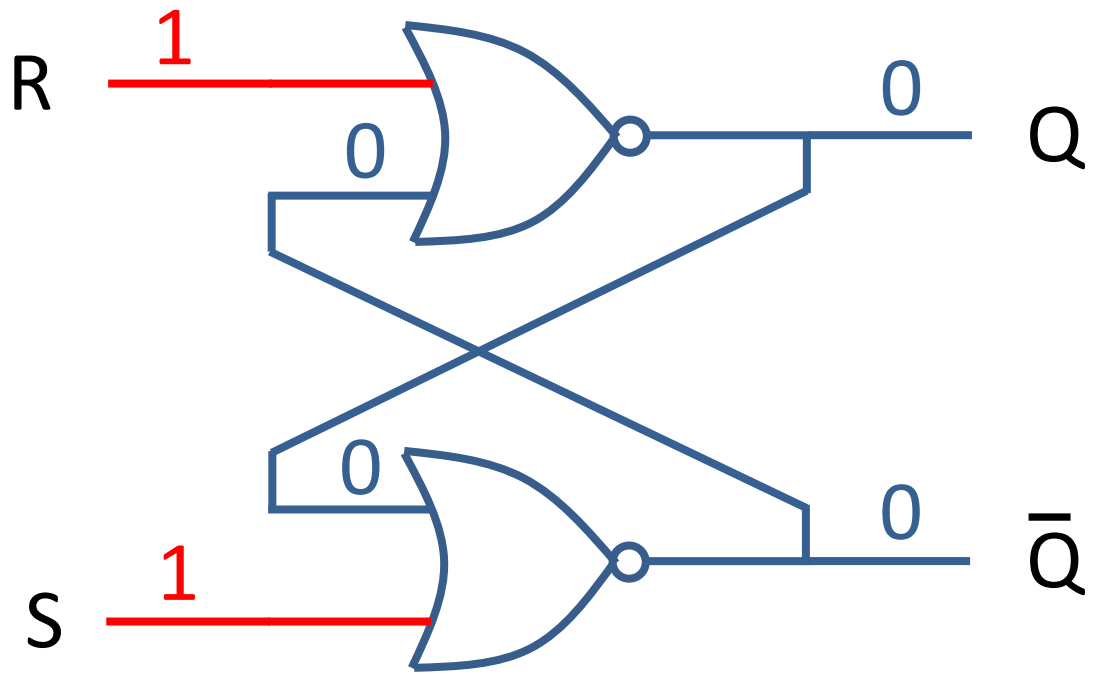
SR Latch

Παραβιάζεται η βασική σχέση των εξόδων και έχουμε: $Q = \overline{Q}$
Δηλαδή οι δύο έξοδοι δεν είναι συμπληρωματικές



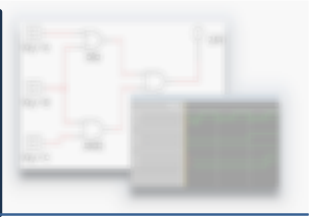
- **Invalid** state

S	R	Q	$\overline{Q}$	
0	0	1 0	0 1	
0	1	0	1	
1	0	1	0	
1	**1**	**0**	**0**	Invalid



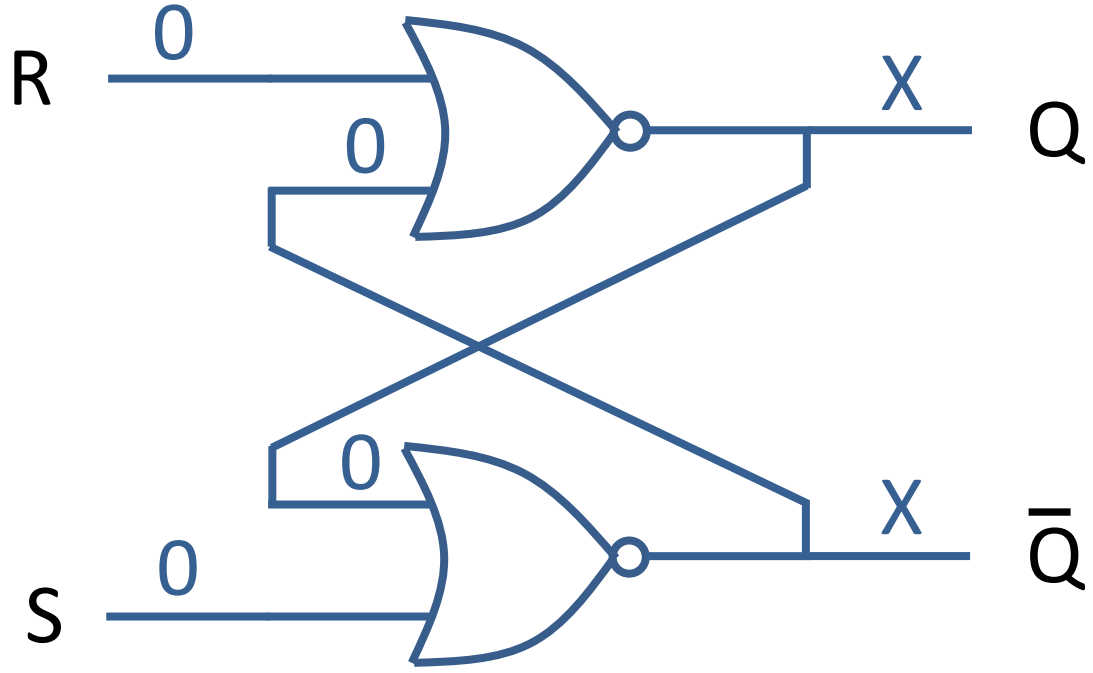
SR Latch

Όταν οι είσοδοι **S** και **R** επιστρέψουν ταυτόχρονα στο 0 μετά την απαγορευμένη κατάσταση, η τελική κατάσταση εξαρτάται από τις καθυστερήσεις διάδοσης των πυλών (race condition). Επομένως, δεν είναι προβλέψιμη από τις λογικές τιμές των εισόδων και μόνο.



- Εάν μετά R = S = 0 <u>ταυτόχρονα</u>: Δεν μπορούμε να ξέρουμε ποια θα είναι η έξοδος! (Race)

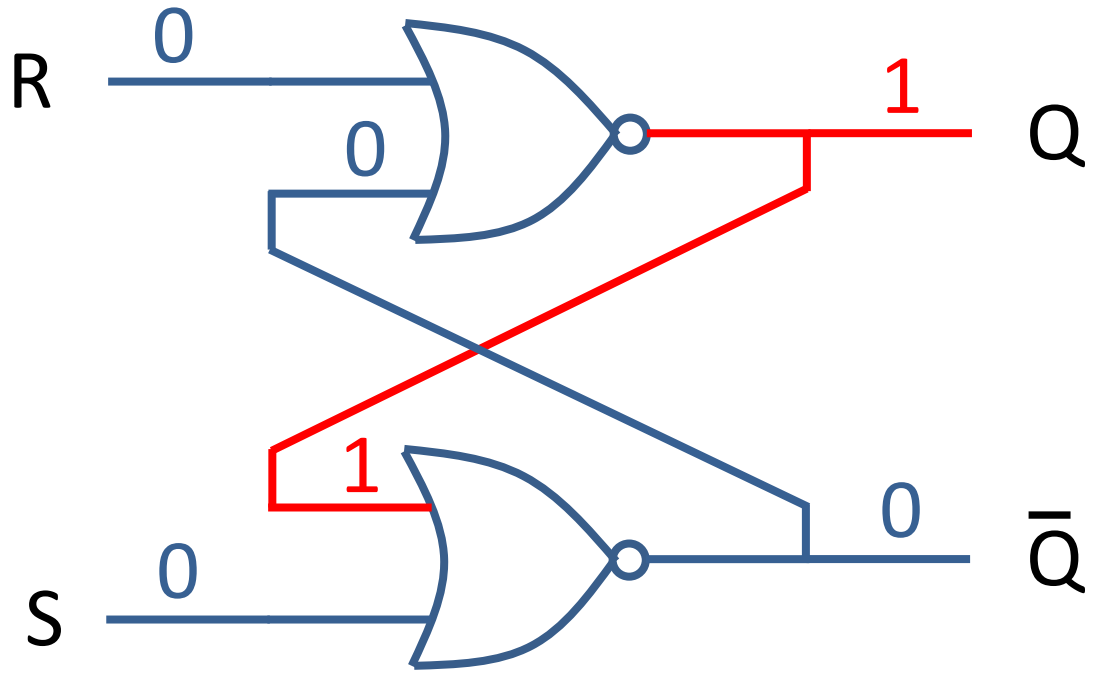
S	R	Q	$\overline{Q}$	
0	**0**	**1** **0**	**0** **1**	**?**
0	1	0	1	
1	0	1	0	
1	1	0	0	Invalid



SR Latch

- **Active-High SR Latch** (NOR-based)

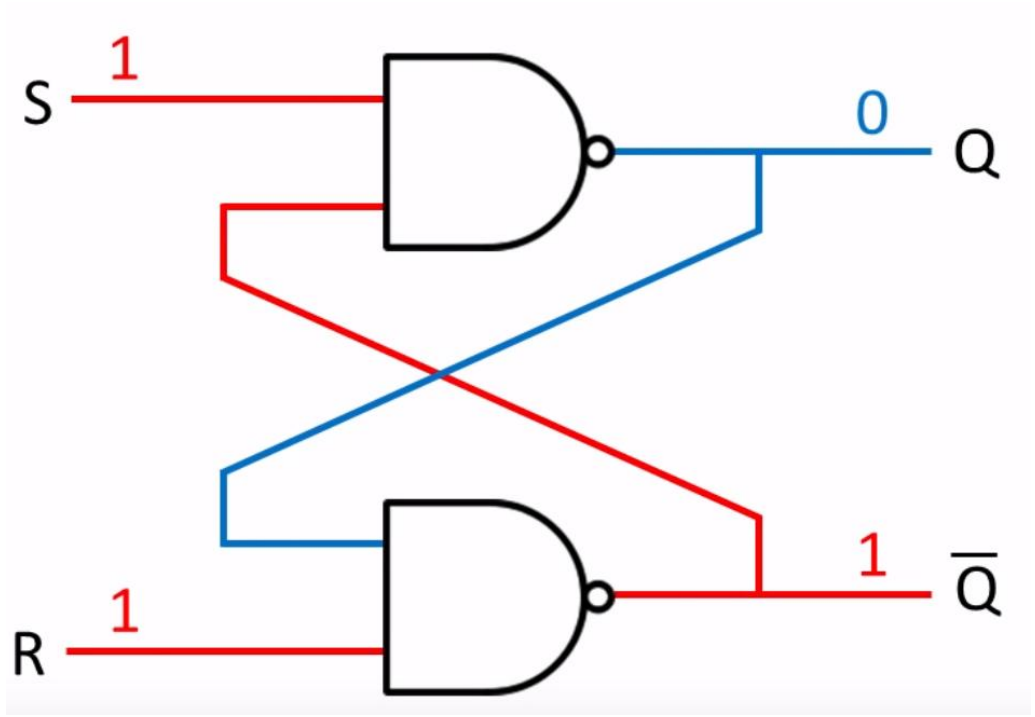
S	R	Q	$\overline{Q}$
0	0	1 0	0 1
0	1	0	1
1	0	1	0
1	1	0	0



SR Latch

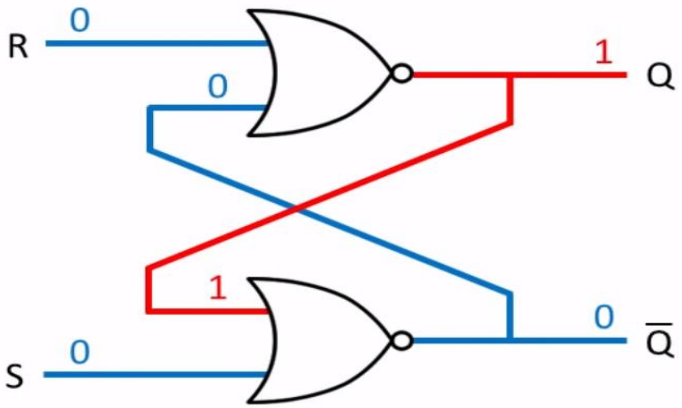
- **Active-Low SR Latch** (NAND-based)

S	R	Q	$\overline{Q}$	
0	0	1	1	Invalid
0	1	1	0	
1	0	0	1	
1	**1**	**0** 1	**1** 0	

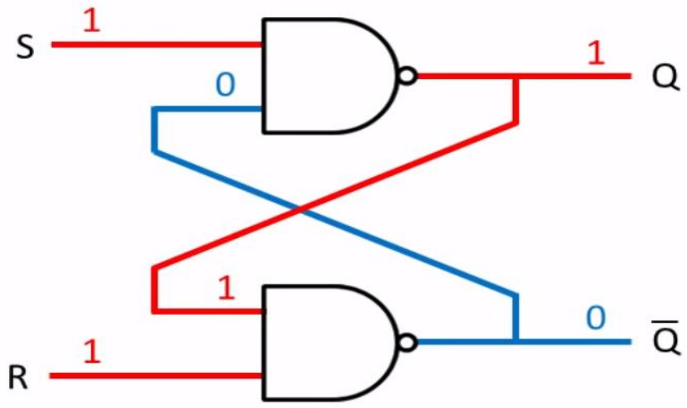


SR Latch

- Εκτελούν την ίδια λειτουργία, αλλά ελέγχονται με διαφορετικό τρόπο.
- Στο **NOR SR latch** οι είσοδοι S και R ενεργοποιούνται με 1, ενώ στο **NAND SR latch** ενεργοποιούνται με 0. Και τα δύο αποθηκεύουν ένα bit πληροφορίας.



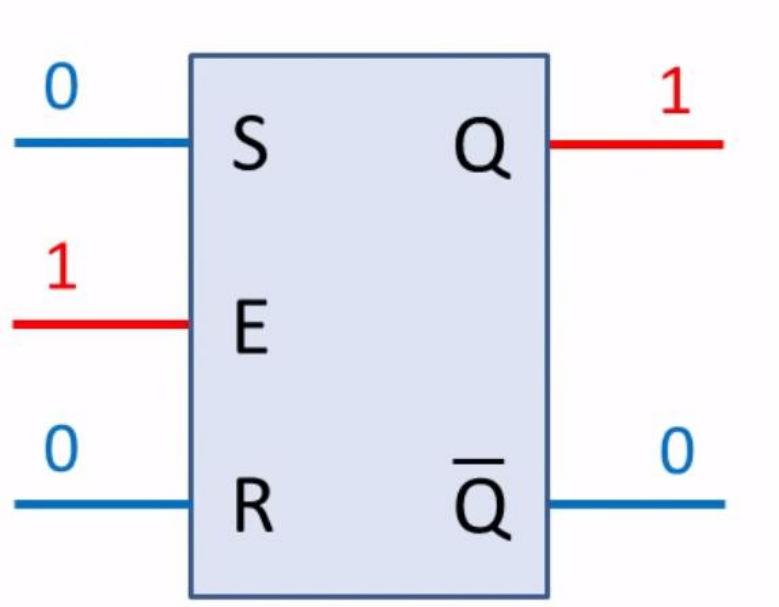
S	R	Q	$\overline{Q}$	
0	0	1 0	0 1	
0	1	0	1	
1	0	1	0	
1	1	0	0	Invalid



S	R	Q	$\overline{Q}$	
0	0	1	1	Invalid
0	1	1	0	
1	0	0	1	
1	1	0 1	1 0	

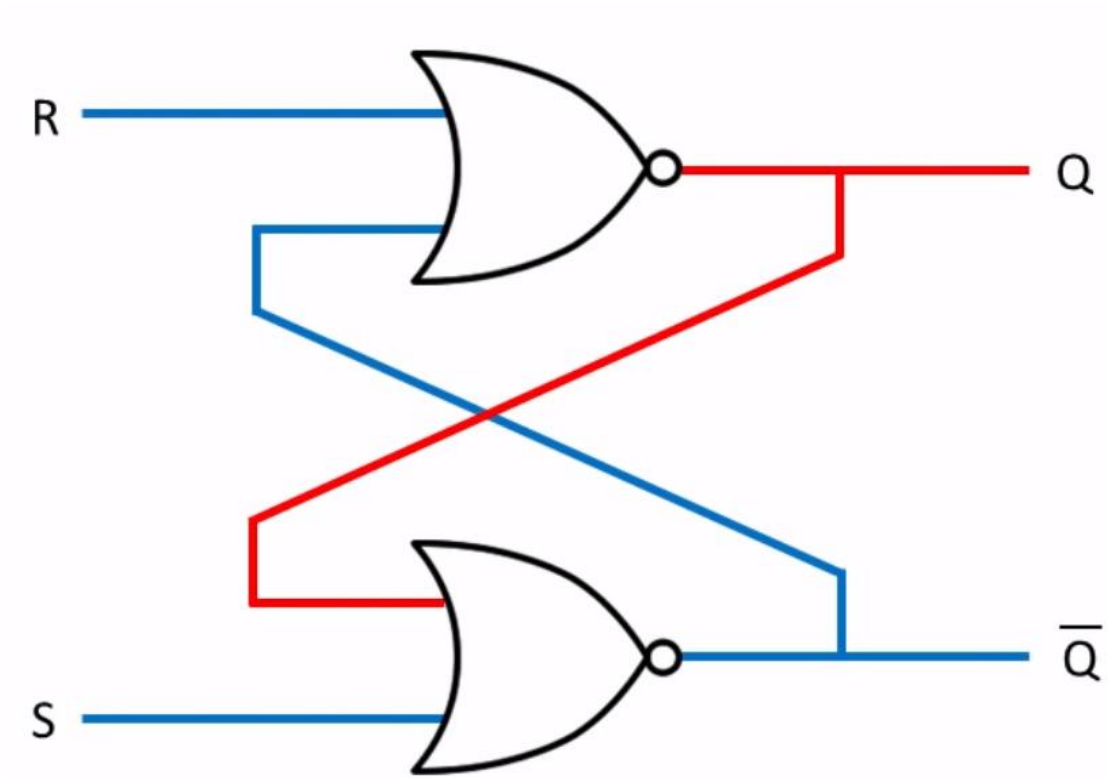
Gated SR Latch

- Το **Gated SR Latch** μπορεί να αλλάξει την αποθηκευμένη κατάστασή του μόνο όταν το σήμα **Enable** (E) βρίσκεται στην ενεργή στάθμη του. Όταν E=0, διατηρεί την προηγούμενη κατάσταση ανεξάρτητα από τις τιμές S και R.



Gated SR Latch

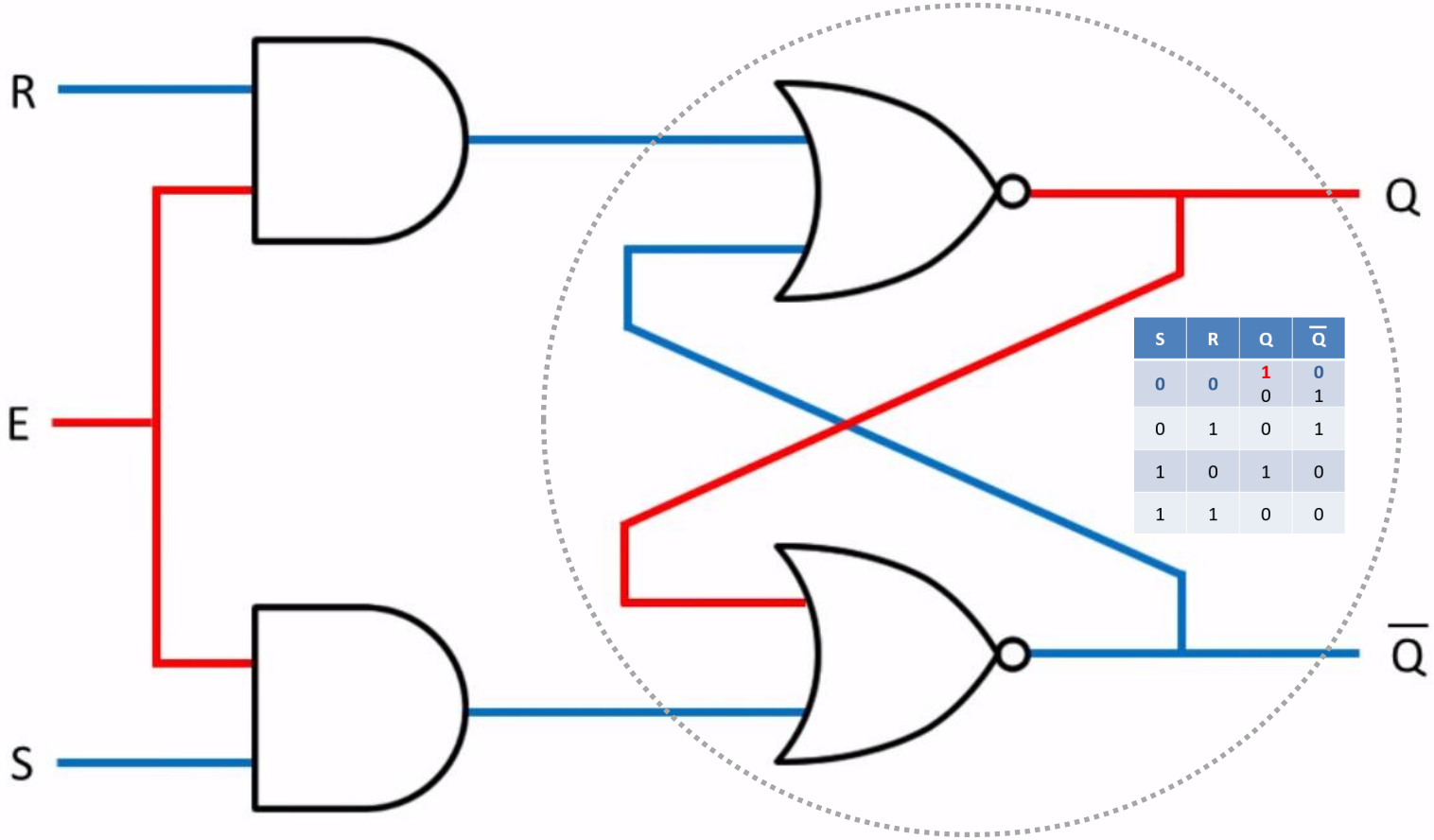
- **Ungated Active-High** SR Latch (NOR-based)



S	R	Q	$\overline{Q}$
0	0	1 0	0 1
0	1	0	1
1	0	1	0
1	1	0	0

Gated SR Latch

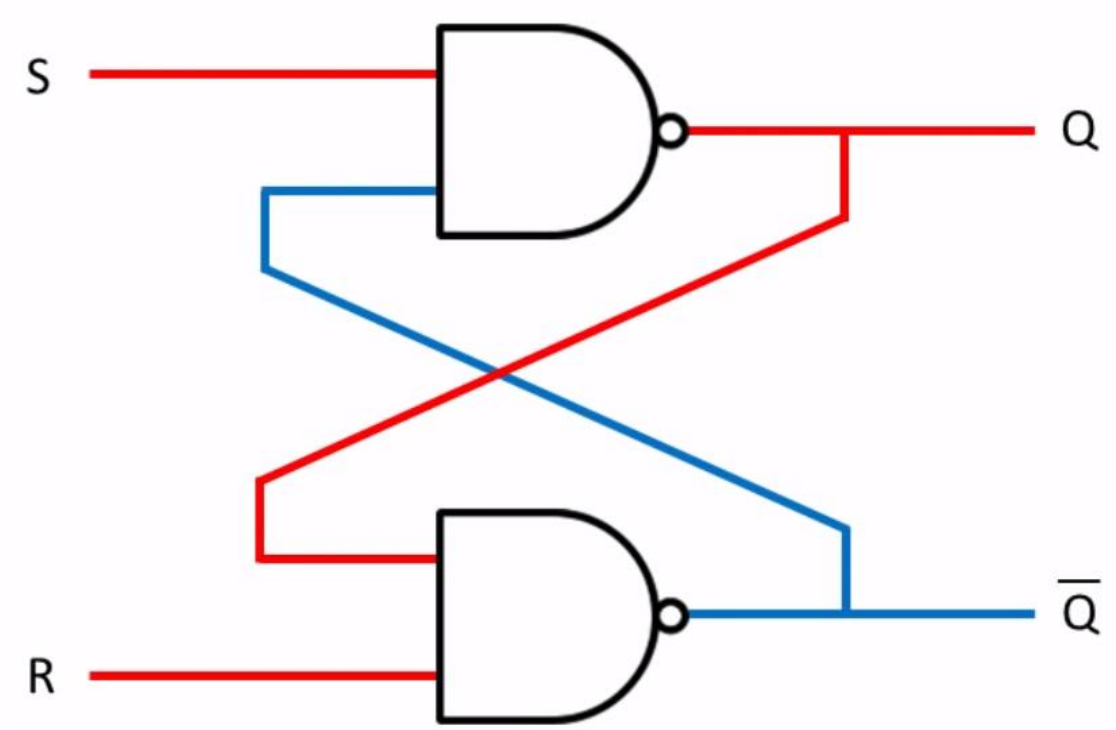
- **Gated Active-High** SR Latch (NOR-based). Η είσοδος E ενεργοποιεί ή όχι το latch



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	Invalid

Gated SR Latch

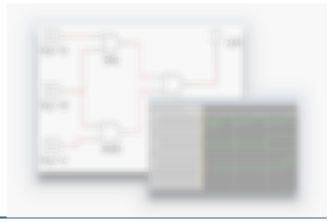
- **Ungated Active-Low** SR Latch (NAND-based)



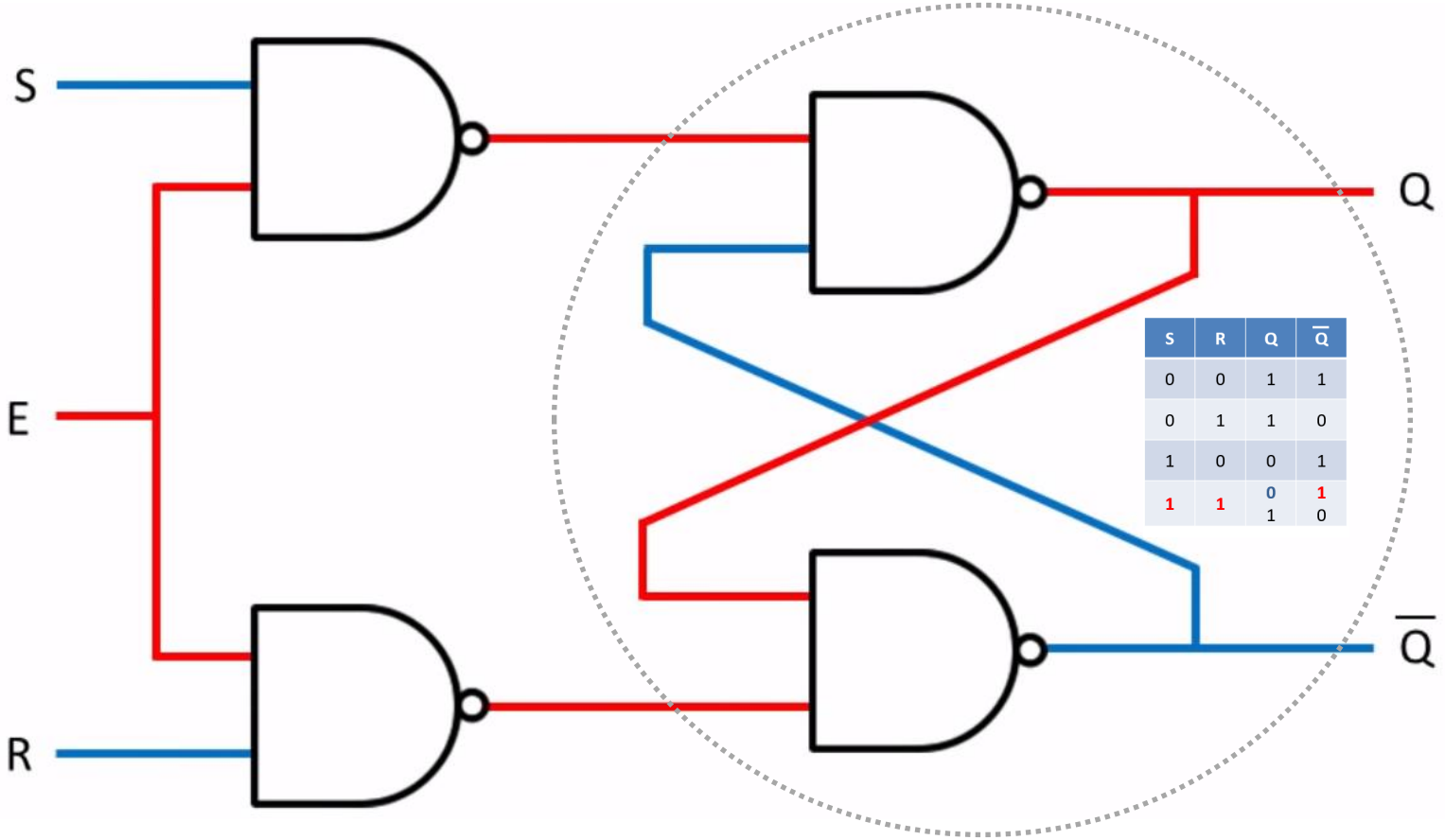
S	R	Q	$\overline{Q}$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0 1	1 0

Gated SR Latch

Παρότι το εσωτερικό NAND SR latch έχει **active-low** εισόδους, το συνολικό gated SR latch έχει **active-high** εξωτερικές εισόδους S, R και E.



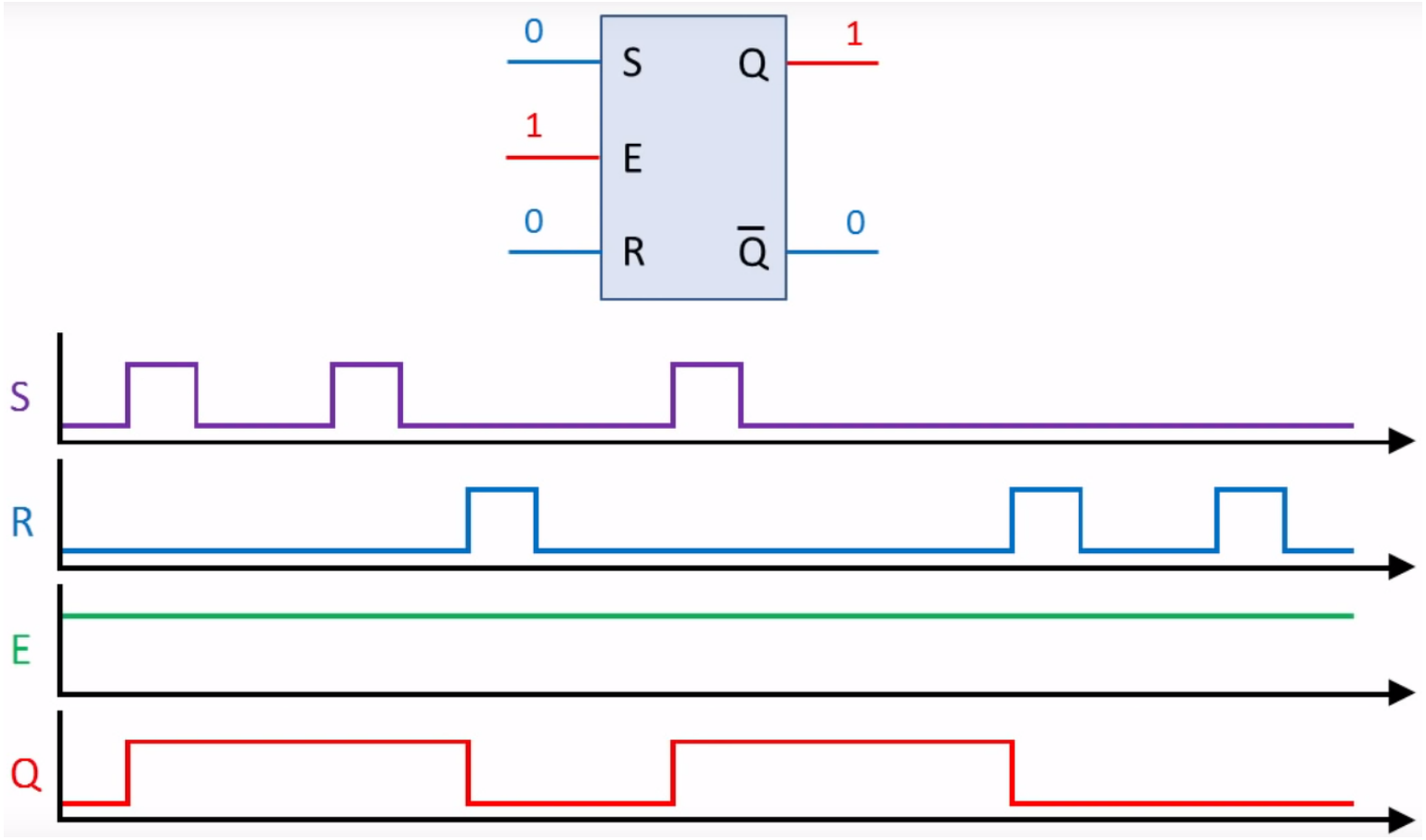
- **Gated Active-High** SR Latch (NAND-based). Η είσοδος E ενεργοποιεί ή όχι το latch



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	Invalid

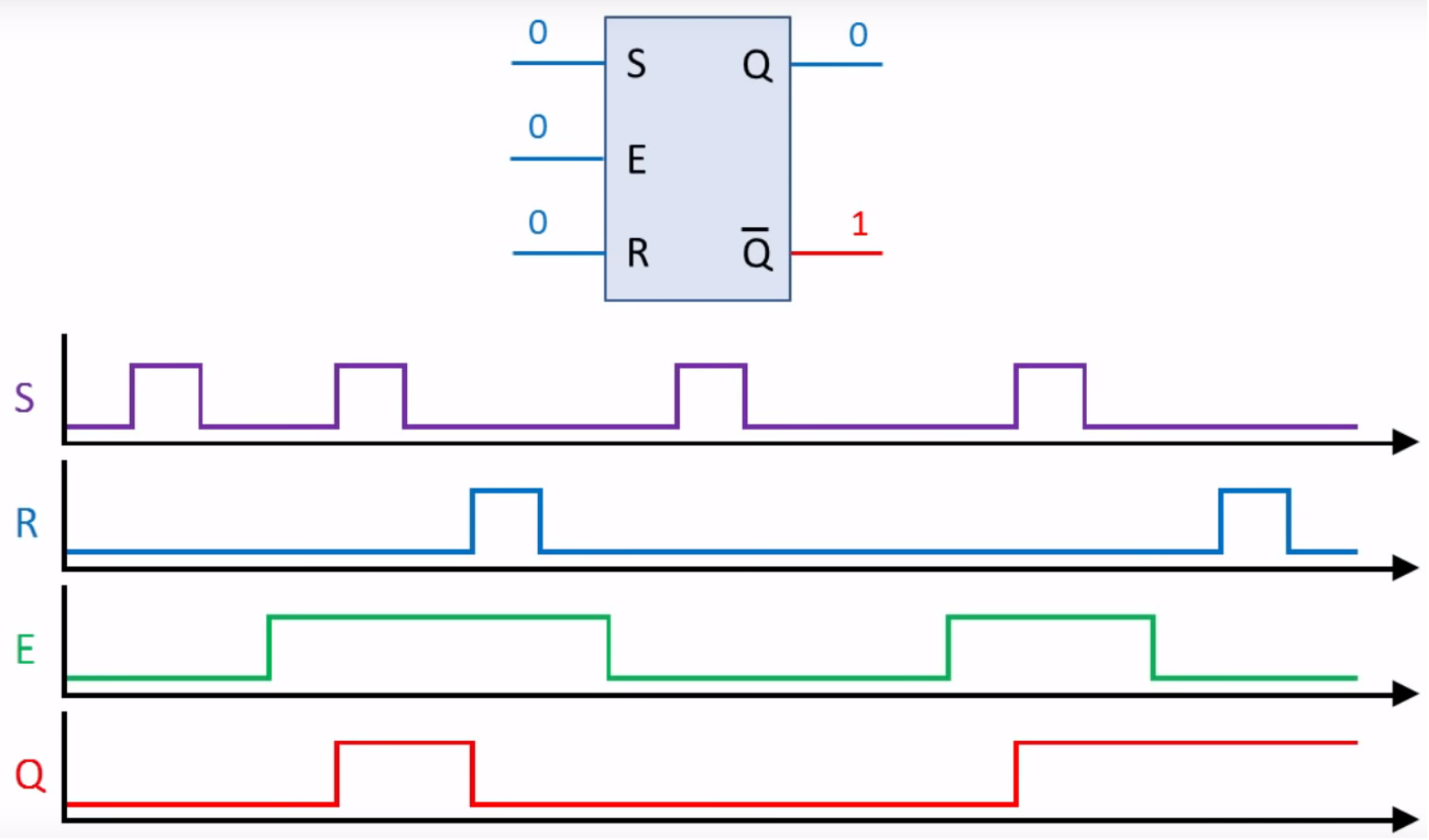
Gated SR Latch

- Παράδειγμα 1: Το latch είναι <u>πάντα enable</u>.



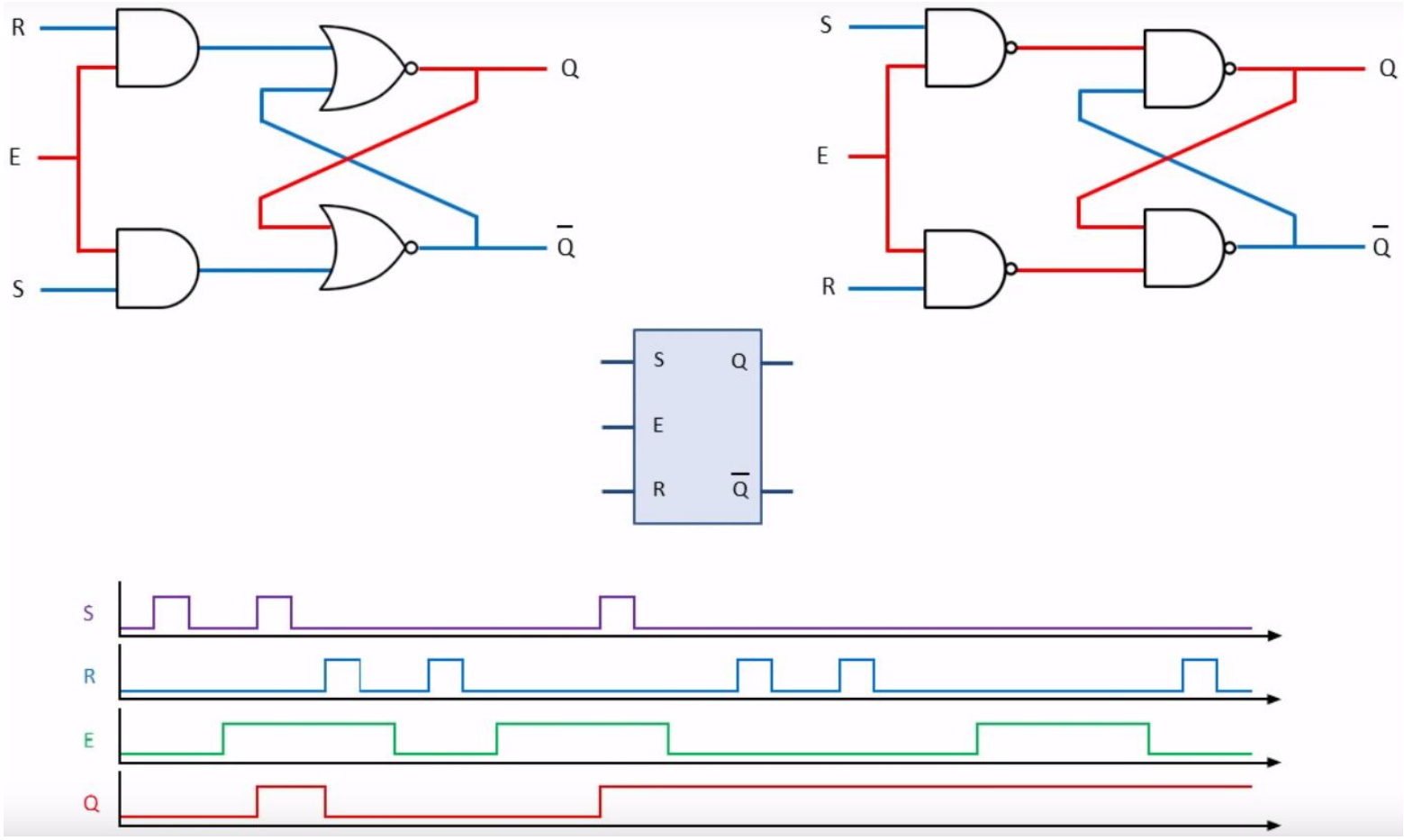
Gated SR Latch

- Παράδειγμα 2: Το σήμα enable μεταβάλλεται.



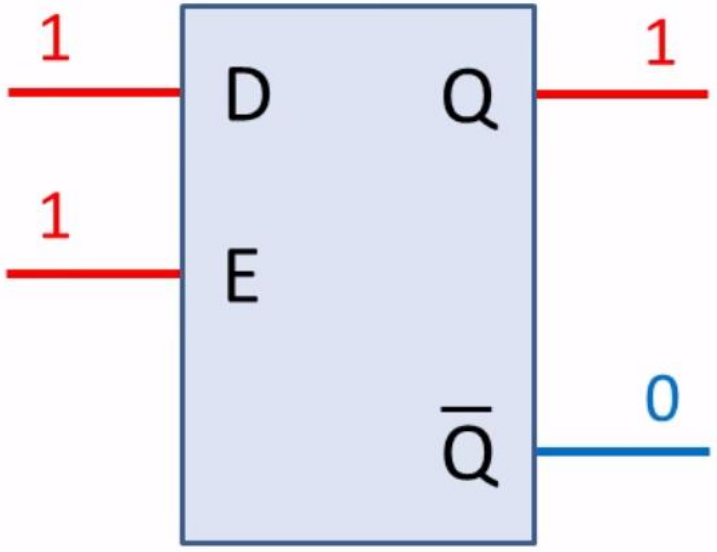
Gated SR Latch

- Εκτελούν την ίδια λειτουργία.



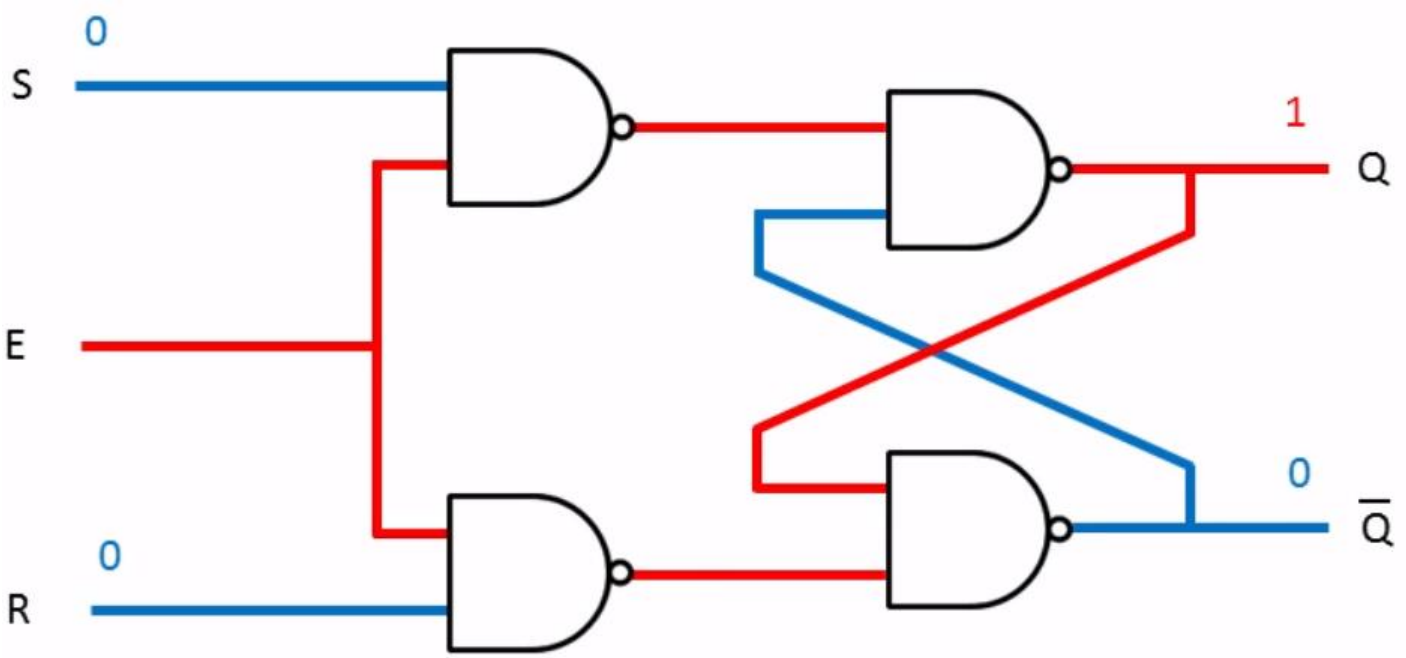
Gated D Latch

- 1-bit memory device.
- Το D latch αποθηκεύει την τιμή της εισόδου D όταν απενεργοποιείται το Enable και τη διατηρεί μέχρι την επόμενη ενεργοποίησή του.
- Όσο $E=1$, η έξοδος Q ακολουθεί την είσοδο D



Gated D Latch

- **Gated Active-High** SR Latch (NAND-based).

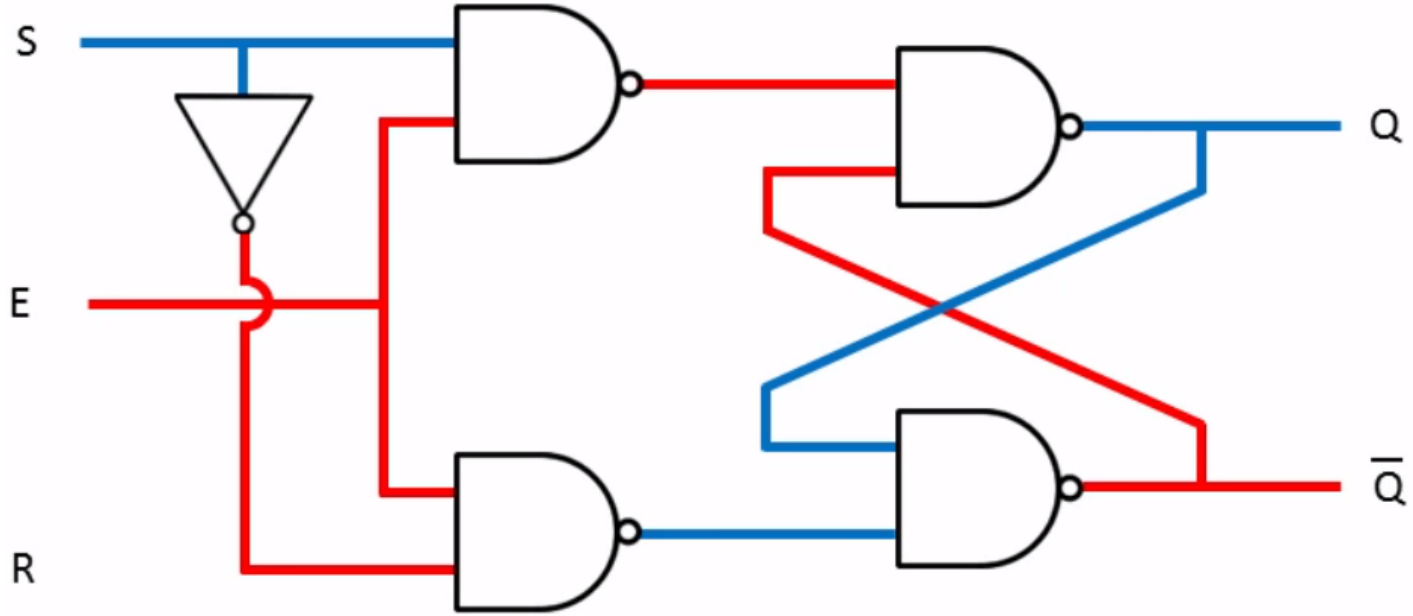


E	S	R	Q(t+1)
0	x	x	**Q(t)**
1	0	0	**Q(t)**
1	0	1	**0**
1	1	0	**1**
1	1	1	Invalid

Gated D Latch

- Συνδέουμε τις εισόδους S και R μέσω ενός αντιστροφέα, ώστε $R = \overline{S}$

  Έτσι, όταν $E=1$, αποκλείεται ο συνδυασμός $S=R=1$ και αποφεύγεται η απαγορευμένη κατάσταση του SR latch.



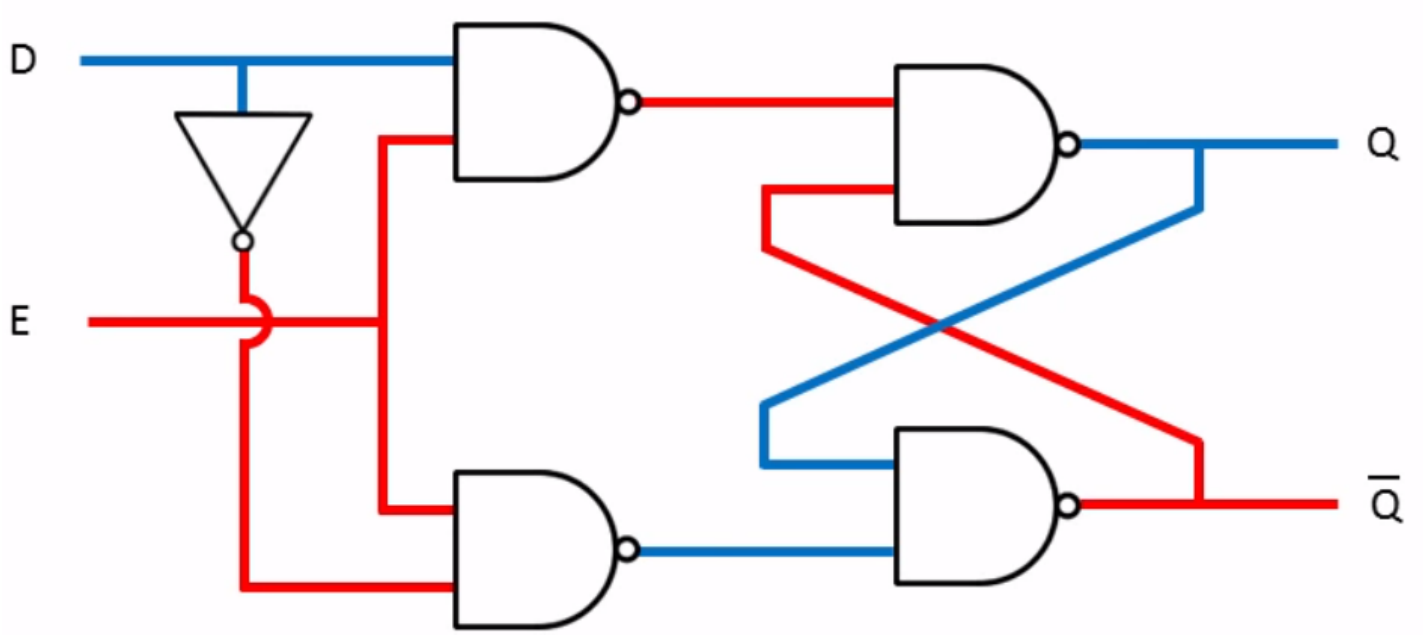
E	S	~~X~~	Q(t+1)
0	x	x	**Q(t)**
~~1~~	~~0~~	~~0~~	~~**Q(t)**~~
1	0	1	**0**
1	1	0	**1**
~~1~~	~~1~~	~~1~~	~~Invalid~~

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

Gated D Latch

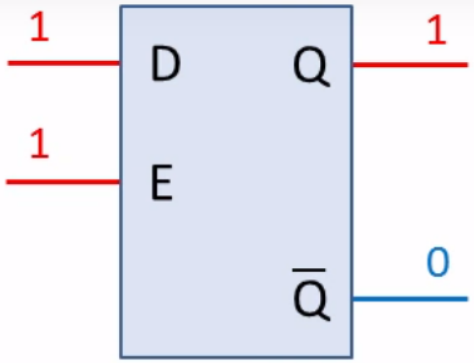
- Αλλάζουμε το όνομα της εισόδου από S σε **D**.

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

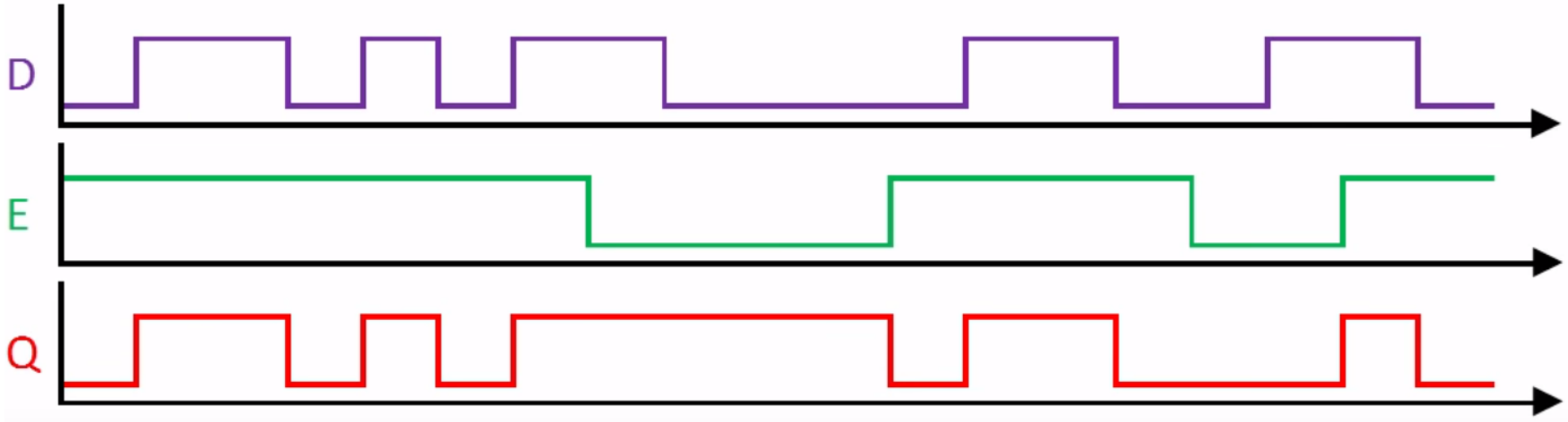


Gated D Latch

Το **D latch** είναι **level-sensitive** και όχι edge-triggered. Όσο το **Enable είναι ενεργό**, κάθε αλλαγή στην είσοδο D μπορεί να εμφανιστεί στην έξοδο Q.

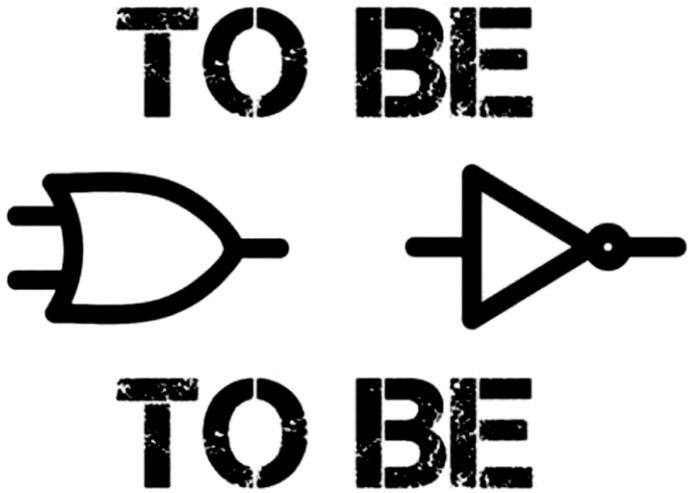


E	Λειτουργία	Έξοδος Q
0	Hold	Διατηρεί την προηγούμενη τιμή
1	Transparent	Ακολουθεί την D



Ευχαριστώ για την προσοχή σας!

- **Ερωτήσεις / Απορίες ;**



Επικοινωνία: ece119.uth@gmail.com