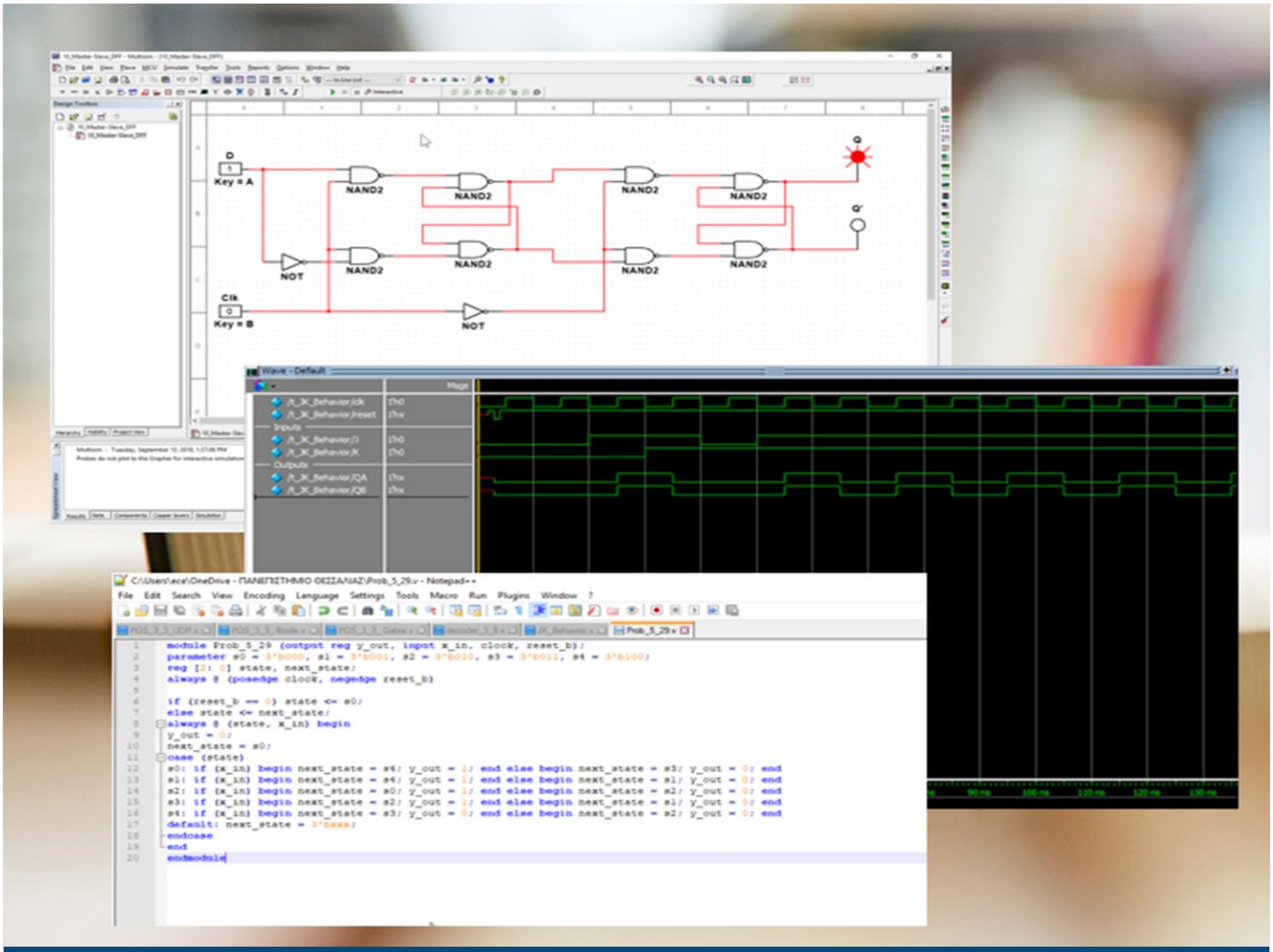
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ



ECE119 Ψηφιακή Σχεδίαση

Εργαστηριακές ασκήσεις, Multisim - Verilog

Lab 8: Latches and Sequential Logic Circuits

Καραμπερόπουλος Δημήτρης

Σεπτέμβριος 2026

Lab 8

Required Tools and Technology 1
Lab 8: Latches and Sequential Logic Circuits 2
Learning Objectives 2
Expected Deliverables 2
8.1 Theory and Background 3
8.2 Implement: Building a Gated D Latch Circuit using Logic Gates 9
8.3 Exercise: Verifying the Gated SR Latch Characteristic Table 12
8.4 Conclusion 15
8.5 Exercise: HDL - Verilog, D-Latch 17

Required Tools and Technology

Software: NI Multisim 14.0 or newer	✓ **Install Multisim:** http://www.ni.com/gate/gb/GB_ACADEMICEVALMULTISIM/US ✓ **View Help:** http://www.ni.com/multisim/technical-resources/
MultisimLive	✓ https://www.multisim.com/

Lab 8: Latches 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:

1. Understand the difference between synchronous and asynchronous sequential circuits.
2. Test and compare circuits for D latches using both logic gates and latches.
3. Confirm the characteristic table of a gated SR latch.
4. Observe and articulate the difference between D latches and SR latches.
5. Explain the transparent and hold behavior of a D latch and implement a multi-bit D latch using Verilog.

Expected Deliverables

In this lab you will collect the following deliverables:

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

Your instructor may expect you to complete a lab report. Refer to your instructor for specific requirements or templates.

8.1 Theory and Background

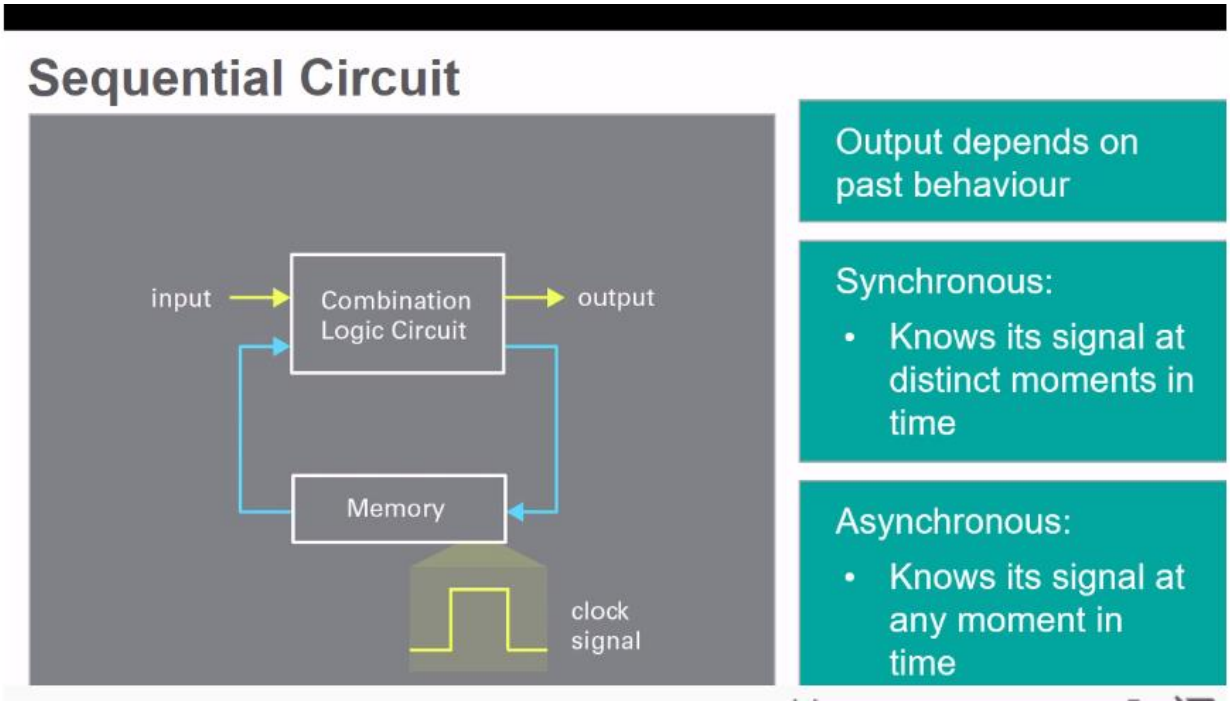


Figure 8-1 Video. View the video here: https://youtu.be/RMY4rTyVso0

Video Summary

- A sequential circuit's output depends on the present combination of inputs, and the past behavior of the circuit
- There are two classes of sequential circuits: synchronous and asynchronous
- The clock signal is a rectangular pulse train

Sequential Circuit

A *sequential circuit* is one whose output depends not only on the present combination of the inputs, but also on the past behavior of the circuit.

- The basic building block of sequential circuits is the bistable, a circuit having two stable states.
- The state of a sequential circuit is represented by a set of bits, called state variables, containing all the information about the past necessary to explain the future behavior of the circuit.
- The outputs of sequential circuits are a function of both their inputs and of the history of these inputs.

The block diagram of a sequential logic circuit is presented below.

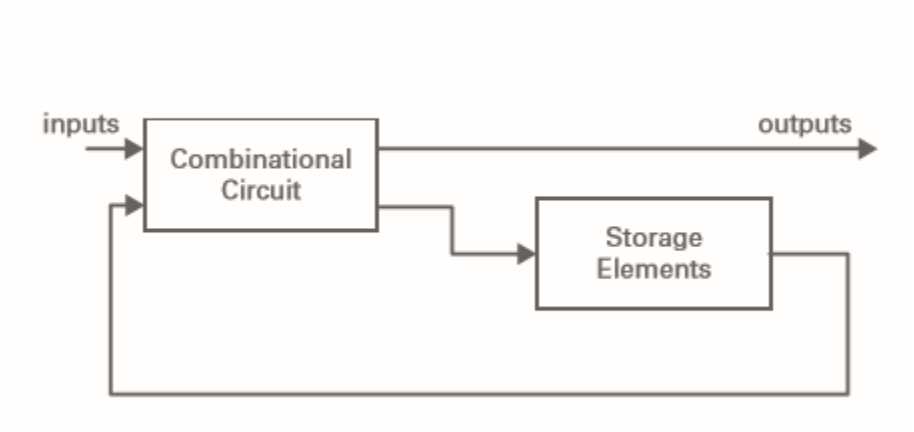


Figure 8-2 Sequential logic circuit block diagram

There are two main classes of sequential circuits, depending on the timing of their signals:

1. *Synchronous sequential circuits* – their behavior can be defined knowing its signals at distinct moments in time.
2. *Asynchronous sequential circuits* – their behavior can be defined knowing the input signals at any moment in time and the order in which they change.

This lab introduces sequential circuits through basic and gated latches, focusing on how stored state is controlled by input and enable signals. Throughout this lab, synchronous sequential circuits will be simply called sequential circuits.

Clock Signals

The clock signal is a rectangular pulse train. It has a precise pulse width and a precise interval between pulses, called clock cycle time.

- These signals are used by sequential circuits by synchronizing the activity within the circuit and for the synchronizing the update of stored values.
- Most sequential circuits change their state on one of the edges of the clock pulses (rising or falling edge), being referred to as *edge-triggered***.**
- The timing events is important when dealing with sequential logic circuits. Therefore, in this case, besides the truth table, one should also know the order in which events occurred, i.e. the *logic timing diagram***.**
- The table presenting the sequential circuits operation is often called a *characteristic table* rather than a truth table, since it does not represent a combinational circuit.

- Many synchronous sequential circuits use edge-triggered flip-flops, which update their stored state on a clock edge. Latches, however, are level-sensitive and may update their state throughout the active level of the enable signal.
- An example pulse train is shown below (left), as well as an example timing diagram for an AND gate (right).

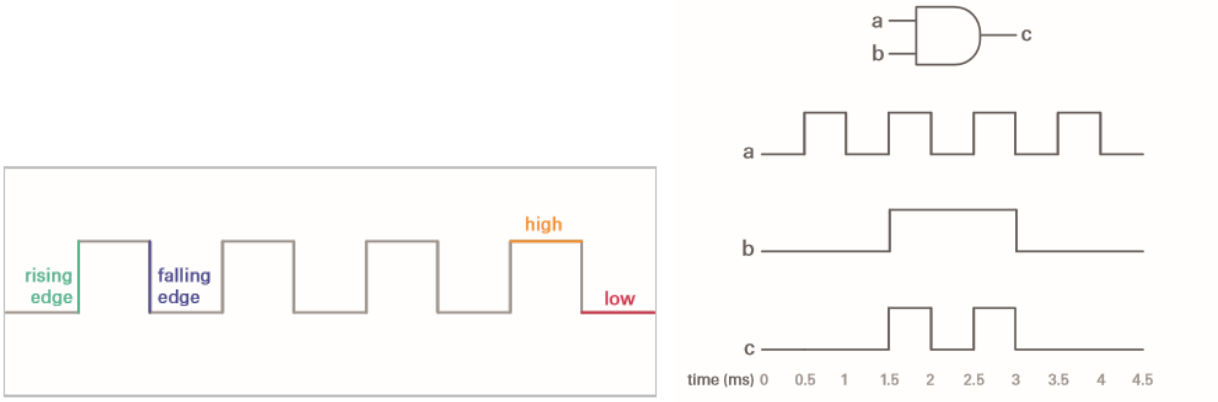


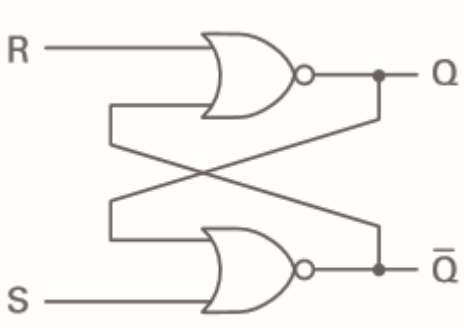
Figure 8-3 Pulse train (left) and timing diagram for an AND gate (right)

The SR Latch

A *latch* is a storage element that operates with signal levels rather than signal transitions.

- A latch is a level sensitive device and the basic building block of flip-flops, which will be presented later.
- One of the simplest sequential circuits is the basic latch.

The basic latch is presented in the figure below and has the functionality described by the accompanying characteristic table.



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

Figure 8-4 Basic latch (left) and characteristic table (right)

- The circuit has two inputs, S (Set) and R (Reset), and two outputs, Q and $\bar{Q}$, with $\bar{Q}$ the complement of Q.

- The state of the latch can be controlled through the S and R inputs, which set and reset the output Q.
- The word “set” denotes the process of making the output Q a logic value 1, while “reset” denotes the process of making the output Q a logic value 0.

Note: The output of the circuit depends not only on the current inputs, but also on the previous value of the output.

- When both the inputs are 0, the output remains unchanged.
- A pulse on the S input sets the output to 1 and a pulse to R resets the latch (Q=0)
- When S=R=1, both outputs become 0, violating their complementary relationship. When both inputs return to 0 simultaneously, the final state may be unpredictable due to propagation delays.
    - In practical applications, setting both inputs to 1 is forbidden.

The symbols for a SR latch are presented in the figure below

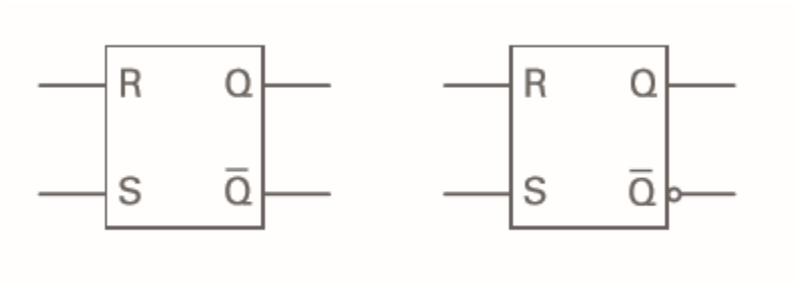


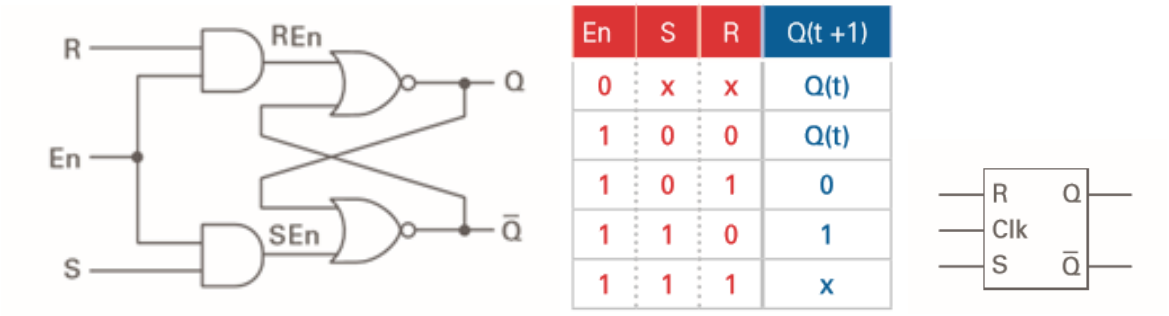
Figure 8-5 SR latch symbols

The SR latch is bistable element with one bit of state stored in Q. Its state can be controlled through the two inputs, S and R. When none of the two inputs is asserted, the state of the circuit remains unchanged. The SR latch represents the basic element for most static memory structures.

The Gated SR Latch and D Latch

An ***ungated SR latch*** can change its state whenever the S or R input is asserted, independently of a clock signal.

- Its operation can be modified in such a way that an input signal *(ENABLE)* controls the moment when the state of the latch changes.
- The circuit is called a gated SR latch.
- The ENABLE signal can be used as an ON/OFF signal.
- The characteristic table (center) presents the operation of the gated SR latch. Q(t) represents the current value of the output and Q(t+1) represents the next state.
- The SR latch circuit is shown below (left) along with its graphical signal (right).



En	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

Figure 8-6 latch circuit (left), characteristic table (center), and graphical symbol (right)

*The **D latch*** eliminates the undesirable condition of the indeterminate state in the SR latch.

- It ensures that the inputs S and R are never equal to 1 in the same time.
- The circuit has only two inputs, D and Clk or En.
- The D input stands for data.
- The D latch circuit is presented in the figure below (left), along with the D latch characteristic table (center) and graphical symbol (right).



Clk	D	Q(t +1)	
0	x	Q(t)	no change
1	0	0	reset state
1	1	1	set state

Figure 8-7 D latch circuit (left), latch characteristic table (center), and graphical symbol (right)

8-1 How are sequential circuits different from combinational logic circuits?

a) Sequential circuits are different from combinational logic circuits as they depend on their previous states in addition to their current inputs.

b) Sequential circuits are different from combinational logic circuits as they depend only on their current inputs.

8-2 What is the difference between synchronous and asynchronous sequential circuits?

a) Asynchronous sequential circuits allow their behavior to be defined knowing its signals at distinct moments in time. Synchronous sequential circuits allow their behavior to be defined knowing the input signals at any moment in time and the order in which they change.

b) Synchronous sequential circuits allow their behavior to be defined knowing its signals at distinct moments in time. Asynchronous sequential circuits allow their behavior to be defined knowing the input signals at any moment in time and the order in which they change.

8.2 Implement: Building a Gated D Latch Circuit using Logic Gates

D Latch

Instructions:

- Launch Multisim.
- Build the following D latch circuit using logic gates:

Note: *E* is a digital constant set a **high**
enable is a button **Electro_Mechanical / FLOW_NO**

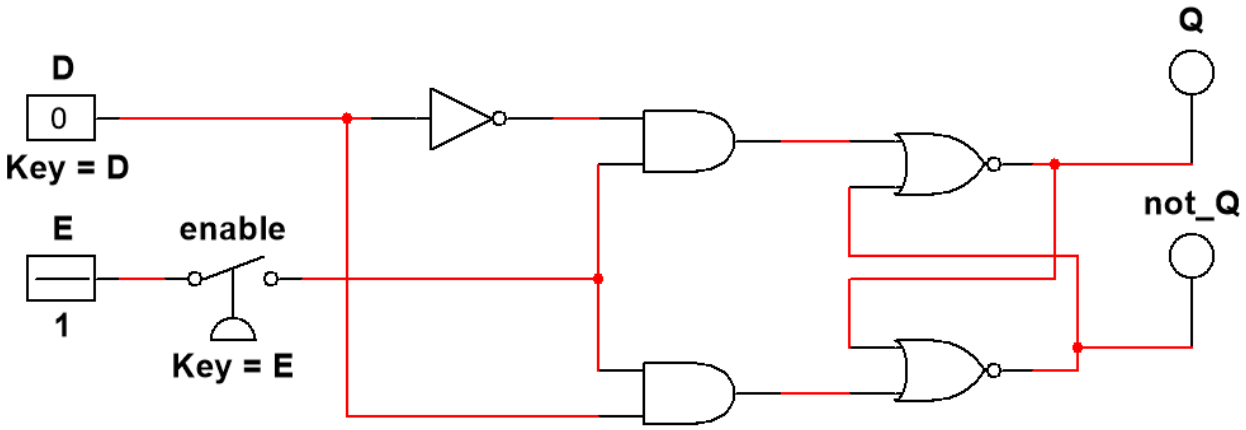


Figure 8-8 D latch circuit diagram

- If you use MultisimLive you can build the following:

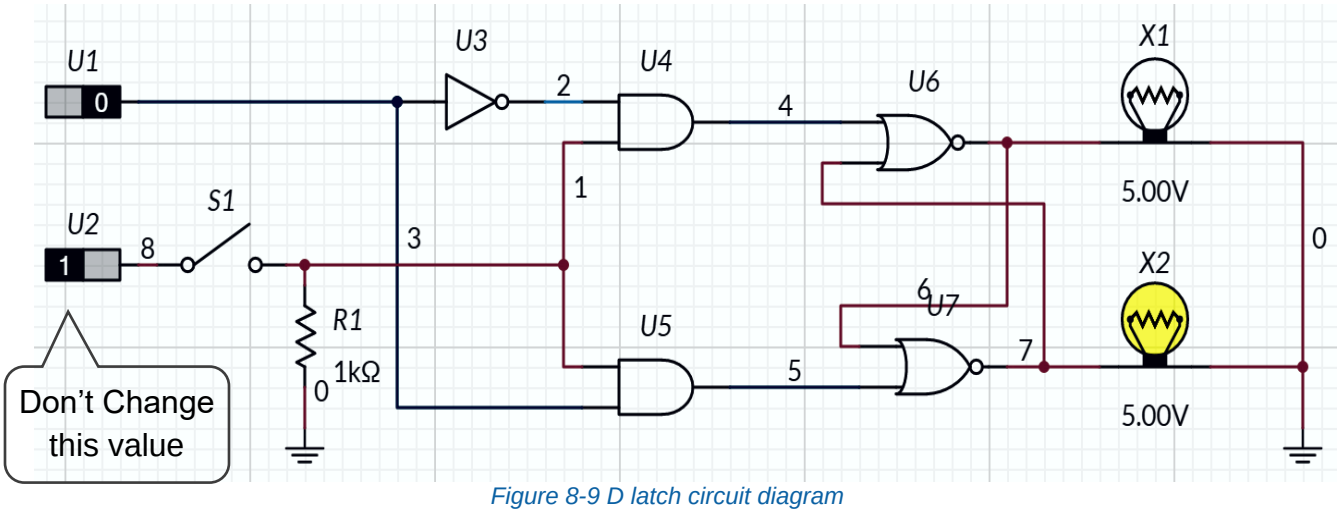


Figure 8-9 D latch circuit diagram

- Multisim:	Όνομα αρχείου “**8_D_Latch.ms14**”.
	Προσθήκη στο zip file με όνομα ”Lab8_ονοματεπώνυμο_AM.zip”
ή	
- MultisimLive:	Όνομα αρχείου “**8_D_Latch.png**”.
Schematic image	Προσθήκη στο zip file με όνομα ”Lab8_ονοματεπώνυμο_AM.zip”

Testing a Gated D Latch Circuit using Logic Gates

Instructions:

- **Start** the Simulation.
- Press and close *enable (E=1)*
- Press and open *enable (E=0)*
- Change *D* to **1.**
- Change *D* to **0.**
- Change *D* to **1.**

8-3 Do the probes change and why?

a) No, because *E*=0 so the circuit cannot activate a change.
b) Yes, because *E*=0 so the circuit can activate a change.

- Press the button *enable* to close *(E=1)*.

8-4 Does the probe change and why?

a) No, the probe does not change because enable is activated and input is “1”.
b) Yes, now the probe has changed because enable is activated and input is “1”.

- Change *D* to **0.**
- Change *D* to **1.**
- Change *D* to **0.**

8-5 Do the probes change and why?

a) No, the probe does not change because enable is activated and input is “0”.
b) Yes, the probe has changed because enable is activated and the input changes.

- Press the button *enable* to open *(E=0)*.

- Change *D* to **1.**

8-6 Does the probe change and why?

a) No, because *E*=0 so the circuit cannot activate a change.
b) Yes, because *E*=0 but input is “1”.

- **Stop** the simulation.

8-7 Compare your observations to figure 8-7’s characteristic table.

a) Is the same.
b) Differ.

8.3 Exercise: Verifying the Gated SR Latch Characteristic Table

Circuit

Instructions:

- Launch Multisim
- Connect the following circuit:

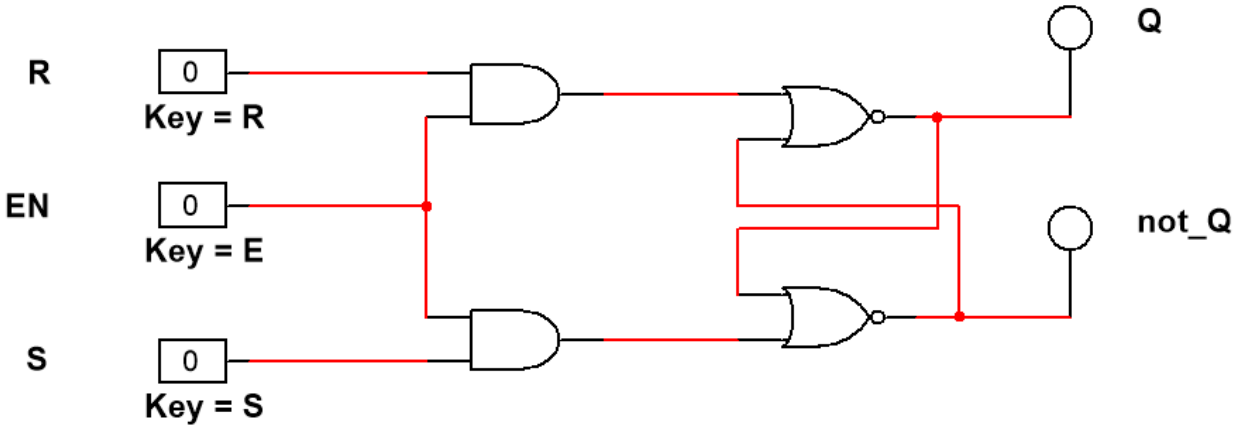


Figure 8-10 SR latch circuit diagram

- If you use MultisimLive you can build the following:

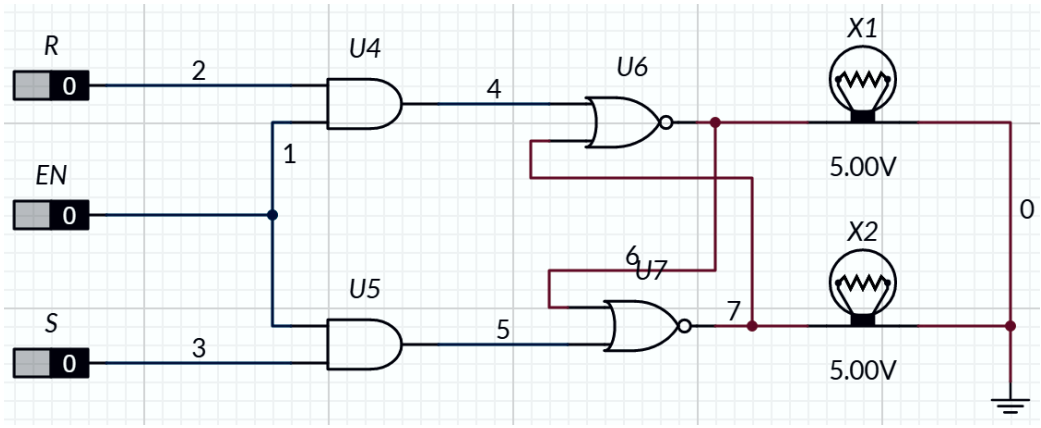


Figure 8-11 SR latch circuit diagram

- Multisim:	Όνομα αρχείου “**8_Gated_SR_Latch.ms14**”.
	Προσθήκη στο zip file με όνομα ”Lab8_ονοματεπώνυμο_AM.zip”
ή	
- MultisimLive:	Όνομα αρχείου “**8_Gated_SR_Latch.png**”.
Schematic image	Προσθήκη στο zip file με όνομα ”Lab8_ονοματεπώνυμο_AM.zip”

- **Start** the simulation.
- Change *R* to **1.**
- Change *EN* to **1.**
- Change *EN* to **0.**

- Using the Characteristic table provided, vary the inputs and observe the probe.

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

Figure 8-12 Characteristic table

8-8 With the *Enable* input equal to 0, does it matter what you do to the *S* and *R* inputs?

a) Yes
b) No

- Set *S* to **1.**
- Set *R* to **0.**
- Set *Enable* to **1.**

8-9 Does the probe *Q* light up?

a) Yes
b) No

- Set *S* to **0.**

8-10 Does the probe *Q* light up?

a) Yes
b) No

8-11 Explain what has happened.

a) Despite *S* changing from 1 to 0, *Q* is maintained due to the latch's memory function when *S*=*R*=0.
b) The probe went out because *S* changing from 1 to 0.

- Set *R* to **1.**

8-12 Does the probe *Q* light up?

a) Yes
b) No

8-13 Explain what has happened.

a) When *R* changed to 1 it reset the probe and thus put it back to off because *S* was false. This change was able to occur because *enable* is set to false.
b) When *R* changed to 1 it reset the probe and thus put it back to off because *S* was false. This change was able to occur because *enable* is set to true.

8-14 When *S*=*R*=1 and *Enable*=1, what happens to *Q* and *not_Q*? Is this a valid operating condition?

a) Both outputs are 0. Yes, this is a valid operating condition
b) Both outputs are 0. No, this is not a valid operating condition

- **Stop** the simulation.

8.4 Conclusion

8-15 Based on your observations, what are the main differences between the behavior of D latches and SR latches?

a) D latches are a simplification of SR latches. With a D latch ‘R’ is wired to be the same with ‘S’ thus eliminating the indeterminate state. With an SR latch, the S and R signals must be coordinated so they are never both 1.
b) D latches are a simplification of SR latches. With a D latch ‘R’ is wired to be the opposite of ‘S’ thus eliminating the indeterminate state. With an SR latch, the S and R signals must be coordinated so they are never both 1.

8-16 A sequential circuit’s behavior depends on:

A. The number and type of latches
B. Only on the inputs
C. Current inputs and previous behavior
D. The previous 4 bits

8-17 Sequential circuits employ which component for synchronization?

A. SR latches
B. Clock signals
C. Gated D latch
D. None of the above

8-18 A gated SR latch is a combination of which logic gates?

A. 2 AND and 2 NOR gates
B. 2 AND and XNOR gates
C. 2 AND and XOR gates
D. 2 AND and NOT gates

8-19 In the characteristic table of a gated SR latch, Q (t+1) represents:

A. A don’t care condition
B. The current value of the output
C. Removing the Q (t+1) state
D. The next state

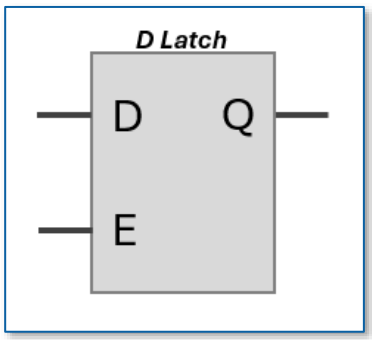
8-20 The D latch eliminates the undesirable condition of the indeterminate state in the SR latch by:

A. Reducing the number of logic gates needed for the circuit
B. Ensuring that the internal Set and Reset signals cannot be asserted simultaneously
C. Removing the Q(t + 1) state
D. Adding another output

8.5 Exercise: HDL - Verilog, D-Latch

1. Υλοποίηση D Latch ενός bit

Γράψτε σε γλώσσα Verilog την περιγραφή ενός μανδαλωτή τύπου D (D latch), σύμφωνα με τη λειτουργία του κυκλώματος του Figure 8-7.



Το κύκλωμα θα διαθέτει:

- Είσοδο δεδομένων D (1 bit).
- Είσοδο ενεργοποίησης enable (1 bit).
- Έξοδο Q (1 bit).

Για την υλοποίηση χρησιμοποιήστε τις εντολές **always** και **if**.

Το κύκλωμα πρέπει να λειτουργεί ως εξής:

- Όταν enable=1, η έξοδος Q ακολουθεί την είσοδο D (Transparent).
- Όταν enable=0, η έξοδος Q διατηρεί την τελευταία αποθηκευμένη τιμή της (Hold).

Ονομάστε το module: "D_latch"

- Verilog:	Όνομα αρχείου "**D_latch.v**". Προσθήκη στο zip file με όνομα "Lab8_ονοματεπώνυμο_AM.zip"

2. Testbench του D Latch

Κατασκευάστε ένα testbench για την επαλήθευση της λειτουργίας του προηγούμενου κυκλώματος, εφαρμόζοντας τις ακόλουθες μεταβολές:

Χρόνος	Μεταβολή
2 ns	enable = 0
10 ns	D = 0
20 ns	D = 1
30 ns	enable = 1
40 ns	D = 0
50 ns	D = 1
60 ns	enable = 0
70 ns	D = 0
80 ns	D = 1
90 ns	$finish

Χρησιμοποιήστε την εντολή $monitor για την παρακολούθηση των σημάτων.

Να χρησιμοποιήσετε τις εντολές **$dumpfile** και **$dumpvars** ώστε να παραχθεί το αρχείο “vcd” με τις κυματομορφές των σημάτων.

Ονομάστε το module: “t_D_latch”

- Verilog:	Όνομα αρχείου “**t_D_latch.v**”. Προσθήκη στο zip file με όνομα ”Lab8_ονοματεπώνυμο_AM.zip”

3. Έλεγχος αποτελεσμάτων

Εκτελέστε την προσομοίωση και παρατηρήστε τις κυματομορφές των D, enable και Q.

Απαντήστε στις παρακάτω ερωτήσεις:

8-21. Γιατί η έξοδος Q εμφανίζεται αρχικά με την τιμή x στην προσομοίωση;

α) Επειδή η είσοδος D είναι 0.

β) Επειδή η έξοδος Q δεν έχει αποκτήσει ακόμη γνωστή τιμή.

γ) Επειδή το enable είναι 1.

δ) Επειδή υπάρχει σφάλμα στο κύκλωμα.

8-22. Τι συμβαίνει στην έξοδο Q όταν μεταβάλλεται η είσοδος D ενώ enable=1;

α) Διατηρεί την προηγούμενη τιμή της.

β) Γίνεται πάντα 0.

γ) Ακολουθεί τις μεταβολές της D.

δ) Αλλάζει μόνο στη θετική ακμή του enable.

8-23. Γιατί η έξοδος Q δεν αλλάζει όταν μεταβάλλεται η D στα 70 ns και 80 ns;

α) Επειδή enable=0.

β) Επειδή η D είναι δυαδικό σήμα.

γ) Επειδή η προσομοίωση έχει ολοκληρωθεί.

δ) Επειδή η Q είναι πάντα 1.

Κάντε ένα **screenshot των κυματομορφών σας** και προσθέστε το στο zip file.
(μπορεί να είναι printscreen του υπολογιστή ή μία φωτογραφία με το κινητό σας)

- Picture:	Όνομα αρχείου “**t_D_latch**”. Προσθήκη στο zip file με όνομα ”Lab8_ονοματεπώνυμο_AM.zip”

4. Υλοποίηση 4-bit D Latch

Τροποποιήστε την προηγούμενη περιγραφή ώστε να υλοποιήσετε έναν μανδαλωτή τύπου D τεσσάρων bits (4-bit D latch).

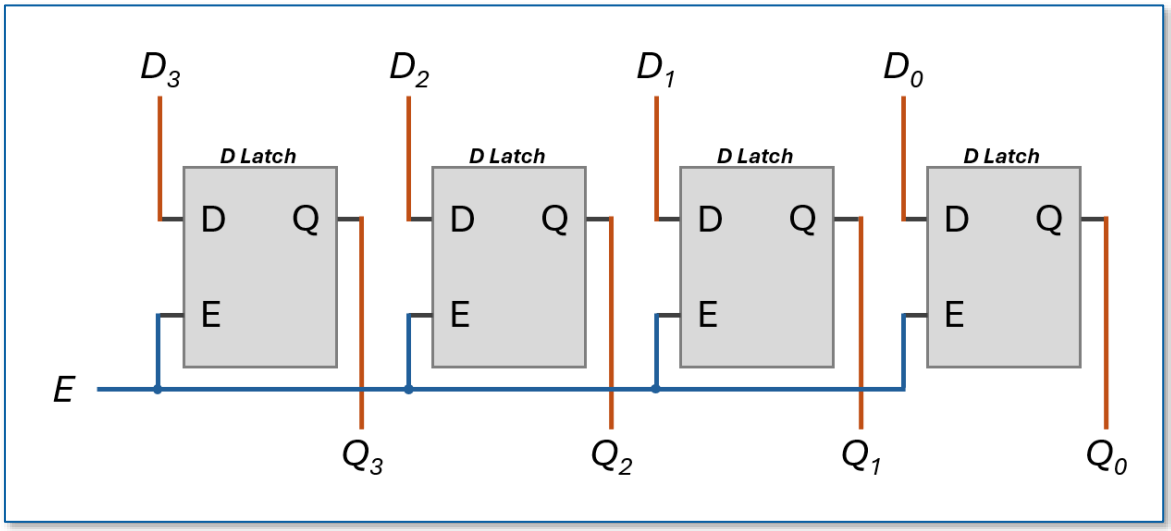
Το νέο κύκλωμα θα διαθέτει:

- Είσοδο δεδομένων D[3:0] (4 bits).
- Κοινή είσοδο ενεργοποίησης enable (1 bit).
- Έξοδο Q[3:0] (4 bits).

Η λειτουργία θα είναι ίδια με εκείνη του D latch ενός bit:

- Όταν enable=1, η έξοδος Q[3:0] ακολουθεί την είσοδο D[3:0].
- Όταν enable=0, η έξοδος διατηρεί την τελευταία αποθηκευμένη τιμή των τεσσάρων bits.

Για την υλοποίηση χρησιμοποιήστε τις εντολές **always** και **if**.



Ονομάστε το module: “D_latch_4bit”

- Verilog:	Όνομα αρχείου “**D_latch_4bit.v**”. Προσθήκη στο zip file με όνομα ”Lab8_ονοματεπώνυμο_AM.zip”

5. Testbench του 4-bit D Latch

Δημιουργήστε νέο testbench, στο οποίο θα ελέγχετε τη λειτουργία του κυκλώματος εφαρμόζοντας τις ακόλουθες μεταβολές:

Χρόνος	D[3:0]	enable	Ενέργεια
0 ns	0000	0	Αρχικοποίηση εισόδων
10 ns	1010	—	
20 ns	1100	1	
30 ns	0011	—	
35 ns	—	0	
40 ns	1111	—	
50 ns	0101	—	
60 ns	—	1	
70 ns	1001	—	
75 ns	—	0	
80 ns	0000	—	
90 ns	—	—	$finish

Η παύλα (—) σημαίνει ότι το αντίστοιχο σήμα διατηρεί την προηγούμενη τιμή του.



Χρησιμοποιήστε την εντολή $monitor για την παρακολούθηση των εισόδων και της εξόδου.



Να χρησιμοποιήσετε τις εντολές **$dumpfile** και **$dumpvars** ώστε να παραχθεί το αρχείο “vcd” με τις κυματομορφές των σημάτων.

Ονομάστε το module: “t_D_latch_4bit”



- Verilog: Όνομα αρχείου “**t_D_latch_4bit.v**”.
Προσθήκη στο zip file με όνομα ”Lab8_ονοματεπώνυμο_AM.zip”

6. Αξιολόγηση αποτελεσμάτων

Εκτελέστε την προσομοίωση και ελέγξτε τις κυματομορφές.

Απαντήστε στις παρακάτω ερωτήσεις:

8-24. Ποια είναι η τιμή της εξόδου Q[3:0] στα 50 ns;

α) 1111

β) 0101

γ) 0011

δ) xxxx

8-25. Ποια είναι η τιμή της εξόδου Q[3:0] στα 70 ns;

α) 0101

β) 1001

γ) 0011

δ) 0000

8-26. Τι συμβαίνει στην έξοδο Q[3:0] όταν enable=0 και μεταβάλλεται η είσοδος D[3:0];

α) Όλα τα bits της εξόδου γίνονται 0.

β) Η έξοδος ακολουθεί αμέσως την D.

γ) Η έξοδος αποκτά άγνωστη τιμή.

δ) Διατηρούνται οι προηγούμενες τιμές και των τεσσάρων bits.

Κάντε ένα **screenshot των κυματομορφών σας** και προσθέστε το στο zip file.
(μπορεί να είναι printscreen του υπολογιστή ή μία φωτογραφία με το κινητό σας)



- Picture: Όνομα αρχείου “**t_D_latch_4bit**”.
Προσθήκη στο zip file με όνομα ”Lab8_ονοματεπώνυμο_AM.zip”