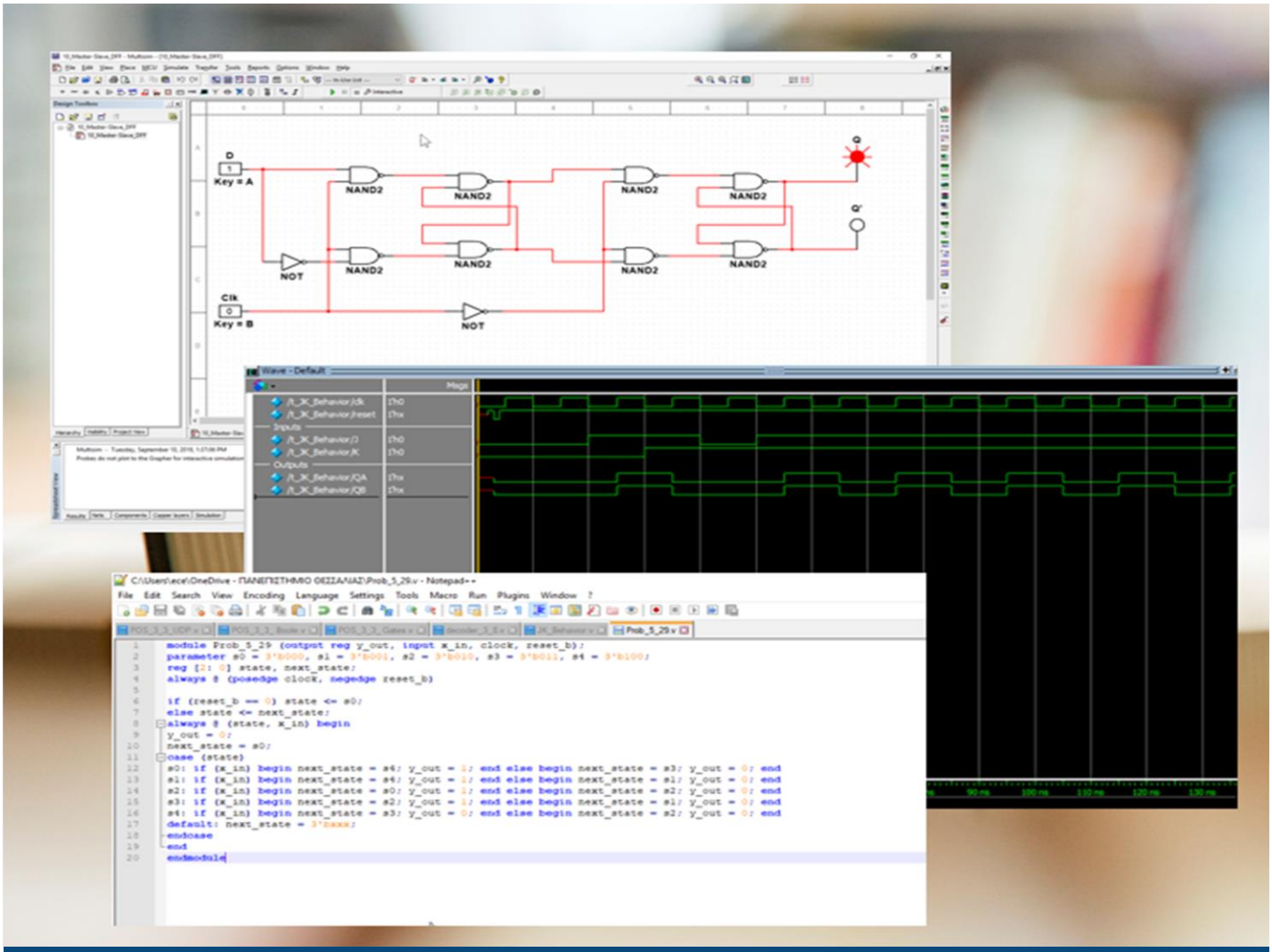
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
χρόνια δημιουργίας

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ




# ECE119 Ψηφιακή Σχεδίαση

Εργαστηριακές ασκήσεις, Multisim - Verilog

**Lab 9**: Flip-Flops

Καραμπερόπουλος Δημήτρης

Σεπτέμβριος 2023

**Lab 9**

## 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 9: Flip-Flops

In the previous lab, you were introduced to the concept of sequential logic circuits by examining latches. Particularly, you explored D and SR latches. In this lab, you will explore sequential logic circuits by focusing on flip-flops. Flip flops differ from latches in that they are edge triggered. Latches are level sensitive devices. Flip-flops can be constructed from latches. Together, flip-flops and latches are the building blocks of sequential logic circuits.

## Learning Objectives

In this lab, students will:

1. Become familiar with the basic behavior of D, JK and T flip-flops
2. Explore variations of flip-flops in greater detail (i.e. master-slave relationships)
3. Gain an awareness of the versatility of the DFF and its application in other circuits.

## Expected Deliverables

In this lab, you will collect the following deliverables:

- Flip-flop analysis
- Conclusion questions

## 9.1 Theory and Background

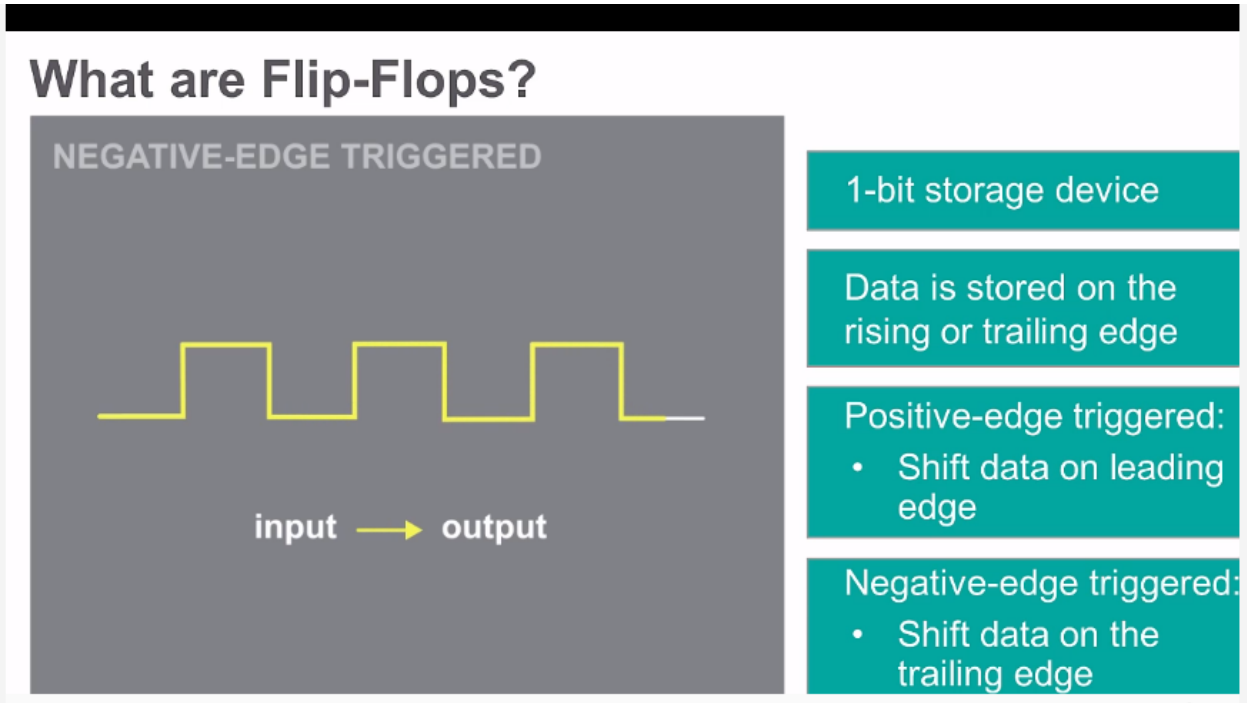



*Figure 9-1 Video Screenshot. View the video here: https://youtu.be/zLZUjC5Zr-w*

### *Video Summary*

- Flip-flops are 1-bit storage devices that store data on the falling or rising edge of a clock signal
- If a signal is transferred from input to output on the rising edge it is positive-edge triggered
- If a signal is transferred from input to output on the falling edge, it is negative-edge triggered

## Flip-flops

- Flip-flops are the fundamental building blocks of sequential circuits.
- They are 1-bit storage devices that, unlike latches (which are level sensitive devices) are edge-triggered.
- Data gets stored into a flip-flop at one of the edges of the clock signal, i.e. when the clock input makes a transition from 0 to 1 or from 1 to 0.

The timing diagram of a clock pulse is shown below:

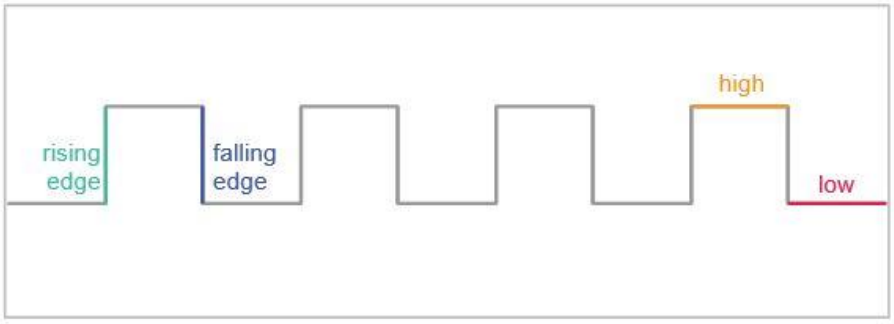



*Figure 9-2 Timing diagram of a clock pulse*

## Latch Feedback

- The state transitions of the latches start in the moment the clock changes to 1 and can continue for the entire period while it is active.
- Because of this, the output of a latch cannot be applied through combinational circuits to the inputs of other latches triggered by the same clock signal.
- This fact represents a serious drawback when using latches as storage elements.
- Flip-flops overcome this problem by triggering only during signal changes (edges).

Latch feedback is illustrated in the figure below.

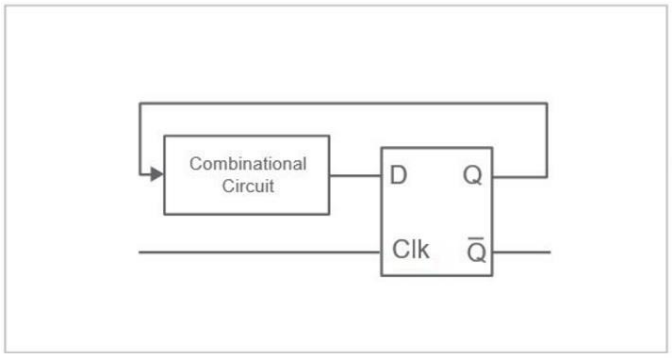



*Figure 9-3 Latch feedback*

## D Flip-flops (DFFs)

*D flip-flops (DFFs)* are very commonly used because they are perfectly suited for the construction of sequential circuits.

- If the DFF transfers the input to the output on the rising edge of the clock signal then it is said to be a *positive-edge triggered* D flip-flop or positive-edge DFF.
- If the DFF transfers the input to the output on the falling edge of the clock signal then it is said to be a *negative-edge triggered* D flip-flop or negative-edge DFF.
- This mode of operation is signaled through the use of a circle on the clock input. The image below shows the symbol for a positive-edge DFF (left) and a negative-edge DFF (right).
- For both DFF circuits shown below, the small triangle on the clock input indicates that the flip-flop will trigger only on one of the clock edges.
- The complemented output, Q is omitted in the graphical symbol when it is not needed.

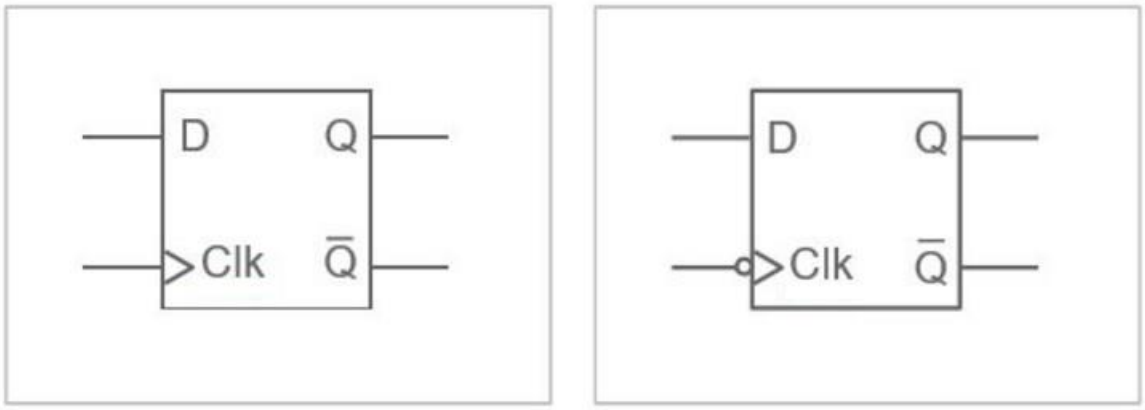



*Figure 9-4 Positive edge DFF (left) and negative edge DFF (right)*

The operation of the positive-triggered D flip-flop is very simple:

- The output **Q** will go to the present state of the **D** input when the clock signal changes from **0** to **1**.
- The level present at the input will be stored in the flip-flop on the rising transition of the **Clk** signal.

The figures below present the functionality of a positive-edge DFF:

| Clk | Q(t +1) |
|---|---|
| 0 | Q |
| 1 | Q |
| ↑ | D |
| ↓ | Q |

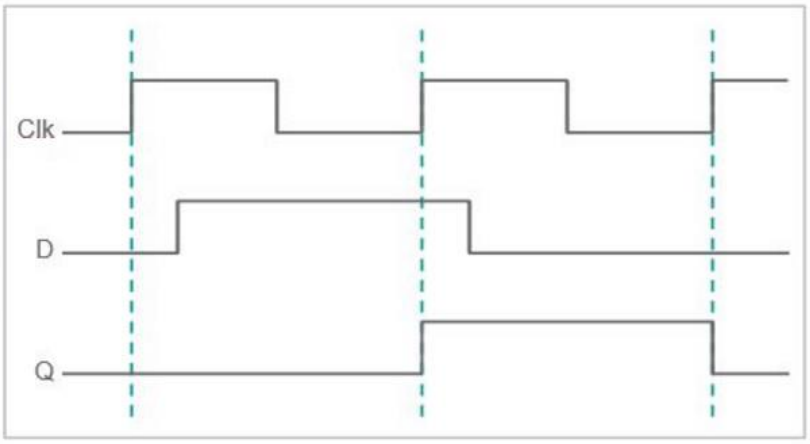



*Figure 9-5 Characteristic table (left) and diagram (right) for a positive-edge DFF*

A flip-flop can be built using the following methods:

- Using two D latches arranged in a master-slave configuration, where one of them is transparent during one semi-period of the clock period and the other transparent during the other semi-period.

**Note:** another circuit having the same operation, but implemented using fewer gates, can be built from three SR latches.

- Using of a single D latch and short pulses derived from the clock signal, which cause of the latch to be transparent for only a brief moment. These circuits are called pulsed latches or pulse-based flip-flops.

The timing characteristics of flip-flops include:

- the propagation delay from **Clk** to **Q** (the time needed for a value present on the input to reach the output when the clock changes from **0** to **1**)
- the setup time
- the hold time
- the maximum frequency that can be applied to the clock signal and still have correct operation

## The J K Flip-flop (JKFF)

Another type of flip-flop, one that is not as widely used as the DFF, but a more versatile one, is the *J K flip-flop (JKFF)*.

- The J K flip-flop is referred to as the *universal* bi-stable because it can easily behave like any one of the others.
- It has two data inputs, (**J** and **K**), a clock input, and two outputs (**Q** and $\bar{\mathbf{Q}}$)
- **J** acts as a **SET** input and **K** acts as a **RESET** input.
- The output changes only on one of the clock edges.
- When both **J** and **K** are **1**, the flip-flop toggles between opposite logic states each time the clock signal changes from **0** to **1** or from **1** to **0**.
- This circuit is like the one of a gated SR latch with an edge detector, with the difference that the *forbidden* state (when both **S** and **R** are set to **1**) is replaced by the *toggle* state.

The golden rules of JKFFs are:

1. If **J** and **K** are different, then **Q** is always the same as **J**.
2. If **J** and **K** are **0**, nothing happens.
3. If **J** and **K** are **1**, **Q** toggles.

The figures below present the way in which a J K flip-flop can be built using a DFF and logic gates. The graphical symbol of the JKFF is also shown below, and its operation is described by the characteristic table.

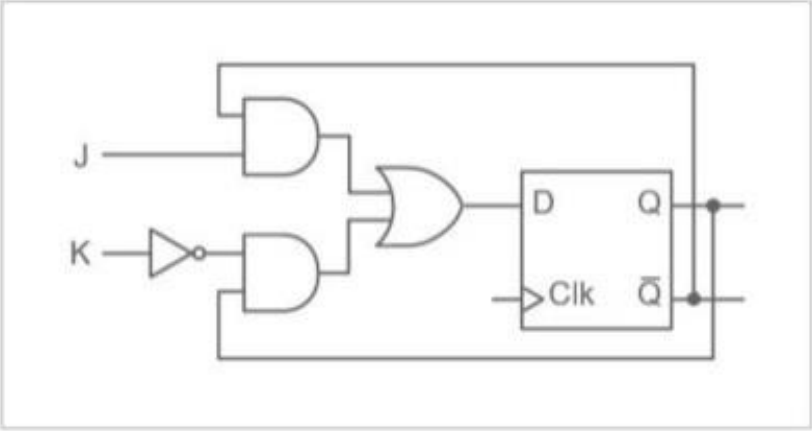



*Figure 9-6 J K flip-flop built using a DFF and logic gates*

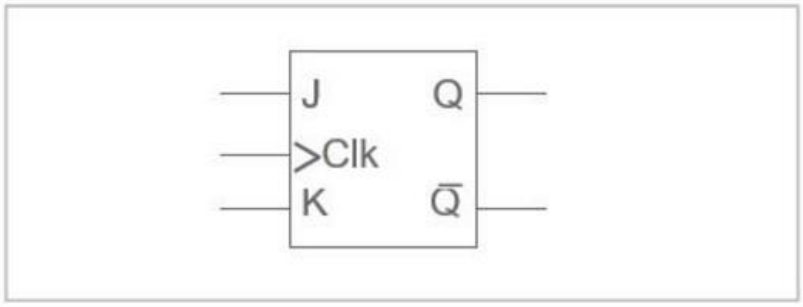



*Figure 9-7 Graphical symbol of the JKFF*

| J | K | Q(t +1) | |
|---|---|---|---|
| 0 | 0 | Q(t) | no change |
| 0 | 1 | 0 | reset |
| 1 | 0 | 1 | set |
| 1 | 1 | $\overline{Q(t)}$ | toggle |

*Figure 9-8 JKFF characteristic table*

## The T flip-flop

A *T flip-flop* is a flip-flop whose output toggles between **1** and **0** each applied clock pulse, when an input called **T** (from toggle) is active.

- The toggle mode of operation of a flip-flop is sometimes useful in synchronous circuits.
- It can be implemented using a DFF.

The figures below show the CLC for a TFF built using a DFF. The TFF graphical symbol is also shown below, along with the TFF characteristic table.

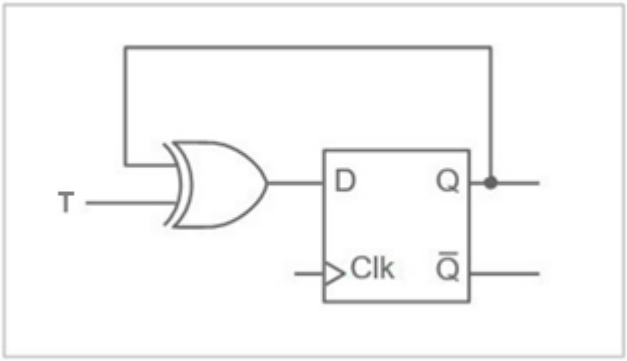



*Figure 9-9 CLC for a TFF built using a DFF*

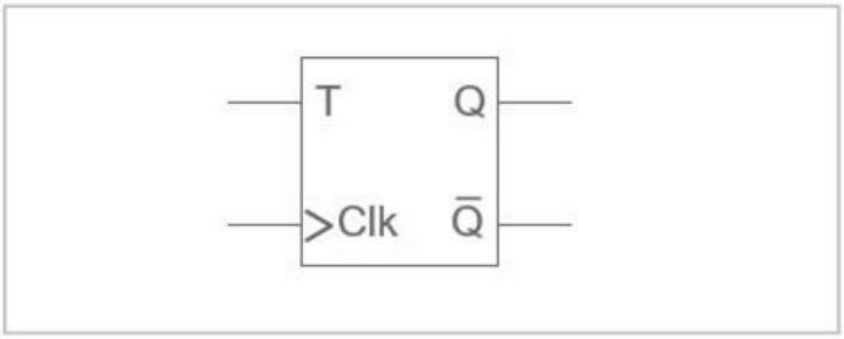



*Figure 9-10 TFF graphical symbol*

| T | Q(t +1) | |
|---|---|---|
| 0 | Q(t) | no change |
| 1 | $\overline{Q(t)}$ | toggle |

*Figure 9-11 TFF characteristic table*

A T flip-flop can also be built using a J K flip-flop, having the inputs **J** and **K** tied together, as shown below.

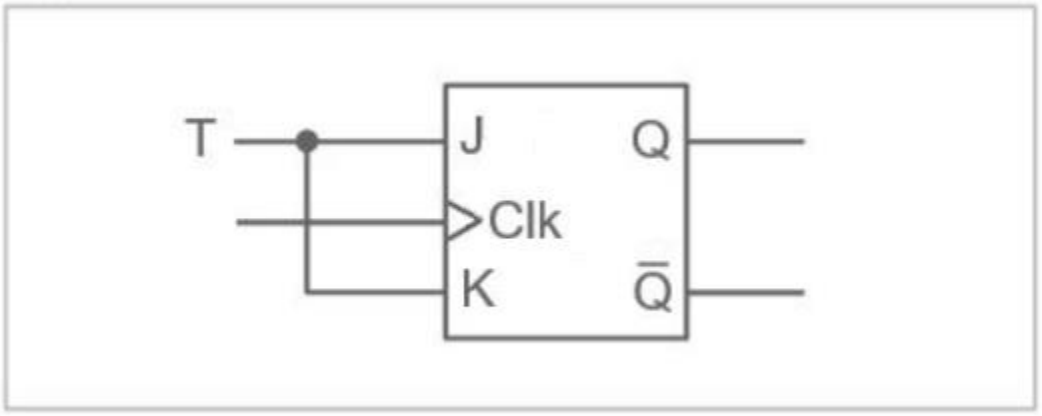



*Figure 9-12 TFF built using a JKFF*

## 9.2 Simulate: Building a Master-Slave D Flip-Flop

### Instructions:

- Launch Multisim
- Connect the following circuit:

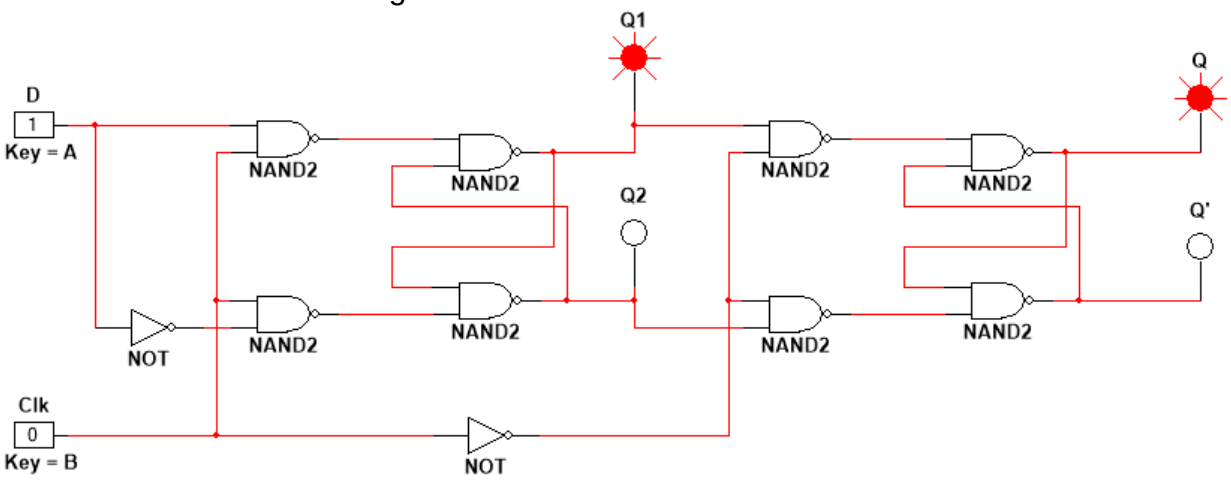



*Figure 9-13 Circuit diagram*

| | |
|---|---|
| **- Multisim:** | Όνομα αρχείου “**9_Master-Slave_DFF.ms14**”. Προσθήκη στο zip file με όνομα ”Lab9_ονοματεπώνυμο_AM.zip” |
| **ή** | |
| **- MultisimLive:** | Όνομα αρχείου “**9_Master-Slave_DFF.png**”. |
| **Schematic image** | Προσθήκη στο zip file με όνομα ”Lab9_ονοματεπώνυμο_AM.zip” |

- **Start** the simulation
- Modify the inputs in order to understand its function

9-1 Is this circuit positive edge triggered or negative edge triggered? Explain why?

a) Positive edge triggered. It transfers the input to the output on the rising edge of the clock signal
b) Positive edge triggered. It transfers the input to the output on the falling edge of the clock signal
c) Negative edge triggered. It transfers the input to the output on the rising edge of the clock signal
d) Negative edge triggered. It transfers the input to the output on the falling edge of the clock signal

## 9.3 Simulate: Building a JK Flip Flop Using a DFF and Logic Gates

- Launch Multisim
- Connect the following circuit:

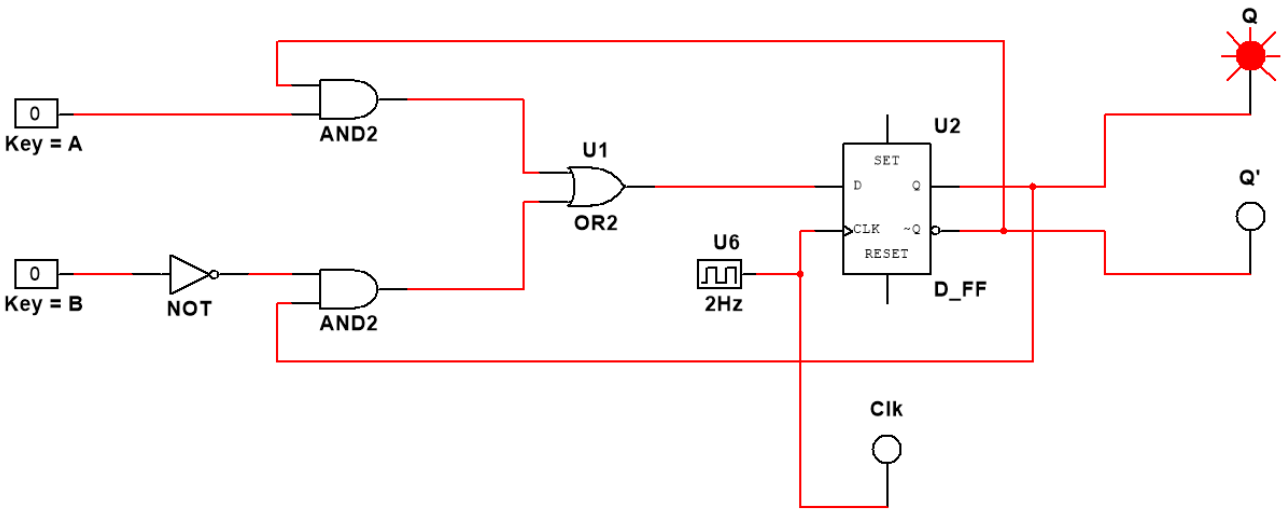



*Figure 9-14 Circuit diagram*

- If you use MultisimLive you can build the following. It does not have a DFF, so you have to build it with gates:

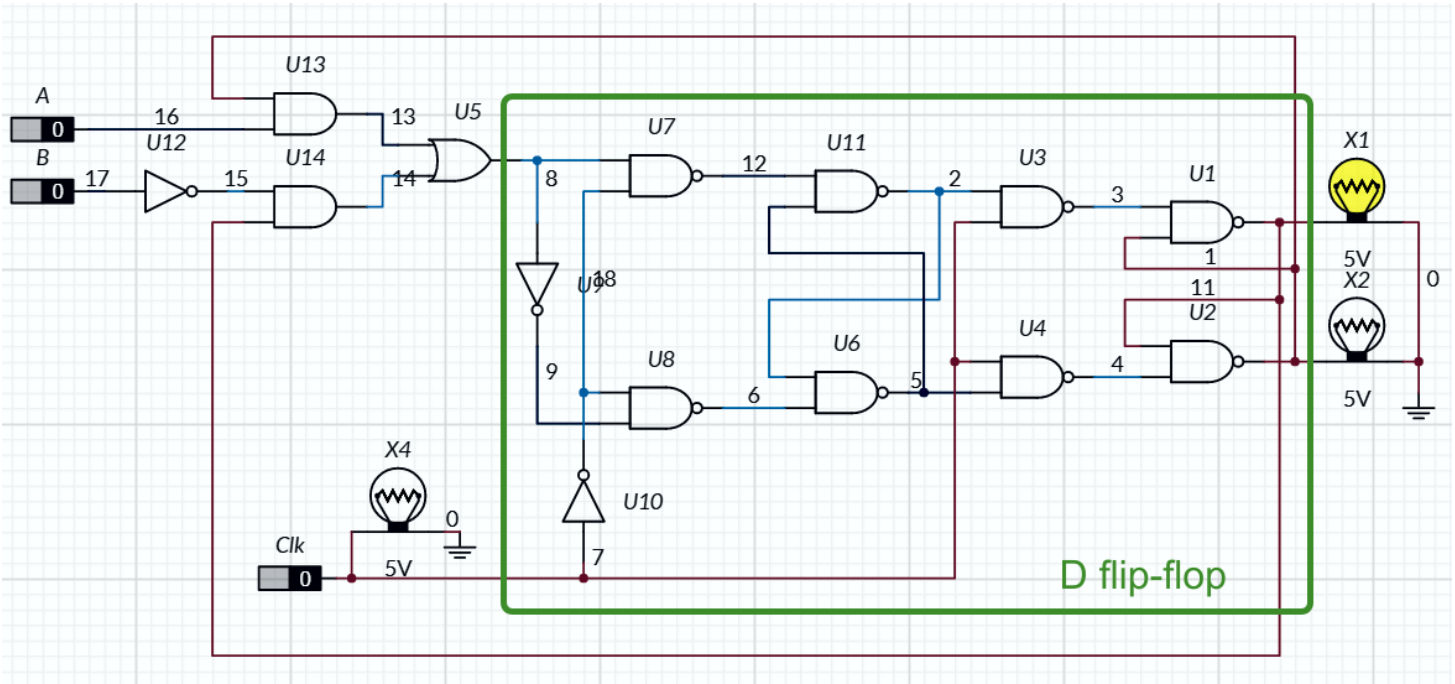



*Figure 9-15 Circuit diagram*

| | |
|---|---|
| **- Multisim:** | Όνομα αρχείου “**9_JK-FF.ms14**”.<br>Προσθήκη στο zip file με όνομα ”Lab9_ονοματεπώνυμο_AM.zip” |
| **ή** | |
| **- MultisimLive:**<br>**Schematic image** | Όνομα αρχείου “**9_JK-FF.png**”.<br>Προσθήκη στο zip file με όνομα ”Lab9_ονοματεπώνυμο_AM.zip” |

- **Start** the simulation
- Vary the two inputs and observe what happens

9-2 Compare your results with the following characteristic table:

| J | K | Q(t +1) |
|---|---|---|
| 0 | 0 | Q(t) |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | $\overline{Q(t)}$ |

*Figure 9-16 JFKK characteristic table*

a) They are the same
b) they are different

9-3 Based on the behavior, which interactive input is the J input and which is the K input?

a) The “J” input is the top interactive input, input key A.
The “K” input is the bottom interactive input, input key B.

b) The “J” input is the bottom interactive input, input key B.
The “K” input is the top interactive input, input key A.

- **Stop** the simulation when you are done

## 9.4 Conclusion

9-4 Flip-flops are storage devices that:

A. Are level sensitive
B. Are edge-triggered
C. Work only when the clock signal is set to 1
D. Work only when the clock signal is set to 0

9-5 Some D flip-flops have a circle on the clock input. This indicates:

A. The current travels into the flip-flop in one direction
B. A negative edge flip-flop
C. The flip-flop will be triggered only on one of the clock edges
D. The Q output is equivalent to the D input

9-6 JK flip-flops are said to be versatile because:

A. They have two inputs, ‘set’ and ‘reset’
B. They can behave like any other flip-flop
C. The output changes only one of the clock edges
D. All of the above

9-7 If both inputs on a JK flip-flop are 1, then:

A. The Q output is always the same as the set input
B. The Q output is always the same as the reset input
C. Q toggles
D. Nothing happens

9-8 T flip-flops can be built with which of the following:

A. JK flip-flop and an OR gate
B. D flip-flop and an XOR gate
C. JK flip-flop and an XOR gate
D. D flip-flop and an OR gate

## 9.5 Exercise: HDL - Verilog, D flip-flop

Γράψτε μια HDL περιγραφή συμπεριφοράς για το D flip-flop με ασύγχρονο μηδενισμό, βάση του πίνακα αληθείας του. Ο ασύγχρονος μηδενισμός να συμβαίνει στο “1” του σήματος “reset”.

**Ονομάστε το module: “D_FF”**


| | |
|---|---|
| **- Verilog:** | Όνομα αρχείου “**D_FF.v**”.<br>Προσθήκη στο zip file με όνομα ”Lab9_ονοματεπώνυμο_AM.zip” |

## 9.6 Exercise: HDL - Verilog, T flip-flop

Γράψτε μια HDL περιγραφή συμπεριφοράς για το T flip-flop κατασκευασμένο από ένα D flip-flop (της προηγούμενης άσκησης) και μία πύλη XOR, όπως στο figure 9.9.

**Ονομάστε το module: “T_FF”**


| | |
|---|---|
| **- Verilog:** | Όνομα αρχείου “**T_FF.v**”.<br>Προσθήκη στο zip file με όνομα ”Lab9_ονοματεπώνυμο_AM.zip” |

## 9.7 Exercise: HDL - Verilog, JK flip-flop

Υλοποιήστε σε γλώσσα Verilog ένα JK flip-flop κατασκευασμένο από ένα D flip-flop (το οποίο ήδη κατασκευάσατε) και πύλες, όπως στο figure 9.6.

**Ονομάστε το module: “JK_FF”**


| | |
|---|---|
| **- Verilog:** | Όνομα αρχείου “**JK_FF.v**”.<br>Προσθήκη στο zip file με όνομα ”Lab9_ονοματεπώνυμο_AM.zip” |

## 9.8 Exercise: HDL - Verilog, JK Behavior

Γράψτε και επαληθεύστε μια HDL περιγραφή συμπεριφοράς για το JK flip-flop, με ασύγχρονο μηδενισμό στο “0” του σήματος reset.

| | |
|---|---|
| **- Verilog:** | Όνομα αρχείου “**JK_Behavior.v**”.<br>Προσθήκη στο zip file με όνομα ”Lab9_ονοματεπώνυμο_AM.zip” |

Το παραπάνω αρχείο θα περιέχει **3 modules**.

Θέλουμε 2 modules τα οποία θα υλοποιούν το JK-FF με 2 τρόπους:

1. Γράψτε πρώτα την χαρακτηριστική εξίσωση του JK flip-flop και έπειτα δείτε τι αποτέλεσμα δίνει όταν Q=0 ή Q=1.
   Χρησιμοποιήστε στο module μια εντολή **if-else** που να βασίζεται στην τιμή της παρούσας κατάστασης Q, όπως π.χ: If (Q==0)…else…

   **Ονομάστε το module: “JK_Behavior_a”.**

2. Εξετάστε πως επηρεάζουν οι είσοδοι J και K την έξοδο του flip-flop σε κάθε παλμό του ρολογιού. Χρησιμοποιήστε και την εντολή **case** για τις διαφορετικές εισόδους.

   **Ονομάστε το module: “JK_Behavior_b”.**

Και ένα τρίτο module:

3. Για το test των παραπάνω modules γράψτε ένα test bench το οποίο θα μεταβάλει τα μεγέθη ως εξής:

   - Στα 0 nsec: clock=0
   - Κάθε 5 nsec: Αναστροφή του clock (Περίοδος 10 nsec)
   - Στα 2 nsec: reset=1
   - Στα 3 nsec: reset=0
   - Στα 4 nsec: reset=1
   - Στα 0 nsec: J=0, K=0
   - Στα 10 nsec: J=1, K=0
   - Στα 20 nsec: J=1, K=1

- Στα 40 nsec: J=0, K=1
- Στα 50 nsec: J=1, K=1
- Στα 70 nsec: $finish

- Δημιουργήστε 2 wire, τα **QA, QB**, ώστε να δοκιμάσετε ταυτόχρονα και τα δύο προηγούμενα modules που περιγράψατε. Δώστε στο **"JK_Behavior_a"** ως έξοδο το QA και στο **"JK_Behavior_b"** ως έξοδο το QB.

- Για το clock χρησιμοποιείστε μία δομή always.

- Να χρησιμοποιήσετε την εντολή "$monitor" για να ελέγξετε τις εισόδους - εξόδους. Απεικονίστε κατά σειρά τα σήματα: **time, reset, clock, J, K, QA, QB**.

- Η παραπάνω μεταβλητή time μας δείχνει την χρονική στιγμή που εκτελείται η εντολή monitor. Μπορεί να χρησιμοποιηθεί π.χ. ως εξής:
  **initial $monitor("time=%3d", $time);**
  Φυσικά μαζί με αυτή τη μεταβλητή μπορείτε να χρησιμοποιήσετε και τις υπόλοιπες όπως, κατά σειρά, ζητείται πιο πάνω.

Δείτε το παράδειγμα HDL 5.5 (Morris Mano)

**Ονομάστε το module: "t_JK_Behavior".**

Σε κάθε περίπτωση ελέγξτε τις κυματομορφές για την σωστή τιμή εξόδου.

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

**- Picture:** Όνομα αρχείου "**JK_Behavior**".
Προσθήκη στο zip file με όνομα "Lab9_ονοματεπώνυμο_ΑΜ.zip"