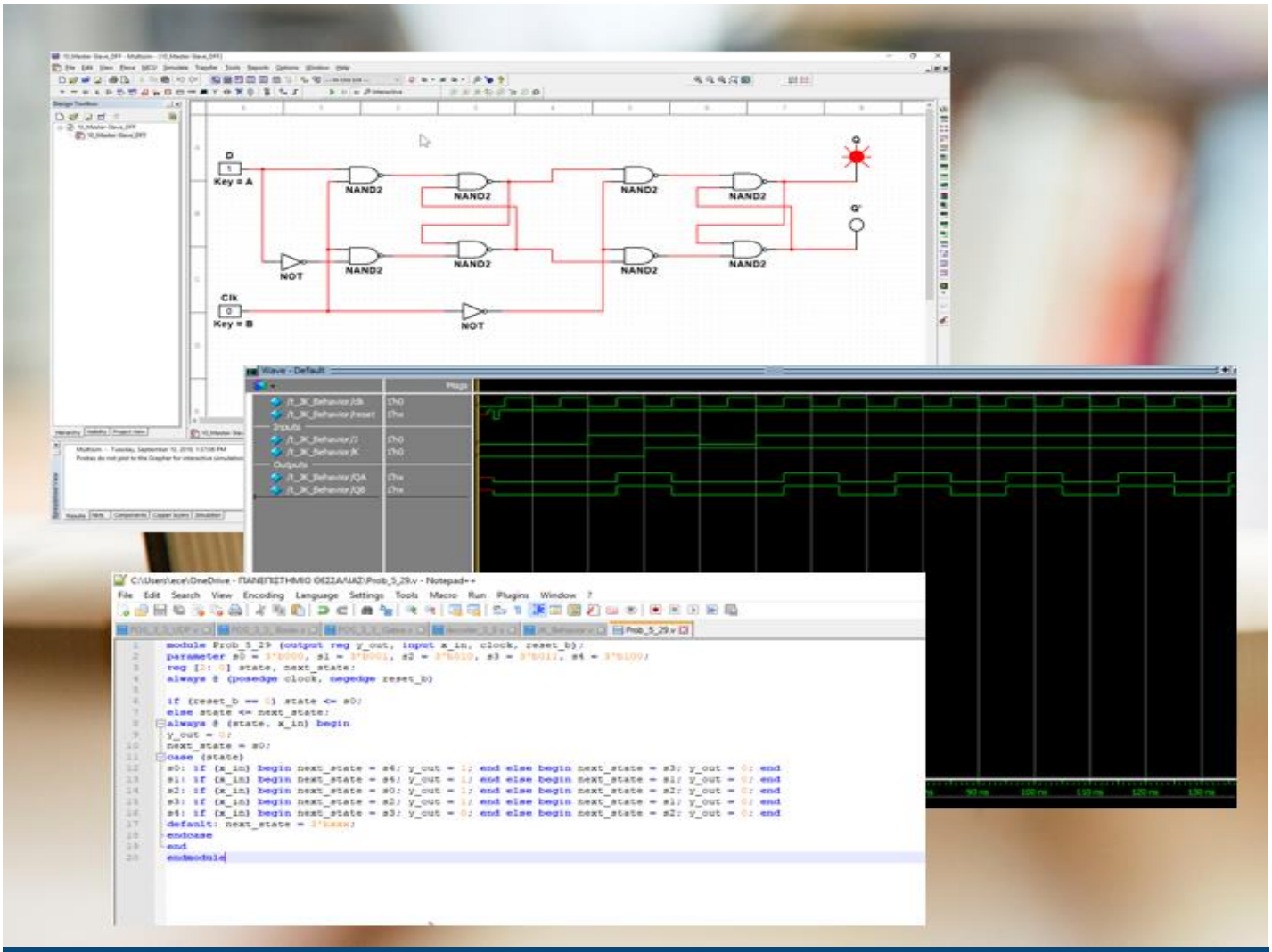
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ



ECE119 Ψηφιακή Σχεδίαση

Εργαστηριακές ασκήσεις, Multisim - Verilog

Lab 2: Truth Tables and Basic Logic Gates

Καραμπερόπουλος Δημήτρης

Σεπτέμβριος 2026

Lab 2

Required Tools and Technology 1
Lab 2: Truth Tables and Basic Logic Gates 2
Learning Objectives 2
Expected Deliverables 2
2.1 Theory and Background 4
2.2 Άσκηση: Θεώρημα 1 8
2.3 Άσκηση: Θεώρημα 3, Διπλή άρνηση 8
2.4 Άσκηση: Θεώρημα 6, Απορρόφησης 8
2.5 Simulate: Building a Circuit with Multiple Gates 9
2.6 Exercise: Determining a Circuit from a Truth Table 11
2.7 Conclusion 13
2.8 Exercise: Γλώσσα Περιγραφής Υλικού HDL - Verilog 14
Βασικές οδηγίες λειτουργίας του Icarus Verilog: 15

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/

Lab 2: Truth Tables and Basic Logic Gates

Nowadays, digital hardware is encountered in almost every aspect of our everyday life, being part of personal computers, household appliances, robots, television networks, etc.

Logic circuits are the building blocks of digital hardware. The logic circuits perform operations on digital signals and are usually implemented as electronic circuits where the signal values are restricted to a few discrete values. The most common are the binary logic circuits, where the only values are 0 and 1.

Although multi-valued logic systems with three or more logic levels exist, binary logic dominates practical digital circuit design because it provides simpler implementation and greater noise immunity.

The three basic logic operations are:

- Logical **AND**
- Logical **OR**
- **NOT** operation (inversion)

The logic operations are implemented with logic gates. A logic gate is an electronic circuit made up of transistors. The information related to the logic gates and logic functions and be described by a *truth table*.

Learning Objectives

In this lab, students will:

1. Explore the behavior of different configurations of logic gates.
2. Configure and build circuits in Multisim.
3. Learn basics in Verilog (HDL)

Expected Deliverables

In this lab, you will collect the following deliverables:

- Probe results
- Truth Tables
- Analysis of gate behavior
- Analysis of circuit behavior
- Circuit calculations
- Conclusion questions
- Multisim Files
- Verilog File

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

2.1 Theory and Background

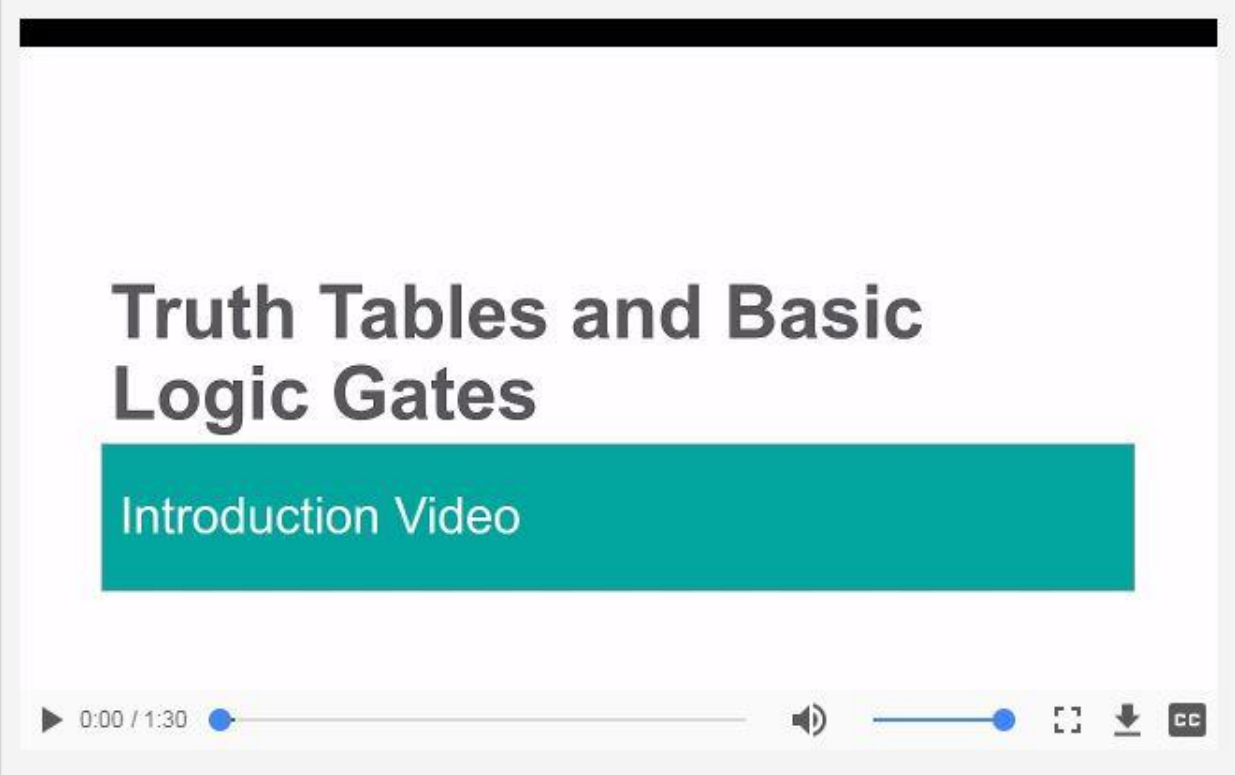


Figure 2-1 Video. View the video here: https://youtu.be/PhlGDrqgmj8

Video Summary

- Logic gates are the building blocks of all digital electronics
- AND and OR gates have at least two inputs and only one output
- NOT gates have one input and one output
- Inputs and outputs are expressed solely in binary (0's or 1's)

Truth Tables

A common way to express the specific function of a logic circuit is the truth table. *(Other methods of presentation include Schematics, Boolean expressions, and Behavioral descriptions.)*

Truth tables show all permutations of the inputs with their corresponding output values in terms of logic level states. Logic level states are typically expressed as:

- 1 and 0
- HIGH and LOW
- True and False

This is an example of a truth table for two inputs:

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

Figure 2-2 Truth table for two inputs

A gate or logic circuit's truth table must have as many rows as there are possibilities of unique input combinations. For a single-input gate, like the inverter, there are only two input possibilities, namely 0 and 1. For a two-input gate there are four possibilities (00, 01, 10, and 11), and thus four rows for the corresponding truth table. For a three-input logic device, there are eight possibilities and so forth. The input columns are typically written in binary order as shown here:

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

Figure 2-3 Truth table for three inputs written in binary

Logic Gates

Logic gates are physical devices that implement the Boolean functions of truth tables. The three most basic logic gates are the “AND”, “OR” and “NOT”.

- In the “**AND**” logic gate, the output is 1 if both the inputs for A and B are also 1. If one or all of the inputs for A and B are 0, then the resulting output is 0. This is summarized in the truth table below.
- Generally, the “AND” logic gate outputs the minimum value between the two input digits.
- The “AND” symbol is represented on the right. In this case, we can see two inputs (A and B) and one output.

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

Fig 2-4 AND Truth Table

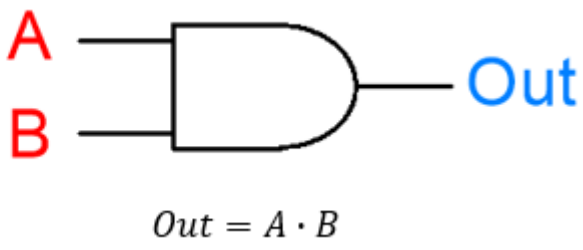


Fig 2-5 AND Logic Gate

- In the “**OR**” logic gate, the output is 0 if both the inputs for A and B are also 0. If one or all of the inputs for A and B are 1, then the resulting output is also 1. This is summarized in the truth table below.
- The “OR” logic gate outputs the maximum value between the two input digits.
- The “OR” symbol is represented below. As above, there are two inputs and one output.

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

Figure 2-6 OR Truth Table

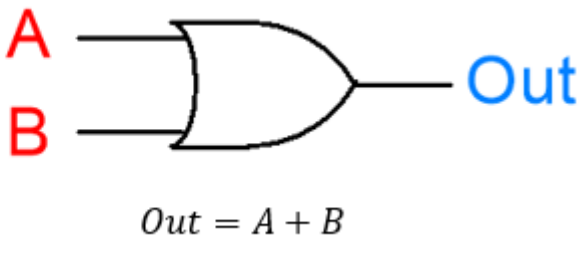


Figure 2-7 OR Logic Gate

- The “**NOT**” logic gate is also known as Inverter.
- It has only one input and one output.
- The truth table for an inverter is simple. The output is always the opposite of the input.
- For example, if the input is 1, the output will be 0 and vice versa. Visually this is depicted by a circle at the input and/or output ends of the logic gates.
- In this situation, the circle is at the output, which means that the output is inverted. If it was at the input, then it is the input that would be inverted.

In	Out
0	1
1	0

Figure 2-8 NOT Truth Table

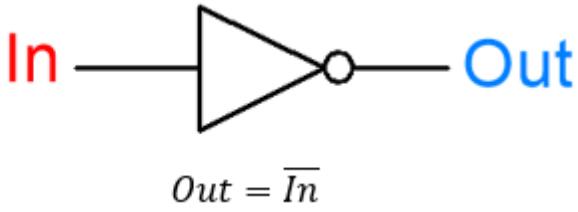


Figure 2-9 NOT Logic Gate

2.2 Άσκηση: Θεώρημα 1

Με μια *πύλη OR 2 εισόδων* επιβεβαιώστε το θεώρημα της άλγεβρας Boole σύμφωνα με το οποίο ισχύει η ακόλουθη σχέση: **x + x = x**. Συμπληρώστε τον κατάλληλο πίνακα αληθείας, αναγράφοντας όλους τους πιθανούς συνδυασμούς τιμών.

Η προσομοίωση να γίνει με το Multisim.



- Multisim: Όνομα αρχείου "**2_Theorem_1.ms14**".
Προσθήκη στο zip file με όνομα "Lab02_ονοματεπώνυμο_ΑΜ.zip"

2.3 Άσκηση: Θεώρημα 3, Διπλή άρνηση

Τοποθετείστε και συνδέστε σε σειρά *δύο αντιστροφείς* και επιβεβαιώστε το θεώρημα της Άλγεβρας Boole: **(x')' = x**. Συμπληρώστε τον κατάλληλο πίνακα αληθείας, αναγράφοντας όλους τους πιθανούς συνδυασμούς τιμών.

Η προσομοίωση να γίνει με το Multisim.



- Multisim: Όνομα αρχείου "**2_Theorem_3.ms14**".
Προσθήκη στο zip file με όνομα "Lab02_ονοματεπώνυμο_ΑΜ.zip"

2.4 Άσκηση: Θεώρημα 6, Απορρόφησης

Χρησιμοποιώντας τα κατάλληλα κυκλωματικά στοιχεία επιβεβαιώστε το θεώρημα απορρόφησης κατά το οποίο ισχύει η ακόλουθη σχέση: **x + xy = x**. Συμπληρώστε τον κατάλληλο πίνακα αληθείας, αναγράφοντας όλους τους πιθανούς συνδυασμούς τιμών.
Η προσομοίωση να γίνει με το Multisim.



- Multisim: Όνομα αρχείου "**2_Theorem_6.ms14**".
Προσθήκη στο zip file με όνομα "Lab02_ονοματεπώνυμο_ΑΜ.zip"

2.5 Simulate: Building a Circuit with Multiple Gates

Circuit Example 1

Build the following circuit using multiple AND/OR Gates in Multisim:

- Place an **OR** gate and **two AND** gates from the **Misc Digital** group.
- Place **three INTERACTIVE_DIGITAL_CONSTANT**s from the **Sources** group.
- Place one **PROBE_DIG_RED** from the **Indicators** group.
- Wire them as shown.

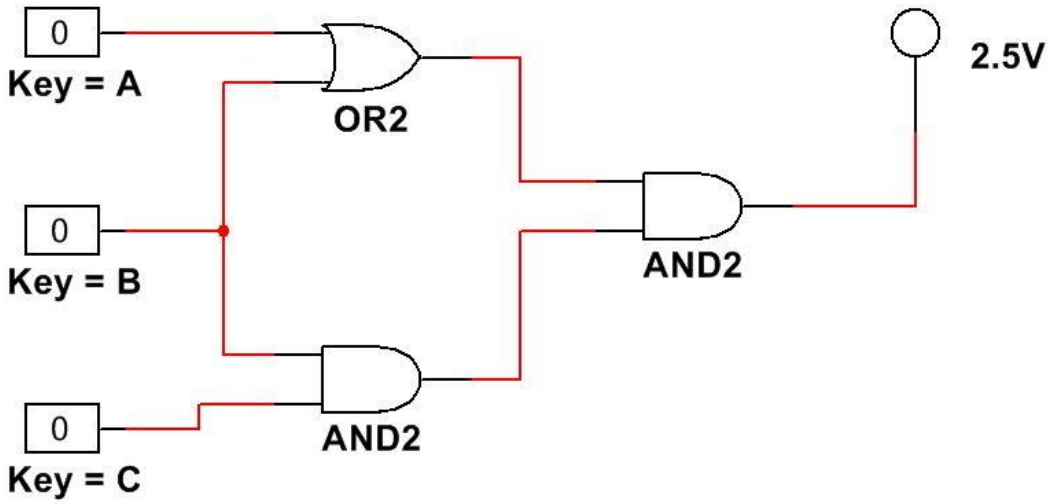


Figure 2-10 Circuit with AND, OR Logic gates

- Click the **Run** button to begin simulating the circuit.



Figure 2-11 Run button

- Using the **A**, **B**, and **C** keys, vary the inputs into the circuit.

2-1 Record the results, as indicated by the probe, in the following truth table.

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

- When you're done, stop the simulation by clicking the **Stop** button.



Figure 2-12 Stop button

2.6 Exercise: Determining a Circuit from a Truth Table

Truth Tables

We have now observed how truth tables define the behavior of a given circuit. However, we often know what type of behavior we need to implement, and we need to figure out what configuration of gates would give us this behavior. This can be a lot more complex.

In this part, you'll look at a few simple truth tables and determine their circuits.

Given the following truth table, determine which configuration of gates would give you these results. You can:

- Use trial and error simulation in Multisim.
- Calculate different circuits on paper.
- Use whatever other methods you find useful.
- **Use gates with 1 or 2 inputs.**

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

Figure 2-13 Truth table

- When you've picked a circuit design that you think will result in this behavior, simulate it in Multisim.



- Multisim: Όνομα αρχείου "**2_Figure_2_13.ms14**".
Προσθήκη στο zip file με όνομα "Lab02_ονοματεπώνυμο_AM.zip"

2-2 Did your circuit design work as expected? (Yes or No)

2-3 Is there more than one configuration of gates that would give you this result? (Yes or No)

Given the following truth table, determine which configuration of gates would give you these results.

You can:

- Use trial and error simulating in Multisim.
- Calculate different circuits on paper.
- Use whatever other methods you find useful.
- **Use gates with 1 or 2 inputs.**

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

Figure 2-14 Truth table

- When you've picked a circuit design that you think will result in this behavior, simulate it in Multisim.

- Multisim: Όνομα αρχείου "**2_Figure_2_14.ms14**".
Προσθήκη στο zip file με όνομα "Lab02_ονοματεπώνυμο_ΑΜ.zip"

2-4 Did your circuit design work as expected? (Yes or No)

2-5 Is there more than one configuration of gates that would give you this result? (Yes or No)

2.7 Conclusion

2-6 Why is it useful to represent the behavior of a gate or series of gates in a table?

A. It is easier for us to design the circuit
B. The table helps to directly determine the output of the circuit based on specific inputs, without the need to simulate or detect logic through a series of portals
C. To better analyze the circuits and to be able to distinguish their differences.
D. To better see the differences between the gates.

2-7 What is the difference between “AND” and “OR” logic gates?

A. AND logic gates return true only when every input is true whereas OR logic gates return true if minimum one input is true.
B. OR logic gates return true only when every input is true whereas AND logic gates return true if minimum one input is true.

2-8 The basic logic operations are:

A. AND
B. OR
C. NOT
D. All the above

2-9 Truth tables:

A. Express the particular function of a logic circuit
B. Show all possible permutations of inputs and corresponding output values
C. Are quantified as logic level states
D. All of the above

2-10 A truth table with 3 inputs can have how many possible outputs?

A. 8
B. 4

C. 6
D. 12

2-11 AND logic gates have:

A. One input and two outputs
B. Two or more inputs and one output
C. A current inverter
D. None of the above

2-12 OR logic gates output:

A. The minimum value between the two inputs
B. The maximum value between the two inputs
C. The average of the two inputs
D. The output does not have any direct relationship to the inputs

2.8 Exercise: Γλώσσα Περιγραφής Υλικού HDL - Verilog

Γράψτε την περιγραφή HDL σε επίπεδο πυλών του κυκλώματος “Figure 2.10”.

Μεταγλωττίστε τον κώδικά σας με την χρήση του Icarus.

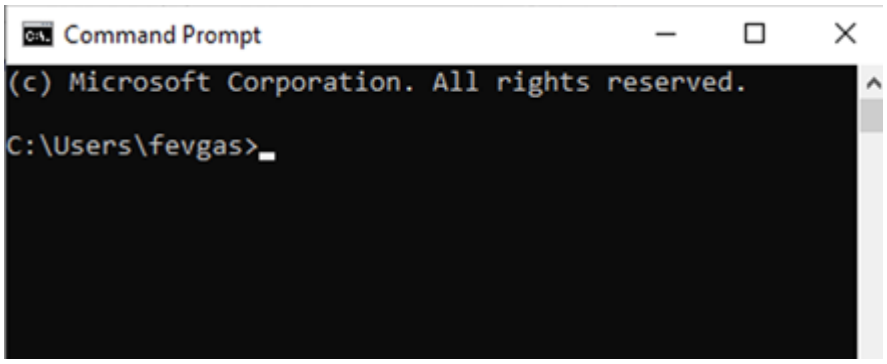
Ονομάστε το module: “Figure2_10”.
(βλέπε: Verilog Παράδειγμα 3.3 – Morris Mano)



- Verilog: Όνομα αρχείου “**Figure2_10.v**”.
Προσθήκη στο zip file με όνομα ”Lab02_ονοματεπώνυμο_AM.zip”

Βασικές οδηγίες λειτουργίας του Icarus Verilog:

- Ανοίξτε την εξερεύνηση των Windows (explorer)
- Μετακινηθείτε στον φάκελο …ECE119\Lab2
- Δεξί κλικ ***New→Text Document***
- Ελέγξτε ότι στο menu του explorer στο ***View*** είναι τσεκαρισμένο το: ***File name extensions***
- Αλλάξτε την κατάληξη του αρχείου από ".txt" σε "**.v**"
- Αλλάξτε το όνομα του αρχείου στο επιθυμητό (π.χ. **Figure2_10**)
- Από την έναρξη των Windows βρείτε και ανοίξτε το **Notepad++**
- ***File→Open*** βρείτε και ανοίξτε το αρχείο ***Figure2_10.v*** που δημιουργήσατε.
- Γράψτε τον κώδικα Verilog που περιγράφει το κύκλωμα της εικόνας Figure 2.10.
- Αποθηκεύστε το αρχείο. (στον προσωπικό σας χώρο …ECE119\lab2)
- Ανοίξτε ένα **Command Window**
  (δείτε την διαδικασία στις διαφάνειες του εργαστηρίου).



<u>Για το σπίτι:</u>

- Αλλάξτε τον φάκελο εργασίας (working directory) στο **C:\ECE119\lab02** δίνοντας στο **Command Window** τίς εντολές:

```
>C:
>cd C:\ECE119\lab02
```

Για το εργαστήριο:

- Αλλάξτε τον φάκελο εργασίας (working directory) στο **S:\ECE119\lab02** δίνοντας στο **Command Window** τίς εντολές:

```
>S:
>cd S:\ECE119\lab02
```

- Δείτε τα περιεχόμενα του καταλόγου με την εντολή:

```
>dir
```

*(Θα πρέπει να δείτε το αρχείο **Figure2_10.v** που δημιουργήσατε)*

- Μεταγλωττίστε (**compile**) τον κώδικα Verilog με την εντολή:

```
iverilog -o Figure2_10 Figure2_10.v
```

- Δείτε την **έξοδο** του **compiler**

- **Διορθώστε** τυχόν σφάλματα

```
C:\ECE119\Lab02>iverilog -o Figure2_10 Figure2_10.v
Figure2_10.v:8: syntax error
Figure2_10.v:8: error: invalid module item.

C:\ECE119\Lab02>iverilog -o Figure2_10 Figure2_10.v

C:\ECE119\Lab02>
```

- Δείτε ξανά τα περιεχόμενα του καταλόγου με την εντολή:

```
>dir
```

*(Θα πρέπει τώρα να δείτε 2 αρχεία. Το **Figure2_10.v** και το **Figure2_10** που δημιούργησε ο compiler)*