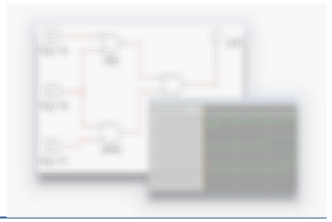



# ECE119 – Ψηφιακή Σχεδίαση

Διδάσκοντες Εργαστηρίου: Δ. Καραμπερόπουλος
Α. Φεύγας

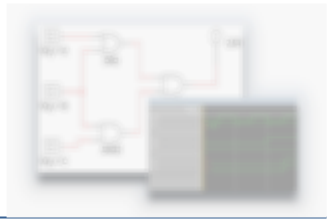
- Lab 3: Logic Gates Explored and Boolean Algebra

# 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)

# Logic Gates Explored and Boolean Algebra



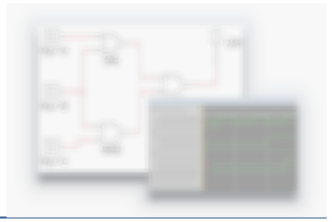
In the previous lab, we were introduced to the three basic logic gates **AND**, **OR** and **NOT** in detail.

Building on these, we can create a few other types of logic gates.

These are: **NAND, NOR, XOR,** and **XNOR**.

Let’s take a look at each one in greater detail.

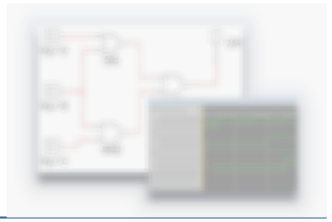
# Learning Objectives



In this lab, students will:

- Explore the function of various different **logic gates**
- Create circuits with varying logic gates in theory and in practice.
- Calculate and build combinational logic circuits from **Sum-of-Products** and **Product-of-Sums** derived from truth tables.
- Learn how to write a **Combinational Logic Circuit** (CLC) in Verilog.
- Learn how to **test a module** and take True Table in Verilog.

# Expected Deliverables



In this lab, you will collect the following deliverables:

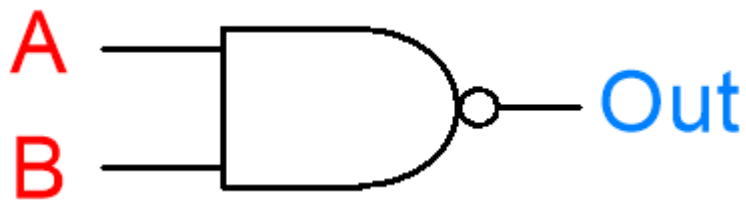
- SOP and POS Boolean expressions
- Design circuits
- Truth Tables
- Multisim Files
- Conclusion questions
- Verilog File

# NAND Logic Gates



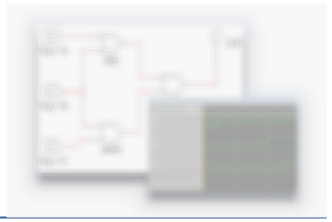
- **NAND** gates invert the output of the AND gate.
- The inputs do not change from those of the AND truth table, but the output is the opposite.
- As a rule, if any of the inputs are **0**, the output will always be **1**.
- See the truth table and the symbol.

| A | B | Out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



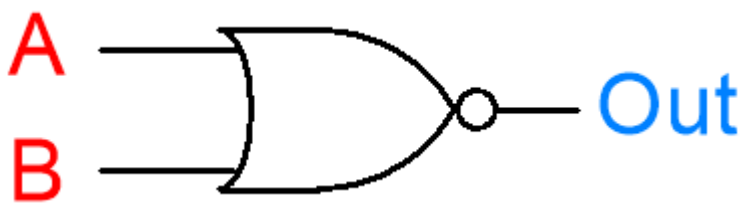

$$Out = \overline{A \cdot B}$$

# NOR Logic Gates



- The **NOR** logic gate inverts the output of the OR gate.
- The inputs of the truth table for the OR gate do not change, but the output is the opposite.
- As a rule, if any of the inputs are 1, the output will always be 0.
- See the truth table and symbol.

| A | B | Out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |



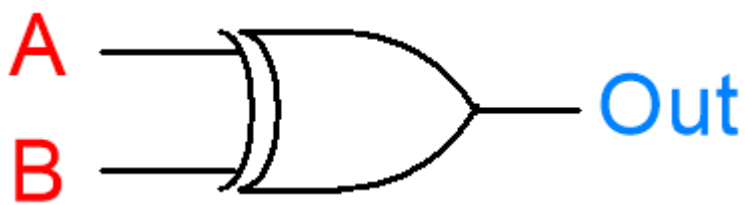

$$Out = \overline{A + B}$$

# XOR Logic Gates



- An **XOR** gate is also known as an **eXclusive OR** gate.
- The output will be 1 if only one of the inputs is 1. The output will be 0 if both inputs are 0 or both are 1.
- See the truth table and symbol.

| A | B | Out |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



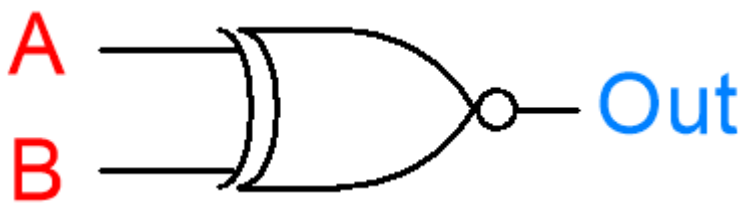

$$Out = \bar{A} \cdot B + A \cdot \bar{B}$$

# XNOR Logic Gates



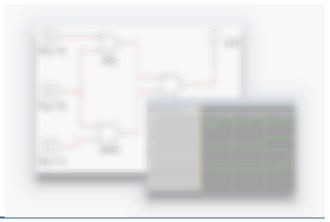
- The **XNOR** gate does the opposite of the XOR gate.
- The output will be 1 if the inputs are the same and the output will be 0 if the inputs are not the same.
- See the truth table and symbol.

| A | B | Out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

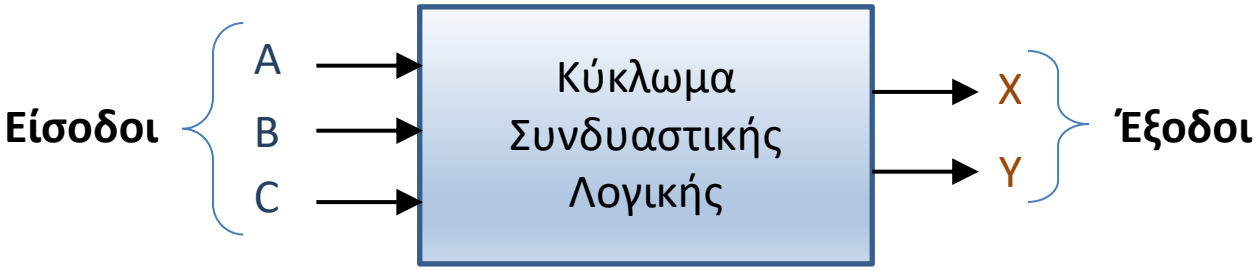



$$Out = A \cdot B + \bar{A} \cdot \bar{B}$$

# Combinational Logic Circuits (CLCs), Συνδυαστικά κυκλώματα

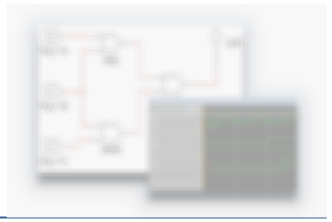


- **CLCs** are a classification of circuits whose output is only **dependent on the current inputs** and are implemented by Boolean circuits.
- Using combinations of logic gates, different results can be achieved.
- A truth table is often used to define the behavior of a CLC, but sometimes we start with a truth table and need to design a CLC.




Συνδυαστικό Κύκλωμα

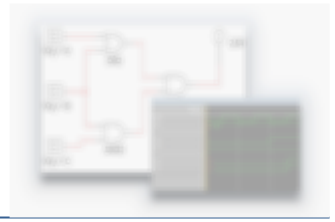
# Boolean Algebra (1/2)



**Boolean algebra** is an algebraic system where two values are used to represent the properties of bi-stable electrical switching circuits, namely **on** and **off**, or simply **1** and **0**. The rules for the two **binary operators** (addition and multiplication) and complement (') for a two-valued Boolean algebraic expression are presented in the tables on the next slide.

- It can be seen that the binary addition, multiplication and complement are the same as the OR, AND and NOT logic operations.
- For the **complement**, several notations are used: apostrophe after the variable, exclamation mark, tilde or the word NOT before the variable or an over-bar on top of it.
- Because it works with digital systems with only the values 0 and 1, the algebra used is simply called **"binary logic"**.

# Boolean Algebra (2/2)



- Any logic function, no matter how complex it is, can be implemented using only the three basic logic operations.
- A function represented by a truth table can be expressed using different methods.
- Knowing the logic expression and the function, the circuit can be then realized.

| x | y | x · y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Binary Multiplication (AND logic operation)**

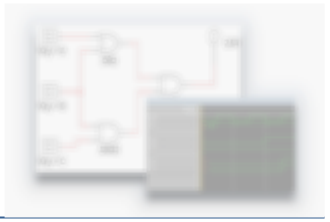
| x | y | x + y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**Binary Addition (OR logic operation)**

| x | x′ |
|---|---|
| 0 | 1 |
| 1 | 0 |

**Compliment (NOT logic operation)**

# Sum-of-Products (άθροισμα γινομένων)



A simple method for **converting a truth table into a CLC** is found in a standard form of Boolean expression called the **Sum-of-Products (SOP).**

- An SOP expression is literally a sum of Boolean terms called **minterms (ελαχιστόροι)**.
- A minterm is a multiplicative combination of Boolean variables whose output equals 1.
- An example of an SOP expression is ABC + AB'C', where ABC, AB'C' are minterms.
- SOP expressions may be generated from truth tables using the following steps:

1. Determine which rows of the table have an output of 1.
2. Derive each row's minterm, such that the output is 1 given that row's input state.
3. Sum the minterms.

# Sum-of-Products (άθροισμα γινομένων)

Below is an example of a **truth table conversion to an SOP** expression.

| A | B | C | O |
|---|---|---|---|
| 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 |

A′B′C

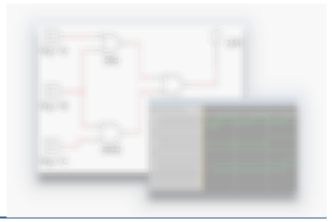
AB′C′

AB′C

ABC′

O = A′B′C + AB′C′ +AB′C + ABC′

# Product-of-Sums (γινόμενο αθροισμάτων)



**Product-of-Sums (POS)** expressions are another way of **representing truth tables**.

- A POS expression is a product of Boolean terms called **maxterms (μεγιστόροι)**.
- A maxterm is a summation of Boolean variables whose output equals 0.
- To generate a POS expression from a truth table, perform the following steps:
  1. Determine which rows of the table have an output of 0.
  2. Derive each row's maxterm, such that the output is 0 given that row's input state.
  3. Multiply the maxterms.

# Product-of-Sums (γινόμενο αθροισμάτων)



Below is an example of a **truth table conversion to an POS** expression.

| A | B | C | O |
|---|---|---|---|
| 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 |

$A + B + C$

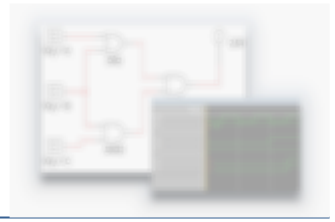
$A + B' + C$

$A + B' + C'$

$A' + B' + C'$

$$O = (A + B + C)(A + B' + C)(A + B' + C')(A' + B' + C')$$

# SOP and POS



The **SOP** and **POS** standard Boolean forms are powerful tools when applied to truth tables.

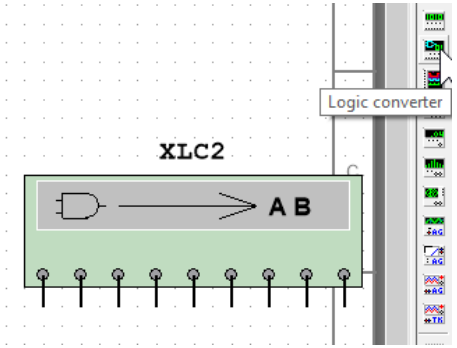
They can be used to derive a Boolean expression—and ultimately, an actual logic circuit.

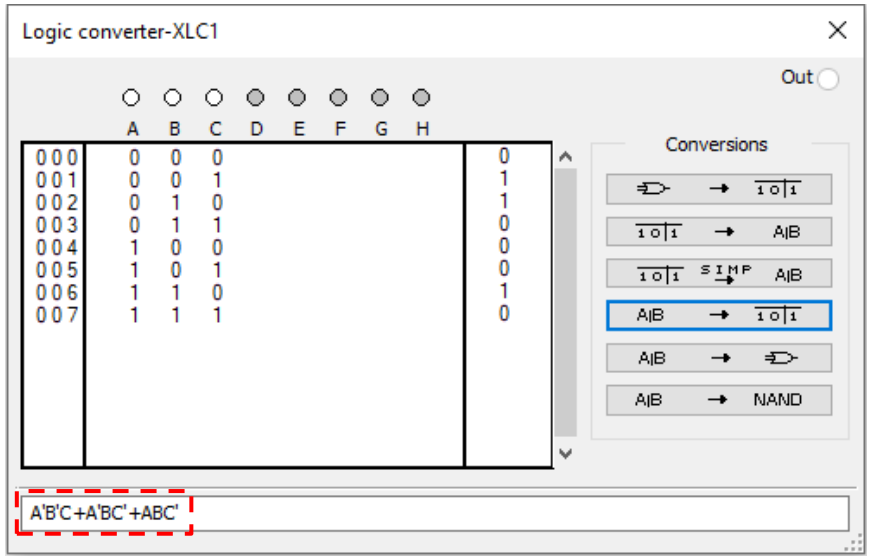
- When creating a circuit from **SOPs**, it would be constructed of **AND gates feeding into an OR gate**.
- When creating a circuit from **POSs**, it would be constructed of **OR gates feeding into an AND gate**.

# MultiSim - Logic Converter (Expression to Truth Table)

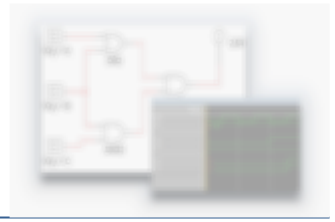
The **Logic Converter** is a great tool for checking truth tables and logic expressions. To build a Logic Converter circuit:

- Place the **Logic Converter** from the instruments toolbar on the right screen onto the circuit.
- **Double click** the Logic Converter to open its user interface.
- **Enter the SOP expression** that you want in the text field at the bottom of the window.
- Click the fourth button, **Expression to Truth Table**.



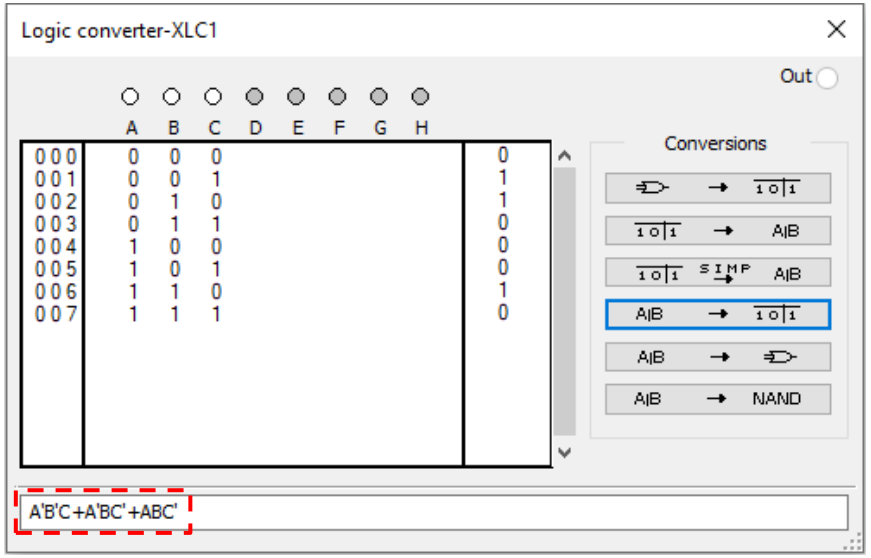



# MultiSim - Logic Converter (Simplified Expression)



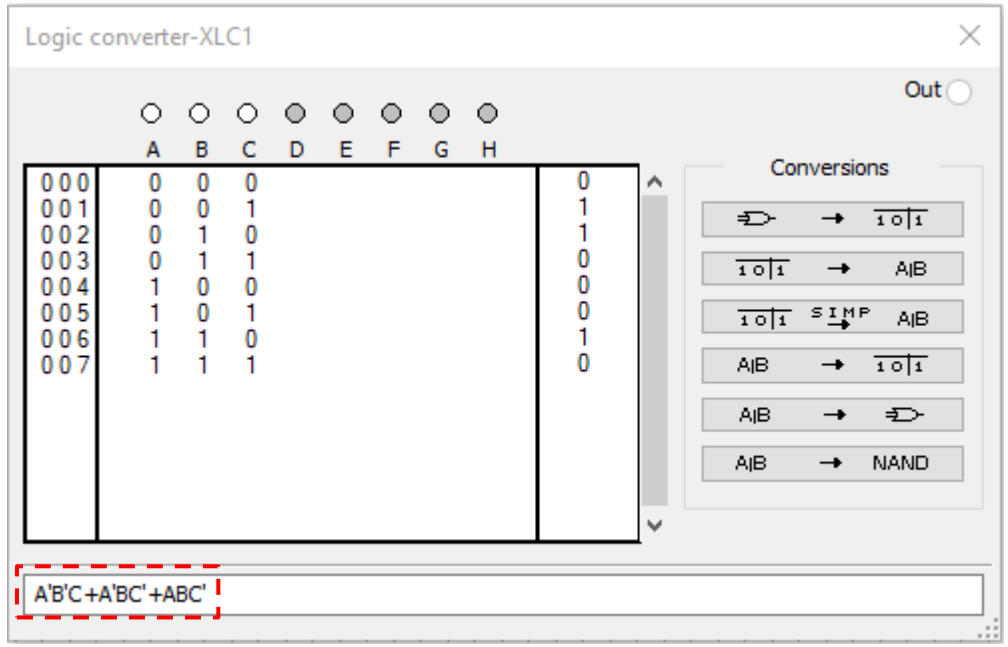
The **Logic Converter** can also **generate circuits** from POS and SOP expressions. This can save some time from doing the work manually.

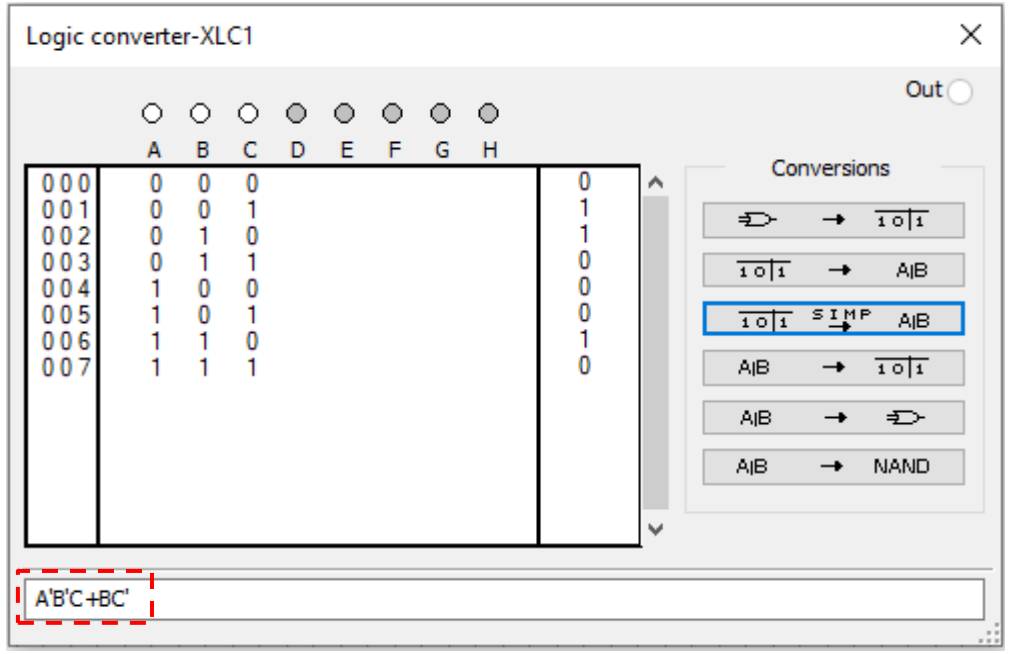
- **Enter the SOP expression** that you want in the text field at the bottom of the window.
- Click the fourth button, **Expression to Truth Table**.



# MultiSim - Logic Converter (Simplified Expression)

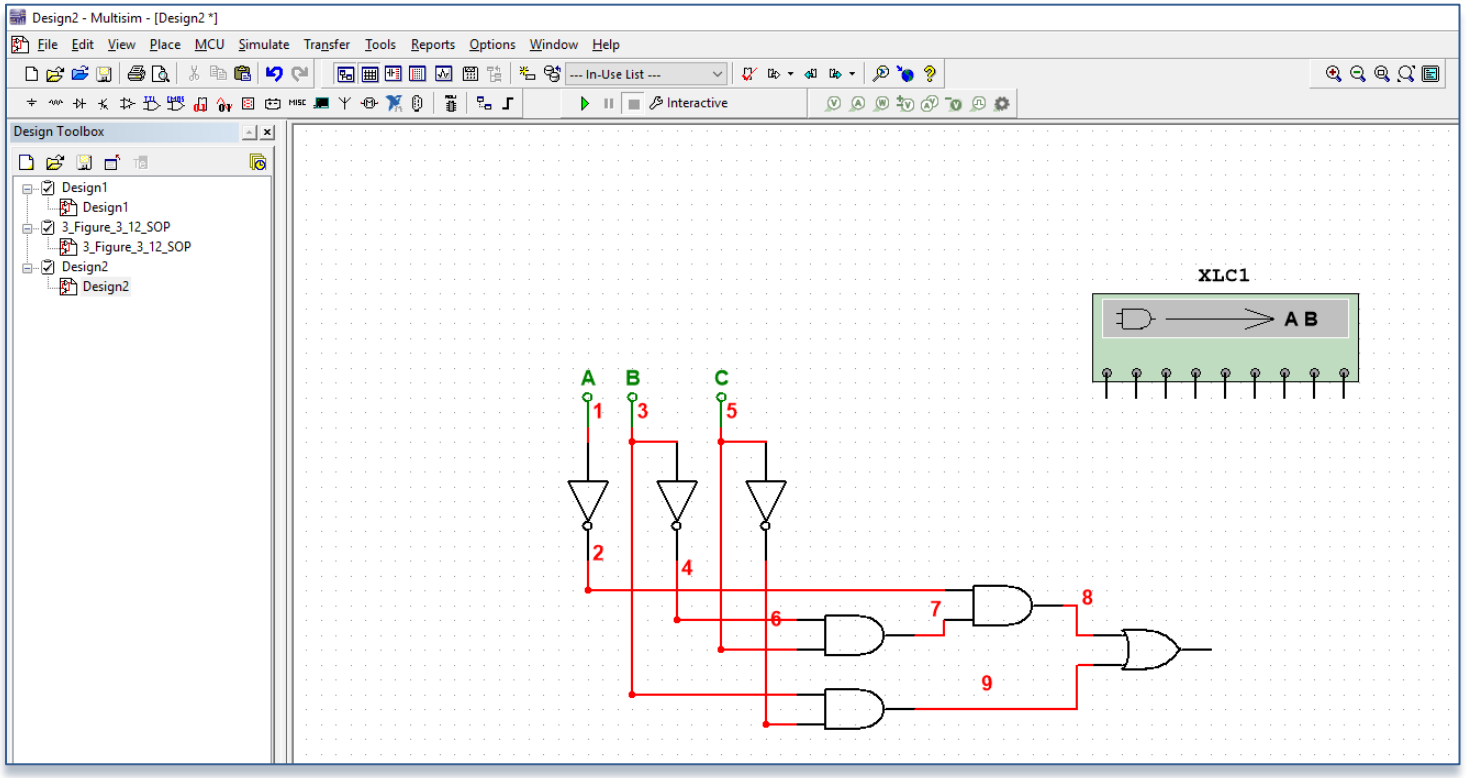
- Click the third button, **Truth Table to Simplified Expression**. This will simplify the expression if it can be simplified.



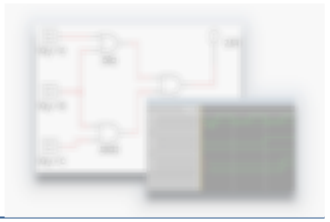



# MultiSim - Logic Converter (Expression to Circuit)

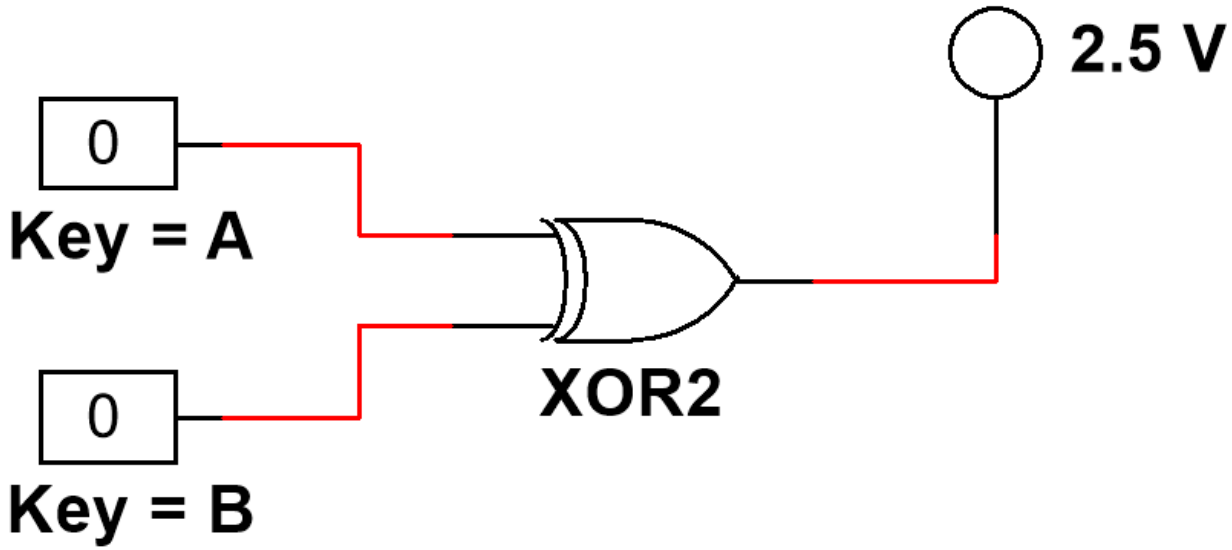
- Next, click the fifth button, **Expression to Circuit**.
- **Place the circuit** that it generates.



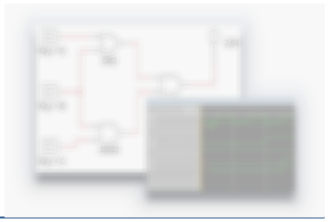
# Exercise: Building an XOR Logic Gate in Multisim



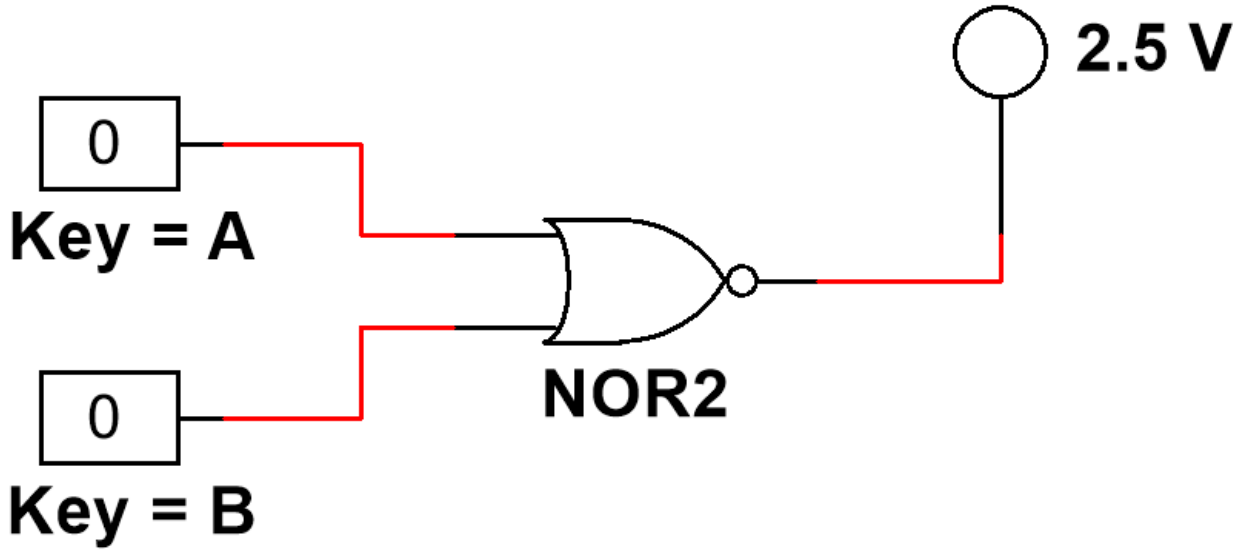
- Build the following circuit using an XOR gate:



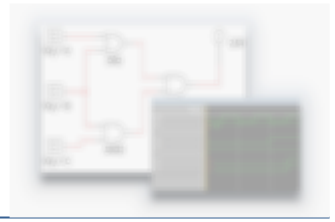
# Exercise: Building a NOR Logic Gate in Multisim



- Build the following circuit using an NOR gate:

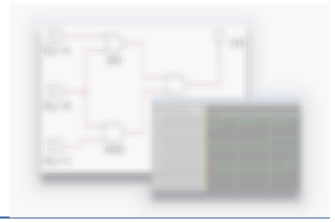


# Exercise: NOT - Gate

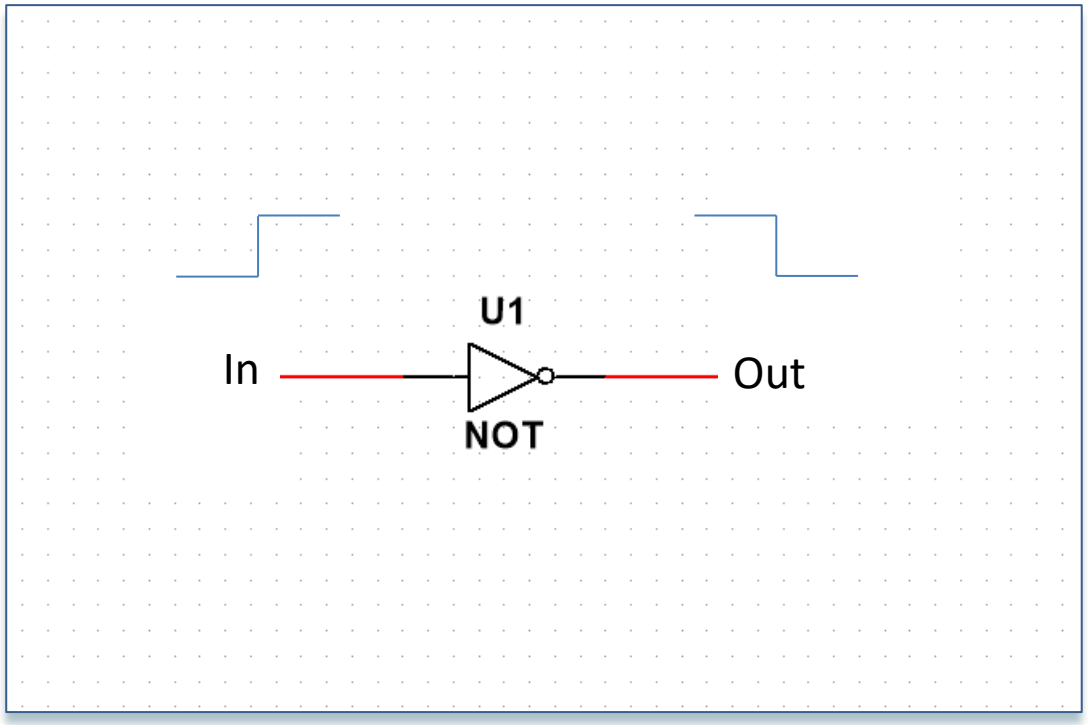


- Θέλουμε να μετρήσουμε και να συγκρίνουμε την **καθυστέρηση διάδοσης** του σήματος από την είσοδο μέχρι την έξοδο δύο αντιστροφέων διαφορετικής τεχνολογίας.
- Οι αντιστροφείς που θα συγκριθούν είναι οι εξής:
  - CMOS / CMOS_5V / **4009BD_5V**
  - TTL / 74STD / **7404N**
- Συνδέστε στην είσοδο και των δύο πυλών ένα ρολόι παραγωγής τετραγωνικών παλμών με συχνότητα 1 MHz.
- Χρησιμοποιώντας έναν **παλμογράφο** παρατηρείστε την είσοδο του παλμού στο κύκλωμα και την έξοδο της κάθε πύλης χρησιμοποιώντας τρία κανάλια της συσκευής.
- Υπολογίστε τον χρόνο καθυστέρησης στη διάδοση του σήματος κάθε αντιστροφέα, για την μετάβαση της εξόδου του από 0→1 καθώς και για μετάβαση από 1→0.
- Η προσομοίωση να γίνει με το Multisim.

# Multisim - Παλμογράφος



- Θέλουμε να μετρήσουμε την **καθυστέρηση διάδοσης** του σήματος από την είσοδο ενός αντιστροφέα στην έξοδό του.



# Multisim - Παλμογράφος

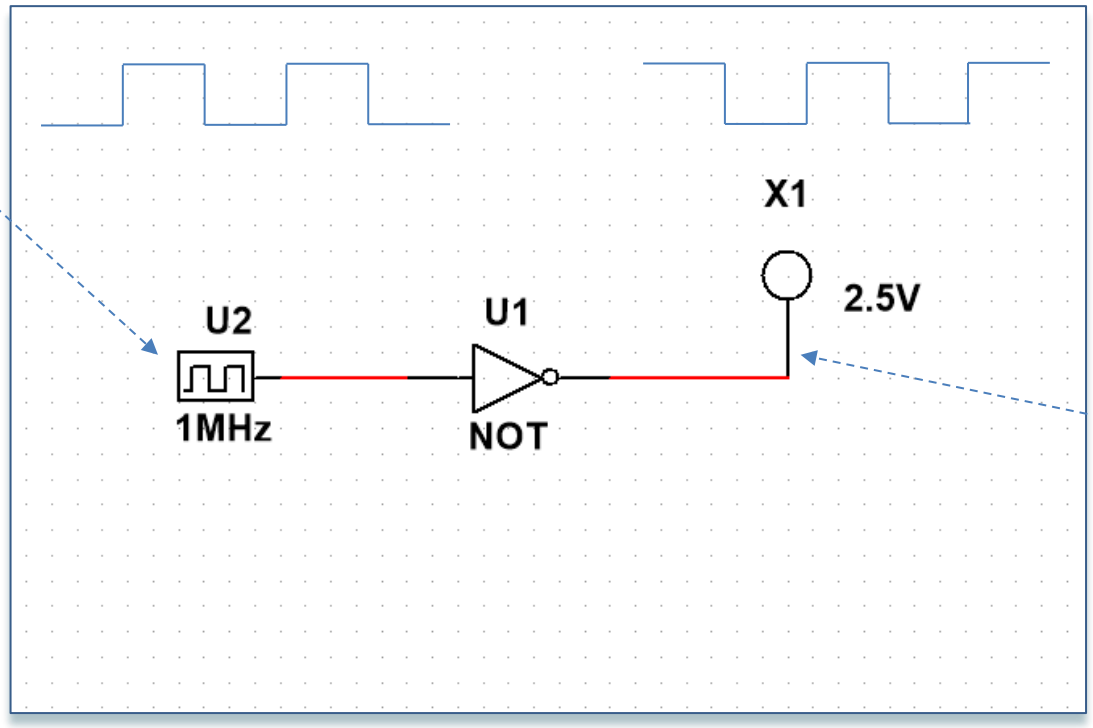
- Θέλουμε να μετρήσουμε την **καθυστέρηση διάδοσης** του σήματος από την είσοδο ενός αντιστροφέα στην έξοδό του.

Συνδέουμε ένα **ρολόι** τετραγωνικών παλμών στην είσοδο

Θα το βρούμε στο: Sources/DIGITAL_SOURCES /DIGITAL_CLOCK

Διπλό κλικ και ρυθμίζουμε την συχνότητα στο 1MHz

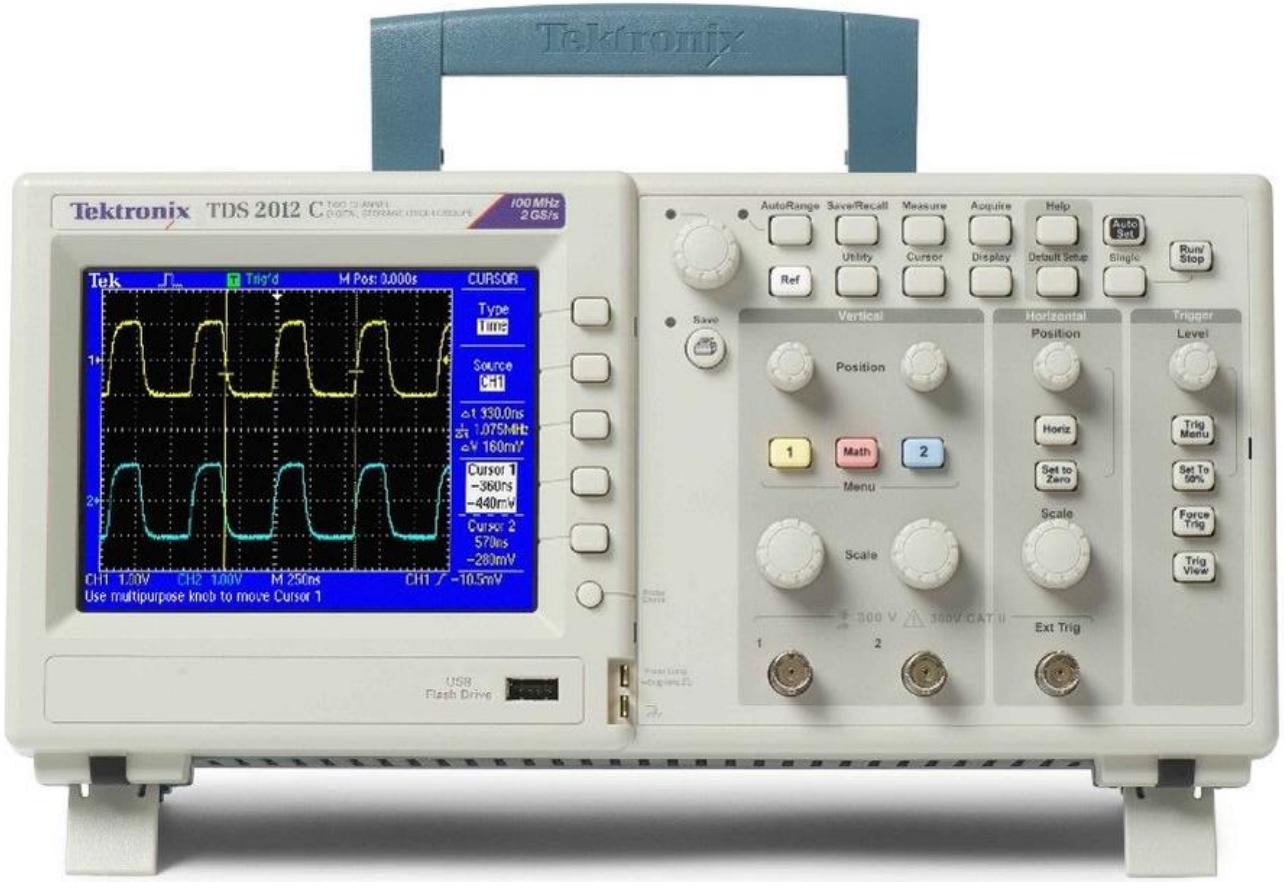
Το **clock** είναι μία γεννήτρια σήματος όπου εναλλάσσει την τιμή του από “0” σε “1” επαναληπτικά στην συχνότητα που το έχουμε ορίσει




Συνδέουμε ένα Probe στην έξοδο

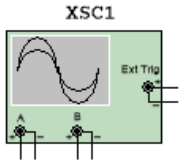
# Multisim - Παλμογράφος

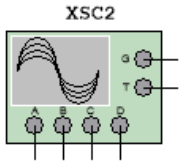
- Ο **Παλμογράφος** είναι ένα εργαστηριακό όργανο για την μέτρηση όχι μόνο της τιμής μεγεθών σε ηλεκτρικά/ηλεκτρονικά κυκλώματα αλλά και της χρονικής τους εξέλιξης τους.

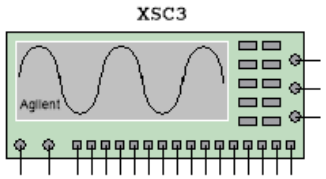


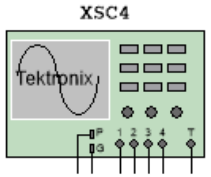
# Multisim - Παλμογράφος

- Στο Multisim υπάρχουν **4 διαφορετικοί παλμογράφοι**
- Βρίσκονται όλοι στην δεξιά μπάρα των οργάνων

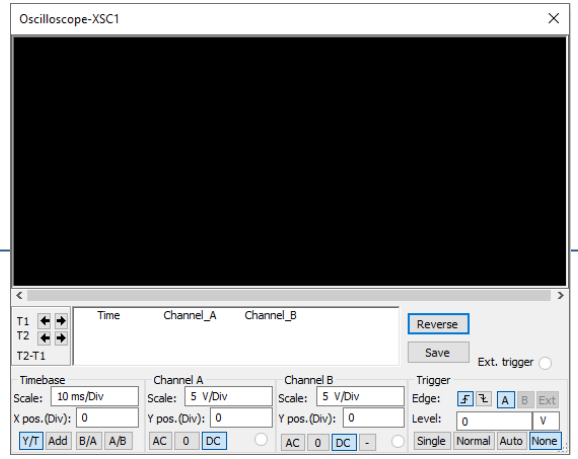


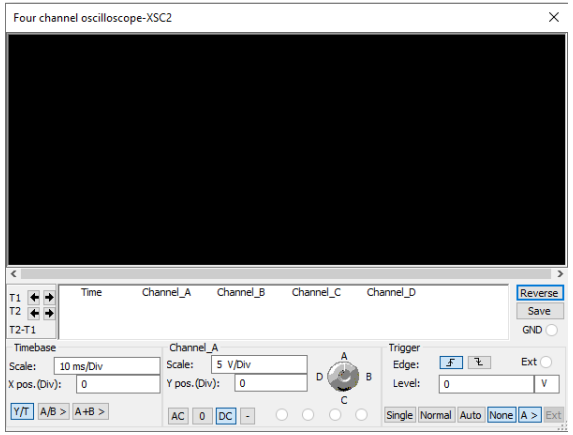



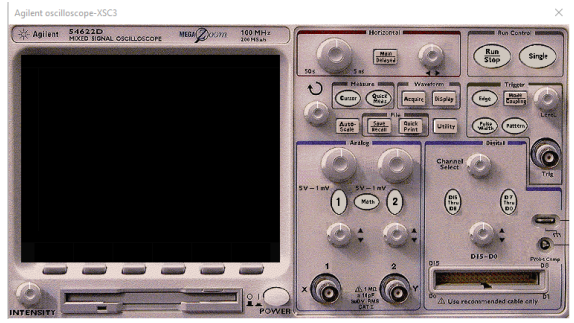



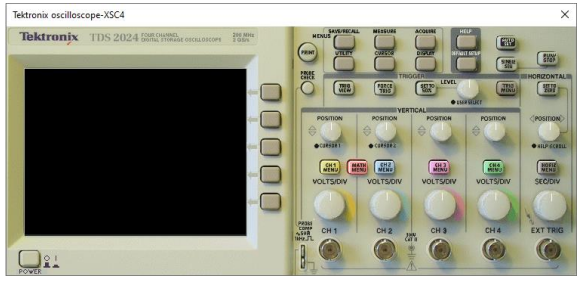




- Με 2 κανάλια (Virtual)
- Με 4 κανάλια (Virtual)
- Agilent oscilloscope – XSC3
- Tektronix oscilloscope – XSC4



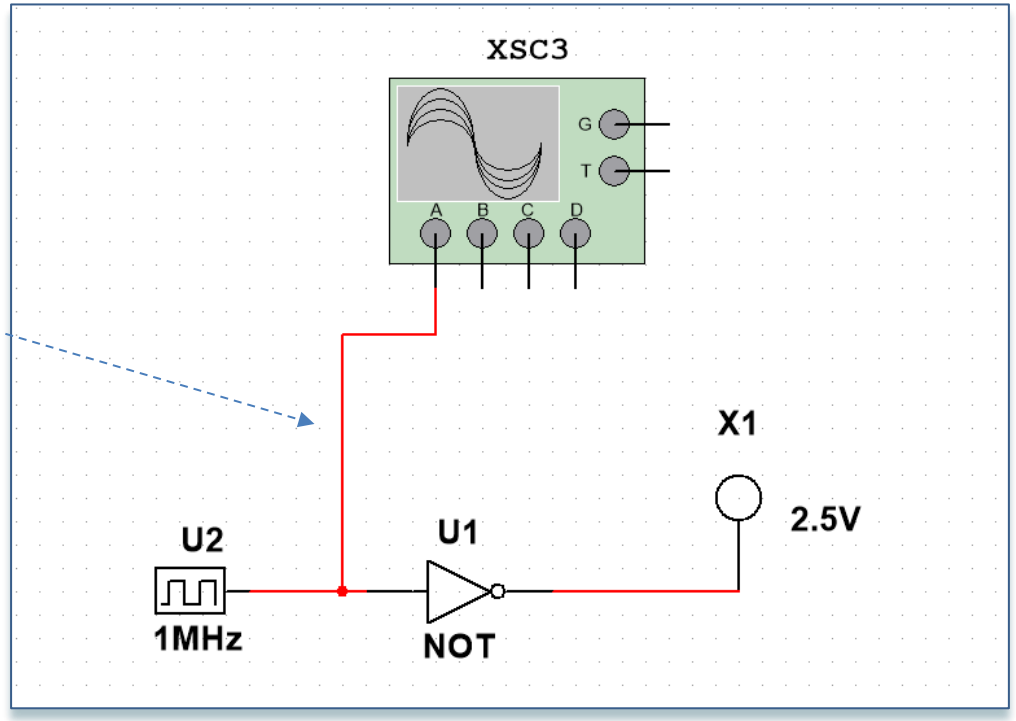



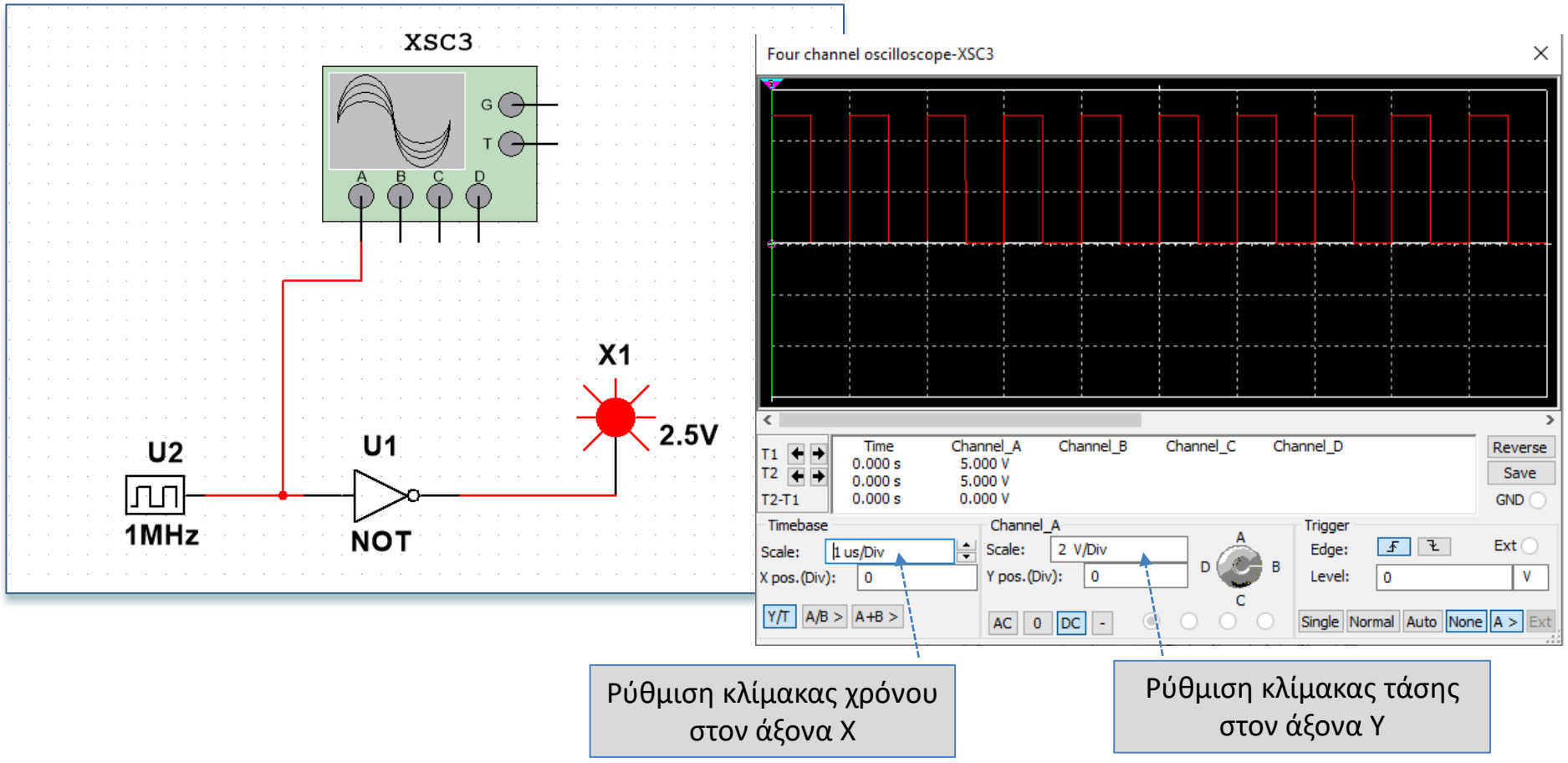
# Multisim - Παλμογράφος

Συνδέουμε το Α κανάλι του παλμογράφου στην είσοδο του αντιστροφέα



# Multisim - Παλμογράφος

Πιέζουμε **Run** και έπειτα από λίγο **Stop**

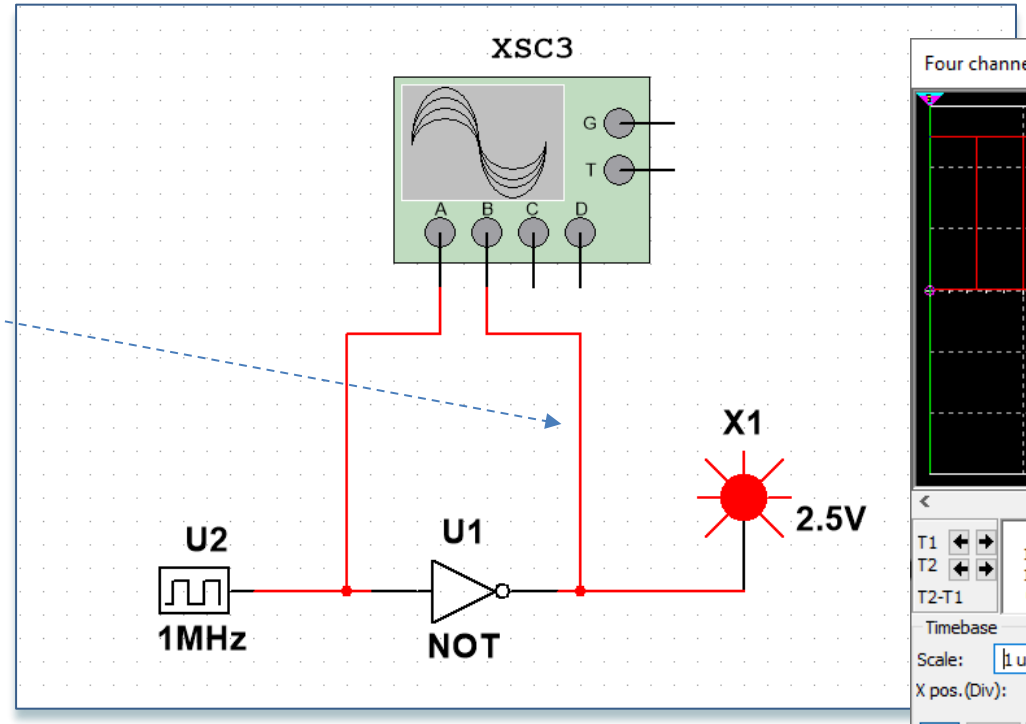


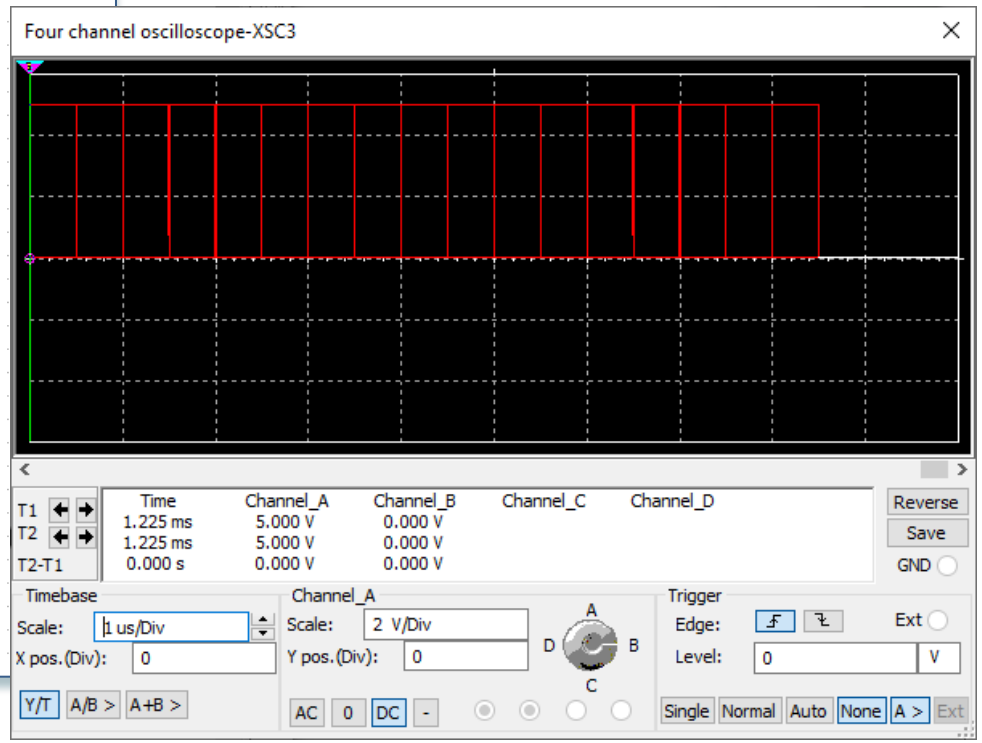

Ρύθμιση κλίμακας χρόνου στον άξονα X

Ρύθμιση κλίμακας τάσης στον άξονα Y

# Multisim - Παλμογράφος

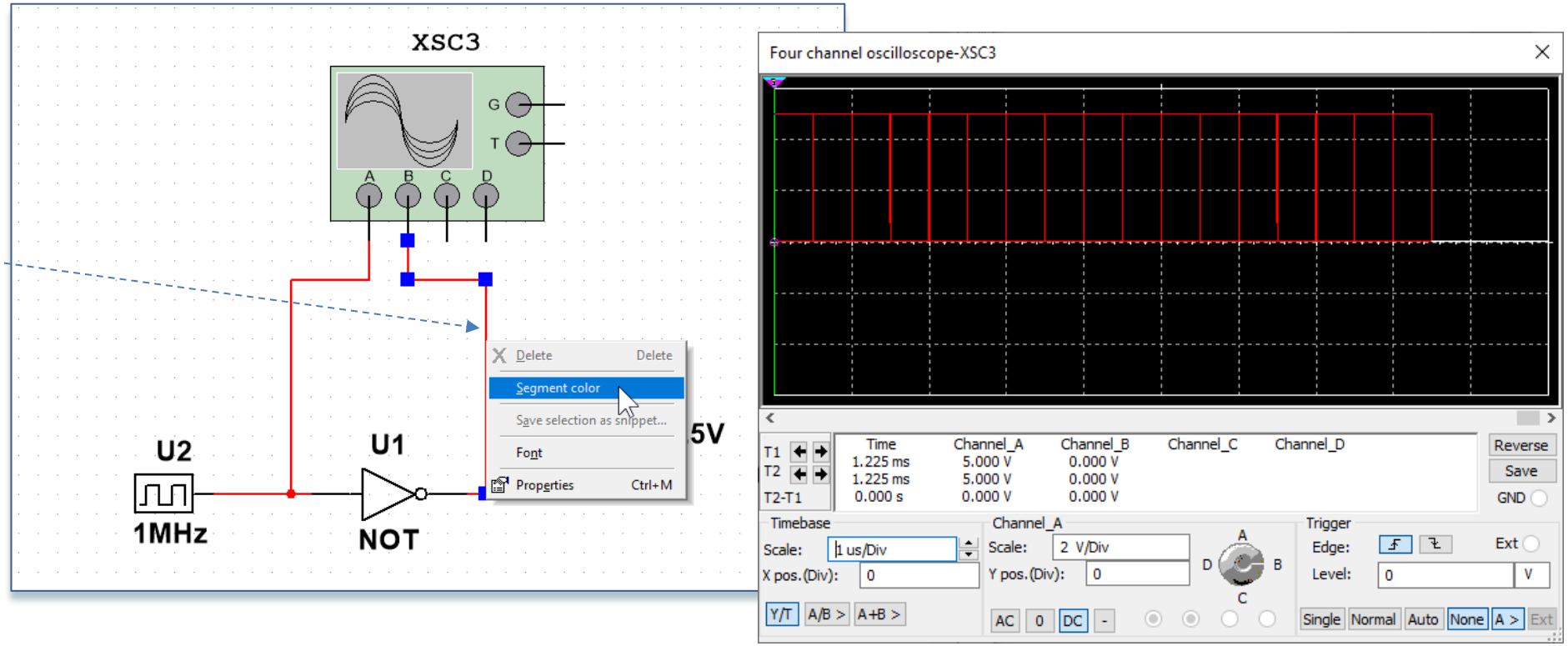
Συνδέουμε και το κανάλι Β του παλμογράφου στην έξοδο του αντιστροφέα



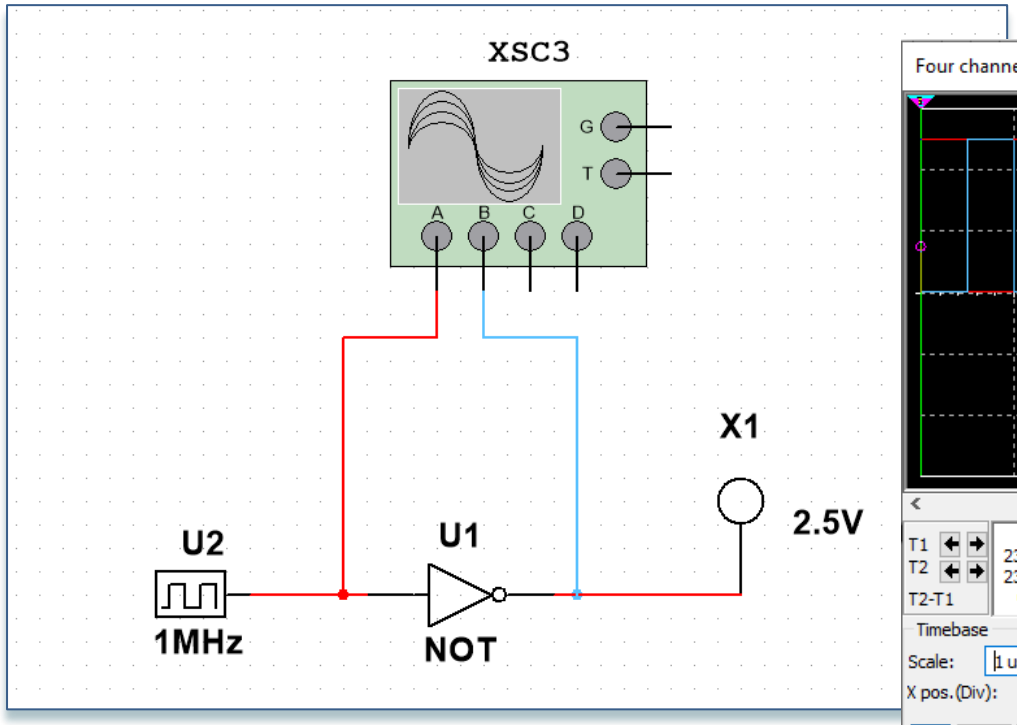



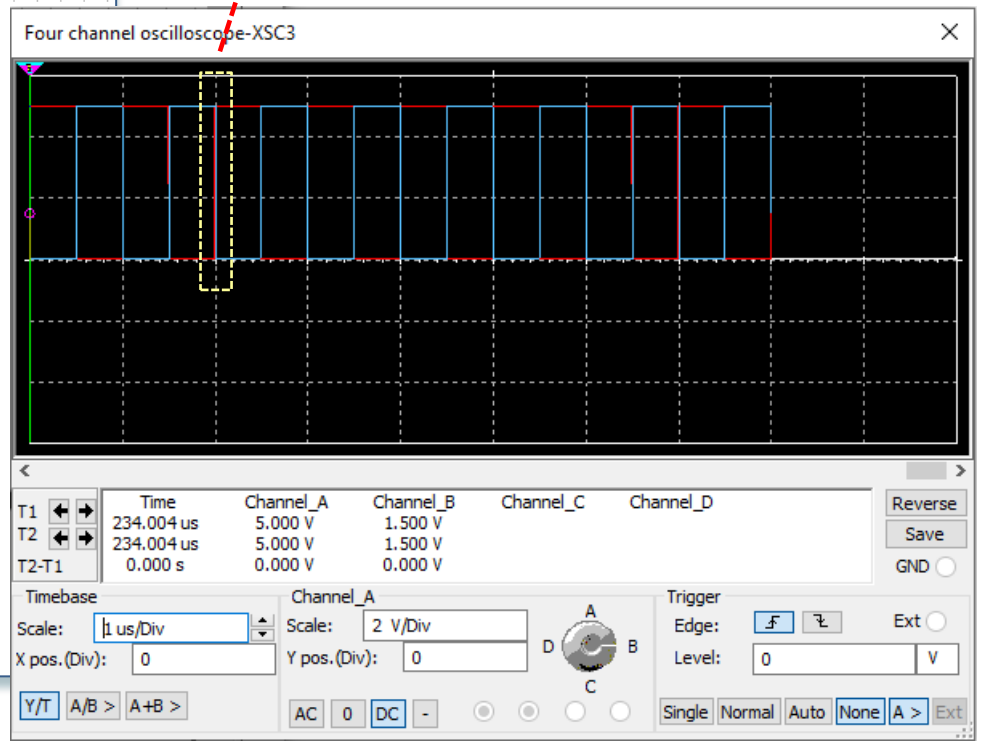
# Multisim - Παλμογράφος

Τροποποιούμε το χρώμα απεικόνισης του Β καναλιού.



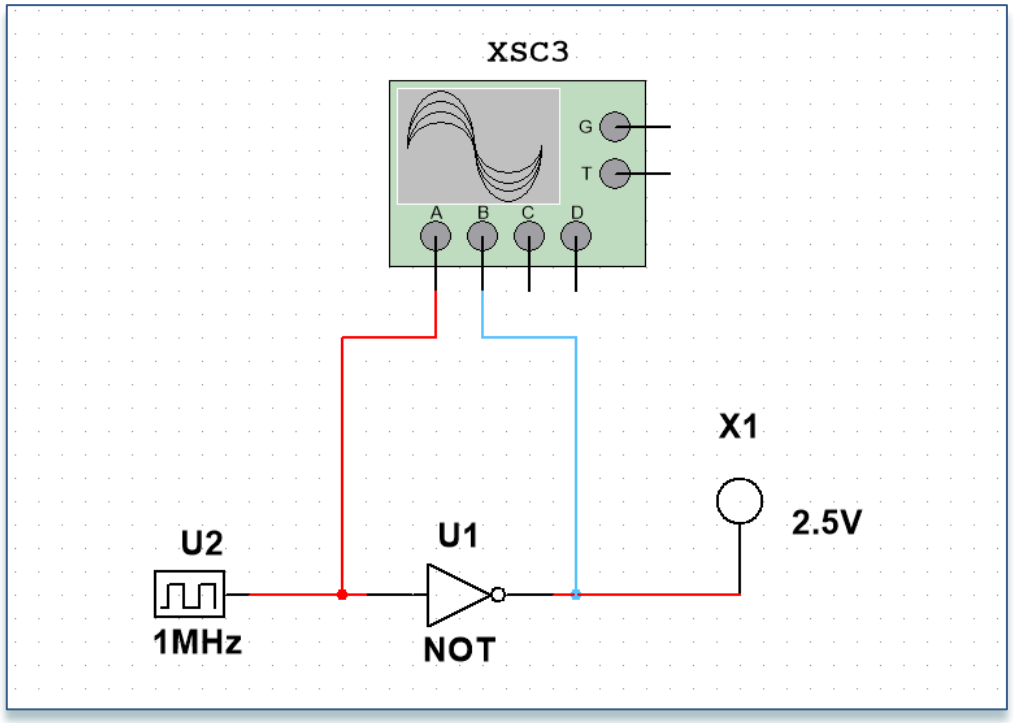
# Multisim - Παλμογράφος

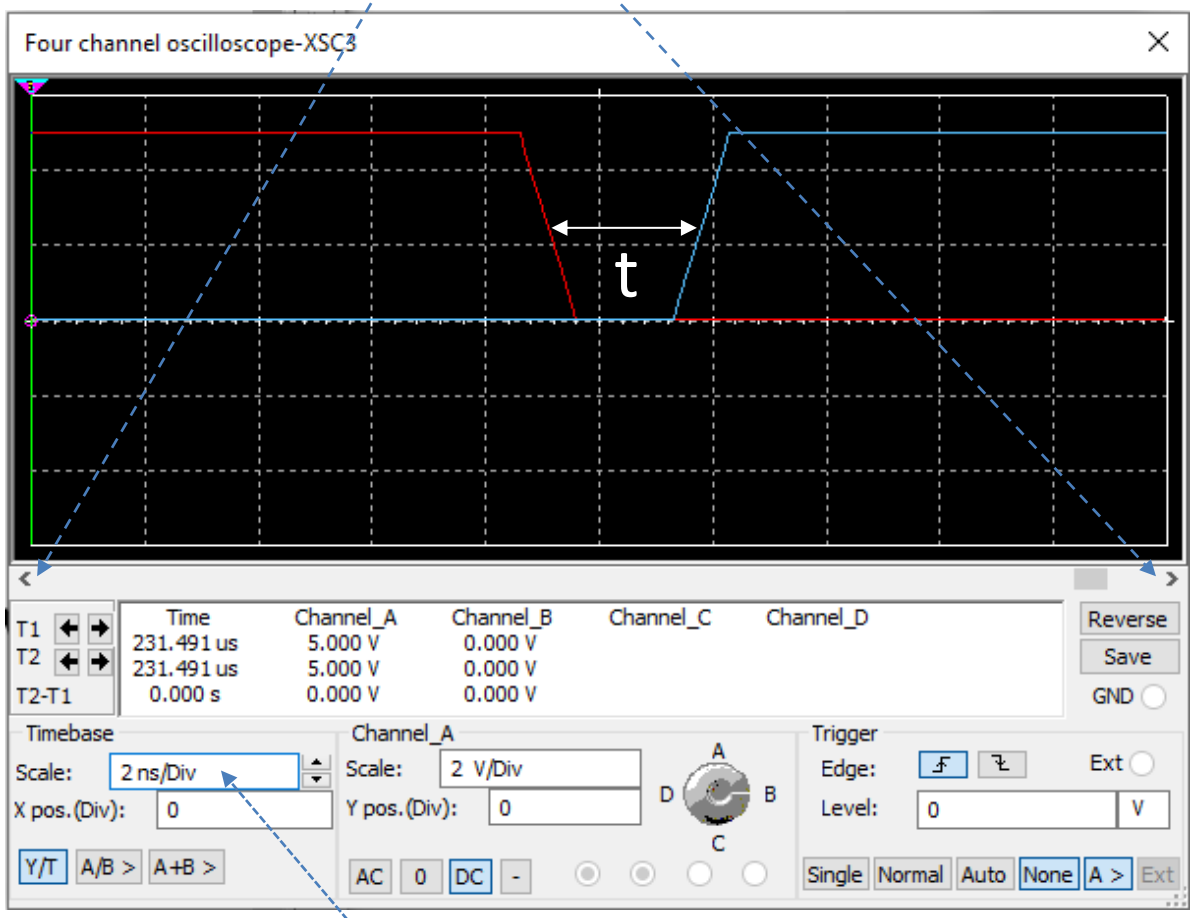


# Multisim - Παλμογράφος

Μετακινούμε το γράφημα δεξιά ή αριστερά ώστε να βρούμε το σημείο της μετάβασης




Μειώνουμε την κλίμακα στον άξονα Χ (χρόνος) ώστε να γίνει ορατή η πολύ μικρή χρονοκαθυστέρηση.

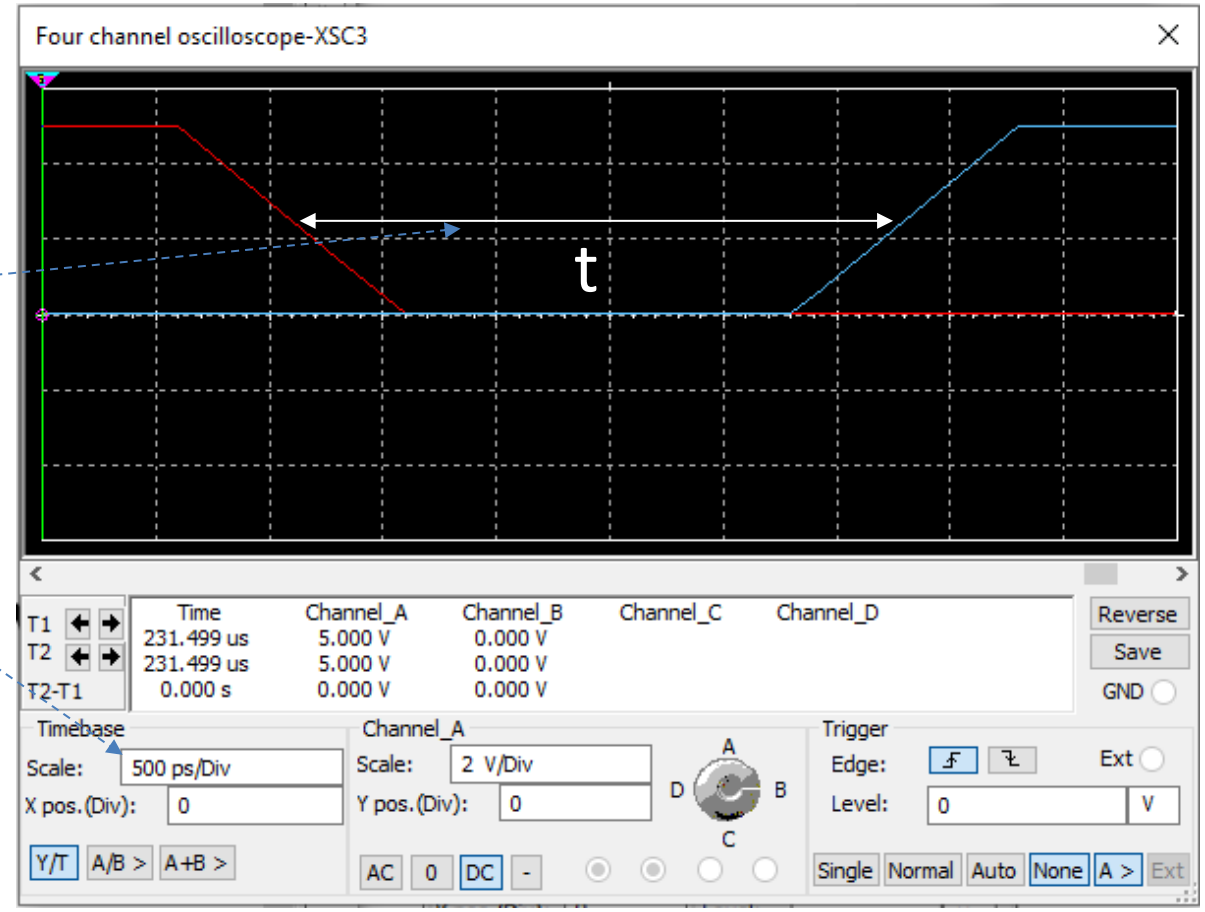
# Multisim - Παλμογράφος

Μειώνουμε την κλίμακα τόσο ώστε να βρίσκονται και οι δύο μεταβάσεις μέσα στο διάγραμμα

Μετράμε περίπου 5,3 “κουτάκια”

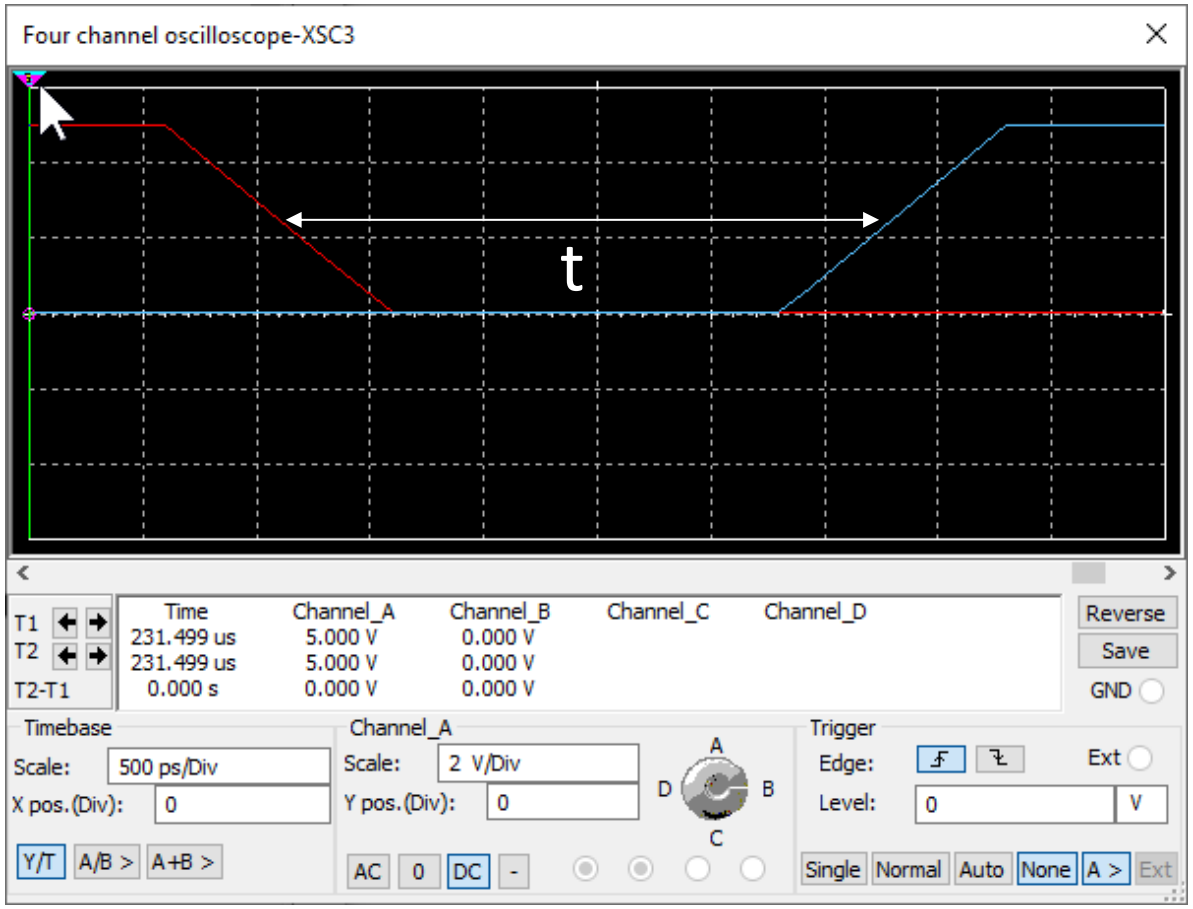
Η κλίμακα είναι 500psec/Div οπότε οπτικά για μετάβαση του αντιστροφέα από 0→1 υπολογίζουμε περίπου καθυστέρηση μετάβασης **t = 5,3 x 500ps = 2,65 nsec**

**Σημείωση**: Μετράμε από το μέσο της μεταβολής του σήματος εισόδου μέχρι το μέσο της μεταβολής του σήματος εξόδου.



# Multisim - Παλμογράφος

Για ακριβέστερη μέτρηση μπορούμε να χρησιμοποιήσουμε τους cursors που διαθέτει ο παλμογράφος.

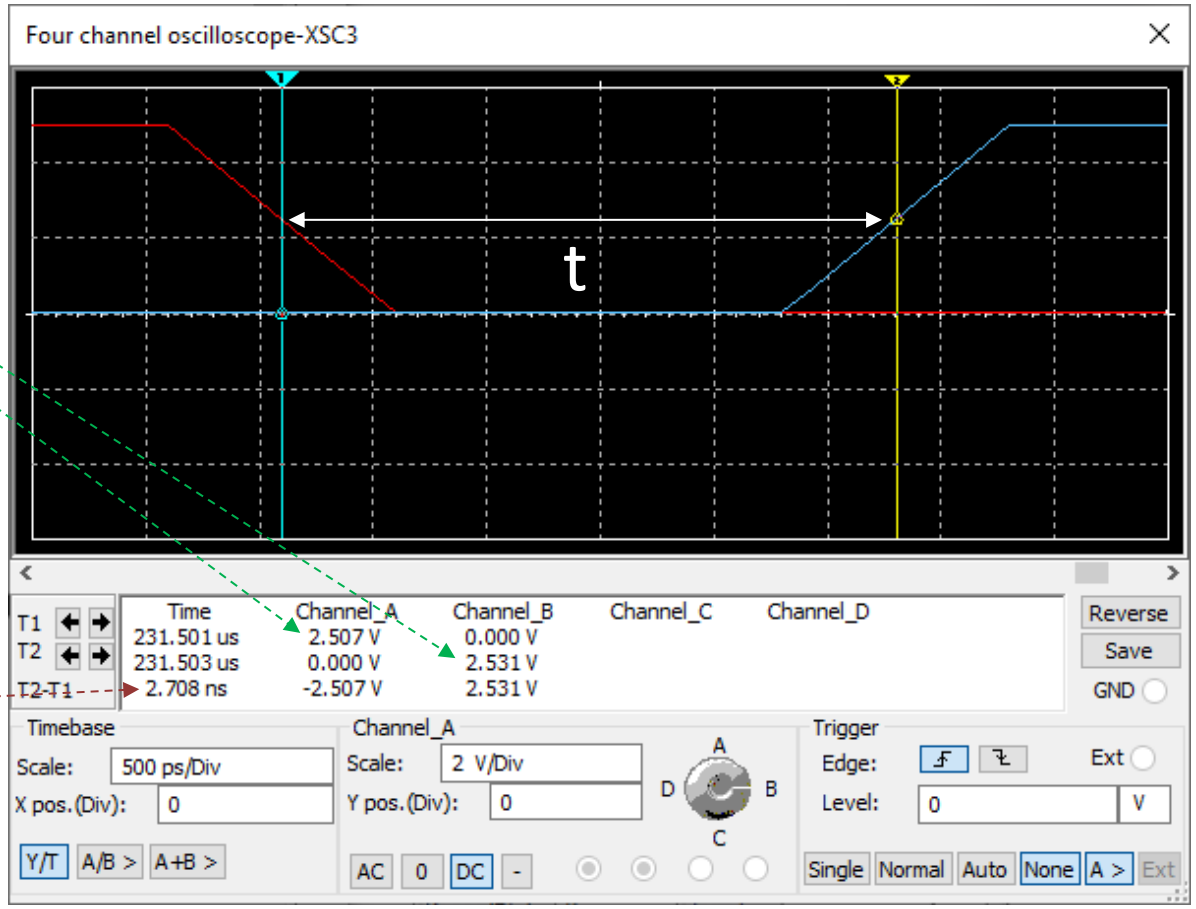


# Multisim - Παλμογράφος

Μετακινούμε τους cursors ώστε να βρίσκονται όσο πιο κοντά στο μέσο της μετάβασης τους, το οποίο βρίσκεται στα 2,5 Volt

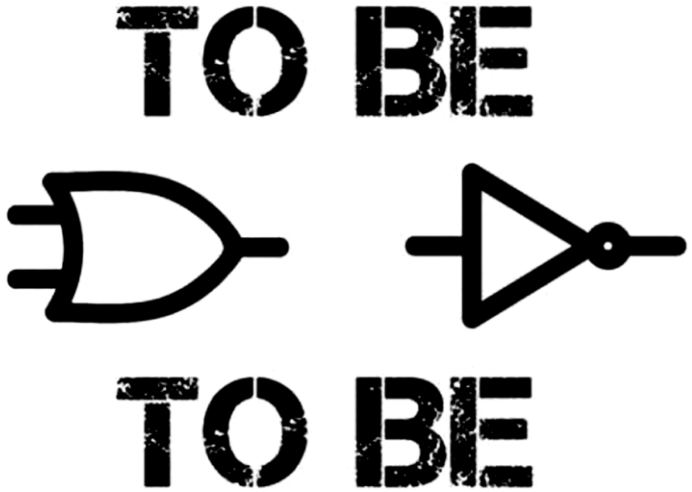
Ο παλμογράφος υπολογίζει την χρονική διαφορά μεταξύ των δύο cursors.

Η χρονική αυτή διαφορά είναι η καθυστέρηση διάδοσης του αντιστροφέα μας για μετάβαση 0→1 και είναι **t=2,708 nsec**



# Ευχαριστώ για την προσοχή σας!

- **Ερωτήσεις / Απορίες ;**




Επικοινωνία: ece119.uth@gmail.com