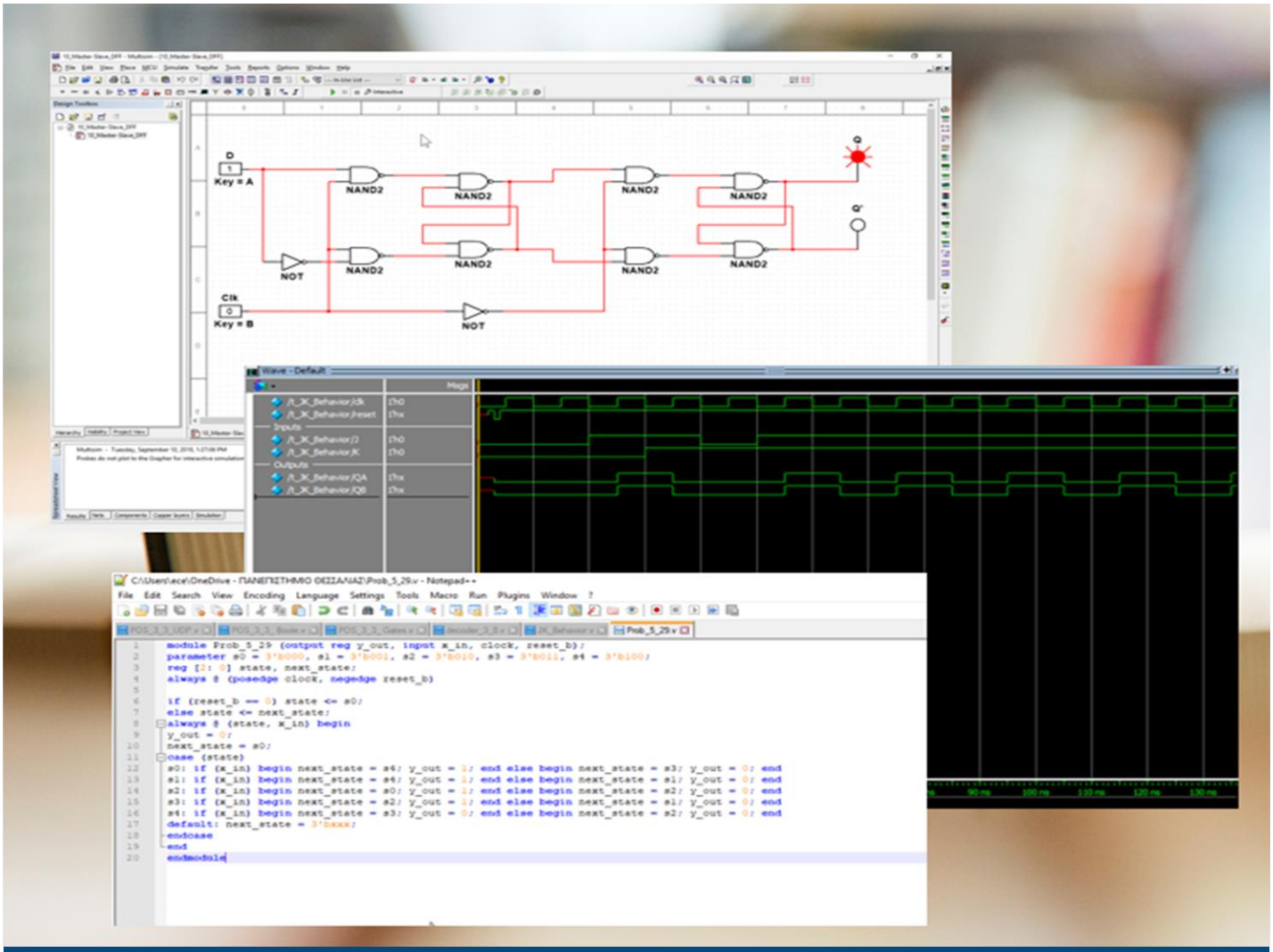
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ



ECE119 Ψηφιακή Σχεδίαση

Εργαστηριακές ασκήσεις, Multisim - Verilog

Lab 7: Multiplexers and Demultiplexers

Καραμπερόπουλος Δημήτρης

Σεπτέμβριος 2024

Lab 7

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 7: Multiplexers and Demultiplexers

Multiplexers are combinational logic circuits for which there are multiple potential inputs but there is always only one output.

Demultiplexers are the opposite in that there is always one input but there are multiple potential outputs.

Both multiplexers and demultiplexers have a bit (or multiple bits) called selector bit(s) which is responsible for determining which input or output is chosen.

Like encoders and decoders, multiplexers and demultiplexers can be broken down into circuit components but are typically represented by chips for visual simplification.

In this lab, we will analyze multiplexers and demultiplexers in both their circuit and chip forms.

Learning Objectives

In this lab, students will:

1. Reflect on the similarities and differences between encoders and multiplexers
2. Examine the function of a basic 2-to-1 Multiplexer using logic gates

Expected Deliverables

In this lab you will collect the following deliverables:

- Sum-of-Products Boolean functions for 2-to-1 Multiplexer
- Sum-of-Products Boolean functions for 1-to4 Demultiplexer
- Image of circuit
- Observations of demultiplexer behavior
- Conclusion questions

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

7.1 Theory and Background

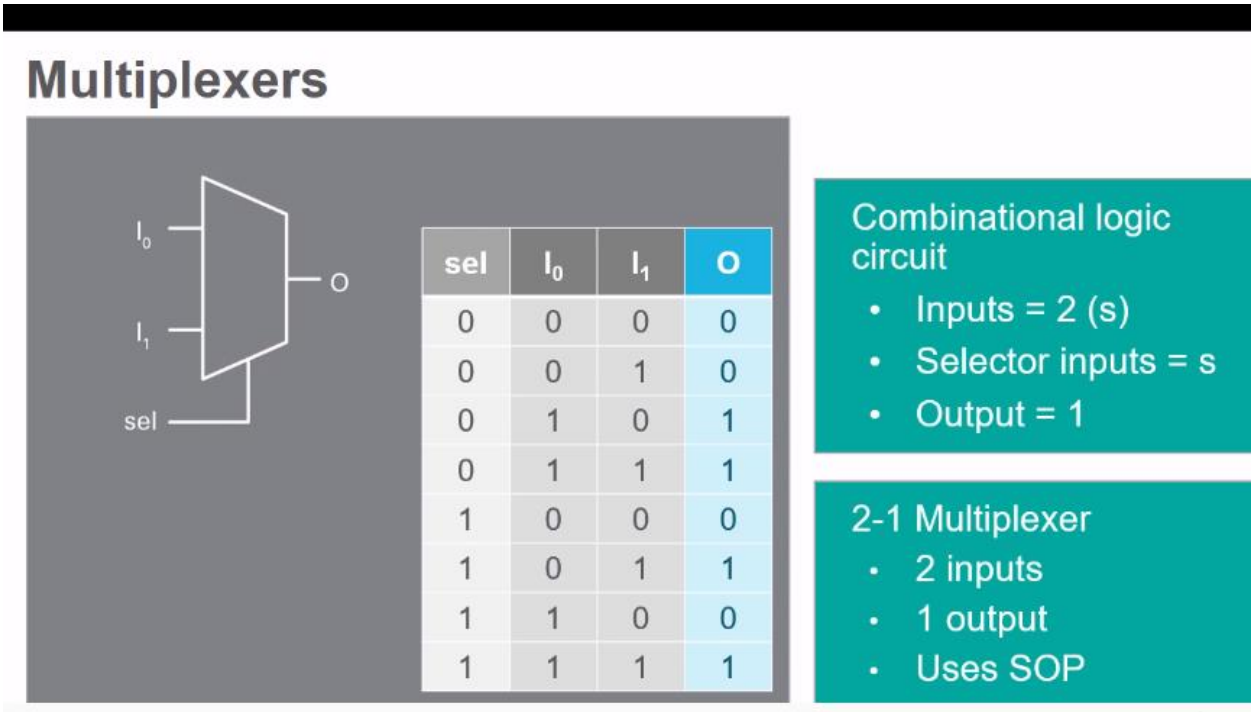


Figure 7-1 Video. View the video here: https://youtube/khmQ-LT_Cxg

Video Summary

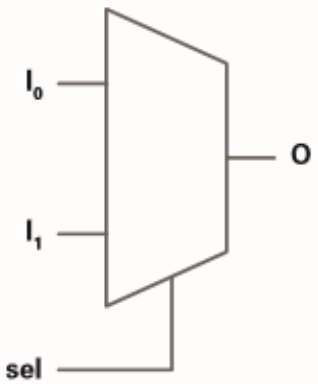
- Multiplexers are combinational logic circuits
- Clock multiplexing is used for operating the same logic function at different clock rates from different sources
- Demultiplexers are combinational logic circuits that have the opposite function of a multiplexer

Multiplexers

The *multiplexer*, abbreviated *MUX*, is a combinational logic circuit which has multiple data inputs, one or more select inputs and one output.

- It passes the data on one of the inputs, depending on the selection signals, to the output
- With the help of this logic circuit, multiple signals can share the same data output
- Multiplexers have 2^s inputs and s selector lines, which determine which of the inputs to output.
- Multiplexers are one of the most widely used combinational circuits, their application areas include:
    - Data routing
    - Operation sequencing
    - Parallel-to-serial conversion
    - Waveform generation

The simplest circuit is the 2-to-1 multiplexer, with the graphical symbol presented in the leftmost figure. Its functionality is described by the joining truth table. The multiplexer below is only 1-bit wide since bit line is connected to a single output bit line.



sel	I_0	I_1	O
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Figure 7-2 Image of 2-to-1 multiplexer (left) and truth table (right)

The truth table can be simplified to the following truth table for a bitter understanding of the circuit's operation:

sel	O
0	I_0
1	I_1

Figure 7-3 Simplified truth table

Using the sum-of-products Boolean function gives the following combinational logic circuit:

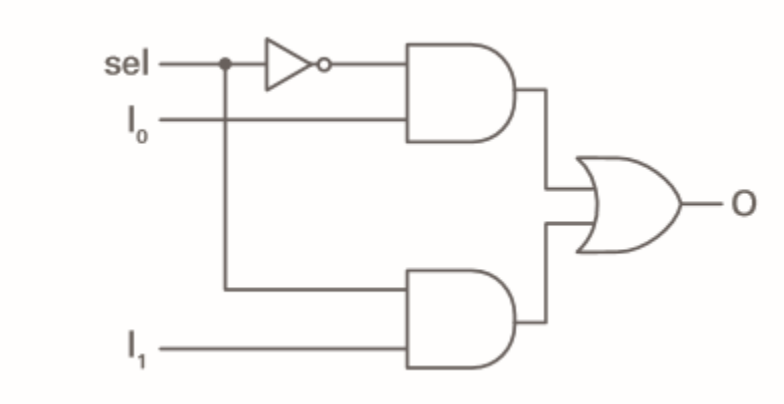
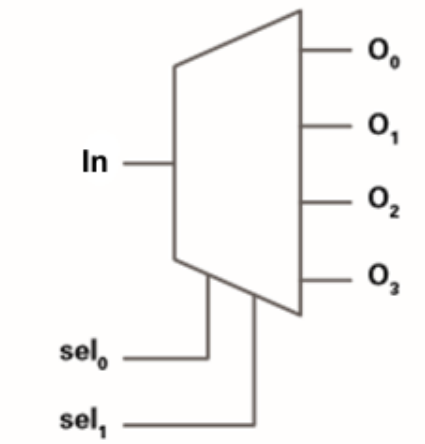


Figure 7-4 Combinational logic circuit

Demultiplexers

Demultiplexers (DEMUX) have the opposite function of a multiplexer

- It places the value of a single data input on several data outputs depending on a selection signal
- Usually demultiplexers have s select inputs and 2^s outputs
- Since demultiplexers take one input and connect it to many outputs, some of their uses are for communication (two-way communication usually includes both multiplexers and demultiplexers) and for serial to parallel converters
- The graphical symbol for a 1-to-4 demultiplexer is shown below as well as the corresponding 1-to-4 DEMUX truth table and the CLC



sel_1	sel_0	O_3	O_2	O_1	O_0
0	0	0	0	0	In
0	1	0	0	In	0
1	0	0	In	0	0
1	1	In	0	0	0

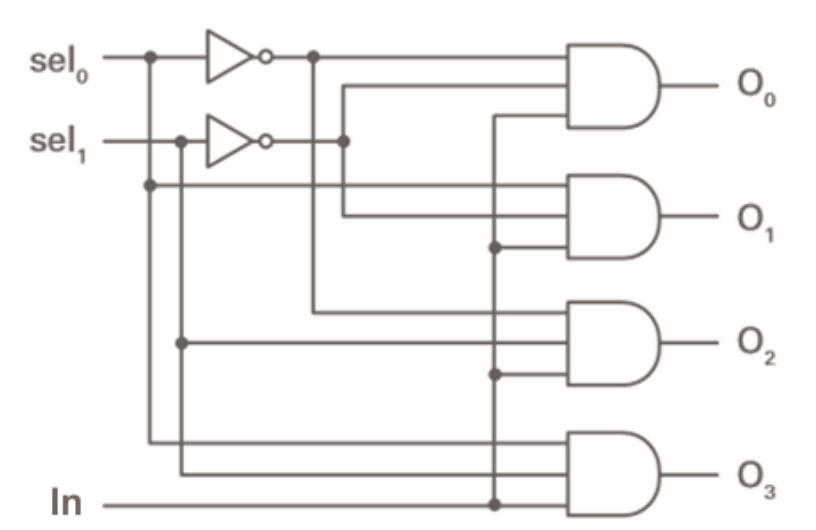


Figure 7-5 Demultiplexer (top left), truth table (top center) and CLC (bottom)

Check Your Understanding

Note: The following questions are meant to help you self-assess your understanding so far. You can view the answer key for all "Check your Understanding" questions at the end of the lab.

7-1 Write the sum-of-products Boolean functions for the 2-to-1 Multiplexer (simplified):
Give names: x, y, sel, out

out=_____

7-2 Write the sum-of-products Boolean functions for the 1-to-4 Demultiplexer:
Give names: x, sel0, sel1, out0, out1, out2, out3

out0=_____

out1=_____

out2=_____

out3=_____

7-3 What is the function of the Selector (Sel) in Multiplexers and Demultiplexers?

a) In Multiplexers the Sel selects the output to which the input is mapped and in Demultiplexers the Sel selects the input that is outputted.

b) In Multiplexers the Sel selects the input that is outputted and in Demultiplexers the Sel selects the output to which the input is mapped.

7.2 Implement: Multiplexers Using Logic Gates

Circuit 1

Build and connect the following circuit in Multisim:

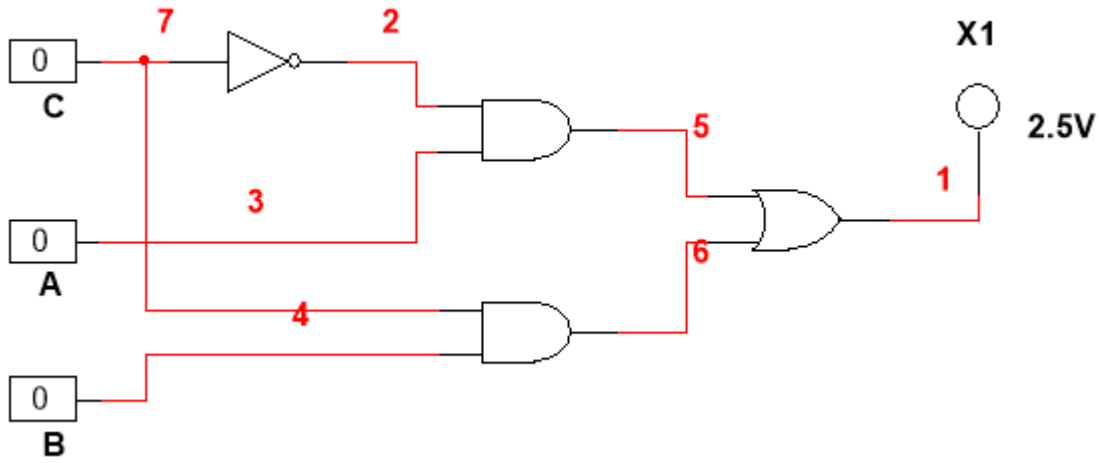


Figure 7-6 Circuit diagram

- Multisim:	Όνομα αρχείου “**7_Circuit_1.ms14**”.
	Προσθήκη στο zip file με όνομα ”Lab7_ονοματεπώνυμο_AM.zip”
ή	
- MultisimLive:	Όνομα αρχείου “**7_Circuit_1.png**”.
Schematic image	Προσθήκη στο zip file με όνομα ”Lab7_ονοματεπώνυμο_AM.zip”

- Set the **Input C** to **0**.
- Toggle **Inputs A** and **B**. Notice that the output is determined by Input A and the behavior of Input B has no effect on the output.
- Set the **Input C** to **1**.
- Toggle **Inputs A** and **B**. Notice that the output is determined by the value of Input B and the value of Input A has no effect on the output.

7-4 What Is this circuit?

a) 2-to-1 demultiplexer

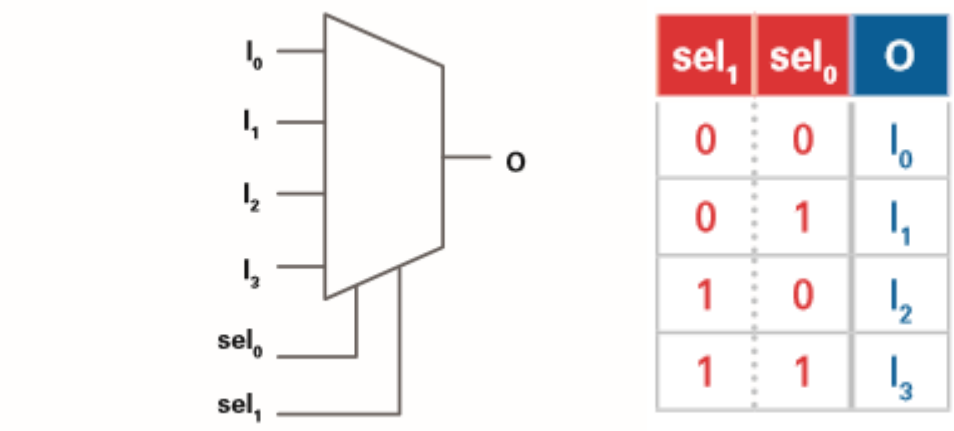
b) 1-to-2 multiplexer

c) 2-to-1 multiplexer

d) 1-to-2 demultiplexer

4-to-1 MUX

Using the following truth table (right) to describe the behavior of a 4-to-1 MUX (left), design and implement the corresponding CLC.



sel_1	sel_0	O
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Figure 7-7 Image of 4-to-1 MUX (left) and truth table (right)

Multisim .ms14

Picture

- Multisim:	Όνομα αρχείου “**7_Mux_4to1.ms14**”.
	Προσθήκη στο zip file με όνομα ”Lab7_ονοματεπώνυμο_AM.zip”
ή	
- MultisimLive:	Όνομα αρχείου “**7_Mux_4to1.png**”.
Schematic image	Προσθήκη στο zip file με όνομα ”Lab7_ονοματεπώνυμο_AM.zip”

7.3 Implement: Demultiplexer

1-to-4 Demultiplexer

Build and run the following 1-to-4 demultiplexer

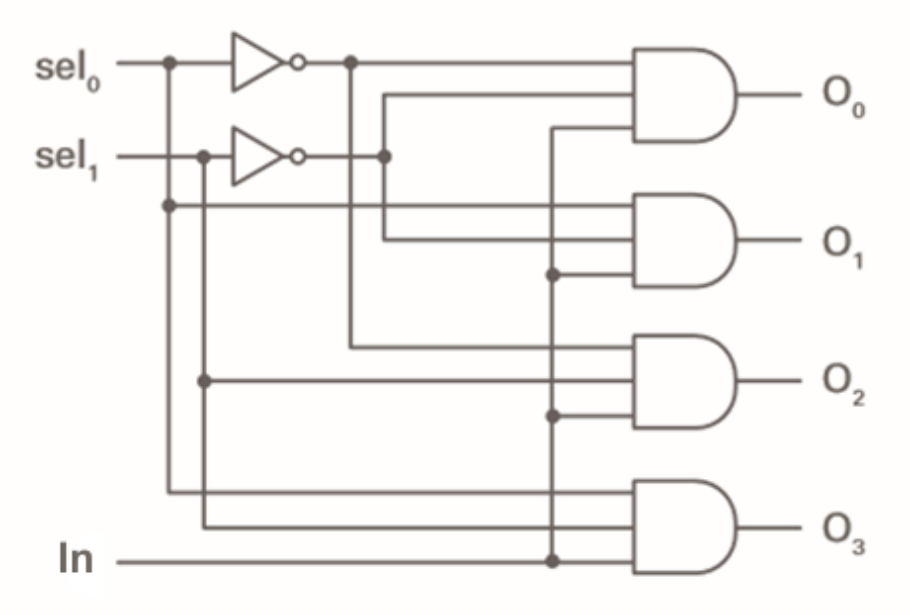


Figure 7-8 Image of 1-to-4 demultiplexer

- Multisim:	Όνομα αρχείου “**7_Demux_1to4.ms14**”.
	Προσθήκη στο zip file με όνομα ”Lab7_ονοματεπώνυμο_AM.zip”
ή	
- MultisimLive:	Όνομα αρχείου “**7_Demux_1to4.png**”.
Schematic image	Προσθήκη στο zip file με όνομα ”Lab7_ονοματεπώνυμο_AM.zip”

- The truth table for a 1-to-4 demultiplexer is below.

sel_1	sel_0	O_3	O_2	O_1	O_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0

Figure 7-9 Truth table

- Notice that the **sel inputs** indicate which output will be On.

- Confirm that your circuit follows this behavior, and record your observations and include them with your completed lab.

7-5 How would you implement a 1-to-8 demultiplexer? Define the pattern for adding more outputs?

a) I would have 8 selectors which would effectively create the numbers 0-7 which would map to the outputs 0-7.
b) I would have 3 selectors which would effectively create the numbers 0-7 which would map to the outputs 0-7.
c) I would have 3 inputs which would effectively create the numbers 0-7 and 1 selector.
d) I would have 8 inputs which would effectively create the numbers 0-7 and 1 selector.

7.4 Conclusion

7-6 How many outputs does a multiplexer have?

A. 1
B. 2
C. 3
D. 2^n

7-7 Why can the truth table of a 2-to-1 multiplexer be simplified depending on whether the selector is set to 0 or 1?

A. There is only one output
B. Some of the outputs of the original truth table are don't care conditions
C. The line that is selected to be inputted will be the only one affecting the output
D. None of the above

7-8 The 1-to-4 demultiplexer has how many selectors?

A. 4
B. 3
C. 2
D. None of the above

7-9 What is the difference between the logic circuit of a 2-to-4 decoder (with enable input) and a 1-to-4 demultiplexer?

A. They use a different combination of logic gates
B. They have a different number of outputs
C. They have different inputs
D. None

7.5 Exercise: HDL - Verilog, Mux 4-bit 2x1

Υλοποιήστε **ένα *πολυπλέκτη* 4-bit, 2 σε 1**, με εισόδους a [3:0], b [3:0], sel και έξοδο out [3:0].

1. Σχεδιάστε το σχηματικό στο Multisim που παρουσιάζει το κύκλωμα που παράγει τις συνδυαστικές συναρτήσεις των σημάτων: out[3], out[2], out[1], out[0].

- Multisim:	Όνομα αρχείου “**7_mux_4bit_2x1.ms14**”.
	Προσθήκη στο zip file με όνομα ”Lab7_ονοματεπώνυμο_AM.zip”
ή	
- MultisimLive:	Όνομα αρχείου “**7_mux_4bit_2x1.png**”.
Schematic image	Προσθήκη στο zip file με όνομα ”Lab7_ονοματεπώνυμο_AM.zip”

Σημείωση:
Μπορείτε να χρησιμοποιήσετε ως είσοδο, εναλλακτικά, τους εξής διακόπτες συνδεδεμένους όπως στο Figure 8-10.

Θα τους βρείτε στη βιβλιοθήκη: Basic / Switch / DSKPK_4

Θα χρειαστεί επίσης στο ένα άκρο τους να δώσετε τάση 5Vdc. Θα την βρείτε: Sources / POWER_SOURCES / VCC

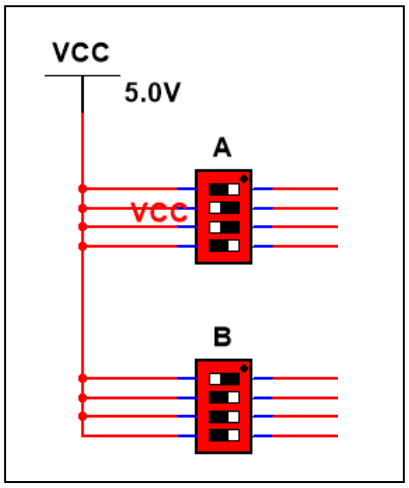


Figure 7-10

2. Υλοποιήστε σε γλώσσα Verilog τον παραπάνω πολυπλέκτη.

Η πρώτη γραμμή του module να είναι η εξής:

```
module    mux_4bit_2x1    (out, a, b, sel);
```

Ονομάστε το module: "mux_4bit_2x1"

- Verilog:	Όνομα αρχείου "**mux_4bit_2x1.v**". Προσθήκη στο zip file με όνομα "Lab7_ονοματεπώνυμο_AM.zip"

3. Έπειτα κατασκευάστε μία μονάδα δοκιμής (test bench) όπου θα δοκιμάζετε το προηγούμενο κύκλωμά ως εξής:

- Στα 10 nsec: a = 0011, b = 1100
- Στα 20 nsec: sel = 0
- Στα 30 nsec: sel = 1
- Στα 40 nsec: $finish

Χρησιμοποιήστε την εντολή "$monitor" μέσα σε μία initial ως εξής:

```
$monitor ("Time = %2t, A = %b, B = %b, sel = %b, out = %b", $time,
a, b, sel, out);
```

Ονομάστε το module: "t_mux_4bit_2x1"

- Verilog:	Όνομα αρχείου "**t_ mux_4bit_2x1.v**". Προσθήκη στο zip file με όνομα "Lab7_ονοματεπώνυμο_AM.zip"

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: Όνομα αρχείου “**t_ mux_4bit_2x1**”.
Προσθήκη στο zip file με όνομα ”Lab7_ονοματεπώνυμο_ΑΜ.zip”