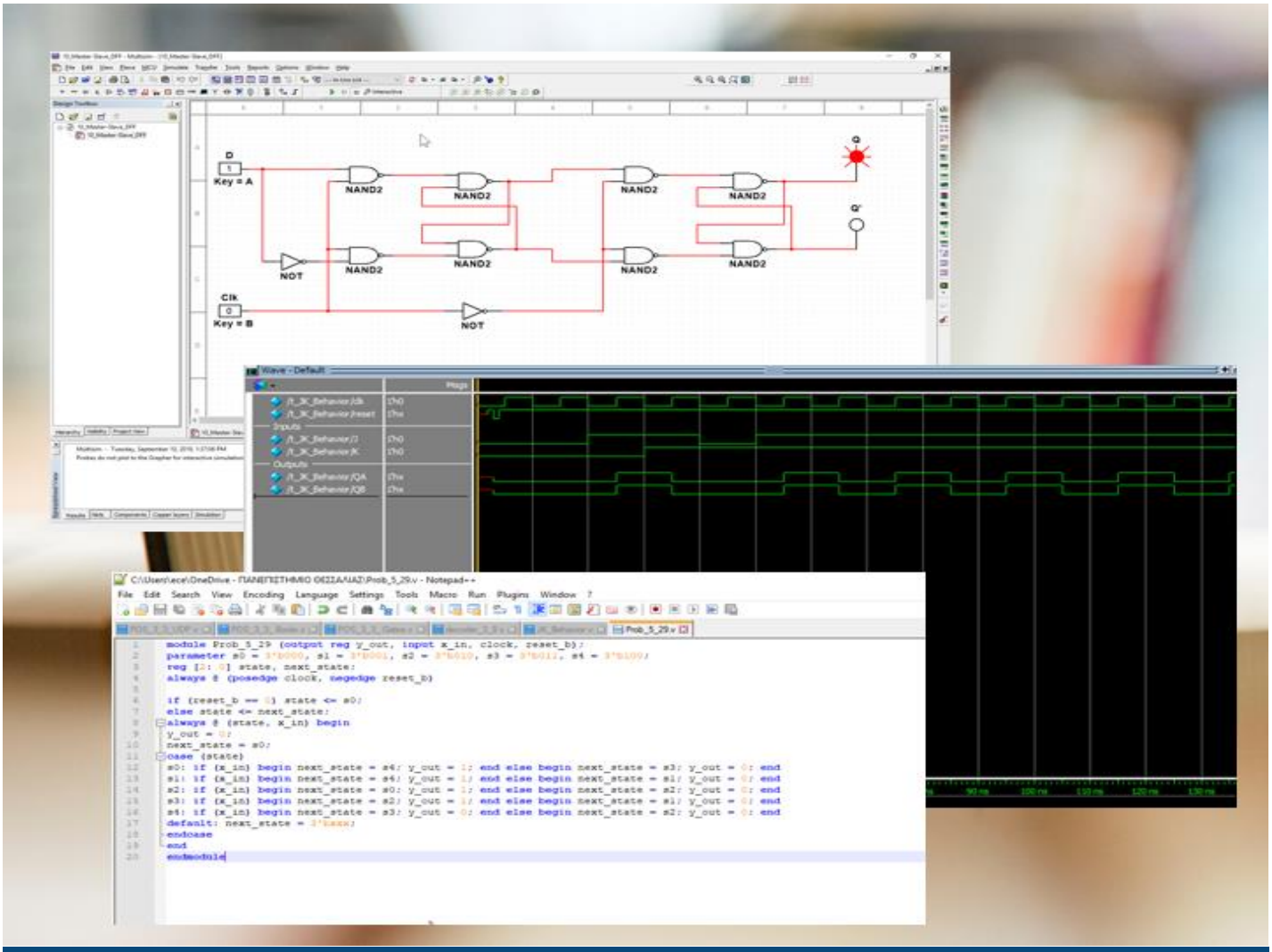
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ



ECE119 Ψηφιακή Σχεδίαση

Εργαστηριακές ασκήσεις, Multisim - Verilog

Lab 6: Encoders and Decoders

Καραμπερόπουλος Δημήτρης

Σεπτέμβριος 2026

Lab 6

Required Tools and Technology 1
Lab 6: Encoders and Decoders 2
Learning Objectives 2
Expected Deliverables 2
6.1 Theory and Background 3
6.2 Exercise: Seven Segment Display Diagram 7
6.3 Implement: Building a BCD to Seven Segment Display Decoder 7
6.4 Implement: Building a Circuit for a 10 to 4 Priority Encoder 10
6.5 Conclusion 12
6.6 Exercise: HDL - Verilog, Decoder 3x8 13

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 6: Encoders and Decoders

In Lab 5, we learned that gates arranged to perform a specific function, such as binary addition, can be represented with a chip. Other applications of this concept include encoders and decoders. Encoders convert information from multiple input lines into a compact binary code. Decoders interpret binary codes to activate specific outputs or control devices. In a previous lab, we learned what a seven-segment display is and in this lab we will explore how decoders apply to this electronic device.

Learning Objectives

In this lab, students will:

- Explain the operation of binary encoders and decoders.
- Distinguish between standard and priority encoders.
- Understand the role of enable signals and active-high/active-low logic.
- Explain how a BCD-to-seven-segment decoder controls a display.
- Build and simulate a BCD-to-seven-segment decoder circuit and verify its operation.

Expected Deliverables

In this lab you will collect the following deliverables:

- Diagram of a BCD to 7 Segment Display Decoder
- Truth tables
- Conclusion questions
- Verilog decoder implementation
- Testbench
- Simulation waveforms.

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

6.1 Theory and Background

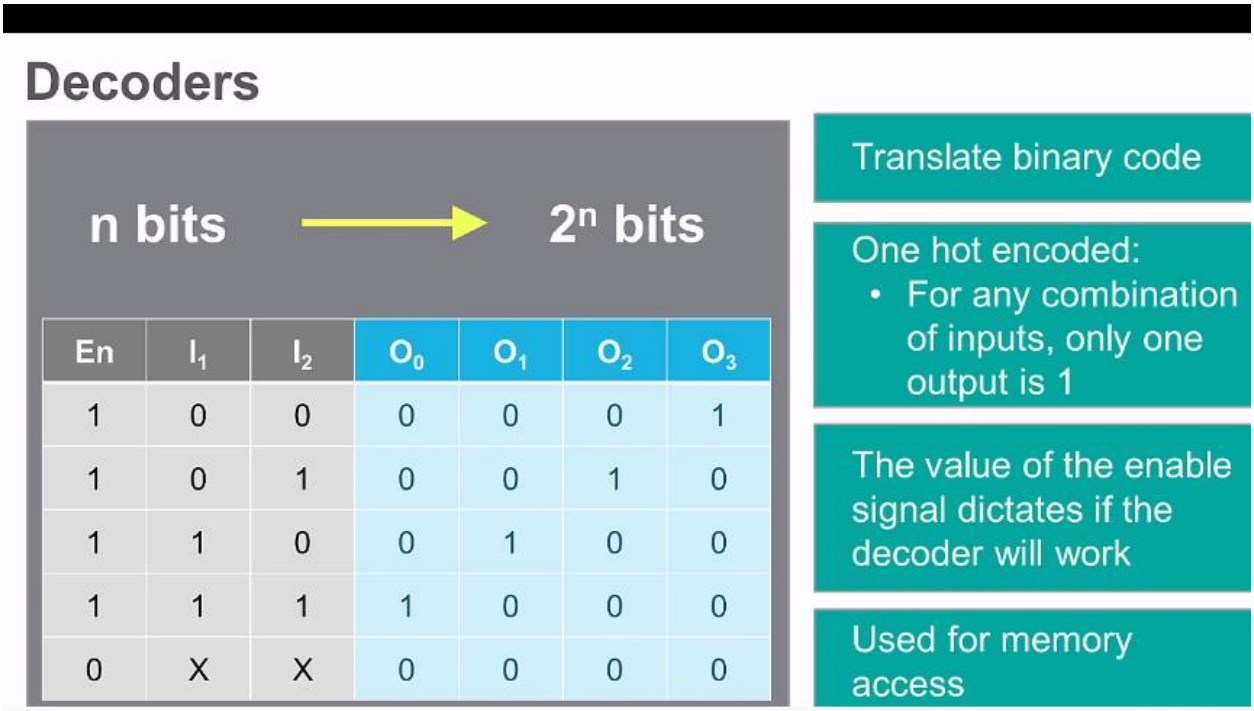


Figure 6-1 Video Screenshot. View the video here: https://youtu.be/RH2SeKV_DKg

Video Summary

- Decoders are devices that translate a binary code of n-bits of information into 2^n bits of information
- For any combination of signals there is only one output that can have a value of 1
- Encoders perform the opposite function of a decoder
- Encoders encode information from 2^n input lines into an n-bit output line

Decoders

A decoder is a combinational logic circuit that converts an n-bit binary input into a specific output pattern. In a standard binary n-to-2^n decoder, exactly one output is activated for each input combination when the circuit is enabled.

- Decoders take binary codes of n bits and have 2^n outputs.
- The outputs of an active-high binary decoder are **one-hot encoded** when the circuit is enabled, since **exactly one output is 1 for each input combination**.
- Decoders can include an *enable signal* for controlling the circuit operation.
- This enable signal can be active-low (meaning that the circuit will operate only when enable is 0) or active-high (the decoder is enabled when enable is 1).
- Decoders with enable inputs can be used for constructing larger decoders.
- One of the most important applications of decoders is memory access, where they are used for decoding the address of the rows in the memory blocks.

Let's take the example of a 2 to 4 decoder enabled when the *En* signal is 1:

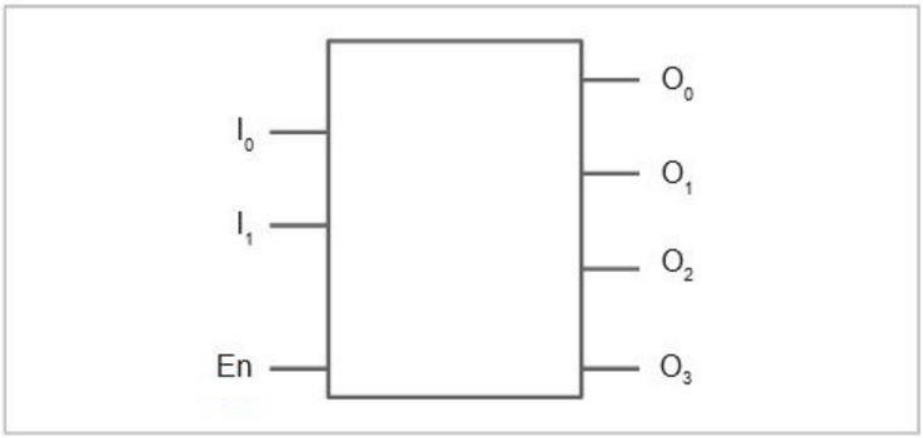


Figure 6-2 Decoder

From this, we can determine the following truth table and logic circuit:

En	I_1	I_0	O_3	O_2	O_1	O_0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0
0	x	x	0	0	0	0

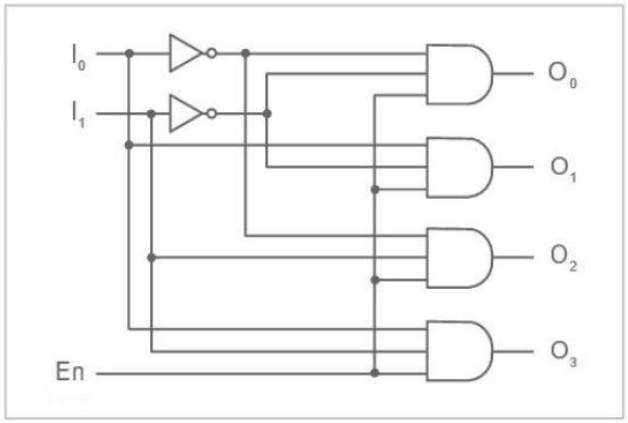
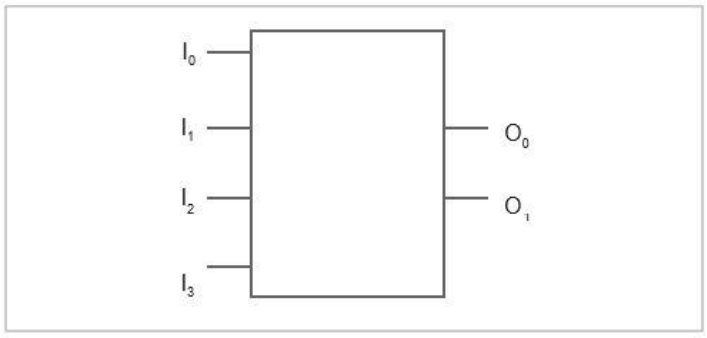


Figure 6-3 Truth table (left) and logic circuit (right)

Encoders

Encoders are logic circuits that perform the opposite function of a decoder. Binary encoders encode information from 2^n input lines, producing an n-bit code.

- In a standard binary encoder, **only one input is assumed to be active at a time** (one-hot input).
- Encoding is used for reducing the number of bits needed to represent information. Encoders are commonly used in applications such as data transmission and storage.
- The graphical symbol of the **4 to 2 binary encoder** is presented below. The cases in which more than one input is 1 are not shown in the truth table because they are treated as don't care conditions (as we guarantee that they will not appear in normal operation).



I_3	I_2	I_1	I_0	O_1	O_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Figure 6-4 Encoder (left) and truth table (right)

- It can be seen in the truth table that the output O_1 is 1 when either I_3 or I_2 is 1 and that the output O_0 is 1 when either I_3 or I_1 is 1.
- The input I_0 does not appear in the output equations, since its activation produces the binary code 00.
- The encoders presented so far are considered to have ***one-hot encoded*** **inputs**.

The corresponding logic circuit is presented below:

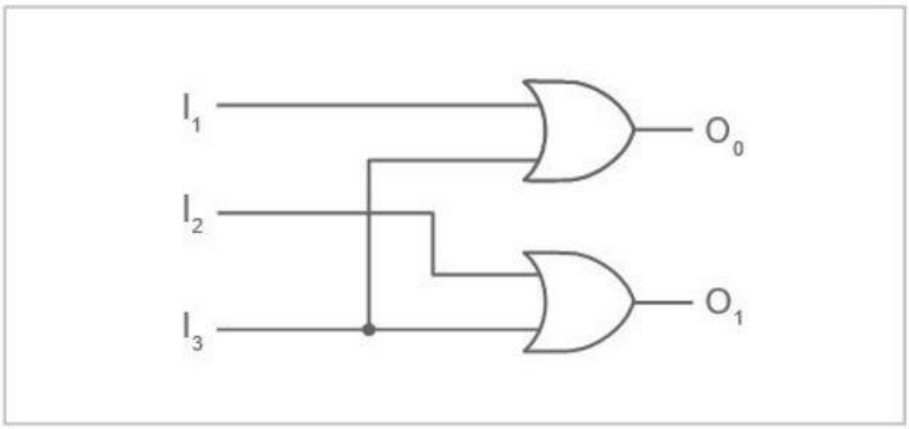


Figure 6-5 Logic circuit

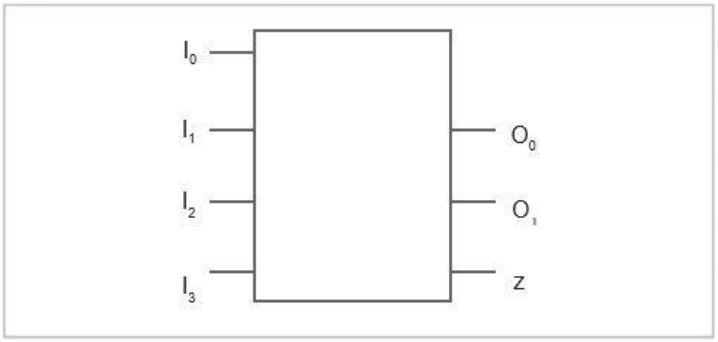
Another commonly used type of encoder is a ***priority encoder*****:**

- Priority encoders are able to prioritize inputs
- This is important because regular encoders can generate the wrong output when there is more than one input present at logic level 1.
- The priority encoder has an additional output, z (valid input indicator), which is 1 when at least one input is active and 0 when no input is active.

The graphical symbol of the priority encoder is presented below. The truth table describes the behavior of a 4-to-2 priority encoder. It can be seen that: When I_3=1, the encoder produces O_1O_0=11 and z=1, regardless of the values of the lower-priority inputs.

The priority is: $I_3 > I_2 > I_1 > I_0$

For example, if I_3 =1 and I_1 =1, the output is 11, because I_3 has higher priority.



I_3	I_2	I_1	I_0	O_1	O_0	z
0	0	0	0	x	x	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

Figure 6-6 Priority encoder (left) and truth table (right)

6.2 Exercise: Seven Segment Display Diagram

Note: Seven Segment Displays (SSD) explored in a previous lab use a Binary Coded Decimal (BCD) to a 7-Segment Display Decoder.

6-1 Explain briefly how a decoder works.

a) A standard binary decoder converts an n-bit binary input into 2^n output lines, activating one output for each input combination when enabled.
b) A decoder analyzes all its input bits to determine the corresponding decoded output. A decoder can generate the wrong output when there is more than one input equal to 1.

6-2 What is the difference between an encoder and a priority encoder?

a) An encoder analyzes all its input bits to determine the corresponding encoded output. A priority encoder can generate the wrong output when there is more than one input equal to 1. A regular encoder, however, can have multiple inputs equal to 1.
b) An encoder analyzes all its input bits to determine the corresponding encoded output. A regular encoder can generate the wrong output when there is more than one input equal to 1. A priority encoder, however, can have multiple inputs equal to 1, as it prioritizes certain inputs and can ignore others.

6.3 Implement: Building a BCD to Seven Segment Display Decoder

BCD to Seven Segment Display Decoder

Build the following circuit:

- From the **TTL** Group, select the **74LS** Family and then the **74LS47N.**
- From the **Basic** Group, select the **RPACK** Family and then the **7Line_Isolated** resistor.
- Right click on the resistor and view **properties.** Change the resistance to **220 Ω**
- From the **Indicators** Group, select **HEX_DISPLAY** and then **SEVEN_SEG_COM_A_GREEN.**

- From the **Sources** Group select **POWER_SOURCES** and then **VCC**. Place one near the bottom of the Decoder and one near the top of the SSD.
- Place four **INTERACTIVE_DIGITAL_CONSTANTS.**
    - Change the keys for toggle to math the ones shown in the figure below.

Write them as shown:

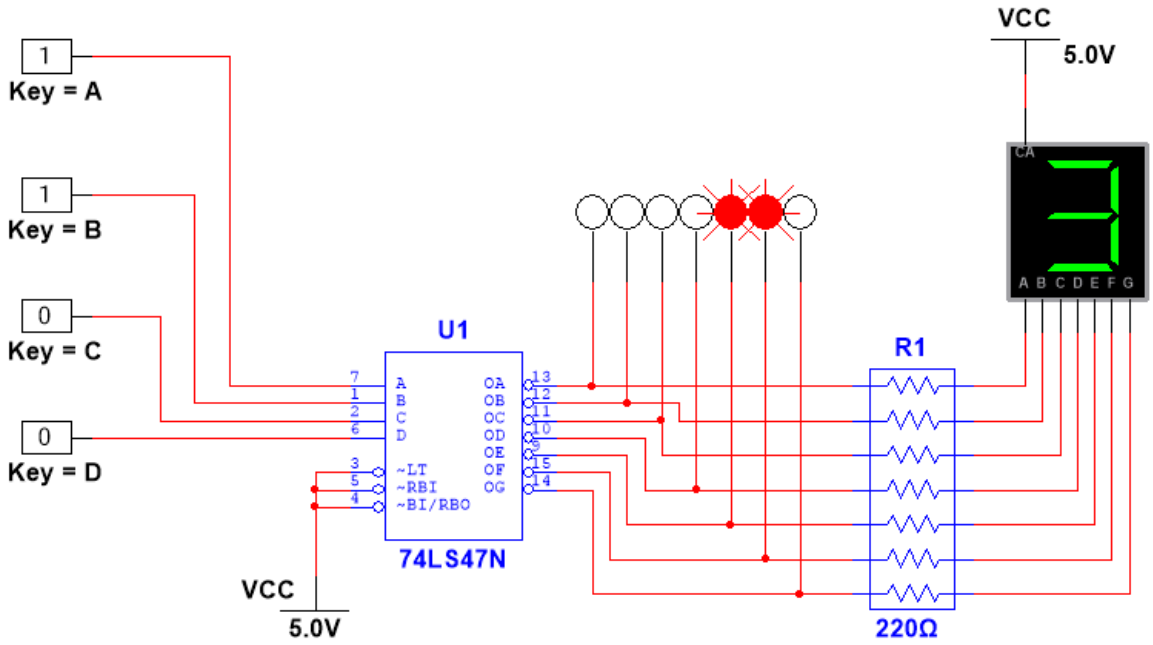


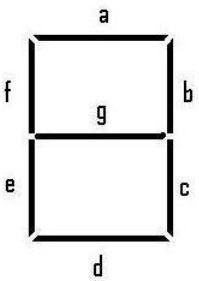
Figure 6-7 Circuit diagram

Brief explanations:

- BCD inputs (A, B, C, D): Represent a decimal digit (0–9) using 4 bits.
- Seven outputs (a–g): Control which segments of the display are illuminated.
- Active-low outputs: A LOW output activates the corresponding segment.
- Current-limiting resistors: These resistors limit the current flowing to the display's LEDs.
- Valid BCD inputs range from 0000 to 1001 (decimal 0–9). The combinations 1010–1111 are not valid BCD digits. Invalid BCD inputs do not necessarily produce undefined outputs. The actual behavior depends on the decoder.

- Multisim:	Όνομα αρχείου “**6_BCD_to_SSD.ms14**”. Προσθήκη στο zip file με όνομα ”Lab06_ονοματεπώνυμο_ΑΜ.zip”

Remember this:



Testing a BCD to Seven Segment Display Decoder

- **Run** the Simulation

6-3 Complete the truth table by testing all 16 input combinations (0000–1111) in Multisim. Record the logic levels at the outputs (a–g) of the 7447N decoder and the digit or pattern displayed on the seven-segment display.

Important: The 7447N has active-low outputs. A LOW output activates the corresponding display segment.

Logic 0 = Segment ON
Logic 1 = Segment OFF

For invalid BCD inputs, record the observed output values and write “invalid” in the Displayed Digit / BCD Status column.

D	C	B	A	a	b	c	d	e	f	g	Displayed Digit / BCD Status
0	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	0	0	0								
1	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								

6.4 Implement: Building a Circuit for a 10 to 4 Priority Encoder

The **74147N** (Family 74STD) is a 10-to-4 priority encoder with **active-low inputs and outputs**. An input is active when its value is 0. When multiple inputs are active, the input with the highest decimal value takes priority.

Circuit

- Create the following circuit:

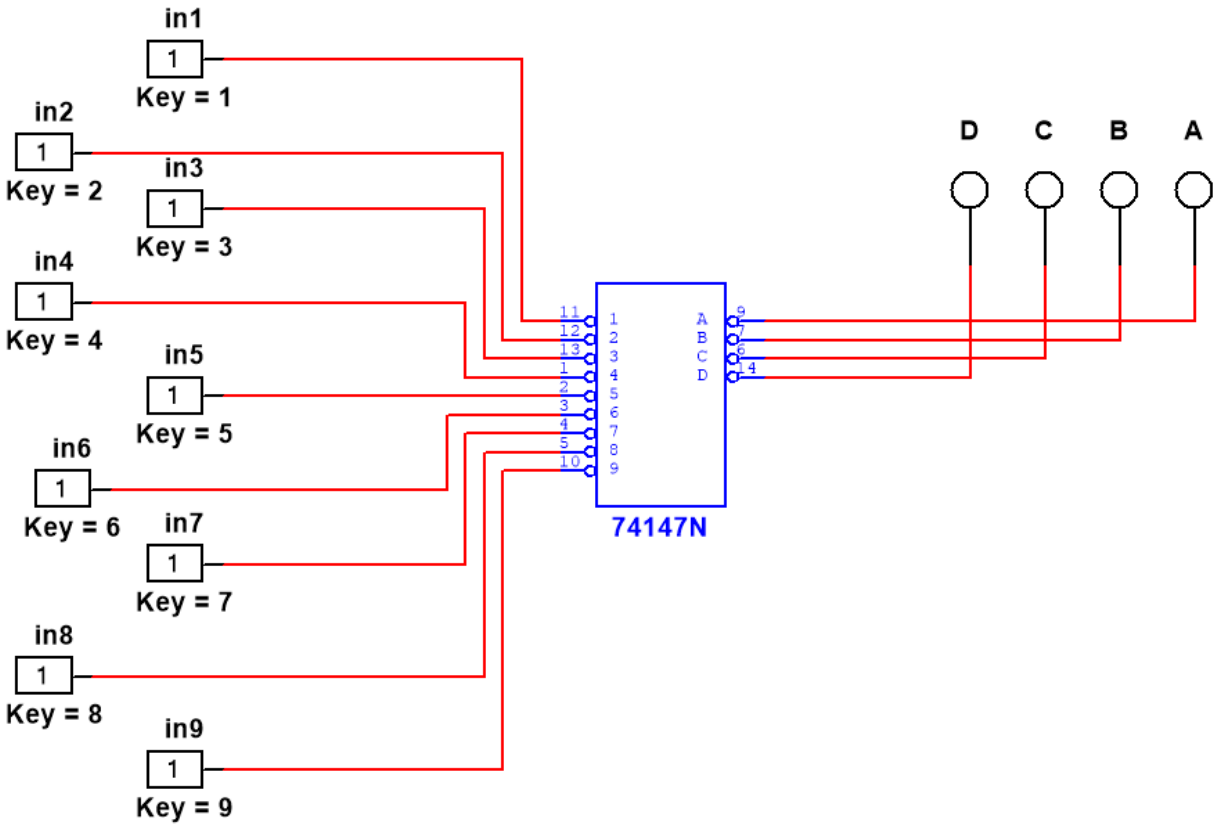


Figure 6-8 Circuit diagram

- Multisim:	Όνομα αρχείου “**6_Encoder.ms14**”. Προσθήκη στο zip file με όνομα ”Lab06_ονοματεπώνυμο_ΑΜ.zip”

Testing 10 to 4 Priority Encoder

- **Run** the circuit.

6-4 Complete the following truth table for the 10 to 4 priority encoder.

in9	in8	in7	in6	in5	in4	in3	in2	in1	D	C	B	A
1	1	1	1	1	1	1	1	1				
1	1	1	1	1	1	1	1	0				
1	1	1	1	1	1	1	0	X				
1	1	1	1	1	1	0	X	X				
1	1	1	1	1	0	X	X	X				
1	1	1	1	0	X	X	X	X				
1	1	1	0	X	X	X	X	X				
1	1	0	X	X	X	X	X	X				
1	0	X	X	X	X	X	X	X				
0	X	X	X	X	X	X	X	X				

Note: X indicates a don't-care input. Its value may be either 0 or 1 without affecting the encoded output, provided that the higher-priority inputs remain as specified.

6-5 What happens when two inputs of the 74147N priority encoder are LOW?

a) When one input is low, the lights that are on display the corresponding binary number. When two inputs are low, lights display the binary number corresponding to the highest of the two input values.

b) When one input is low, the lights that are off display the corresponding binary number. When two inputs are low, lights display the binary number corresponding to the highest of the two input values.

6-6 Can you determine how the encoder defines priorities?

a) The larger input number takes priority.

b) The smaller input number takes priority.

Stop the circuit when you're done.

6.5 Conclusion

6-7 Why do encoders and decoders have different ratios of inputs and outputs?

a) Encoders translate a number from n bits to 2^n bits whereas decoders translate a number from 2^n bits to n bits. This is because the goal of an encoder is to increase the number of bits whereas the goal of a decoder is to translate a larger, encoded number back into its short form.

b) Encoders translate a number from 2^n bits to n bits whereas decoders translate a number from n bits to 2^n bits. This is because the goal of an encoder is to reduce the number of bits whereas the goal of a decoder is to translate a shortened, encoded number back into its long form.

6-8 What is the purpose of an enable input in a decoder?

A. It determines the order in which inputs are processed.
B. It increases the number of output lines.
C. It controls whether the decoder is enabled or disabled.
D. All of the above

6-9 Which logic gates are needed to create a 2 to 4 decoder?

A. 2 NOT gates and 4 AND gates
B. 2 NOT gates and 4 OR gates
C. 2 AND gates and 4 NOT gates
D. 2 AND gates and 4 OR gates

6-10 In a standard binary encoder, which condition is assumed for correct operation?

A. Only one output can be HIGH at a time
B. Only one input is HIGH at a time (one-hot input)
C. All inputs must be HIGH simultaneously
D. The encoder must have an enable input

6.6 Exercise: HDL - Verilog, Decoder 3x8

A. Υλοποιήστε σε γλώσσα Verilog έναν Αποκωδικοποιητή, 3 σε 8, με εισόδους in[2:0] και enable (active-high) και έξοδο out[7:0]. Οι έξοδοι είναι επίσης active-high, δηλαδή όταν enable=0, πρέπει όλες να είναι 0.

Ονομάστε το module: "decoder_3x8"

- Verilog:	Όνομα αρχείου "**decoder_3x8.v**". Προσθήκη στο zip file με όνομα "Lab06_ονοματεπώνυμο_AM.zip"

B. Έπειτα κατασκευάστε μία μονάδα δοκιμής (test bench) όπου θα δοκιμάζετε το προηγούμενο κύκλωμά σας δίνοντας όλες τις δυνατές εισόδους ως εξής:

- Στα 10 nsec: enable = 0 και in = 000
- Στα 20 nsec: in = 001
- (κάθε 10nsec το in να αυξάνει κατά 1 μέχρι να γίνει in =111).
- Στα 100 nsec: enable = 1 και in = 000
- Στα 110 nsec: in = 001
- (κάθε 10nsec το in να αυξάνει κατά 1 μέχρι να γίνει in =111).
- Στα 200 nsec: Τέλος προσομοίωσης

Μπορείτε να χρησιμοποιήσετε την εντολή "repeat".
π.χ. **repeat (10) #5 xxx**;
Η εντολή "xxx" θα εκτελεστεί 10 φορές και μάλιστα κάθε φορά με αναμονή 5nsec

Να χρησιμοποιήσετε την εντολή **$monitor** για να ελέγξετε τις εισόδους - εξόδους. Μπορείτε να χρησιμοποιήσετε και τη διαδικασία συστήματος **$time** η οποία επιστρέφει τον χρόνο της προσομοίωσης.

π.χ. **$monitor ("Time = %3d x = %b y = %b", $time, x, y);**
Κάθε φορά που αλλάζουν οι x, y στο terminal θα έχουμε και την χρονική στιγμή της εκτέλεσης της εντολής.



Να χρησιμοποιήσετε τις εντολές **$dumpfile** και **$dumpvars** ώστε να παραχθεί το αρχείο “vcd” με τις κυματομορφές των σημάτων. Έπειτα να ανοίξετε αυτό το αρχείο με το VSCode ή το gtkwave για να δείτε τις κυματομορφές και να ελέγξετε οπτικά τον τρόπο που μεταβάλλεται η έξοδος στους συνδυασμούς της εισόδου.

Ονομάστε το module: “t_decoder_3x8”.



- Verilog:	Όνομα αρχείου “**t_decoder_3x8.v**”. Προσθήκη στο zip file με όνομα ”Lab06_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_decoder_3x8**”. Προσθήκη στο zip file με όνομα ”Lab06_ονοματεπώνυμο_AM.zip”