AROR UNIVERSITY
OF ART, ARCHITECTURE,
DESIGN & HERITAGE,
SUKKUR, SINDH

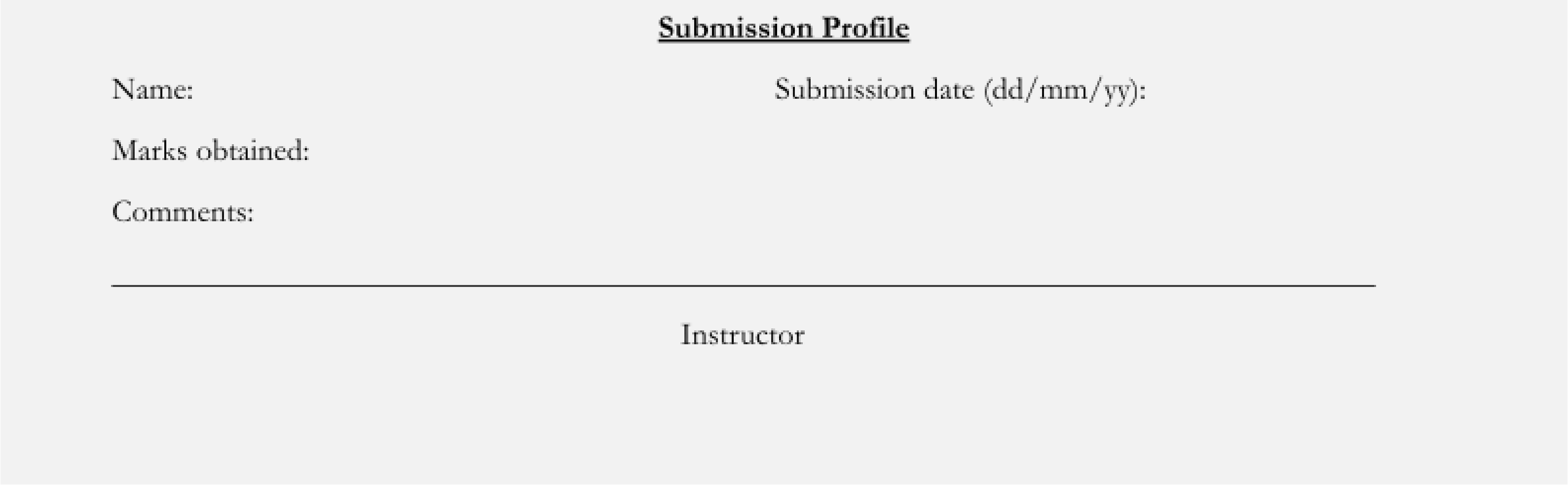
Faculty of Artificial Intelligence & Multimedia Gamming

BS – Artificial Intelligence (Section A)

Digital Logic Design Lab

Lab # 07: Encoder and Decoder

Mr. Abdul Ghafoor



Submission Profile

Name:

Submission date (dd/mm/yy):

Marks obtained:

Comments:

Instructor

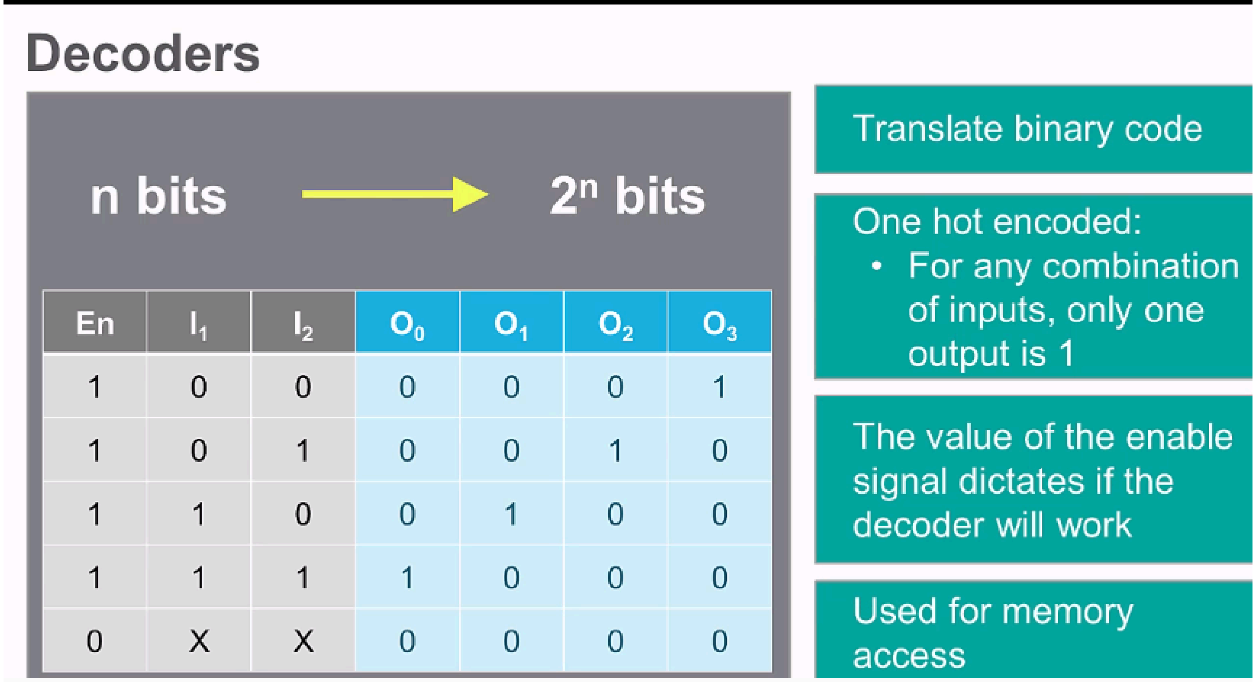
Lab Learning Objectives:

Upon successful completion of this experiment, the student will be able:

- Explain how decoders works, specifically in an SSD
- Create a circuit with a BCD to Seven Segment Display Decoder and verify its truth table

Lab Hardware and Software Required:

Platform: NI ELVIS III	✔ View User Manual: http://www.ni.com/en-us/support/model.ni-elvis-iii.html ✔ View Tutorials: https://www.youtube.com/playlist?list=PLvcPIuVaUMIWm8ziaSxv0gwtshBA2dh_M
Software: NI Multisim 14.0.1 Education Version or newer	✔ Install Multisim: http://www.ni.com/gate/gb/GB_ACADEMICEVALMULTISIM/US ✔ View Help: http://www.ni.com/multisim/techn



Decoders

n bits → 2^n bits

En	I_1	I_2	O_0	O_1	O_2	O_3
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

Translate binary code

One hot encoded:
- For any combination of inputs, only one output is 1

The value of the enable signal dictates if the decoder will work

Used for memory access

Background Theory:

Figure 1-1 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

The process of translating ambiguous information into something understood by a device receiving the data is called *decoding***.**
Therefore, the resulting device is known as a *decoder***.**

- Decoders take binary codes of n bits and generate 2^n outputs.
- The outputs of a binary decoder are said to be *one-hot encoded* because for any combination of the input signals there is only one output having the value 1.
- 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:

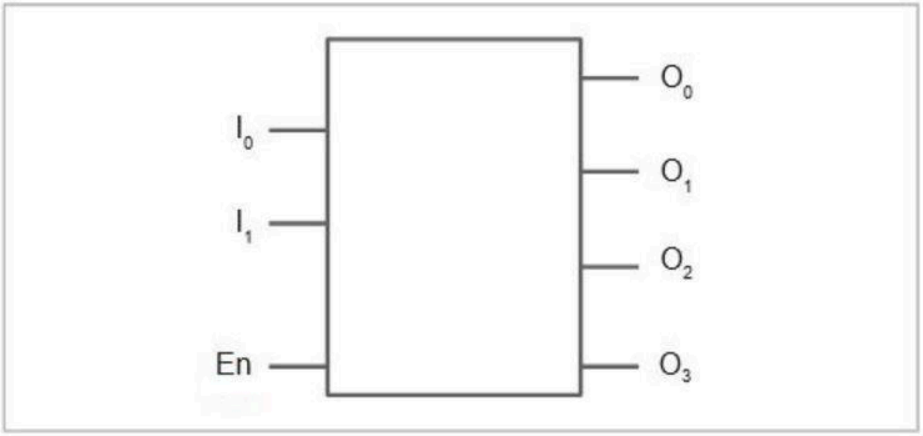
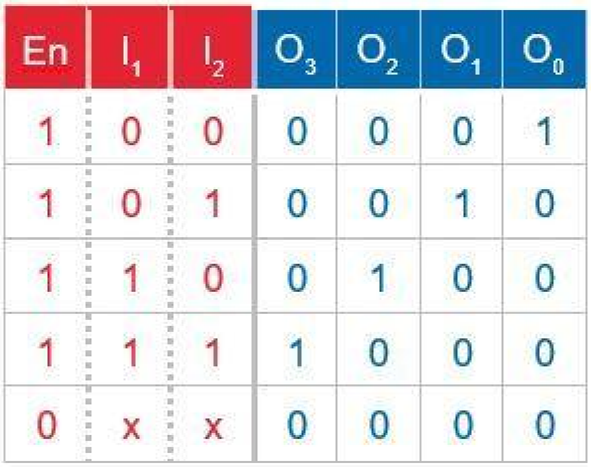


Figure 1-2 Decoder

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



En	I_1	I_2	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

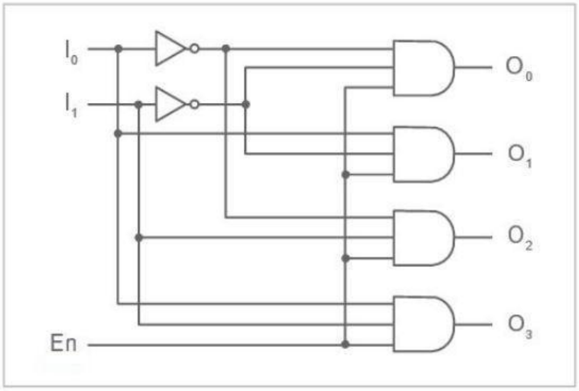
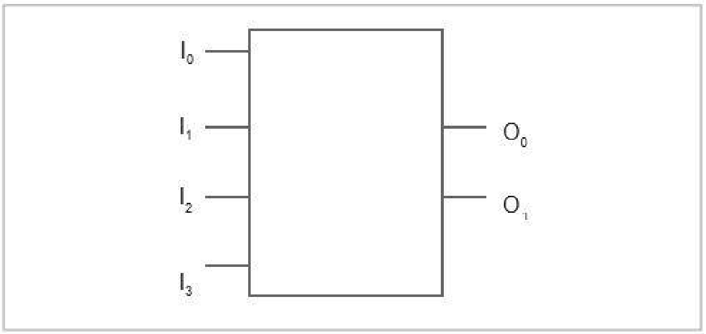


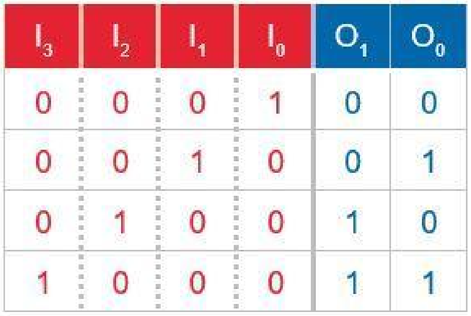
Figure 1-3 Truth table (left) and logic circuit (right)

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.

- At any given time, only one of the 2^n inputs can be 1.
- Encoding is used for reducing the number of bits needed to represent information. They are often used in application such as data transmission and data storing.
- 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.





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 1-4 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.
- It can also be seen that the input I_0 can be ignored
- The encoders presented so far are considered to have *one-hot encoded* inputs.

The corresponding logic circuit is presented below:

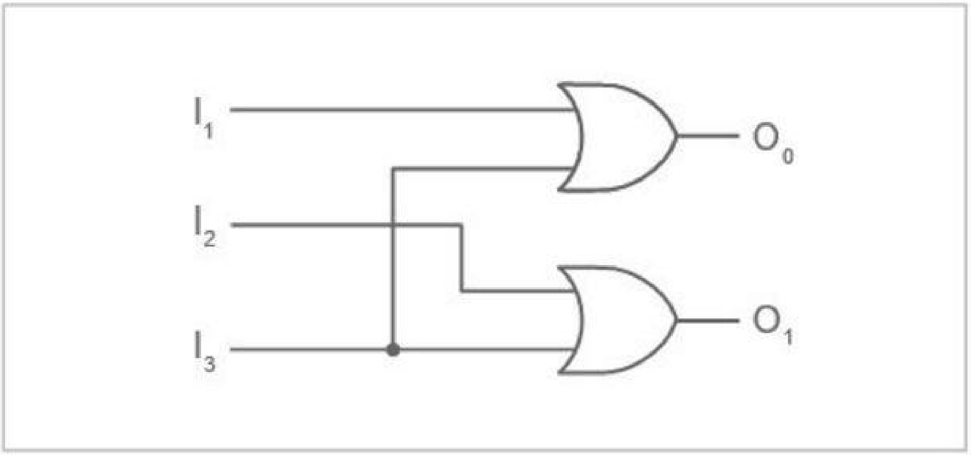
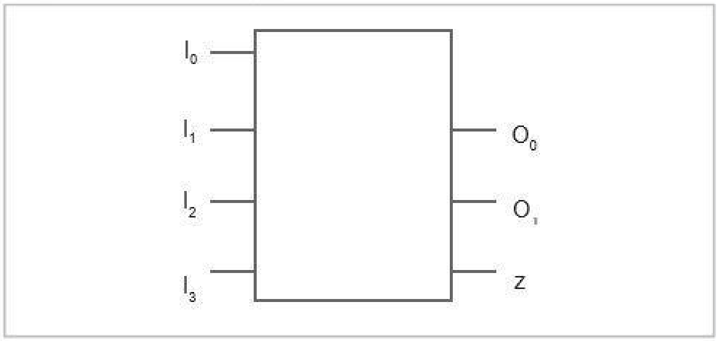


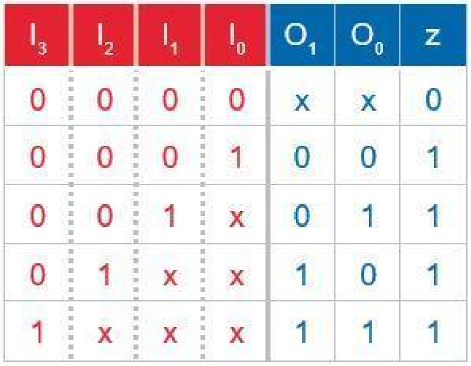
Figure 1-5 Logic circuit

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.
- This type of encoder has an additional output, z, which indicates the case in which none of the inputs is 1.

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 on the last line of the truth table that if the input I_3 is 1, the outputs are all 1 and the values on the other inputs of the decode do not matter and are denoted by 'x'.





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 1-6 Priority encoder (left) and truth table (right)

Lab Activities:

1.1 Implement: Binary to Octal Decoder

A	B	C	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Image of circuit

Lab Activities:

1.2 Implement: Octal to Binary Encoder

D7	D6	D5	D4	D3	D2	D1	D0	A	B	C
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Image of circuit

Lab Activities:

1.2 Implement: Building a BCD to Seven Segment Display Decoder

BCD to Seven Segment Display Decoder

Build the following circuit:

- Click the **Misc Digital** button and from the **TTL** Group, select the **7447N Decoder.**
- Click the **Misc Digital** button and from the **Basic** Group, select the **RPACK** Family and then the **7Line_Isolated** resistor.
- Right click on the resistor and view **priorities.** Change the resistance to **220 Ω**
- Click the **Misc Digital** button from the **Indicators** Group, select **HEX_DISPLAY** and then **SEVEN_SEG_COM_A_GREEN.**
- Click the **Misc Digital** button from the **Sources** Group select **POWER_RESOURCES** 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 match the ones shown in the figure below.

Write them as shown:

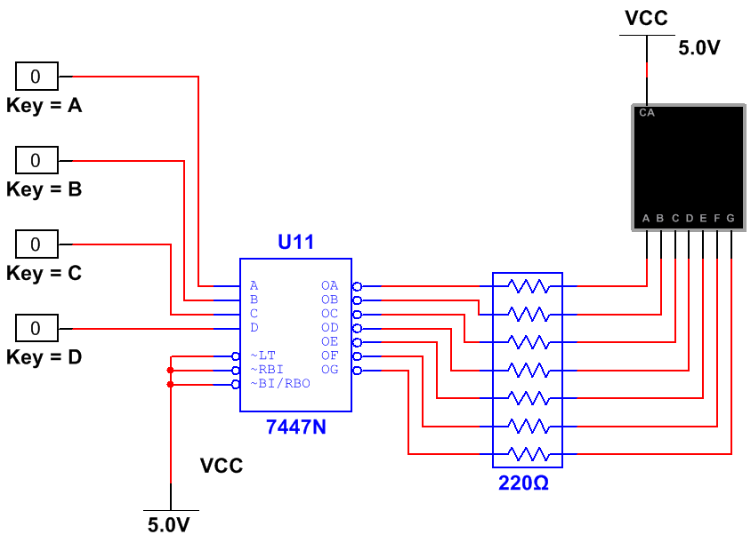


Figure 1-7 Circuit diagram

Testing a BCD to Seven Segment Display Decoder

- Run the Simulation

1-3 Vary the inputs to complete the following truth table of an SSD

A	B	C	D	a	b	c	d	e	f	g	Numeric Output
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	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	X	X	X	X	X	X	X	none
1	0	1	1	X	X	X	X	X	X	X	none
1	1	0	0	X	X	X	X	X	X	X	none
1	1	0	1	X	X	X	X	X	X	X	none
1	1	1	0	X	X	X	X	X	X	X	none
1	1	1	1	X	X	X	X	X	X	X	none