## Laboratory Report Cover Sheet

**18ECE206J ADVANCED DIGITAL SYSTEMS DESIGN**

**Fourth Semester, 2021-22 (Even semester)**

SRM Institute of Science and Technology College of Engineering and Technology

Department of Electronics and Communication Engineering

#### Name :

**Register No. :**

**Day/ Session :**

**Venue :**

**Title of Experiment :**

**Date of Conduction :**

**Date of Submission :**

|  |  |  |
| --- | --- | --- |
| **Particulars** | **Max. Marks** | **Marks**  **Obtained** |
| Pre lab and Post lab | 10 |  |
| Lab Performance | 20 |  |
| Simulation and results | 10 |  |
| Total | 40 |  |

**REPORT VERIFICATION**

**Staff Name : Signature :**

**9. Design of Encoder andDecoder**

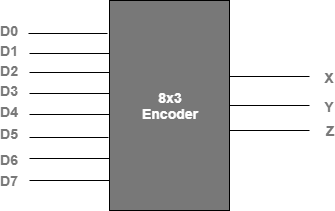
**Aim:** To design and implement Encoder and Decoder in VHDL.

**Software Required**: Xilinx ise& Model Sim

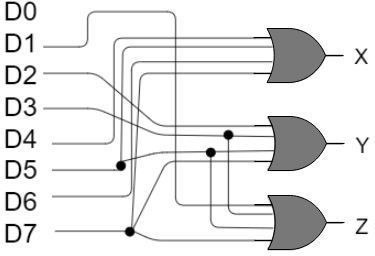
#### Theory:

An **Encoder** is a combinational circuit that performs the reverse operation of Decoder. It has maximum of 2n input lines and ‘n’ output lines. It will produce a binary code equivalent to the input, which is active High. Therefore, the encoder encodes 2n input lines with ‘n’ bits. It is optional to represent the enable signal in encoders

**Decoder** is a combinational circuit that as ‘n’ input lines and maximum of 2n output lines. One of these outputs will be active High based on the combination of inputs present, when the decoder is enabled. That means decoder detects a particular code. The outputs of the decoder are nothing but the **min terms** of ‘n’ input variables lines, when it is enabled.



**Fig. 9.1 Block diagram of Encoder**

****

**Fig. 9.2 Logic diagram of Encoder**

**Expression**

**X = D4 + D5 + D6 + D7**

**Y = D2 +D3 + D6 + D7**

**Z = D1 + D3 + D5 + D7**

**VHDL Code for a 8 x 3 Encoder**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

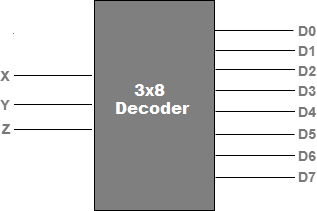
entity enc is

port(i0,i1,i2,i3,i4,i5,i6,i7:in bit; o0,o1,o2: out bit); end enc;

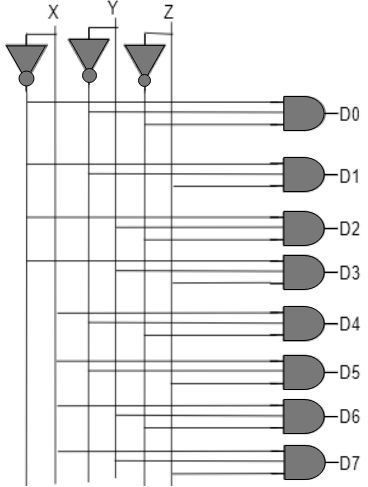
architecture dataflow of enc is begin

o0<=i4 or i5 or i6 or i7; o1<=i2 or i3 or i6 or i7; o2<=i1 or i3 or i5 or i7;

end dataflow;



**Fig. 9.3 Block diagram of Decoder**



**Fig. 9.4 Logic diagram of Decoder**

Expression D0 = X’ Y’ Z’ D1 = X’ Y’ Z D2 = X’ Y Z’ D3 = X’ Y Z D4 = X Y’ Z’ D5 = X Y’ Z D6 = X YZ’

D7 = X Y Z

#### VHDL Code for a 3 x 8 Decoder

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity dec is

port(i0,i1,i2:in bit; o0,o1,o2,o3,o4,o5,o6,o7: out bit); end dec;

architecture dataflow of dec is begin

o0<=(not i0) and (not i1) and (not i2); o1<=(not i0) and (not i1) and i2; o2<=(not i0) and i1 and (not i2); o3<=(not i0) and i1 and i2;

o4<=i0 and (not i1) and (not i2); o5<=i0 and (not i1) and i2; o6<=i0 and i1 and (not i2); o7<=i0 and i1 and i2;

end dataflow;

**Pre-lab questions:**

1. What is the difference between simple encoder and priority encoder?
2. What is the purpose of encoder and decoder?
3. Is encoder a multiplexer? Explain.
4. List the applications of Encoder and Decoder.

**Post-lab questions:**

1. Write a VHDL code to implement 4:1 Multiplexer using a 2:4 decoder.
2. Write a VHDL code to implement Full subtractor using 3:8 decoder.
3. Write VHDL code to implement 8: 3 encoder using case statement.
4. Write VHDL code to implement 3:8 Decoder using structural modeling.

**Result:**