|  |
| --- |
| **DIGITAL SYSTEM DESIGN LABORATORY** |
| **LAB 1\_Extra** |

**IMPLEMENTATION OF COMBINATION LOGIC CIRCUIT (2) USING VERILOG IN FPGA KIT**

### I. LAB OBJECTIVES

### This Lab experiments are intended to implement Combination Logic Circuits (2) in Verilog. Students are require to write test bench to simulate the given example code and Top level module to implement these codes in DE2-115 FPGA Kit.

### II. LAB EXPERIMENT EXERCISES

**AIM: WRITE VERILOG HDL CODES TO SIMULATE AND IMPLEMENT THE FOLLOWING DIGITAL SEQUENTIAL LOGIC CIRCUITS:**

1. Multiplexer 8 to 1
2. 1:8 Demultiplexer
3. 4-to-2 bit Encoder
4. 2-to-4 Binary Decoders
5. 8:3 Encoder
6. 3:8 Decoder
7. 4 Bit Comparator
8. 8 Bit Comparator
9. N bit comparator

For each circuit, Do the following steps:

Step 1 : Draw the Schematic of this circuit

### (Show Schematic in Lab report)

Step 2 : Write the truth Table for this circuit

### (Show The Truth Table in Lab report)

Step 3 : Write the Verilog Module to implement this circuit ( using structural, data flow, and behavior modeling)

### (Show Verilog codes of this module in Lab report)

Step 4 : Write the testbench to simulate the Verilog modules of this circuit

### (Show simulation results in Lab report)

Step 5 : Write the Top-level Verilog Code to implement the Verilog modules of this circuit in DE2-FPGA Kit

### (Show implementation results in Lab report)

**IV. LAB REPORT GUIDELINES**

Students write up a report on the Verilog HDL implementation experiment projects created in this lab. The lab report should include Circuit Schematics, Verilog Module Codes, Verilog test bench codes, Top level module to implement the required circuit in FPGA KIT and evidences of data output evidences to validate the experiments (The Captured Screens, Photo of FPGA Kit implementation results).

**1)Multiplexer 8 to 1**

module Lab1\_extra\_1(SW,LEDR,LEDG);

input [17:0] SW;

output [17:0] LEDR;

output [0:7] LEDG;

assign LEDR=SW;

mux81str DUT(.i0(SW[0]),.i1(SW[1]),.i2(SW[2]),.i3(SW[3]),.i4(SW[4]),.i5(SW[5]),.i6(SW[6]),.i7(SW[7]),.s0(SW[8]),.s1(SW[9]),.s2(SW[10]),.y(LEDG[0]));

endmodule

module mux81str(i0,i1,i2,i3,i4,i5,i6,i7,s0,s1,s2,y);

input i0,i1,i2,i3,i4,i5,i6,i7,s0,s1,s2;

output y;

wire a,b,c,d,e,f,g,h,n\_s0,n\_s1,n\_s2;

not N0(n\_s0,s0);

not N1(n\_s1,s1);

not N2(n\_s2,s2);

and G0(a,i0, n\_s2, n\_s1, n\_s0);

and G1(b,i1, n\_s2, n\_s1, s0);

and G2(c,i2, n\_s2, s1, n\_s0);

and G3(d,i3, n\_s2, s1, s0);

and G4(e,i4, s2, n\_s1, n\_s0);

and G5(f,i5, s2, n\_s1, s0);

and G6(g,i6, s2, s1, n\_s0);

and G7(h,i7, s2, s1, s0);

or G8(y,a,b,c,d,e,f,g,h);

A screenshot of a computer

Description automatically generatedendmodule