ECE3300L Lab 2

4x16 Decoder

Group X
Czyrone Agbayani (BID: 014766336)
Caleb Jala-Guinto (BID: 015446587)

Objective:

We will design, simulate and implement a 4-to-16 decoder with an enable input (SW4). When enabled, exactly one of the 16 outputs goes HIGH based on the 4-bit input. On the other hand, if disabled, all outputs are LOW.

Gate-Level Implementation:

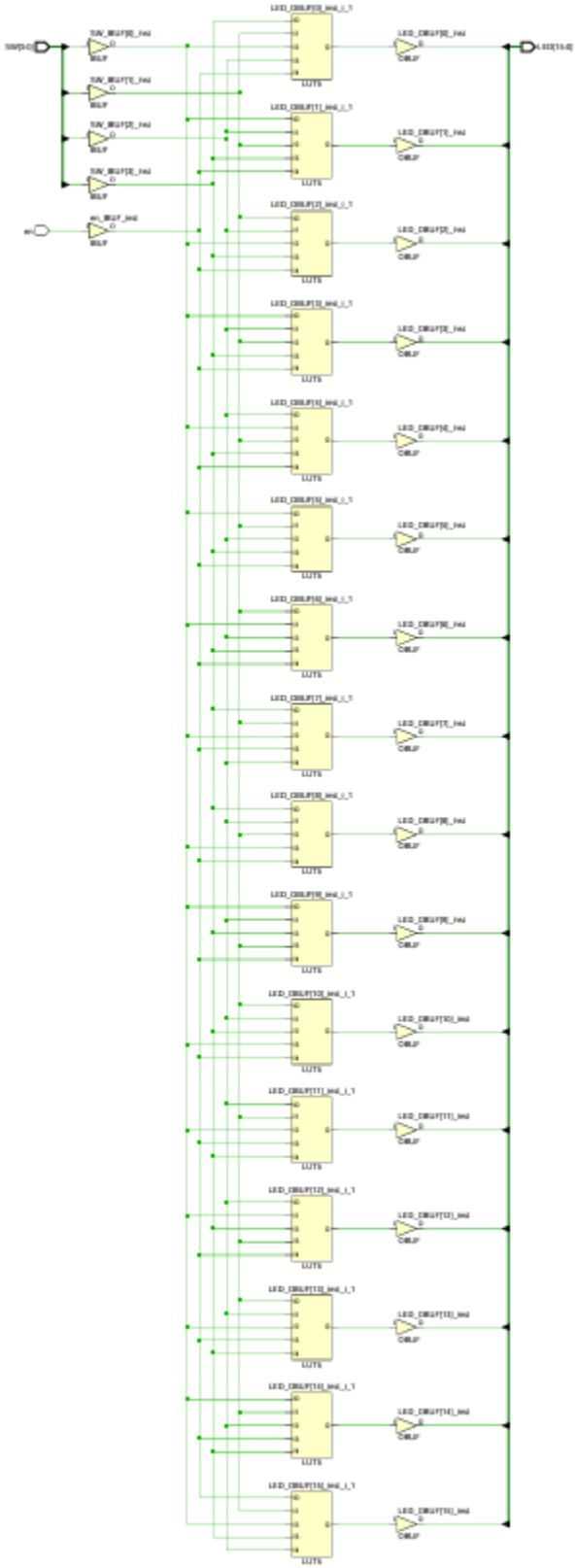
3-to-8 Decoder

```
23  module decoder3to8 (
24      input [2:0] SW,
25      input en,
26      output [7:0] LED
27      );
28      assign LED[0] = en & ~SW[2] & ~SW[1] & ~SW[0]; //3x8 truth table
29      assign LED[1] = en & ~SW[2] & ~SW[1] & SW[0];
30      assign LED[2] = en & ~SW[2] & SW[1] & ~SW[0];
31      assign LED[3] = en & ~SW[2] & SW[1] & SW[0];
32      assign LED[4] = en & SW[2] & ~SW[1] & ~SW[0];
33      assign LED[5] = en & SW[2] & ~SW[1] & SW[0];
34      assign LED[6] = en & SW[2] & SW[1] & ~SW[0];
35      assign LED[7] = en & SW[2] & SW[1] & SW[0];
36
37  endmodule
```

4-to-16 Decoder using two 3-to-8 Decoders

```
23  module decoder4to16(
24      input [3:0] SW,
25      input en,
26      output [15:0] LED
27      );
28      wire [7:0] low_out;
29      wire [7:0] high_out;
30
31      wire en_low = en & ~SW[3];
32      wire en_high = en & SW[3];
33
34      decoder3to8 lower_decoder (
35      .SW(SW[2:0]),
36      .en(en_low),
37      .LED(low_out)
38      );
39
40      decoder3to8 upper_decoder (
41      .SW(SW[2:0]),
42      .en(en_high),
43      .LED(high_out)
44      );
45
46      assign LED[7:0] = low_out;
47      assign LED[15:8] = high_out;
48
49  endmodule
```

Schematic:



Behavioral Implementation:

4-to-16 Decoder Behave:

```
module decoder4to16_behave(
    input [3:0] SW,
    input en,
    output reg [15:0] LED
    );

    always @(*) begin
    LED = 16'b0;
    if (en) begin
        case (SW)
        4'b0000: LED = 16'b0000_0000_0000_0001;
        4'b0001: LED = 16'b0000_0000_0000_0010;
        4'b0010: LED = 16'b0000_0000_0000_0100;
        4'b0011: LED = 16'b0000_0000_0000_1000;
        4'b0100: LED = 16'b0000_0000_0001_0000;
        4'b0101: LED = 16'b0000_0000_0010_0000;
        4'b0110: LED = 16'b0000_0000_0100_0000;
        4'b0111: LED = 16'b0000_0000_1000_0000;
        4'b1000: LED = 16'b0000_0001_0000_0000;
        4'b1001: LED = 16'b0000_0010_0000_0000;
        4'b1010: LED = 16'b0000_0100_0000_0000;
        4'b1011: LED = 16'b0000_1000_0000_0000;
        4'b1100: LED = 16'b0001_0000_0000_0000;
        4'b1101: LED = 16'b0010_0000_0000_0000;
        4'b1110: LED = 16'b0100_0000_0000_0000;
        4'b1111: LED = 16'b1000_0000_0000_0000;
        endcase
        end
        end

endmodule
```

Constraint file:

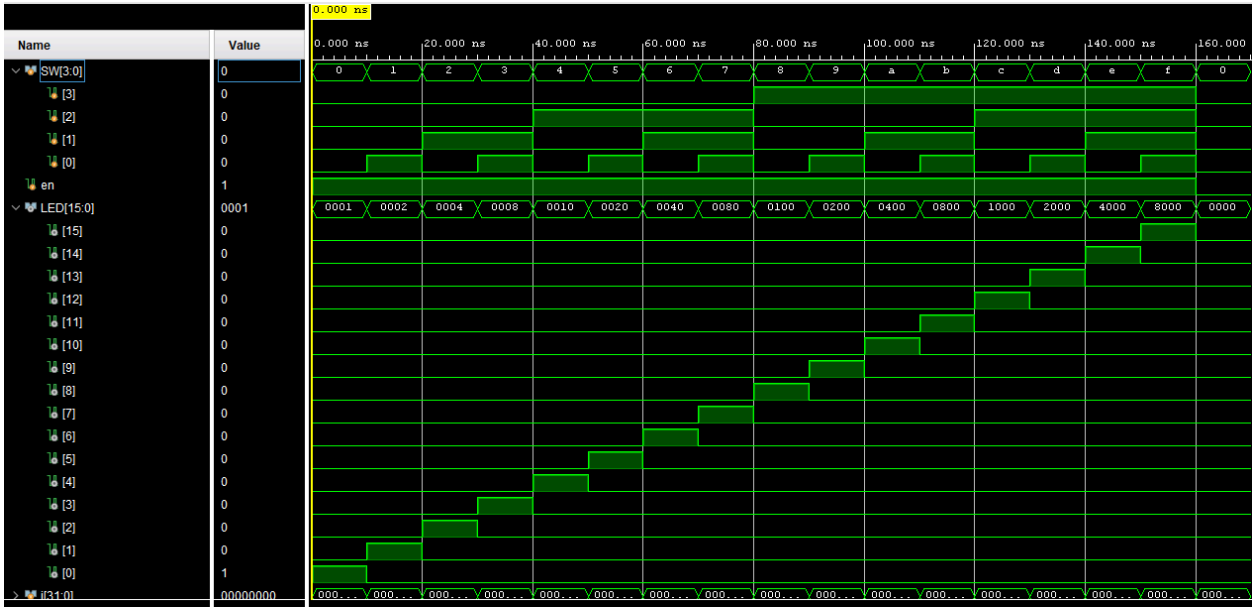
```
##Switches

set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports { SW[0] }]; #IO_L24N_T3_RS0_15 Sch=sw[0]
set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports { SW[1] }]; #IO_L3N_T0_DQS_EMCCLK_14 Sch=sw[1]
set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports { SW[2] }]; #IO_L6N_T0_D08_VREF_14 Sch=sw[2]
set_property -dict { PACKAGE_PIN R15   IOSTANDARD LVCMOS33 } [get_ports { SW[3] }]; #IO_L13N_T2_MRCC_14 Sch=sw[3]
set property -dict { PACKAGE_PIN R17   IOSTANDARD LVCMOS33 } [get ports { en }]; #IO L12N T1 MRCC 14 Sch=sw[4]
```

And we had all LEDs turned on

```
set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports { LED[0] }]; #IO_L18P_T2_A24_15 Sch=led[0]
set_property -dict { PACKAGE_PIN K15   IOSTANDARD LVCMOS33 } [get_ports { LED[1] }]; #IO_L24P_T3_RS1_15 Sch=led[1]
set_property -dict { PACKAGE_PIN J13   IOSTANDARD LVCMOS33 } [get_ports { LED[2] }]; #IO_L17N_T2_A25_15 Sch=led[2]
set_property -dict { PACKAGE_PIN N14   IOSTANDARD LVCMOS33 } [get_ports { LED[3] }]; #IO_L8P_T1_D11_14 Sch=led[3]
set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 } [get_ports { LED[4] }]; #IO_L7P_T1_D09_14 Sch=led[4]
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports { LED[5] }]; #IO_L18N_T2_A11_D27_14 Sch=led[5]
set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports { LED[6] }]; #IO_L17P_T2_A14_D30_14 Sch=led[6]
set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports { LED[7] }]; #IO_L18P_T2_A12_D28_14 Sch=led[7]
set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports { LED[8] }]; #IO_L16N_T2_A15_D31_14 Sch=led[8]
set_property -dict { PACKAGE_PIN T15   IOSTANDARD LVCMOS33 } [get_ports { LED[9] }]; #IO_L14N_T2_SRCC_14 Sch=led[9]
set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports { LED[10] }]; #IO_L22P_T3_A05_D21_14 Sch=led[10]
set_property -dict { PACKAGE_PIN T16   IOSTANDARD LVCMOS33 } [get_ports { LED[11] }]; #IO_L15N_T2_DQS_DOUT_CSO_B_14 Sch=led[11]
set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports { LED[12] }]; #IO_L16P_T2_CSI_B_14 Sch=led[12]
set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports { LED[13] }]; #IO_L22N_T3_A04_D20_14 Sch=led[13]
set_property -dict { PACKAGE_PIN V12   IOSTANDARD LVCMOS33 } [get_ports { LED[14] }]; #IO_L20N_T3_A07_D23_14 Sch=led[14]
set property -dict { PACKAGE_PIN V11   IOSTANDARD LVCMOS33 } [get ports { LED[15] }]; #IO L21N T3 DQS A06 D22 14 Sch=led[15]
```

Test bench:



Video Link:
https://youtu.be/DPPkRD5WqPA

Contributions:
Czyrone (50%) - 3 to 8 decoder code, 4 to 16 behavior code, test bench.
Caleb (50%) - Board Demo, 4 to 16 decoder using upper and lower 3 to 8 decoders.
Both worked on the lab report together.

Reflections:

In this lab, a 4-to-16 decoder with an enable input was designed and implemented using the Nexys A7-100T FPGA board. The decoder was created using two different methods, which were a gate-level approach with basic logic gates and a behavioral approach using a case statement. Both versions were tested to make sure they worked correctly. To help build the decoder more efficiently, two 3-to-8 decoder modules were used. The most significant input bit (A3) was used to choose which 3-to-8 block to activate. This made the design easier to manage and understand. A self-checking testbench was created to automatically verify that only one output was high when the decoder was enabled, and that all outputs stayed low when disabled. After testing in simulation, the design was programmed onto the FPGA. Flipping the switches on the board correctly lit up one LED at a time, showing that the decoder worked as expected. This lab helped build a stronger understanding of Verilog coding, testbench creation, and how to bring a digital design from simulation to working hardware.