ECE 3300L

Lab Report #2

Group E

Paul Kim (ID: 015236949)

Winson Zhu (ID: 016416790)

June 25, 2025

Design:

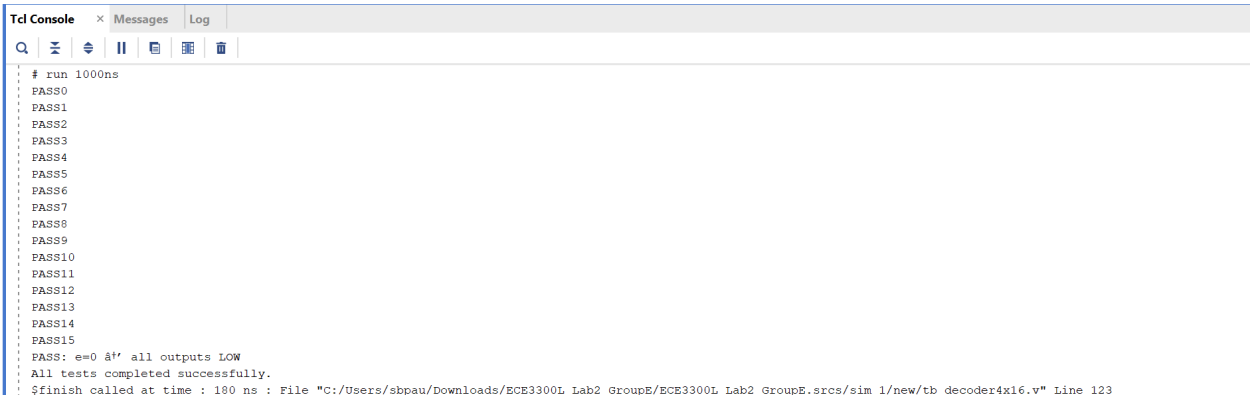
The gate-level implementation manually defines each output using Boolean expressions with AND and NOT logic. Every output bit corresponds to one unique combination of the 4-bit input A, gated by the enable E.

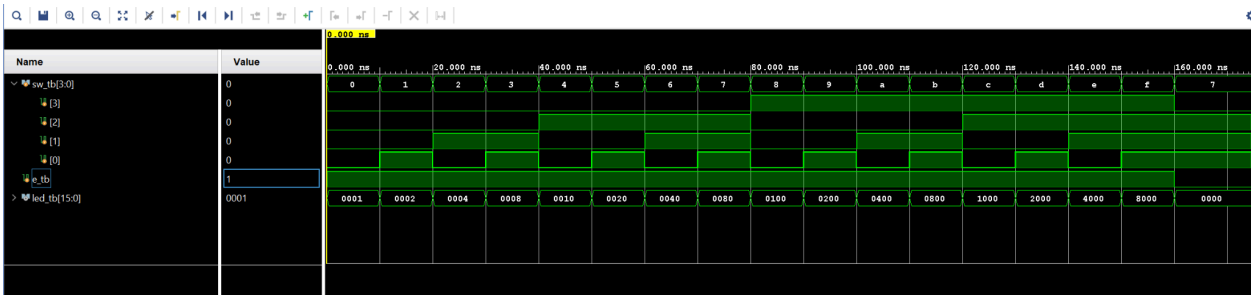
```
module decoder4x16_gate(
    input wire [3:0] sw,
    input e,
    output wire [15:0] led
    );

    assign led[0] = e&~sw[3]&~sw[2]&~sw[1]&~sw[0];
    assign led[1] = e&~sw[3]&~sw[2]&~sw[1]&sw[0];
    assign led[2] = e&~sw[3]&~sw[2]&sw[1]&~sw[0];
    assign led[3] = e&~sw[3]&~sw[2]&sw[1]&sw[0];
    assign led[4] = e&~sw[3]&sw[2]&~sw[1]&~sw[0];
    assign led[5] = e&~sw[3]&sw[2]&~sw[1]&sw[0];
    assign led[6] = e&~sw[3]&sw[2]&sw[1]&~sw[0];
    assign led[7] = e&~sw[3]&sw[2]&sw[1]&sw[0];
    assign led[8] = e&sw[3]&~sw[2]&~sw[1]&~sw[0];
    assign led[9] = e&sw[3]&~sw[2]&~sw[1]&sw[0];
    assign led[10] = e&sw[3]&~sw[2]&sw[1]&~sw[0];
    assign led[11] = e&sw[3]&~sw[2]&sw[1]&sw[0];
    assign led[12] = e&sw[3]&sw[2]&~sw[1]&~sw[0];
    assign led[13] = e&sw[3]&sw[2]&~sw[1]&sw[0];
    assign led[14] = e&sw[3]&sw[2]&sw[1]&~sw[0];
    assign led[15] = e&sw[3]&sw[2]&sw[1]&sw[0];

endmodule
```

The behavioral implementation uses an always block and case statement to set the output Y. It begins by resetting all outputs to zero, then selectively enables the correct bit based on the 4-bit input A, only when E is active.

```
module decoder4x16_behav(
     input wire [3:0] sw, e,
     output reg [15:0] led
    );
    always @(e, sw, led) begin

    led = 16'b0; // reset all outputs to 0
        if (e) begin // only decode when enabled
        case (sw)
        4'b0000:led=16'b0000_0000_0000_0001; // ▶ output 0
        4'b0001:led=16'b0000_0000_0000_0010; // ▶ output 1
        4'b0010:led=16'b0000_0000_0000_0100; // ▶ output 2
        4'b0011:led=16'b0000_0000_0000_1000; // ▶ output 3
        4'b0100:led=16'b0000_0000_0001_0000; // ▶ output 4
        4'b0101:led=16'b0000_0000_0010_0000; // ▶ output 5
        4'b0110:led=16'b0000_0000_0100_0000; // ▶ output 6
        4'b0111:led=16'b0000_0000_1000_0000; // ▶ output 7
        4'b1000:led=16'b0000_0001_0000_0000; // ▶ output 8
        4'b1001:led=16'b0000_0010_0000_0000; // ▶ output 9
        4'b1010:led=16'b0000_0100_0000_0000; // ▶ output 10
        4'b1011:led=16'b0000_1000_0000_0000; // ▶ output 11
        4'b1100:led=16'b0001_0000_0000_0000; // ▶ output 12
        4'b1101:led=16'b0010_0000_0000_0000; // ▶ output 13
        4'b1110:led=16'b0100_0000_0000_0000; // ▶ output 14
        4'b1111:led=16'b1000_0000_0000_0000; // ▶ output 15
        endcase
        end
    end
endmodule
```

Simulation:





Implementation:

Resource	Used	Available	Utilization
Slice LUTs	8	63,400	0.013%
Slices	4	15,850	0.025%
LUT as Logic	8	63,400	0.013%
Bonded IOBs	21	210	10.0%

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	inf	Worst Hold Slack (WHS):	inf	Worst Pulse Width Slack (WPWS):	NA
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	NA
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	NA
Total Number of Endpoints:	16	Total Number of Endpoints:	16	Total Number of Endpoints:	NA

There are no user specified timing constraints.

Contributions:

Paul Kim - Testbench, Behavioral Code, Simulation - 50% contribution

Winson Zhu - Gate-Level Design, Hardware Demo - 50% contribution

Group Video Link:

https://youtu.be/VDF3Er3c-Ik