Julio Flores (ID# : 016326856)
Victor Perez (ID# : 016196050)

Group I

Session E02

Lab 2

4×16 Decoder Design on Nexys A7-100T FPGA

WEDNESDAY

JUNE 25, 2025

ECE 3300L

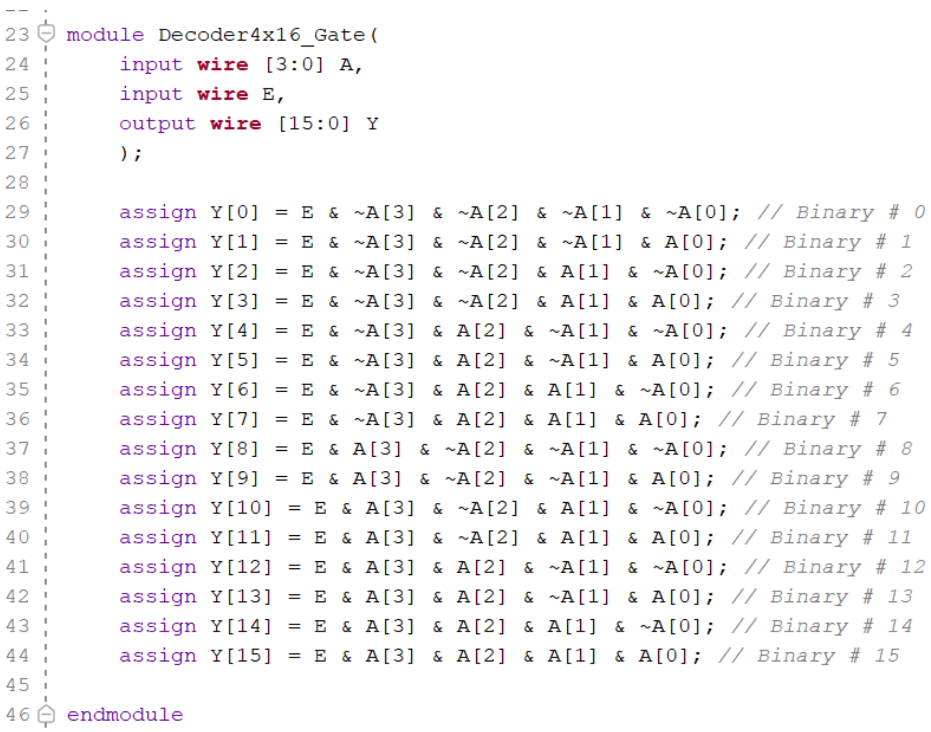
Summer 2025

Objective:

The objective of this lab is to design and implement a 4 to 16 decoder with an enable input. This lab will utilize gate level structural coding and behavioral coding in verilog along with using Vivado resources such as testbench for self-checking and timing reports. Finally implementing this code into the Nexys A7-100T FPGA board for a physical demonstration of the 4 to 16 decoder.

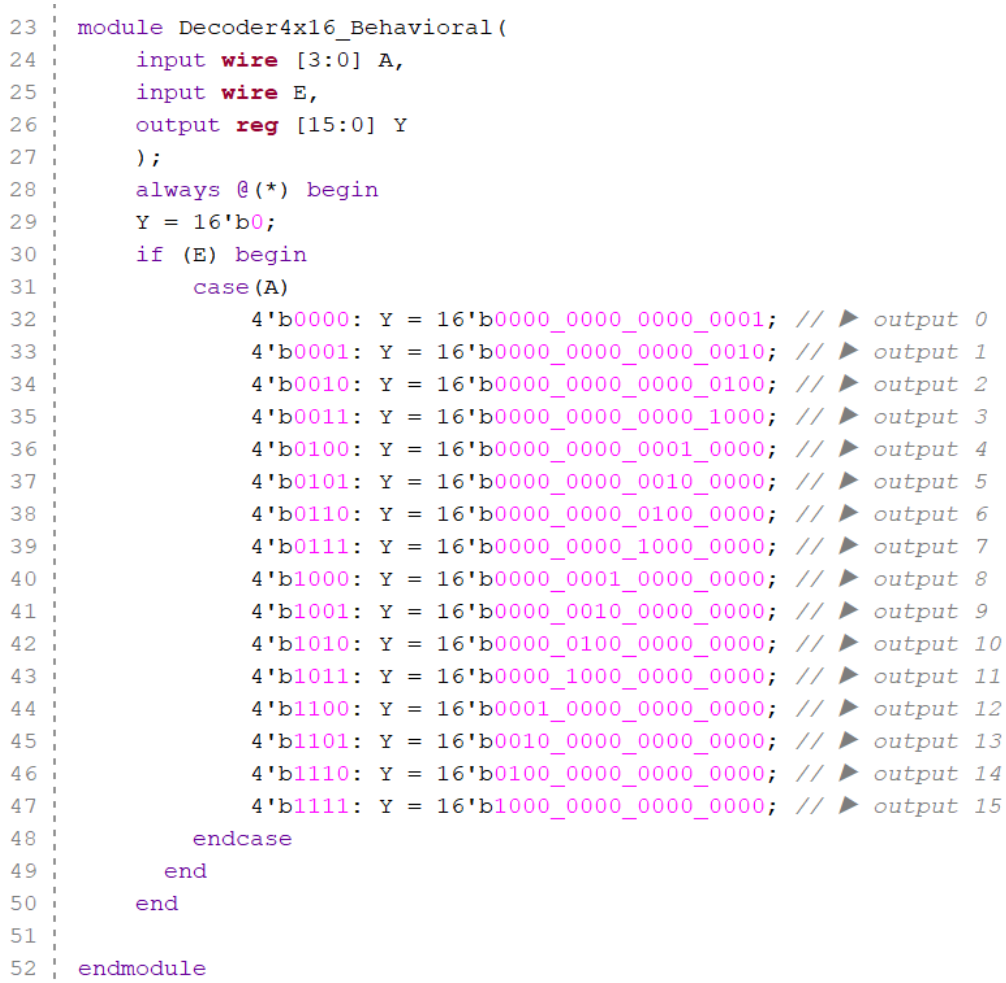
Design

Gate level code:



```verilog
module Decoder4x16_Gate(
    input wire [3:0] A,
    input wire E,
    output wire [15:0] Y
    );

    assign Y[0] = E & ~A[3] & ~A[2] & ~A[1] & ~A[0]; // Binary # 0
    assign Y[1] = E & ~A[3] & ~A[2] & ~A[1] & A[0]; // Binary # 1
    assign Y[2] = E & ~A[3] & ~A[2] & A[1] & ~A[0]; // Binary # 2
    assign Y[3] = E & ~A[3] & ~A[2] & A[1] & A[0]; // Binary # 3
    assign Y[4] = E & ~A[3] & A[2] & ~A[1] & ~A[0]; // Binary # 4
    assign Y[5] = E & ~A[3] & A[2] & ~A[1] & A[0]; // Binary # 5
    assign Y[6] = E & ~A[3] & A[2] & A[1] & ~A[0]; // Binary # 6
    assign Y[7] = E & ~A[3] & A[2] & A[1] & A[0]; // Binary # 7
    assign Y[8] = E & A[3] & ~A[2] & ~A[1] & ~A[0]; // Binary # 8
    assign Y[9] = E & A[3] & ~A[2] & ~A[1] & A[0]; // Binary # 9
    assign Y[10] = E & A[3] & ~A[2] & A[1] & ~A[0]; // Binary # 10
    assign Y[11] = E & A[3] & ~A[2] & A[1] & A[0]; // Binary # 11
    assign Y[12] = E & A[3] & A[2] & ~A[1] & ~A[0]; // Binary # 12
    assign Y[13] = E & A[3] & A[2] & ~A[1] & A[0]; // Binary # 13
    assign Y[14] = E & A[3] & A[2] & A[1] & ~A[0]; // Binary # 14
    assign Y[15] = E & A[3] & A[2] & A[1] & A[0]; // Binary # 15

endmodule
```

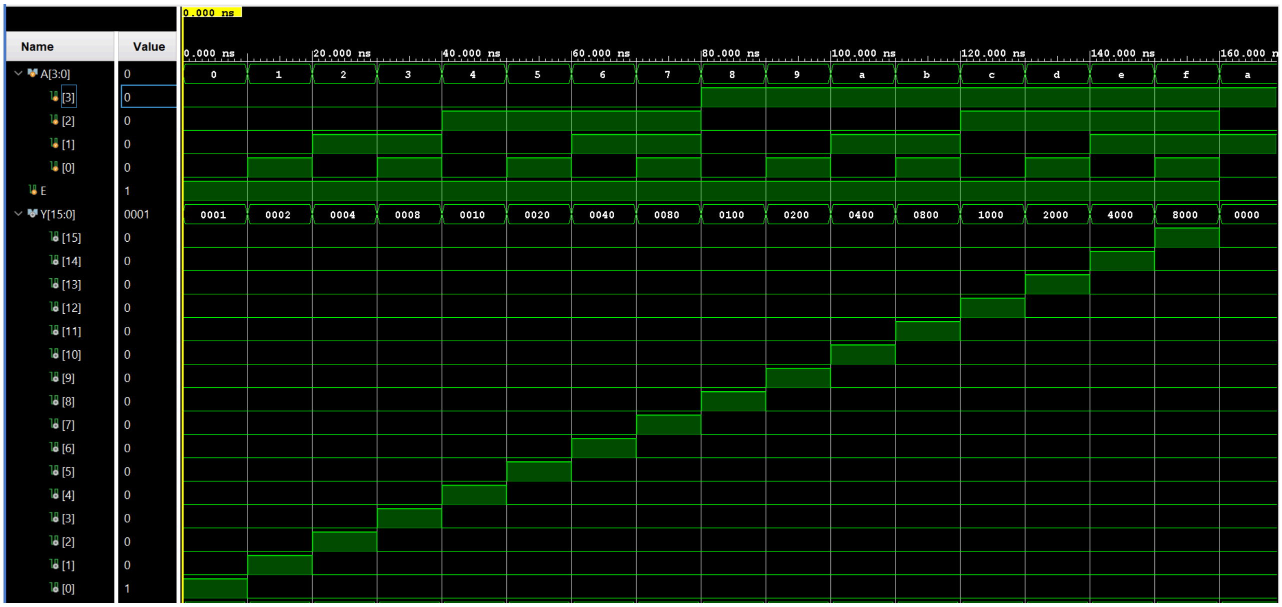
Behavioral code:



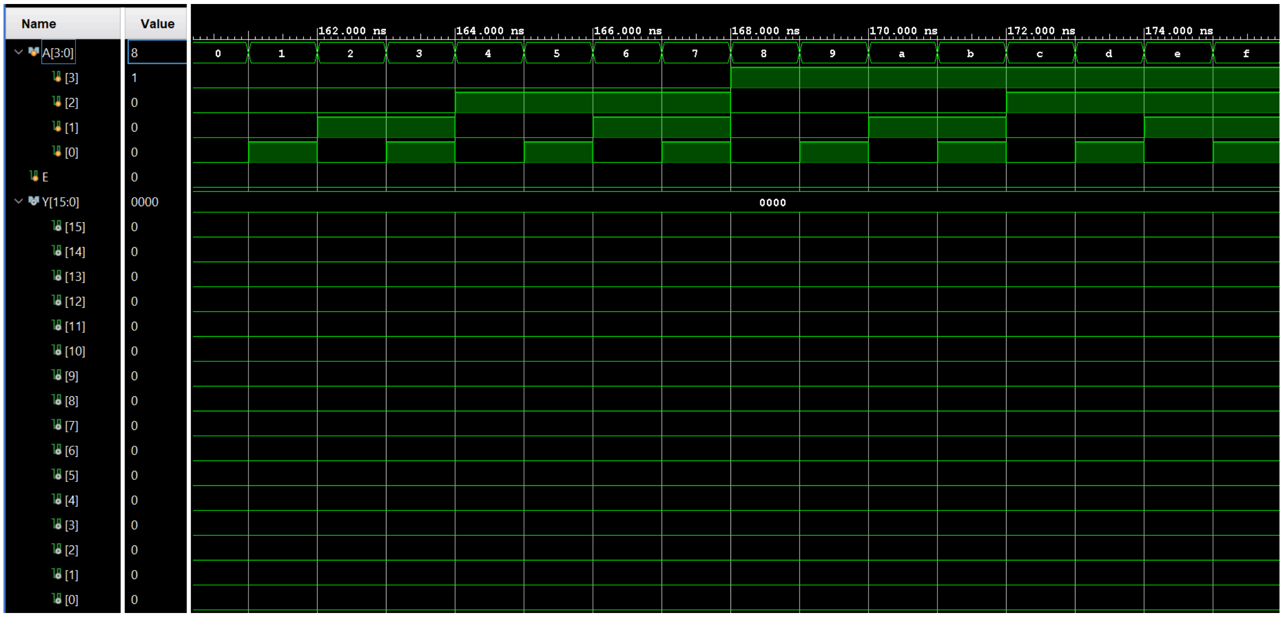
```
module Decoder4x16_Behavioral(
    input wire [3:0] A,
    input wire E,
    output reg [15:0] Y
    );
    always @(*) begin
    Y = 16'b0;
    if (E) begin
        case(A)
            4'b0000: Y = 16'b0000_0000_0000_0001; // ▶ output 0
            4'b0001: Y = 16'b0000_0000_0000_0010; // ▶ output 1
            4'b0010: Y = 16'b0000_0000_0000_0100; // ▶ output 2
            4'b0011: Y = 16'b0000_0000_0000_1000; // ▶ output 3
            4'b0100: Y = 16'b0000_0000_0001_0000; // ▶ output 4
            4'b0101: Y = 16'b0000_0000_0010_0000; // ▶ output 5
            4'b0110: Y = 16'b0000_0000_0100_0000; // ▶ output 6
            4'b0111: Y = 16'b0000_0000_1000_0000; // ▶ output 7
            4'b1000: Y = 16'b0000_0001_0000_0000; // ▶ output 8
            4'b1001: Y = 16'b0000_0010_0000_0000; // ▶ output 9
            4'b1010: Y = 16'b0000_0100_0000_0000; // ▶ output 10
            4'b1011: Y = 16'b0000_1000_0000_0000; // ▶ output 11
            4'b1100: Y = 16'b0001_0000_0000_0000; // ▶ output 12
            4'b1101: Y = 16'b0010_0000_0000_0000; // ▶ output 13
            4'b1110: Y = 16'b0100_0000_0000_0000; // ▶ output 14
            4'b1111: Y = 16'b1000_0000_0000_0000; // ▶ output 15
        endcase
      end
    end

endmodule
```

Simulation

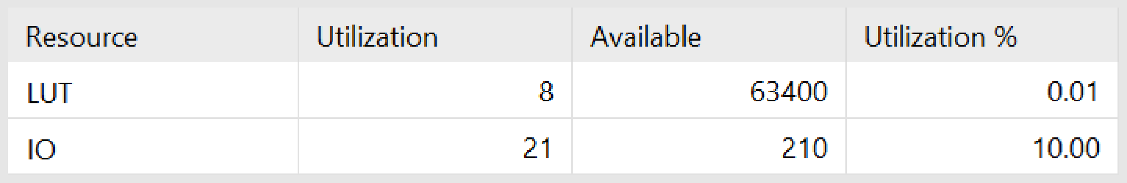
Enable = 1



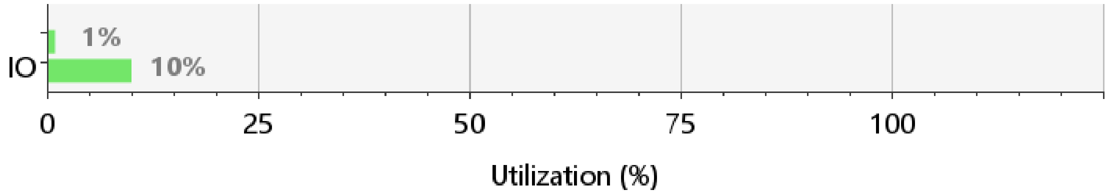
Enable = 0



Implementation



Resource	Utilization	Available	Utilization %
LUT	8	63400	0.01
IO	21	210	10.00



Contributions:

Victor Perez: 50% - Verilog code(gate) and test demo

Julio Flores : 50% - Verilog code(Behavioral) and test bench

Reflection:

This lab provided a comprehensive experience in verilog and FPGA implementation by designing a 4-to-16 decoder with an enable input. Using both gate level and behavioral coding in addition to a testbench for self-checking we were able to verify the design before implementation of the FPGA board.

Demo/Youtube Link:

https://youtu.be/HgmHyZUB2rc