4x16 Decoder Design on Nexys A7-100T FPGA

3300L
Dia Agrawal and Robert Lainez Torres

Objective

The objective of this lab is to implement a 4-16 decoder with an enable input. We will put to practice gate-level coding, behavioral coding, self-checking testbench, understanding what resource utilization reports mean, and documentation.

Hardware Connections

Hardware connections are controlled via the constraint file which is shown below.

```
## Decoder Inputs (A[3:0])
set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports { A[0] }]; # SW[0]
set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports { A[1] }]; # SW[1]
set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports { A[2] }]; # SW[2]
set_property -dict { PACKAGE_PIN R15   IOSTANDARD LVCMOS33 } [get_ports { A[3] }]; # SW[3]

## Decoder Enable (E)
set_property -dict { PACKAGE_PIN R17   IOSTANDARD LVCMOS33 } [get_ports { E }];    # SW[4]

## Decoder Outputs (Y[15:0])
set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports { Y[0] }];   # LED[0]
set_property -dict { PACKAGE_PIN K15   IOSTANDARD LVCMOS33 } [get_ports { Y[1] }];   # LED[1]
set_property -dict { PACKAGE_PIN J13   IOSTANDARD LVCMOS33 } [get_ports { Y[2] }];   # LED[2]
set_property -dict { PACKAGE_PIN N14   IOSTANDARD LVCMOS33 } [get_ports { Y[3] }];   # LED[3]
set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 } [get_ports { Y[4] }];   # LED[4]
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports { Y[5] }];   # LED[5]
set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports { Y[6] }];   # LED[6]
set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports { Y[7] }];   # LED[7]
set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports { Y[8] }];   # LED[8]
set_property -dict { PACKAGE_PIN T15   IOSTANDARD LVCMOS33 } [get_ports { Y[9] }];   # LED[9]
set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports { Y[10] }];  # LED[10]
set_property -dict { PACKAGE_PIN T16   IOSTANDARD LVCMOS33 } [get_ports { Y[11] }];  # LED[11]
set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports { Y[12] }];  # LED[12]
set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports { Y[13] }];  # LED[13]
set_property -dict { PACKAGE_PIN V12   IOSTANDARD LVCMOS33 } [get_ports { Y[14] }];  # LED[14]
set_property -dict { PACKAGE_PIN V11   IOSTANDARD LVCMOS33 } [get_ports { Y[15] }];  # LED[15]
```

In the image above, we show the decoder inputs as SW[0], SW[1], SW[2] and, SW[3] and the output as LEDs 0 to 15 (16-bit output)

Gate-Level and Behavioral implementation:

The gate level implementation shows the logic using gates to express the logic whereas the behavioral implementation shows this expressed directly in binary form.

Test-Bench:

The testbench cycles through all 16 possible input values twice — once with enable off to verify zero output, and once with enable on to confirm one-hot decoding. I used a task to check the gate and behavioral outputs against expected values at every step.

Gate-Level implementation

```
module decoder4x16_gate(
    input [3:0] A,       //[BLUE]
    input E,             //[BLUE]
    output [15:0] Y       //[GREEN]
    );

//assigning output values

// Enable LOW
    assign Y[0]  = E & ~A[3] & ~A[2] & ~A[1] & ~A[0];//[GREEN]
    assign Y[1]  = E & ~A[3] & ~A[2] & ~A[1] &  A[0];//[GREEN]
    assign Y[2]  = E & ~A[3] & ~A[2] &  A[1] & ~A[0];//[GREEN]
    assign Y[3]  = E & ~A[3] & ~A[2] &  A[1] &  A[0];//[GREEN]
    assign Y[4]  = E & ~A[3] &  A[2] & ~A[1] & ~A[0];//[GREEN]
    assign Y[5]  = E & ~A[3] &  A[2] & ~A[1] &  A[0];//[GREEN]
    assign Y[6]  = E & ~A[3] &  A[2] &  A[1] & ~A[0];//[GREEN]
    assign Y[7]  = E & ~A[3] &  A[2] &  A[1] &  A[0];//[GREEN]
// Enable HIGH
    assign Y[8]  = E &  A[3] & ~A[2] & ~A[1] & ~A[0];//[GREEN]
    assign Y[9]  = E &  A[3] & ~A[2] & ~A[1] &  A[0];//[GREEN]
    assign Y[10] = E &  A[3] & ~A[2] &  A[1] & ~A[0];//[GREEN]
    assign Y[11] = E &  A[3] & ~A[2] &  A[1] &  A[0];//[GREEN]
    assign Y[12] = E &  A[3] &  A[2] & ~A[1] & ~A[0];//[GREEN]
    assign Y[13] = E &  A[3] &  A[2] & ~A[1] &  A[0];//[GREEN]
    assign Y[14] = E &  A[3] &  A[2] &  A[1] & ~A[0];//[GREEN]
    assign Y[15] = E &  A[3] &  A[2] &  A[1] &  A[0];//[GREEN]
endmodule
```

Behavioral Implementation

```
module decoder4x16_behav(
    input [3:0] A,           // [BLUE] Input Address
    input E,                 // [BLUE] Enable Signal
    output reg [15:0] Y      // [GREEN] Output
    );

    always @(*) begin          //[ORANGE]
        Y = 16'b0;             // [ORANGE]
        if (E) begin           // [ORANGE]
            case (A)
                4'b0000: Y = 16'b0000000000000001; //
                4'b0001: Y = 16'b0000000000000010;
                4'b0010: Y = 16'b0000000000000100;
                4'b0011: Y = 16'b0000000000001000;
                4'b0100: Y = 16'b0000000000010000;
                4'b0101: Y = 16'b0000000000100000;
                4'b0110: Y = 16'b0000000001000000;
                4'b0111: Y = 16'b0000000010000000;
                4'b1000: Y = 16'b0000000100000000;
                4'b1001: Y = 16'b0000001000000000;
                4'b1010: Y = 16'b0000010000000000;
                4'b1011: Y = 16'b0000100000000000;
                4'b1100: Y = 16'b0001000000000000;
                4'b1101: Y = 16'b0010000000000000;
                4'b1110: Y = 16'b0100000000000000;
                4'b1111: Y = 16'b1000000000000000;
            endcase // [ORANGE]
        end
    end

endmodule
```

Testbench

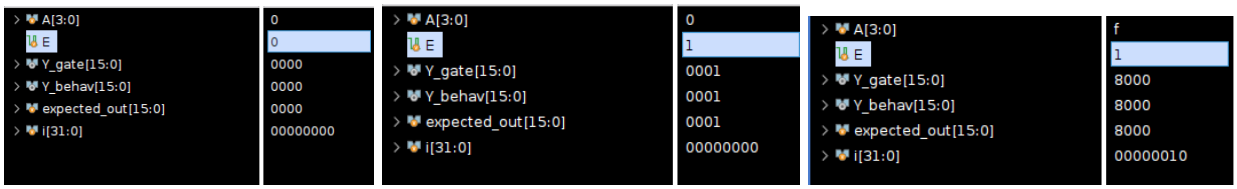
```
# run 1000ns
starting 4-to - 16 decoder testbench..
PASS GATE: E=0 A=0000 Y=0000000000000000
PASS BEHAV: E=0 A=0000 Y=0000000000000000
PASS GATE: E=0 A=0001 Y=0000000000000000
PASS BEHAV: E=0 A=0001 Y=0000000000000000
PASS GATE: E=0 A=0010 Y=0000000000000000
PASS BEHAV: E=0 A=0010 Y=0000000000000000
PASS GATE: E=0 A=0011 Y=0000000000000000
PASS BEHAV: E=0 A=0011 Y=0000000000000000
PASS GATE: E=0 A=0100 Y=0000000000000000
PASS BEHAV: E=0 A=0100 Y=0000000000000000
PASS GATE: E=0 A=0101 Y=0000000000000000
PASS BEHAV: E=0 A=0101 Y=0000000000000000
PASS GATE: E=0 A=0110 Y=0000000000000000
PASS BEHAV: E=0 A=0110 Y=0000000000000000
PASS GATE: E=0 A=0111 Y=0000000000000000
PASS BEHAV: E=0 A=0111 Y=0000000000000000
PASS GATE: E=0 A=1000 Y=0000000000000000
PASS BEHAV: E=0 A=1000 Y=0000000000000000
PASS GATE: E=0 A=1001 Y=0000000000000000
PASS BEHAV: E=0 A=1001 Y=0000000000000000
PASS GATE: E=0 A=1010 Y=0000000000000000
PASS BEHAV: E=0 A=1010 Y=0000000000000000
PASS GATE: E=0 A=1011 Y=0000000000000000
PASS BEHAV: E=0 A=1011 Y=0000000000000000
PASS GATE: E=0 A=1100 Y=0000000000000000
PASS BEHAV: E=0 A=1100 Y=0000000000000000
PASS GATE: E=0 A=1101 Y=0000000000000000
PASS BEHAV: E=0 A=1101 Y=0000000000000000
PASS GATE: E=0 A=1110 Y=0000000000000000
PASS BEHAV: E=0 A=1110 Y=0000000000000000
PASS GATE: E=0 A=1111 Y=0000000000000000
PASS BEHAV: E=0 A=1111 Y=0000000000000000
PASS GATE: E=1 A=0000 Y=0000000000000001
PASS BEHAV: E=1 A=0000 Y=0000000000000001
PASS GATE: E=1 A=0001 Y=0000000000000010
PASS BEHAV: E=1 A=0001 Y=0000000000000010
PASS GATE: E=1 A=0010 Y=0000000000000100
PASS BEHAV: E=1 A=0010 Y=0000000000000100
PASS GATE: E=1 A=0011 Y=0000000000001000
PASS BEHAV: E=1 A=0011 Y=0000000000001000
PASS GATE: E=1 A=0100 Y=0000000000010000
PASS BEHAV: E=1 A=0100 Y=0000000000010000
PASS GATE: E=1 A=0101 Y=0000000000100000
PASS BEHAV: E=1 A=0101 Y=0000000000100000
PASS GATE: E=1 A=0110 Y=0000000001000000
PASS BEHAV: E=1 A=0110 Y=0000000001000000
PASS GATE: E=1 A=0111 Y=0000000010000000
PASS BEHAV: E=1 A=0111 Y=0000000010000000
PASS GATE: E=1 A=1000 Y=0000000100000000
PASS BEHAV: E=1 A=1000 Y=0000000100000000
PASS GATE: E=1 A=1001 Y=0000001000000000
PASS BEHAV: E=1 A=1001 Y=0000001000000000
PASS GATE: E=1 A=1010 Y=0000010000000000
PASS BEHAV: E=1 A=1010 Y=0000010000000000
PASS GATE: E=1 A=1011 Y=0000100000000000
PASS BEHAV: E=1 A=1011 Y=0000100000000000
PASS GATE: E=1 A=1100 Y=0001000000000000
PASS BEHAV: E=1 A=1100 Y=0001000000000000
PASS GATE: E=1 A=1101 Y=0010000000000000
PASS BEHAV: E=1 A=1101 Y=0010000000000000
PASS GATE: E=1 A=1110 Y=0100000000000000
PASS BEHAV: E=1 A=1110 Y=0100000000000000
PASS GATE: E=1 A=1111 Y=1000000000000000
PASS BEHAV: E=1 A=1111 Y=1000000000000000
Testbench completed.
```

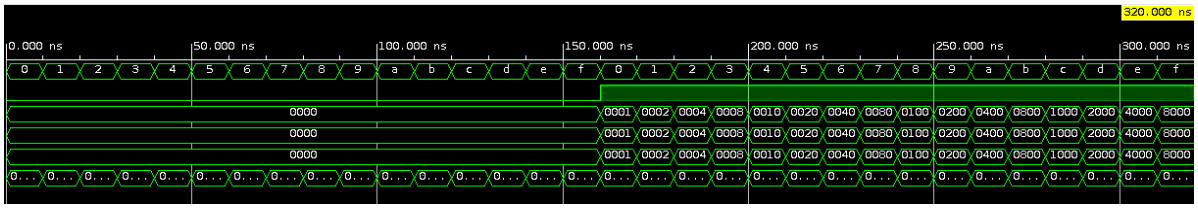
Utilization Report

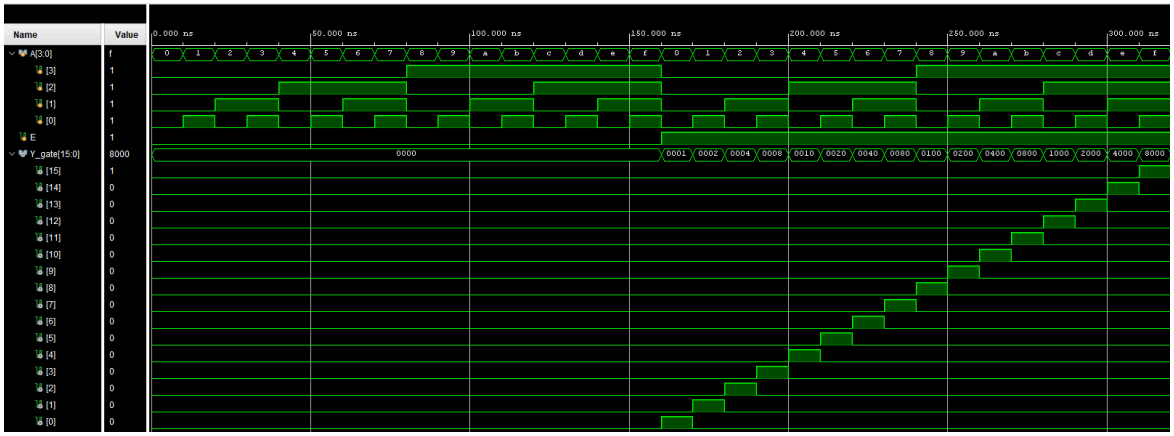
Name	Slice LUTs (63400)	Bonded IOB (210)
decoder4x16_gate	8	21

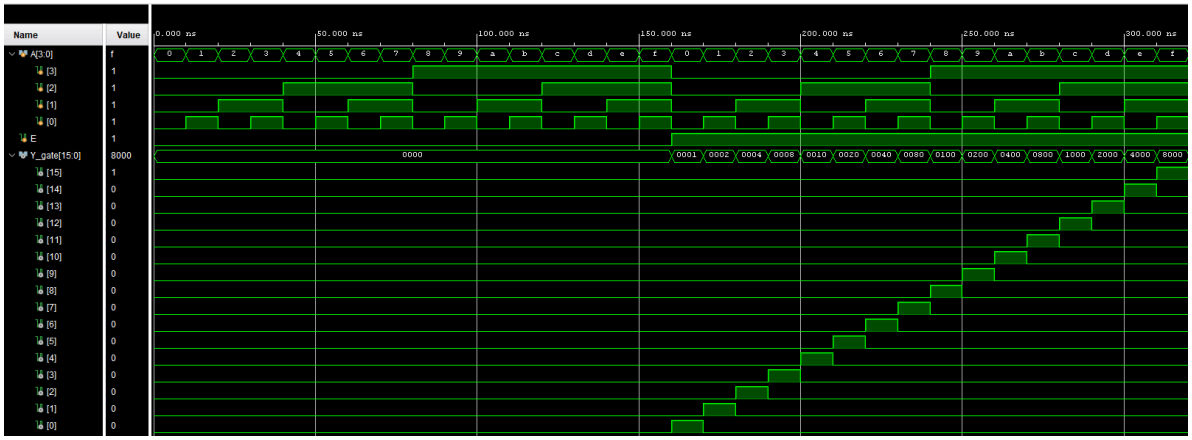
Resource usage was minimal, the design only used 8 LUTs out of 63,400 and 21 I/O pins out of 210. That makes sense since the logic is fully combinational with no registers or memory, and only one enable plus 4 input lines are needed to drive 16 outputs.

Simulation Report









The waveform and log output both confirm that the decoder behaves exactly as expected. Every output matches the intended value, and there were no failed assertions across the full range of inputs.

Video Presentation

https://www.youtube.com/watch?v=AU0JlCCPny4

Conclusion

Overall, we have been able to put into practice concepts like decoder. We have a better understanding of utilization reports, testbenching, and simulation. We have been able to apply new coding concepts like color coded comments and display messages during testbench.