California Polytechnic State University, Pomona
Department of Electrical and Computer Engineering

Digital Circuit Design Using Verilog Laboratory
ECE 3300L

Lab 2

4×16 Decoder Design

By

Jetts Crittenden (ID# 015468128) and
Evan Tram(#ID 016404570)

6/25/2025

Design:

The gate design works by instantiating individual wires that act exactly as they would in the physical circuit. This gives us a good understanding of what gates are being used, how many we would have to implement, and the logic block connections that are being made. The behavioral design forgoes all of this in favor of a simpler design that represents each LED as a bit on a 16-bit long character. This model shows the actual behavior that is intended for the design and shows various properties of how the circuit only provides one output at a time. Both designs provided the same outputs, but their code is vastly different. In the end, it will have no impact on performance since both designs will be synthesized down into the same blocks.

Design Snippets:

Behavioral-

```
task check_output;
    input [3:0] a_in;
    input e_in;
    input [15:0] expected;
    begin
        A_tb = a_in;
        E_tb = e_in;
        #10;
        if (Y_tb !== expected) begin
            $display("FAIL: A=%b E=%b → Y=%b (expected %b)", A_tb, E_tb, Y_tb, expected);
            $fatal;
        end else begin
            $display("PASS: A=%b E=%b → Y=%b", A_tb, E_tb, Y_tb);
        end
    end
endtask

initial begin
    $display("=== Starting Decoder Testbench ===");

    // Enabled: only one output should be high
    check_output(4'b0000, 1'b1, 16'b0000_0000_0000_0001);
```

Gate-

```
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];
    assign Y[1] = E & ~A[3] & ~A[2] & ~A[1] & A[0];
    assign Y[2] = E & ~A[3] & ~A[2] & A[1] & ~A[0];
    assign Y[3] = E & ~A[3] & ~A[2] & A[1] & A[0];
```

Simulation:

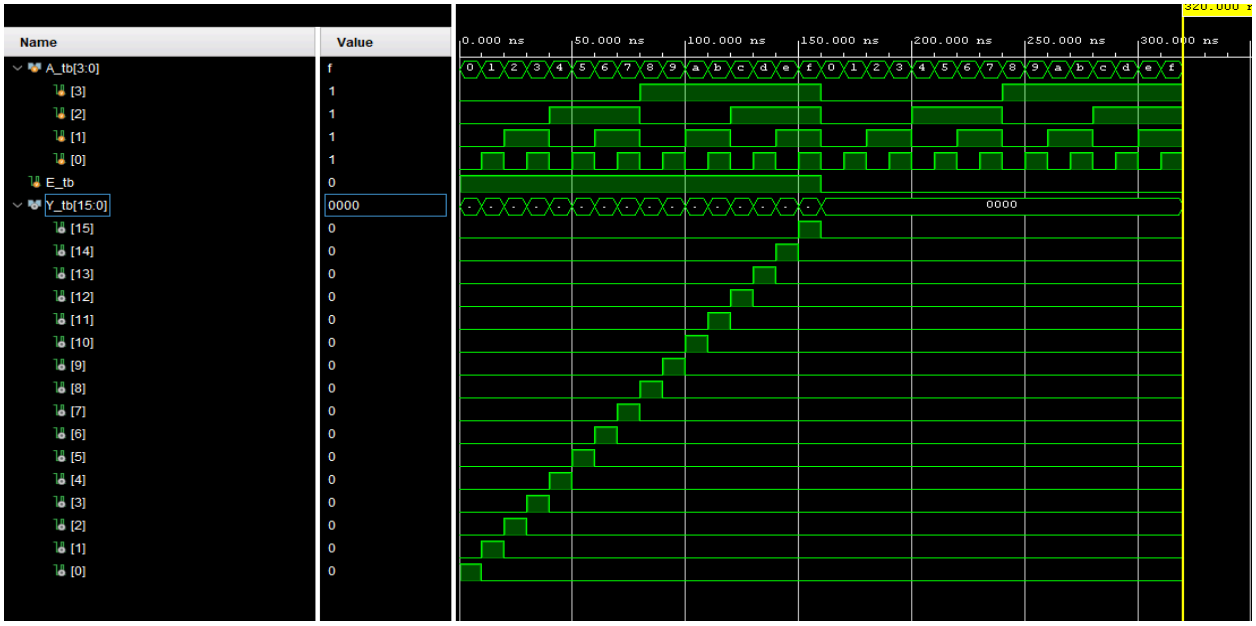
Test Bench Description:

The test bench works by looping the 16 different test values and checking the output with the correct output. If it passes, it continues and if it fails a check the routine and exited in the console. Its a simple way to check without having to read the and analyze the waveform.

Console Output:

```
PASS: A=1001 E=1 → Y=0000001000000000
PASS: A=1010 E=1 → Y=0000010000000000
PASS: A=1011 E=1 → Y=0000100000000000
PASS: A=1100 E=1 → Y=0001000000000000
PASS: A=1101 E=1 → Y=0010000000000000
PASS: A=1110 E=1 → Y=0100000000000000
PASS: A=1111 E=1 → Y=1000000000000000
PASS: A=0000 E=0 → Y=0000000000000000
PASS: A=0001 E=0 → Y=0000000000000000
PASS: A=0010 E=0 → Y=0000000000000000
PASS: A=0011 E=0 → Y=0000000000000000
PASS: A=0100 E=0 → Y=0000000000000000
PASS: A=0101 E=0 → Y=0000000000000000
PASS: A=0110 E=0 → Y=0000000000000000
PASS: A=0111 E=0 → Y=0000000000000000
PASS: A=1000 E=0 → Y=0000000000000000
PASS: A=1001 E=0 → Y=0000000000000000
PASS: A=1010 E=0 → Y=0000000000000000
PASS: A=1011 E=0 → Y=0000000000000000
PASS: A=1100 E=0 → Y=0000000000000000
PASS: A=1101 E=0 → Y=0000000000000000
PASS: A=1110 E=0 → Y=0000000000000000
PASS: A=1111 E=0 → Y=0000000000000000
=== All Decoder Tests Passed ===
```

Sample Waveform:



Implementation: Resource utilization table(s):

Site Type	Used	Fixed	Prohibited	Available	Util%
Slice LUTs*	8	0	0	63400	0.01
LUT as Logic	8	0	0	63400	0.01
LUT as Memory	0	0	0	19000	0.00
Slice Registers	0	0	0	126800	0.00
Register as Flip Flop	0	0	0	126800	0.00
Register as Latch	0	0	0	126800	0.00
F7 Muxes	0	0	0	31700	0.00
F8 Muxes	0	0	0	15850	0.00

4. IO and GT Specific

Site Type	Used	Fixed	Prohibited	Available	Util%
Bonded IOB	21	0	0	210	10.00
Bonded IPADs	0	0	0	2	0.00
PHY_CONTROL	0	0	0	6	0.00
PHASER_REF	0	0	0	6	0.00
OUT_FIFO	0	0	0	24	0.00
IN_FIFO	0	0	0	24	0.00
IDELAYCTRL	0	0	0	6	0.00
IBUFDS	0	0	0	202	0.00
PHASER_OUT/PHASER_OUT_PHY	0	0	0	24	0.00
PHASER_IN/PHASER_IN_PHY	0	0	0	24	0.00
IDELAYE2/IDELAYE2_FINEDELAY	0	0	0	300	0.00
ILOGIC	0	0	0	210	0.00
OLOGIC	0	0	0	210	0.00

Timing summary:

```
------------------------------------------------------------------------------------------------
| Design Timing Summary
| ---------------------
------------------------------------------------------------------------------------------------

    WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints      WHS(ns)      THS(ns)  THS Failing Endpoints  THS Total Endpoints     WPWS(ns)     TPWS(ns)  TPWS Failing Endpoints  TPWS Total Endpoints
    -------      -------  ---------------------  -------------------      -------      -------  ---------------------  -------------------     --------     --------  ----------------------  --------------------
        inf        0.000                      0                   16        inf        0.000                      0                   16           NA           NA                      NA                    NA


There are no user specified timing constraints.
```

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack:                    inf
  Source:                 A[0]
                            (input port)
  Destination:            Y[13]
                            (output port)
  Path Group:             (none)
  Path Type:              Max at Slow Process Corner
  Data Path Delay:        10.142ns  (logic 5.384ns (53.082%)  route 4.759ns (46.918%))
  Logic Levels:           3  (IBUF=1 LUT5=1 OBUF=1)
```

Worst Negative Slack (WNS): ∞ — perfect timing, no violations
Total Negative Slack (TNS): 0 ns — no failing paths
Data Path Delay (Worst Case): 10.142 ns

- **Logic delay**: 5.384 ns
- **Routing delay**: 4.759 ns
- **Total logic levels**: 3

Critical path: From input A[0] to output Y[13]

Group video link:

https://youtu.be/KDXrhcK-BLk?si=C7Dqzie8rYXwDHWG

Contributions:

Our team, Jetts Crittenden and Evan Tran, went 50/50 on contribution. Since we had two styles of HDL development, we split our tasks into gate and behavioral. The testbench was developed and shared between users because it could work with both gate and behavioral implementations. The writing responsibilities were also shared.