**ECE 3300L**

**California State Polytechnic University, Pomona**

**Group G**

**Nathaniel Garcia and Mikael Parsmyan**

**Lab Report #2**

**06/25/2025**

**Summary:**

The main goal of this lab was to design a 4-to-16 decoder with an enable input. A decoder takes binary input and turns on a single output line. For our lab, the input was 4 bits (A[3:0]), and the output was 16 bits (Y[15:0]). We also had an enable signal (E) that controls the decoder to be active. When E = 0, then all of the outputs are 0. When E = 1, then only one output turns on, based on the value of A.  
  
We created two versions of the decoder: gate-level using modules and logic gates, and behavioral using a case statement. The gate-level version used two 3-to-8 decoder modules. The lower 3 bits went in both of the decoders, and the highest bit was used to pick which decoder to enable. If A[3] = 0, then the lower outputs Y[0–7] were used. If A[3] = 1, then the upper outputs Y[8–15] were active. Each output line was created using AND and NOT logic. The behavioral version used an “always block” and a case statement to set the correct output line to 1, depending on the input, which was easier to write and simulate. The testbench was used to check that the outputs were correct for all of the 16 input values, and that all outputs stayed low when E = 0.

**Design:**

**Decoder 4x16 (Made of two 3x8 Decoders):**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A close-up of a computer screen

AI-generated content may be incorrect.**

**Decoder 4x16 Behavioral:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Simulation:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Implementation:**

**A screenshot of a white box

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Contribution:**

**Nathaniel Garcia:** Xdc file, code for testbench, simulation and Demo

**Mikael Parsmyan:** Code for 3x8 Decoder, 4x16 Decoder, and 4x16 Decoder behavioral

Youtube Demo Link: https://www.youtube.com/watch?v=jyJIWkoec7k