CDA 4203L

Computer System Design Lab

Lab 3 Report

Programmable BCD Counter

|  |  |
| --- | --- |
| Today’s Date: | 02/10/2022 |
| Team Members: | Thomas Bivins |
| Patrick Cook |
| Josue Lugo Roldan |
| Work Distribution: | Briefly explain the tasks completed by each team member  Thomas and Patrick – Code writing and video recording  Josue – Code troubleshooting and lab report  All 3 worked on troubleshooting and synthesizing into board. |
| No. of Hours Spent: | 10 |
| Exercise Difficulty:  (Easy, Average, Hard) | Average |
| Any Other Feedback: |  |

**Problem 1:** Draw overall block diagram of the programmable counter. Briefly explain how your design works. For each block, include the Verilog code. *Use as many pages as needed.*

This programmable BCD counter contains a control unit that controls the clock and the run signal which tells the counter when to start counting and how far to count. The following code shows the section of code that controls the clock and the run signals that tell the counter when to start start counting

Diagram

Description automatically generated



This module contains the logic to stop a counter when it reaches a designated value. The maximum value is 99.

Text, application

Description automatically generated

Background pattern

Description automatically generated with low confidence

This is the top module for the programmable BCD counter. It implements a programmable 7-bit counter and a binary-to-bcd converter that can output two digits.

Text

Description automatically generated

**Problem 2:** Testbench code: Include your Verilog testbench. *Use as many pages as needed.*

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Problem 3:** Simulation Waveforms. Include waveforms that demonstrate the functionality. *Use as many pages as needed.*

As the waveforms show, the count does not start until after the run signal goes up to 1 and it stops at the max\_count and ignores any input until the run signal goes down to 0, it then resets to 0 and starts the count over until the new max\_count is reached, if the max\_count is greater to 99, it counts until 99 and then stops.

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

Graphical user interface, diagram

Description automatically generated