ECEN2350 Digital Logic - Project 2

Initial Diagrams Due Friday March 23rd (midnight)
Final Due Sunday April 8th (Midnight) - 30 Points [+3pts EC]

Overview

For this project you will gain more experience with using verilog to design and implement a reaction timer with digital readout, similar to the one described in **Section 5.14.1** in the book.

You must use the DE10-LITE board (Available in the EE Store) with the MAX 10 10M50DAF484C7G Device. Groups of 1-2 students, with **1 Report / person.** Grading will be done on the CCC Rubric (See D2L).

Outcomes:

- Design your system using finite state machines
- Design a technical portfolio site entry
- Implement counters, dividers and shift registers in verilog

Requirements

Read Section 5.14 in the book. The basic operation of the reaction timer to be designed and built for this project is as follows. To start a measurement a "Start" button on the DE0 board is pushed. After releasing the Start button a random amount of time passes before a LED (or an other suitable start signal) is turned on and the reaction timer starts counting milliseconds. Once the LED is on, the person measuring their reaction time has to press a "Stop" button on the DE0 board as quickly as possible. Pushing the Stop button stops the timer and the reaction time gets displayed in milliseconds on the 7-segment display of the DE0 board. If the Start button is pushed before the start signal occurred, then the start of the measurement should be delayed until the button is released. You need to implement an asynchronous reset button that should take you back to the first state.

You are required to provide the following modules at a minimum:

- Project2_top.v
    - The top level verilog file
- BCD_decoder.v
    - The decoders used to print decimal numbers to the displays
- BCD_counter.v
    - The count of the BCD counter using the 7-segment display. You will need to show at least three digits or more.
- Clock_divider.v

    - A method to divide your clock down. **Needs to use parameters and procedural assignment.**
- LFSR.v
    - To truly test the reaction time, there has to be a random delay between pushing the Start button and the actual start of the test. To generate random numbers on the DE10 board you can use a LFSR Generator (linear feedback shift register generator).

Verify that the initial delay is indeed random. Run a couple of trials and make a histogram from at least 10 delay times. Run your reaction tester for 10 reaction time measurements (invite some friends to try it!). List the 10 reaction times in your report.

Questions that should think about with your design and answer on your technical portfolio site:

1. How do you generate a 1kHz clock from the 50 MHz clock on the DE0 board?
2. How do you implement a BCD counter that counts the milliseconds.
3. Describe your strategy to generate a random delay between pushing the Start button and lighting up the LED that indicates that the measurement of the reaction time has actually started.

Start with designing a block diagram for the project before you go into coding. This is a requirement for the report. This should show all your modules connect (basically a picture of what the Project2_top.v file is doing. **DO THIS BEFORE STARTING VERILOG**

You need to design a state machine for this system. Diagram all states, state variables, and state transitions. **DO THIS BEFORE STARTING VERILOG**

You will have to add an extra state in order to track the "high score" of the project. This should automatically compare and update if your reaction time is faster the current high score. You need to be able to display the high score with a new input. So switch between your initial state (idle/ready) and the high score menu.

Finally, you should record a short video of your working project showing how the reaction timer works.

Extra credit embellishments for your reaction timer

1. (2pts) Write a scrolling message that says "Go Buffs" on your display when the actual test starts.
2. (1pts) Add more randomness by specifying (e.g., in the form of a binary code) a randomly chosen button or slide switch that needs to be actuated to stop the timer.

Deliverables

There are two due dates for this assignment:

- Block diagrams and state machine design are Due Friday March 23rd (midnight) on the project_2_diagrams dropbox page. Please turn in a PDF of these diagrams. You can update these diagrams for the final submission.
- Final Report Portfolio Site is Due Sunday April 8th.

Report:
A simple report with a link to your portfolio site and your partner's site. You should also include your verilog files nicely formatted in the report and explain the operation. In addition, you should zip up all of your verilog files and submit those to dropbox.

Technical Portfolio Entry: You need to create a portfolio entry of this project. You need to create a 1-2 pager entry for your reaction timer in the form of an online how-to/blog. You can create a technical portfolio entry on this site: https://ecetp.colorado.edu/. You site has to have your name in the url (Example https://ecetp.colorado.edu/AlexFosdick). **Every person must create their own.** You may reuse diagrams, but you must write your own content and film your You should create a tab specific for projects. Then you should create a sub-tab for you Delay Timer Project.

Your project overview site must include the following

- **Project Details:** Title, partners, class, semester, software/hardware requirements
- **Introduction**: 1-paragraph introduction on what the project is including a block diagram of your design.
- **Block Diagram.** This should show all your modules connect (basically a picture of what the Project2_top.v file is doing. Please submit this as a digital drawing (use draw.io or your favorite program to design this). **THIS IS NOT A STATE MACHINE DIAGRAM.**
- **State machine Diagram:** How did your reaction timer function? Show the transitions, mealy/moore machine, inputs/outputs in addition to your system diagram.
- **Results**: Explain how your design functioned
- Data on randomness (histogram) and data from at least 10 users.
- Pictures (if applicable)
- **Video:** A link or an embedded video of the delay timer working
- **(Extra Credit)** A short demo to the instructor team of the extra credit circuit in action or an extra video on your portfolio site.
- **Conclusion/Takeaways:** 1-paragraph conclusive statement on the project. What was the most difficult part of this project? If something did not work, what is the reason and how would you correct it? What would you do different next time?

Do NOT put your verilog files on your site. A zero will be awarded for this.

Technical Report Options

Each group member will need to implement their own technical portfolio. You can use the same videos and images, but you must write your own content. You will need to register your technical report on this google form:
https://goo.gl/forms/nyGMWz9y0W22nuZG3

Many different sites exist for hosting your technical report. Here are a few suggestions

- Wix.com
- ecetp.colorado.edu (sign in with identikey)
- Google sites
- https://jekyllrb.com/
- Other: Please share with me your options

Resources

- Quartus 16.1 Prime Software Lite Edition (Free): https://www.altera.com/downloads/download-center.html
- Be sure to install Modelsim and the components for the DE10-LIT MAX10 FPGA chipset.
- Terasic Informaiton: http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=1021
- Here are a few reflex/reaction games to improve if you're looking for more inspiration.