Computer Systems Design - Lab 5

Handed Out: Thursday, 05 Feb. 2015
Due: Thursday, 12 Feb. 2015 by 11:00 AM; Demo During Lab 12 Feb.
Submit via USF Canvas
This is a team assignment. Teams can have up to 3 members
Late submissions will not be accepted

Objective:

To design and construct a *synthesizable* Finite State Machine and Datapath which computes the greatest common denominator (GCD) of two numbers as explained & discussed in lecture.

Problem:

Implement one of the two GCD FSMD designs discussed in the class lecture and outlined in the slide set. Those two designs are your only options.

Description:

Your design will take in two 4-bit numbers and output the binary value of the greatest common divisor of those two numbers. The numbers will be input via the dip switches, the result will be output via the LEDs, and control is done via pushbuttons.

- Input **X** uses switches 7-4. SW7 is the MSB.
- Input **Y** uses dip switches 3-0. SW3 is the MSB.
- Input **START** uses the Center pushbutton. This button needs to be debounced.
- Input **RESET** uses the RESET[1] special pushbutton. This button *does not* need to be debounced[2].
- Output **GCD_OUT** uses LEDs 3-0; LED 3 is the MSB.
- Output **DONE** uses the LED 7.

Your design will consist of two components - the controller and the datapath. The controller is to be a *pure* FSM. The datapath operates based on signals generated by the controller FSM - it should have no independent controlling logic.

IMPORTANT: The datapath must be constructed ***structurally***. This means that you can only instantiate components, and connect them with wires. You will need to create individual components (Adder, Comparator, Register, etc) for each element in the datapath. Registers are the only clocked component, and they should be triggered by the clock on the *opposite* edge than that of the Controller FSM. All non-register components must be *purely* combinational - no sequential elements. Your registers should have some type of enable signal for control, and a separate reset signal. Check the Template library for examples/starting points.

- **FSMD** *[50 points]*: Design, build and verify the GCD FSMD. ***DEMO REQUIRED***

[1] Be aware that the RESET pushbutton is **active low**

[2] CPU RESET is mechanically debounced at the hardware level. As a result, this pushbutton can uniquely be used as a trigger to an always block. Useful for asynchronous reset.

Design Constraints: The designs must properly function given the following constraints:

- The design must both function correctly in simulation *and* on the FPGA board.
- The Start input must be properly debounced.
- USER_CLK is used as the system clock.
- The GCD_OUT and DONE output should be held until RESET is pressed.
- Inputs X and Y, and output GCD_OUT are 4-bit unsigned numbers.
- RESET must be properly handled.
- The design *must* be one of the two GCD FSMD implementations (normal and optimized) found in the lecture slides.

Lab Report: You must submit a written lab report. The lab report will consist of (at least) the following sections:

- **Cover:** Include a list of team members (with U#'s) and work distribution for each team member.
- **Introduction:** In your own words, briefly describe the lab assignment.
- **Design:** Briefly describe the structure and function of your design. Discuss your design decisions and any problems you encountered during the design process.
- **Testing:** Discuss how you verified the functionality of the design. This should include a description of your testbench and test vectors, and a brief justification of the completeness of the tests you chose. Also discuss any problems you had with the testing process.
- **Synthesis:** Discuss the synthesis process and results, including any warnings or errors encountered. Describe the function of your design on the FPGA board and discuss how you verified the functionality. Be descriptive.
- **Results:** Discuss the results of the design process, testing and the assignment overall. This should include a discussion of any simulation results and an overall analysis of your design.
- **Conclusion:** Briefly summarize your accomplishments and discuss any failures. Briefly discuss your thoughts, concerns, and feedback for this assignment. If you choose, you may rant about the software, programming language, the assignment itself, or the TAs responsible for the lab.

All reports MUST be submitted in PDF format. No exceptions.

Deliverables: Submit your report PDF and the code ZIP file separately on Canvas. The PDF should be named <TeamName>_lab05_report.pdf. To generate the code ZIP file:

- Select **Project** → **Cleanup Project Files...** from the menu.
- Select **OK** on the pop-up box.
- Select **Project** → **Archive...** from the menu.
- On the new window, make sure **Exclude Generated Files From Archive** is *selected*.
- Choose a location for the archive, name it <TeamName>_lab05_src.zip.
- Select **OK**

Insert any images and waveforms into the report PDF. Submit the report PDF and source code ZIP separately on Canvas. If you wish, you may include copies of all images in a separate ZIP file, submitted through Canvas. Name it <TeamName>_lab05_images.zip.

No re-grading will be done for failure to follow the guidelines. Points will be lost, never to return.

This lab assignment requires a live demonstration.

Demonstrations will occur during lab time on the day the assignment is due.