Lab 13

Part B – 8-Bit Serial Multiplier

Description

- In this module, you will synthesize, implement, upload, and test a hardware description for an 8-bit serial multiplier.
  - This multiplier operates sequentially (i.e., in series). It is *not* encapsulated within a serial transceiver, unlike 2LMR from 12C.

Procedure

Project Creation

1. Download the files below from Canvas and place them in a new folder titled `lab13b`.
   - `lab13b.xdc`
   - `lab13b.vhd`
   - `adder_8.vhd`
   - `controller.vhd`
   - `dataflow.vhd`
   - `display_controller.vhd`
   - `f_abs.vhd`
   - `full_adder.vhd`
   - `sign_magnitude_f.vhd`
2. Open Vivado and create a new project titled appropriately.
   - For your convenience, the board identifier is `xc7a35wtcpg236-1`.
     - You can also find the board with the same options as before: `General Purpose`, `Artix-7`, `cpg236`, `-1`.

Hardware Review, Upload, and Test – 4 Bit CLA Adder

3. Open `lab13b.vhd` in the editor, and follow along with the instructor to review the hardware description of it and the other files.

4. `Open Elaborated Design`, review the schematic alongside the instructor, and take screenshots for your report as follows:.
   - Take one screenshot of the entire multiplier.
   - Take one screenshot zoomed in to just the `dataflow` component.
   - Take one screenshot zoomed in to just the `controller` component.
   - Take one screenshot zoomed in to just the `sign_magnitude_f` component.
     - This should show 14 `f_abs` components.
5. Run synthesis and implementation and then generate the bitstream to program the device. Then, open the Hardware Manager.
   - Take a screenshot of the resource utilization (LUTs and FFs) to add to your report.
6. Before proceeding, plug the FPGA development board into your computer using the provided Micro-USB to USB-A cable.
7. At the top, in the green banner (or under the `Open Hardware Manager` dropdown) click `Open Target`, then `Auto Connect`.
8. Once the device is connected, select `Program Device` (in either of the locations where `Open Target` was previously).
   - See 06B-handout for troubleshooting steps, or ask the instructor or TA.
9. Click `Program` to upload the bitstream to the FPGA.
10. The GPIO is organized as follows:
    - The left half of the switches represent the first term (multiplicand).
    - The right half of the switches represent the second term (multiplier).
    - The 7-segment displays show the 16-bit result of the multiplication.
11. If you have not already, make a copy of the template spreadsheet provided at the following link: https://docs.google.com/spreadsheets/d/1LUeCaYJ9kc3Up1m0VH3xfaVEhl34mjmvGhT8vcU10Sg/edit?usp=sharing
    - Test each of the inputs on the board and record your results.
    - During one of your tests, take a picture of the board.
12. What is the difference between the pipelined array multiplier implementation and the serial multiplier implementation? Did any of your results differ between the two? How do these implementations differ from the two-level multiplier from 12C?
13. Why is the Mealy state machine cheaper than the Moore state machine?
    - Hint: The Moore state machine implemented in the serial multiplier requires 27 states, while if we have chosen Mealy we would have used only 10 states.

14. Why is the Moore state machine faster than the Mealy state machine but requires more clock cycles to be done?
    - Hint: for a Moore state machine you need to pass through 19 states from start to done, for a Mealy state machine you need to pass through 10 states from start to done.
15. Create a table evaluating the two multipliers (array and serial) from Parts A and B based on the following criteria:
    - Area (Complexity)
    - Speed
    - Power
    - Latency
    - Throughput
    - Modularity

Deliverables

- Include as part of your **informal report**:
    - Screenshots of Elaborated Design schematics (Step 4)
    - Screenshot of FPGA resource utilization (Step 5)
    - A picture of the FPGA development board (Step 11)
    - Completed testing spreadsheet from Parts A and B (Step 11)
    - Answers to discussion questions (Steps 12, 13, 14)
    - A table evaluating the hardware from Parts A and B (Step 15)

Outcomes

- Practice working with VHDL.
- Practice using Vivado for hardware synthesis and implementation.
- Practice programming and testing a hardware description on an FPGA development board.
- Understand the implementation of a serial multiplier.
- Practice comparing different hardware.
- Practice evaluating competing hardware designs based on a set of metrics.