Lab 08

Part D – 8-Bit Signed Array Multiplier on 7SD

Description

In this lab module, you will implement and test an 8-bit signed array multiplier on the multiplexed seven-segment display, building upon 08B and 08C.

As usual, for this lab module, you will require a PC with a USB-A port to program the FPGA development board.

Procedure

Project Creation

1. Download `lab08d.vhd`, `display_controller.vhd`, `full_adder.vhd`, `half_adder.vhd`, `adder_9.vhd`, `part_prod_gen.vhd`, and `lab08d.xdc` from Canvas and place them in a new folder titled `lab08d`.
   - You can also copy `display_controller.vhd`, `full_adder.vhd`, and `half_adder.vhd` from 08C, but you must download the other new files.
2. Open Vivado and create a new project titled appropriately.
   - For your convenience, the board identifier is `xc7a35tcpg236-1`.
     - You can also find the board with the same options as before: `General Purpose`, `Artix-7`, `cpg236`, `-1`.

Hardware Review and Simulation

3. Open `lab08d.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.

5. Setup the behavioral simulation.
    - In Vivado, from the left side pane, click `Run Simulation`, then `Run Behavioral Simulation`.
    - Minimize the two panes on the left with `Scope` and `Objects` tabs to make the wave view bigger.
    - Optionally, minimize the bottom pane (`Tcl Console/Messages/Log`).
    - Optionally, drag the handle next to `Value` to make the columns larger.
    - Set the radix for all of the signals that are by default included in the waveform to `Signed Decimal`.
        - To select: within the waveform graph, under the `Name` column, click first signal, then hold `Shift` and click last signal.
        - Right-click one input and click `Radix`.
        - Select `Signed Decimal`.
6. Test the following values for the `multiplicand` and `multiplier` and check the output signal. **Do not close the simulation yet.**
    - You only need to use the `multiplicand` and `multiplier` inputs, and look at the `product` output. For this simulation, everything else can be ignored.
        - If desired, you can remove the other signals from the waveform by right-clicking them and clicking `Delete`.
    - Right-click an input (ex. `multiplicand`) and select `Force constant`.
    - Change the value radix to `Signed Decimal`.
    - Enter the decimal value of the input (ex. -1) as the `Force value`.
    - Click `OK`.
    - To run the simulation for a specified time, click the play button with `(T)` underneath, or press `Shift+F2`.
    - While holding `Ctrl`, you can use the mouse scroll wheel to zoom in and out in the waveform view.
    - While holding `Shift`, you can use the mouse scroll wheel to navigate horizontally along the waveform view.
    - If you accidentally close the waveform view, go to `Window` in the toolbar, and click `Waveform`.

Multiplicand	-1	-128	-128	-127	127	127	-128	0	-128
Multiplier	-1	-128	-127	-128	-128	127	127	127	0

7. At this point, take screenshots of your simulation results for your report.
8. Now, switch the radix of your signals to `Hexadecimal` and take another screenshot. This will help you compare your simulation results to what you will see on the FPGA development board.
9. Calculate each of the output values using a standard calculator (physical, online, in Excel, etc.) Do the values from your simulation match the expected output?

Hardware Upload

10. 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.
11. Before proceeding, plug the FPGA development board into your computer using the provided Micro-USB to USB-A cable.
12. At the top, in the green banner (or under the `Open Hardware Manager` dropdown) click `Open Target`, then `Auto Connect`.
13. 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.
14. Click `Program` to upload the bitstream to the FPGA.

Hardware Testing and Analysis

15. 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.
16. Test the hardware using the same inputs from the table above.
    - Compare each output to the screenshot you took of your simulation with hexadecimal radix. Do the results match? Discuss in your report.
    - Take a picture of the FPGA development board during one of your tests.

Deliverables

- Include as part of your **informal report**:
  - A screenshot of the Elaborated Design schematic (Step 4)
  - Screenshots of simulation results (Steps 7, 8)
  - A screenshot of the FPGA resource utilization (LUTs, FFs) (Step 10)
  - A picture of the FPGA development board (Step 17)
  - Answers to handout questions (Steps 9, 16)

Outcomes

- Practice working with VHDL.
- Practice using Vivado for hardware simulation.
- Practice using Vivado for hardware synthesis and implementation.
- Practice programming and testing a hardware description on an FPGA development board.
- Practice working with two's complement on an actual piece of hardware.
- Understand how an array multiplier works.
- Understand VHDL components, instantiation, and structural connections.