Lab 3: Chipscope

Joseph Carlos, Alice Tsai, Elon Bauer

Design

We used a very simple design to test the Chipscope tool, which was just a counter with an increment and decrement signal asserted whenever a button was pressed and released, to avoid long synthesis times or other possible problems.

Chipscope Cores

We used the Chipscope ILA and ICON cores in our lab. We integrated them using ISE Project Navigator and explicitly adding the module definitions to our code. These cores enable the capture of simple triggered data. The ICON core is the Chipscope controller, and the ILA core actually gathers the triggered data.

Unused Chipscope Cores

We didn’t use the ATC2 or IBERT Cores. The ATC2 allows connection to external probes, and the IBERT allows debugging of the Gigabit Ethernet transceiver. Neither of these cores are even supported by our board.

Process

The process for using Chipscope through ISE is surprisingly simple. We were luckily able to find the Chipscope user manual using Google at http://www.xilinx.com/support/documentation/sw_manuals/xilinx13_1/chipscope_pro_sw_cores_ug029.pdf. Using the sections entitled “Generating an ILA Core” and “Generating an ICON Core,” we were able to determine the following flow to get Chipscope up and running on our Spartan-5:

1. Specify the existing design. Ensure that the preferred language is Verilog, not VHDL.
2. Go to “Add Source” in ISE and select Coregen Core for each of the ILA and ICON cores.
3. Follow the guide mentioned above for using Coregen.
4. Ensure the cores were generated in Verilog by right-clicking on their entries in Project Navigator and selecting the option to view HDL source.
5. Use the HDL definition to connect the ILA to the ICON and instantiate both within the design.
6. Ensure the FPGA is on and connected to the machine.
7. Double-click on the Chipscope debug target, under Program Device.
8. Magic

9. Use Chipscope per the manual mentioned above.

Limits

The Chipscope user interface is pretty janky. Unfortunately you can’t name trigger signals in Chipscope, only data signals, and you can only rename them after programming the device – you can’t specify names in the code.