Lab 2: Integrated Logic Analyzer

Objective

During this lab, you will learn how the crazy awesome ILA tool can aid your debugging trials. I hope to convince you that this is a debugging tool that you should employ easily and often.

Theory

A logic analyzer is a standard digital design debugging tool. It allows the user to view a large selection of signals in a time-correlated fashion, with the signal waveforms captured and presented to the user. It works much like an oscilloscope, except it has many, many channels and the channels are digital. We have several Mixed Signal Oscilloscopes (MSO) in the lab for your use. You'll notice it has lots and lots of wires to connect to digital signals in your circuit.

Unfortunately, such a useful tool has problems when faced with FPGA debugging, primarily because of the limits of the chip packaging. When debugging an FPGA design, the signals you want to examine are, more often than not, buried inside the FPGA chip and have never been attached to one of the package pins. As the logic analyzer can only connect to the outside of the FPGA package at the pins, it doesn't have access to the internals of your circuit.

Enter ILA. By using the MarkDebug tools in Vivado, you can insert additional logic into your synthesized design. This logic will collect signal information from your design and store the actual signal values into RAM on the board-under-test. A connection back to the host computer, running the ILA Vivado tools, allows the user to view the signal information.

Procedures

By this point in the course, you should have some portions of your design working on your FPGA board. If not, build a simple FSM (or look ahead to Lab 3 and build an AC97 decoder).

1. Read through the ILA documentation from the Xilinx website for your version of Vivado.
2. Instrument several signals using ILA, view your system under test.
3. Experiment with ILA. Try different triggering options, match unit types, multiple triggers, storage qualification conditions, etc. You may not be able to do protocol violation monitoring, but you should definitely know about it for your future debugging work.

Report

Document what has occurred in a simple lab writeup, one per team. Describe the mechanisms you used to mark signals for debug, how you set trigger conditions (and how complex the triggers you chose to play with were), etc. Describe any limits you bumped up against, or ones that you imagine you might bump up against that would keep you from using ILA in particular situations.