F15 18-545 Lab 1 Report (09/16/2015)

Team M-x butterfly

Lab Summary:

For this Lab we mostly used the ILA to debug the behavior of the AVG system we were building, more specifically the behavior of the BRAM, VGA Controller and Rasterizer modules. Overall, we found it was relatively useful for observing how the signals behaved on the board without having to do complex testbenches to replicate certain bugs. At our current stage though, the code seems simple enough that the Simulator is more efficient. However we expect this to change as we begin extending our software.

One of the first ways we used the ILA was to observe the Hsync and Vsync components to the VGA, so as to monitor possible timing issues resulting in incorrect input. We set up a simple FSM to trigger on the Vsync going low, followed by the Hsync, then zoomed in 4000 samples left and right of the trigger to check that the timings matched up.

Going from here, we experimented with using it to check the behavior of certain states. There were some difficulties here because we did not quite understand the debug setup correct and had trouble trying to get state output. We ended up just tying a debug wire to the state which we used for the ILA as a workaround. Doing this early allowed us to realize that the BRAM switching code was not working correctly, as we created an FSM to flip and count the number of times the state flipped from using on BRAM to the next, and realized it did not change. We then changed it to trigger on the row/col signal being input at the point where it should have flipped, studied the other signals, and eventually found the bug (a port width mismatch leading to truncated values)

There were also some unusual timing bugs encountered during the initial integration of the Rasterizer with the BRAM module. More specifically, certain addresses were not being sent, while the color data for different lines was incorrect. Using the ILA to match specific values on the color_in and addr_in lines allowed us to check if those values were being propagated correctly and at the right time. This helped me notice a timing error where our ready_in signal was being asserted twice over four clock cycles when it should have only be asserted once, leading to duplicate color information and the skipping of an entire section of addresses. Some discussion over the states in our respective FSMs and the handshake was fixed.

We didn't do any protocol violation monitoring as our system was largely working at this point, but will definitely keep it in mind for future developments. One of the major limits we found about the ILA was the sheer time needed to use it – since it operates on the board itself, every time we made a change, we needed to re-synthesize, re-map the XDC, implement and program the board. As the project grows larger, it'll only take longer. For future uses, we will likely use the simulator to check if our bug fix worked in an isolated environment (by forcing the clock/constraints environment we glean from the initial ILA test), then use the ILA to confirm our fix worked. The ILA also has some limitations on which variables it can view – this might be more due to our lack of experience than a functional limitation, something we will explore more carefully this week.