California State University, Northridge
ECE 520L - System on Chip
Fall 2022

Lab 6: Create a Custom IP

October 16th, 2022
Instructor: Saba Janamian

Troy Israel

Introduction:

The purpose of this lab is to learn how to create a custom AXI IP block in Vivado Design Suite and create firmware in Vitis that we can add into our custom IP. Since we are creating our own block design we will also learn some more TCL generation that will make sure that if someone wants to recreate our block design it will be recreated in all its entirety. So far the process for which we do our TCL generation in the previous labs will not be enough to recreate our entire design.

Procedure:

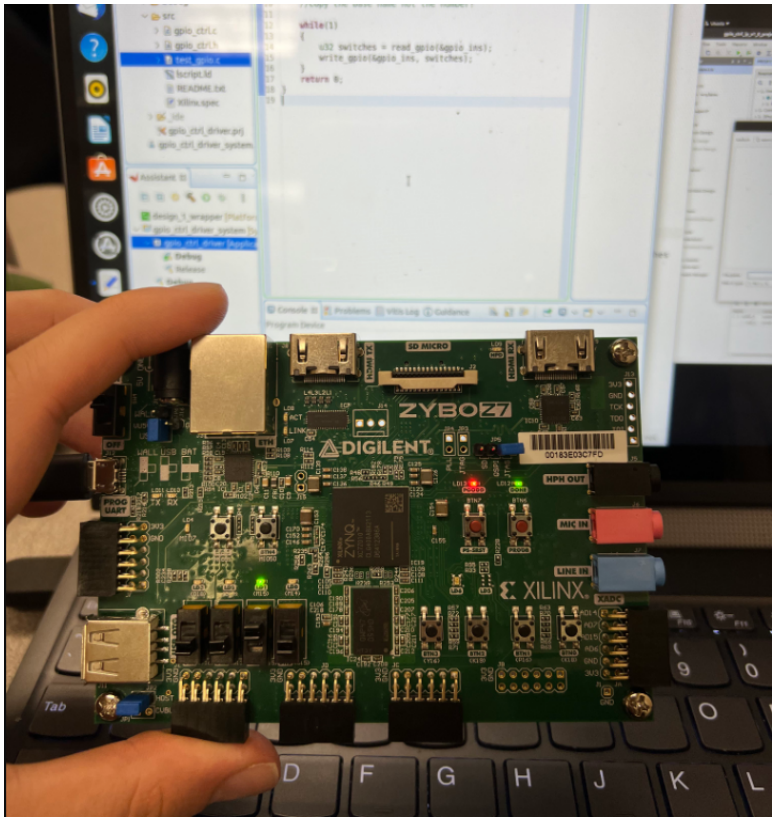
First, we begin the design of this lab in Vivado. We create the custom IP by going inside the tools menu and choosing “Create and Package New IP”. Next we choose the option to create a new AXI4 peripheral. We set it as a AXI Lite and the interface mode to Slave and we then go directly to editing the IP. When you check the pathing inside the linux terminal for the project name and inside the ip_repo file that is created, there is a group of files that are generated when editing the IP. When we close the ip editor, those files will disappear. Only the 1.0 version will remain.

Inside the design sources, we will be editing the Verilog modules that were created when we first started up the IP editor for the AXI4 peripheral. We are going to be doing port mapping in Verilog and adjusting the slv_reg files that were generated. The write to slave register is one of the things that we will editing. We want the values of the LEDs to be written into the slave. Whatever wires that we create in the top module we also need to create them in the control module and vice versa. And we need to make sure that we do the port mapping for any new wires or signals that are made. Once we make our changes to the Verilog code we shall make our changes in the “Customization GUI” window.

After working on the Verilog files we can start building the block design by first adding the Zynq processing system and the custom IP block we made earlier. We then make the necessary external ports for the design after we run the block automation. Now we can do the HDL wrapper for the design and the port pin mappings. After the pins have been mapped correctly we can generate the bitstream and launch Vitis

In Vitis we will first design a header file that will contain all our function prototypes. We then create a C file that will contain each of the function parameters and their makeup. Then we will make a test C file that will allow us to test the switches and LEDs on the Zybo Z7. After we create all the C files we will then transfer them over to our Vivado project. In this case we are trying to implement firmware and drivers into our overall design.

Results:



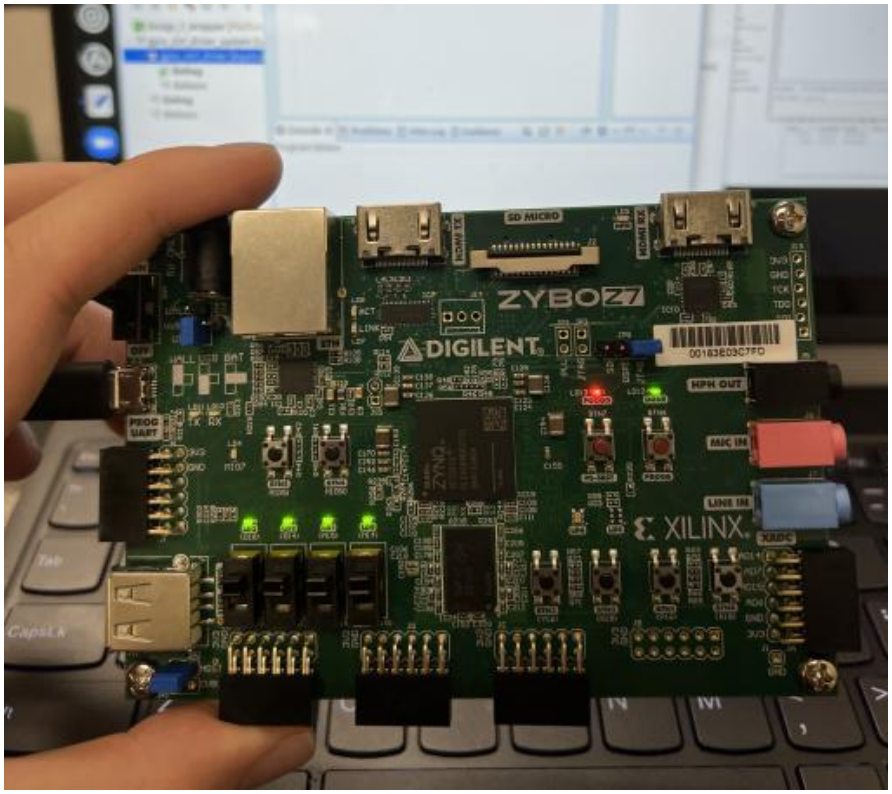


Fig 6.1 Verifying that the switches work

In the above two screenshots I was able to verify that switches and leds are working properly. It is able to work with 16 different switch combinations. In this case I showed that turning all the switches to 1 will turn on each individual led. Again the main purpose of this lab was to showcase various stages of a custom IP from the block design, implementing firmware and drivers and then to test it on the board. Obviously from the above screenshots I am showcasing that the design software works.

Conclusion:

In retrospect this was a very important lab because it taught me how to create my own IP modules. It also showed me that I can even modify my design even further by editing the generated HDL code that Vivado creates. There was also the important showcase of how we can even parameterize our designs and make it more user friendly. Instead of having a hard coded input size we can design so that we can make the design accept a variety of input from a defined range. It was also interesting to see how we can take code written in Vitis and actually add it to the Vivado project. This lab showcased how we can not only create the block design but also how to develop firmware and drivers.

Appendix:

Github: https://github.com/csun-ece/fa22-e520-lab6-troykerim

Youtube: https://youtube.com/shorts/Zu24YGW1Mfs?feature=share