Embedded Systems 2

Lab 3

Alec Bakholdin

Submitted 10/20/2021

Github: https://github.com/embedded-systems-2-fall-2020-labs/lab-3-submission-Alec-Bakholdin-Rutgers

**Purpose**

The purpose of this lab was to introduce us to using external or custom IP in Vivado designs and then exposing us to more complicated hardware setups which would require more complicated debugging than we'd previously done. It also held our hand through reading some more complex C code and required us to try a number of different solutions when attempting to get the hardware to perform as we would expect it to.

**Theory of Operation**

The expectation was that the board would be controlled from the command line of the PC it was connected to. More specifically, 'q' would open the main menu, then there were three options that would keep playing until you returned to the main menu with 'q', at which point the hardware would stop. 's' would play whatever audio was fed to it, 'n' would add noise (a high-pitched sound in the background of the audio) based on which switches were flipped at the time, and 'f' would both add the noise then allow the user to filter the noise out when any of the buttons were pressed.

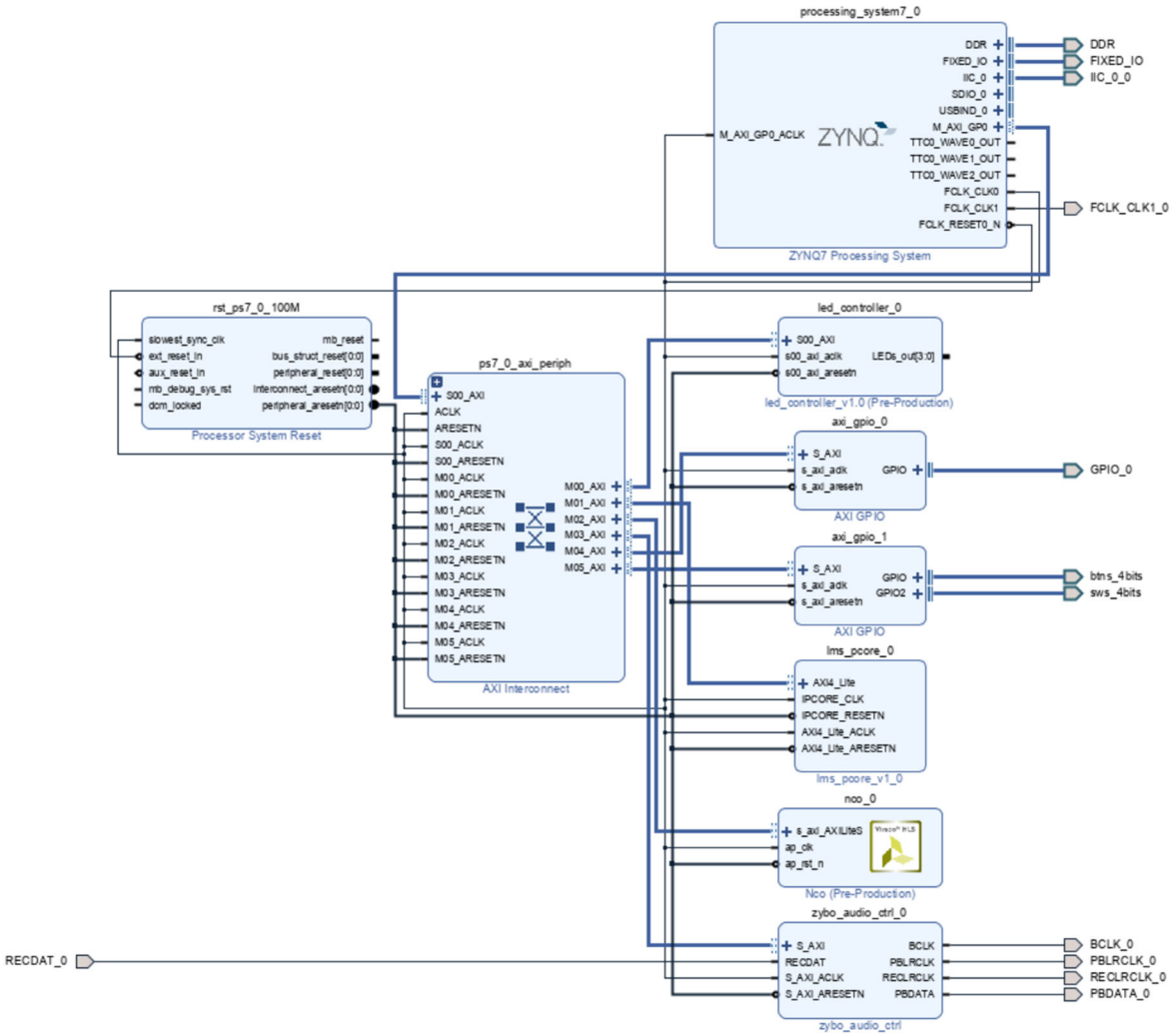


**Figure 1. Block Diagram**

**Design**

The only changes made to the C code were adding extern to the 4 variables at the bottom of adventures_with_ip.h and adding those same variables as global variables in adventures_with_ip.c. All C code is located under “Source Code” in the repo that was submitted above.

**Test**

To test my implementation, I used an AUX cable to connect the output of the computer I was using to the input of the Zybo board, then hooked up a speaker to the output of the board. Then, I connected the Vitis console to the serial output of the device and ran through the options that were described earlier. I hit ‘s’, then played music to make sure it output sound through the speaker. Then I hit ‘q’ followed by ‘n’, which allowed me to introduce noise using the switches. Once I heard the telltale high-pitched noise in the background, I pressed ‘q’ and ‘f’. The high-pitched noise remained until I was holding any of the buttons down, at which point it disappeared.

**Implementation**

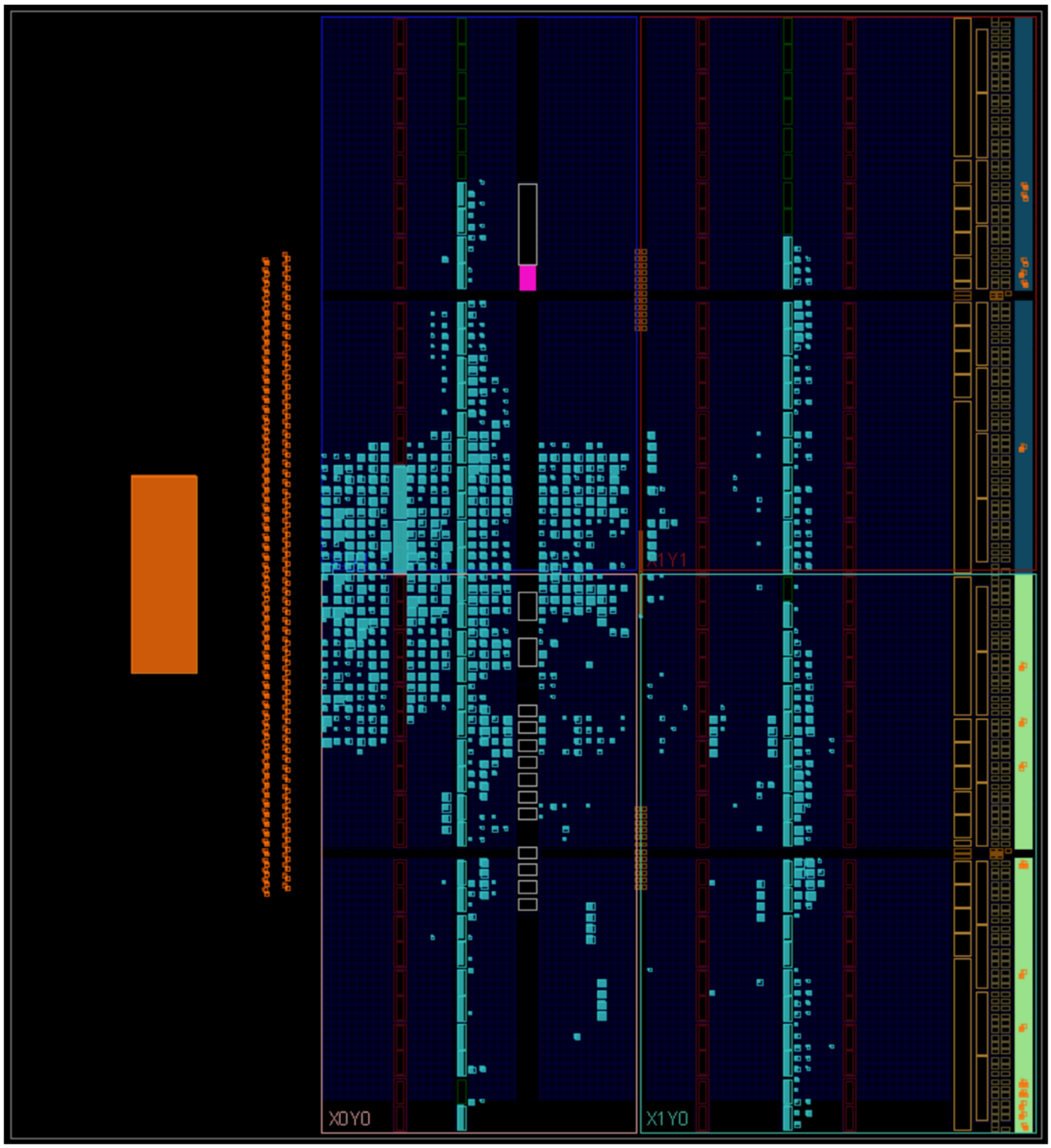


**Figure 2. Implemented Design**

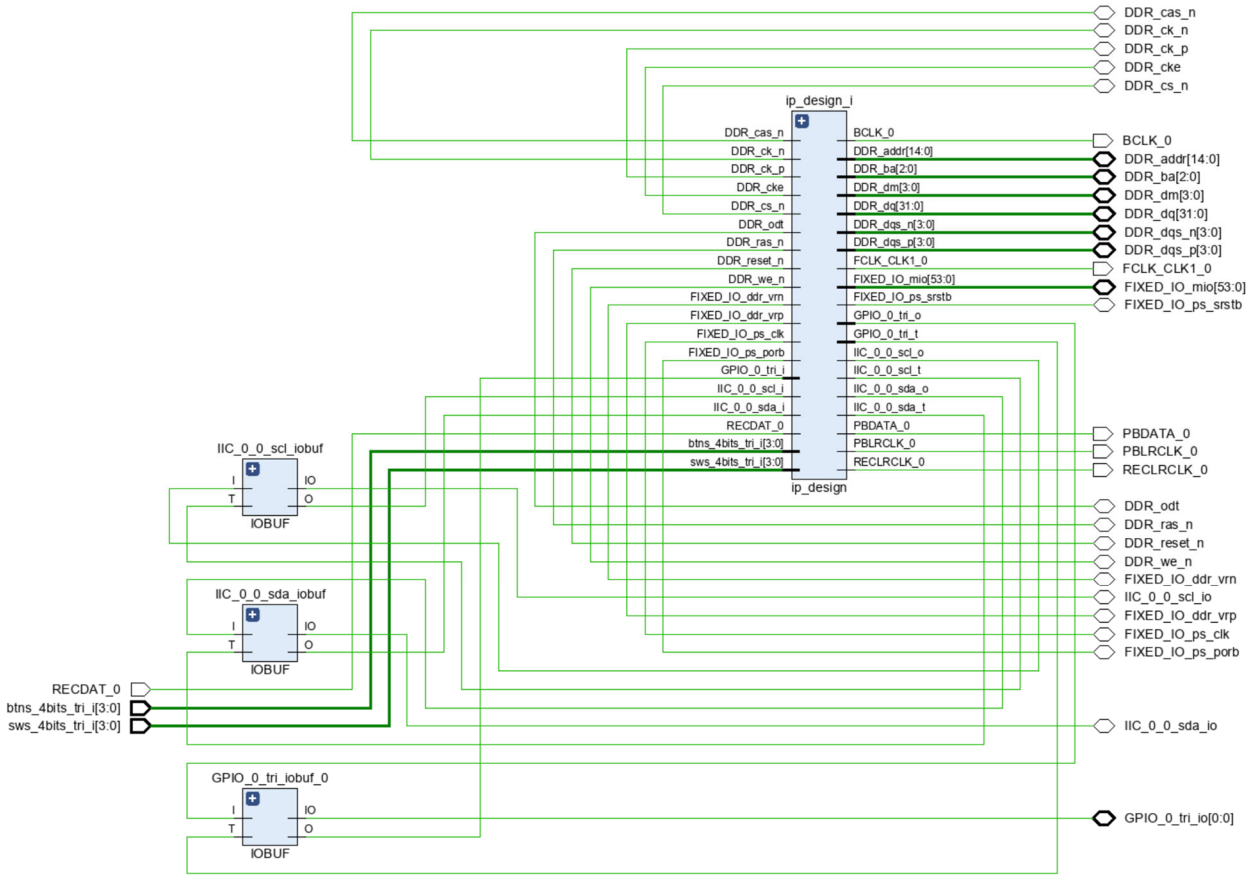


**Figure 3. Elaboration Schematic**

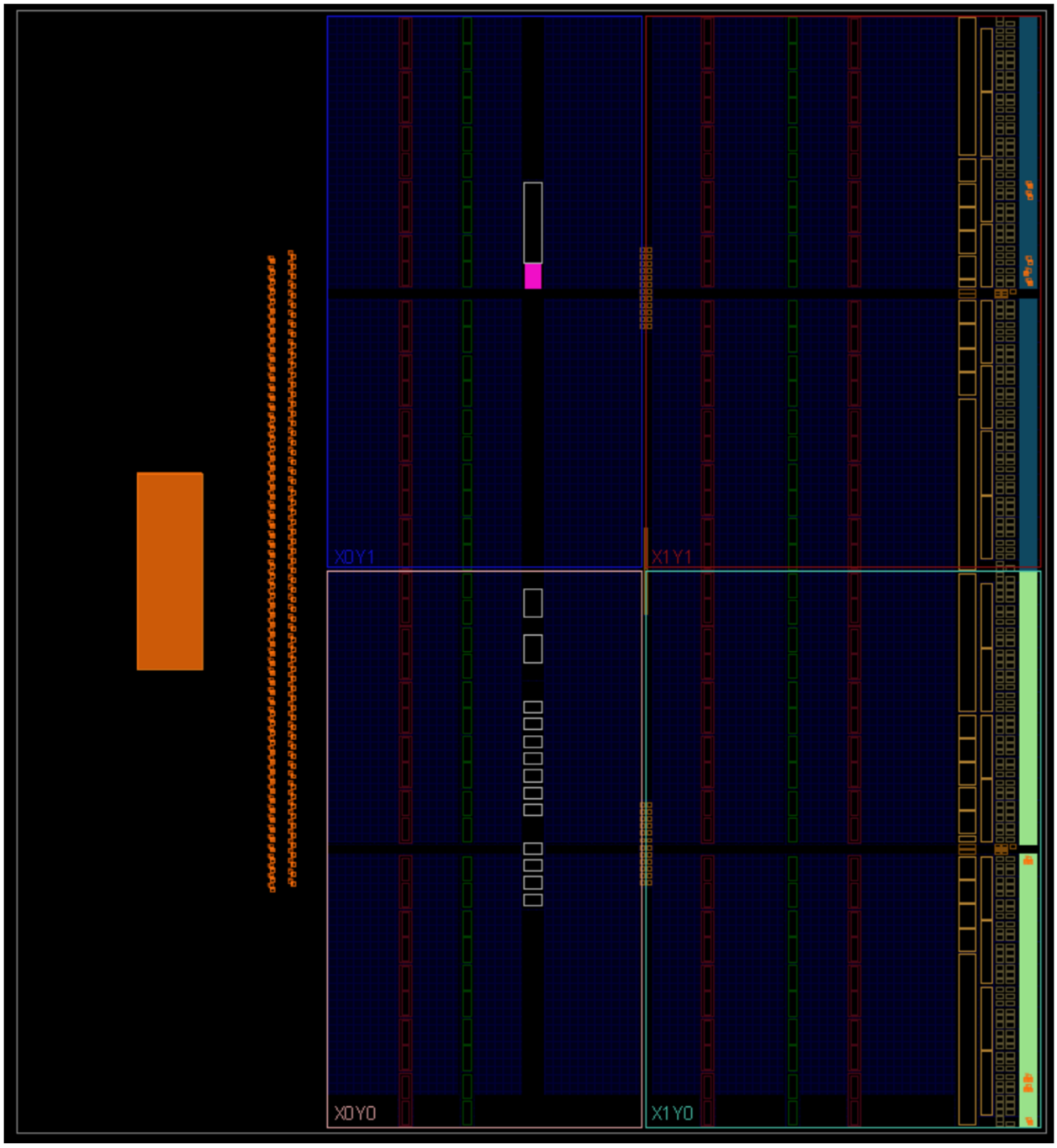


**Figure 4. Synthesis Schematic**

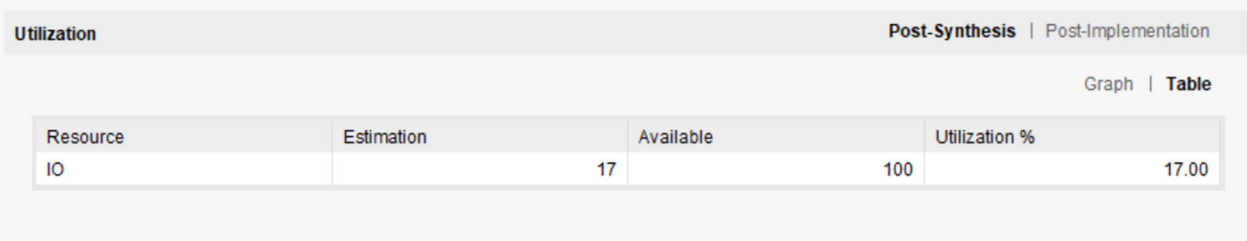


**Figure 5. Post-Synthesis Utilization Table**

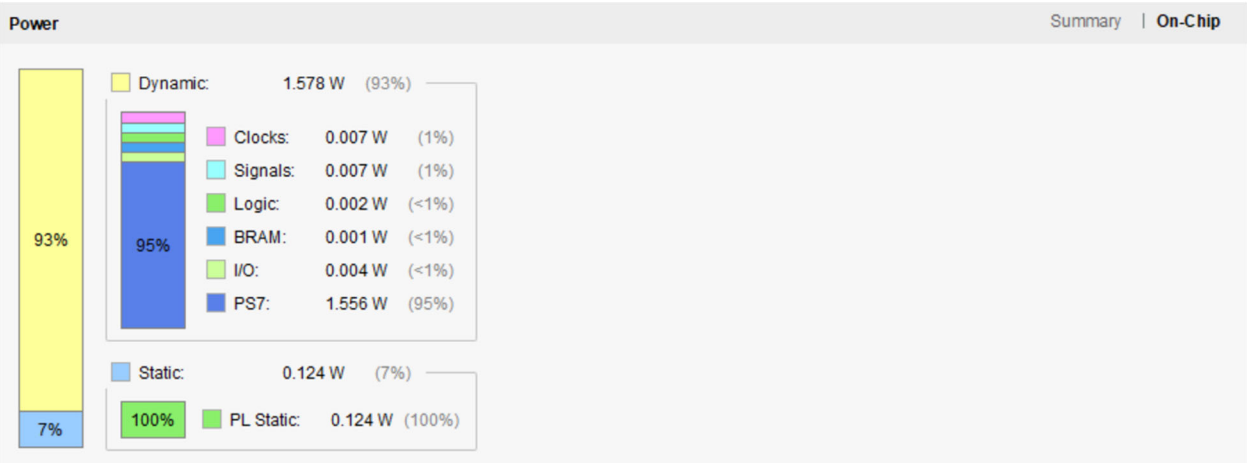


**Figure 6. On-chip Power Graph**

**Discussion**

Just like in any lab, I gained experience with the tools we'll be using: Vivado and Vitis, which is the overall goal. More specifically, I gained a better understanding of how to interact with the hardware through C code and gained more experience in creating more complex block diagrams using IP from sources beyond the included packages.

I feel that I have a very solid grasp of how Vivado works now at a very basic level, and how it assists in generating hardware packages for Vitis. While I can read the code that was provided for me, I still don't feel wholly confident when I think about actually *writing* code for the hardware that was generated in Vivado.