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

Lab 3: Using UART and GPIO in ZYNQ

September 20, 2022
Instructor: Saba Janamian

Troy Israel

Introduction

In this experiment we are creating a block diagram that will include the ZYNQ7 processing system to provide board support while we program the ZYBO Z7 FPGA board. We will create IP block diagrams that will be integrated together used on the board. We will be setting parameters inside the processing system to suit the tasks that will be done. Lastly we will create a Vitis IDE project that will utilize our block design and later on our Vitis project will be programmed on to the FPGA board.

Procedure

For this experiment we will be using documentation from Xilinx and following their steps for creating the project. This experiment is broken down into two parts. The first part is working directly with the Vivado software. We start by creating a block design from the IP list, specifically the ZYNQ7 processing system. We then edit the parameters inside the processing system block. We turn off everything inside the I/O peripherals except for the UART 1. Next we modify the clock configuration and adjust the clock frequency to 100 MHz. Then we begin to add more IP blocks to the overall design first with an AXI GPIO. With this IP block we create an external port connect and have it set to 4 switches. Then we add another AXI GPIO block and adjust the settings so that it is set for 4 buttons.

Now we add the Processor System Reset as well as the AXI interconnect. We start connecting all of the input ports to these IP blocks together using the connection automation. All the resets and clocks get connected together and are directly connected to the AXI interconnect. The AXI interconnect is acting as a Master for the GPIOs while at the same time a slave for the Processing system. Once all the ports that we need to use from the Xilinx documentation have been connected, we can then validate the design, create the HDL wrapper file and use the run implementation tool. Finally we can generate bitstream once the previous steps are finished.

The second part of the experiment is in the Vitis IDE. Here we will create a project that will use an exported XSA file made in vivado. The XSA file contains the board support that we will use for the Z7 board soon. Inside Vitis, we will create a C application where we can set the push buttons and the switches to be used. Basically if we flip a switch or press a button the console should have a message pop up and the board's LEDs should light up.

Results:

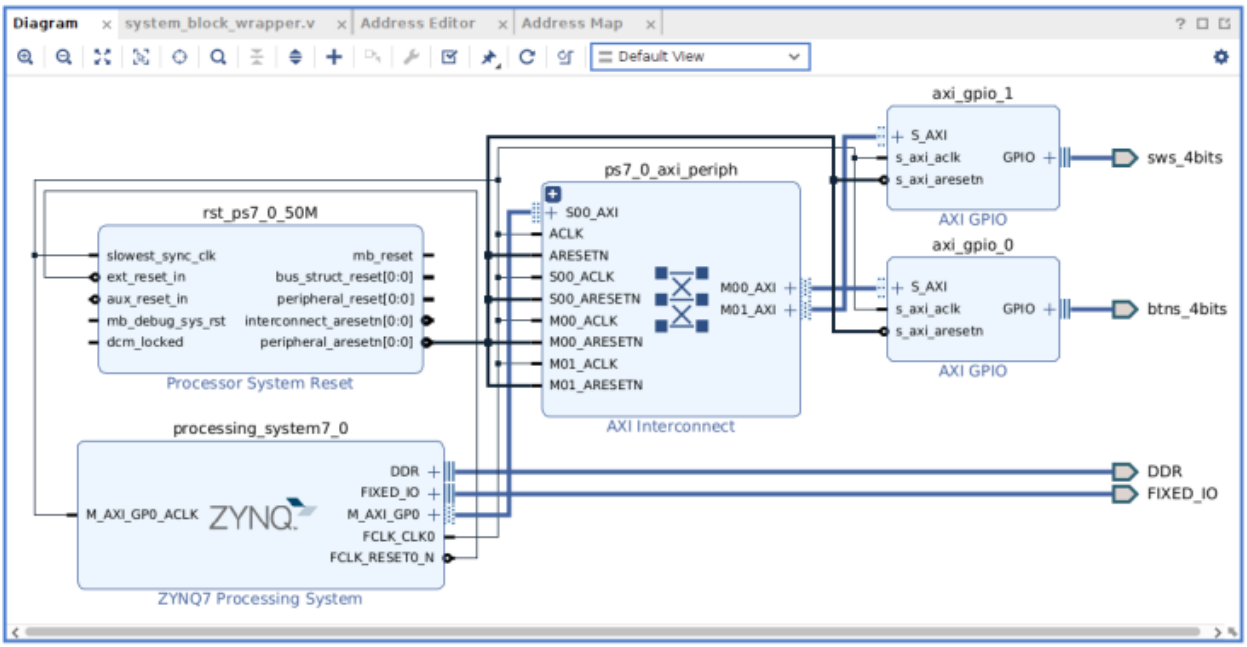

Fig 3.1 Overall IP Design

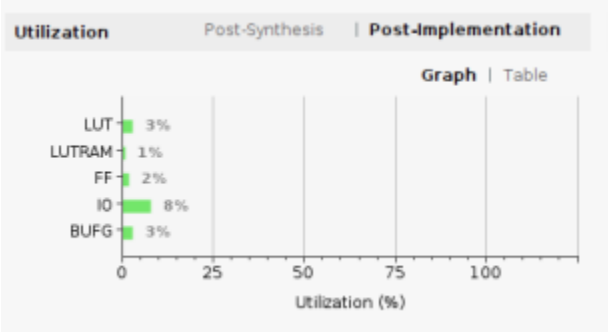

Fig 3.2 Utilization

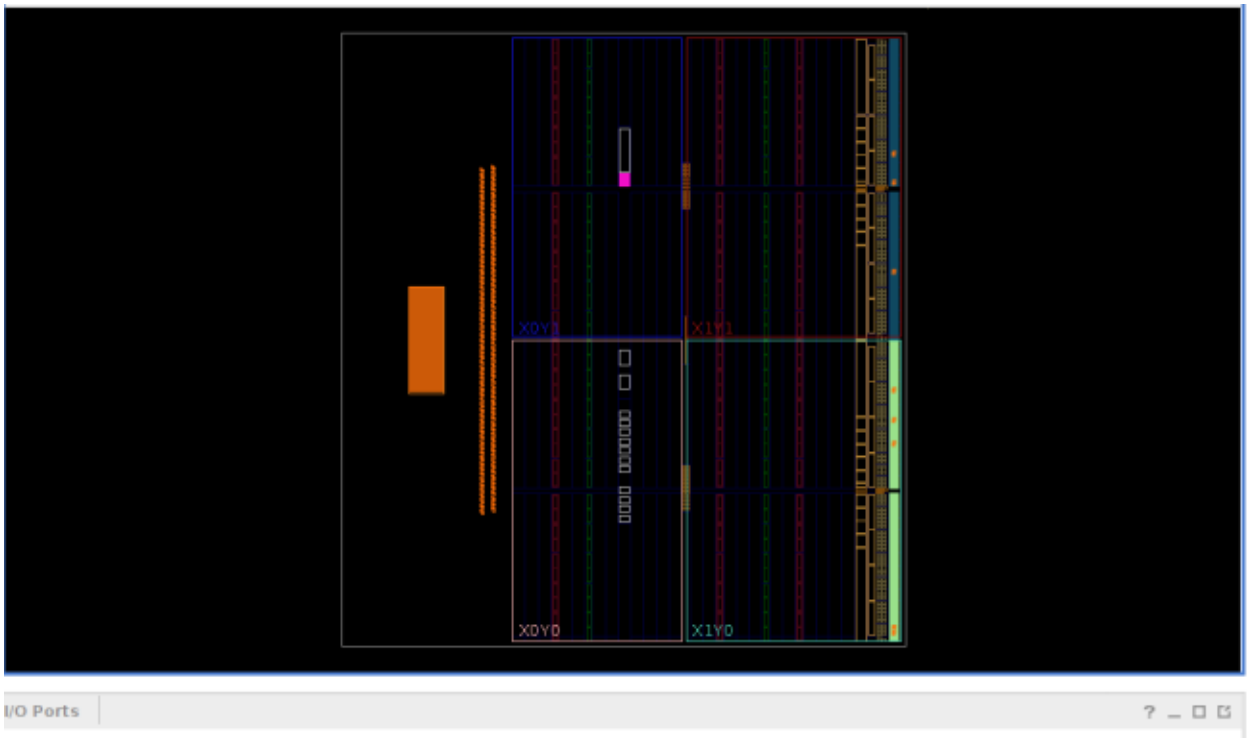

Fig 3.3 Device Mapping

For Vitis results see the unlisted youtube video:
https://www.youtube.com/watch?v=Hj3w_Sm5XKA

Conclusion:

In this lab we learned how to set up an IP block diagram in which would create board support that we would later on use in Vitis. We also learned more about writing C code to interact with the hardware, mainly on a Zybo Z7 board. While the Zybo Z7 board would get programmed from Vitis however there was something not quite right. While I checked to make sure that the C code was correct and it was built, there is a chance that the XGpio parameters were not correct. We did have to make an adjustment from the code that was given to us during class. There is a chance that inside the actual block design there could be a slight problem with it.

Appendix:
https://github.com/csun-ece/fa22-e520-lab3-troykerim