Embedded System Zynq PL and PS Project

Leonardo Aviles, *tue51778@temple.edu*

***Abstract*—This project was a proposed to Dr. Silage and approved with some changes. The goal of this project to use the Maxim Proximity Sensor in order to change which LEDs will be turned on and off. The original project was proposed as a PL project at first but using both PL and PS the project was attempted.**

I. INTRODUCTION

In this lab the goal was using the Proximity Sensor in order to change the LEDs. In order to do this different intervals are used. An example of this is if there is a light source if the light source is at its brightest LED0 will turn on and as the light source begins to dim the next bits will turn on in different intervals. Once the light is either completely gone or dim all the LEDs will be turned on.

The specifications of the project are as follows.

- Using Vivado studio a block diagram will be created. The IP blocks used were the Zynq,axi_timer,rst_processing system, and processing system_axi periph.
- The LEDs would increment depending on the value of the proximity reading.

To complete the project, code needs to be manipulated in the following way.

- The example that was given to the class was used and manipulated in order to fit the needs of the lab.
- A axi was added for LED_Data and the variables for address of the proximity were changed.

II. DISCUSSION

A. Lab9.c: "Using a Proximity Sensor to change LEDs"

In order to achieve this goal the addresses of Proximity, Infrared ALS photodiode, and green ALS photodiode were needed to activate the devices and if needed the interrupts needed. In order to find the ambient light sensing the infrared ALS signal must be subtracted from the green ALS after the respective gains are applied.

The device ID was found using the XPAR_PS7_I2C_0_DEVICE_ID and defined in the variable Prox_Device_Id. The ALS upper and lower thresholds are set. These values are 16 bits each with the values being split into 2 high bytes and 2 low bytes. The interrupt status address and the interrupt status was found and turned on. The main configuration was set to a 0x33. In order to complete this lab 3 different functions were found from the PMODs manufacturers website. These functions were set_prox_mode,set_als_mode, and readProx. In this program the data was read in a constant loop.

Once all the code was written it was found that there was a error that would not allow the program to be written on the board. An error was found which stated a function was not found or valid. The first iteration of the project found the example given to the class with the different variables and functions changed. This was the project where Maxim header files were included and the sample code given from Maxim was used. The second iteration of the project was using the example from the blog emailed. In this project the block diagram was created and I2C data paths were used. For this project the block diagram was followed but when the address of the constraint files were created Vivado would produce an error stating address were not valid. A way that this could have been changed was to manipulate the address in Vivado in order to match with the JE Pmod interface in the Zynq board.

III. CONCLUSION

The project was not successfully implemented with many errors and problems occurring. A cause of these problems was trying to create an I2C system with little experience using this type of communication. At first a PL program was going to be created but it was unknown how to implement the addresses on the PL side of things. Another problem was trying to create actual I2C communication protocol.

IV. APPENDIX

Figure 1: Shows the block diagram created for the lab.