Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

3C7 – Lab D

Student Name:	**Aaron Dinesh**
Student ID Number:	**20332661**
Board Number	**8**
Assessment Title:	**Lab D**
Lecturer (s):	**Shreejith Shanker**
Date Submitted	28/02/23

I hereby declare that this assessment submission is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I hereby declare that I have not shared any part of this submission with any other student or person.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at: http://www.tcd.ie/calendar

I have also completed the Online Tutorial on avoiding plagiarism 'Ready, Steady, Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write

I am aware that the module coordinator reserves the right to submit my exam to Turnitin and may follow up with further actions if required should I be found to have breached College policy on plagiarism.

Signed: 

Date: 28/02/23

Introduction

Lab D was the first introduction into the synthesis and constraint tools that Vivado provides us. These allow us to turn our Verilog code into designs that can then be programmed onto an FPGA. In this lab, we flashed our programs onto a Basys 3 board.

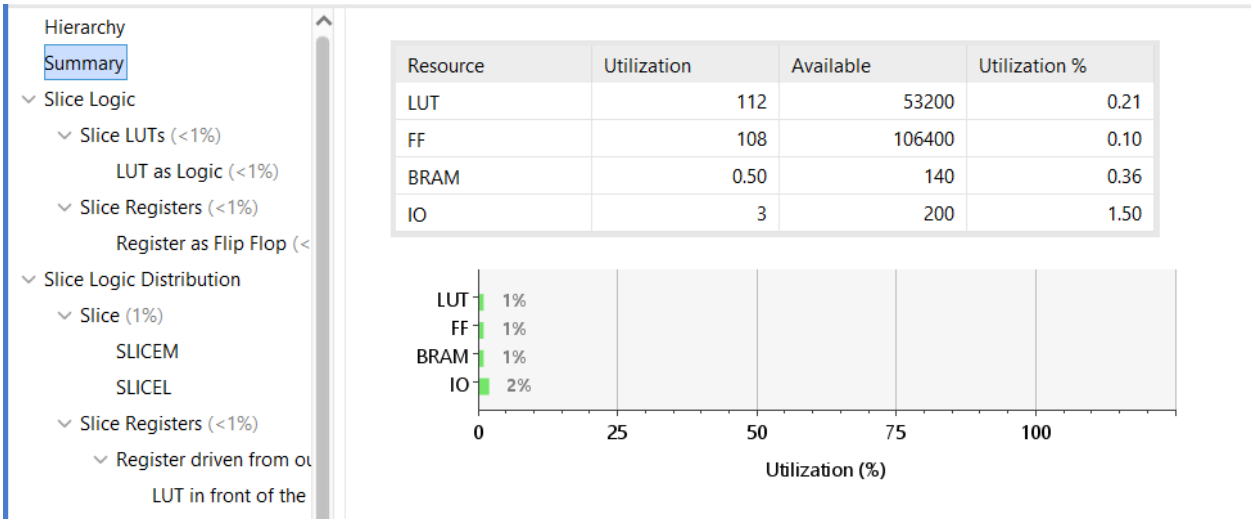
Experiments and Observations

The lab was split into two parts, the first part allowed us to look at the synthesis and implementation tools that were available in vivado and the second part allowed us to actually implement a design on the Basys 3 board. As such the Experiments and Observations section will be split into two sections, one talking about the tutorial, and the other talking about the Basys 3 implementation.

Tutorial

The tutorial required us to create a new Vivado project called “modulator” and load the files contained in the Tutorial Code folder as sources. There was an error in the “pwm_rtl.vhd” file which prevented the design from being synthesised. This was due to a semicolon in the wrong place on line 12. Once this was removed the design was synthesized easily and the rest of the tutorial of Lab D.

Once the design was synthesized, we could generate a utilization report which is shown below:

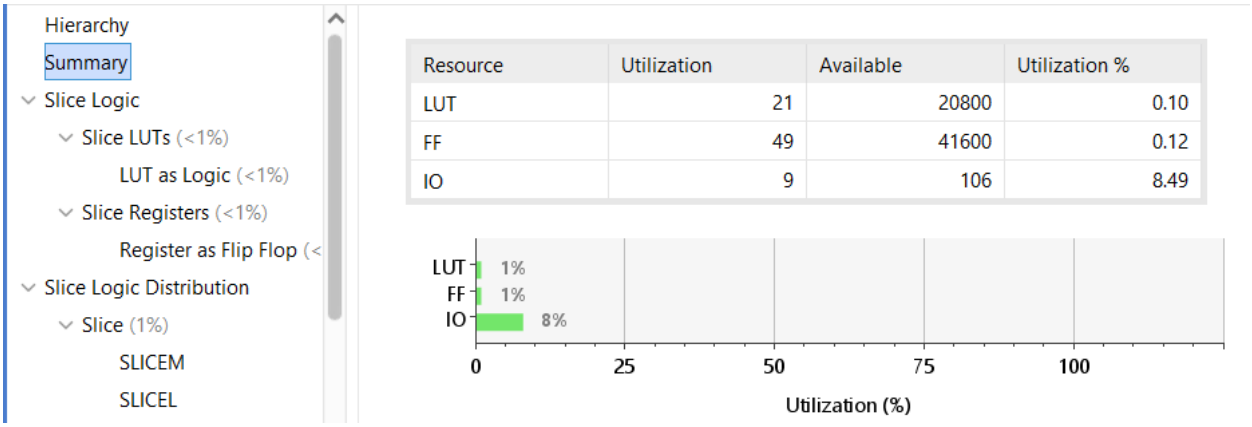


This utilization report lists the Look Up Tables, Flip-Flops, BRAM and Inputs and Outputs used on the FPGA. The BRAM in this case was used to store the values for the sine wave. The 3 IO ports map to the clk_p (input), pwm_out (output) and sw0 (output). The IO would have been 4 if the clk_n (input) was being used, however this was left unused.

Basys 3 Implementation

The rest of the lab required us to interface with and program our Basys 3 boards with the bargraph test code given to us. Once the code was uploaded the LEDs along the bottom began flashing according to the uploaded code. After analysing the constraint files and the sources file, I deduced

that the LEDs were lighting up according to whether or not a certain bit of the data register was set. Ie. If the bit in register position 2 was set (corresponding to a number > 2^2) the LEDs up to position 2 was turned on; that is LEDs LD0, LD1 and LD2. After analysing the clock code, I determined that the clock was running at $\frac{5\times10^6}{625\times10^3} = 80\text{Hz}$. Where 625×10^3 was a clock prescaler dividing the 50MHz clock rate to the 80Hz rate that we used for our design. Attached below is the utilization report for the bargraphtest design:



The LUT useage was mainly down to the comparasion operators used in the clock code and in the bargraph.v code. The FF useage is for the registers that get used in the source files to store values eg clkq register in clock.v. The IO pins are used for the 8 LEDs (LD0 to LD7) and the 50MHz clock input from the crystal oscillator on the board.

The lab handout also asked us to set the switches to the binary equivalent of our board number. Given that my board number was 8, I set the SW3 switch, however setting this switch made no difference to the code running on the board. Upon inspecting the code, this theory was confirmed as there was no use of the switches in the constraints file or in the code.

Conclusion

The main takeaways from this lab were getting comfortable with the synthesis and implementation tools that were provided in Vivado. Also, it allowed me to learn how to use the constraints wizard to set the inputs and outputs correctly. Aside from this I also got to look at how we can use clocks and always block in synchronous designs and how we can use the main clock and a clock prescaler to derive any clock frequencies that we might need for our design.