Lab #1

Introduction to Quartus Schematic Capture, ModelSim Simulation, and the Intel DE-10 Lite Board

Lab Report

Bradley Manzo
916953788

TABLE OF CONTENTS

INTRODUCTION

OBJECTIVES

PART I

- Learn how to initialize the DE10-LITE Board and connect it with your PC, learn how to create a schematic using the Quartus Schematic Capture tool, Learn how to configure and generate an assignment for the board, and learn how to upload to the board.

PART II

- Implement a 5.0 MHz Clock into the schematic and upload it to the DE10-LITE Board to create a pseudo-random number generator.

PART III

- Learn how to simulate the circuit using ModelSim-Altera, and how to view the signals in the Wave Window over a given interval.

PROCEDURE

DESIGN AND TEST PROCEDURE

PART I

- Using the provided instructions, I created schematic lab1b and after generating an assignment for my DE10-LITE Board, then I uploaded said schematic to the board. I then demonstrated the functionality of the system to a TA, showing that the value incremented by one with each press of the button.

PART II

- Using the provided instructions I added to my previous schematic and called the new version lab1b. This circuit includes a 5.0 MHZ clock and uses the GPIO, CLOCK, and SW signals that were initialized in Part I. I then uploaded the schematic to my DE10-LITE board and demonstrated the functionality to my TA, showing that flipping the switch essentially generates a random value.

PART III

- Using the provided instructions I uploaded my schematic to the ModelSim-Altera Program and with the help of my TA, I plotted several important signals. I then generated two unique waveforms by forcing st1 and then st0. I shared my screen with the TA and verified the accuracy of my wave.

ANALYSIS/RESULTS

DATA

PART III:

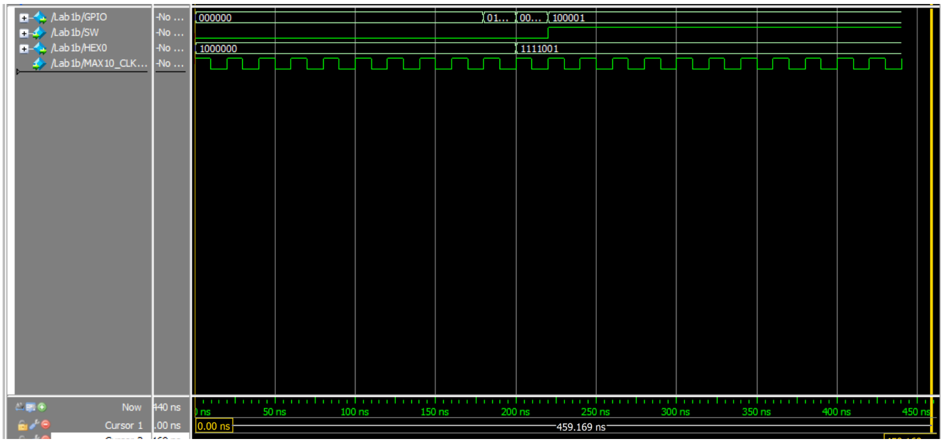

Figure 1: Forcing SW to value 1

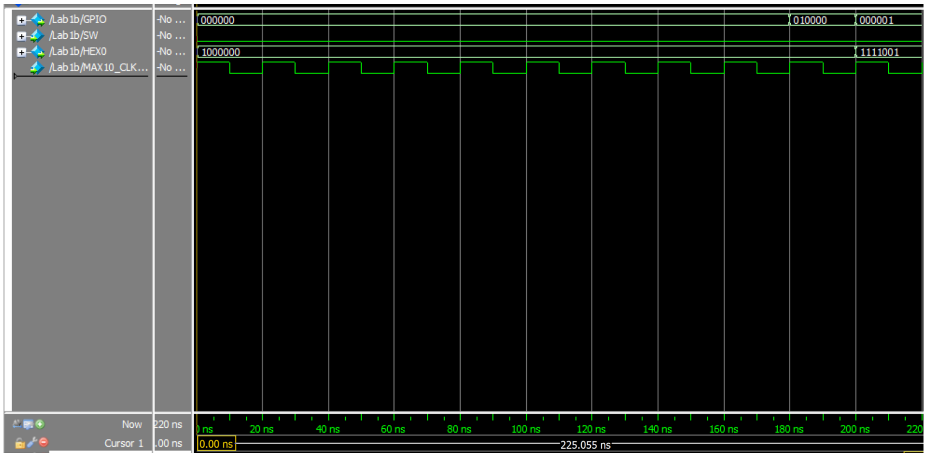

Figure 2: Forcing SW to value 0

ANALYZING RESULTS

PART I

- After completing Part I, it has become clear how exactly to create and configure a new project in Quartus, generate a pin assignment using System Builder, and how to connect and flash the DE10-Lite Board with my PC. The provided schematic makes sense in the context of the board's behavior, as the input of KEY[0] increments HEX[0], just as the button increments the number on the board. The result was just as expected.

PART II

- After completing Part II, I have a greater understanding of the pins on the chips we are using in the schematic, and a greater understanding of how frequency and period can have a drastic effect on the performance of a system. This part helped me reinforce the lessons of the previous part, and the result was just as expected.

PART III

- After completing Part III, I've learned how to simulate the behavior of a system using ModelSim, and how to spot similarities between the physical and virtual systems. It is clear that forcing SW to 0 or 1 is akin to flipping the physical SW[0] switch on the board. Forcing SW to 1 ceases the board's background GPIO and HEX signals, essentially pausing the program on a single number. When SW is forced to 0, these processes continue seemingly at random intervals, however this is most likely due to the chosen 220 ns interval being out of phase with the period of the GPIO and HEX[0] signals.

ANSWERS TO HIGHLIGHTED QUESTIONS

- Does each waveform match the behavior of the corresponding signal when the circuit operates on the DE10-LITE Board?
    - Yes it does. HEX0 represents the display value, GPIO represents input/output, the clock is the 50 MHz clock, and SW1 is the switch. When the switch is off at 0, the GPIO and HEX0 change values every 220 ns cycle. However when SW1 is forced to on at 1, internal processes cease, essentially pausing the program.

CONCLUSION

- After completing Lab 1, I feel I have developed an understanding of the process of creating a new project, generating pin assignments, constructing schematics, connecting to the device, compiling and uploading the circuit; as well as projecting said schematic to ModelSim and plotting specific signals on the wave window. Additionally, I've learned how to spot similarities between the physical and virtual forms of the schematic, and developed a greater understanding of how logic gates can be implemented in a realistic fashion. This lab was rather straightforward and despite having troubles with drivers and installing the incorrect version of Quartus, I was able to reach the desired result. Due to the nature of this lab, I wasn't entirely to design my own solutions and therefore I dont have any way to improve upon them. However, the next lab will no doubt go much smoother now that I've solved most if not all of my technical issues.