

Lab 1: Utility Board Assembly

Peter Johnson¹

E155 Fall 2018 Section 2, September 12, 2018

Abstract— This paper describes the, assembly, programming and testing of the MuddPi Mark IV such that it can control its on-board LEDs and display hexadecimal numbers on an off-board 7-segment display based on inputs from switches.

I. INTRODUCTION

In this lab, the MuddPi Mark IV utility board was assembled. This board includes an Altera Cyclone IV FPGA, Flash configuration memory, an array of LEDs, switches, an oscillator as the clock, H bridges and an ADC. The assembly was done through a combination of regular soldering as well as surface mount soldering using solder paste and a heat gun. Once assembly was complete, SystemVerilog was written to control the onboard LEDs as well as a 7-segment display which would display the hexadecimal number conveyed by the four switches.

II. DESIGN AND TESTING METHODOLOGY

While assembling the board, I tried to be very systematic in my testing, checking the resistance between Vin and GND each time I added a new component. I also checked for connectivity between joints when appropriate. Once assembled, I approached the design of this system by first considering the requirements of the onboard LEDs. The logic was simple only requiring assign statements and not gates. For the flashing LED I referred to the E85 Ch.5 lecture slides which went over a counter used to toggle the most significant bit of a variable at a desired rate. Given the relationship:

$$f_{out} = f_{clk} * p / 2^N$$

As well as the know values: the desired frequency was 2.4 Hz, Clock was 40 MHz and word size was 32 bits I then solved for p:

$$2.4 \text{ Hz} = 40 \text{ MHz} * p / 2^{32}$$
$$p \approx 258$$

This counter went into its own module since it seemed appropriate to reduce the complexity at my top level module. Next I simulated all of the LED logic on Modelsim. I just forced each of the values of the switches and checked their simulated outputs. The blinking LED I left for testing on the hardware since this would be simpler.

The next part of the design was the 7-segment display. I used a HDSP-5721 common anode 7-segment display. I started with the analog requirements to turn on each of the LED diodes. To turn on a diode segment, a 0 not a 1 needed to be passed to the pin of that segment since the pins were the cathodes. By passing a 0 to the cathode, the cathode would be pulled to 0 producing a voltage drop across the diode resulting in it producing light. The desired current flowing through an LED is roughly 5-20mA. The drop across an LED

is roughly 1.7-2.7 V. This allowed me to solve for an appropriate resistor using Ohm's Law:

$$V=IR$$

I chose I=5 mA and V=3.3-1.7V, so R=320Ω. There were 390Ω resistors available and they work so I used them instead. They are a little big though so the LED is dim. Once this was set up, I wrote the SystemVerilog to implement the display. I used a combinational logic set of cases to implement the truth table. Initially I wrote out the table using 1's as turn on a segment, but realized I would need extra hardware to not all of the 1's so I swapped the 1's and 0's. I then programmed the FPGA via JTAG thinking it was quicker to test by actually flipping the switches. I then went through all 16 positions ensuring the correct value was displayed.

III. CODE AND SCHEMATICS

Refer to attached schematics and code.

IV. RESULTS AND DISCUSSIONS

All of the prescribed tasks were accomplished: the LEDs respond appropriately to the switches, the 7-segment display correctly displays hexadecimal values from the switches and when reset, the board is able to return to the program since PROM was correctly programmed. One small potential issue that did arise was that I slightly melted the housing of the reset pushbutton when using a heat gun. I do not believe the internals were damaged since it is still able to correctly reset the board, but I would be more concerned about it losing functionality at some point compared to a non melted pushbutton.

V. CONCLUSIONS

The lab was successfully completed. The Mudd Pi Mark IV was fully assembled and tested. The on-board LEDs turn on in the correct patterns, and the 7-segment display is able to correctly display hexadecimal values based on inputs from the switches. Once the board is rest or turned on and off, the LEDs continue to function, showing the PROM was successfully programed. I gained practice surface mount soldering, analog circuit design and brushed up on my SystemVerilog. In total I spent 6 hours, mainly on the soldering but it wasn't bad.

¹Peter Johnson is with Department of Engineering Harvey Mudd College, Claremont, CA 91711 pjohnson@hmc.edu