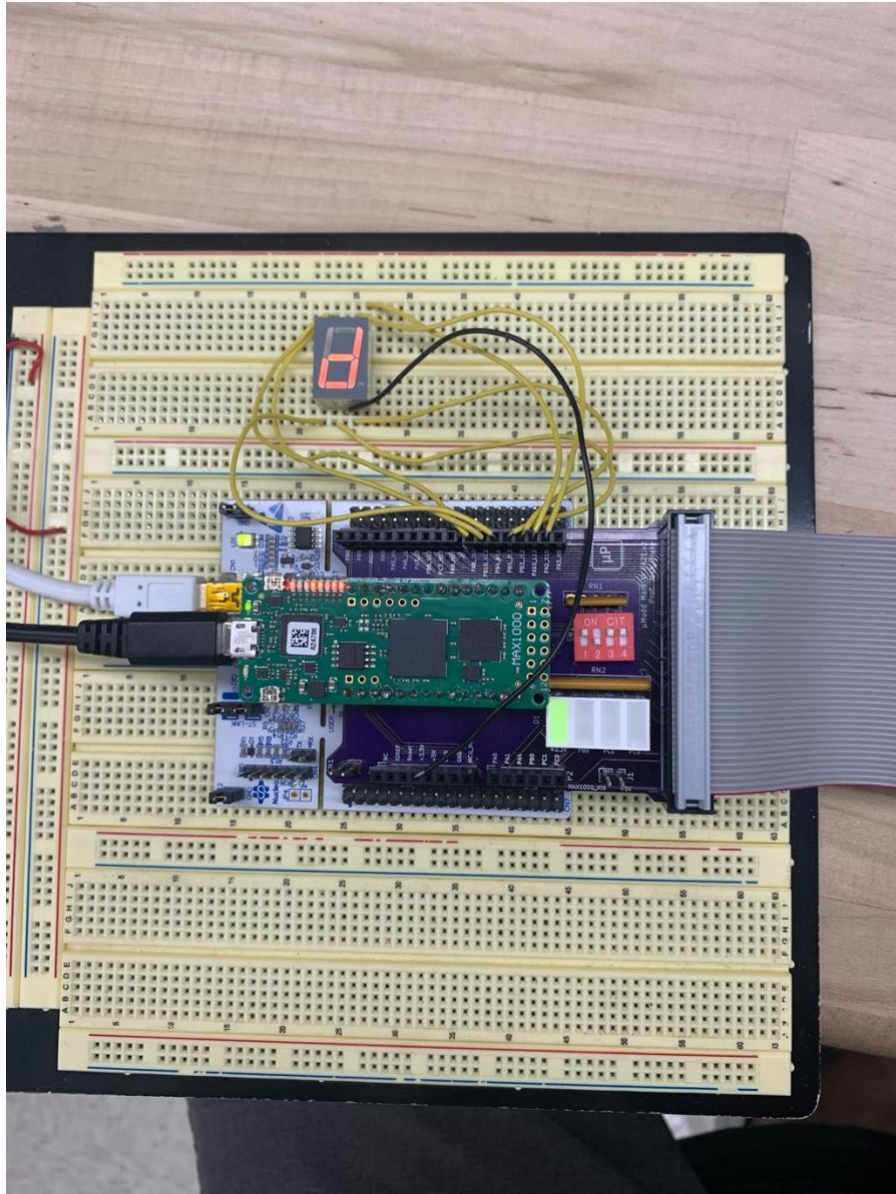


Board Assembly and Testing

Engineering 155 Lab I Report

September 7th, 2021

John Hearn



Introduction:

The goal of this lab was to assemble and test the μ Mudd Mark IV boards and test them with the Nucleo F401RE MCU and MAX1000 FPGA boards. We were to solder the boards with the appropriate components (header pins, resistor networks, dip switches, etc.) before programming the FPGA with Verilog code and interfacing a seven-segment display to the board which showed a single hexadecimal digit corresponding to the input on the switches.

Design Methodology:

In designing the hardware for the system, the main point of concern was the maximum current that the 7-segment display could handle. Consultation of the UA5651-11EWRS datasheet made it known that the acceptable range for current through the component is between 5 and 20mA. For simplicity of calculation, 10mA was chosen. The 3.3V power source on the μ Mudd Mark IV was used to power the component, and so it was found that a 330-ohm resistor was necessary to limit the current to 10mA as shown in Figure 1 below.

$$R = \frac{V}{I} \quad R = \frac{3.3V}{10mA} \quad R = 330\Omega$$

Fig. 1: Calculation for strength of resistor

In designing the software for the system, two SystemVerilog modules were created using Quartus. One main module for calculating outputs of LEDs and segments of the 7-segment display and one counter module aiding in blinking an LED at 2.4Hz. The given parameters for the software portion are outlined in Figure 3 below.

Signal	Signal Type	Description
clk	input	12 MHz clock on FPGA
s[3 : 0]	input	the four DIP switches
led[7 : 0]	output	the 8 lights on the LEDs on the MAX1000 board
seg[6 : 0]	output	the segments of a common-anode 7-segment display

Fig. 3: Given parameters for the main module

The input clk was driven by the FPGA's built-in 12MHz clock, and the input was taken in by assigning the correct pin on the board to drive the bit. We used the clk input to blink LED 7 at 2.4Hz using a divide-by- 2^N counter whose implementation will be discussed later. However, using the equation $f_{out} = f_{clk} * (p / 2^N)$, it was found that the combination of $p = 214$, $N = 30$ gives an accurate approximation of 2.4Hz.

Each bit of the output seg[6:0] corresponded to one of the segments in the display, which was to display a single hexadecimal digit specified by the input s[3:0]. The logic for led[7:0] is shown in Figure 4 below.

S0	LED0	LED1
0	OFF	ON
1	ON	OFF

S1	LED2	LED3
0	OFF	ON
1	ON	OFF

S2	LED4	LED5
0	OFF	ON
1	ON	OFF

S3	S2	LED6
0	0	OFF
0	1	OFF
1	0	OFF
1	1	ON

LED7
Blink at 2.4 Hz

Fig. 4: Logic for the output of each LED

Testing of the software component was conducted by simulating the logic in ModelSim. By using the force command in the simulation terminal, every case for was tested and was found to yield the correct result for both the seg[6:0] and led[7:0] outputs.

Technical Documentation:

```
1  module lab1_jh(input clk,
2                input s[3:0],
3                output led[7:0],
4                output [6:0] seg);
5
6      // Compute LED logic
7      assign led[0] = s[0];
8      assign led[1] = ~s[0];
9      assign led[2] = s[1];
10     assign led[3] = ~s[1];
11     assign led[4] = s[2];
12     assign led[5] = ~s[2];
13     and g5(led[6], s[2], s[3]);
14
15     counter(clk, led[7]);
16
17     // Number segment display logic
18     // bits of seg[6:0] go in order of gfedcba, which correspond to a segment
19     always_comb
20     case(s[3:0])
21         4'b0000: seg = 7'b1000000;
22         4'b0001: seg = 7'b1111001;
23         4'b0010: seg = 7'b0100100;
24         4'b0011: seg = 7'b0110000;
25         4'b0100: seg = 7'b0011001;
26         4'b0101: seg = 7'b0010010;
27         4'b0110: seg = 7'b0000010;
28         4'b0111: seg = 7'b1111000;
29         4'b1000: seg = 7'b0000000;
30         4'b1001: seg = 7'b0011000;
31         4'b1010: seg = 7'b0001000;
32         4'b1011: seg = 7'b0000011;
33         4'b1100: seg = 7'b1000110;
34         4'b1101: seg = 7'b0100001;
35         4'b1110: seg = 7'b0000110;
36         4'b1111: seg = 7'b1001110;
37         default: seg = 7'b1111111;
38     endcase
39
40 endmodule
41
42 // Get clock input and create a divide-by-2^N counter
43 // Was found that N = 30 and p = 214 is a precise combination for a 2.4Hz output
44 module counter (input logic clk,
45                output logic high);
46     logic [29:0] q;
47     always_ff @(posedge clk)
48         q <= q + 214;
49     assign high = q[29];
50
51 endmodule
```

Fig. 5: SystemVerilog code

On the previous page is the SystemVerilog code used to operate the system, and below is the logical circuit diagram associated with it.

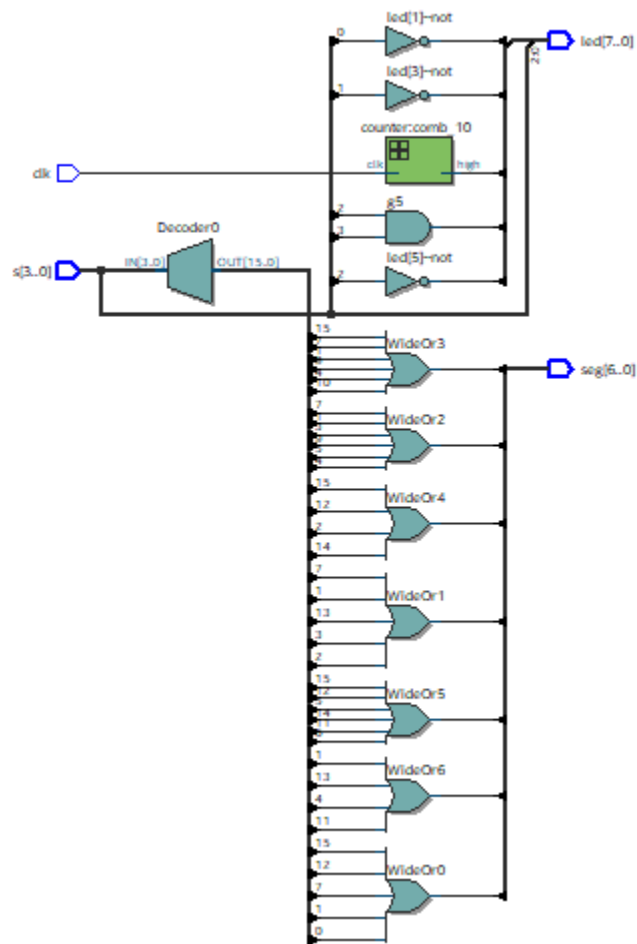


Fig. 6: RTL schematic showing logic elements

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate
in clk	Input	PIN_H6	2	B2_NO	PIN_H6	2.5 V		12mA (default)	
out led[7]	Output	PIN_D8	8	B8_NO	PIN_D8	2.5 V		12mA (default)	2 (default)
out led[6]	Output	PIN_C10	8	B8_NO	PIN_C10	2.5 V		12mA (default)	2 (default)
out led[5]	Output	PIN_C9	8	B8_NO	PIN_C9	2.5 V		12mA (default)	2 (default)
out led[4]	Output	PIN_B10	8	B8_NO	PIN_B10	2.5 V		12mA (default)	2 (default)
out led[3]	Output	PIN_A10	8	B8_NO	PIN_A10	2.5 V		12mA (default)	2 (default)
out led[2]	Output	PIN_A11	8	B8_NO	PIN_A11	2.5 V		12mA (default)	2 (default)
out led[1]	Output	PIN_A9	8	B8_NO	PIN_A9	2.5 V		12mA (default)	2 (default)
out led[0]	Output	PIN_A8	8	B8_NO	PIN_A8	2.5 V		12mA (default)	2 (default)
in s[3]	Input	PIN_D1	1A	B1_NO	PIN_D1	2.5 V		12mA (default)	
in s[2]	Input	PIN_C1	1A	B1_NO	PIN_C1	2.5 V		12mA (default)	
in s[1]	Input	PIN_C2	1A	B1_NO	PIN_C2	2.5 V		12mA (default)	
in s[0]	Input	PIN_E1	1A	B1_NO	PIN_E1	2.5 V		12mA (default)	
out seg[6]	Output	PIN_J10	5	B5_NO	PIN_J10	2.5 V		12mA (default)	2 (default)
out seg[5]	Output	PIN_K12	5	B5_NO	PIN_K12	2.5 V		12mA (default)	2 (default)
out seg[4]	Output	PIN_K11	5	B5_NO	PIN_K11	2.5 V		12mA (default)	2 (default)
out seg[3]	Output	PIN_J13	5	B5_NO	PIN_J13	2.5 V		12mA (default)	2 (default)
out seg[2]	Output	PIN_J12	5	B5_NO	PIN_J12	2.5 V		12mA (default)	2 (default)
out seg[1]	Output	PIN_L12	5	B5_NO	PIN_L12	2.5 V		12mA (default)	2 (default)
out seg[0]	Output	PIN_K10	5	B5_NO	PIN_K10	2.5 V		12mA (default)	2 (default)

Fig. 7: Pin placements

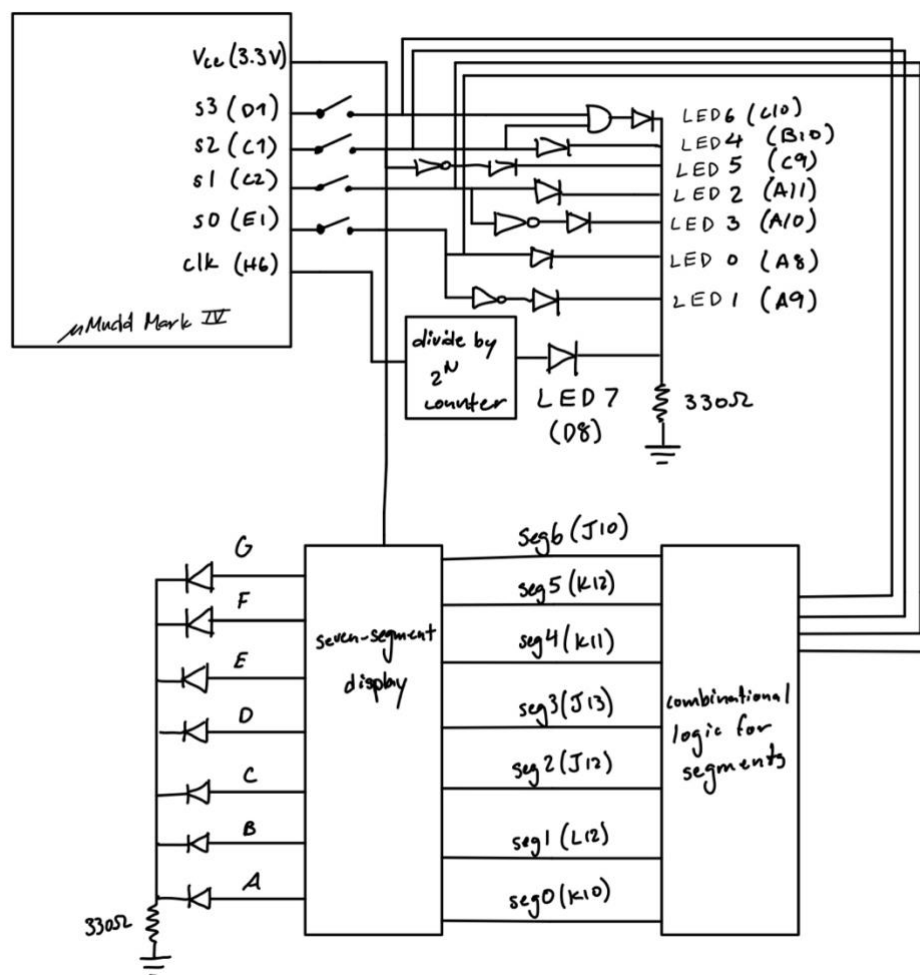


Fig. 8: Circuit schematic

Results and Discussion:

I was successful in accomplishing all the prescribed tasks. The board's LEDs lit up properly and every possible digit was tested and successful.

If I were to redo this lab, I would solder my board correctly the first time. I had a nightmare resoldering my board for hours on end.

Conclusion:

I was successful in assembling and testing my board. I successfully interfaced a 7-segment display to the board controlled by the DIP switches I soldered onto the board. The system performed as I had hoped, and I was able to get it to work perfectly by the end of the lab.

This lab took 16 hours to complete, and the report alone took 4 hours. I fear I have no choice but to become a basement-dwelling troll this semester...

As for the future, I think the lab could be a little more thoroughly explained. I spent a lot of confused over instructions I felt were unclear and found myself frequently needing the help of proctors.