ELC 2137 Lab 6: MUX and 7-segment Decoder

CJ Jones, Ashlie Lackey

February 26, 2020

Summary

This lab explored using a Basys3 board to produce an 8-bit number on a 7-segment display through a MUX combinational logic design. The MUX used had two inputs that were four bits each. Using Verilog, some skills gained in this lab include: writing a multiplexer utilizing the conditional operator, using always block, using multi-bit signals, producing a toplevel module, using constraint files,and creating a design on a FPGA board. Overall, this lab demonstrated how to utilize software and programmable logic to produce a hardware output.

Q&A

1. How many wires are connected to the 7-segment display?

   In our version of the seven segment display, only 1 wire is connected to the 7-segment display

2. If the segments were not all connected together, how many wires would there have to be?

3. Why do we prefer the current method vs. separating all of the segments?

Results

Below are the results for running the half-, full-, and 2-bit adders. The design and testbench code for each adder is included along with each expected results table and the block diagram of each digital circuit. As the ERT's and screenshots each indicate, the results from the testbench simulations do match what is expected.

The results match what was expected as the screenshotted waveform results confirm the expected results table.

Code

Listing 1: mux2-4b Code

```
// Ashlie Lackey and Chris Jones , ELC 2137, 2020 -2-26
module mux2_4b(
input [3:0]in0,
input [3:0]in1,
input sel,
output [3:0]out
);

assign out = sel?in1:in0;
endmodule
```

Listing 2: MUX Testbench Code

```
// Ashlie Lackey and Chris Jones , ELC 2137, 2020 -2-26
module mux2_4b_test();
reg [3:0]in0_t, in1_t;
reg sel_t;
wire [3:0]out_t;

mux2_4b Mux(.in1(in1_t), .in0(in0_t), .sel(sel_t), .out(out_t));

initial begin
in0_t = 4'b0000;
in1_t = 4'b0001;
sel_t = 0;
#10;

in0_t = 4'b0001;
in1_t = 4'b0010;
sel_t = 1;
#10;

in0_t = 4'b0010;
in1_t = 4'b0001;
sel_t = 0;
#10;

in0_t = 4'b0010;
in1_t = 4'b0001;
sel_t = 1;
#10;

$finish ;
end
endmodule   //mux2_4b_test
```