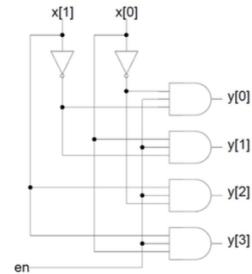


Willis Allstead
CPE 301-1001
Lab #11
December 6, 2016

Assignment Description:

In this lab we were supposed to learn the basics of using the Altera Quartus-II FPGA compiler along with the Verilog language. We were to follow the supplied tutorial after reading some chapters of the user manual. Then, we were supposed to implement and test the two-input logic gates supplied in the lab description. A copy of the logic diagram is shown to the right. As you can see, it is a 2x4 decoder with an enable line.



Problems Encountered:

The only minor problem we encountered in this lab was figuring out how to use Verilog in such a way that the inputs could be read, and then using predefined boolean equations, light up corresponding LEDs in the 7-segment display on the supplied FPGA.

Lessons Learned:

We learned how to really implement an admittedly simple program through using Verilog, which means that we can learn more in the future. We learned how to declare the inputs and outputs in verilog, along with how to assign the corresponding addresses to those inputs and outputs before compiling.

Description of Completed Lab:

The code below is what we ended up with for our final product. It probably isn't the most efficient code possible for the program, but we were happy with it and it got the job done. As you can see we declared the two inputs a and b, then an array of outputs which held the 6 possible outputs of the decoder, (should have been 5 because of the enable being 0). Then we declared 7 LED outputs for each segment in the 7-segment display. We used the outputs in the decoded array as inputs to the boolean equations of the LED array.

```
module gate2 (in_a, in_b, en, out, LED_7);
    input wire in_a; // declare input variable in_a
    input wire in_b; // declare input variable in_a
    input wire en; //switch 9 enable
    output wire [5:0] out; // declare out variable out
    output wire [6:0] LED_7; //declare led variable
    assign out[4] = ~en; //if enable isn't active
    assign out[3] = en & (in_a & in_b); // decoder output 3
    assign out[2] = en & (~in_a & in_b); // decoder output 2
    assign out[1] = en & (in_a & ~in_b)); // decoder output 1
    assign out[0] = en & (~in_a & ~in_b); // decoder output 0
    //makes the seven segment display work with decoder
    assign LED_7[0] = ~(out[1] | out[2] | out[4]); //makes top LED light up
    assign LED_7[1] = ~(out[0] | out[1] | out[2] | out[3] | out[4]); //makes top right led light up
    assign LED_7[2] = ~(out[0] | out[2] | out[3] | out[4]); //makes bottom right led light up
    assign LED_7[3] = ~(out[1] | out[2] | out[4]); //makes bottom led light up
    assign LED_7[4] = ~(out[1] | out[4]); //makes bottom left led light up
    assign LED_7[5] = ~(out[3] | out[4]); //makes top left led light up
    assign LED_7[6] = ~(out[1] | out[2] | out[3]); //makes middle led light up

endmodule
```

The final pin table we ended up with is below.

Name	Type	Pin	Value	Pad	Notes
LED_7[6]	Output	PIN_E2	2	B2_N1	3.3-V LVTTL (default)
LED_7[5]	Output	PIN_F1	2	B2_N1	3.3-V LVTTL (default)
LED_7[4]	Output	PIN_F2	2	B2_N1	3.3-V LVTTL (default)
LED_7[3]	Output	PIN_H1	2	B2_N1	3.3-V LVTTL (default)
LED_7[2]	Output	PIN_H2	2	B2_N1	3.3-V LVTTL (default)
LED_7[1]	Output	PIN_J1	2	B2_N1	3.3-V LVTTL (default)
LED_7[0]	Output	PIN_J2	2	B2_N1	3.3-V LVTTL (default)
en	Input	PIN_L2	2	B2_N1	3.3-V LVTTL (default)
in_a	Input	PIN_L22	5	B5_N1	3.3-V LVTTL (default)
in_b	Input	PIN_L21	5	B5_N1	3.3-V LVTTL (default)
out[5]	Output	PIN_V19	6	B6_N1	3.3-V LVTTL (default)
out[4]	Output	PIN_T18	6	B6_N1	3.3-V LVTTL (default)
out[3]	Output	PIN_Y19	6	B6_N1	3.3-V LVTTL (default)
out[2]	Output	PIN_U19	6	B6_N1	3.3-V LVTTL (default)
out[1]	Output	PIN_R19	6	B6_N0	3.3-V LVTTL (default)
out[0]	Output	PIN_R20	6	B6_N0	3.3-V LVTTL (default)

To the right is a picture of us getting the LEDs to work from right to left on top of the switches. This was the step before attempting to implement the code above. Below is a picture of the implemented final code. The 7-segment to the far right is the one we used, and the switch to the far left was our enable. When enabled, you can see that when b was enabled and a wasn't, we were at 3 on the display. On the last page of this report you can see that 4 is displayed when both inputs are enabled.

