CSE100 Lab 1 – 7-Segment Display

Contents

1 Submission

- Due: Thursday 10/3/2024 5:00pm.
- Submit your code to the Gradescope autograder. (You have unlimited submission attempts).
- Demonstrate your implementation to a TA.
- Submit your pre-lab and post-lab PDF to Gradescope.

2 Goals

You will implement a 7 Segment Display Decoder in Verilog.

Write a decoder that will display a 4-bit hex nibble onto the on-board 7-Segment Display.

Once you finish your implementation, you will program it to a Basys 3 FPGA Board.

3 Prelab

Complete the following questions before starting the lab.

1. Give a 1-sentence definition of each of the following aspects of the chip-developmment flow:

- Hardware Description Language
- Simulation
- Synthesis
- Hardware Target
- Place and Route
- Bitstream Generation
- FPGA
- FPGA Development Board

2. What is the name of the FPGA Development Board we are using in this class?
3. What is the name of the FPGA we are using in this class?
4. How many Logic Cells does this class' FPGA contain? (https://docs.amd.com/v/u/en-US/ds180_7Series_Overview).

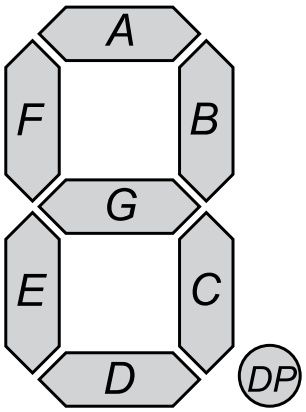

Figure 1: 7 Segment Display with Labeled Segments (By Uln2003 - Own work, CC0, https://commons.wikimedia.org/w/index.php?curid=69807877)

5. Complete the following truth table for the 7 Segment Display Circuit.

Display	d3	d2	d1	d0	A	B	C	D	E	F	G
0	0	0	0	0							
1	0	0	0	1							
2	0	0	1	0							
3	0	0	1	1							
4	0	1	0	0							
5	0	1	0	1							
6	0	1	1	0							
7	0	1	1	1							
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1							
A	1	0	1	0							
B	1	0	1	1							
C	1	1	0	0							
D	1	1	0	1							
E	1	1	1	0							
F	1	1	1	1							

4 Lab

You need to complete and submit the following files:

- `"rtl/hex7seg.sv"`
- `"synth/basys3/Basys3_Master.xdc"`
- `"synth/basys3/basys3.sv"`

Read about the 7-Segment Displays in the Basys3 Board Reference Manual. The FPGA pin names used for the 7-segment display controls in this manual are `A`, `B`, `C`, `D`, `E`, `F`, `G`, `dp`, `an[3]`, `an[2]`, `an[1]`, `an[0]`.

1. Complete the `hex7seg` module. Use the truth table from the prelab to enter the equations for `A`, `B`, `C`, `D`, `E`, `F`, `G`. Note that for segments that are on more often than they are off, you can
2. Complete the `basys3` module. You need to figure out what the values should be for `dp`, `an[3]`, `an[2]`, `an[1]`, `an[0]` using the Basys3 Board Reference Manual: https://reference.digilentinc.com/reference/programmable-logic/basys-3/reference-manual. The decimal point should not be lit.
3. You are provided a Basys3 constraints file `"synth/basys3/Basys3_Master.xdc"`. It was downloaded from from Digilent's GitHub: https://github.com/Digilent/Basys3/blob/master/Resources/XDC/Basys3_Master.xdc. Complete it as specified.
4. Simulate, synthesize, and program your design. Observe the `"README.md"`.

5 Post-lab

Complete the following questions after finishing the lab.

1. Run the Makefile script to generate the bitstream. Now, open the Xilinx Project file in the Vivado GUI. Likely, the command will be

```
vivado -nolog -nojournal synth/basys3/build/basys3/basys3.xpr
```

Now open the Implementation Schematic that Vivado generated. Be sure to expand your `hex7seg` module to reveal its contents. Include a screenshot of the schematic in your lab report.

2. How many LUTs does your design use? Does this make sense? What does that number represent?