

EECS 140: Lab 6 Lab
VHDL Design 7-Segment Display

Morgan Bergen

KUID: 3073682

Date submitted: 10/18/2022

1. Introduction and Background

The objective of this lab was to use the scripting language VHDL to create a generic 7-segment output component and implement it in a FPGA chip. The implementation of VHDL was used on the IDE Xilinx Vivado Software and after configuring the component, its functionality was used to display a hexadecimal number as output for a 4-binary input. When considering the different combinations that can be generated with 4-bits there are 2^n different combinations for every n providers therefore with $n = 4$ bits there are 16 different possible generated combinations allowed. After testing the holistic functionality of the chip the FPGA board's slide switches was being used for inputting a 4-bit binary number, and the hexadecimal equivalent outputted on one of the seven segment displays.

For additionally background knowledge, a basic understanding of the difference between 'ENTITY' and 'ARCHITECTURE' in a VHDL module is necessary. First, a FPGA stands for field programmable gate arrays and can be used to build a processor or digital circuit board so they it can allow for logical expressions to be implemented on. Secondly an ENTITY defines the interface of the module and the ARCHITECTURE defines the function of the model. The architecture statement describes that underlying functionality of the entity, and is always related to the entity's behavior and thus the internal aspect of the entity, however the ENTITY is the interface for which the external view of the component is described and defined.

2. Implementation Process

The implementation process comprised of 4 main tasks, defining a truth table for the seven-segment display prior to design, scripting the Architecture of the device in VHDL, defining the constraints file for setting each property type and mapping each input/output a physical component of the device, and lastly programming it into the FPGA.

Truth Table: First writing a truth table that defines the truth state of a given output continent upon each configuration of input combinations, please note that all outputs run on Negative Logic. The truth table is as follows,

VAL	INPUTS <u>wxyz</u>	OUTPUTS <u>abcdefg</u>	INPUT VAR Equivalence Mapping
0	0000	0000001	sw[0] == w
1	0001	1001111	sw[1] == x
2	0010	0010010	sw[2] == y
3	0011	0000110	sw[3] == z
4	0100	1001100	
5	0101	0100100	
6	0110	0100000	OUTPUT VAR Equivalence Mapping
7	0111	0001111	seg[0] == a
8	1000	0000000	seg[1] == b
9	1001	0001100	seg[2] == c
A	1010	0001000	seg[3] == d
B	1011	1100000	seg[4] == e
C	1100	0110001	seg[5] == f
D	1101	1000010	seg[6] == g
E	1110	0110000	
F	1111	0111000	

After defining the truth table for the 7-segment display, the project from Vivado was required to be initialized. Upon initializing the project I was required to define a module and specific the I/O Ports to add to my source file (specifically the file for running the script onto the device). The entity name was SevenSeg and the Architecture name was Behavioral. We had to define 3 ports called switches, segments, and anodes.

The switches created a 4-bit bus with each pin named switches[0], switches[1], switches[2], switches[3] that were inputs that ranged from 0 - 3 (therefore defining that there are four input switches).

The segments that were the outposts that ranged from 0 - 6 (therefore defining that there are seven output segments).

Then thereafter we needed to associate the inputs and outputs of our design to the right pins of the FPGA. In the unit description we were required to script the behavior of the driver, the name of my entity was titled sevenSeg.vhd. The figure 1 illustrates the SevenSeg.vhd file, and figure 2 illustrates the constraints.xdc file. Then thereafter we used with and when statements for assigning the values for the outputs, the script was as follows in the architecture file. The following figure 2 illustrates the constraints.xdc file.

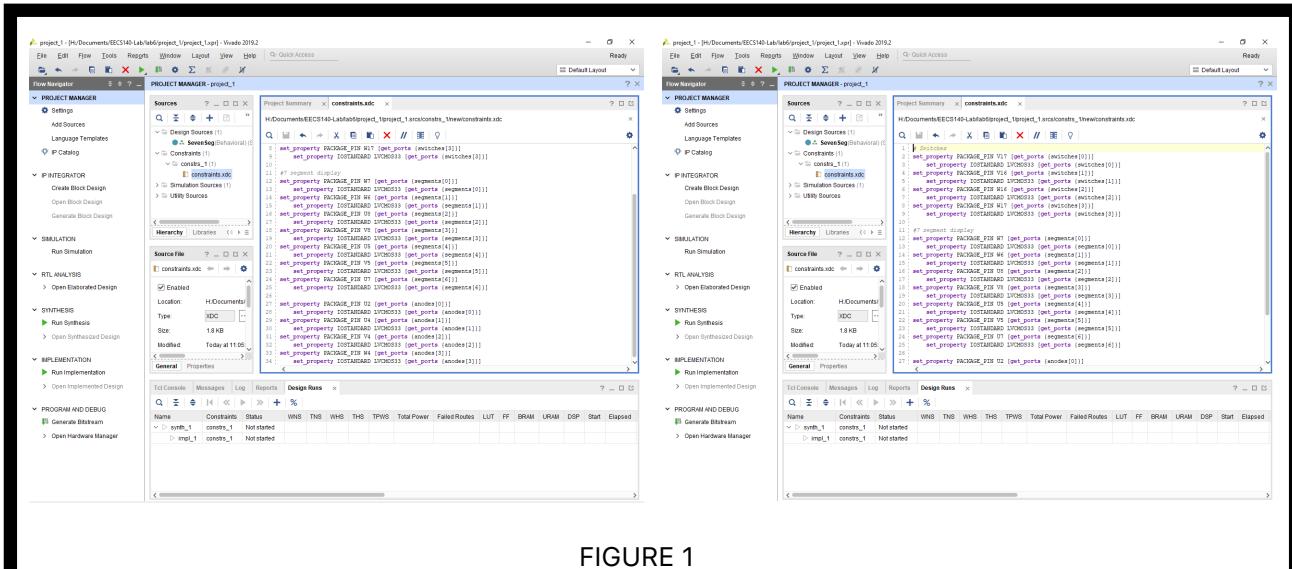


FIGURE 1

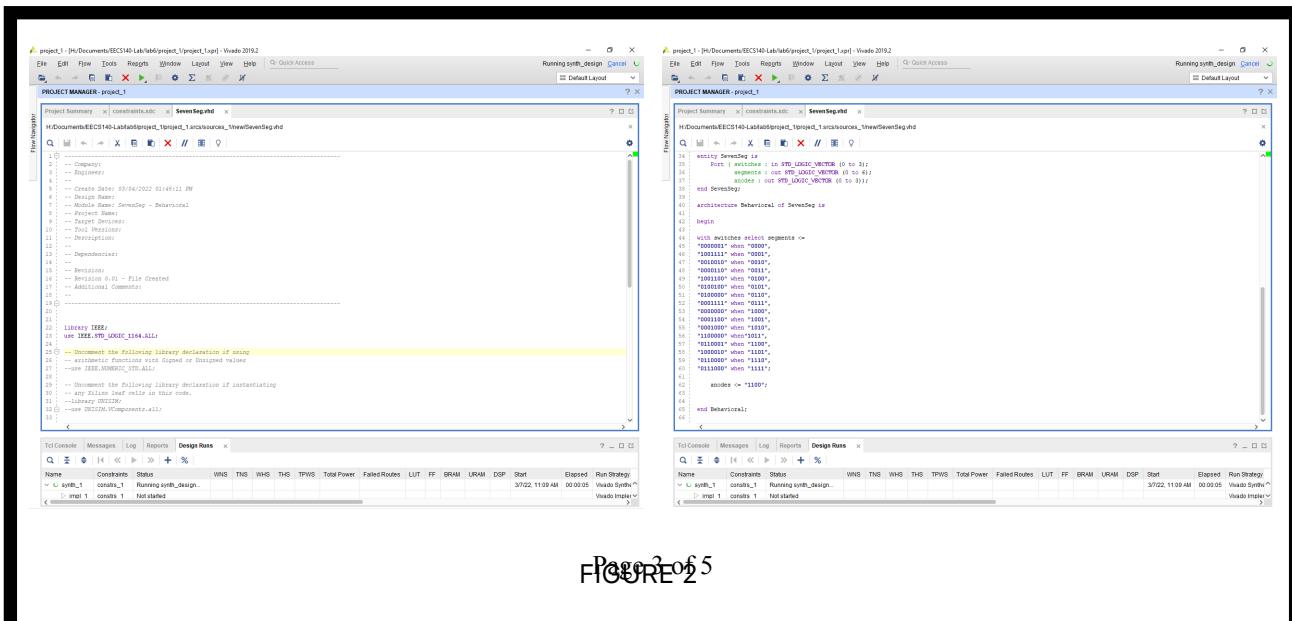
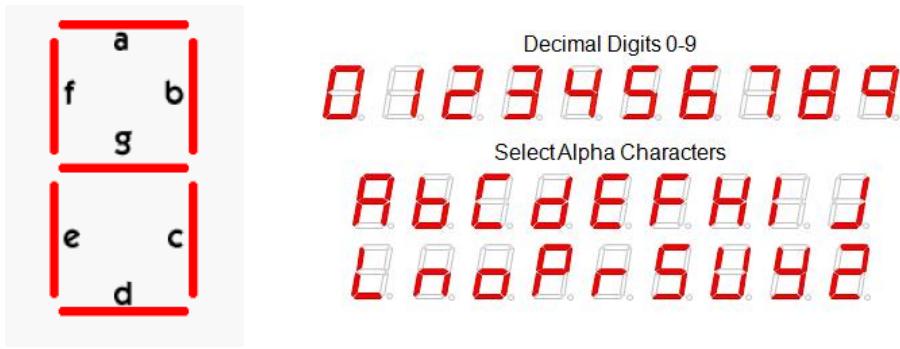


FIGURE 2



3. Evaluation Process

The Evaluation process entailed connecting the Basis3 board to the windows machine and powering it on, for ensuring that the jumper was set to the usb, the slide switches and 7 segment display was used to test the design of my implementation. If the board is operating correctly then the verification process should provide that the inputs using the slide switches were correctly corresponding to the output on the 7 segment display selected. Because this was a negative logic board all bits were inverted.

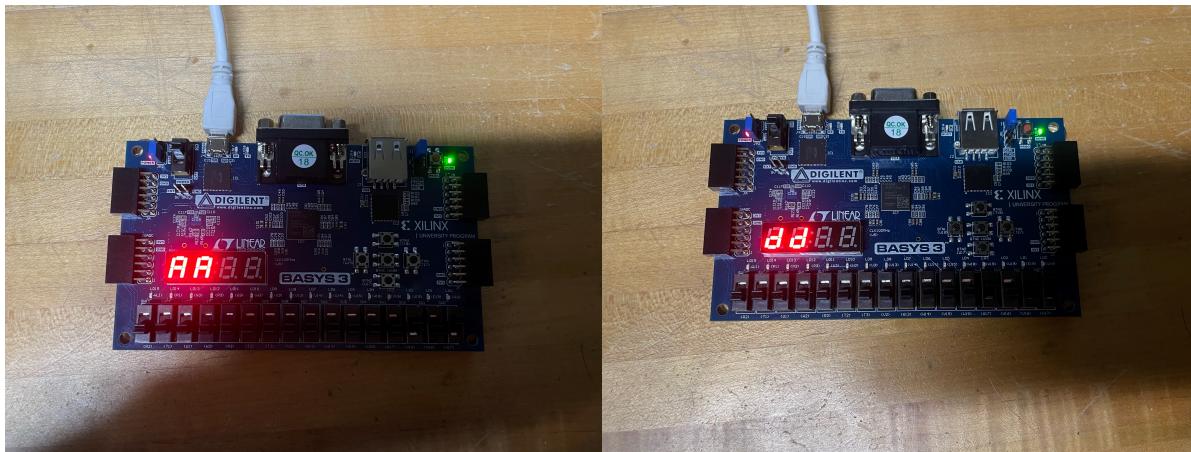


Figure 3

Figure 4

The following Figures figure 3, figure 4 illustrates the negatives logic that allows for a 'b' and 'a' output display.

value	input	segment output
A	1010	0001000
B	1011	1100000

VAL	INPUTS	OUTPUTS
	wxyz	abcdefg
0	0000	0000001
1	0001	1001111
2	0010	0010010
3	0011	0000110
4	0100	1001100
5	0101	0100100
6	0110	0100000
7	0111	0001111
8	1000	0000000
9	1001	0001100
A	1010	0001000
B	1011	1100000
C	1100	0110001
D	1101	1000010
E	1110	0110000
F	1111	0111000

4. Results & Discussion

The results of this lab demonstrated that the driver had the correct functionality to take in a 4-bit binary number from the slide switches and display its actual value onto the display illustrated by the hexadecimal formed value on the led. The characters using the seven segment display appeared to be illuminated as follows.

5. Conclusions and Recommendations

This lab helped me as a programmer understand how to program the seven segment display and map it through light emitting diodes denoted by the hexadecimal number system, also made my knowledge base for VHDL more robust. No recommendations are necessary.