EECS 140: Lab 11

Encoder and Decoder

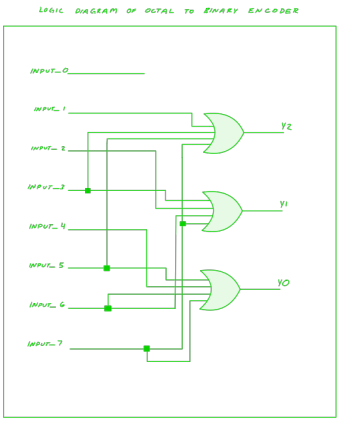
Morgan Bergen

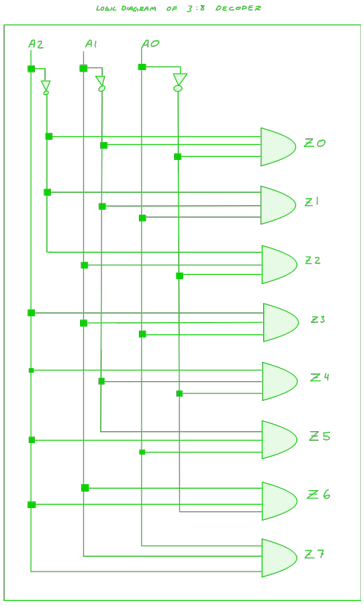
KUID: 3073682

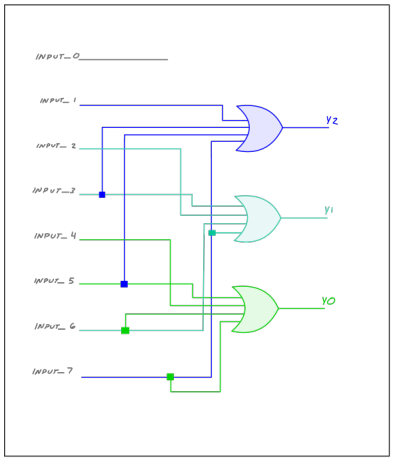
Date submitted: 12/3/2022

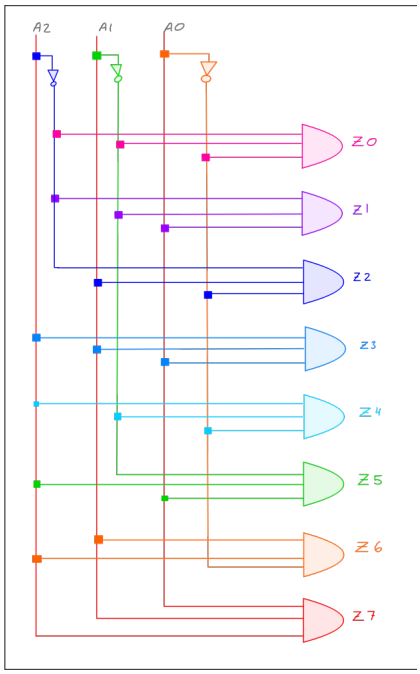
1. Introduction and Background

This lab comprised of decoder and encoder circuits. The decoder circuits had n inputs and up to 2n outputs, where output was associated with each possible binary input. Encoders on the other hand were the opposite of decoders, whereby encoders had 2n inputs and n outputs. The design and implementation of this program was a final 3-to-8 decoder and 8-to-3 encoder. The following shows a diagram of the logic diagram of octal to binary encoder & logic diagram of 3:8 decoder, one colored and not colored for visibility purposes.









For explanation purposes the decoder took the form of multi-input, multi-output logic circuit that converted coded inputs into coded outputs, where the input and output codes were different e.g. n-to-2n, binary-coded decimal decoders. Decoding is necessary for example in applications such as data multiplexing, 7-segment display, and memory address decoding.
Encoders on the other hand were combinational logic circuits and they are exactly opposite of decoders. They accept one or more inputs and generate a mult-ibit output code. An encoder has m input and n output lines. Out of the m input lines only one is activated at a time and produces equivalent code on output n lines. If a device output code has fewer bits than the input code has, the device is usually called an encoder. The logic circuit of a 8-to-3 encoder and a 3-to-8 decoder is presented below.
encode 2^n => n
decode n => 2^n

2. Implementation Process

The first task was to generate a truth table for the 3-to-8 decoder, here's that following table and equations

3-to-8 decoder truth table

input				output								
	x_0	x_1	x_2		y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
	U1	T1	R2		V13	V3	W3	U3	P3	N3	P2	L1
	0	0	0		1	0	0	0	0	0	0	0
	0	0	1		0	1	0	0	0	0	0	0
	0	1	0		0	0	1	0	0	0	0	0
	0	1	1		0	0	0	1	0	0	0	0
	1	0	0		0	0	0	0	1	0	0	0
	1	0	1		0	0	0	0	0	1	0	0
	1	1	0		0	0	0	0	0	0	1	0
	1	1	1		0	0	0	0	0	0	0	1

output eqtions of the 3:8 decoder

$$Z_0 = \overline{A_0 A_1 A_2}$$

$$Z_1 = A_0 \overline{A_1 A_2}$$

$$Z_2 = \overline{A_0} A_1 \overline{A_2}$$

$$Z_3 = A_0 A_1 A_2$$

$$Z_4 = \overline{A_0 A_1} A_2$$

$$Z_5 = A_0 \overline{A_1} A_2$$

$$Z_6 = \overline{A_0} A_1 A_2$$

$$Z_7 = A_0 A_1 A_2$$

Next we need to create a new project for the decoder in my H:// drive. And I generated a VHDL file for the above schematic provided and include it in the project.

Next I created the truth table for 3-to-8 encoder shown in the figure below

input(x)									output(y)			
	I7	I6	I5	I4	I3	I2	I1	I0		Y2	Y1	Y0
	0	0	0	0	0	0	0	0		0	0	0
	0	0	0	0	0	0	1	0		1	0	0
	0	0	0	0	0	1	0	0		0	1	0
	0	0	0	0	1	0	0	0		1	1	0
	0	0	0	1	0	0	0	0		0	0	1
	0	0	1	0	0	0	0	0		1	0	1
	0	1	0	0	0	0	0	0		0	1	1
	1	0	0	0	0	0	0	0		1	1	1

$$Y_0 = I_4 + I_5 + I_6 + I_7$$

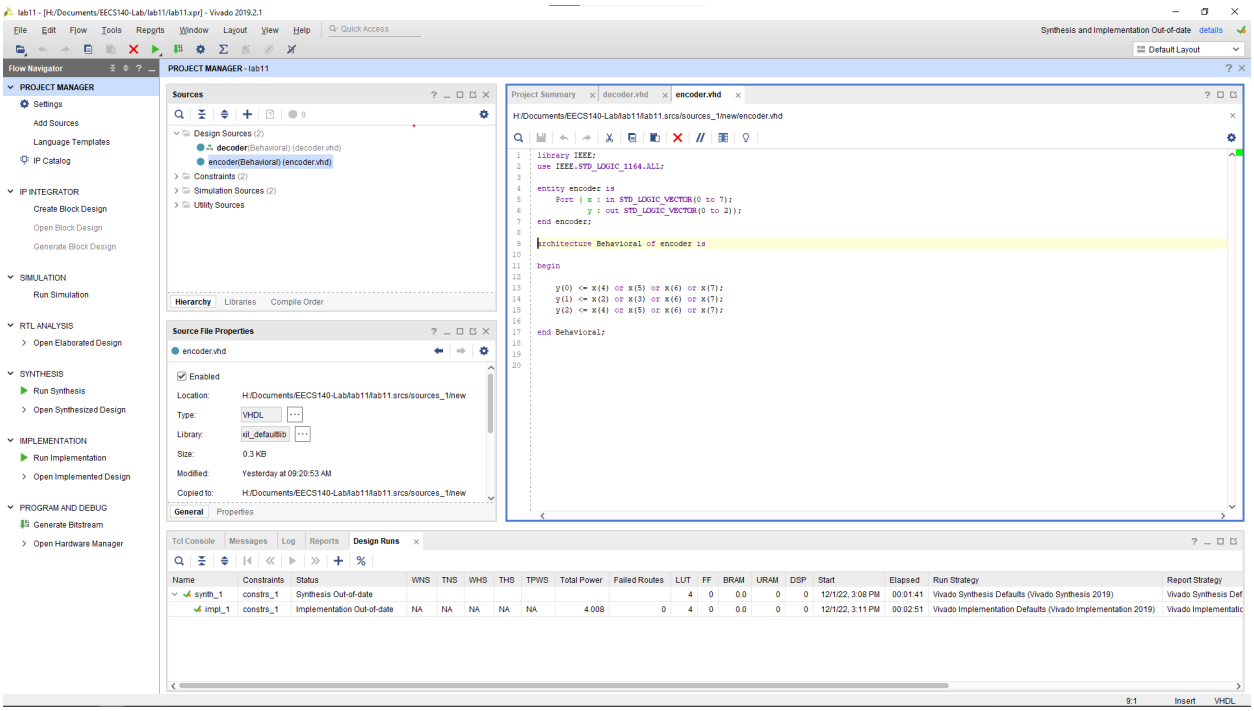
$$Y_1 = I_2 + I_3 + I_6 + I_7$$

$$Y_2 = I_1 + I_3 + I_5 + I_7$$

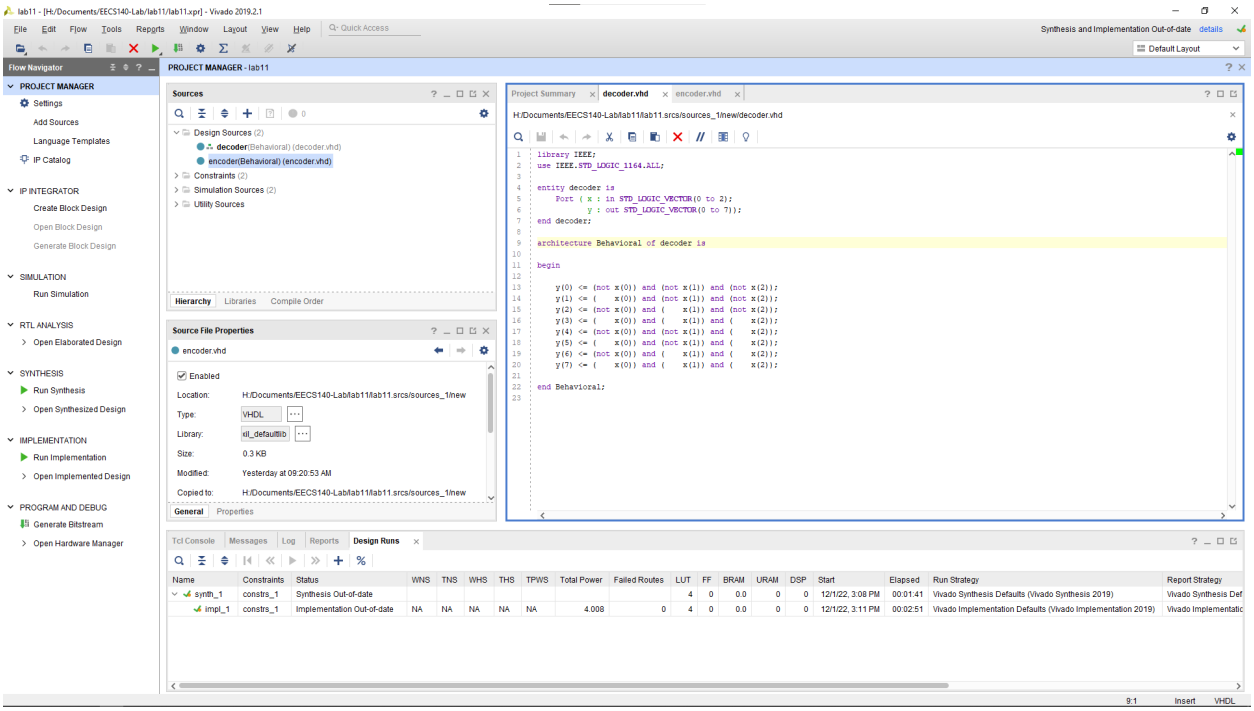
I then created the encoder behavior for the Basys3 Board. and the final aspects of the decoders and encoders are the necessary constraints files. The final components are declared on page 5 & 6.

3. Evaluation Process

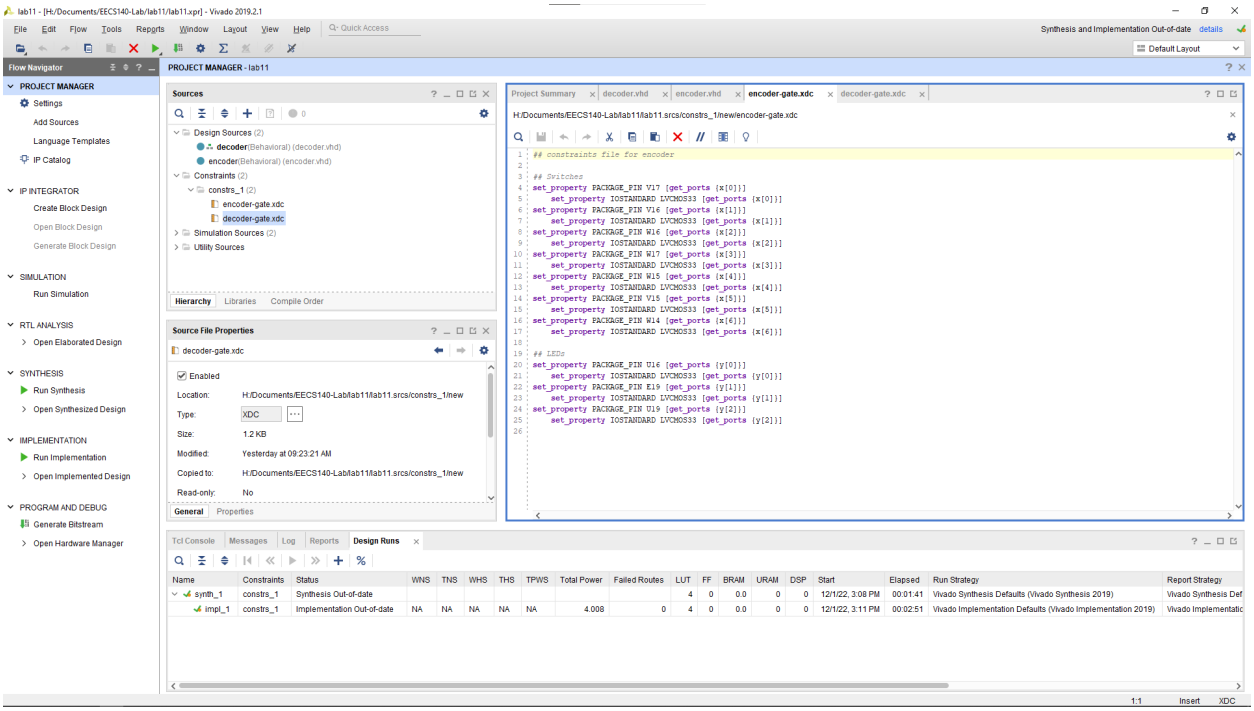
In order to verify the correctness of the design on the FPGA board you must test the design with different inputs and compare the outputs with the expected outputs. The following files were declared encoder.vhd, decoder.vhd, encoder.xdc, decoder.xdc



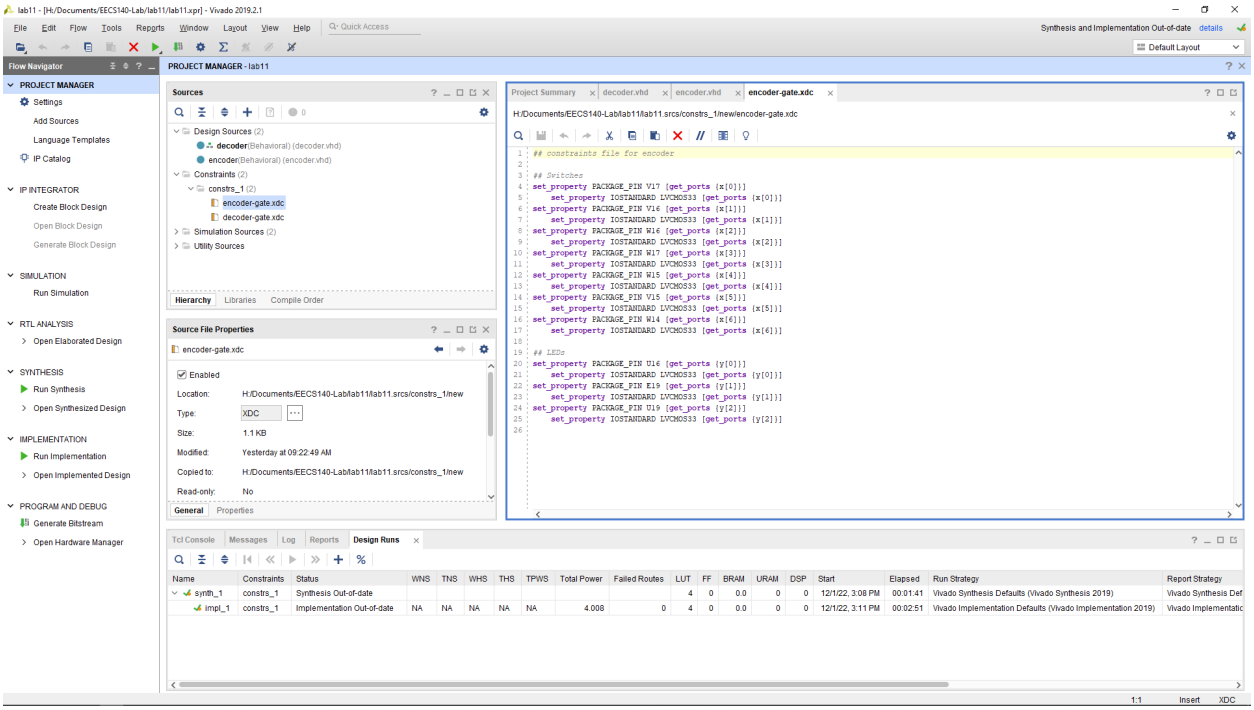
encoder.vhd



decoder.vhd



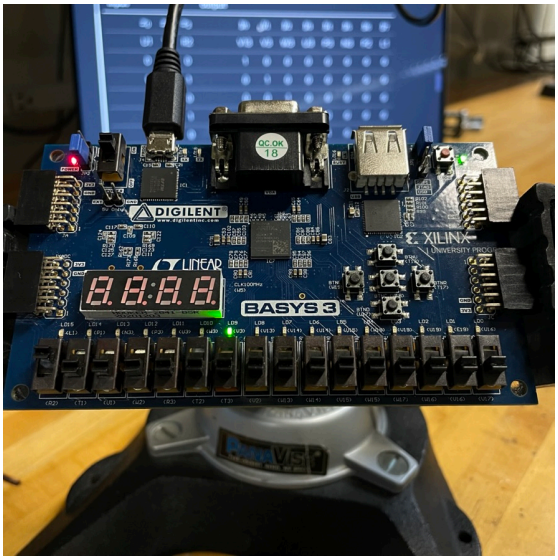
encoder.xdc



decoder.xdc

4. Results and Discussion

The results were light indicators that reflected the switches and outputs from the table.



5. Conclusion and Recommendations

I learned how decoders and encoders worked in relation to implementing VHDL code. And I made sure that all outputs were checked by the TA that I attended to. I would recommend that the class had a bit more time in order to implement, because it took me so long to finish. Also to state that hierarchy mattered when running bit streams.