**ELEC 204 Digital Design Lab Report**

Lab 3

Name: Metehan Gelgi

Date: 11/09/2019

1. **Introduction and objectives**

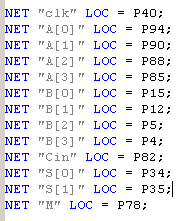
The main objective in this experiment is to get familiar with the elementary logic gates and to use them for implementing hierarchical logic circuit design. In this experiment we learned:

* design a 4-bit arithmetic and logic unit (ALU),
* implement the ALU on FPGA using a modular design,
* experimentally demonstrate the operation of the ALU.

In my code first of all I tried to divide my problem into pieces. First, we have 4-bit ALU with Logic and Arithmetic Units. I divided this 1-bit ALUs (like doing 4-bit fulladder uses 4 1-bit fulladder). 1-bit ALU have 2 main part one of them Logic Unit (and, or, xor, xnor) other one Arithmetic Unit (transfer, +, - and so on). I can switch between them using Multiplexer with selector M.

Logic Unit is so simple it shows our output after some logic operations on Leds if M=’0’.

For Arithmetic part(M=’1’), after 4-bit ALU I have 1 output with 4-bit binary number and Cout (which is carry). Finally, I checked most significant bit of output whether 1 or 0. If it is 1, it means I have negative number, so I translated (-2’s) complement, then I give x”a” to my negativeSign (which display – on 7-segment) . Otherwise I displayed my output directly.

1. **Methods**

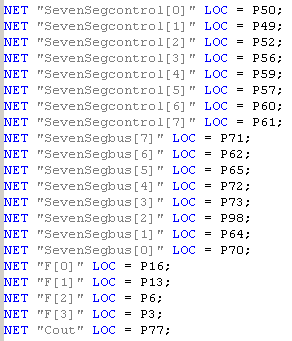
A and B are my 4-bit signed binary inputs. left most bit shows sign.

Cin is 1-bit input for Arithmetic Operation.

S(2-bit) and M(1-bit) selectors between operations and Logic, Arithmetic Units.

clk(1-bit) used for 7-segment display clock

**Figure 1.** ucf locations for inputs



F is my 4-bit output which is comes from Logic Unit or Arithmetic Unit

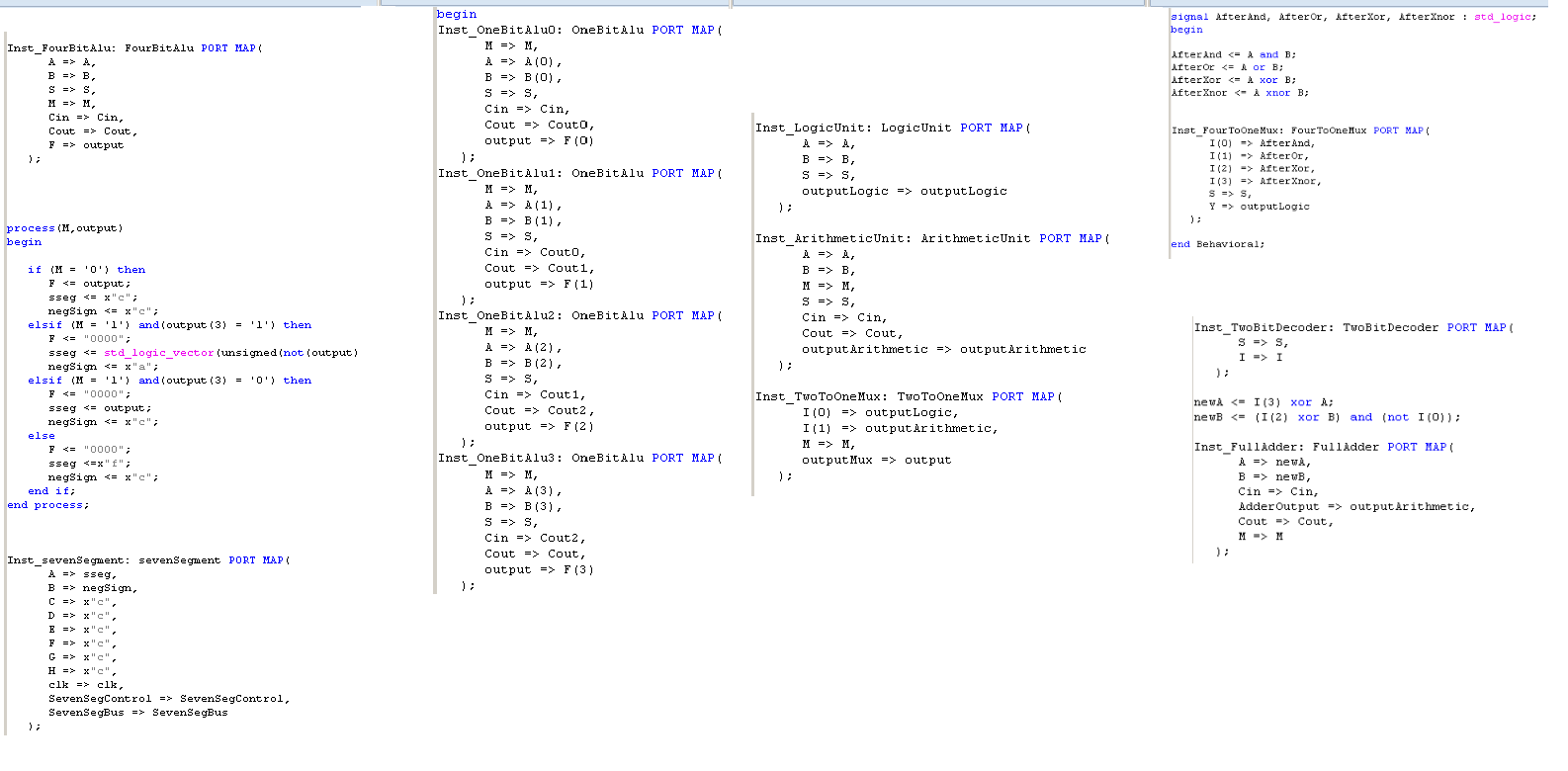
SevenSegBus(8-bit) and SevenSegControl(8-bit) used to display on 7-seg displaying on FPGA.

Cout is a 1-bit carry from Arithmetic Operations.

**Figure 2.** ucf locations for outputs

My Code has property of divide and conquer model. Because I have to do 4-bit ALU. To do this I used 4 1-bit ALU with Logic and Arithmetic Units. For Logic Unit first, I did all operations then, I choose needed one by using 4-to-1 Multiplexer.

On the other hand, I am doing Arithmetic operations inside of Arithmetic Unit. I used 1 Full-Adder. Before sending 1-bit inputs to the full-adder chose operation (transfer, +, -) by using 2-bit selector (S -> decoded with 2-to-4 decoder). Then I sent into the Full-Adder. After getting output from 4-bit ALU. I checked whether leftmost bit of output is 0 or 1(sign bit). If it is 1, I took 2’s complement. And I gave negSign to x”a” which comes from -2’s complement( I used 2’s complement for output) of negative number. Otherwise (number is positive, that’s why negSign is x”c”) I directly used this output. Finally, I switch between Logic and Arithmetic units by using M selector.

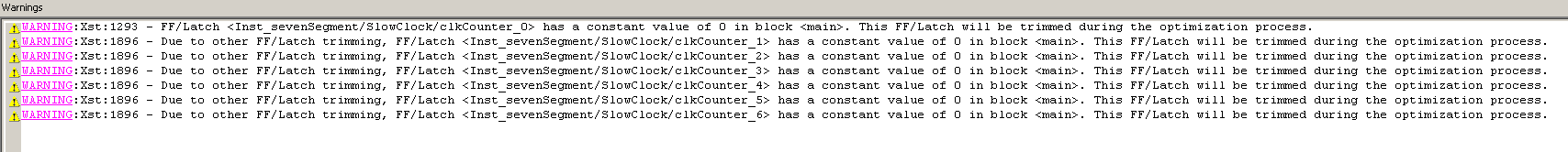


**Figure 3.**  VHDL code for LAB3 4-bit ALU



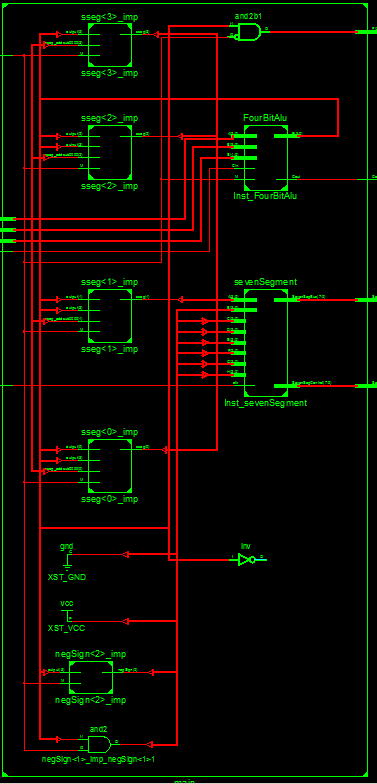
**Figure 4.** Truth Table for 1-bit ALU

1. **Problems encountered, errors and warnings resolved**

****

**Figure 5.** Warnings I observed

I got 1 main Warning during my LAB. I think, this warning occurs from timing problem because while doing experiment 7-segment display tries to show output. Maybe giving wait can solve this problem.

**elektronik eşyalar içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Initially I designed code for this LAB, however this code was so complicated and needed so much pre-functions. In the final code I can get my output directly by using 4-bit ALU then using some small functions.

**Figure 7.** RTL of final VHDL code

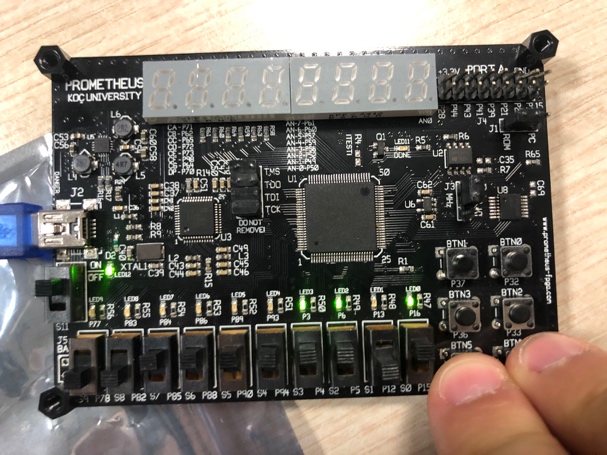
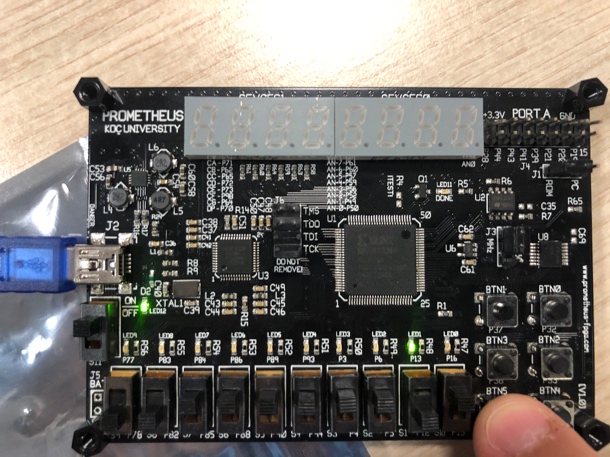
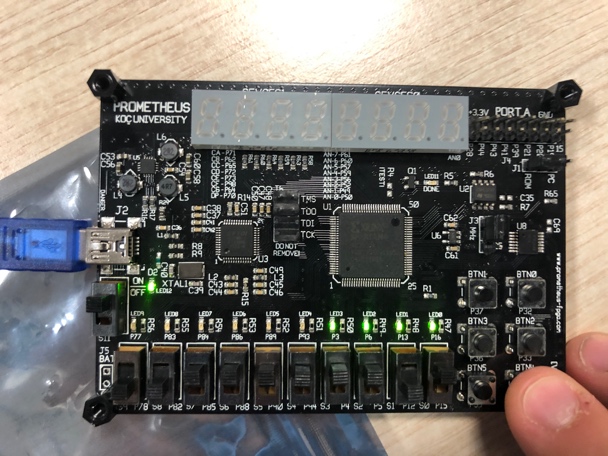
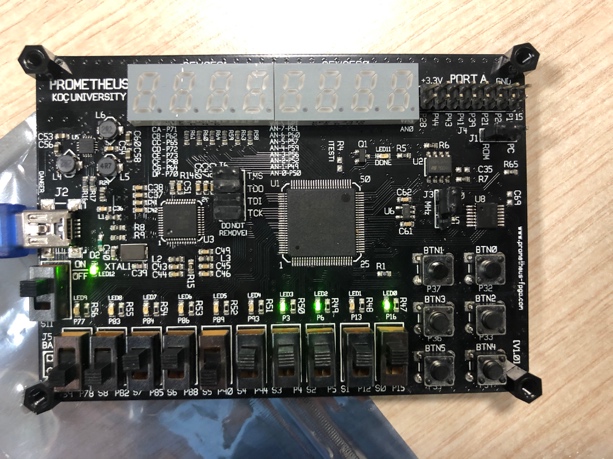
**Figure 6.** RTL forPre-Designed code

1. **Conclusion**

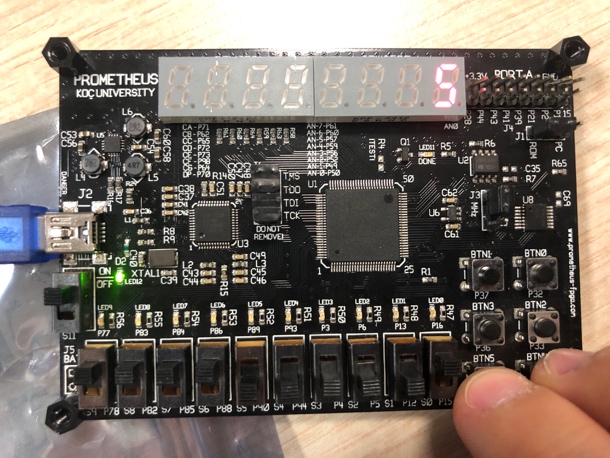
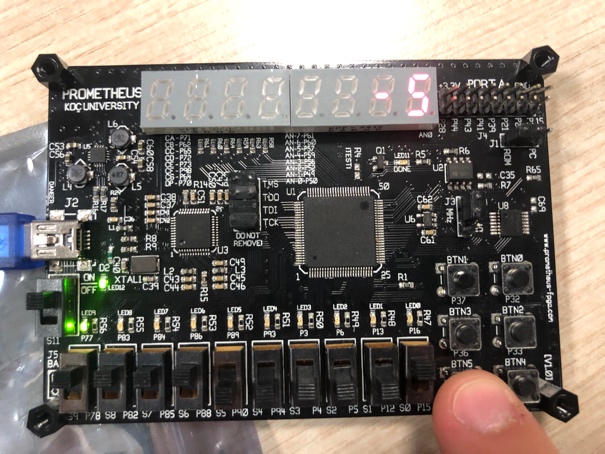
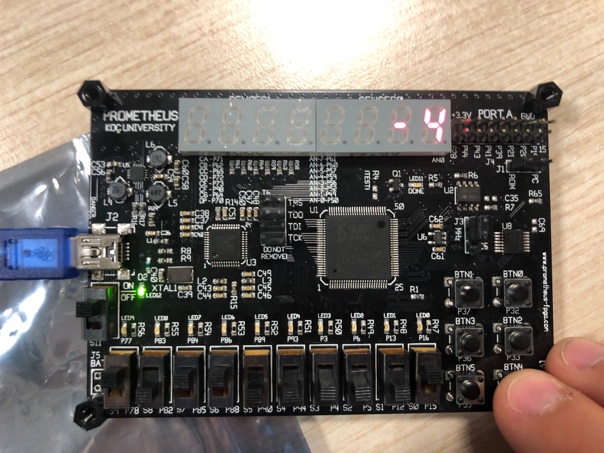
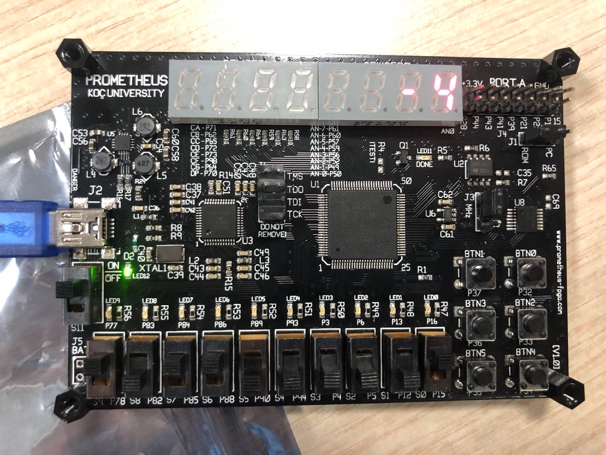
main objective in this experiment was to get familiar with the elementary logic gates and to use them for implementing hierarchical logic circuit design. And I hope that I got that idea very well. Because I can use port mapping and logic gates efficiently after that LAB. Port mapping is kind of boxing our problems, we are giving inputs and outputs inside of this box. Then if don’t want to lose in mess functions we don’t care about what is happening inside of these ports as long as gave right output for us. Moreover, I learned how to use 7-segment display on FPGA board. Now we can experimentally demonstrate the operation of the ALU.

References

1. Marro, Ciletti (2018). Digital Design. New York, NY; Pearson
2. <https://www.youtube.com/watch?v=dYZ-Hwbcnq4&t=852s>
3. <https://www.youtube.com/watch?v=pBcQrLe4IYg>
4. <http://www2.elo.utfsm.cl/~lsb/elo211/aplicaciones/katz/chapter5/chapter05.doc3.html>

****

**Figure 6.** FPGA Boards for Logic Units (A=1111, B=1101)

****

**Figure 7.** FPGA Boards for Arithmetic Units (A=1100, B=0001, Cin=0,0,1,1)