**ELEC 204 Digital Design Lab Report**

Lab 01

Name: Umur Demircioğlu

Date: 13/10/2019

1. **Introduction and objectives**

I designed 3 bit comparator circuit using the elementary logic gates. I used FPGA board, Xilinx ISE software and Prometheus to test my combinational logic design and optimization methodologies were true or not. My code compares two 3 bit numbers starting from the MSB.

1. **Methods**

I have 2 inputs A and B which are 3 bit numbers and 3 outputs E,GAB,GBA which all are 1 bit. E is for equality, GAB is for A bigger than B and GBA for B bigger than A. First I checked if A and B are equal by checking them bit by bit I assigned E0 for LSB being equal, E1 for the middle bit equal or not , E2 for the MSB being equal or not. Then I assigned E to (E0) and (E1) and E(2). So if they are all 1 E is one so both numbers are equal. For A bigger than B or B bigger than A I used the same logic. I will explain for one of them. To check if A is bigger than B we first check if A has 1 and B has 0 for the MSB by writing A(2) and NOT B(2). If this gives 0 than there is a possibility that MSB for both of them are equal so we check the second bit by writing A(1) and NOT B(1) and E2. If this one also gives 0 there is a possibility that the MSB and the second bit being equal so we check the LSB by writing (E2 and E1 and ANB0). We combine this three possibilities with OR gates so even if one of them gives 1 A is bigger than B. The same logic applies for B bigger than A.

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

In this lab I had no problems with the code because the logic behind it was simple however I had problems while uploading the bit file to the board.

1. **Conclusion**

In this experiment we got used to using Xilinx ISE software, Prometheus and the FPGA board. We learned more about different logic gates and how to create logic designs and how to verify if they are true or not using FPGA board and from simulation programs.

Code

entity MComparator is

Port ( A : in  STD\_LOGIC\_VECTOR (2 downto 0);

B : in  STD\_LOGIC\_VECTOR (2 downto 0);

E : out  STD\_LOGIC;

GAB : out  STD\_LOGIC;

GBA : out  STD\_LOGIC);

end MComparator;

architecture Behavioral of MComparator is

signal E0 : STD\_LOGIC;

signal E1 : STD\_LOGIC;

signal E2 : STD\_LOGIC;

signal BNA0 : STD\_LOGIC;

signal BNA1 : STD\_LOGIC;

signal BNA2 : STD\_LOGIC;

signal ANB0 : STD\_LOGIC;

signal ANB1 : STD\_LOGIC;

signal ANB2 : STD\_LOGIC;

begin

E0 <= (A(0) and B(0)) or (NOT A(0) and NOT B(0));

E1 <= (A(1) and B(1)) or (NOT A(1) and NOT B(1));

E2 <= (A(2) and B(2)) or (NOT A(2) and NOT B(2));

ANB0 <= A(0) and NOT B(0);

ANB1 <= A(1) and NOT B(1);

ANB2 <= A(2) and NOT B(2);

BNA0 <= B(0) and NOT A(0);

BNA1 <= B(1) and NOT A(1);

BNA2 <= B(2) and NOT A(2);

E <= E0 and E1 and E2;

GAB <= ANB2 or (E2 and ANB1) or (E2 and E1 and ANB0);

GBA <= BNA2 or (E2 and BNA1) or (E2 and E1 and BNA0);

end Behavioral;