

ECEN 429: Introduction to Digital Systems Design Laboratory

North Carolina Agricultural and Technical State University

Department of Electrical and Computer Engineering

Ian Parker (Reporter)

Tayanna Lee (Lab Partner)

January 24, 2018

Lab #1

Introduction

For this lab we familiarized ourselves with Vivado and the Basys3 FPGA Development Board. We learned how to set up the Vivado for the Basey 3 board and how to load programs onto the board.

In part 1 we wrote a VHDL program using an AND gate with two inputs and one output. For part 2 we wrote a VHDL program using four inputs that are one bit each and creating a variable to represent it. For part 3 we had to find an equation to implement a the given circuit in VHDL. By completing the lab we showed the ability to take circuit designs and implement them using VHDL code using Vivado and the Basys3 board.

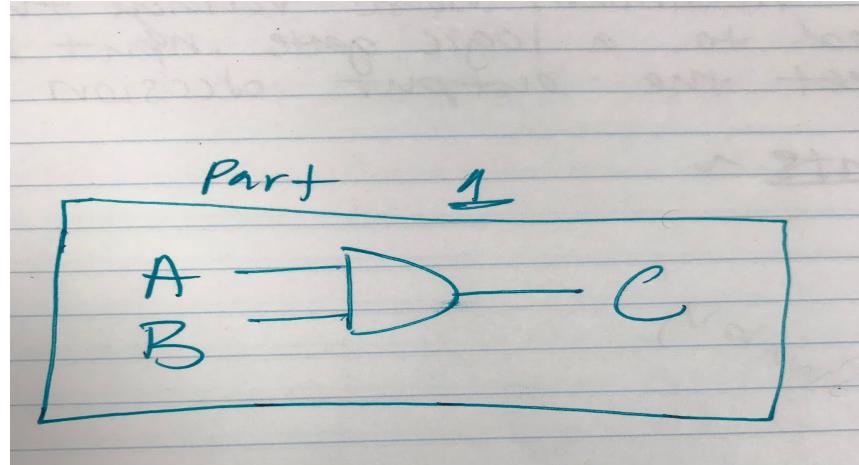
Background, Design Solution, and Results

Part 1: Design an AND gate with 2 inputs and 1 output bits.

Truth Table 1

Inputs		Outputs
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

Schematic



FPGA Pin Assignment

I/O	Pin Assignments
A	V17
B	V16
C	U16

Results

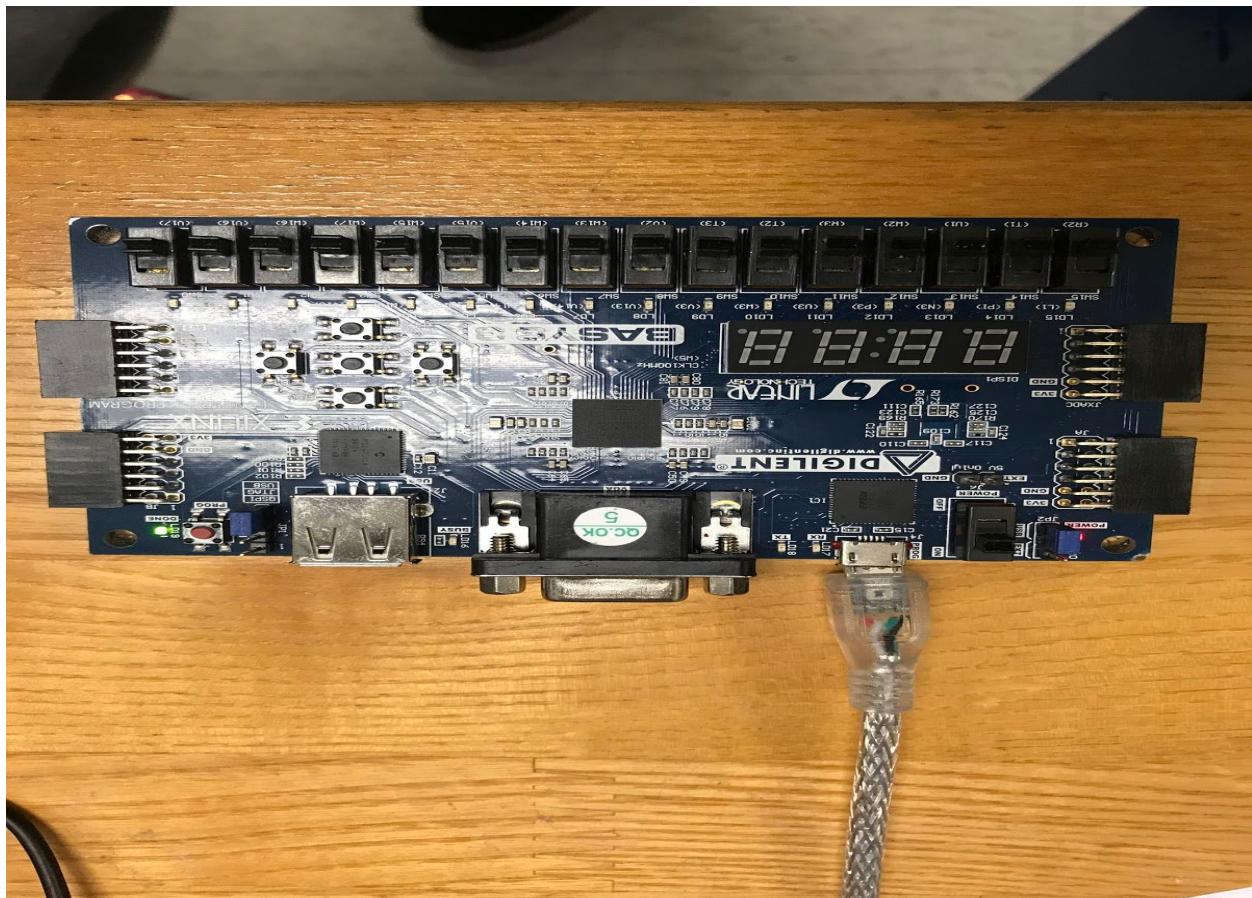


Figure 1.1: When $A = 0$ and $B=0$, and are AND together output $C = 0$.

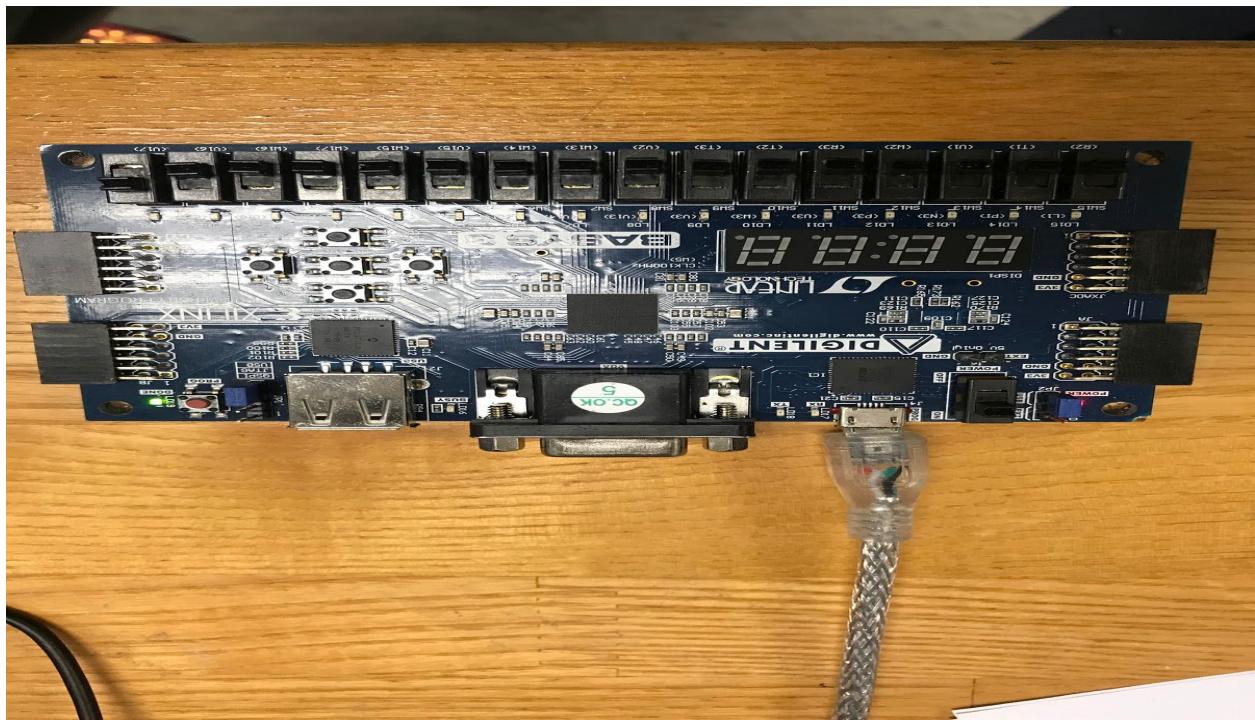


Figure 1.2: When $A = 0$ and $B=1$, and are AND together output $C = 0$.

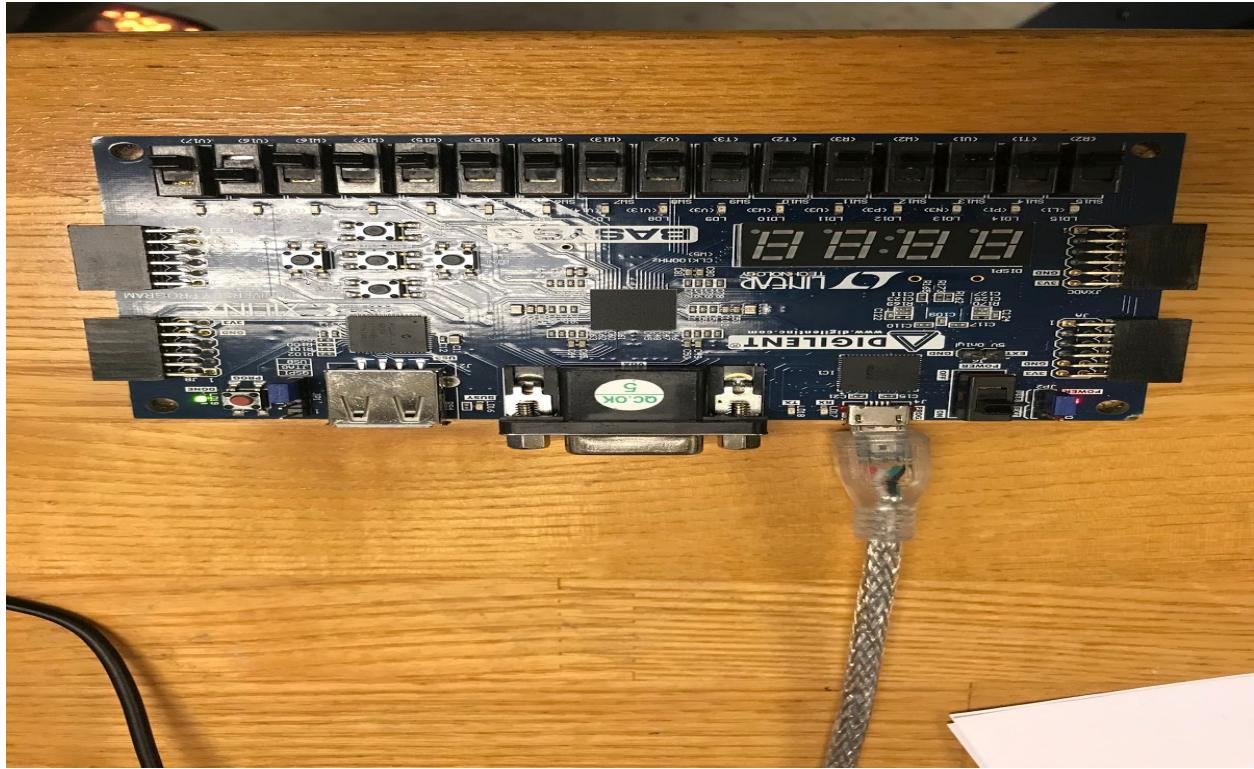


Figure 1.3: When A = 1 and B=0, and are AND together output C = 0.

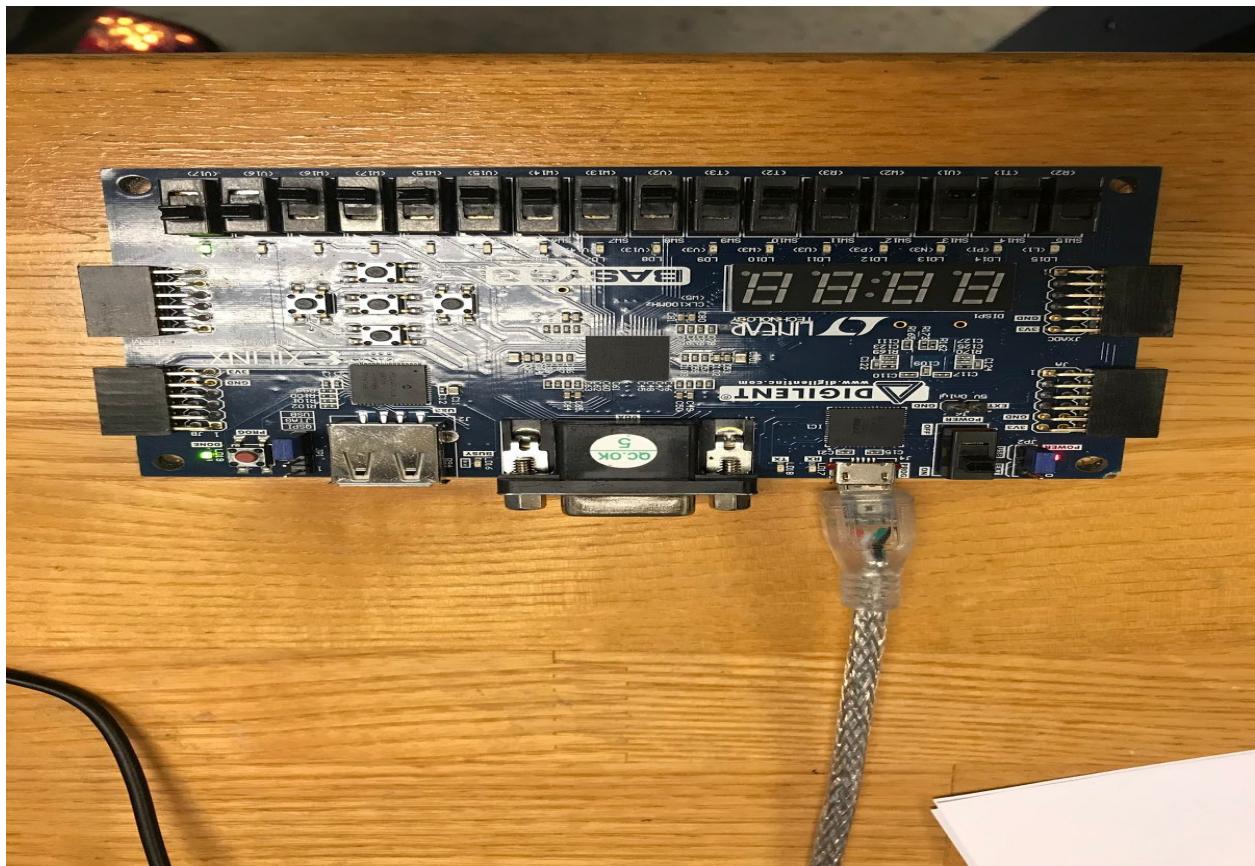


Figure 1.4: When $A = 1$ and $B=1$, and are AND together output $C = 1$.

Part 2:

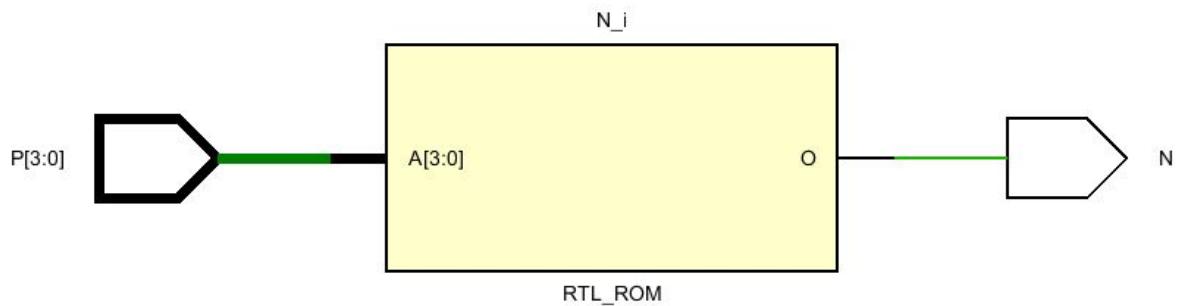
Using VHDL , design four inputs that are one bit each and create a variable to represent it and implement it on the Bayes 3.

The resulting output lights up since the 4 bit number is a prime number.
four inputs that are one bit each and creating a variable to represent it.

Truth Table 2

P	N
0000	0
0001	0
0010	1
0011	1
0100	0
0101	1
0110	0
0111	1
1000	0
1001	0
1010	0
1011	1
1100	0
1101	1
1110	0
1111	0

Schematic



Pin Assignment

I/O	Pin Assignment
P[0]	V17
P[1]	V16
P[2]	W16
P[3]	W17
N	U16

Result

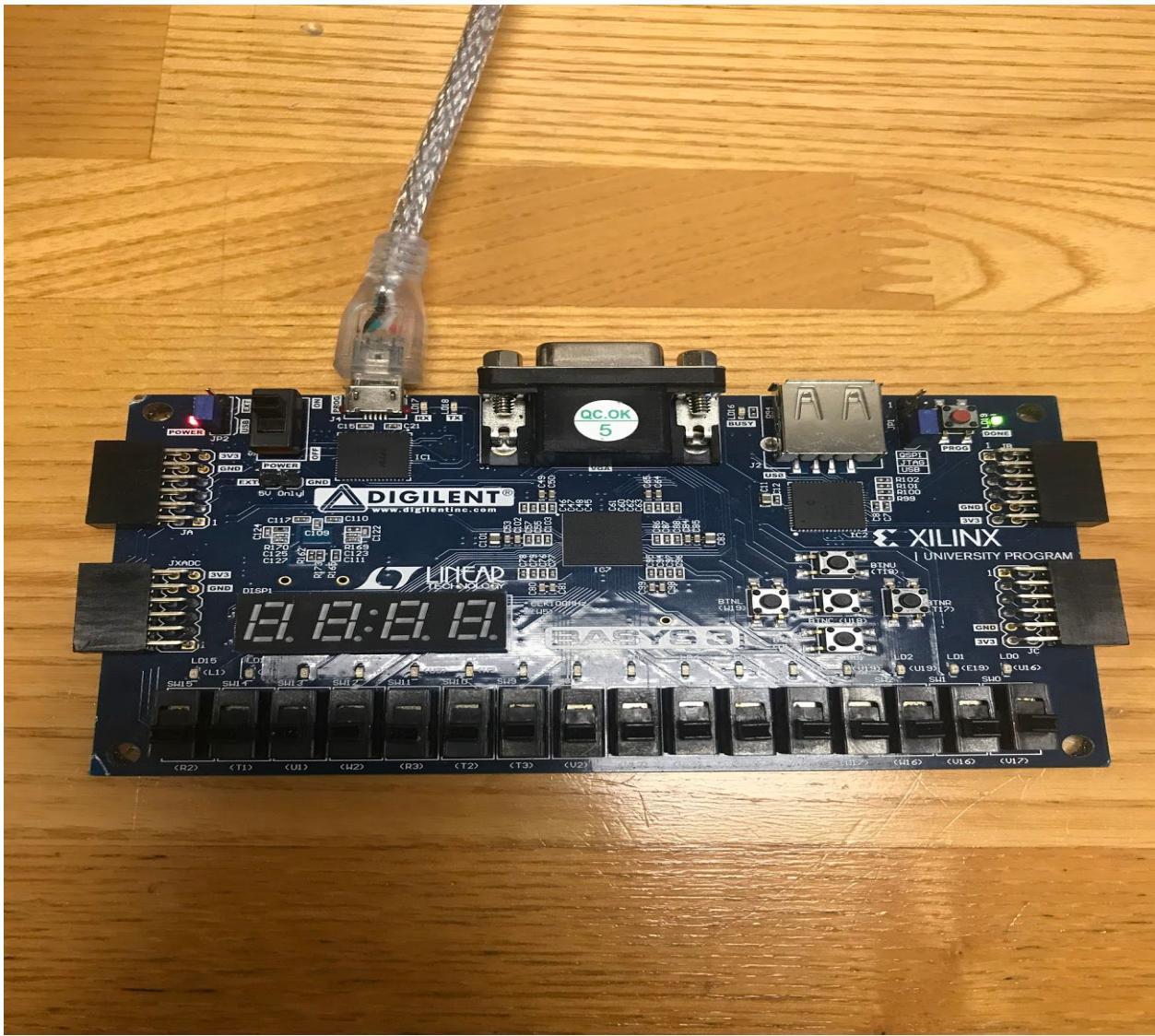


Figure 2.1 When P = “0000”, N is “0”

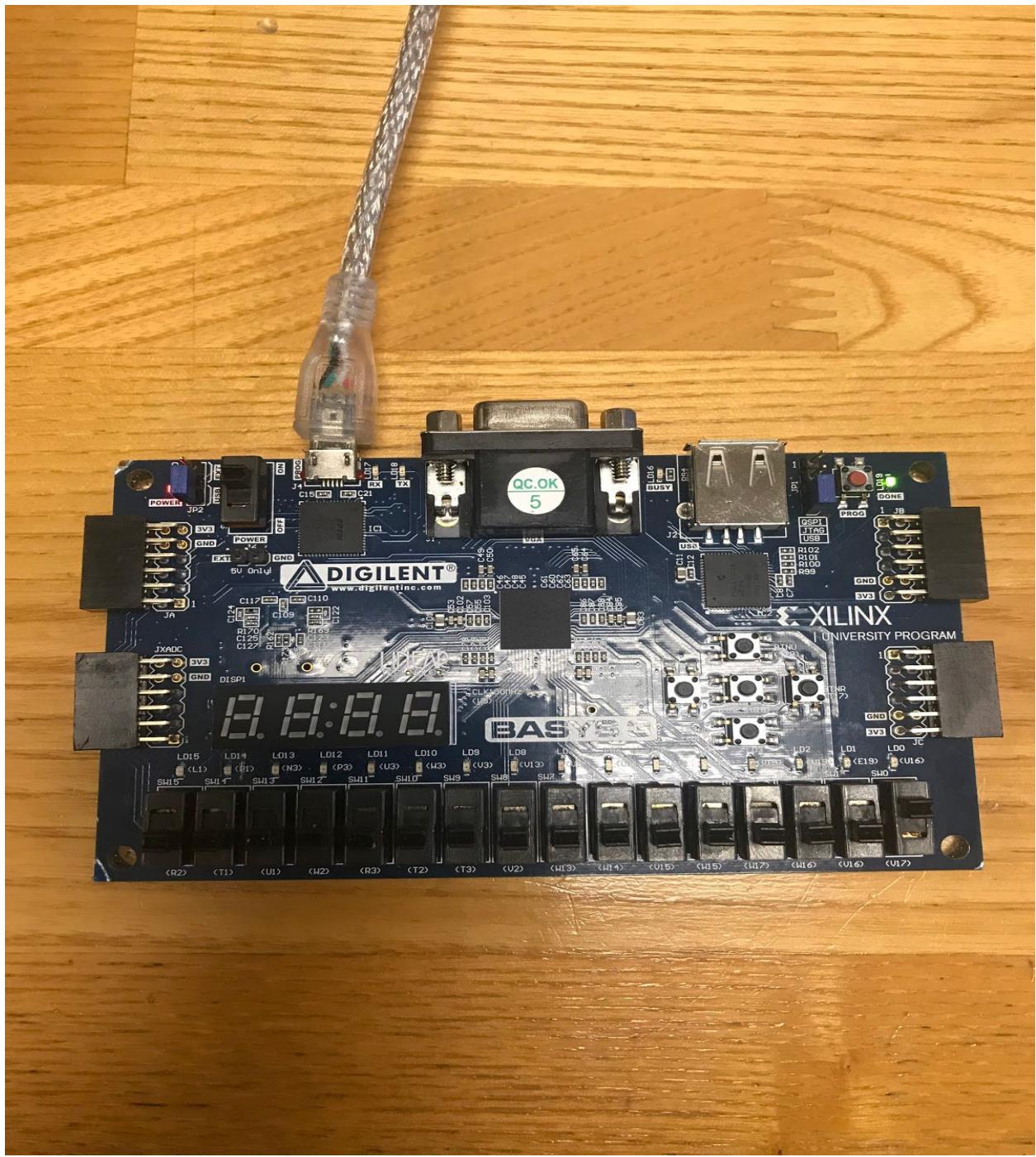


Figure 2.2 When P = “0001”, N is “0”

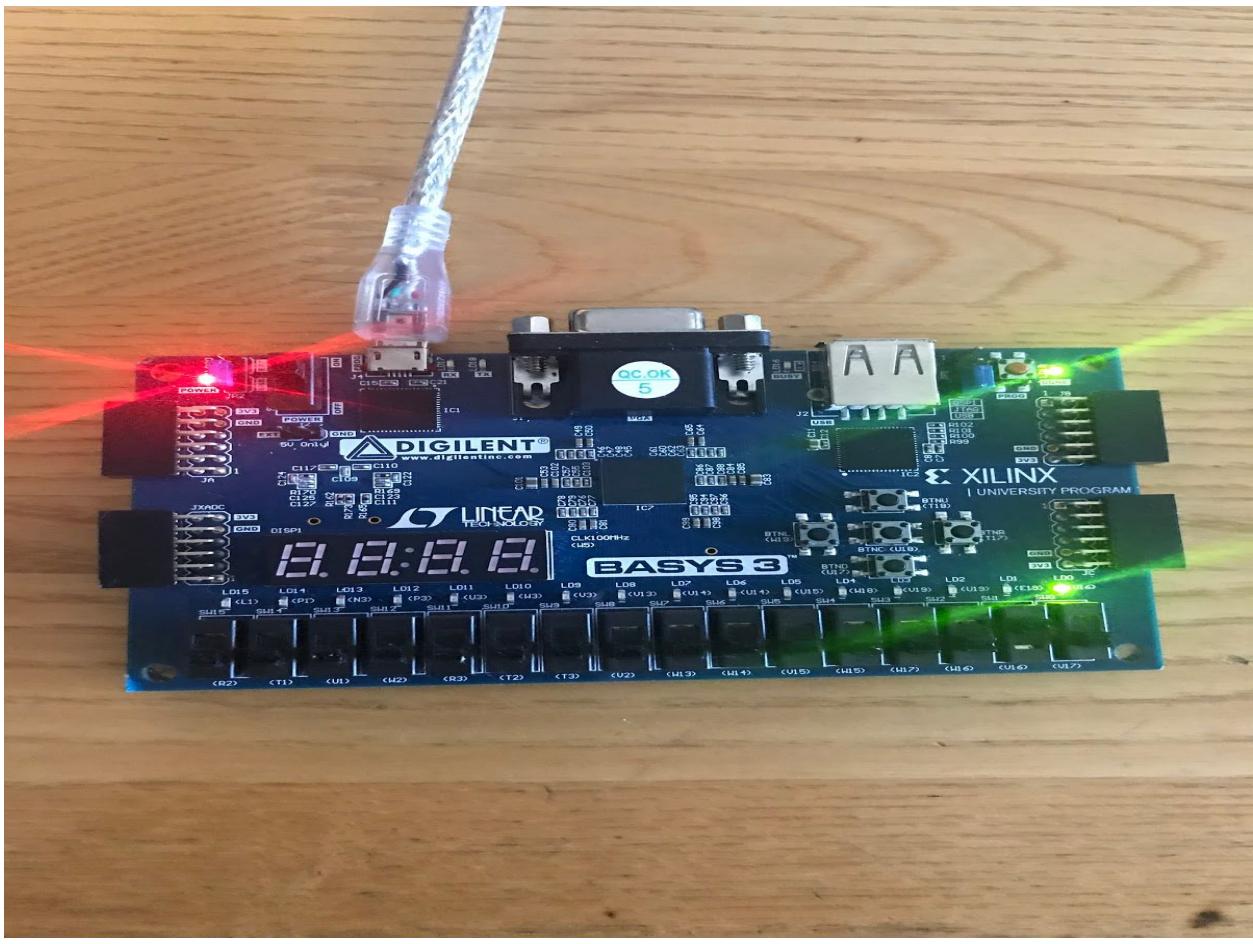


Figure 2.3 When P = “0010”, N is “1”

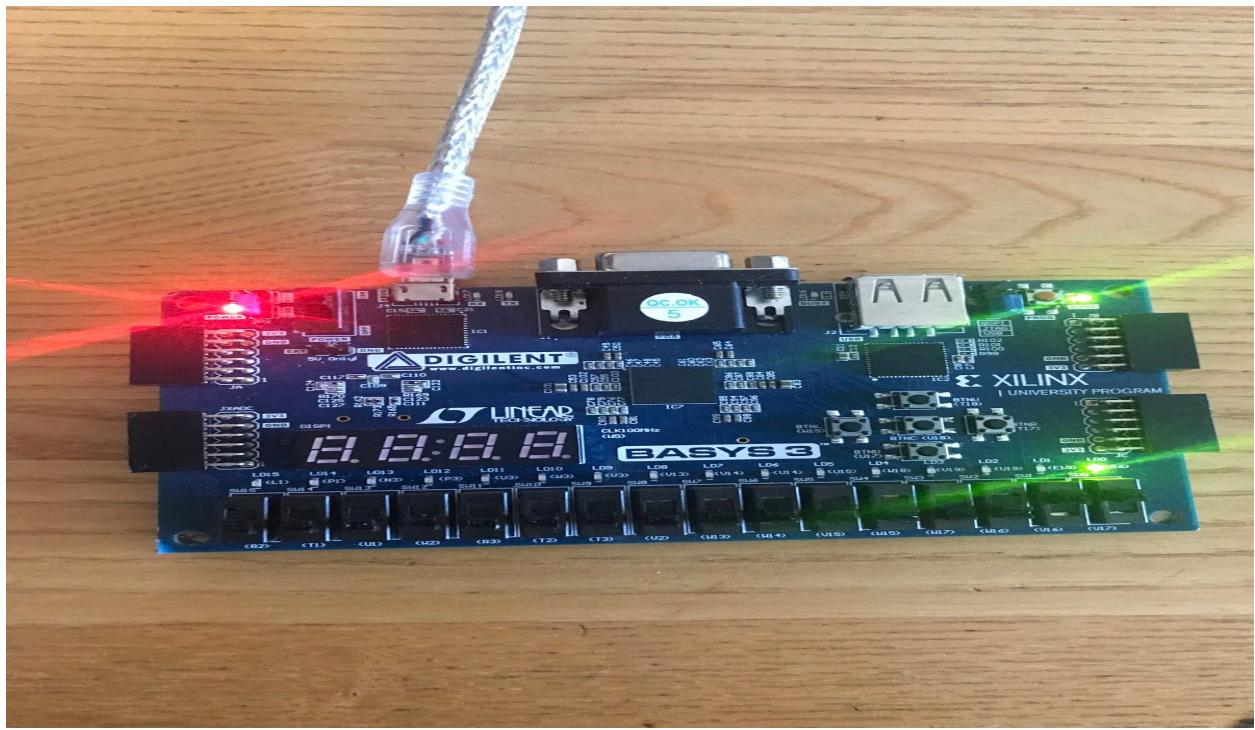


Figure 2.4 When P = “0011”, N is “1”

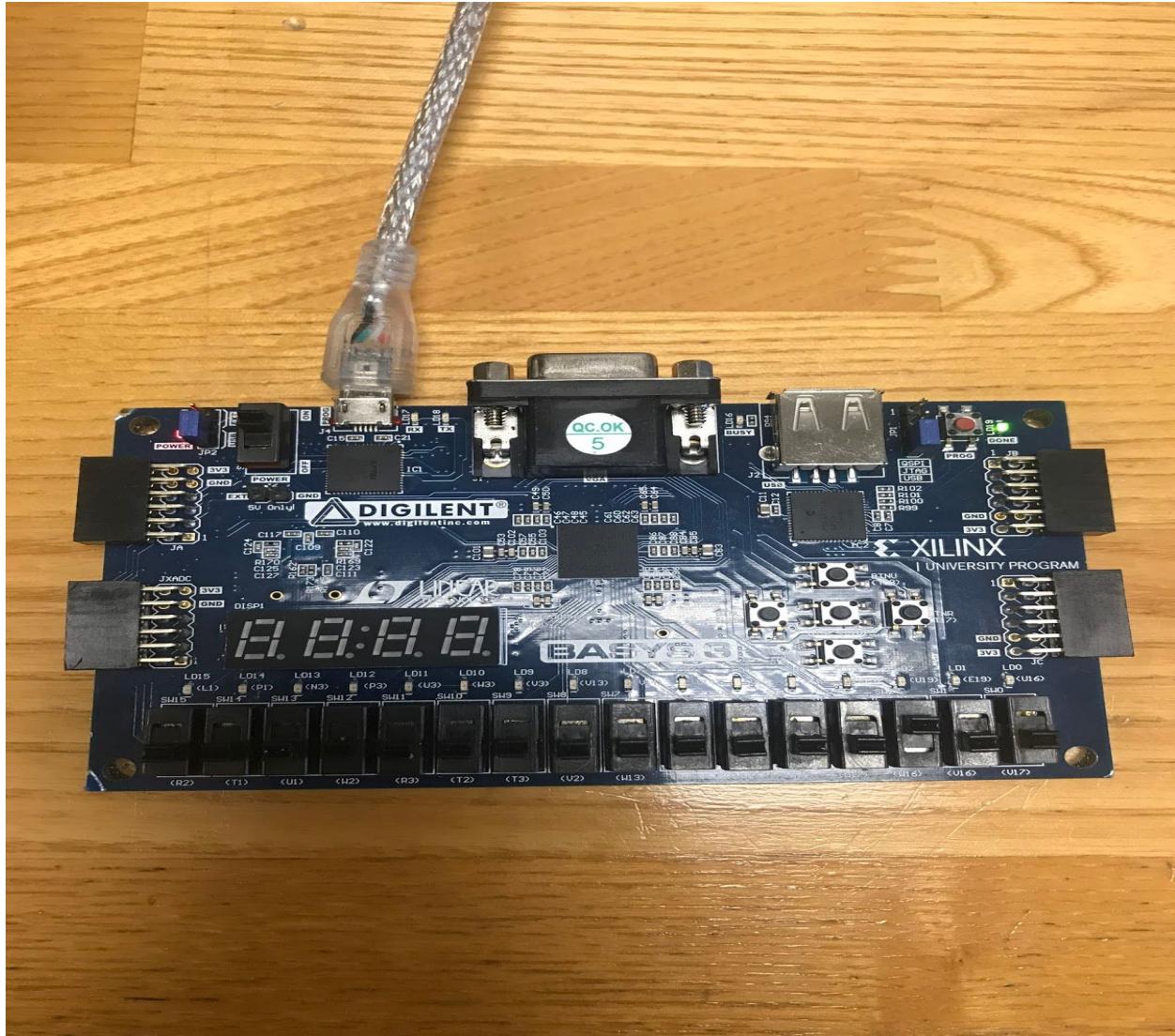


Figure 2.5 When P = “0100”, N is “0”



Figure 2.6 When P = “0101”, N is “1”

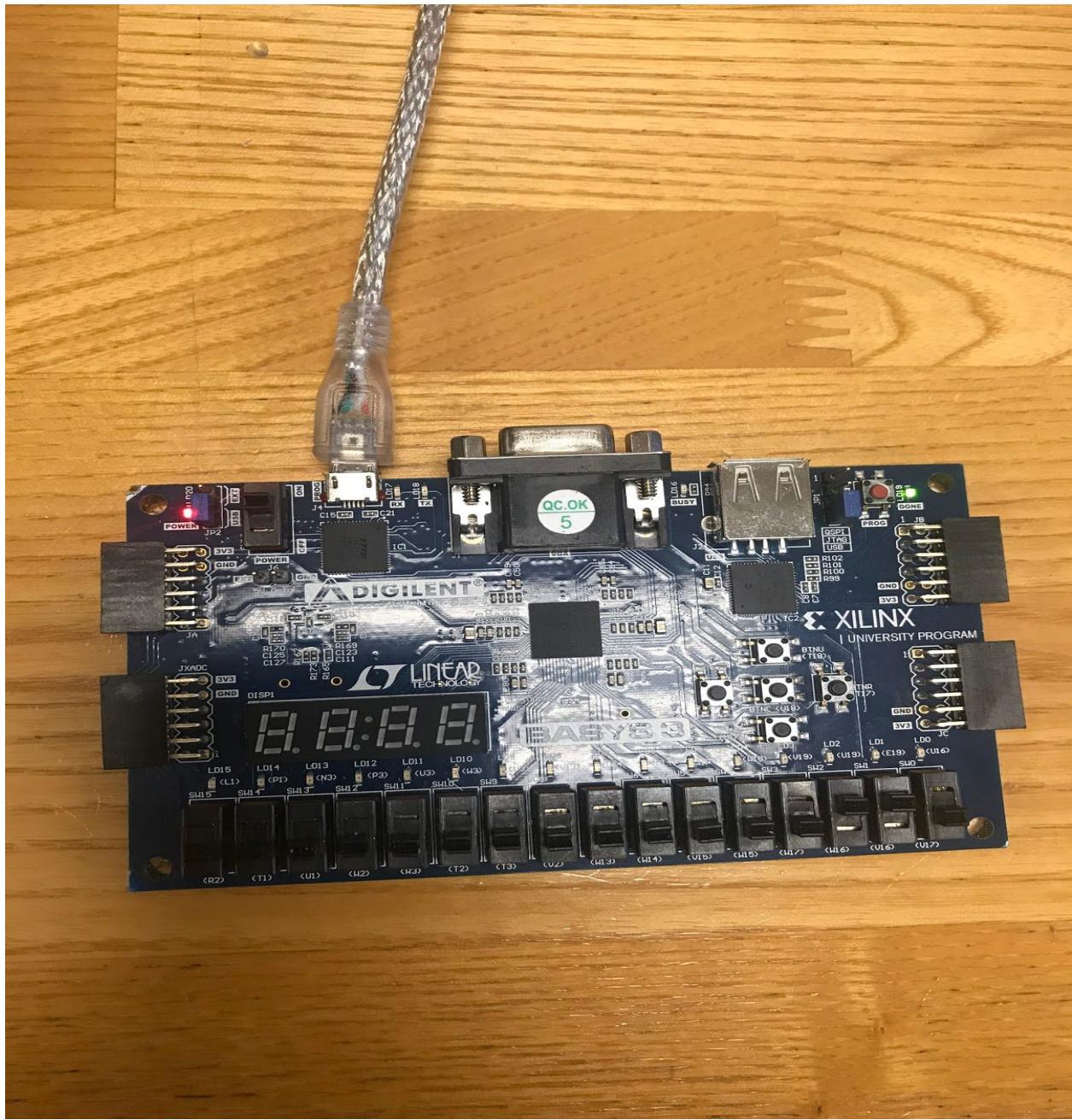


Figure 2.7 When P = "0110", N is "0"

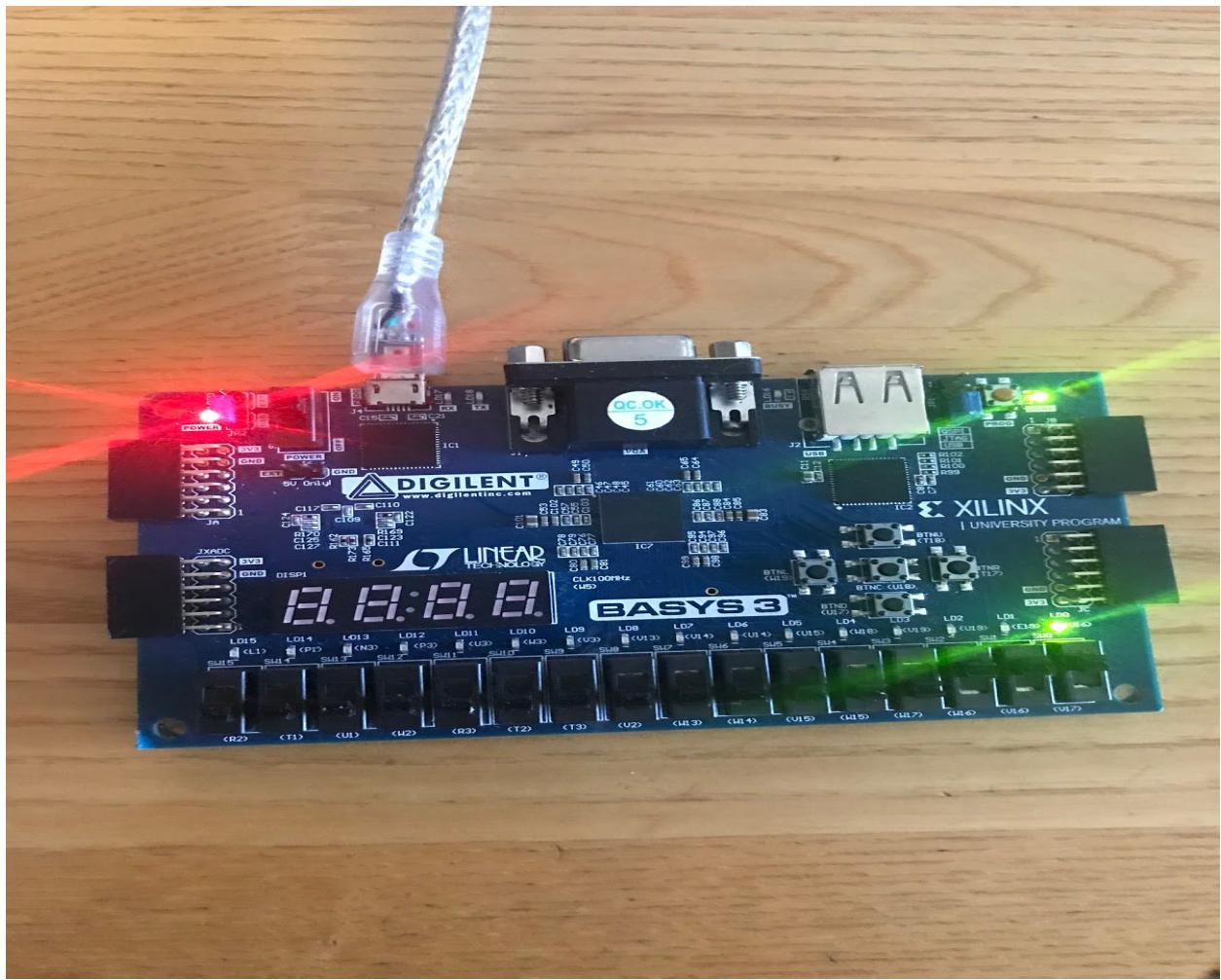


Figure 2.8 When P = “0111”, N is “1”

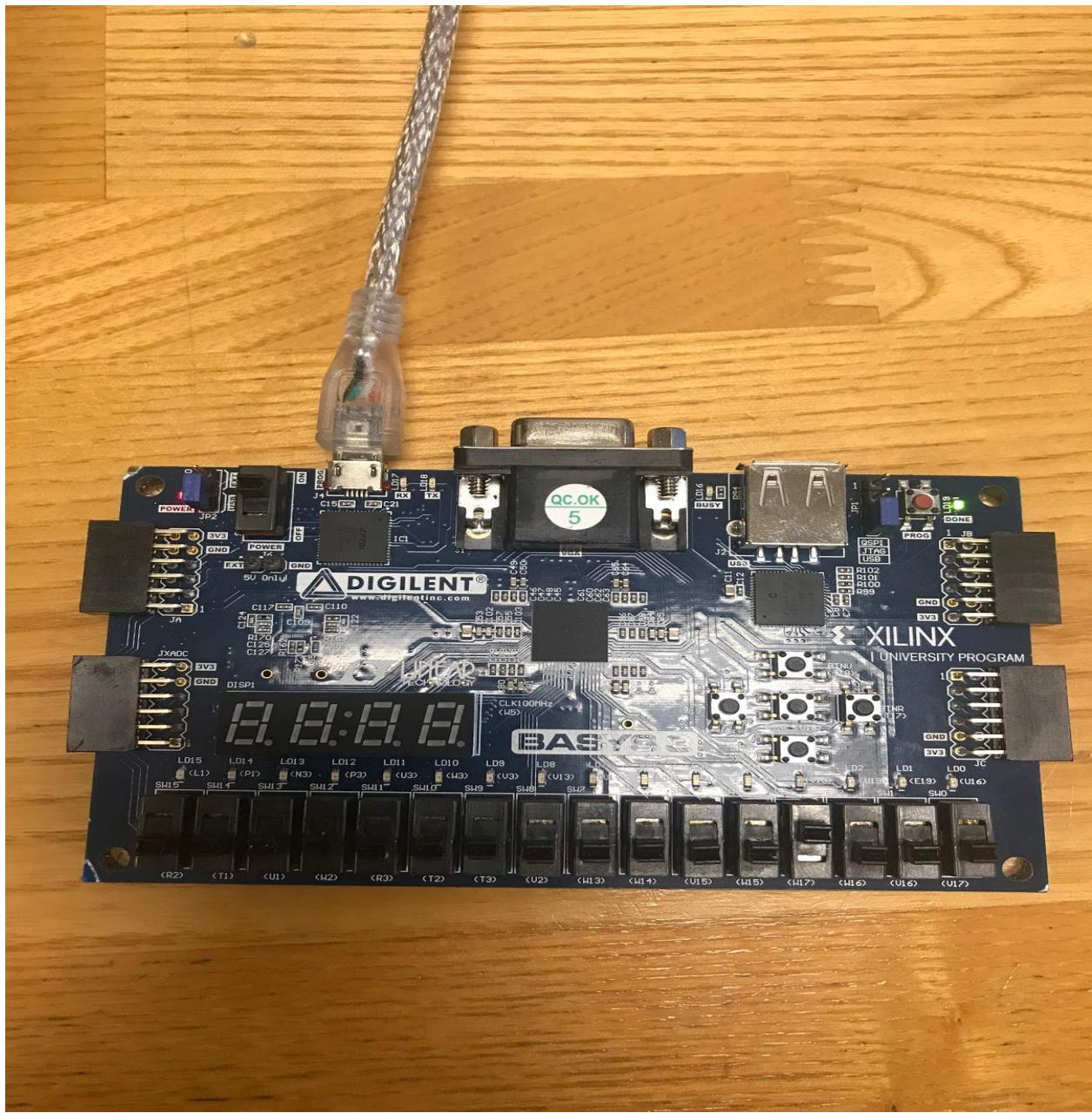


Figure 2.9 When P = “1000”, N is “0”

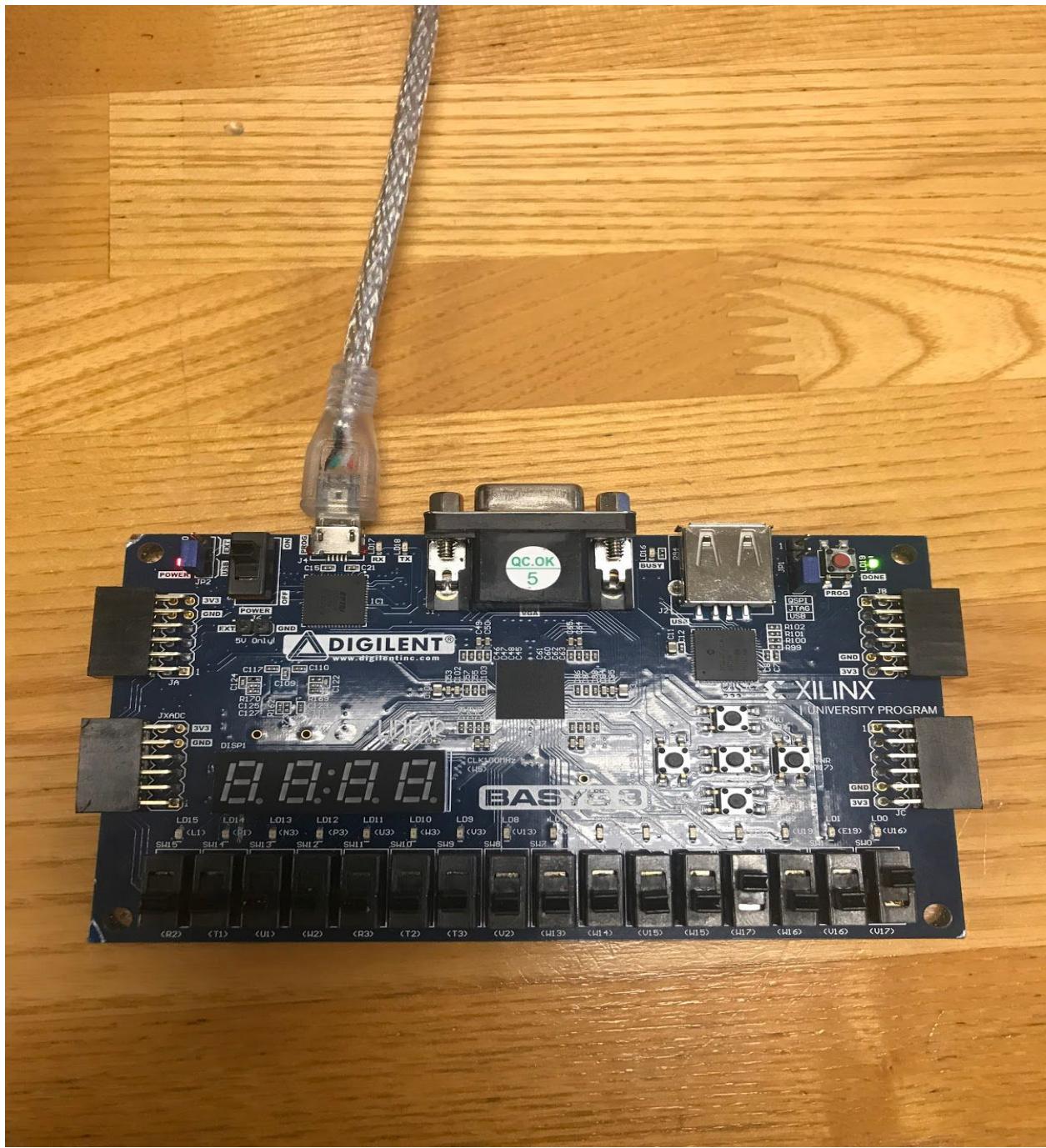


Figure 2.10 When P = “1001”, N is “0”

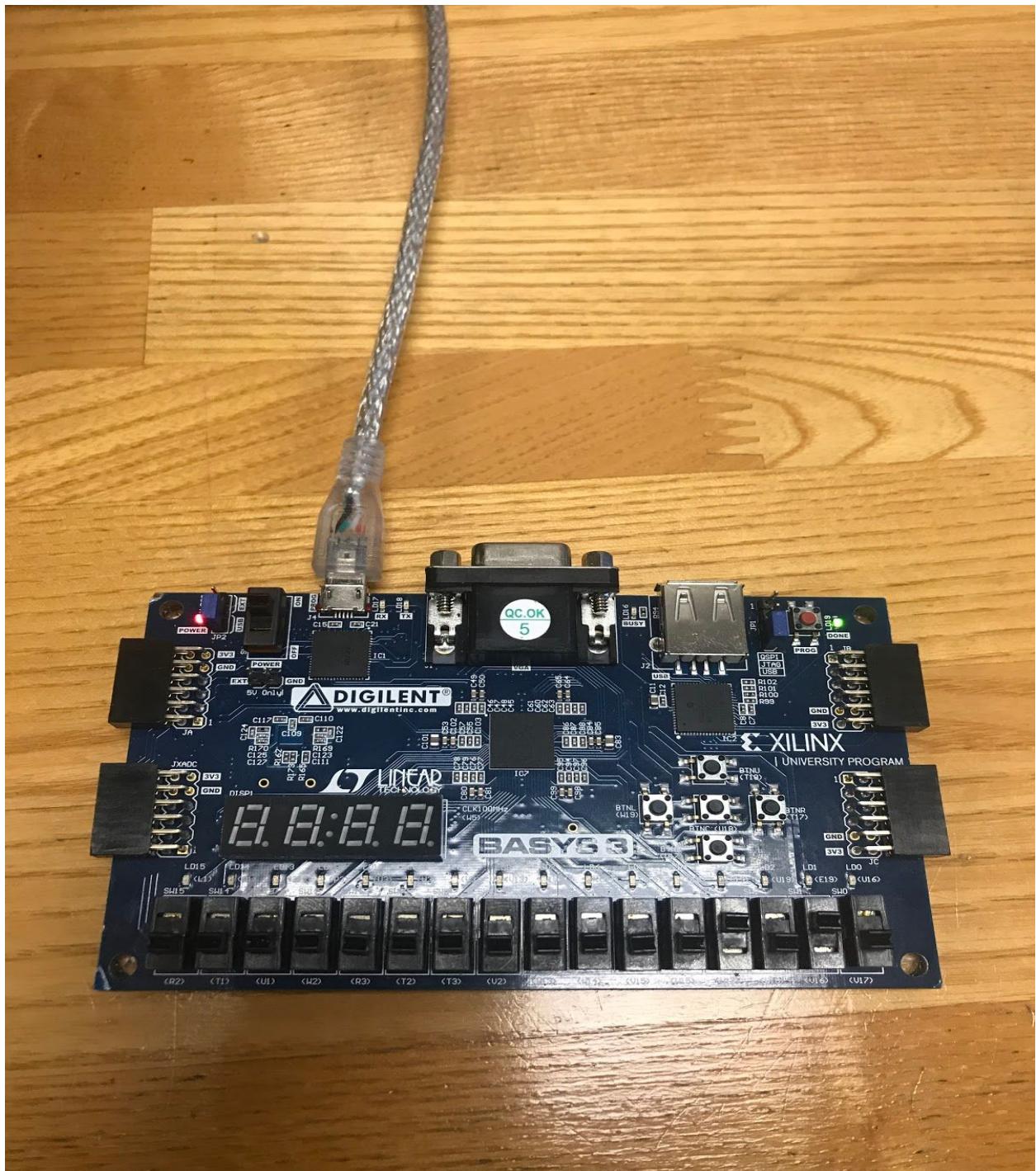


Figure 2.11 When P = “1010”, N is “0”

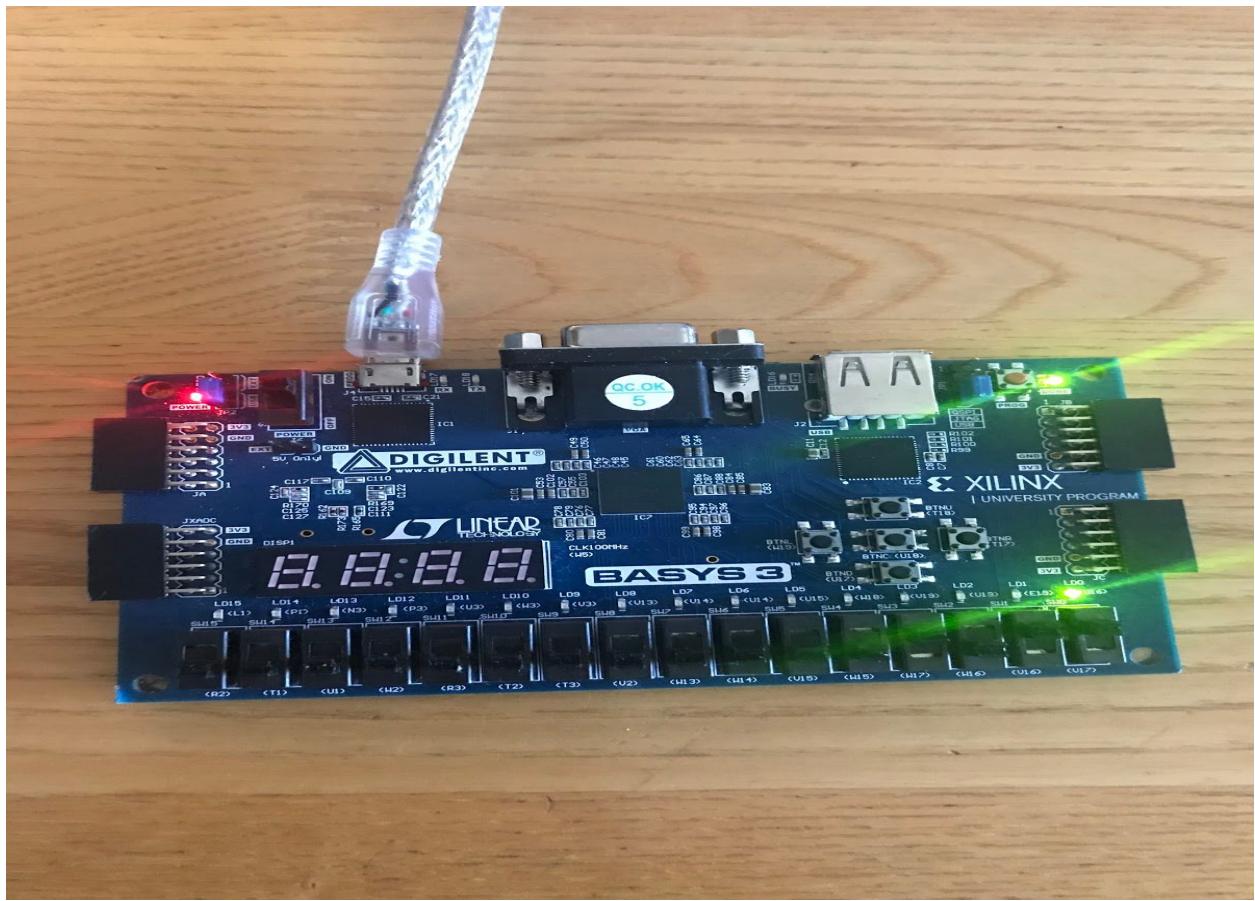


Figure 2.12 When $P = "1011"$, N is " 1 "

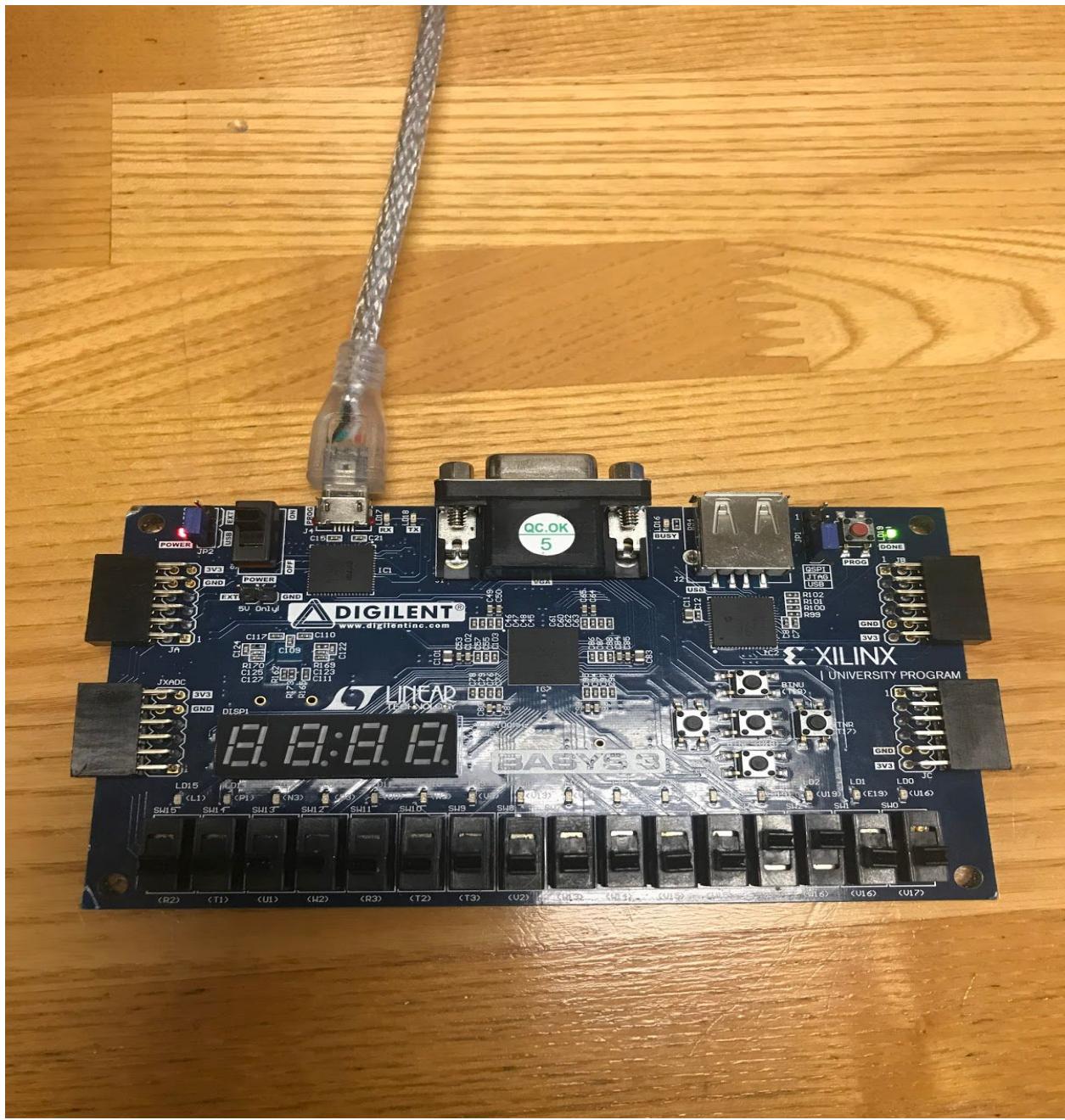


Figure 2.13 When P = "1100", N is "0"

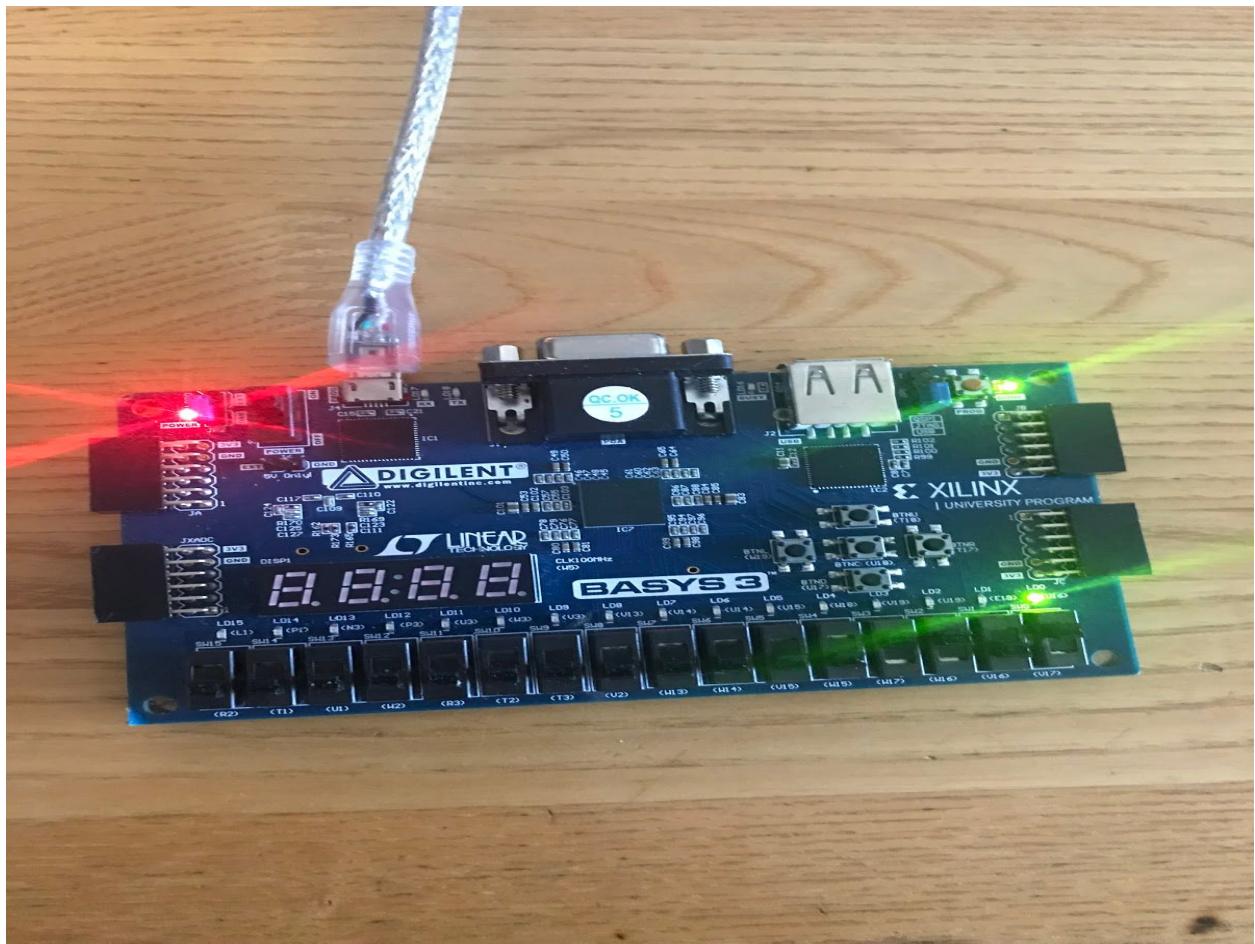


Figure 2.14 When P = “1101”, N is “1”

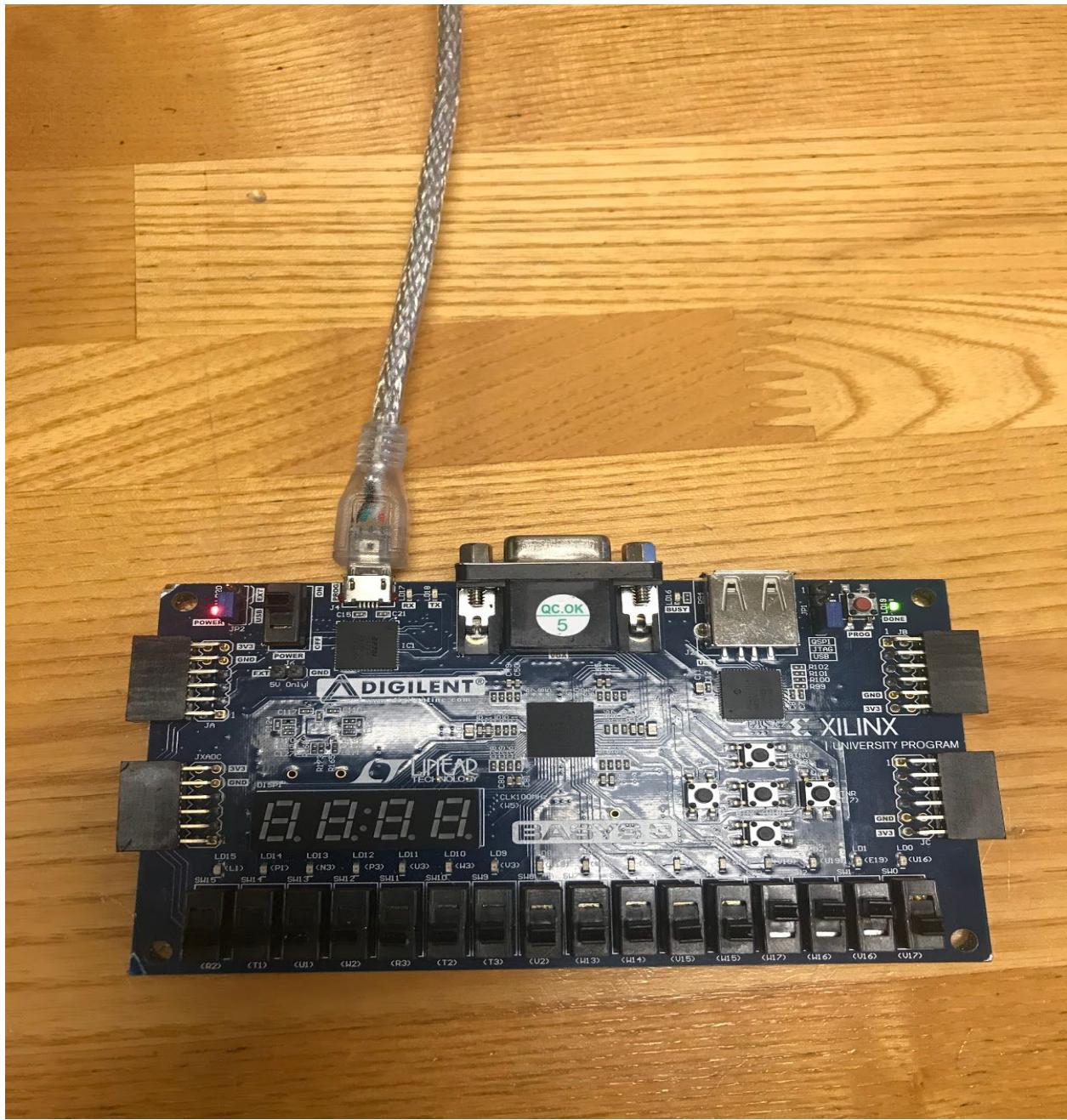


Figure 2.15 When P = “1110”, N is “0”

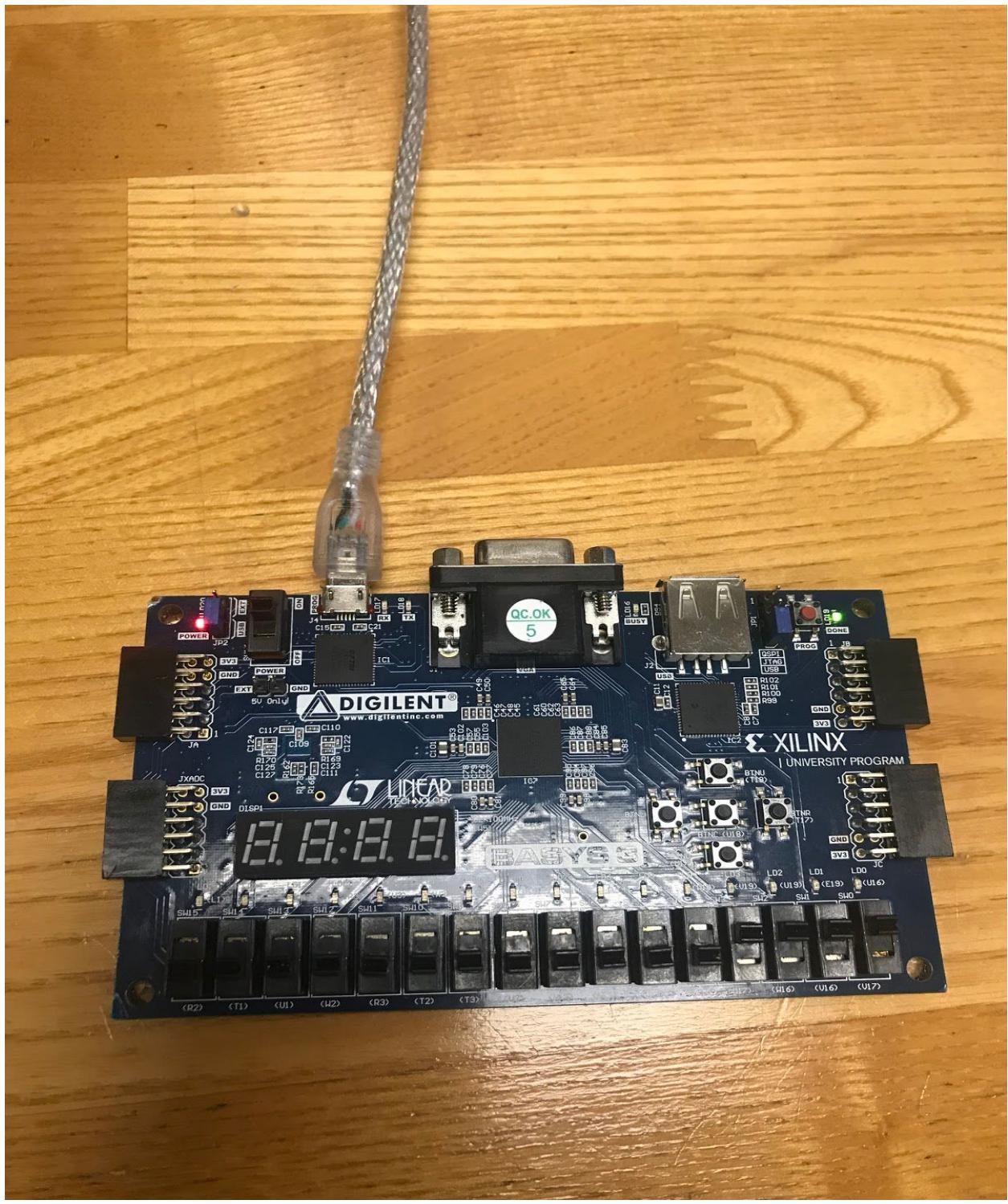


Figure 2.16 When P = “1111”, N is “0”

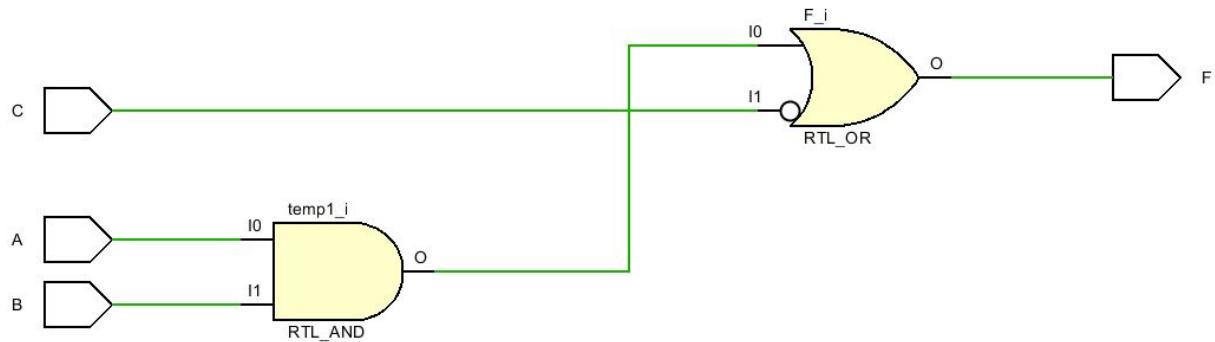
Part 3:

Finding the equation for the output the circuit is implemented in VHDL.

Truth Table 3

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Schematic



Pin Assignment

I/O	Pin Assignments
A	V17
B	V16
C	W16
F	U16

Result

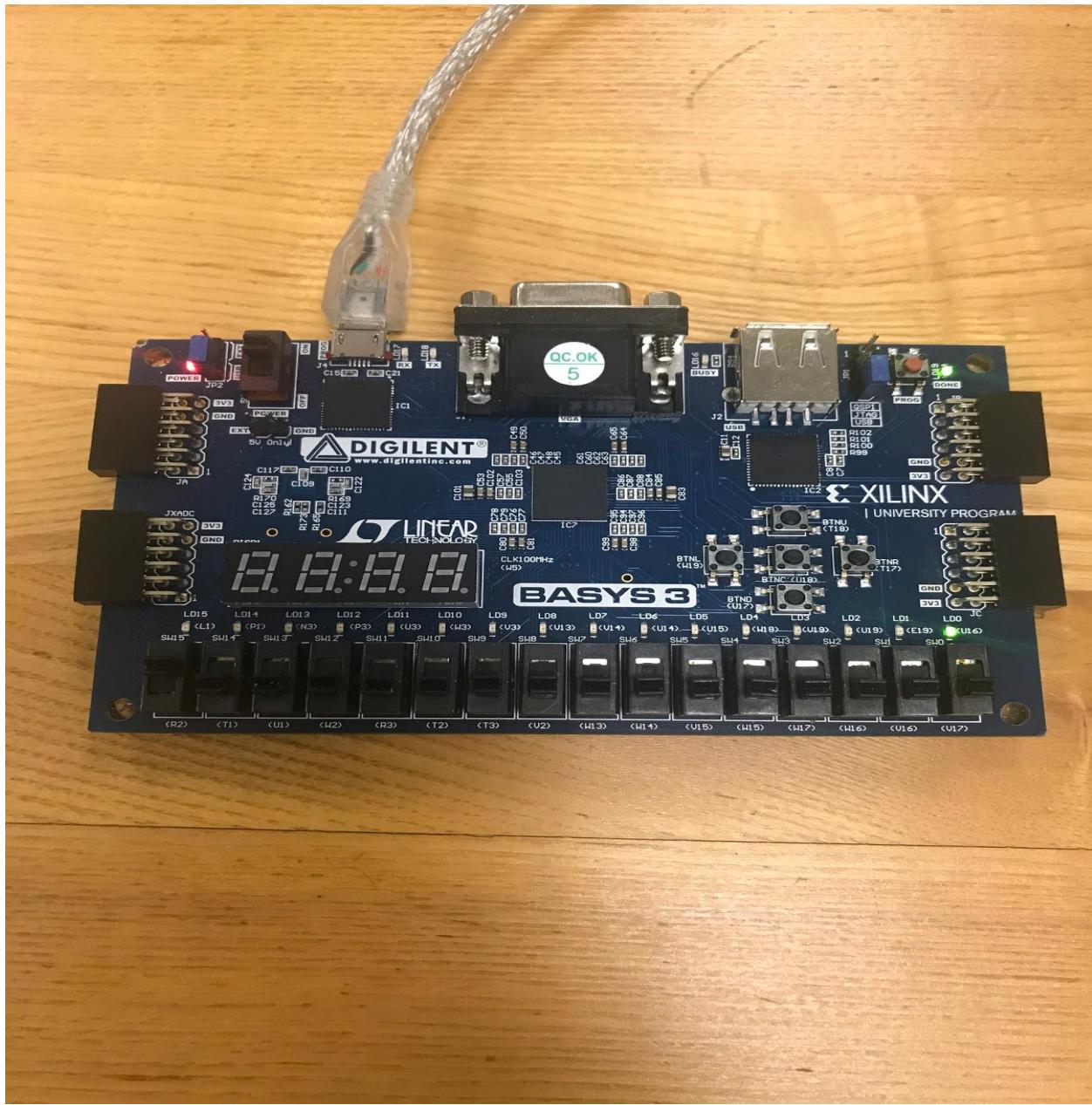


Figure 3.1 When $A=0$ and $B=0$ and are AND together and $C =0$ is NOT C, and their respective outputs are OR together, the output $F=1$.

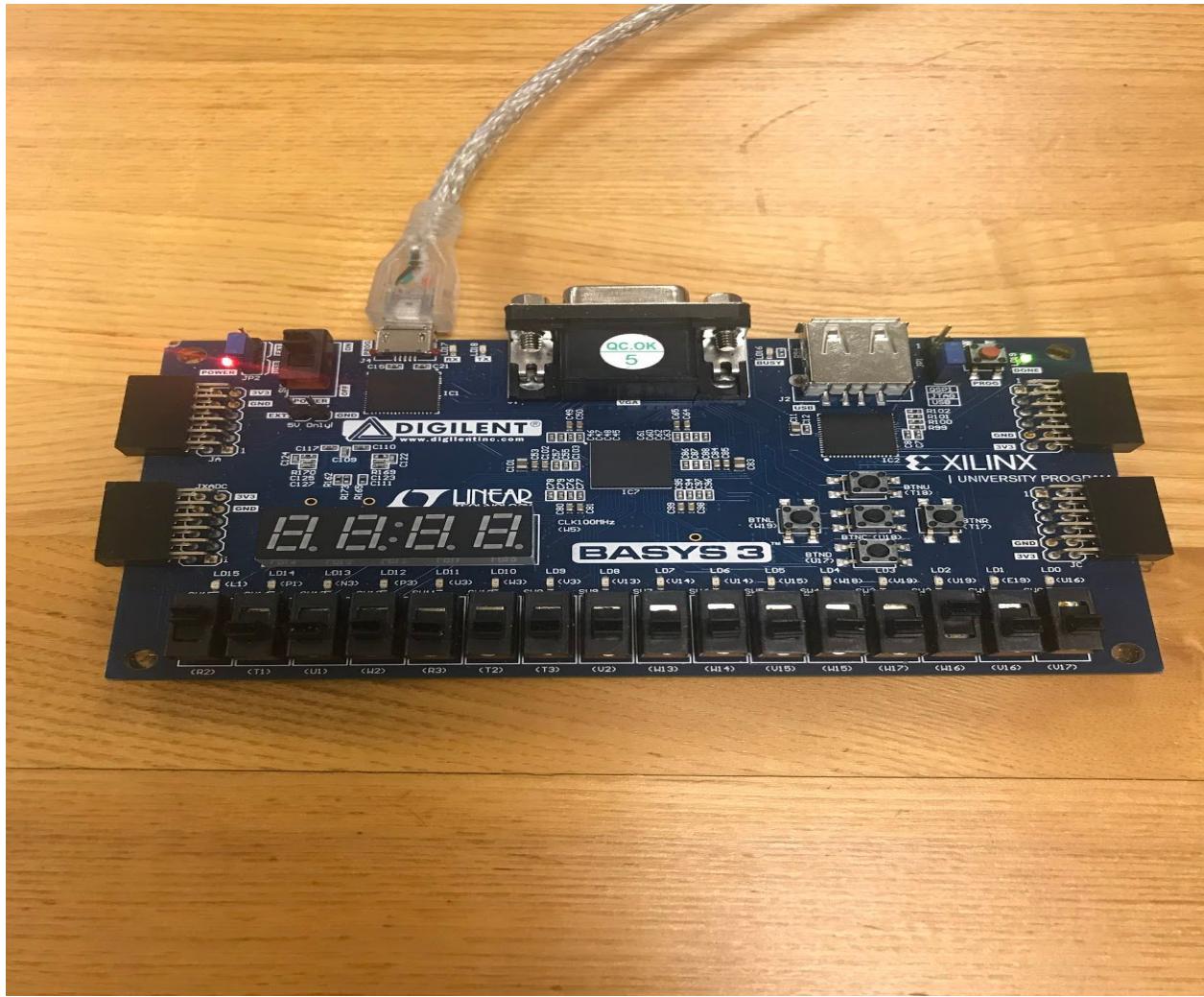


Figure 3.2 When $A=0$ and $B=0$ and are AND together and $C = 1$ is NOT C, and their respective outputs are OR together, the output $F=0$.

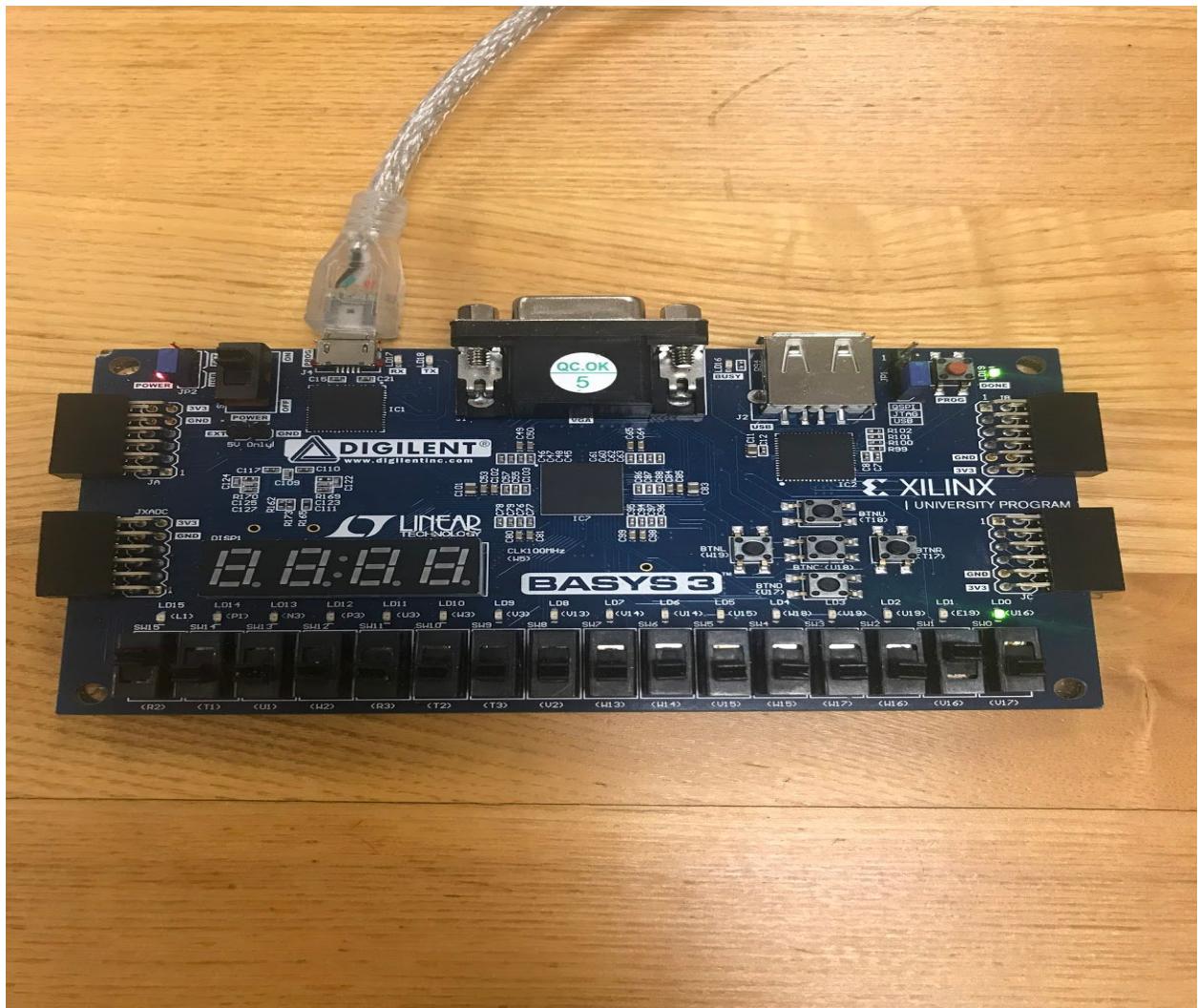


Figure 3.3 When $A=0$ and $B=1$ and are AND together and $C =0$ is NOT C, and their respective outputs are OR together, the output $F=1$.

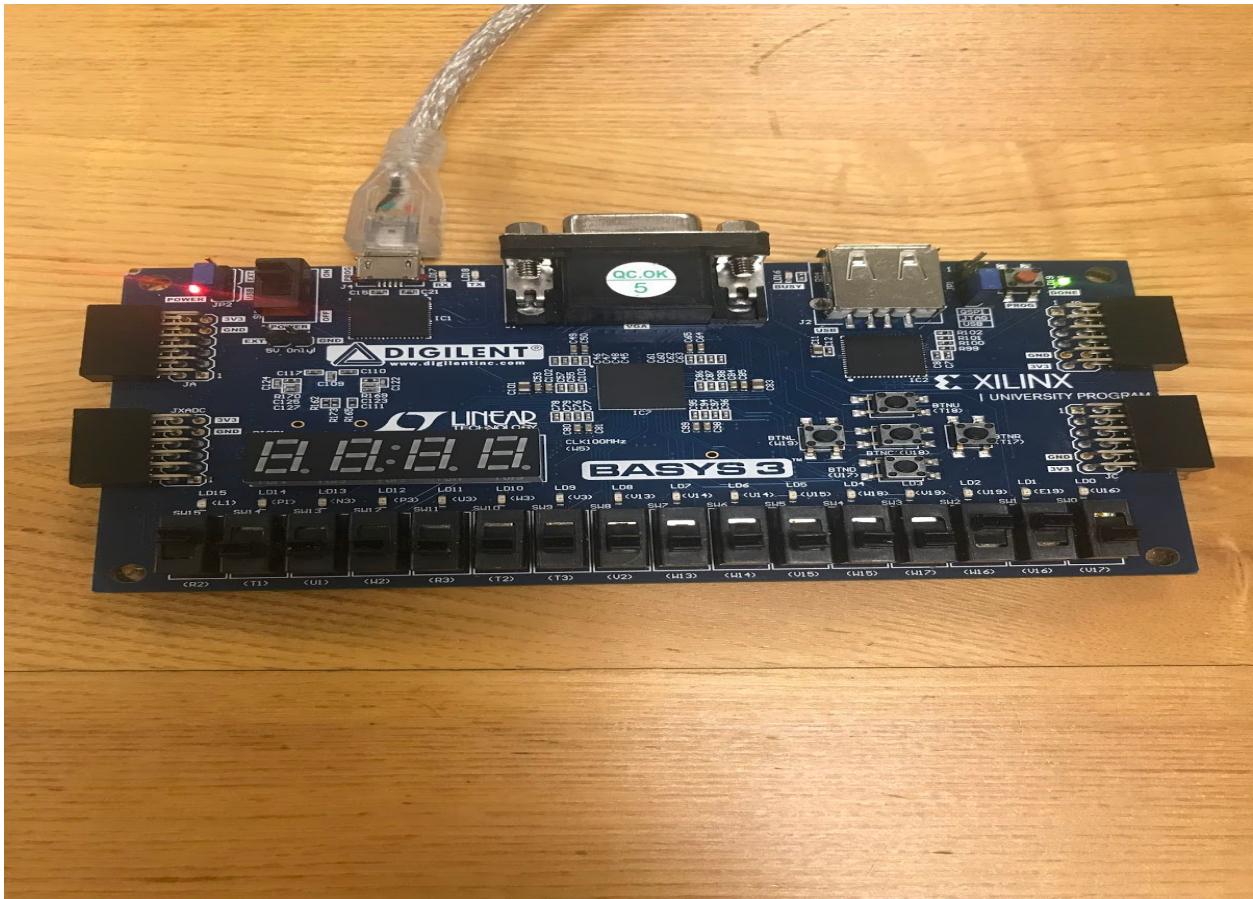


Figure 3.4 When $A=0$ and $B=1$ and are AND together and $C = 1$ is NOT C, and their respective outputs are OR together, the output $F=0$.

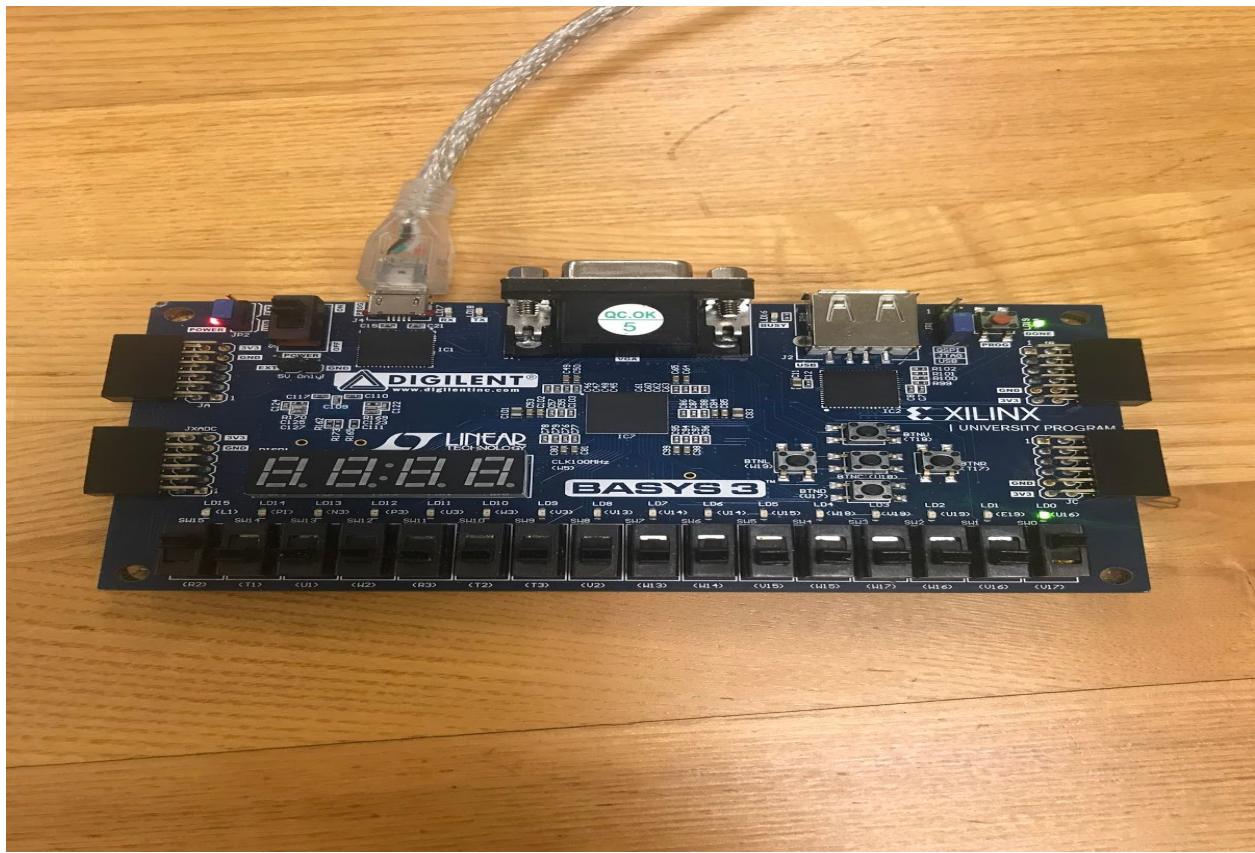


Figure 3.5 When $A=1$ and $B=0$ and are AND together and $C = 0$ is NOT C, and their respective outputs are OR together, the output $F=1$.

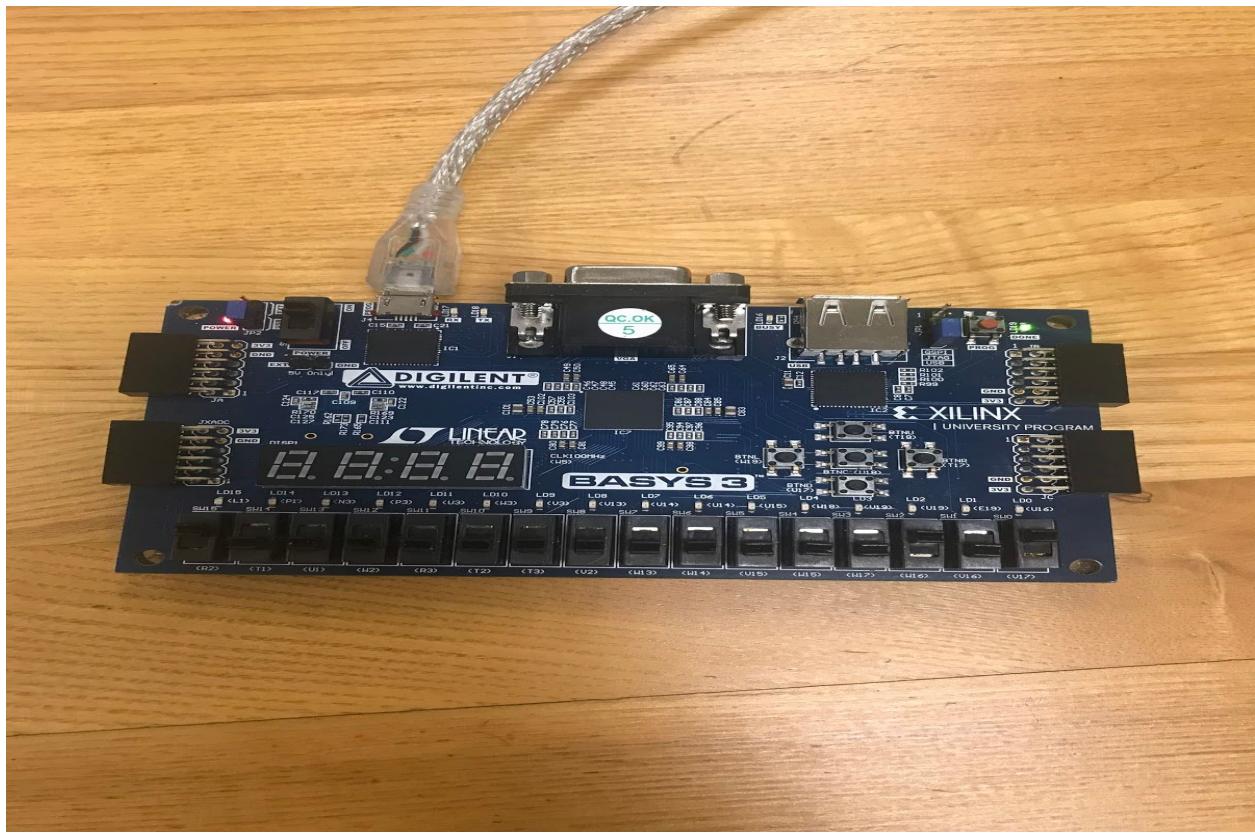


Figure 3.6 When $A=1$ and $B=0$ and are AND together and $C = 1$ is NOT C, and their respective outputs are OR together, the output $F=0$.

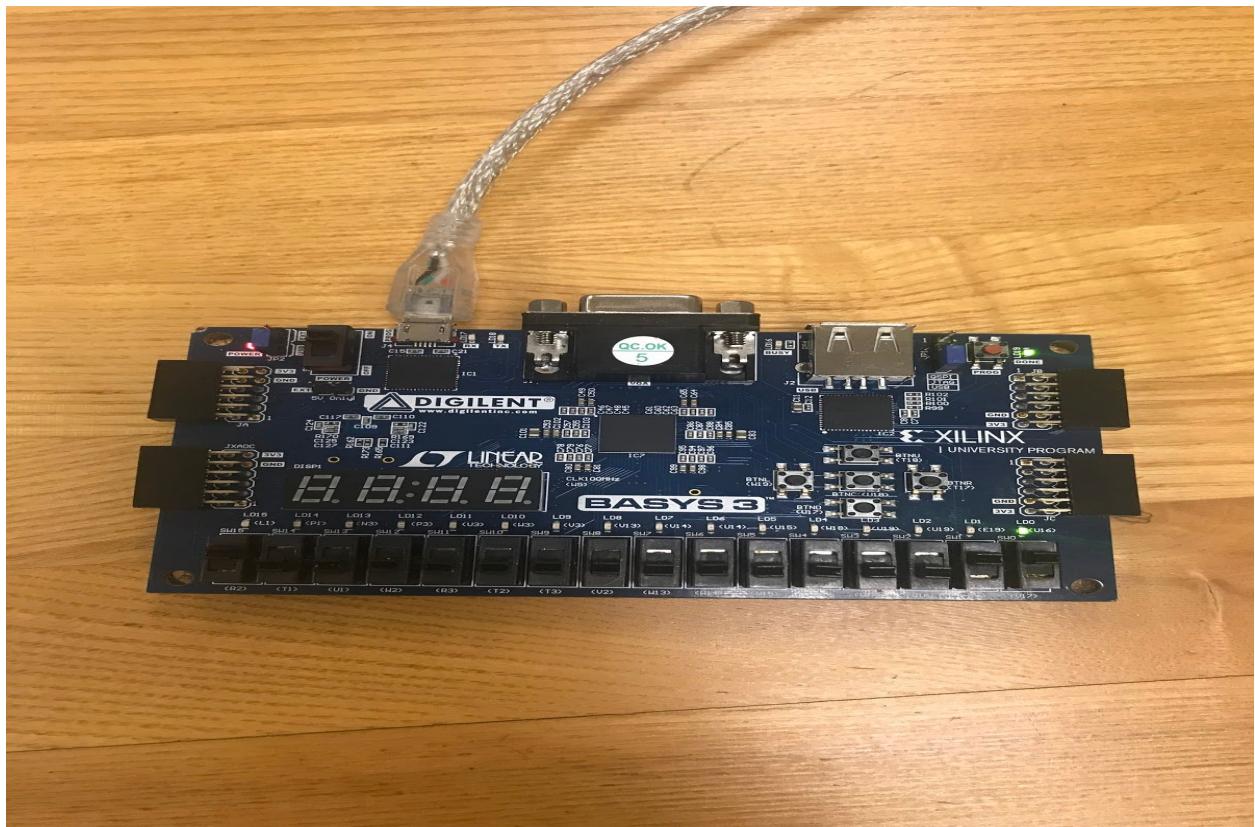


Figure 3.7 When $A=1$ and $B=1$ and are AND together and $C =0$ is NOT C, and their respective outputs are OR together, the output $F=1$.

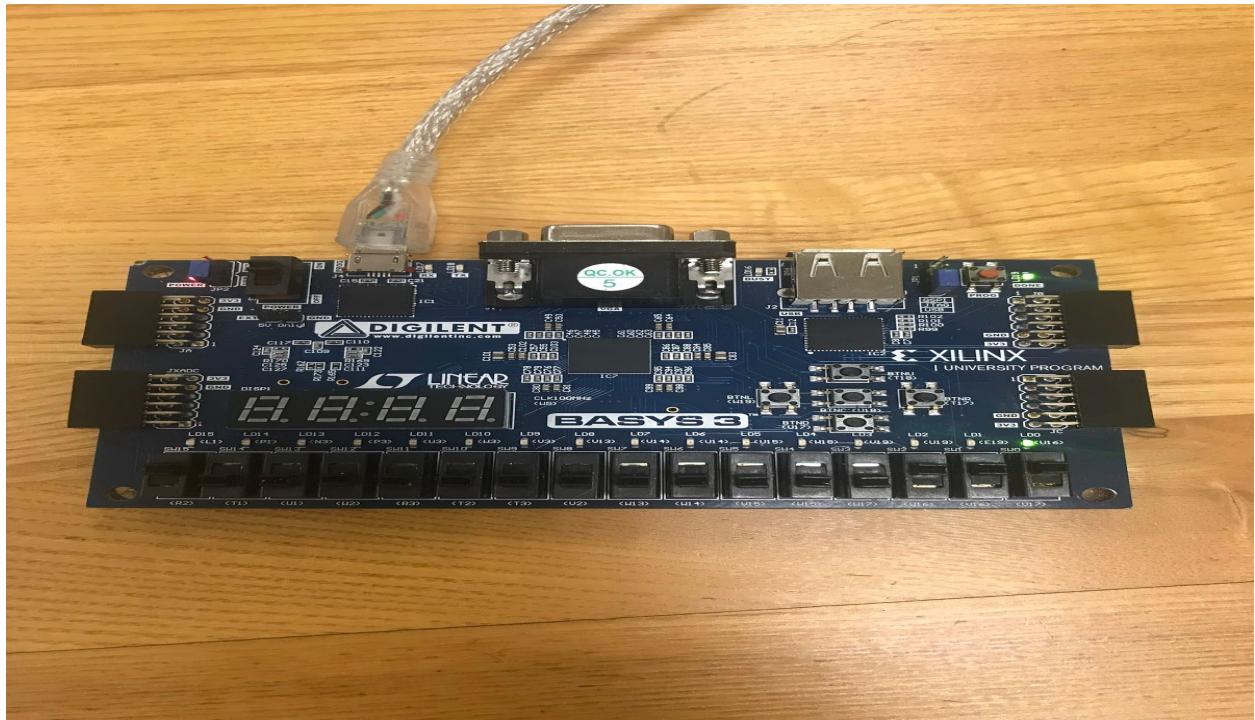


Figure 3.8 When $A=1$ and $B=1$ and are AND together and $C = 1$ is NOT C, and their respective outputs are OR together, the output $F=1$.

Conclusion

In conclusion, we're learning how to create truth tables from the given information to then, draw up a schematic to, finally, produce VHDL code. We were challenged, when trying to develop efficient VHDL code, because, as you know, there are various ways to write it.

Appendices

Part 1:

This code is performing the AND operation.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Tayanna Lee and Ian Parker

entity part1 is
    Port ( A : in STD_LOGIC;      --input A
           B : in STD_LOGIC;      --input B
           C : out STD_LOGIC);   --output C
end part1;
```

architecture Behavioral of part1 is

```
begin
    C <= A and B;      --A & B AND together to get C
end Behavioral;
```

Part 2:

This code is performing an operation to take the input of a 4 bit number and output whether or not it is a prime number.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity part2sourcefile is
    Port ( P : in STD_LOGIC_VECTOR(3 downto 0);      --four bit input P
           N : out STD_LOGIC);                      --1 bit output N
end part2sourcefile;
```

architecture Behavioral of part2sourcefile is

```
begin
process(P)
begin
    case P is          --begin when case statements
        when "0010" => N <= '1';      --2 is prime
        when "0011" => N <= '1';      --3 is prime
        when "0101" => N <= '1';      --5 is prime
        when "0111" => N <= '1';      --7 is prime
        when "1011" => N <= '1';      --11 is prime
        when "1101" => N <= '1';      --13 is prime
        when others => N <= '0';      -- other numbers between 0-15 are not prime
    end case;
    end process;
end Behavioral;
```

Part 3:

This code is to implement the circuit provided in VHDL.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity part3 is
    Port ( A : in bit;      --3, one bit input
           B : in bit;
           C : in bit;
           F : out bit);   --1, one bit output
end part3;

architecture Behavioral of part3 is
    signal temp1 : bit;          --2, temporary signal bits
    signal temp2 : bit;

begin
    temp1 <= A and B;        -- AND A & B together, and set it to temp1
    temp2 <= not C;          -- set, not C, to temp2

    F <= temp1 or temp2;      -- OR both temp signal bits

end Behavioral;
```