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)

February 7, 2019

Lab #3

Introduction

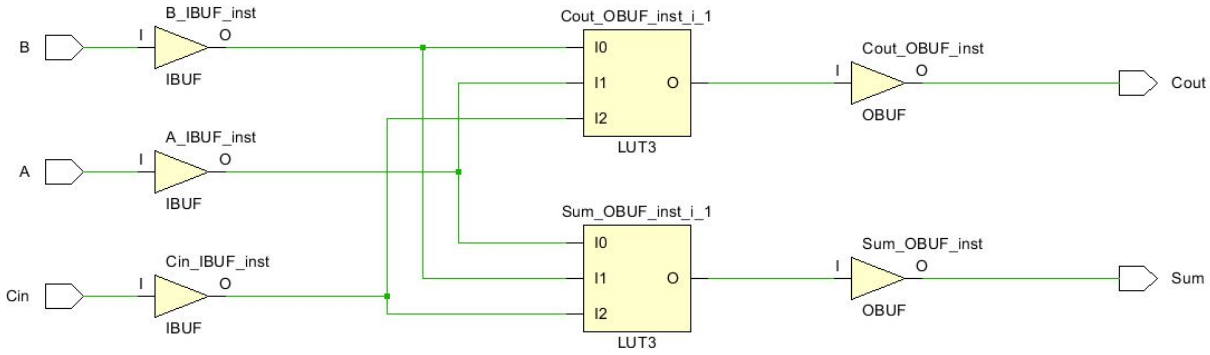
The focus of this Lab is to learn to design complex on top of one another. This lab will give us more experience with integrating design systems by using components. In practical application, the reuse of code is essential in developing larger systems that handle complex tasks. This lab focuses on Adders, Mux, and 7 Segment Displays and how they can all be combined.

Part 1) Create a full adder bit slice that will add A,B, and handle the carry Cin and implement the bit slice adder on the Basys3 Board

Truth Table

Input (A,B,Cin)	Cout	Sum
000	0	0
001	0	1
010	0	1
011	1	0
100	0	1
101	1	0
110	1	0
111	1	1

Schematic



FPGA Pin Assignment

Name	I/O	Pin
A	IN	W16
B	IN	V16
Cin	IN	V17
Cout	OUT	E19
Sum	OUT	U16

Results

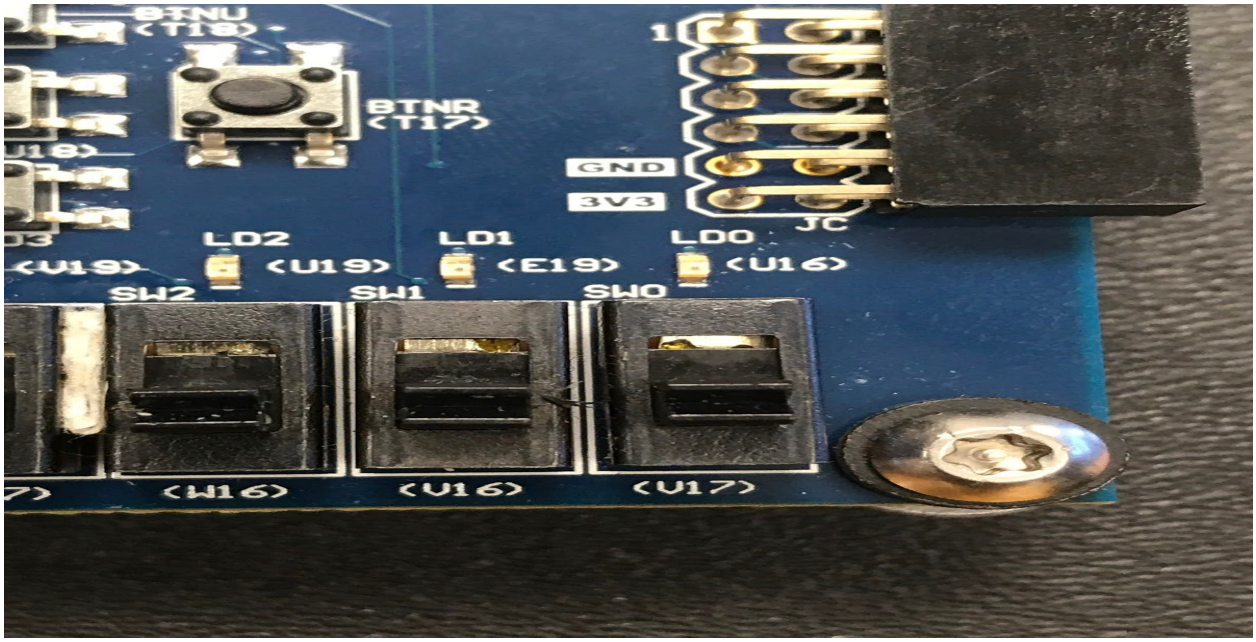


Figure 1.0) A=0; B=0; Cin=0; Cout=0; Sum=0;

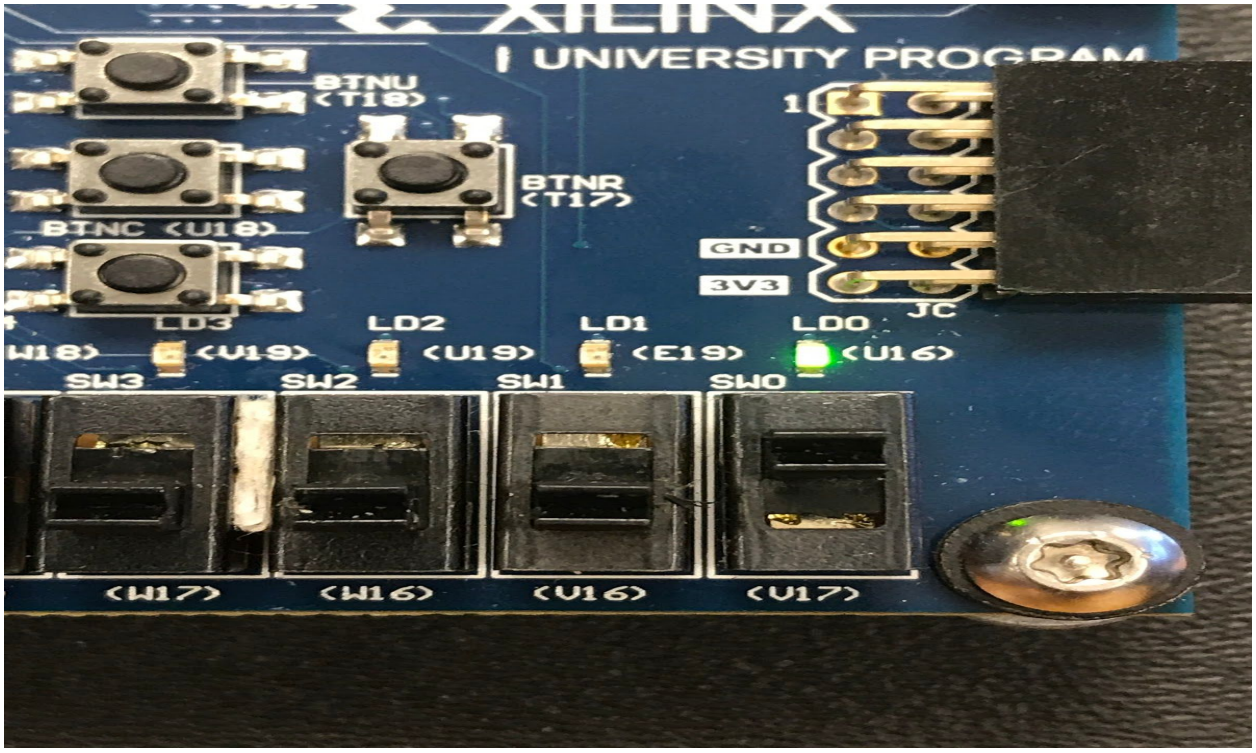


Figure 1.1)A=0; B=0; Cin=1; Cout=0; Sum=1;

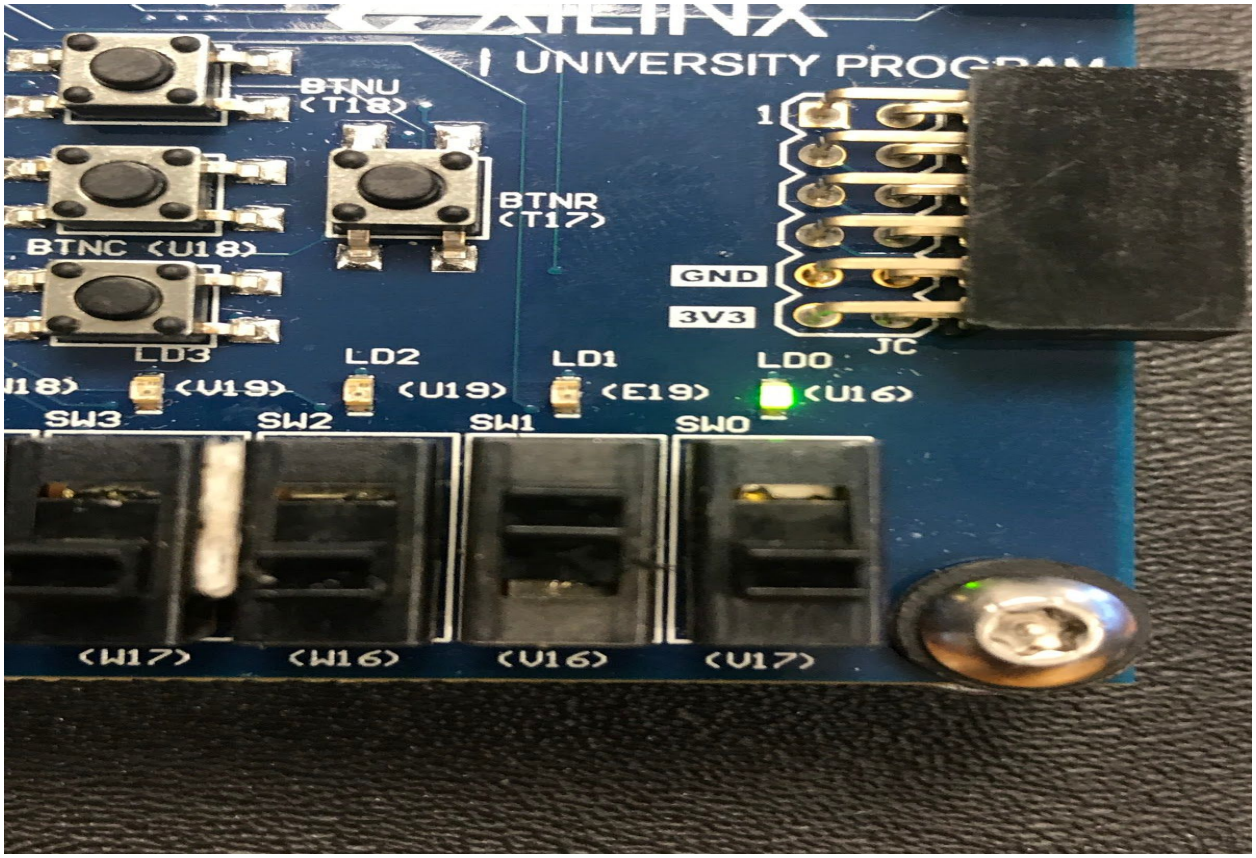


Figure 1.2)A=0; B=1; Cin=0; Cout=0; Sum=1;

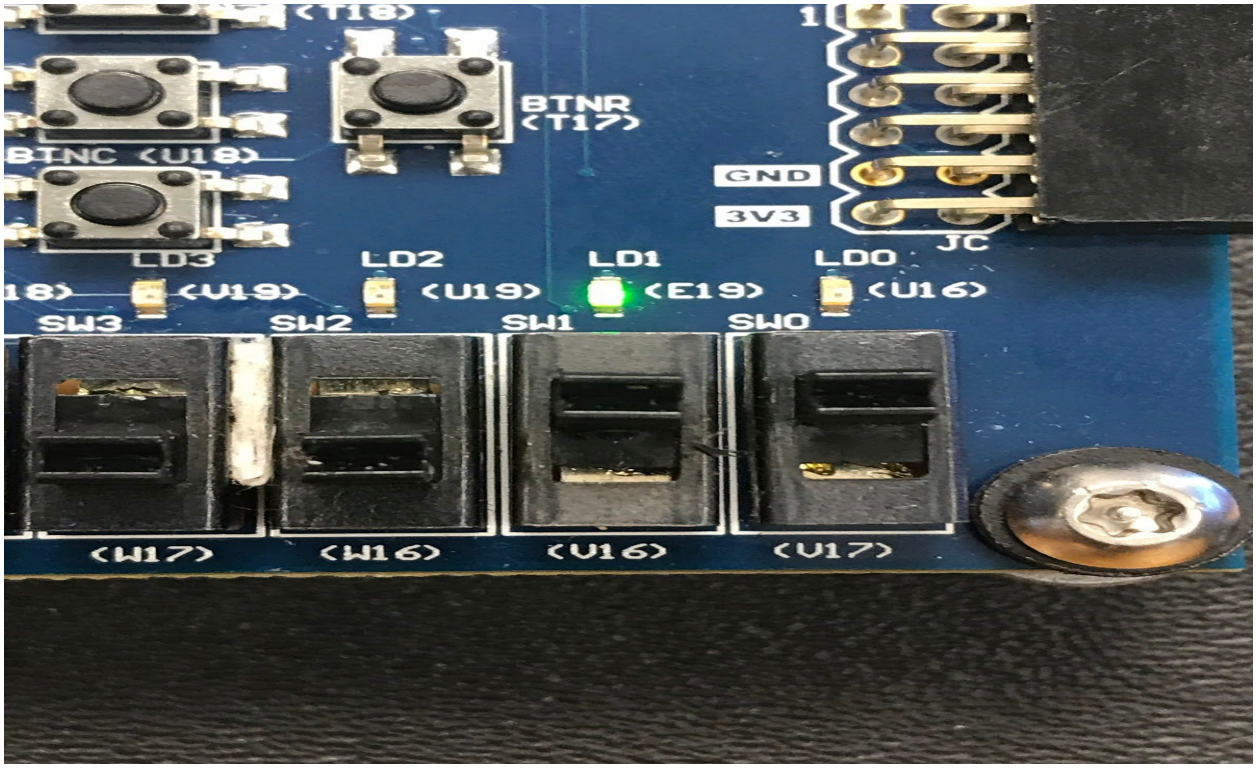


Figure 1.3)A=0; B=1; Cin=1; Cout=1; Sum=0;

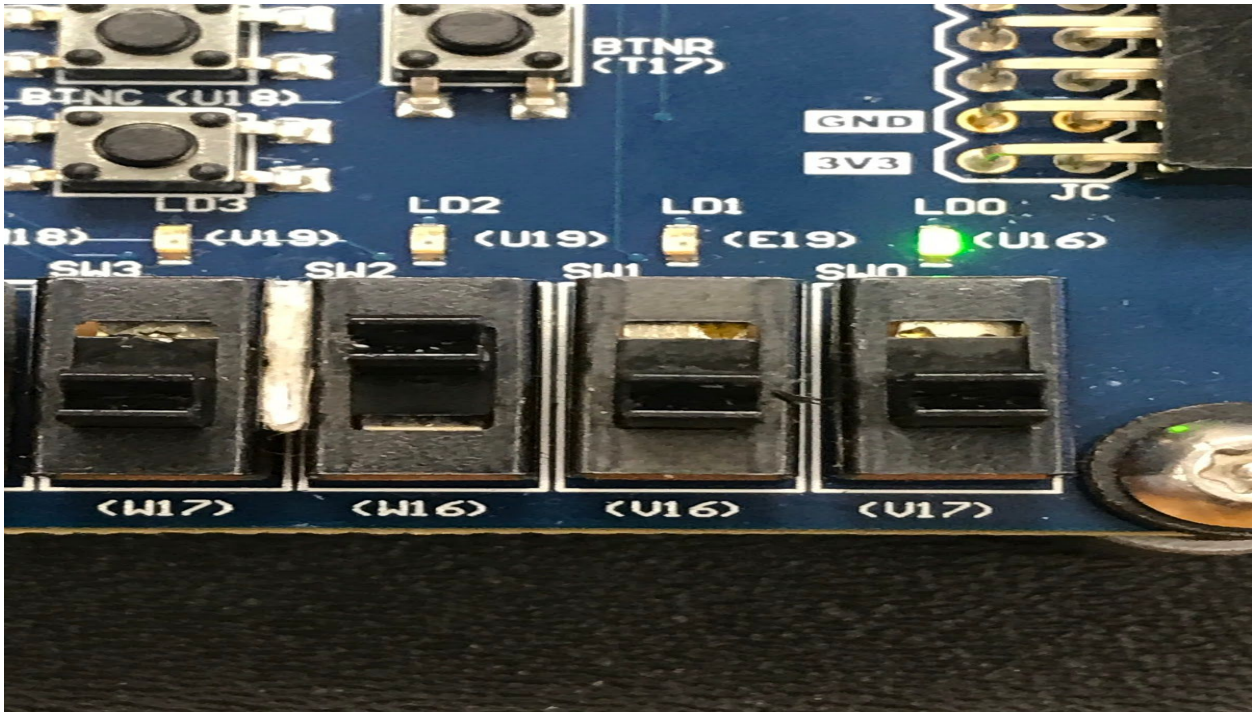


Figure 1.4)A=1; B=0; Cin=0; Cout=0; Sum=1;

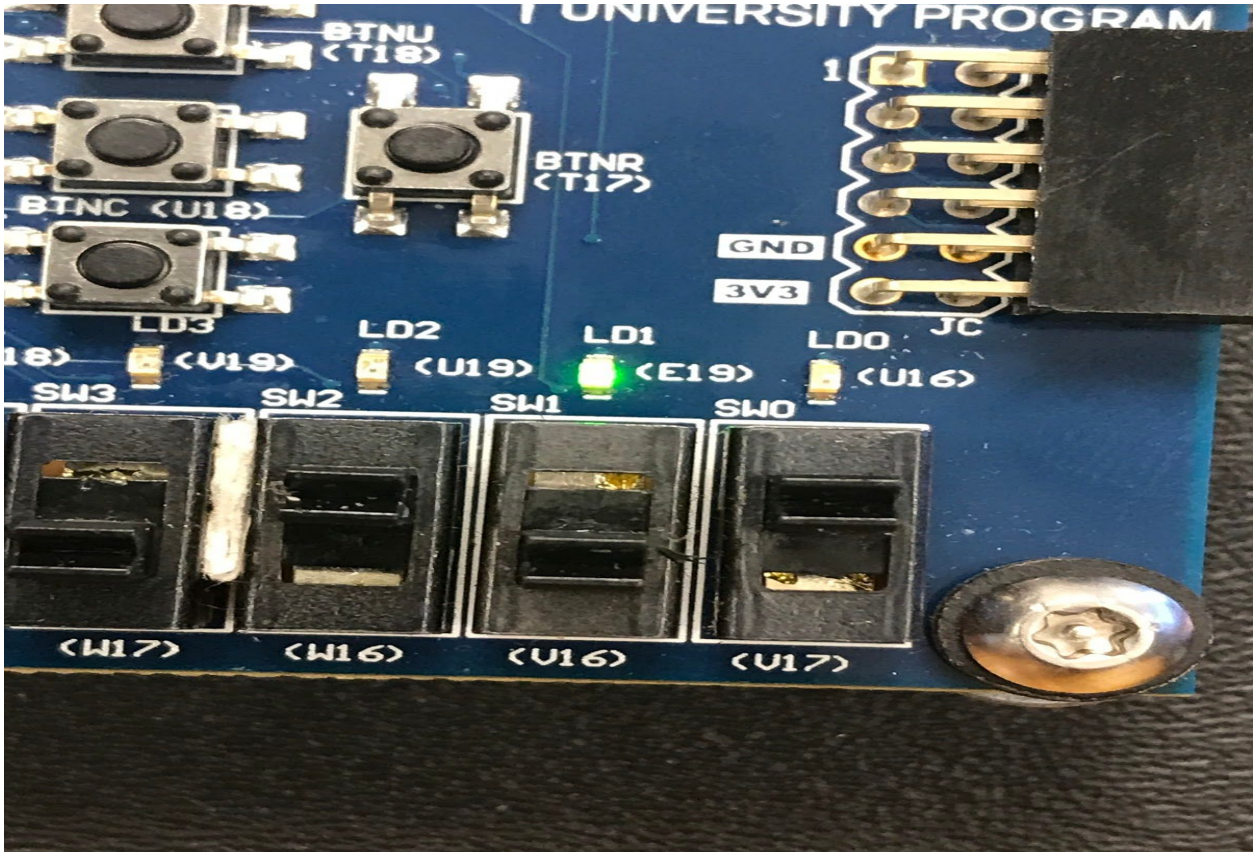


Figure 1.5)A=1; B=0; Cin=1; Cout=1; Sum=0;

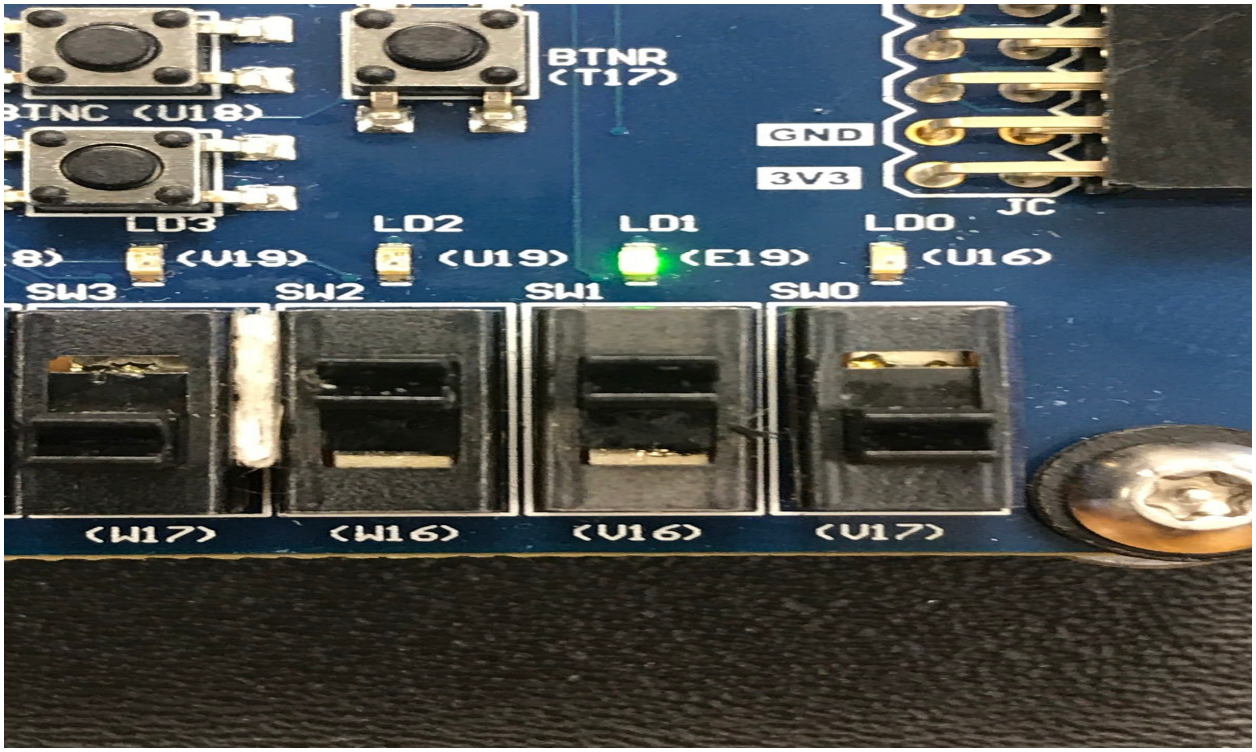


Figure 1.6)A=1; B=1; Cin=0; Cout=1; Sum=0;

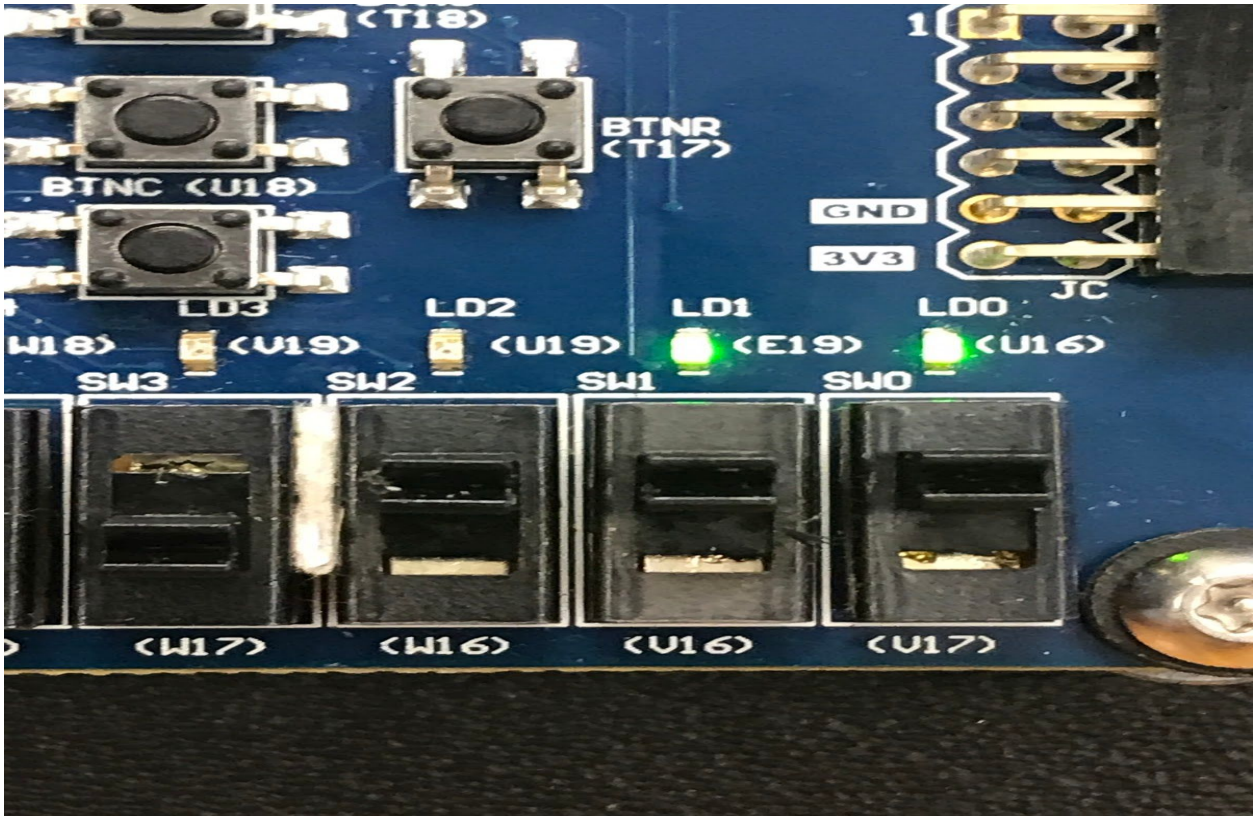


Figure 1.7)A=1; B=1; Cin=1; Cout=1; Sum=1;

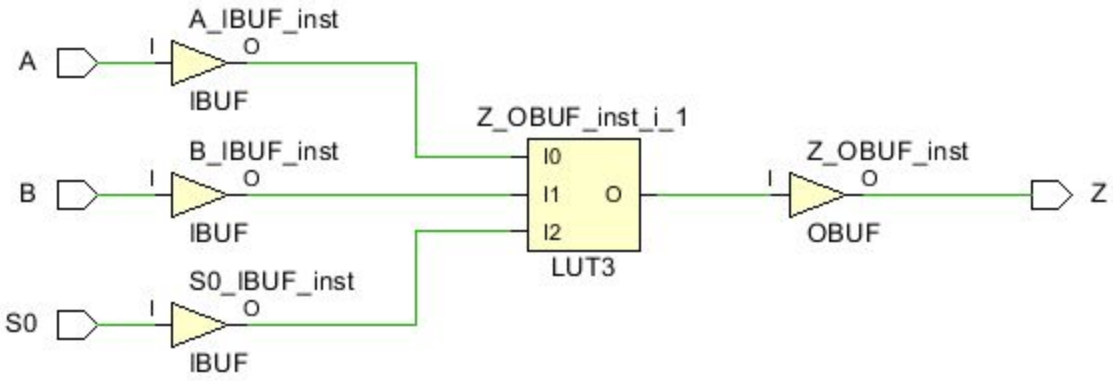
Part 2) In a separate project, create a 2:1 mux and implement it on the board.

Truth Table

Inputs			**Outputs**
S0	A	B	Z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

S0 = 0, Select Bit B;
When S0 = 1 Select bit A;

Schematic



FPGA Pin Assignment

Name	I/O	
A	IN	V16
B	IN	V17
S0	IN	W16
Z	OUT	E19

Results

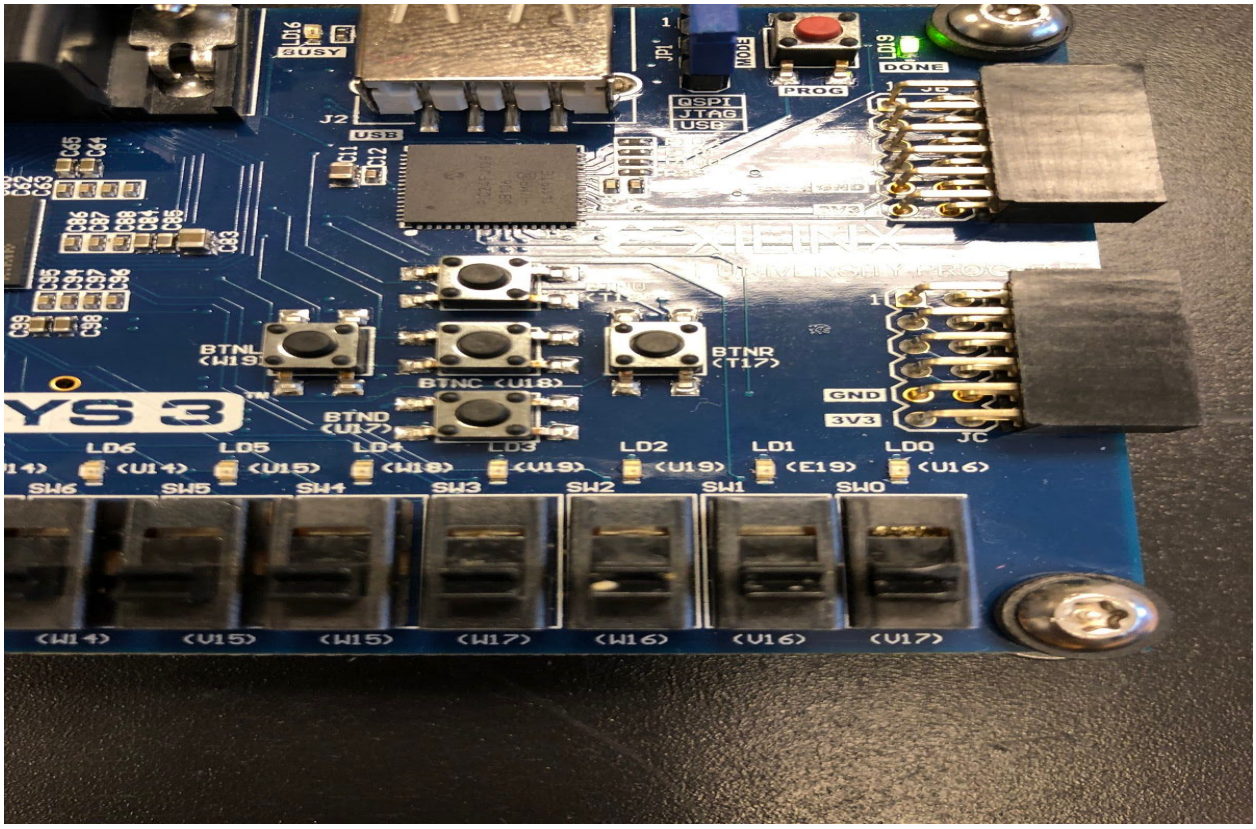


Figure 2.0)S0 = 0; A=0; B=0; Z=0;

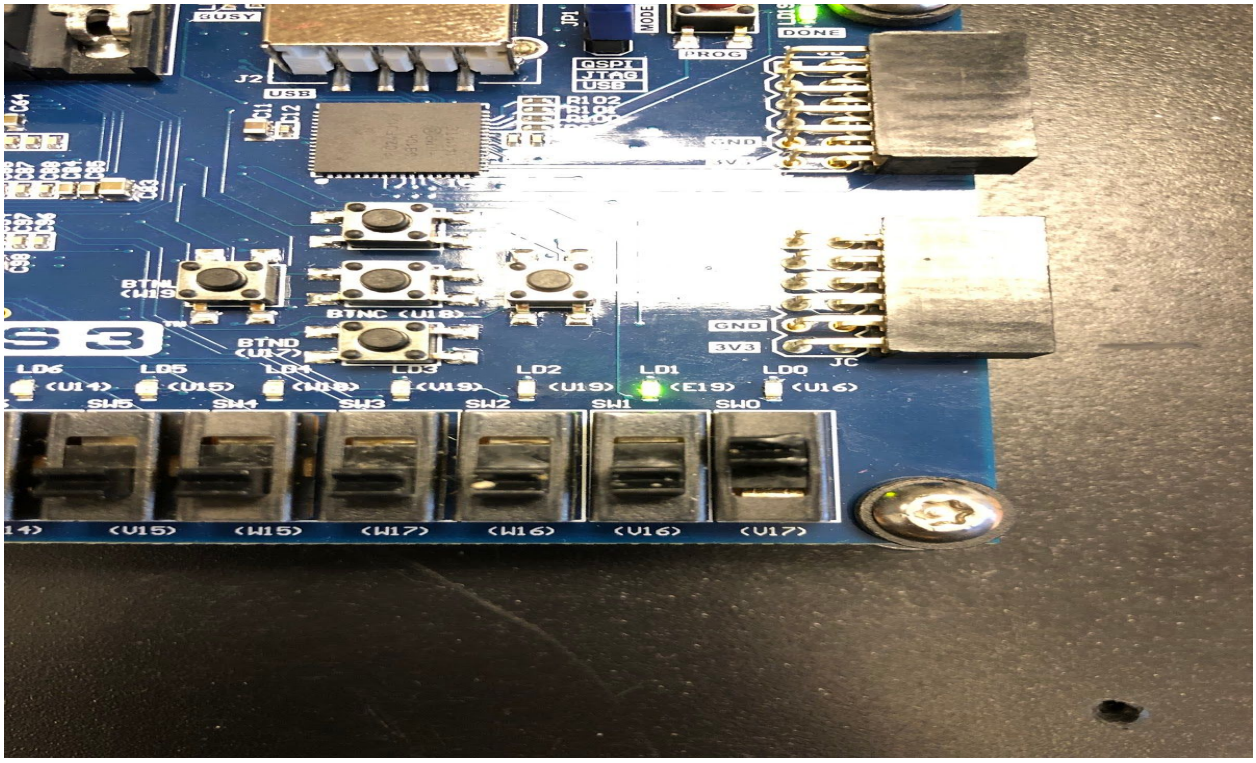


Figure 2.1)S0 = 0; A=0; B=1; Z=1;

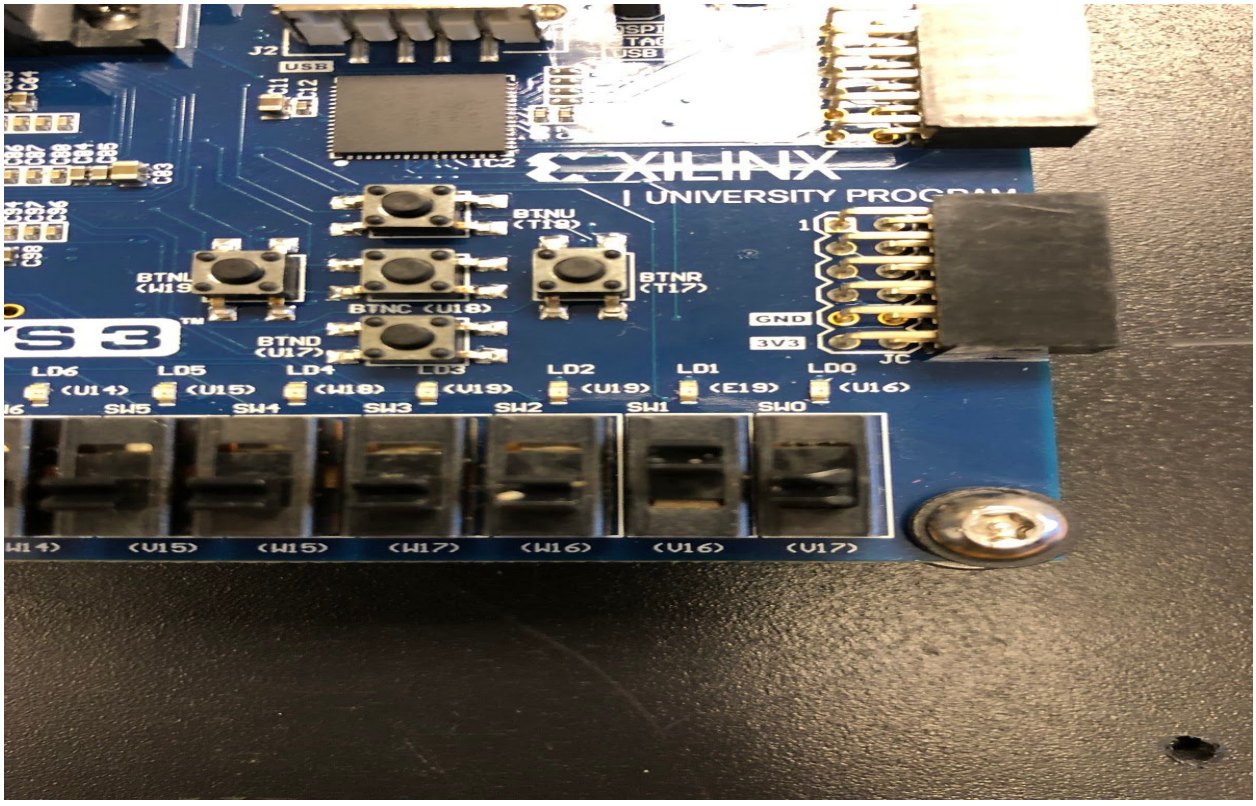


Figure 2.2)S0 = 0; A=1; B=0; Z=0;

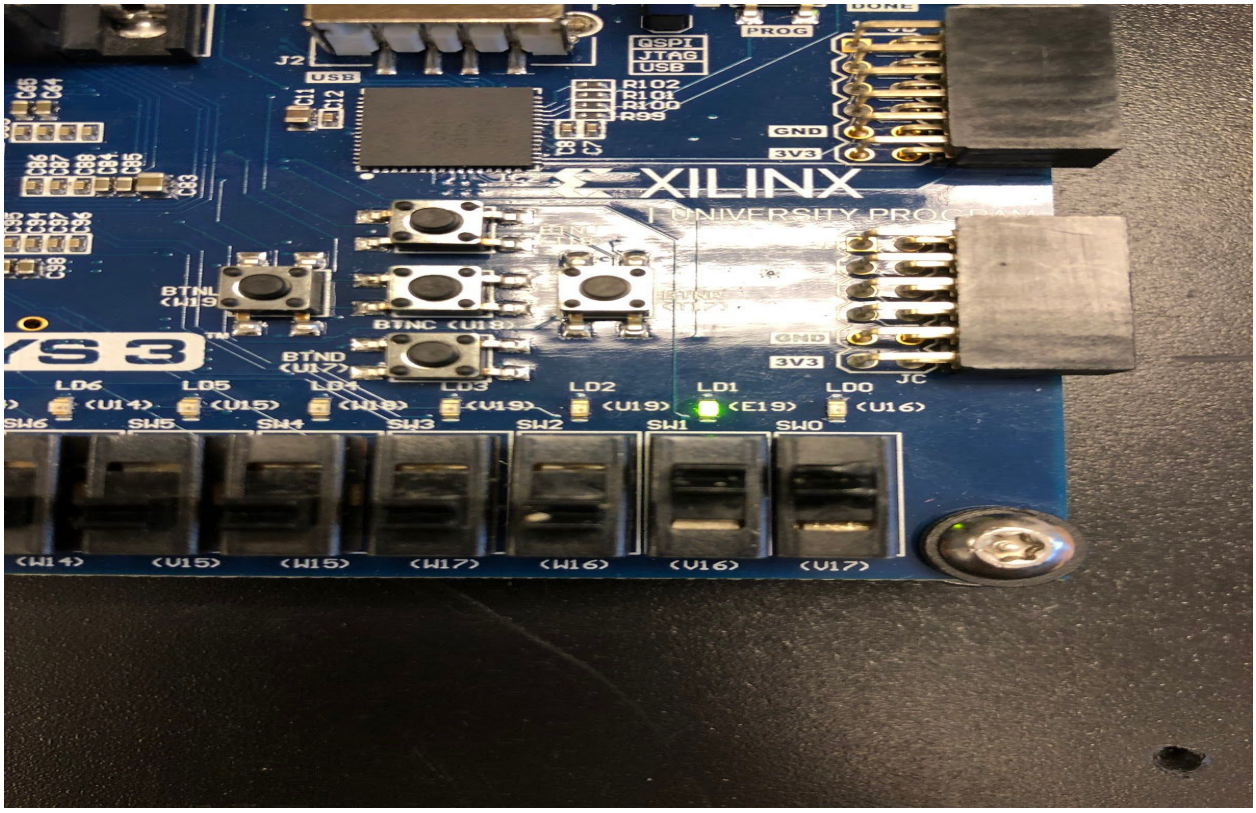


Figure 2.3)S0 = 0; A=1; B=1; Z=1;

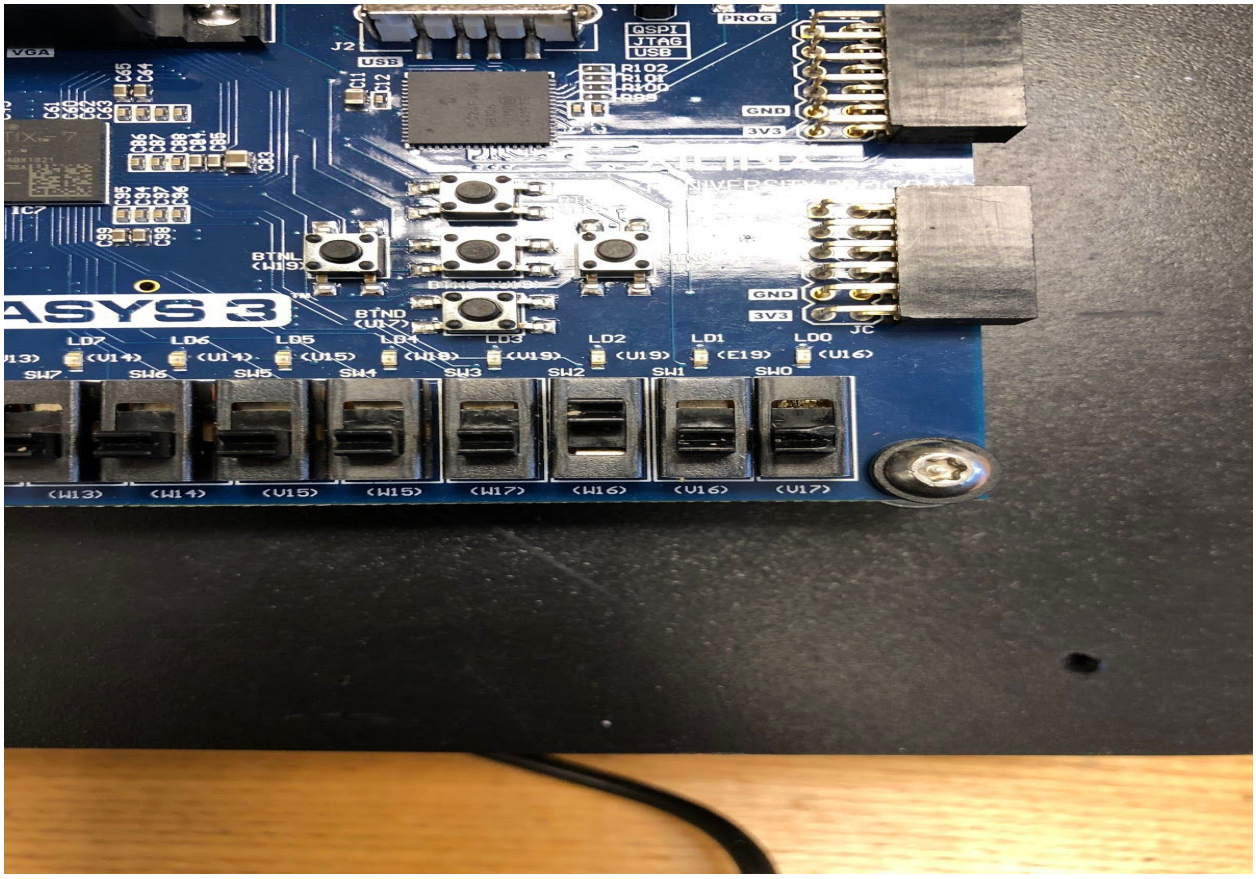


Figure 2.4)S0 = 1; A=0; B=0; Z=0;

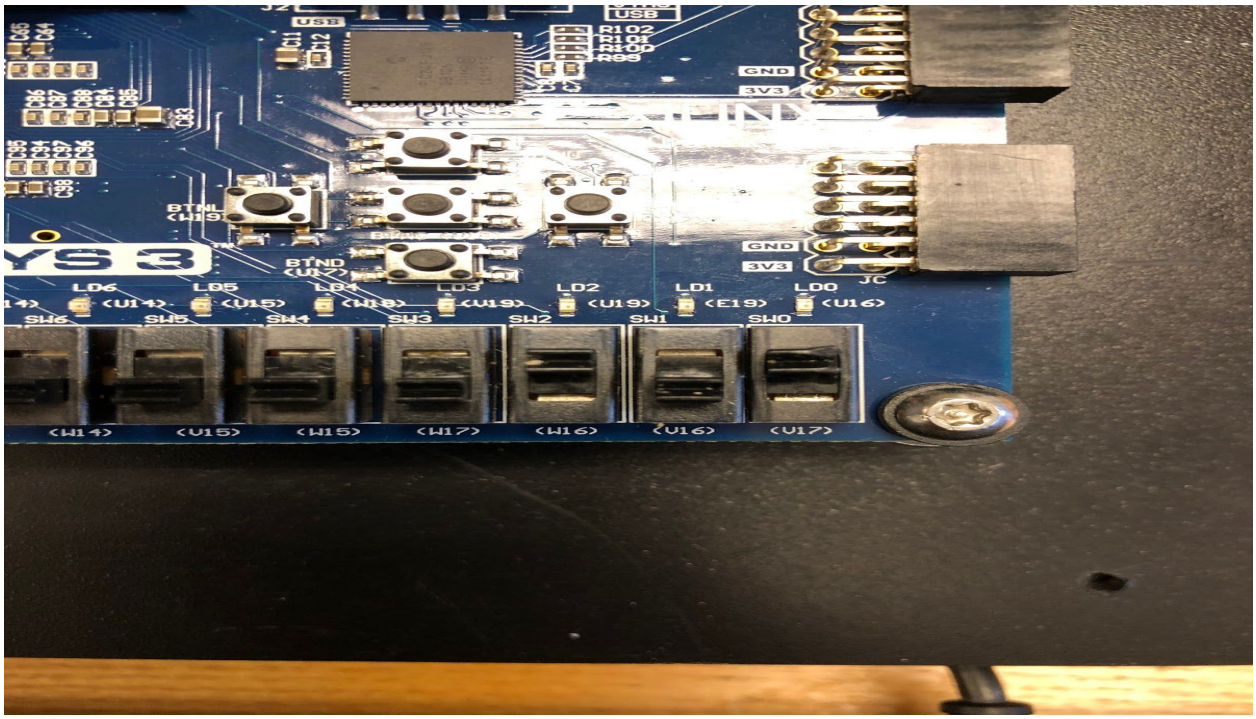


Figure 2.5)S0 = 1; A=0; B=1; Z=0;

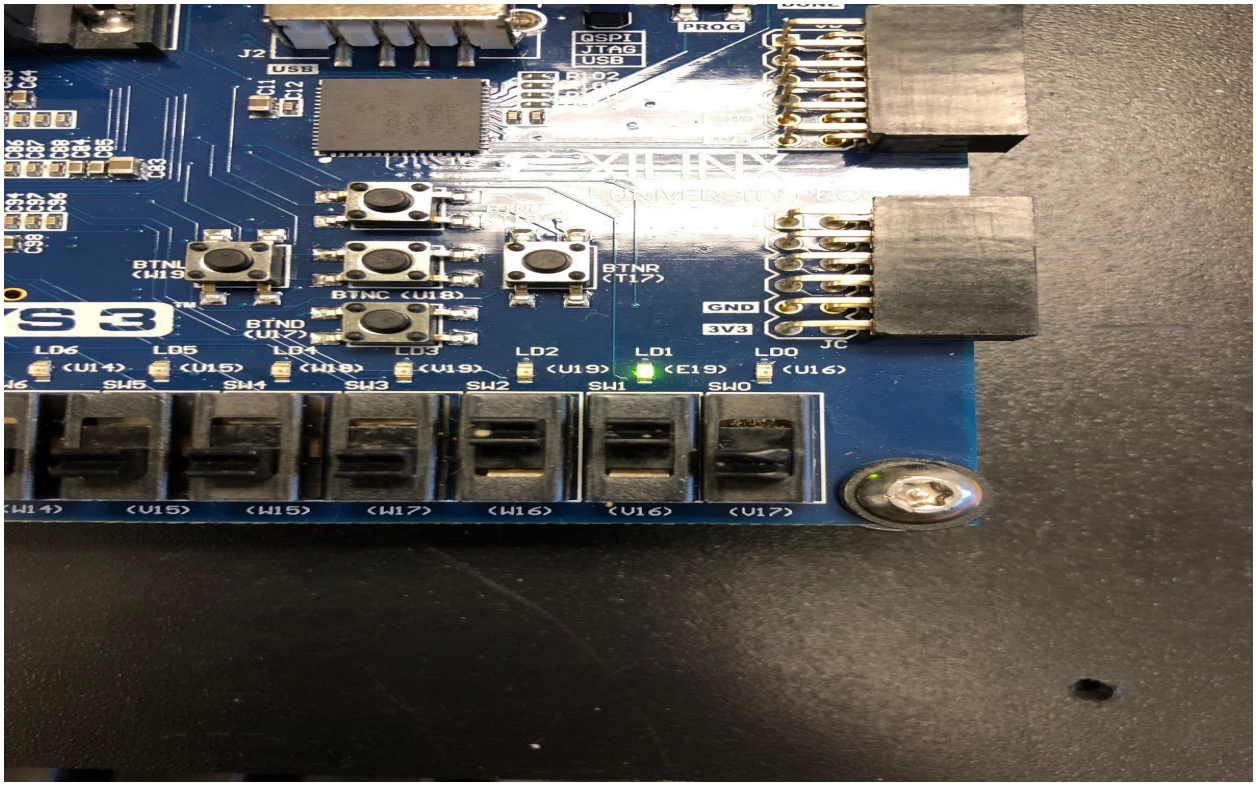


Figure 2.6)S0 = 1; A=1; B=0; Z=1;

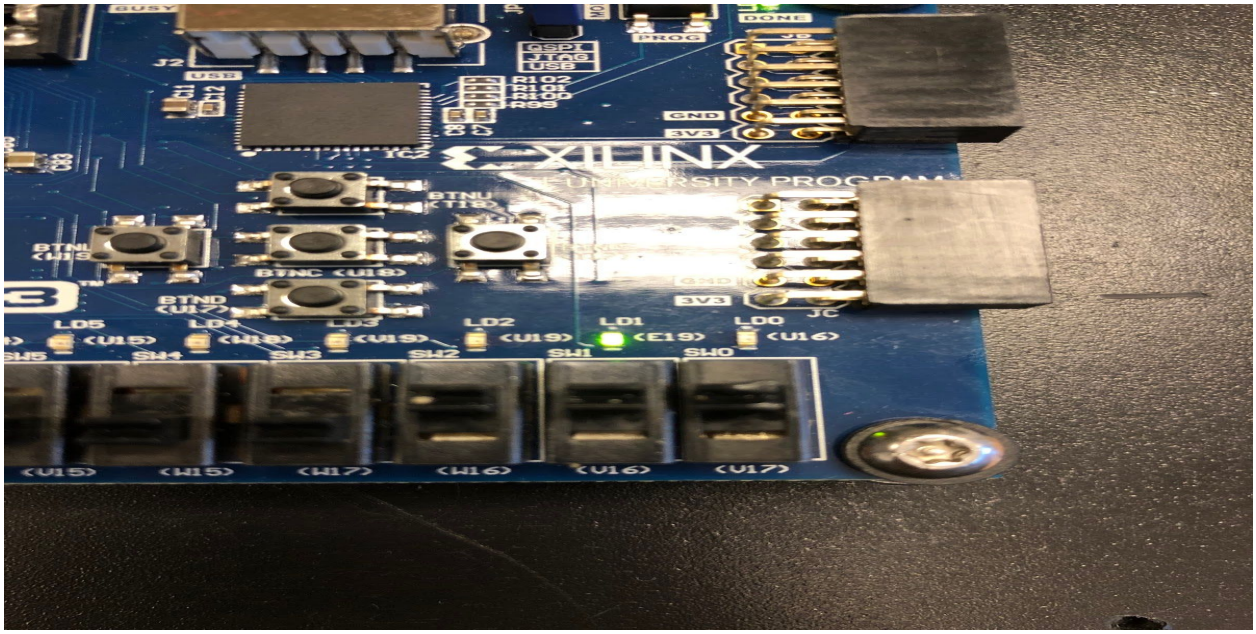


Figure 2.7)S0 = 1; A=1; B=1; Z=1;

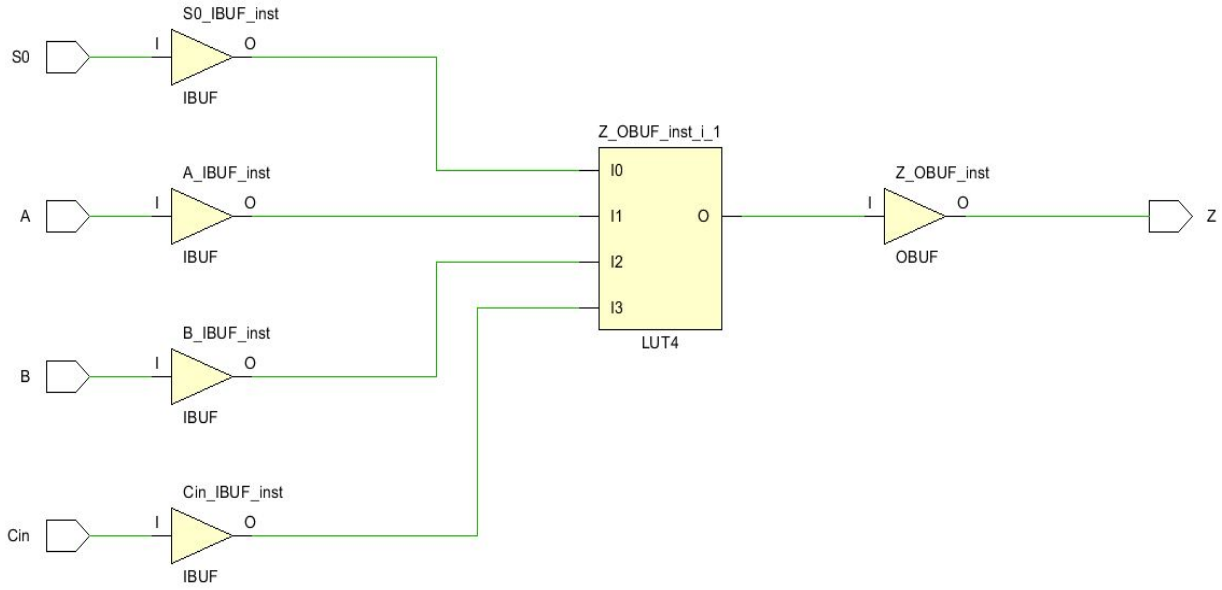
Part 2B) Join the 2:1 MUX (from part 2) and the bit slice ADDER (from part 1) as components and demonstrate the circuit's functionality using the LEDs and switches/buttons on the board.

Truth Table

S0	**A**	**B**	**Cin**	Cout	Sum	**Z**
0	**0**	**0**	**0**	0	0	**0**
0	**0**	**0**	**1**	0	1	**1**
0	**0**	**1**	**0**	0	1	**1**
0	**0**	**1**	**1**	1	0	**0**
0	**1**	**0**	**0**	0	1	**1**
0	**1**	**0**	**1**	1	0	**0**
0	**1**	**1**	**0**	1	0	**0**
0	**1**	**1**	**1**	1	1	**1**
1	**0**	**0**	**0**	0	0	**0**
1	**0**	**0**	**1**	0	1	**0**
1	**0**	**1**	**0**	0	1	**0**
1	**0**	**1**	**1**	1	0	**1**
1	**1**	**0**	**0**	0	1	**0**
1	**1**	**0**	**1**	1	0	**1**
1	**1**	**1**	**0**	1	0	**1**
1	**1**	**1**	**1**	1	1	**1**

When S0 = 0; the Output is Sum.
When S0 = 1; the Output is Cout

Schematic



FPGA PIN Assignment

Name	I/O	PIN
S0	IN	W17
A	IN	W16
B	IN	V16
Cin	IN	V17
Cout	OUT	V19
Sum	OUT	E19
Z	OUT	U16

Results

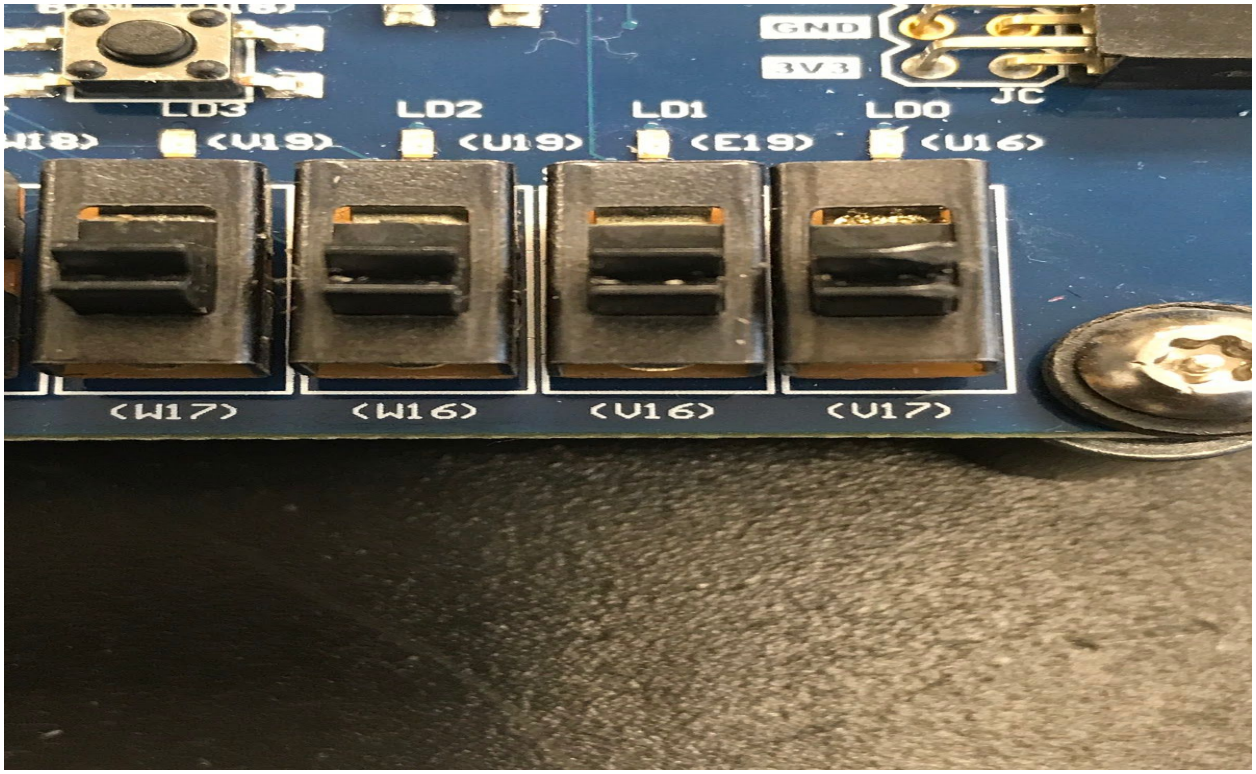


Figure 2B.0) [S0,A,B,Cin] = [0000] ; [Z]= [0]

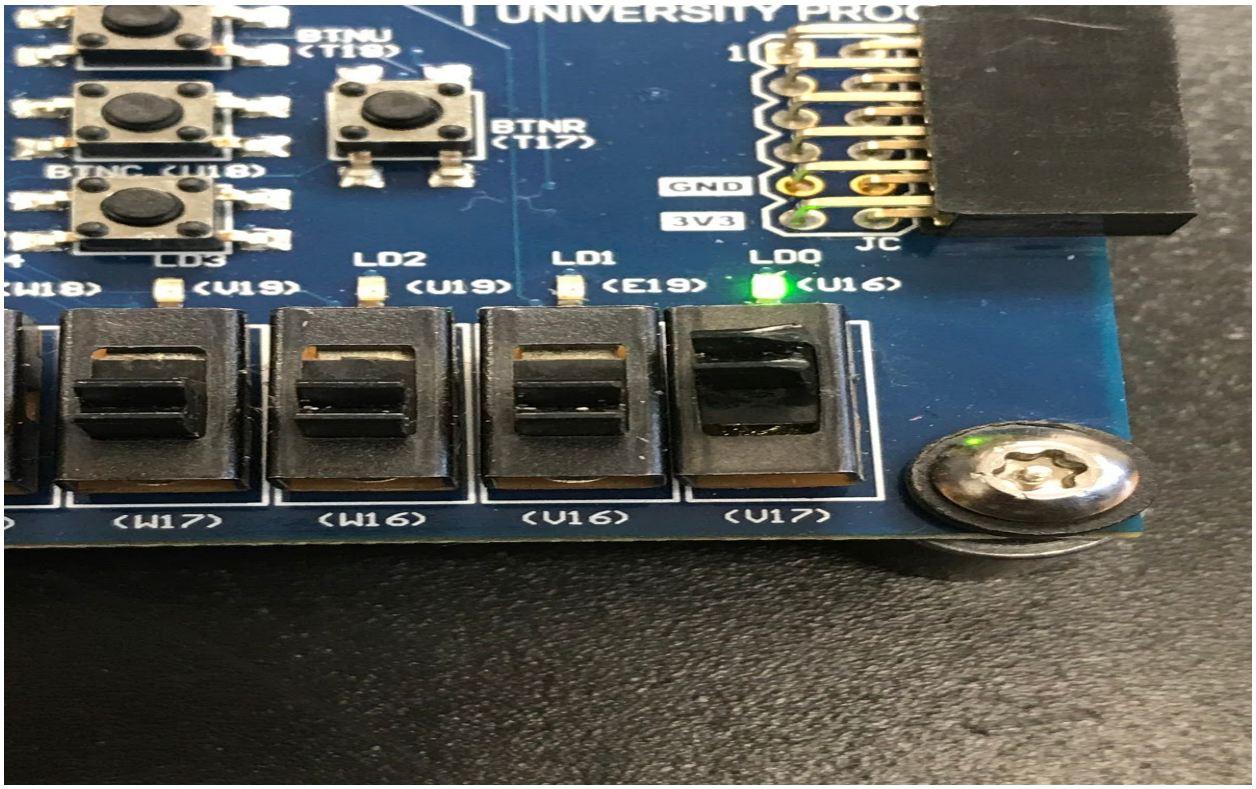


Figure 2B.1)[S0,A,B,Cin] = [0001] ; [Z]= [1]

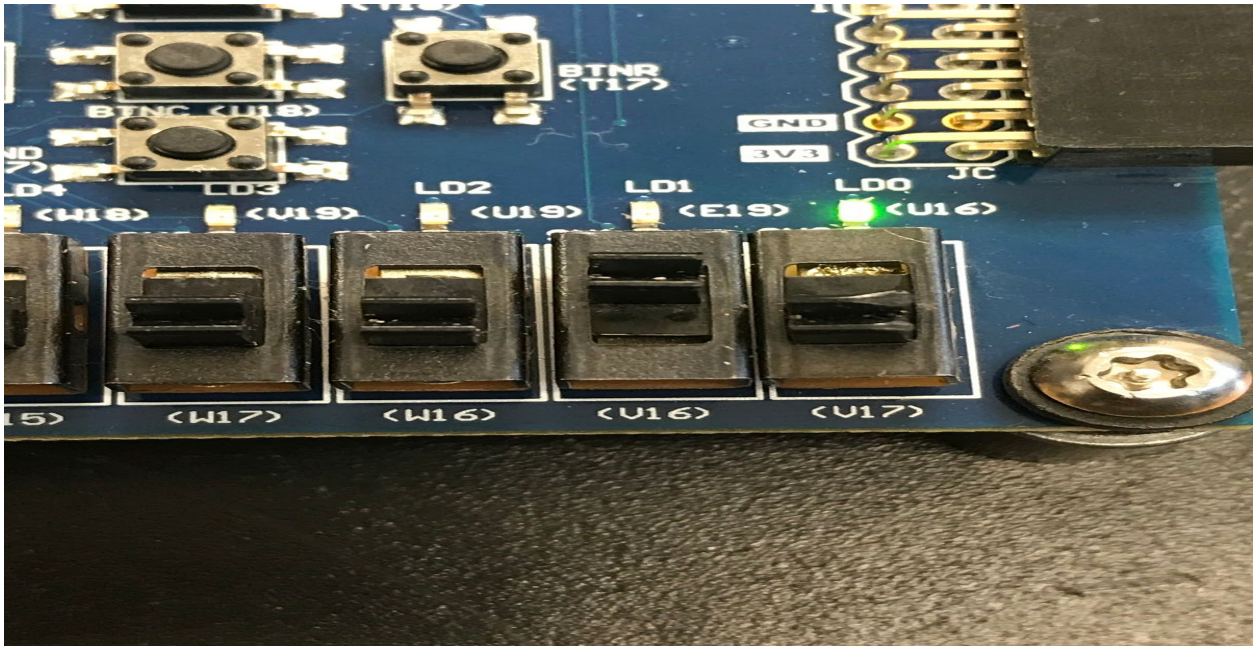


Figure 2B.2) [S0,A,B,Cin] = [0010] ; [Z]= [1]

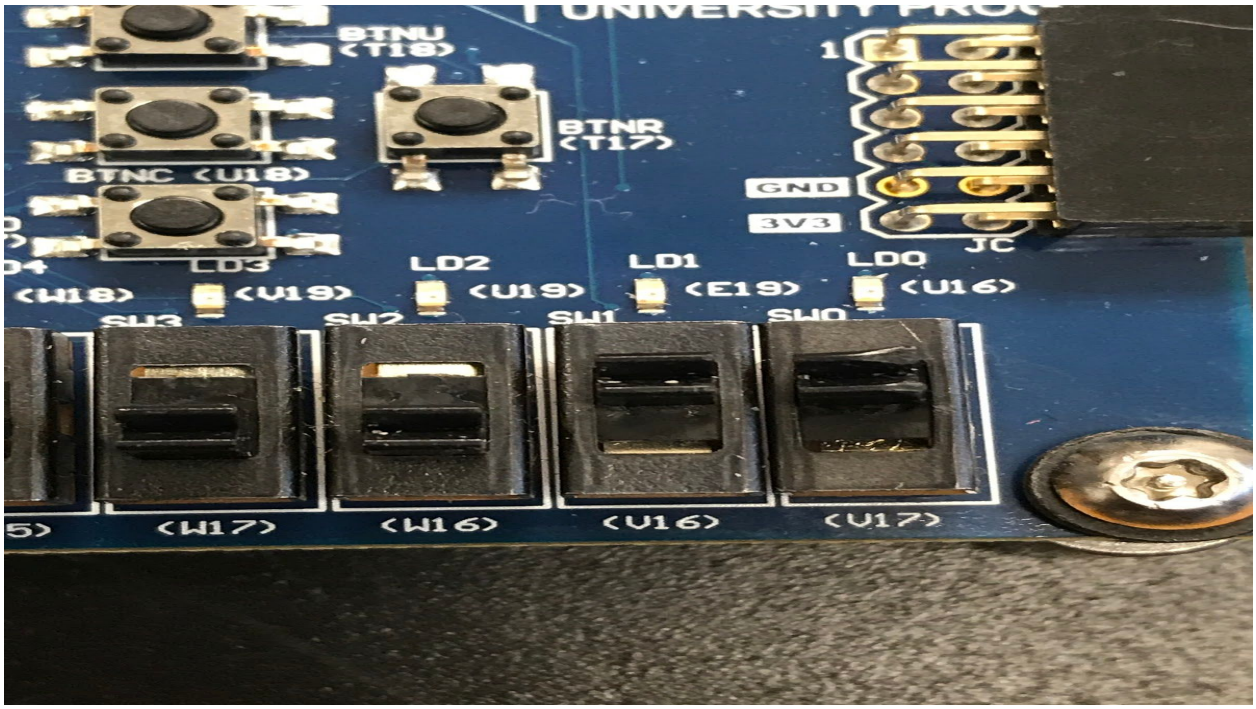


Figure 2B.3) [S0,A,B,Cin] = [0011] ; [Z]= [0]

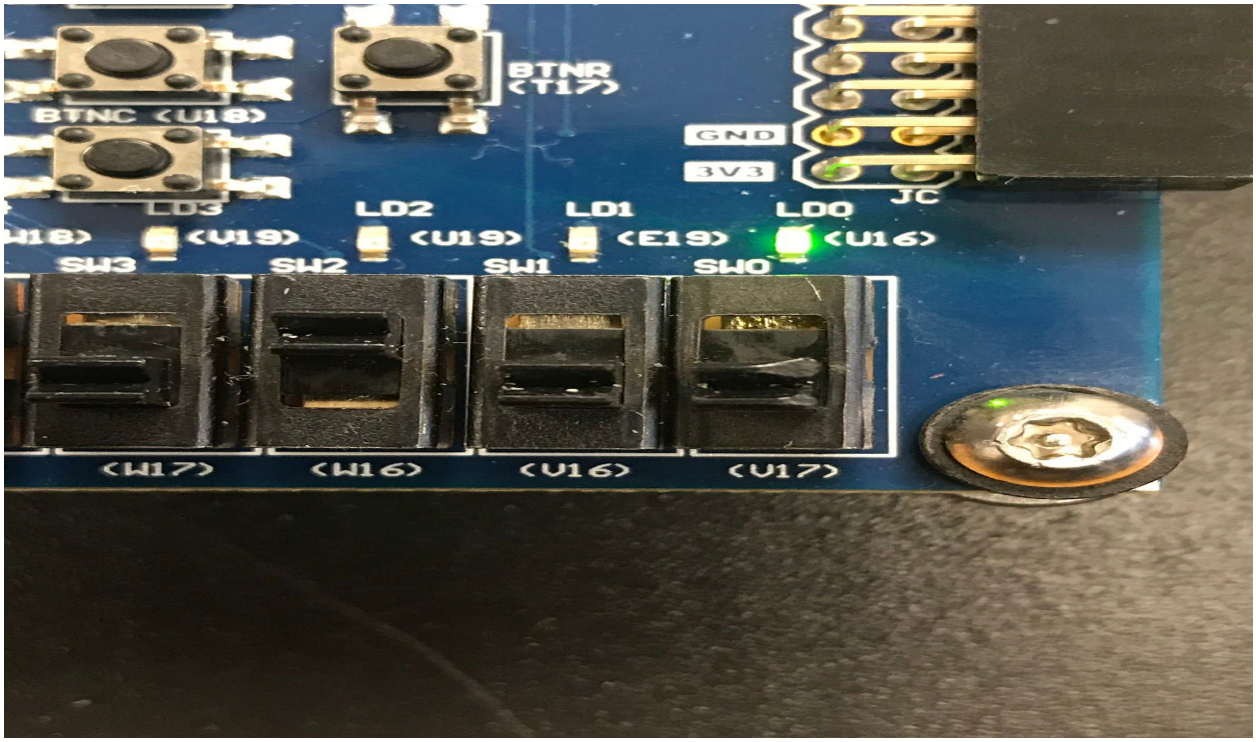


Figure 2B.4) [S0,A,B,Cin] = [0100] ; [Z]= [1]

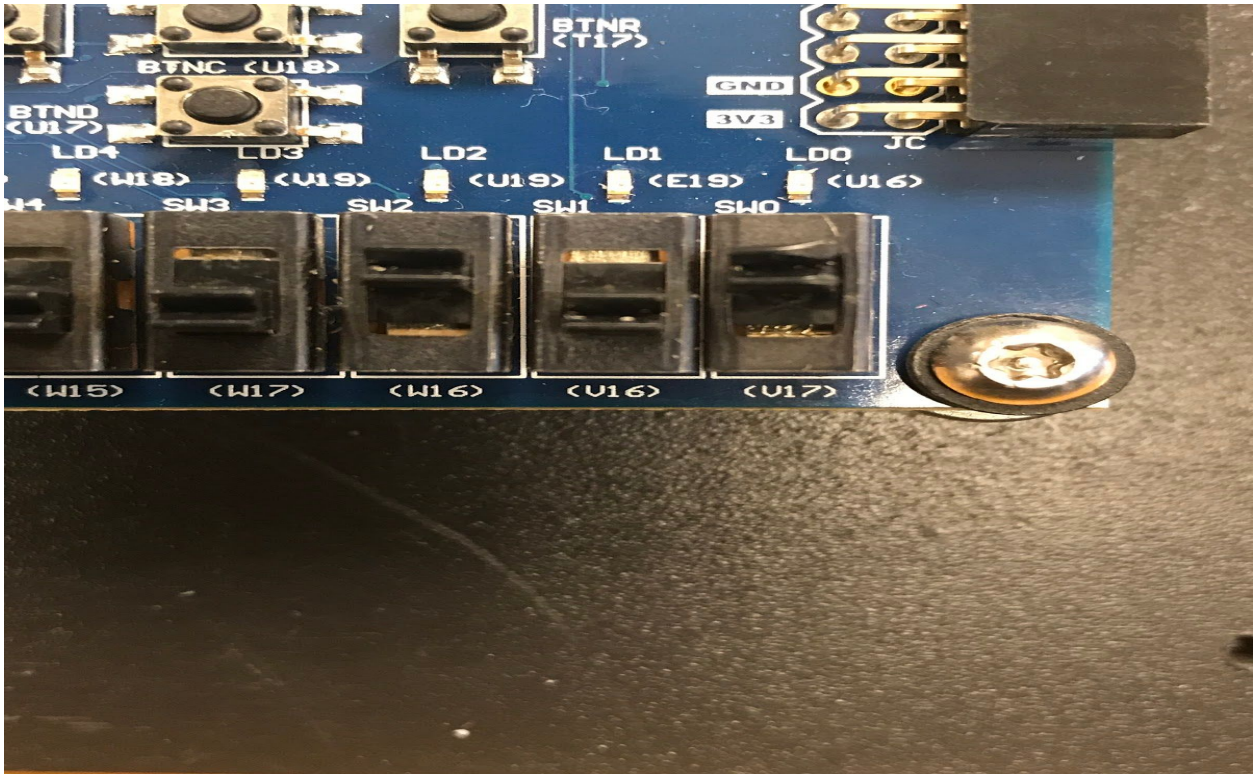


Figure 2B.5) [S0,A,B,Cin] = [0101] ; [Z]= [0]

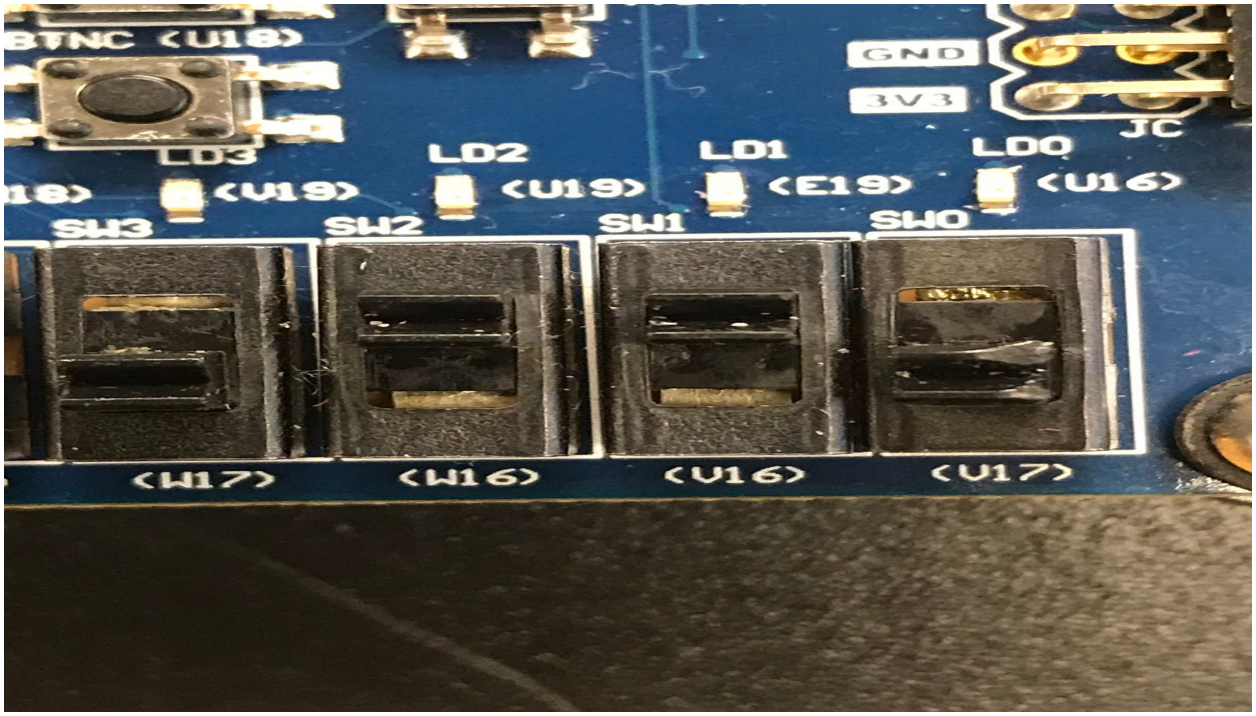


Figure 2B.6) [S0,A,B,Cin] = [0110] ; [Z]= [0]

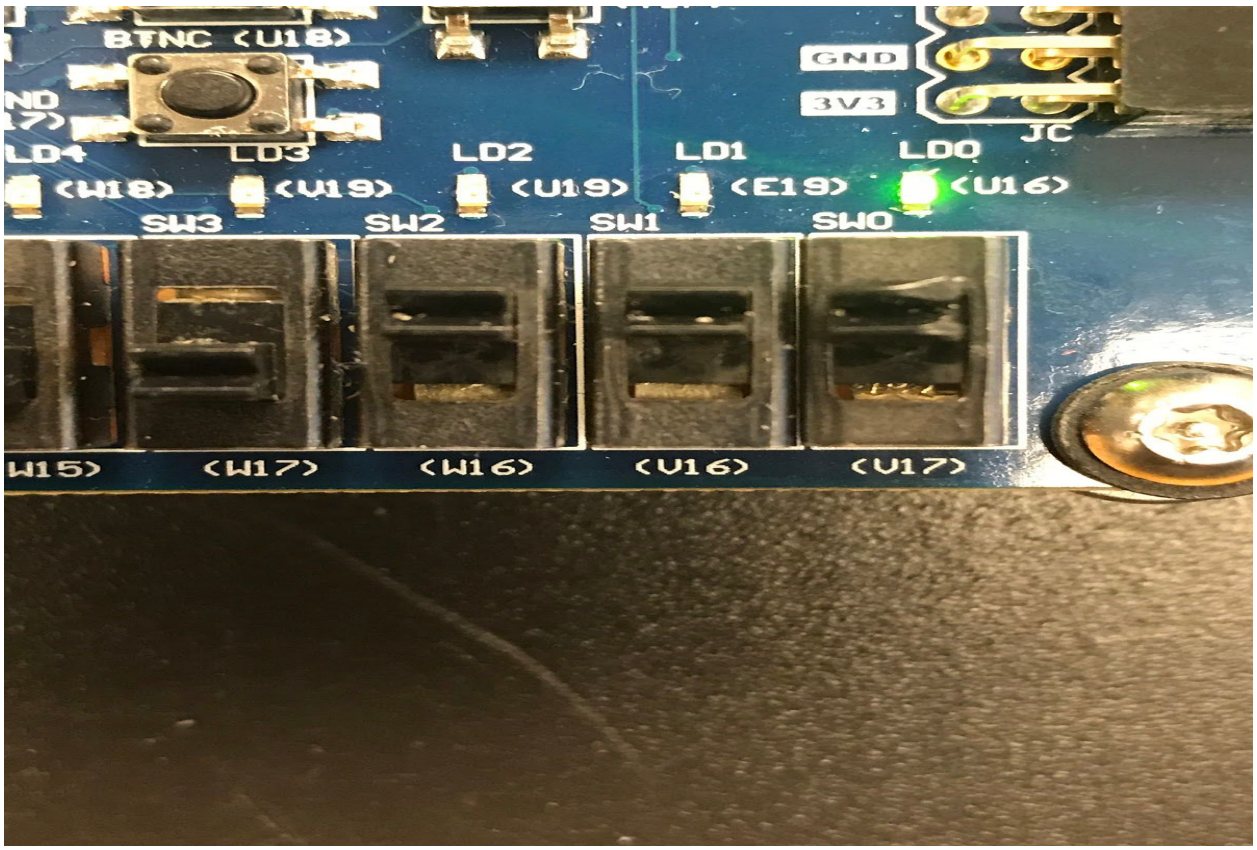


Figure 2B.7) [S0,A,B,Cin] = [0111] ; [Z]= [1]

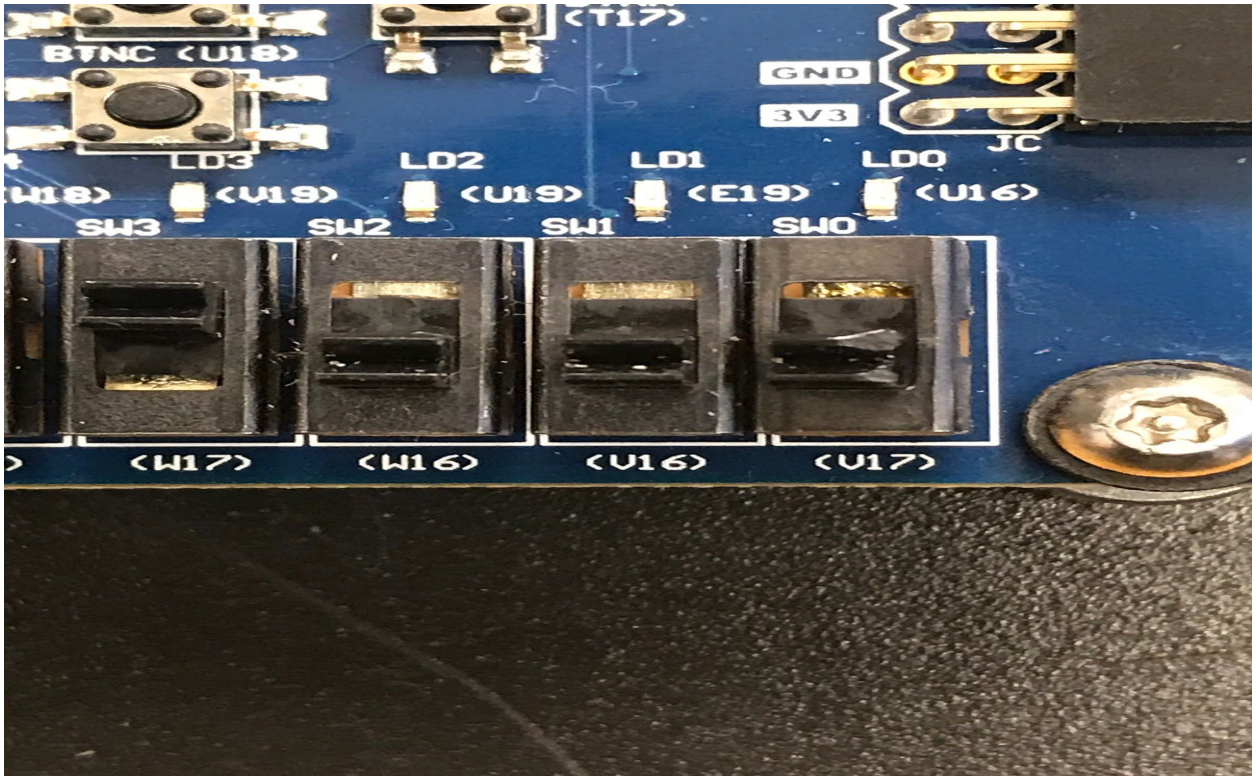


Figure 2B.8) [S0,A,B,Cin] = [1000] ; [Z]= [0]

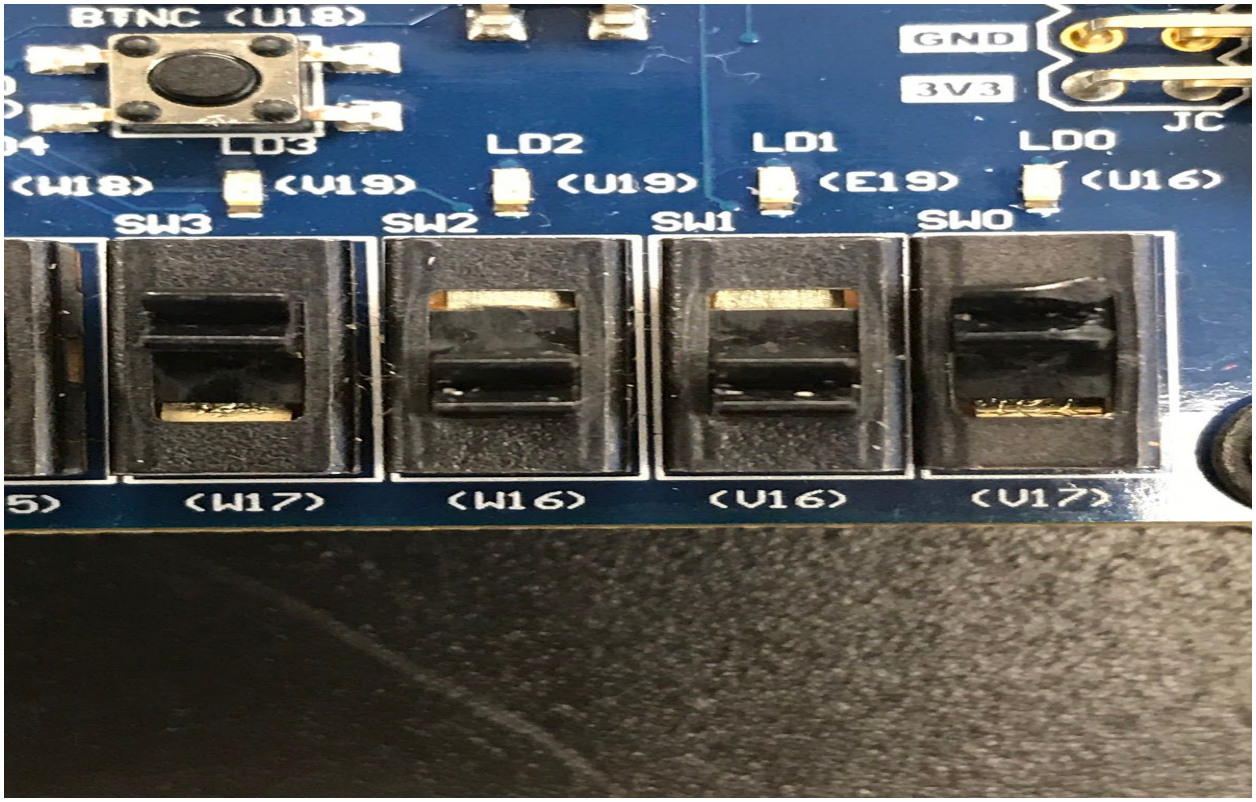


Figure 2B.9) [S0,A,B,Cin] = [1001] ; [Z]= [0]

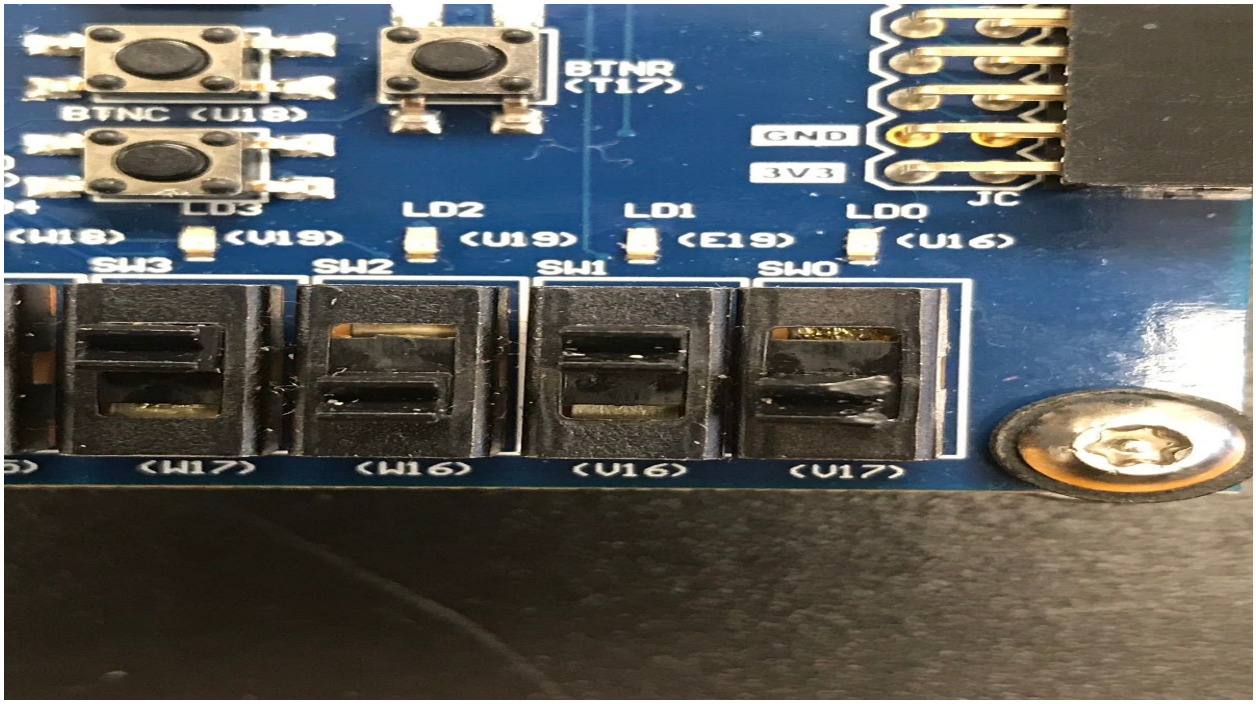


Figure 2B.10) [S0,A,B,Cin] = [1010] ; [Z]= [0]

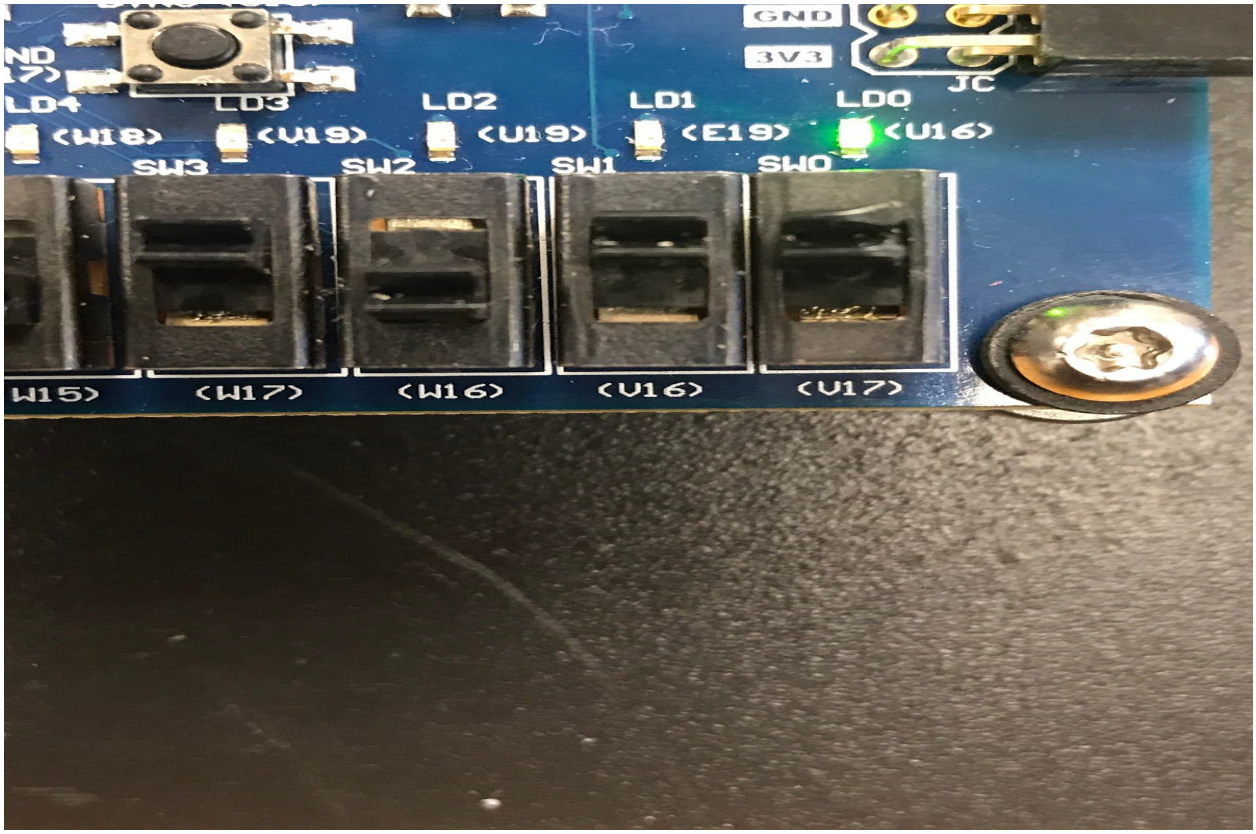


Figure 2B.11) [S0,A,B,Cin] = [1011] ; [Z]= [1]

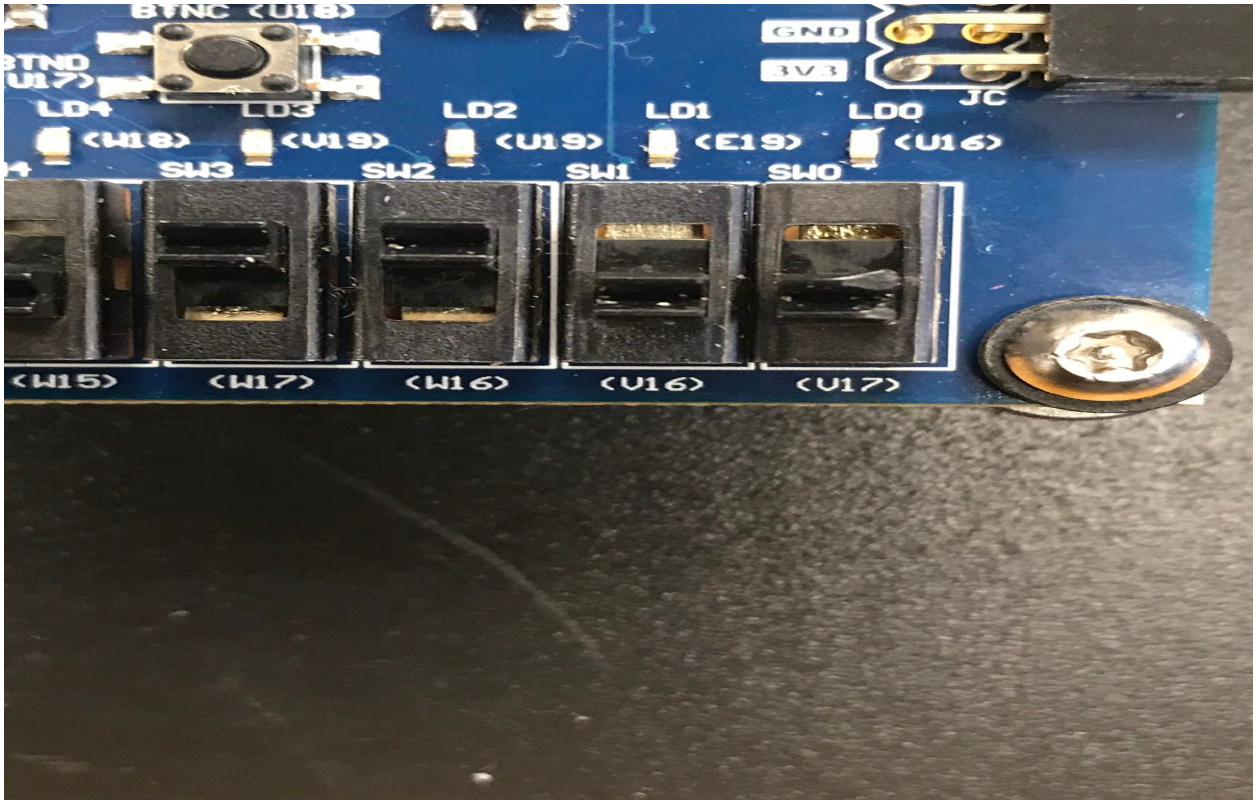


Figure 2B.12) [S0,A,B,Cin] = [1100] ; [Z]= [0]

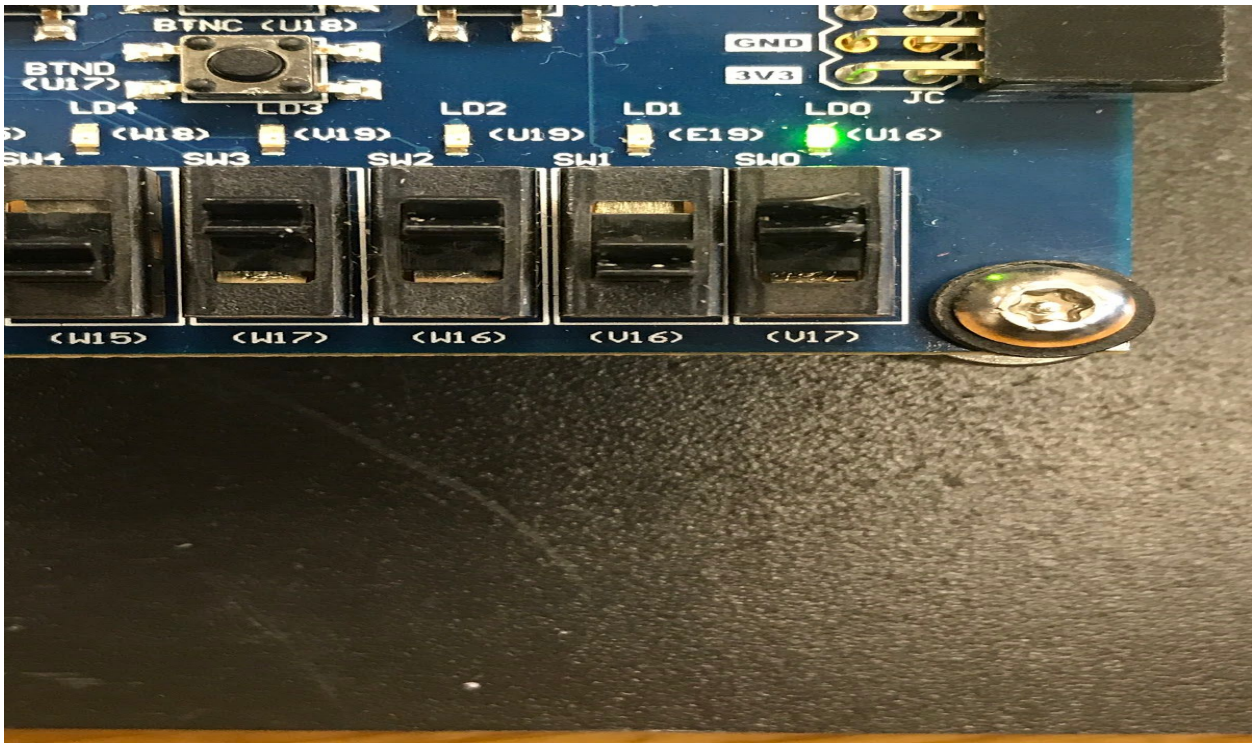


Figure 2B.13) [S0,A,B,Cin] = [1101] ; [Z]= [1]

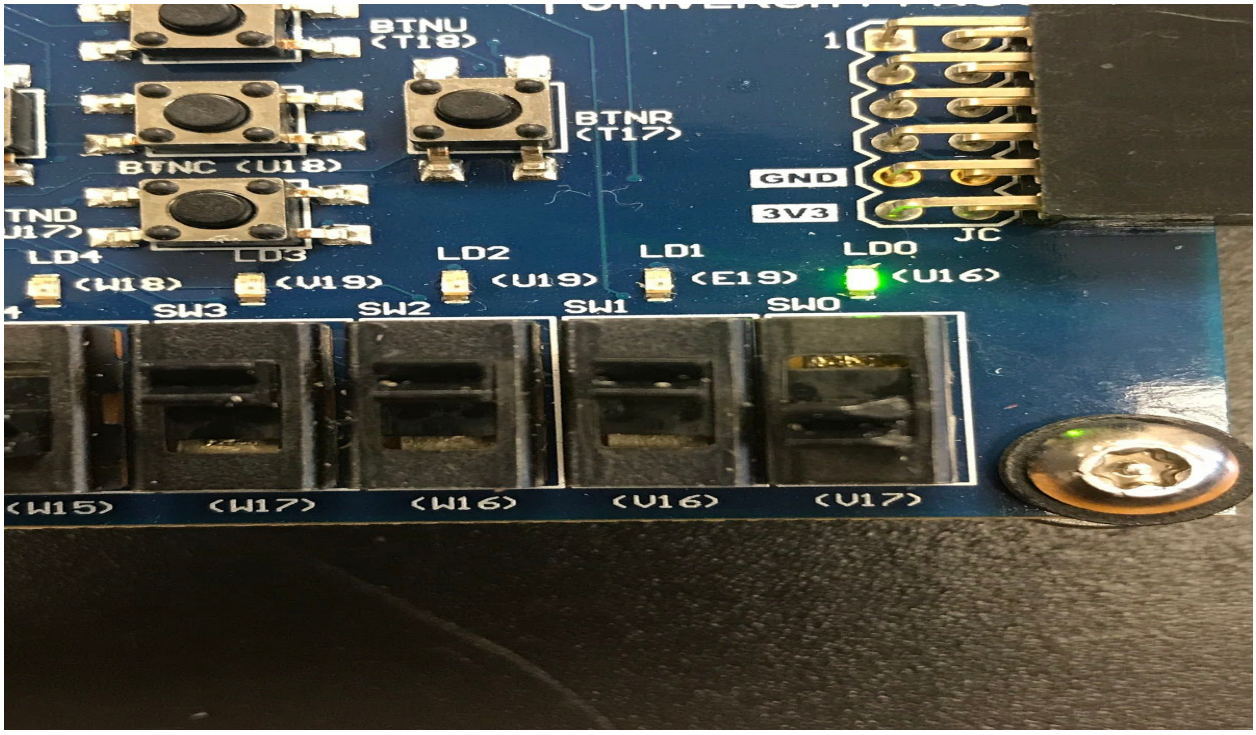


Figure 2B.14) [S0,A,B,Cin] = [1110] ; [Z]= [1]

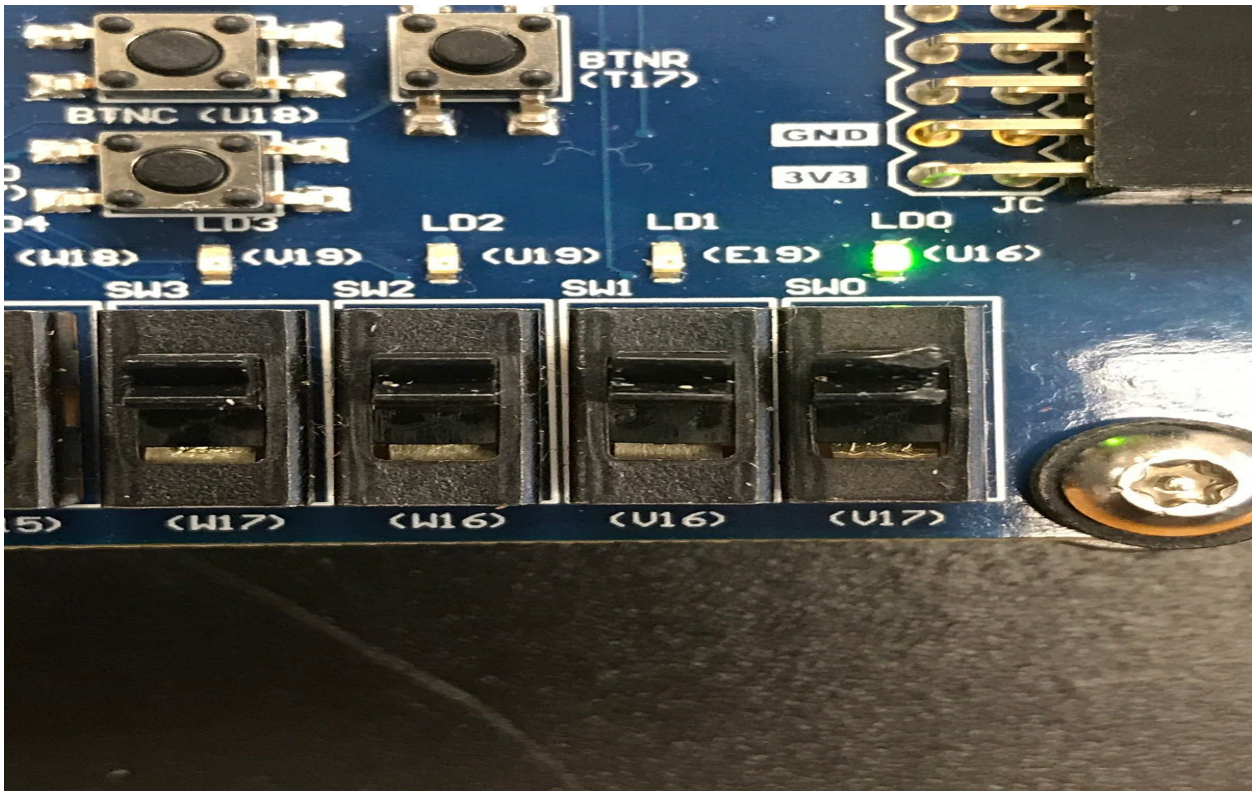


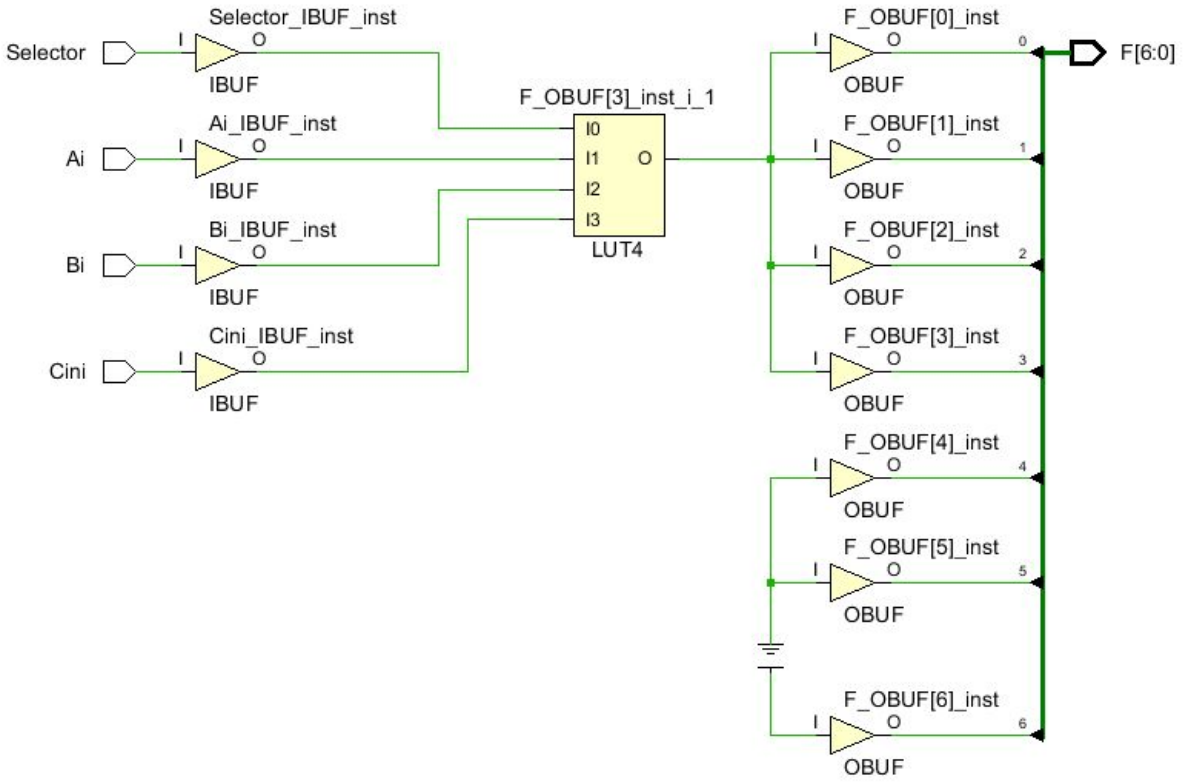
Figure 2B.15) [S0,A,B,Cin] = [1111] ; [Z]= [1]

Part 3) Connect the output of the 2:1 Mux and Adder combination from Part2B to the 7-Segment Display as a component. A fully functioning circuit should display sum/cout(0/1) on the 7-Segment display.

Truth Table

Selector	**Ai**	**Bi**	**Cini**	Cout	Sum	**F**
0	**0**	**0**	**0**	0	0	**0**
0	**0**	**0**	**1**	0	1	**1**
0	**0**	**1**	**0**	0	1	**1**
0	**0**	**1**	**1**	1	0	**0**
0	**1**	**0**	**0**	0	1	**1**
0	**1**	**0**	**1**	1	0	**0**
0	**1**	**1**	**0**	1	0	**0**
0	**1**	**1**	**1**	1	1	**1**
1	**0**	**0**	**0**	0	0	**0**
1	**0**	**0**	**1**	0	1	**0**
1	**0**	**1**	**0**	0	1	**0**
1	**0**	**1**	**1**	1	0	**1**
1	**1**	**0**	**0**	0	1	**0**
1	**1**	**0**	**1**	1	0	**1**
1	**1**	**1**	**0**	1	0	**1**
1	**1**	**1**	**1**	1	1	**1**

Schematic



FPGA Pin Assignments

Name	I/O	PIN
Selector	IN	W17
Ai	IN	W16
Bi	IN	V16
Cini	IN	V17
F[0]	OUT	W7
F[1]	OUT	V5
F[2]	OUT	U5
F[3]	OUT	V8
F[4]	OUT	U8
F[5]	OUT	W6
F[6]	OUT	U7

Results

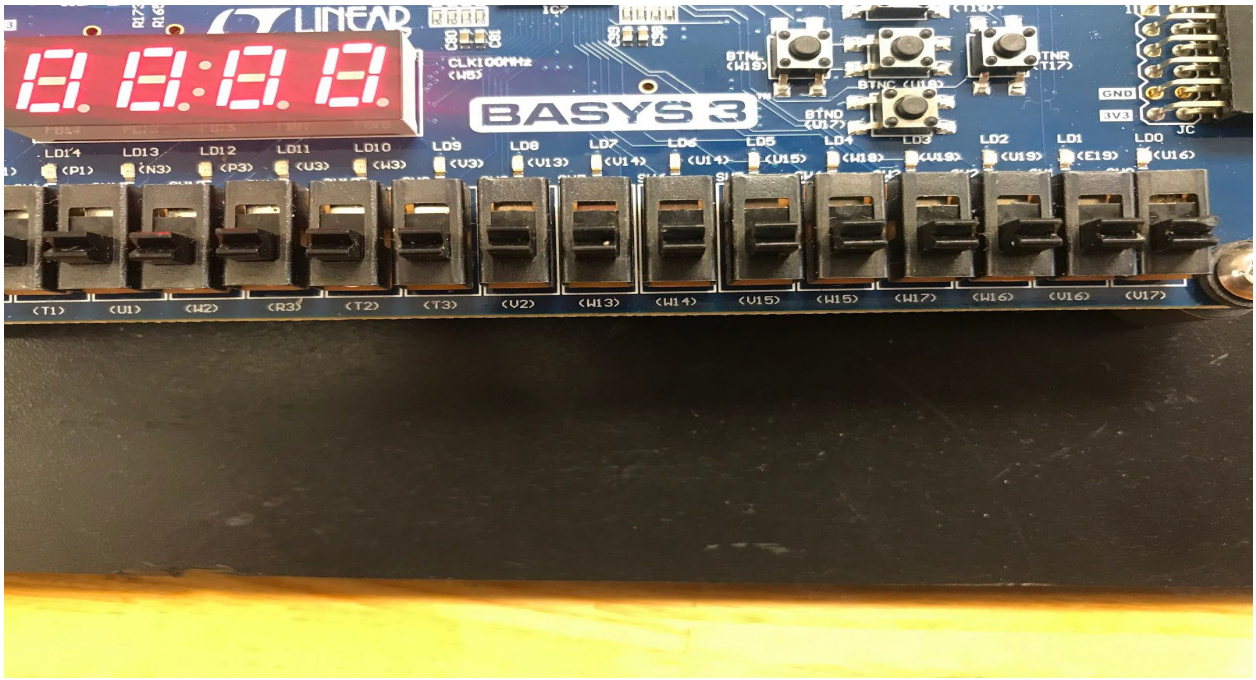


Figure 3.0) [Selector, Ai, Bi, Cini] = [0000]; [F] = [0]

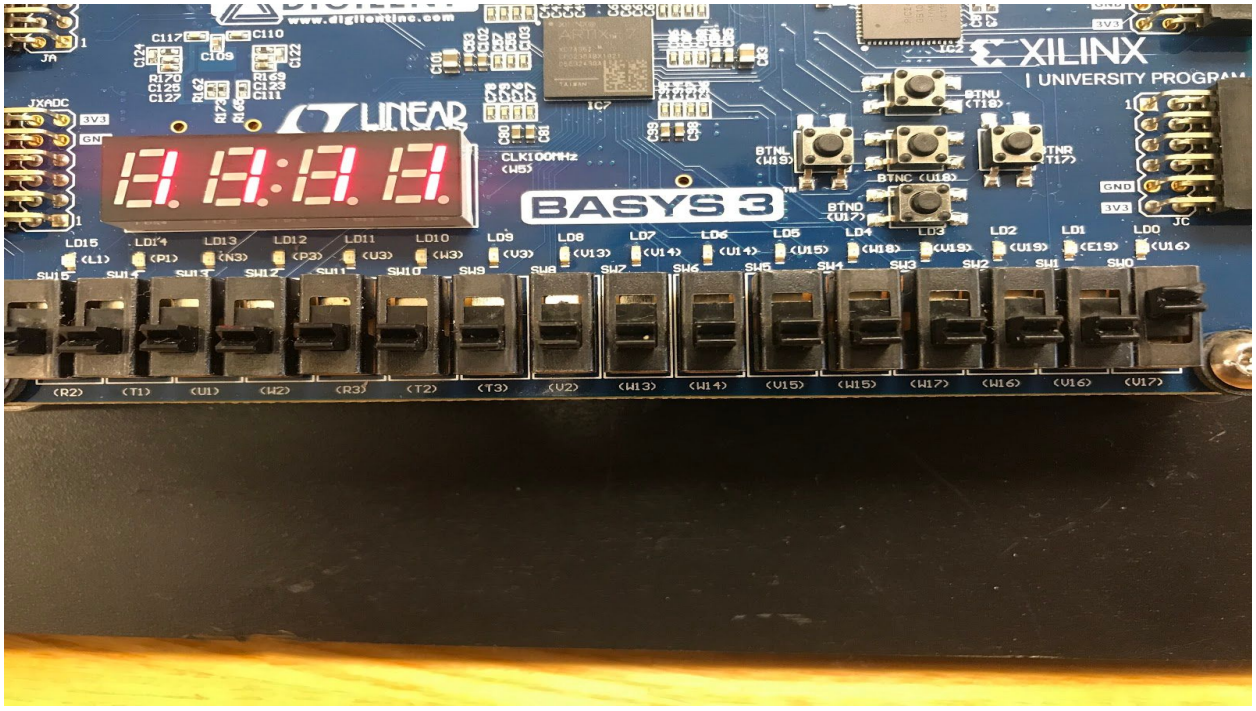


Figure 3.1) [Selector, Ai, Bi, Cini] = [00001]; [F] = [0]

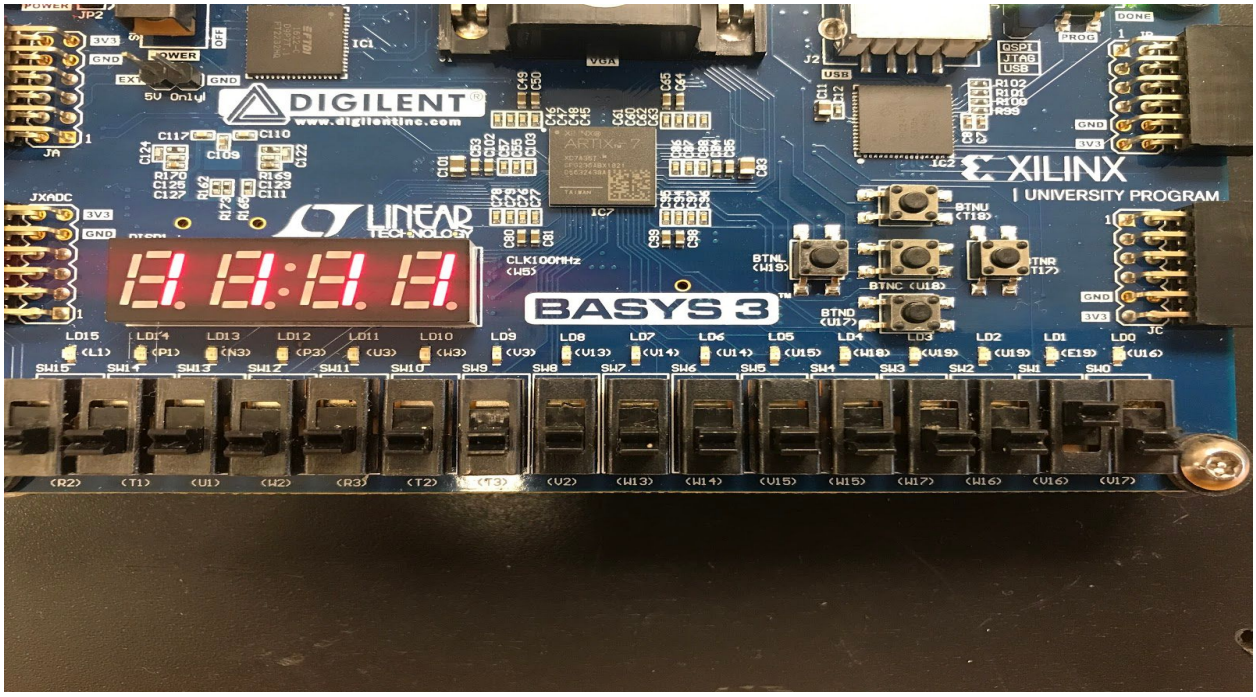


Figure 3.2) [Selector, Ai, Bi, Cini] = [0010]; [F] = [1]

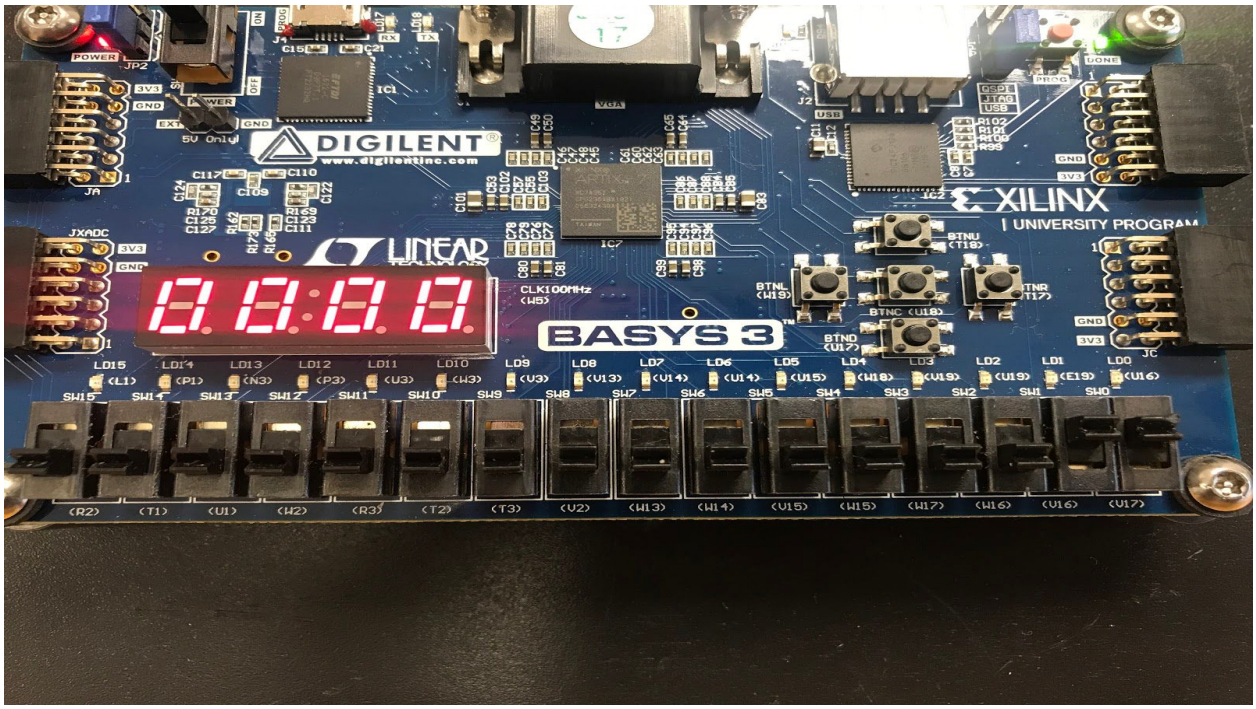


Figure 3.3) [Selector, Ai, Bi, Cini] = [0011]; [F] = [0]

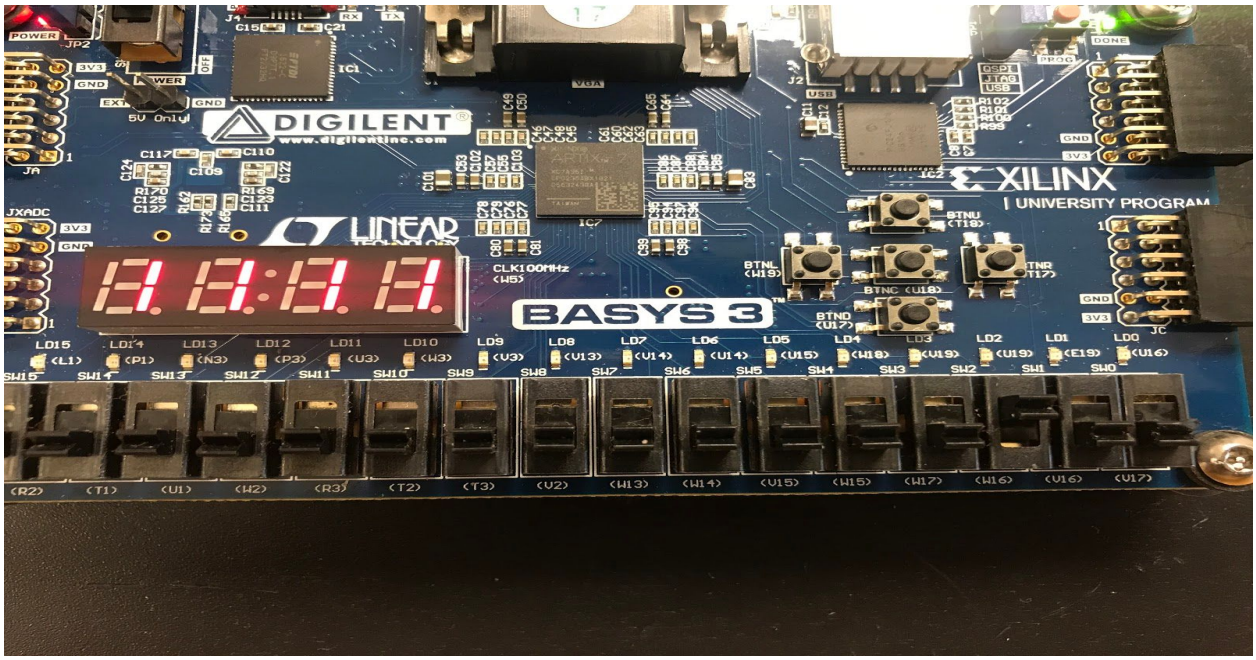


Figure 3.4) [Selector, Ai, Bi, Cini] = [0100]; [F] = [1]

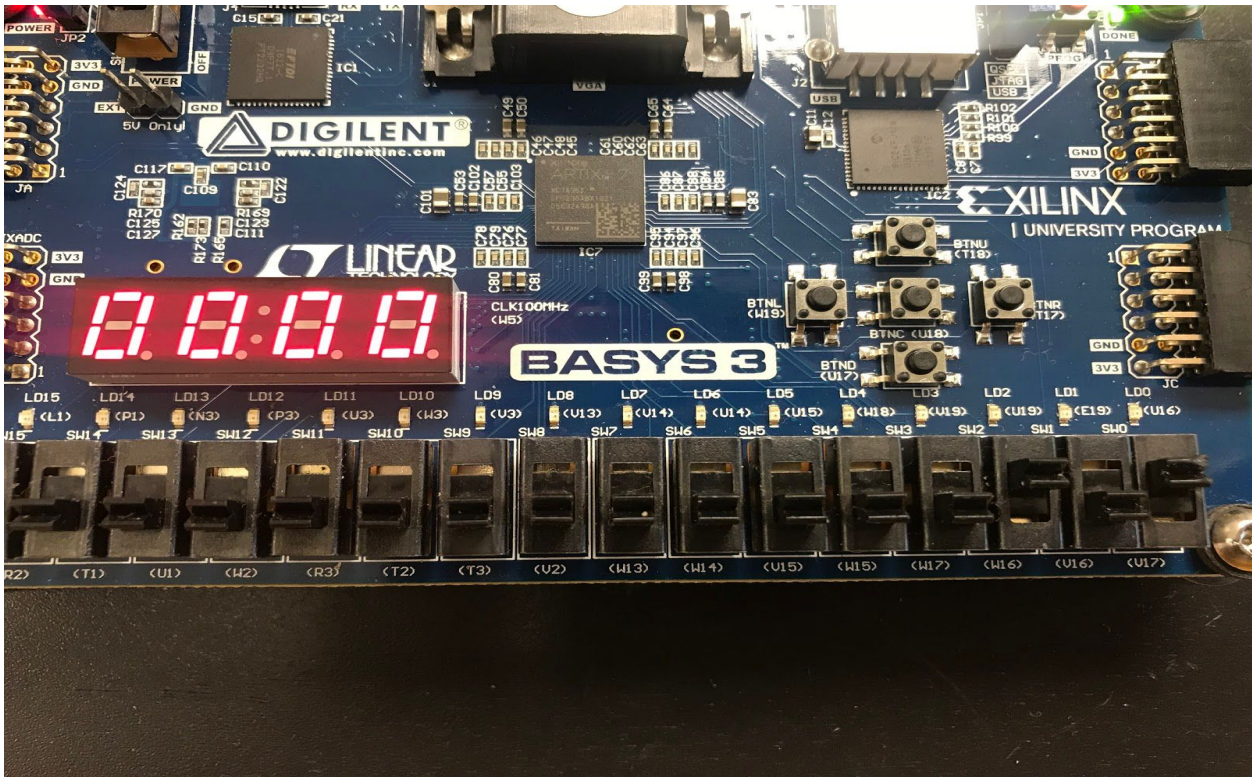


Figure 3.5) [Selector, Ai, Bi, Cini] = [0101]; [F] = [0]

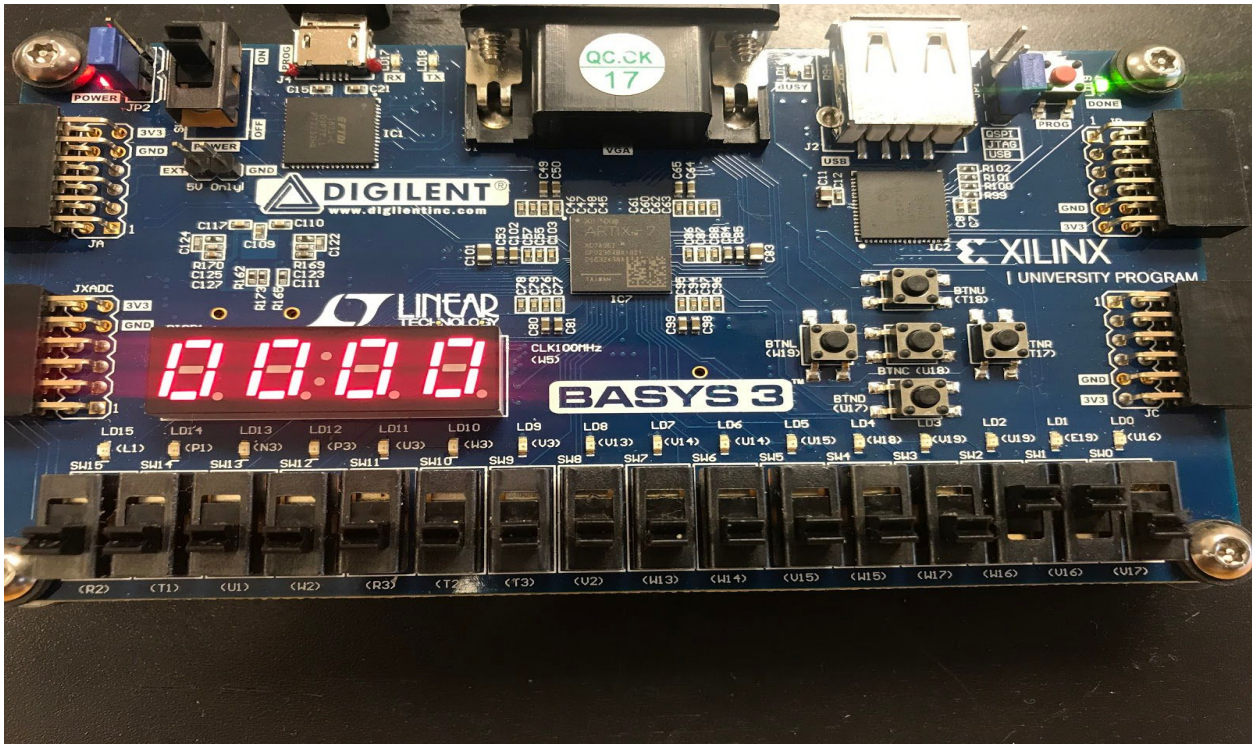


Figure 3.6) [Selector, Ai, Bi, Cini] = [0110]; [F] = [0]

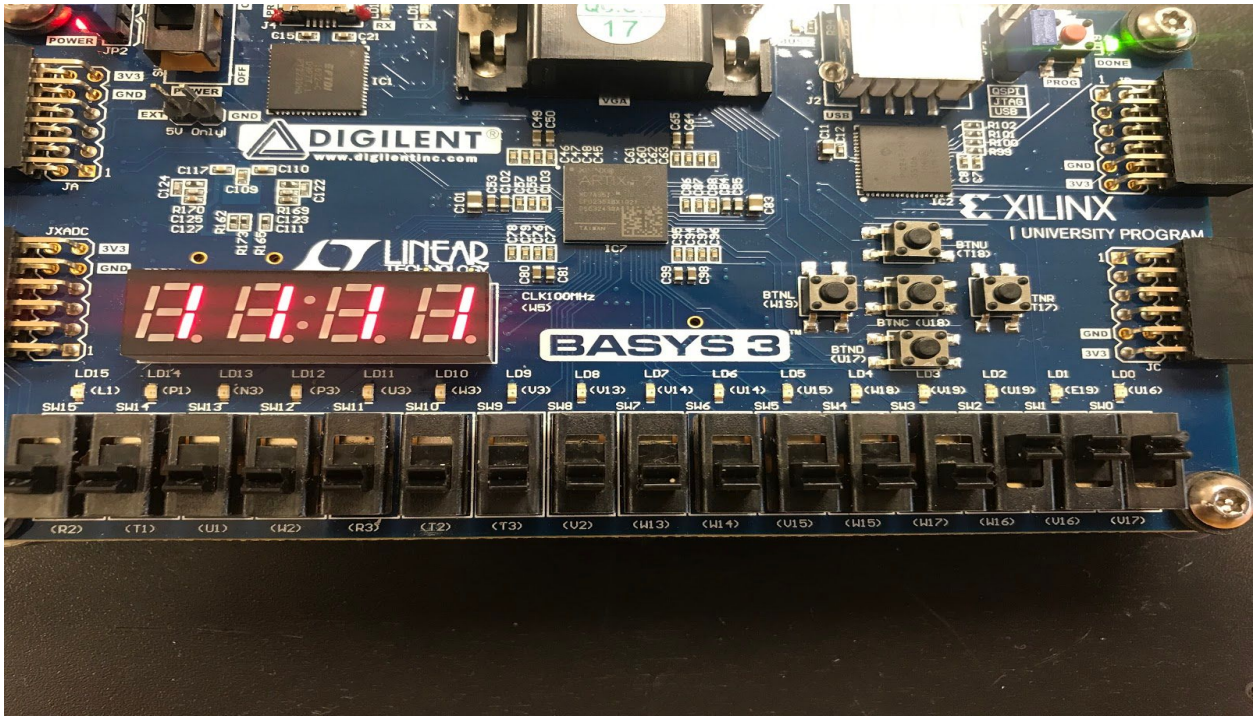


Figure 3.7) [Selector, Ai, Bi, Cini] = [0111]; [F] = [1]

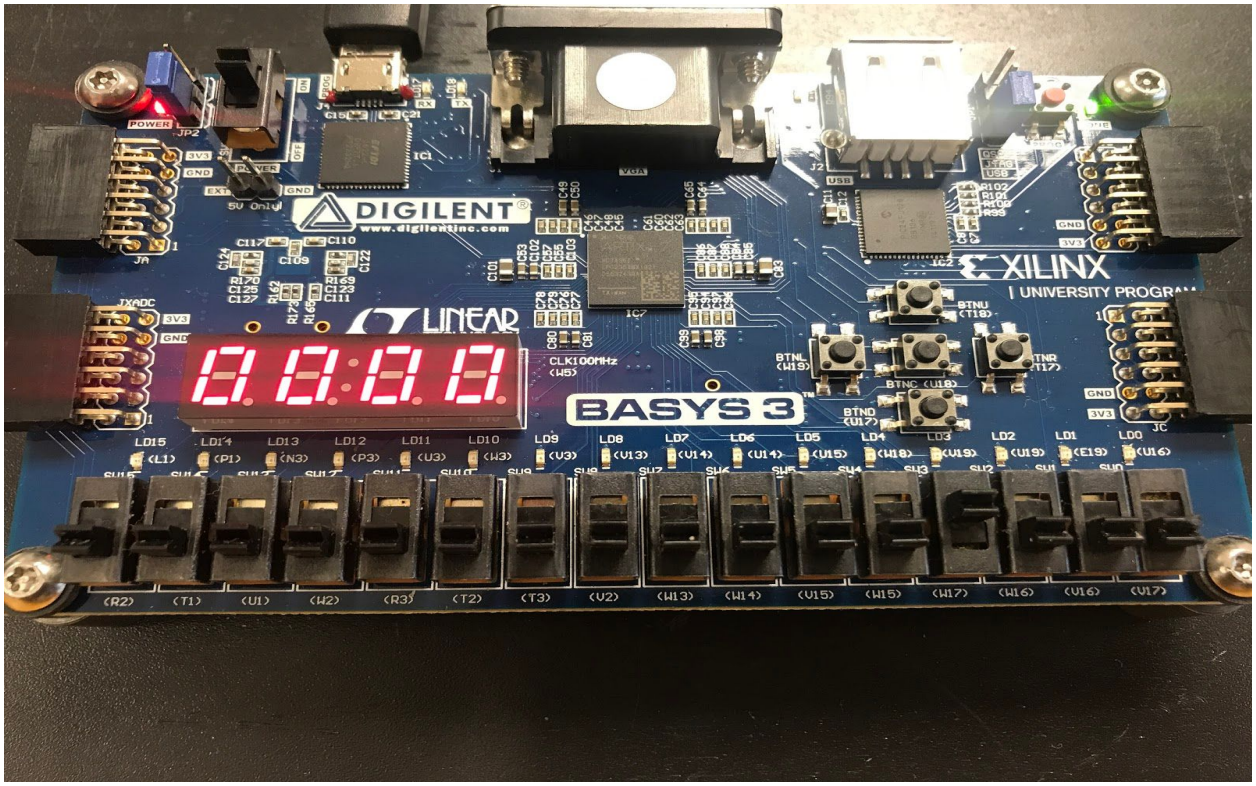


Figure 3.8) [Selector, Ai, Bi, Cini] = [1000]; [F] = [0]

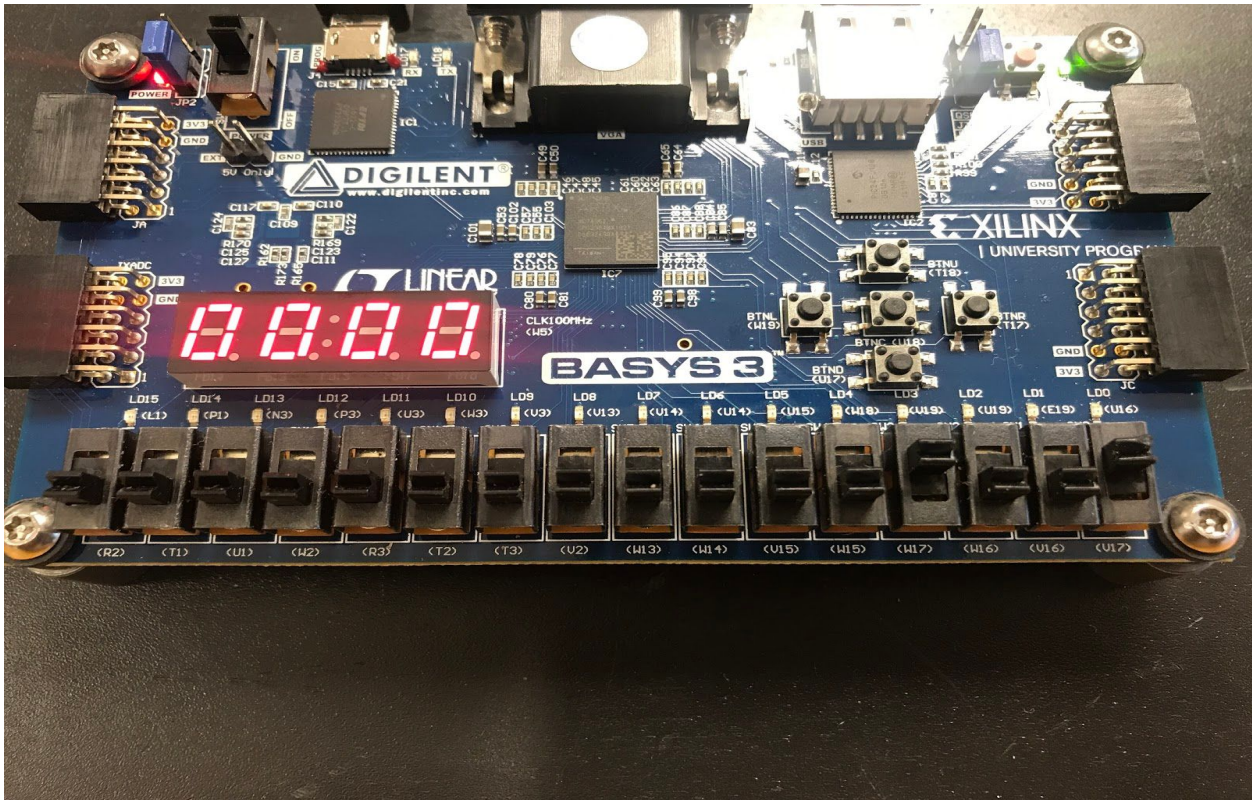


Figure 3.9) [Selector, Ai, Bi, Cini] = [1001]; [F] = [0]

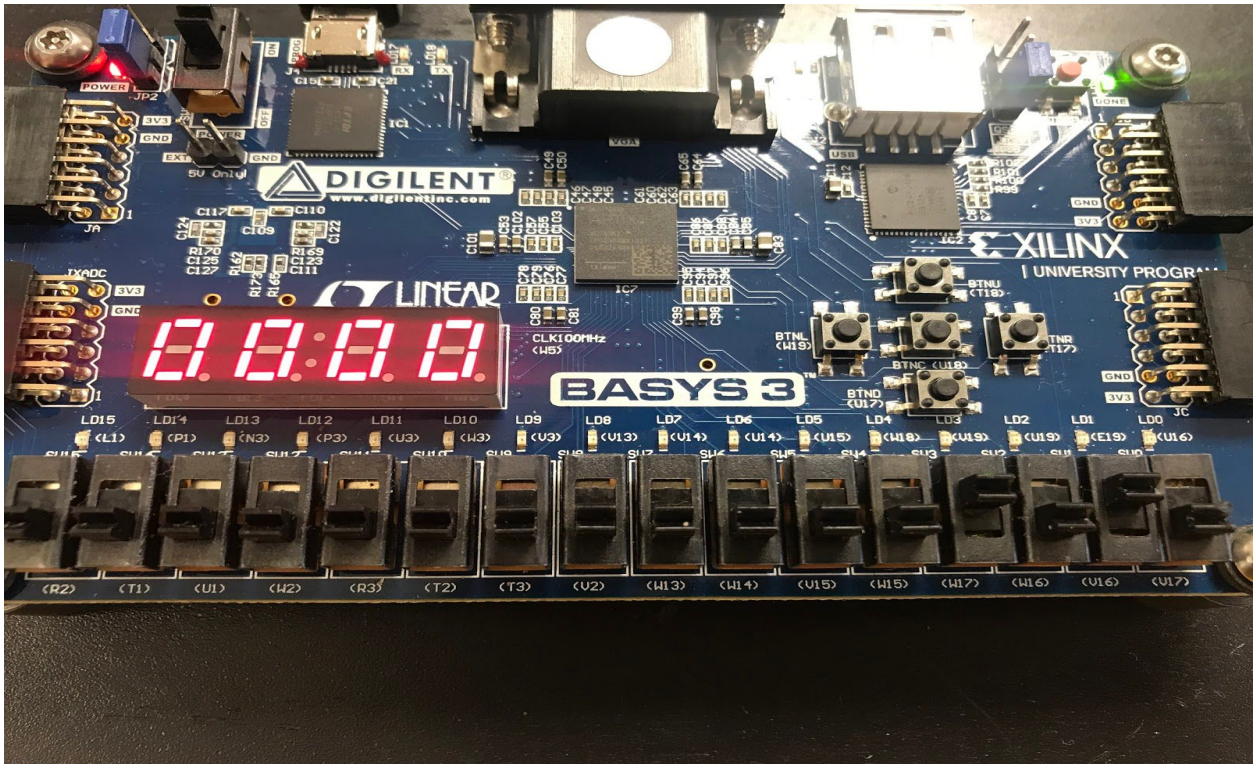


Figure 3.10) [Selector, Ai, Bi, Cini] = [1010]; [F] = [0]

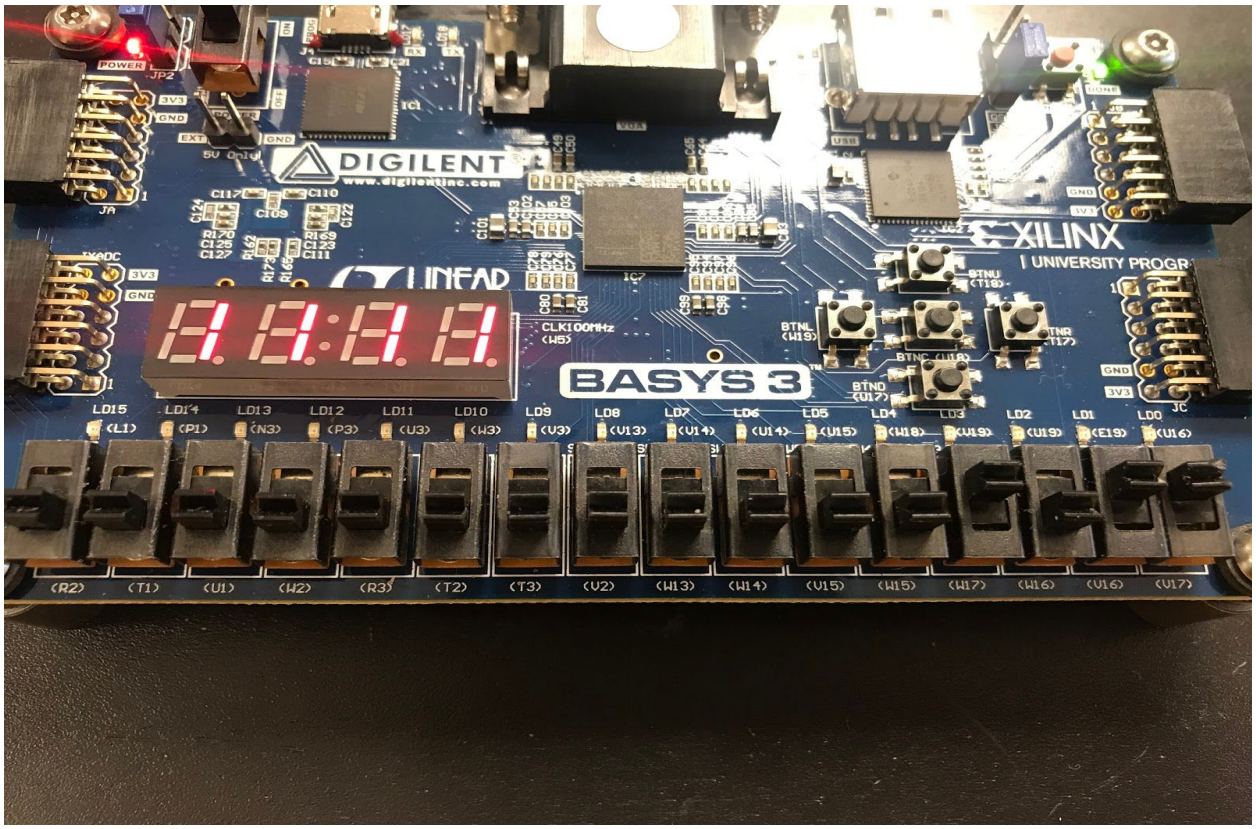


Figure 3.11) [Selector, Ai, Bi, Cini] = [1011]; [F] = [1]

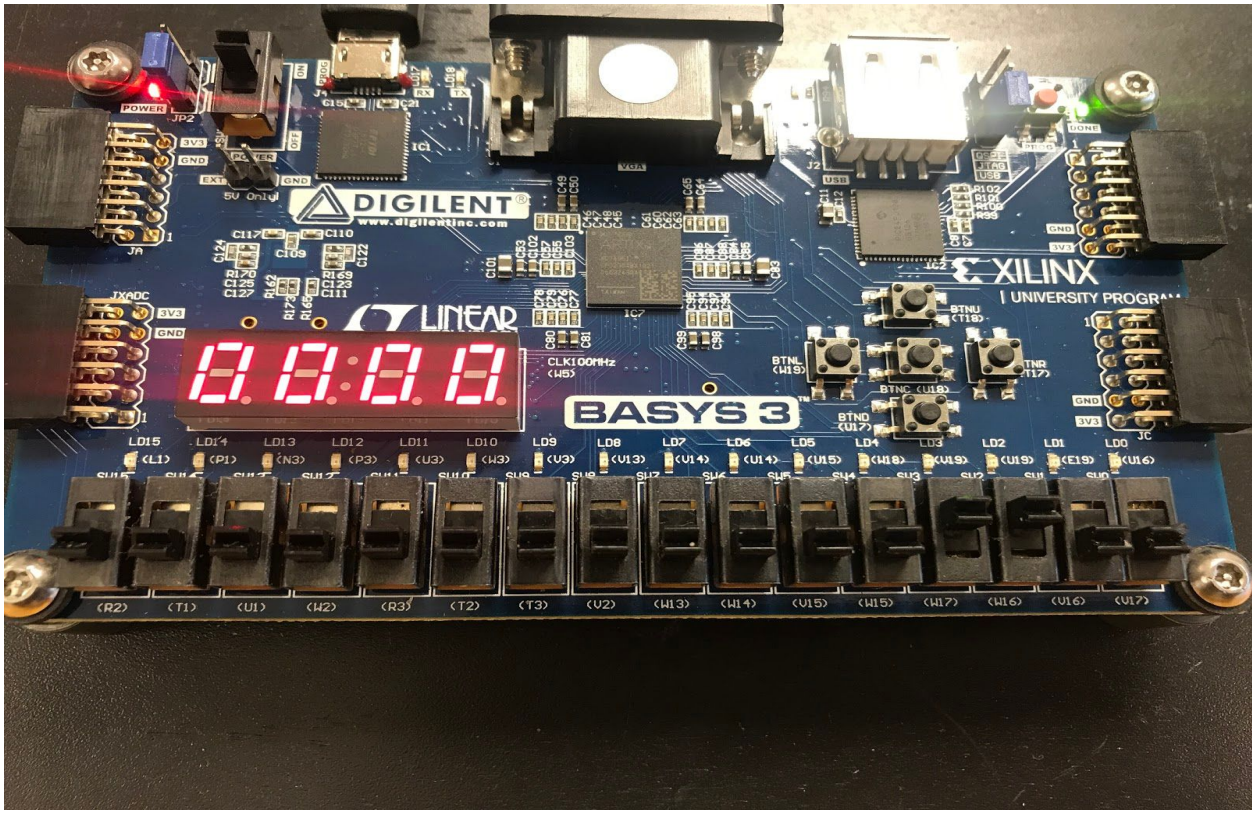


Figure 3.12) [Selector, Ai, Bi, Cini] = [1100]; [F] = [0]

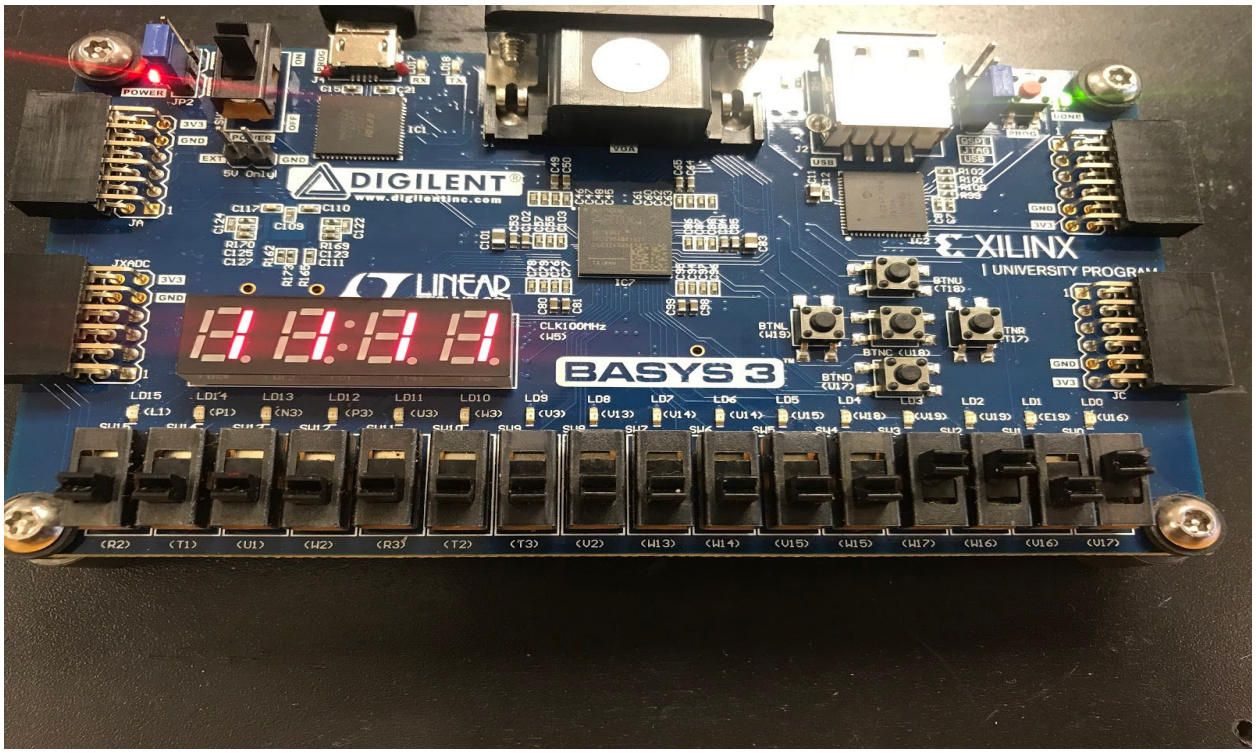


Figure 3.13) [Selector, Ai, Bi, Cini] = [1101]; [F] = [1]

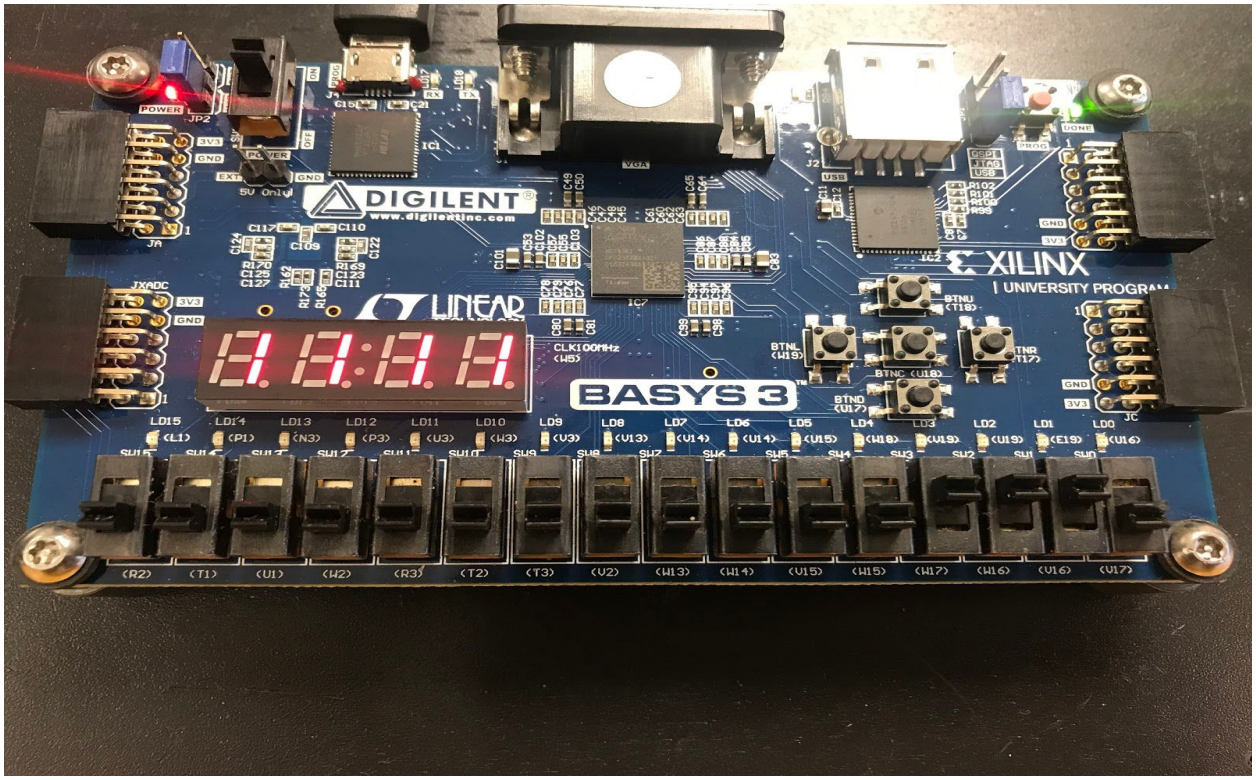


Figure 3.14) [Selector, Ai, Bi, Cini] = [1110]; [F] = [1]

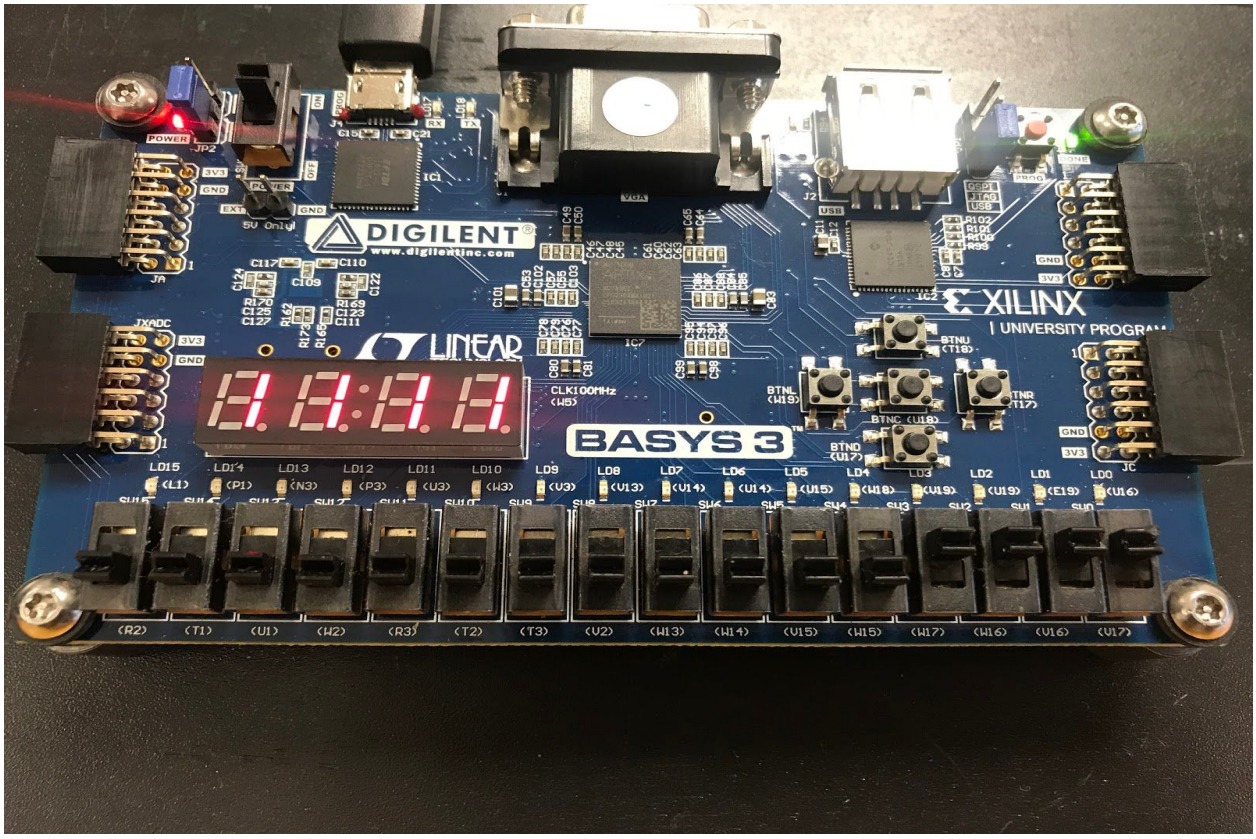


Figure 3.15) [Selector, Ai, Bi, Cini] = [1111]; [F] = [1]

Conclusion:

In this lab we learned about using components to build more complex systems. One difficult part of this lab was figuring out how to add source files from the Adder and Mux to combine them into an Adder to Mux system and later doing a similar process to feed that system into a 7 Segment Display. By completing this lab we feel more confident in our ability to work with components in VHDL to design more complex systems.

Appendices

Part 1 Constraint File)

```
set_property IOSTANDARD LVCMOS33 [get_ports A]
set_property IOSTANDARD LVCMOS33 [get_ports B]
set_property IOSTANDARD LVCMOS33 [get_ports Cin]
set_property IOSTANDARD LVCMOS33 [get_ports Cout]
set_property IOSTANDARD LVCMOS33 [get_ports Sum]
set_property PACKAGE_PIN W16 [get_ports A]
set_property PACKAGE_PIN V16 [get_ports B]
set_property PACKAGE_PIN V17 [get_ports Cin]
set_property PACKAGE_PIN E19 [get_ports Cout]
set_property PACKAGE_PIN U16 [get_ports Sum]
```

Part 1 VHDL Code)

```
entity Lab3Part1 is
        Port (  A : in STD_LOGIC;
                B : in STD_LOGIC;
                Cin : in STD_LOGIC;
                Cout : out STD_LOGIC;
                Sum : out STD_LOGIC);
end Lab3Part1;

architecture Behavioral of Lab3Part1 is
         begin

                Sum <= (A AND B AND Cin) OR (A AND (NOT B) AND (NOT Cin)) OR (B
        AND (NOT A) AND (NOT Cin)) OR (Cin AND (NOT A) AND (NOT B));
                Cout <= (Cin AND (A OR B)) OR (A AND B);


end Behavioral;
```

Part 2 Constraint File)

```
set_property IOSTANDARD LVCMOS33 [get_ports A]
set_property IOSTANDARD LVCMOS33 [get_ports B]
set_property IOSTANDARD LVCMOS33 [get_ports S0]
set_property IOSTANDARD LVCMOS33 [get_ports Z]
set_property PACKAGE_PIN W16 [get_ports S0]
set_property PACKAGE_PIN V17 [get_ports B]
set_property PACKAGE_PIN E19 [get_ports Z]
set_property PACKAGE_PIN V16 [get_ports A]
```

Part 2 VHDL Code)

```
entity Lab3Part2SourceFile is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           S0 : in STD_LOGIC;--selector for the MUX
           Z : out STD_LOGIC);
end Lab3Part2SourceFile;

architecture Behavioral of Lab3Part2SourceFile is

          begin
               process(A, B, S0)
                 begin
                    if ( S0 = '1') then
                     Z <= A;
                    else
                     Z <= B;
                    end if;
               end process;
end Behavioral;
```

Part 2 B Constraint File)

```
set_property IOSTANDARD LVCMOS33 [get_ports A]
set_property IOSTANDARD LVCMOS33 [get_ports B]
set_property IOSTANDARD LVCMOS33 [get_ports Cin]
set_property IOSTANDARD LVCMOS33 [get_ports S0]
set_property IOSTANDARD LVCMOS33 [get_ports Z]
set_property PACKAGE_PIN W17 [get_ports S0]
set_property PACKAGE_PIN W16 [get_ports A]


set_property PACKAGE_PIN V17 [get_ports Cin]
set_property PACKAGE_PIN V16 [get_ports B]
set_property PACKAGE_PIN U16 [get_ports Z]
```

Part 2 B VHDL Code)

```
entity LAb3Part2BSourceFile is
    Port ( A : in bit;
           B : in bit;
           Cin : in bit;
           S0 : in bit;
           Z : out bit);
end LAb3Part2BSourceFile;

architecture Behavioral of LAb3Part2BSourceFile is

        component Lab3Part2SourceFile                          --MUX  Same convention
from part 2 MUX
                Port (A,B,S0: in bit;
                Z: out bit );
          end component;
        component Lab3Part1                          --ADDER Same Convention from part 1
                 Port (A, B, Cin : in bit;
                      Cout, Sum : out bit );
        end component;

     signal CoutTemp, SumTemp : bit;

begin


ADD: Lab3Part1 port map(A, B, Cin, CoutTemp, SumTemp);

MUX2to1: Lab3Part2SourceFile port map(CoutTemp, SumTemp, S0, Z);

end Behavioral;


—if select bit S0 is high then the Cout is high, otherwise the Sum is high

—In order to use the previous adder mux from part 1 and 2. You need to add them add source
files in the current project
```

Part 3 Constraints)

```
set_property IOSTANDARD LVCMOS33 [get_ports Ai]
set_property IOSTANDARD LVCMOS33 [get_ports Bi]
set_property IOSTANDARD LVCMOS33 [get_ports Cini]
set_property IOSTANDARD LVCMOS33 [get_ports Selector]
set_property IOSTANDARD LVCMOS33 [get_ports {F[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {F[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {F[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {F[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {F[2]}]
set_property PACKAGE_PIN U7 [get_ports {F[6]}]
set_property PACKAGE_PIN W6 [get_ports {F[5]}]
set_property PACKAGE_PIN U8 [get_ports {F[4]}]
set_property PACKAGE_PIN V8 [get_ports {F[3]}]
set_property PACKAGE_PIN U5 [get_ports {F[2]}]
set_property PACKAGE_PIN V5 [get_ports {F[1]}]
set_property PACKAGE_PIN W7 [get_ports {F[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {F[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {F[0]}]
set_property PACKAGE_PIN W17 [get_ports Selector]
set_property PACKAGE_PIN W16 [get_ports Ai]
set_property PACKAGE_PIN V16 [get_ports Bi]
set_property PACKAGE_PIN V17 [get_ports Cini]
```

Part 3 VHDL Code)

```
entity Lab3Part3SourceFile is --7Seg
    Port ( Ai : in bit;
           Bi : in bit;
           Cini : in bit;
           Selector : in bit;
            F: out bit_vector(6 downto 0));
end Lab3Part3SourceFile;

architecture Behavioral of Lab3Part3SourceFile is

      component Lab3Part2BSourceFile                                    --Mux joined with Adder
           Port (A,B,Cin,S0: in bit;
                 Z: out bit );
      end component;

           signal  Results : bit;

      begin
           MUXandAdder: Lab3Part2BSourceFile port map(Ai, Bi,Cini, Selector, Results);

           process(Results)
              begin
                 case Results is
                    when '0' => F <= "1000000";--0
                    when '1' => F <= "1001111";--1
                    when others => F <= "0000001"; --null
                  end case;
           end process;
end Behavioral;
```