Lab 3 Report

ECE 4525

Mack Usmanova

Jena Francis

9/17/25

Table of contents

Introduction

The purpose of this lab was to test fault modeling and timing behavior in combinational logic circuits through VHDL code and FPGA implementation. The main objectives for this lab were creating and detecting time delays in combinational circuits and devising and applying test vectors to identify assumed stuck-at and bridging faults. In task one, delays are introduced into a circuit and verified through simulation and hardware testing with the oscilloscope. In task two, faulty circuits are models and test vectors are devised to distinguish them from correct circuits using dip-switches inputs and LED outputs. These tasks reinforce concepts of fault detection, timing analysis, and digital system verification.

Procedure

Task One:

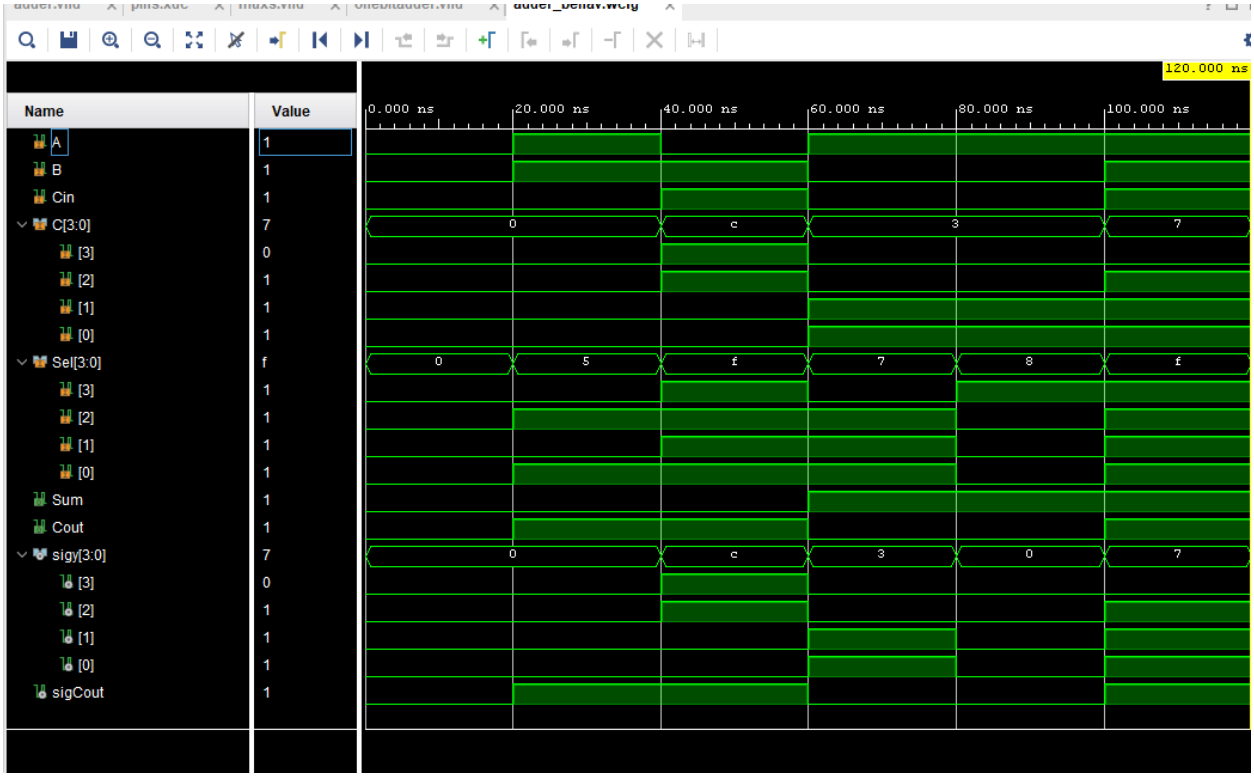

Fig 1: This shows the behavioral simulation for task one

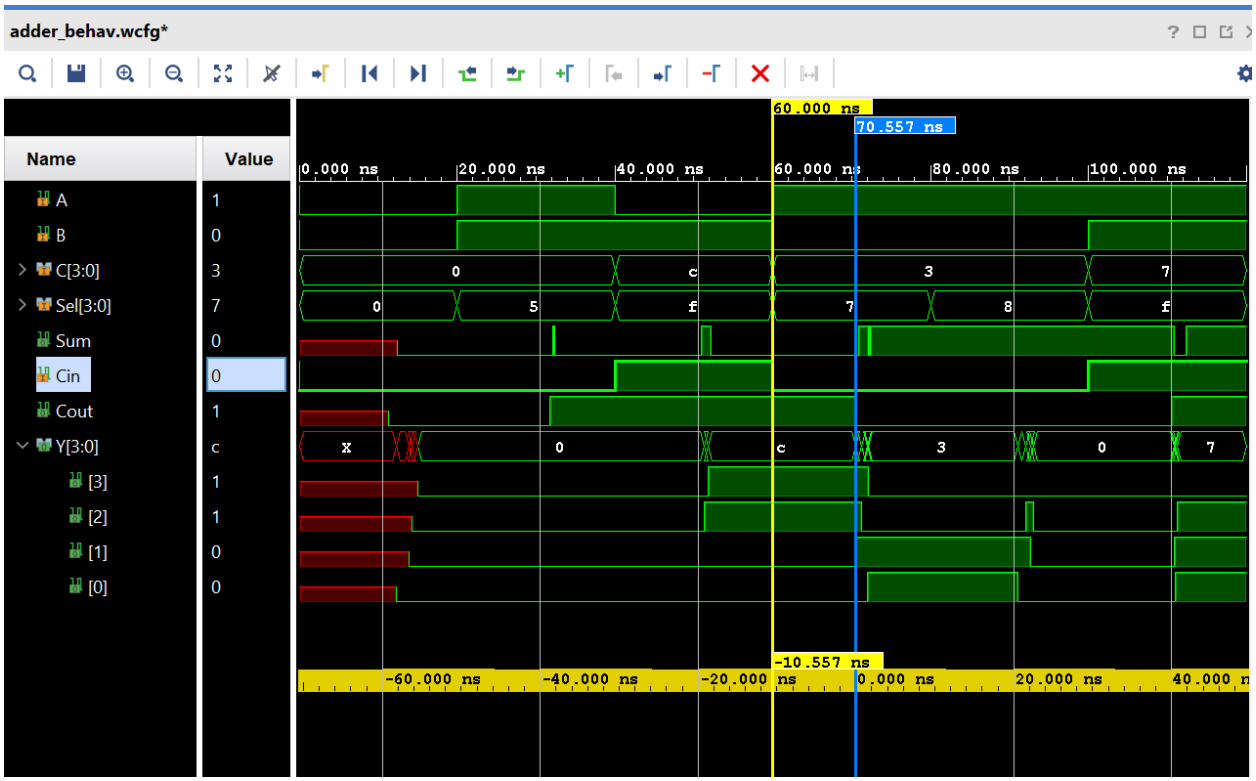

Fig 2: post route simulation Cin>Cout

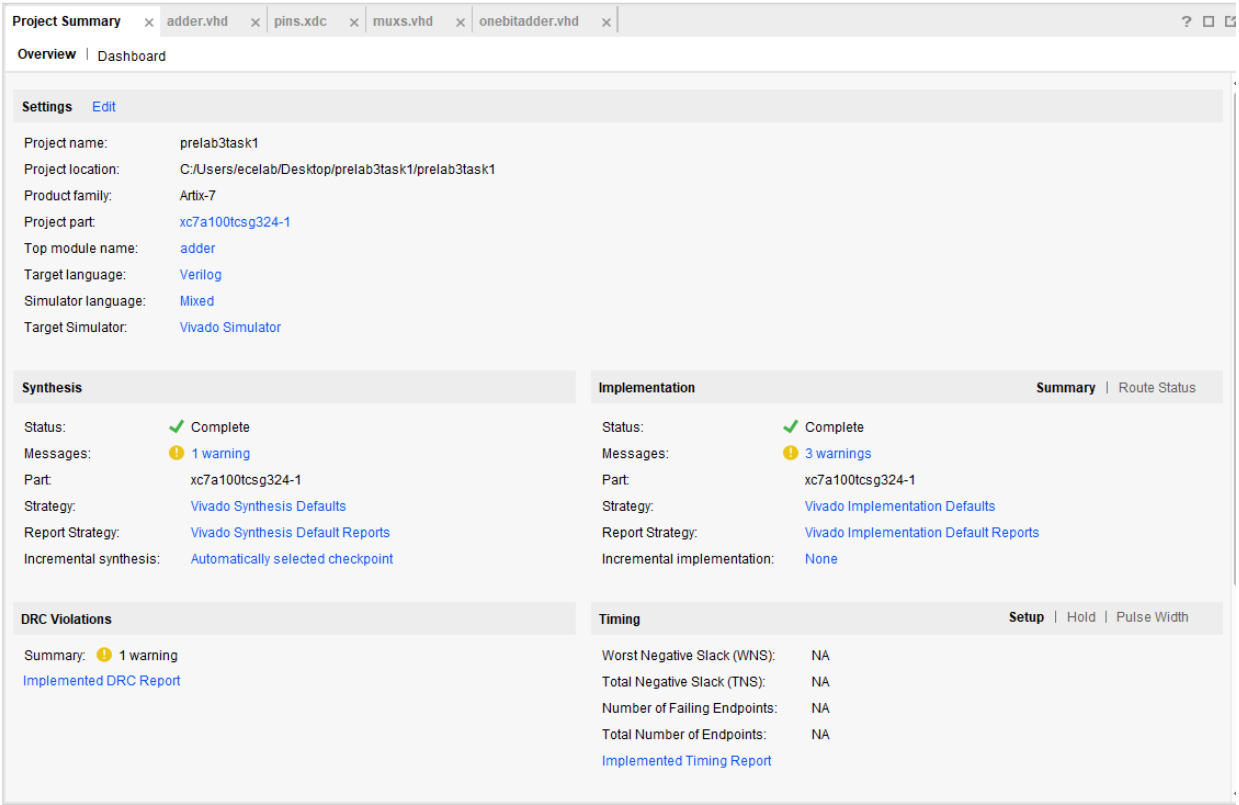

Fig 3: This is the Top of page summary for task one

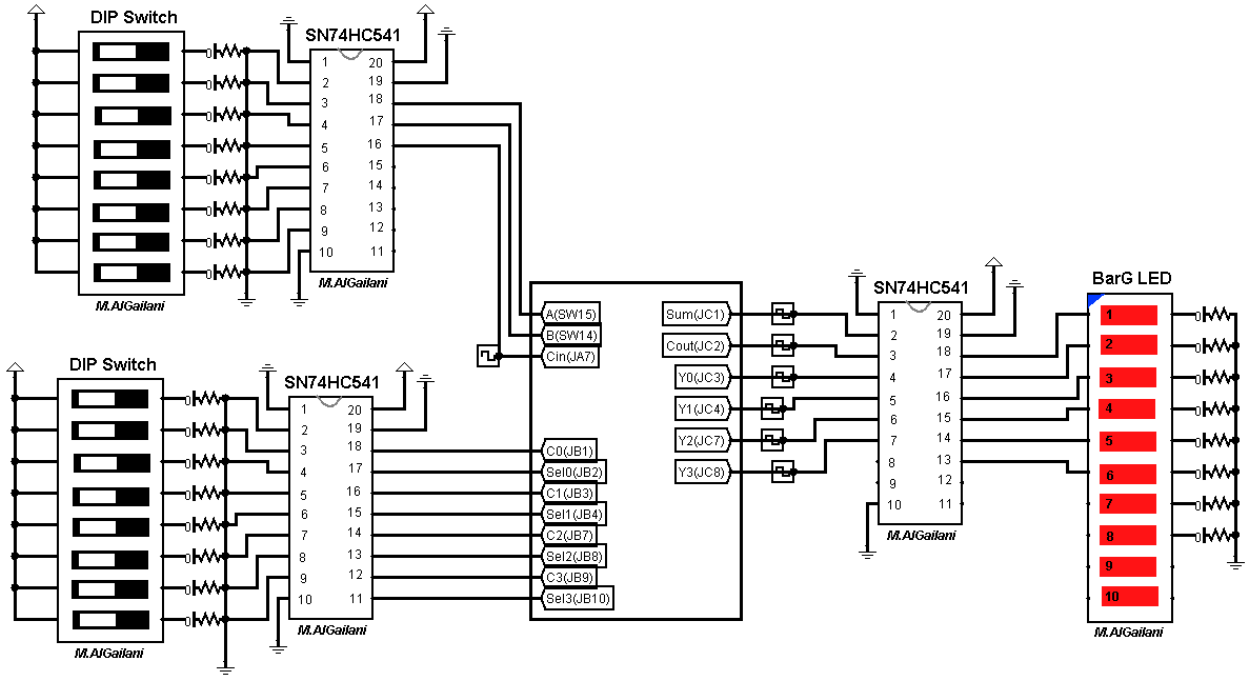

Fig 4: This is the schematic for task one showing the pins with input Cin and outputs Sum, Cout, Y0, Y1, Y2, Y3 connected to logic analyzer.

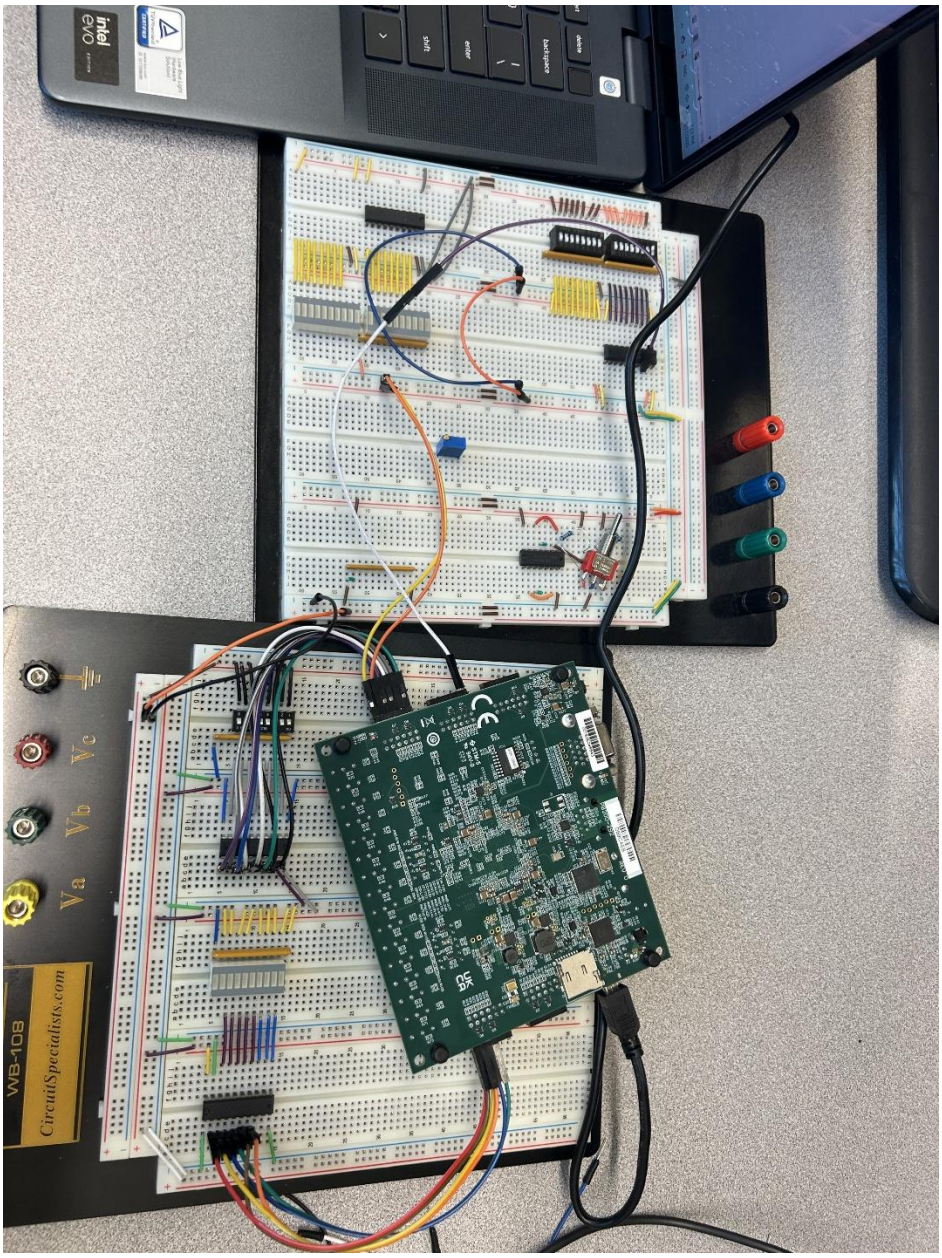

Fig 5: This is the breadboard set for task one. Cin, C0, C1, C2, C3, sel0, sel1, sel2, and sel3 are set as dipswitches for inputs. Sum, Cout, y0, y1, y2, and y3 are set as LED for outputs (left to right). In this photo A, B and Cin (SW14, SW15, and dipswitch in the left breadboard) are set low so all outputs are low.

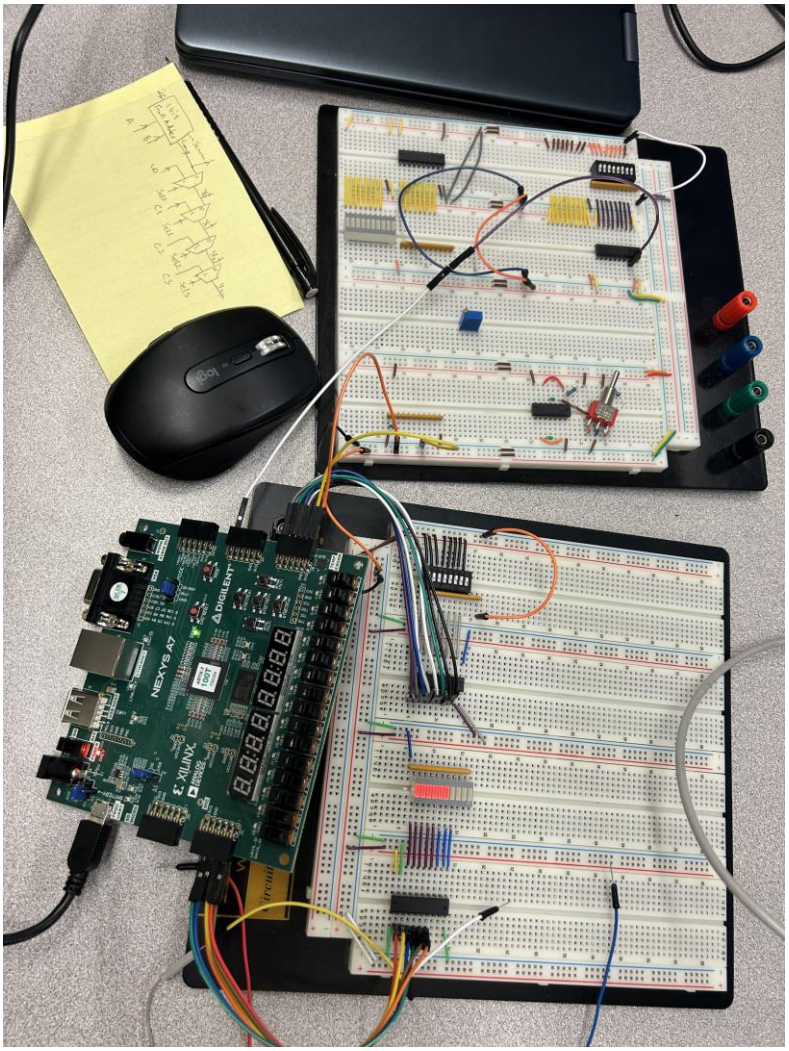

Fig 6: This is the bread board set up for task one. Cin, C0, C1, C2, C3, sel0, sel1, sel2, and sel3 are set as dipswitches for inputs. Cout, sum, y0, y1, y2, and y3 are set as LED for outputs. In this photo A, B and Cin (SW14, SW15, and dipswitch in the left breadboard) are set high so all outputs are high.

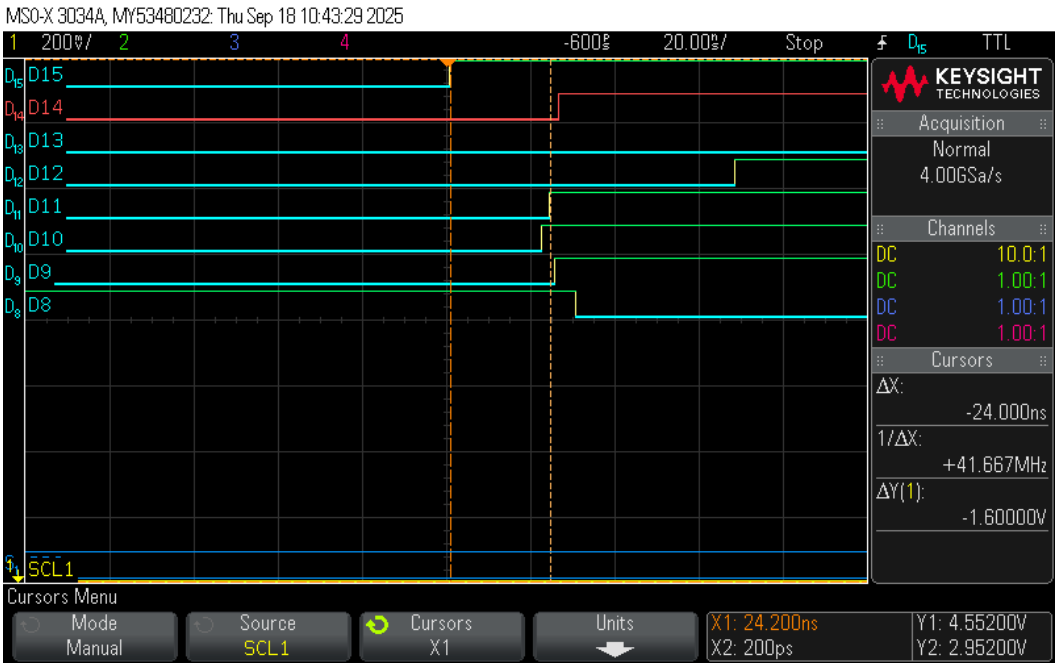

Fig 7: This shows the logic analyzer view showing the delays

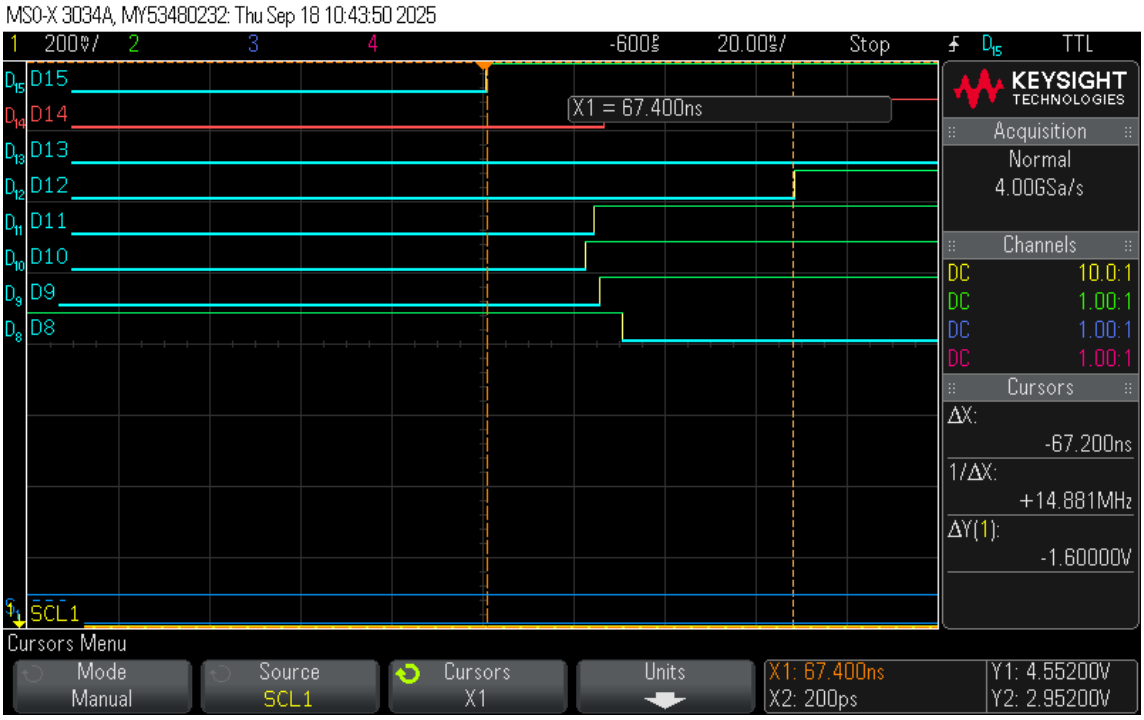

Fig 8: This shows the logic analyzer view showing the delays

Fig 9: This shows the logic analyzer view showing the delays

Task Two:

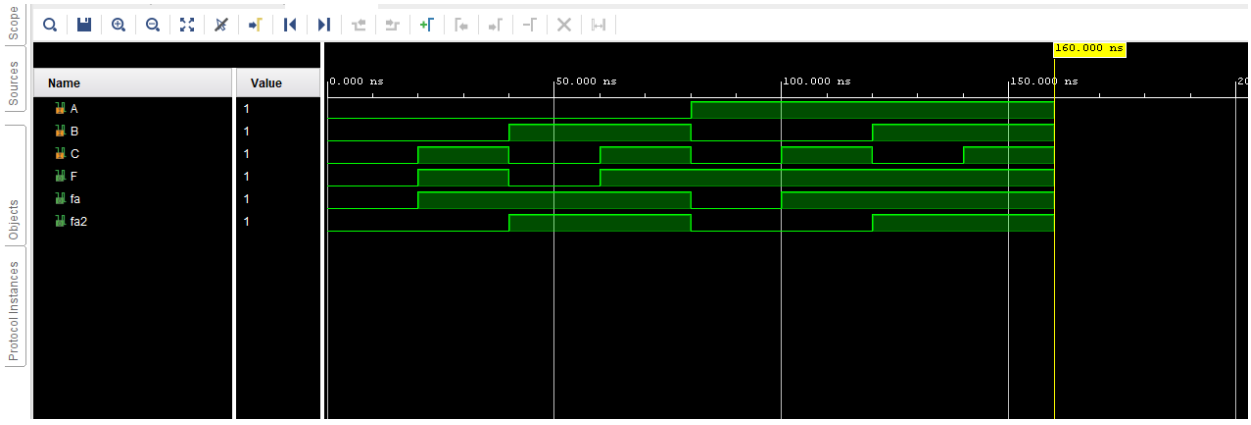

Fig 10: Behavioral simulation for task two which shows both stuck-at-fault(fa) and bridging fault(fa2).

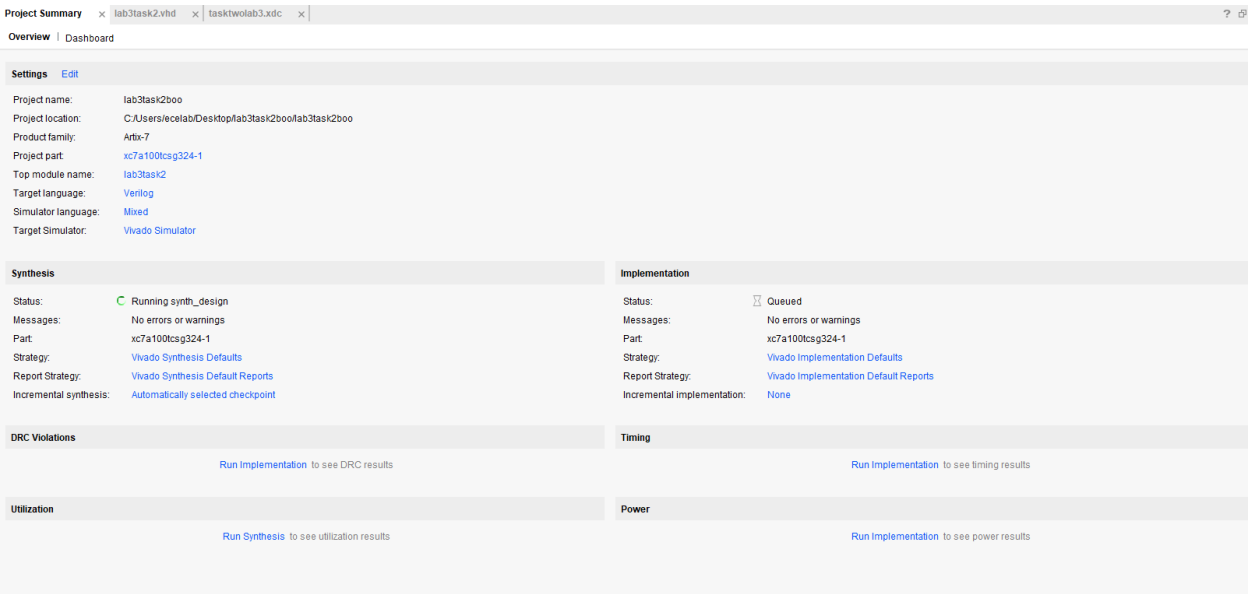

Fig 11: This is the Top of summary report for task two

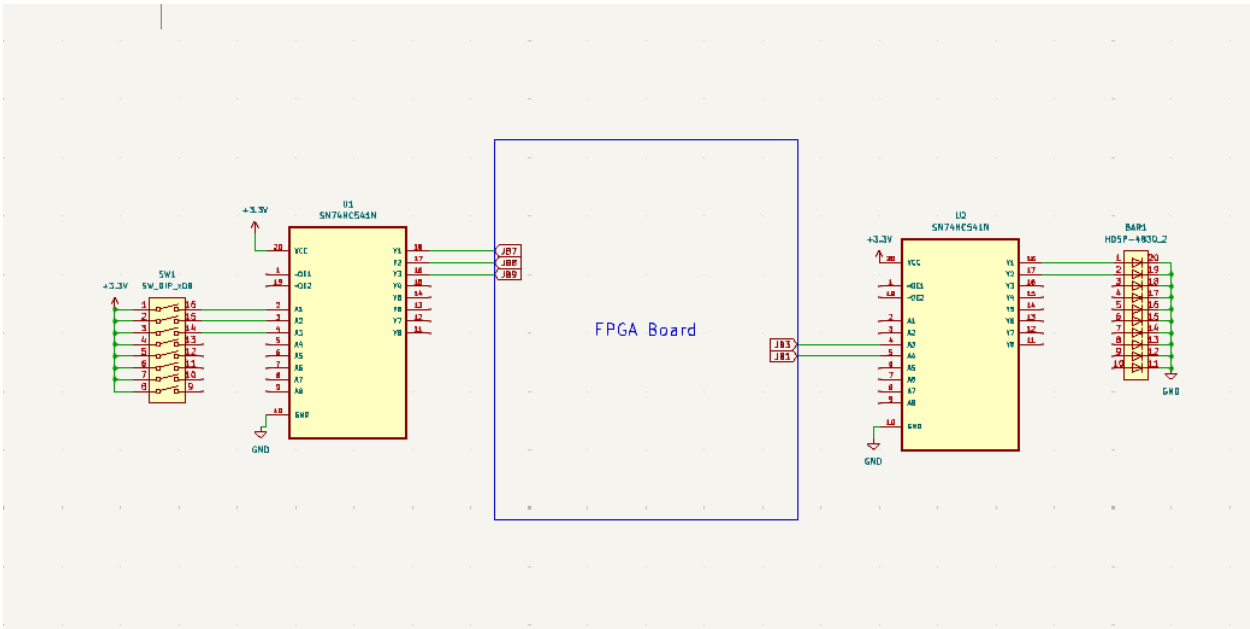

Fig 12: This is the schematic for task two. Inputs A, B, and C are dipswitches and are set to JB7, JB8, and JB9. Outputs are LEDs and are put to JB3 and JB1.

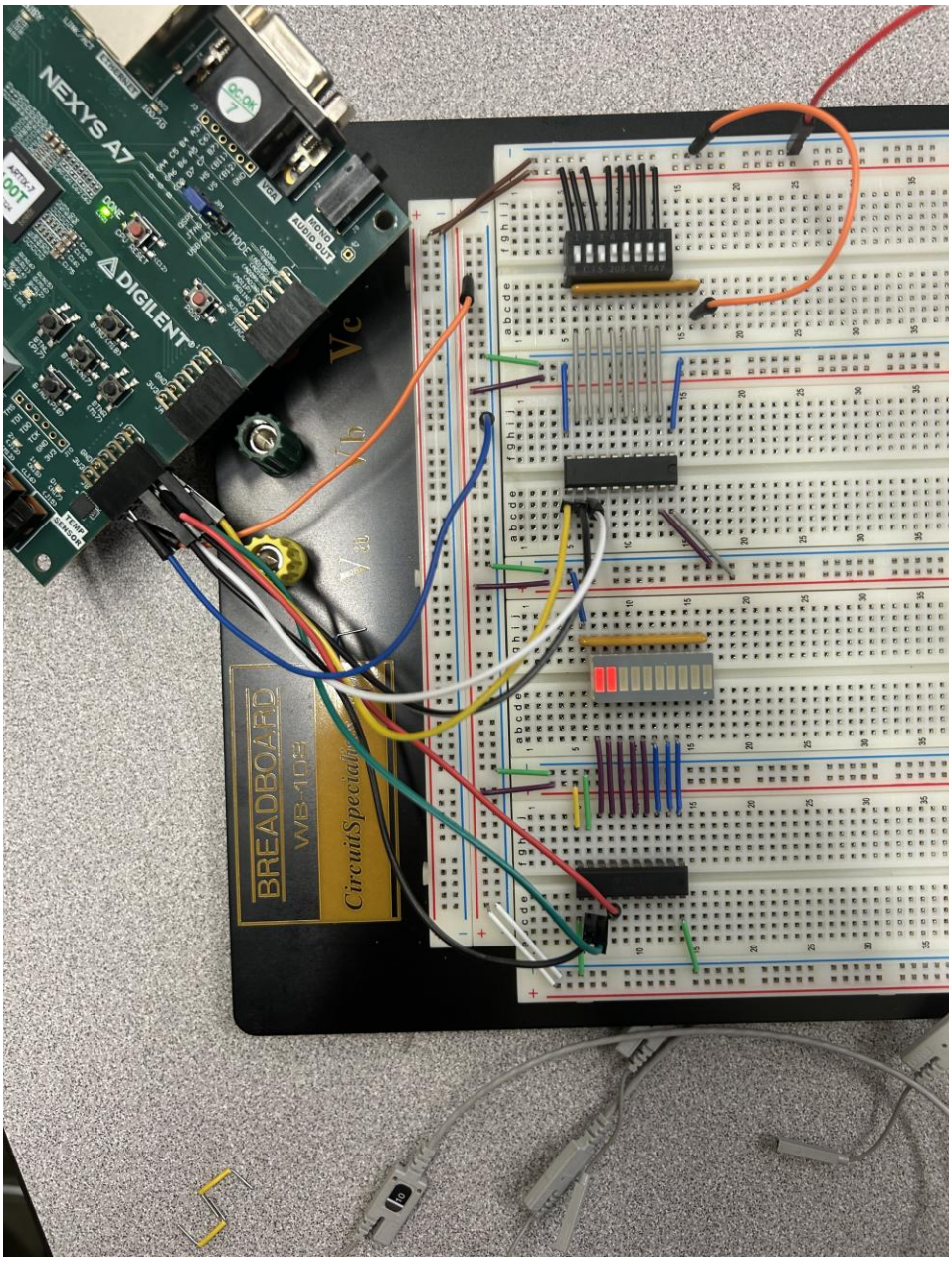

Fig 13: This shows when ABC=010 then f=0 fa=1 and fa2=1

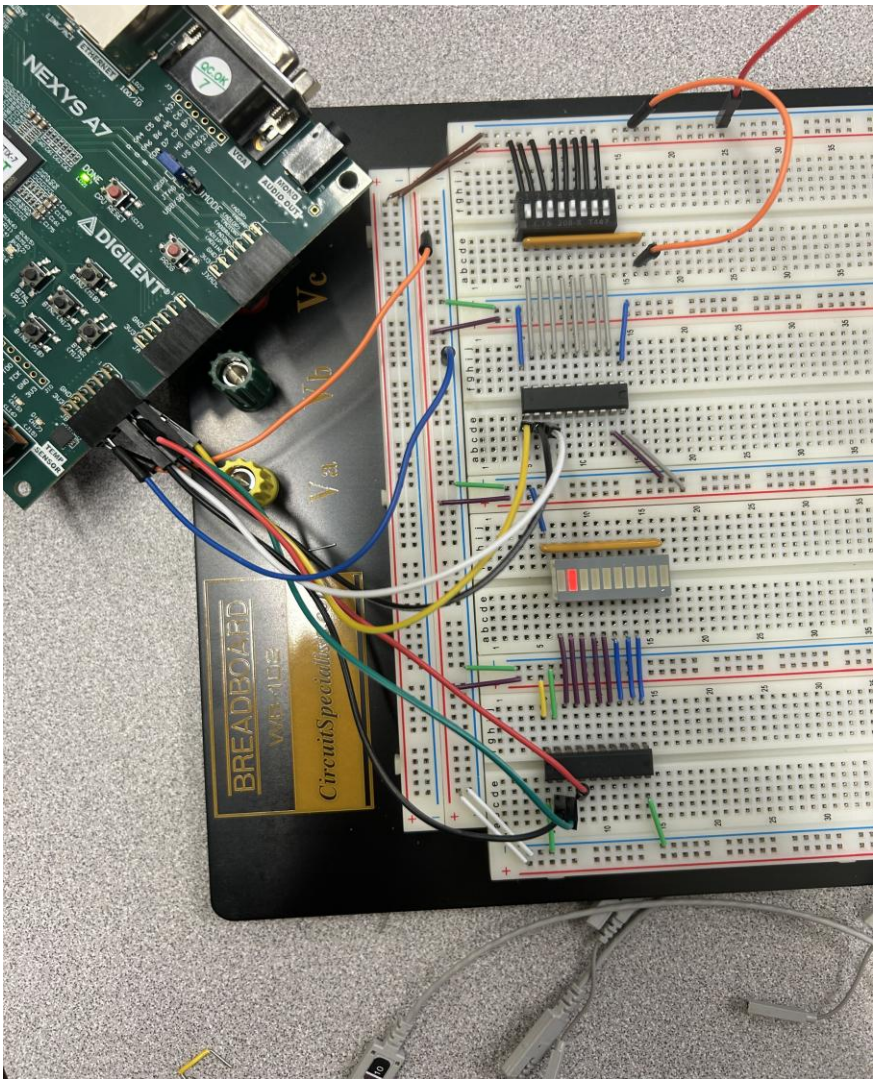

Fig 14: This shows when ABC=100 then f=1 fa=0 and fa2=0

Conclusion

In this lab, combinational circuits were designed, simulated, and implemented on the FPGA board to observe both timing delays and fault behavior. Task one demonstrated how multiplexers can be used both to introduce and measure signal delays, with the results compared between simulation and hardware. Task two modeled stuck-at and bridging faults, where test vectors were developed to clearly distinguish faulty circuits from correct ones. Overall, this lab provided valuable experience in VHDL coding, FPGA implementation, and verification techniques for fault detection and timing analysis.

Appendix

Task one

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 adder is
  Port (
    A,B,Cin: IN std_logic;
    C, Sel: IN std_logic_vector(3 downto 0);
    Sum,Cout: OUT std_logic;
    Y: OUT std_logic_vector(3 downto 0)
   );
end adder;

architecture Behavioral of adder is
    signal sigy: std_logic_vector (3 downto 0);
    signal sigCout: std_logic;
    component onebitadder
    port (
        A,B,Cin: IN std_logic;
        Sum, Cout: OUT std_logic
        );
        end component;
    component muxs
        port(
        Sel, yin, C: IN std_logic;
        Yout: OUT std_logic
        );
        end component;
begin
        Carry: onebitadder
            port map (A, B, Cin, Sum, sigCout);
        Cout <= sigCout;
        Y0: muxs
            port map (Sel(0), sigCout, C(0), sigy(0));
        Y(0) <= sigy(0);
        Y1: muxs
            port map (Sel(1), sigy(0), C(1), sigy(1));
        Y(1) <= sigy(1);
        Y2: muxs
            port map (Sel(2), sigy(1), C(2), sigy(2));
        Y(2) <= sigy(2);
        Y3: muxs
            port map (Sel(3), sigy(2), C(3), sigy(3));
        Y(3) <= sigy(3);
end Behavioral;


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 onebitadder is
  Port (
  A,B,Cin: IN std_logic;
  Sum,Cout: OUT std_logic
  );
end onebitadder;

architecture Behavioral of onebitadder is

begin
    Sum <= (A XOR B) XOR Cin;
    Cout <= (A AND B) OR (A AND Cin) OR (B AND Cin);

end Behavioral;



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 muxs is
  Port (
  Sel, Yin, C: in std_logic;
  Yout: OUT std_logic
  );
end muxs;

architecture Behavioral of muxs is

begin
    PR_MUX: process (Sel, Yin, C)
        begin
            case Sel is
                when '0' => Yout <= Yin;
                when '1' => Yout <= C;
                when others => Yout <= Yin;
            end case;
        end process PR_MUX;

end Behavioral;
```

TCL

```
restart

add_force Sel {0000 0ns}
add_force A {0 0ns}
add_force B {0 0ns}
add_force Cin {0 0ns}
add_force C {0000 0ns}
run 20ns
add_force Sel {0101 0ns}
add_force A {1 0ns}
add_force B {1 0ns}
add_force Cin {0 0ns}
```

```
add_force C {0000 0ns}
run 20ns
add_force Sel {1111 0ns}
add_force A {0 0ns}
add_force B {1 0ns}
add_force Cin {1 0ns}
add_force C {1100 0ns}
run 20ns
add_force Sel {0111 0ns}
add_force A {1 0ns}
add_force B {0 0ns}
add_force Cin {0 0ns}
add_force C {0011 0ns}
run 20ns
add_force Sel {1000 0ns}
add_force A {1 0ns}
add_force B {0 0ns}
add_force Cin {0 0ns}
add_force C {0011 0ns}
run 20ns
add_force Sel {1111 0ns}
add_force A {1 0ns}
add_force B {1 0ns}
add_force Cin {1 0ns}
add_force C {0111 0ns}
run 20ns
```

XDC

```
##Switches
set_property -dict { PACKAGE_PIN U11 IOSTANDARD LVCMOS33 } [get_ports { A }]; #IO_L19N_T3_A09_D25_VREF_14
Sch=sw[14]
set_property -dict { PACKAGE_PIN V10 IOSTANDARD LVCMOS33 } [get_ports { B }]; #IO_L21P_T3_DQS_14 Sch=sw[15]

##Pmod Header JA
set_property -dict { PACKAGE_PIN D17 IOSTANDARD LVCMOS33 } [get_ports { Cin }]; #IO_L16N_T2_A27_15
Sch=ja[7]

##Pmod Header JB
set_property -dict { PACKAGE_PIN D14 IOSTANDARD LVCMOS33 } [get_ports { C[0] }]; #IO_L1P_T0_AD0P_15
Sch=jb[1]
set_property -dict { PACKAGE_PIN F16 IOSTANDARD LVCMOS33 } [get_ports { Sel[0] }]; #IO_L14N_T2_SRCC_15
Sch=jb[2]
set_property -dict { PACKAGE_PIN G16 IOSTANDARD LVCMOS33 } [get_ports { C[1] }]; #IO_L13N_T2_MRCC_15
Sch=jb[3]
set_property -dict { PACKAGE_PIN H14 IOSTANDARD LVCMOS33 } [get_ports { Sel[1] }]; #IO_L15P_T2_DQS_15
Sch=jb[4]
set_property -dict { PACKAGE_PIN E16 IOSTANDARD LVCMOS33 } [get_ports { C[2] }]; #IO_L11N_T1_SRCC_15
Sch=jb[7]
set_property -dict { PACKAGE_PIN F13 IOSTANDARD LVCMOS33 } [get_ports { Sel[2] }]; #IO_L5P_T0_AD9P_15
Sch=jb[8]
set_property -dict { PACKAGE_PIN G13 IOSTANDARD LVCMOS33 } [get_ports { C[3] }]; #IO_0_15 Sch=jb[9]
set_property -dict { PACKAGE_PIN H16 IOSTANDARD LVCMOS33 } [get_ports { Sel[3] }]; #IO_L13P_T2_MRCC_15
Sch=jb[10]

##Pmod Header JC
set_property -dict { PACKAGE_PIN K1 IOSTANDARD LVCMOS33 } [get_ports { Sum }]; #IO_L23N_T3_35 Sch=jc[1]
set_property -dict { PACKAGE_PIN F6 IOSTANDARD LVCMOS33 } [get_ports { Cout }]; #IO_L19N_T3_VREF_35
Sch=jc[2]
set_property -dict { PACKAGE_PIN J2 IOSTANDARD LVCMOS33 } [get_ports { Y[0] }]; #IO_L22N_T3_35 Sch=jc[3]
set_property -dict { PACKAGE_PIN G6 IOSTANDARD LVCMOS33 } [get_ports { Y[1] }]; #IO_L19P_T3_35 Sch=jc[4]
set_property -dict { PACKAGE_PIN E7 IOSTANDARD LVCMOS33 } [get_ports { Y[2] }]; #IO_L6P_T0_35 Sch=jc[7]
set_property -dict { PACKAGE_PIN J3 IOSTANDARD LVCMOS33 } [get_ports { Y[3] }]; #IO_L22P_T3_35 Sch=jc[8]
```

Task Two

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 lab3task2 is

Port (

A, B, C : in std_logic;

F, fa, fa2: out std_logic

);

end lab3task2;

architecture Behavioral of lab3task2 is

begin

F <= (A AND C) Or (A AND B) OR ( A XOR C);

fa<= B OR C;

fa2 <= B;

end Behavioral;
```

TCL

```
restart

add_force A {0 0ns}
add_force B {0 0ns}
add_force C {0 0ns}
run 20ns

add_force A {0 0ns}
add_force B {0 0ns}
add_force C {1 0ns}
run 20ns

add_force A {0 0ns}
add_force B {1 0ns}
add_force C {0 0ns}
run 20ns

add_force A {0 0ns}
add_force B {1 0ns}
add_force C {1 0ns}
run 20ns

add_force A {1 0ns}
add_force B {0 0ns}
add_force C {0 0ns}
run 20ns
```

```
add_force A {1 0ns}
add_force B {0 0ns}
add_force C {1 0ns}
run 20ns

add_force A {1 0ns}
add_force B {1 0ns}
add_force C {0 0ns}
run 20ns

add_force A {1 0ns}
add_force B {1 0ns}
add_force C {1 0ns}
run 20ns
```

XDC

```
set_property -dict { PACKAGE_PIN E16 IOSTANDARD LVCMOS33 } [get_ports { A }];
#IO_L11N_T1_SRCC_15 Sch=jb[7]

set_property -dict { PACKAGE_PIN F13 IOSTANDARD LVCMOS33 } [get_ports { B }];
#IO_L5P_T0_AD9P_15 Sch=jb[8]

set_property -dict { PACKAGE_PIN G13 IOSTANDARD LVCMOS33 } [get_ports { C }]; #IO_0_15
Sch=jb[9]


set_property -dict { PACKAGE_PIN D14 IOSTANDARD LVCMOS33 } [get_ports { F }];
#IO_L1P_T0_AD0P_15 Sch=jb[1]

set_property -dict { PACKAGE_PIN F16 IOSTANDARD LVCMOS33 } [get_ports { fa2 }];
#IO_L14N_T2_SRCC_15 Sch=jb[2]

set_property -dict { PACKAGE_PIN G16 IOSTANDARD LVCMOS33 } [get_ports { fa }];
#IO_L13N_T2_MRCC_15 Sch=jb[3]
```