

# Lab-8 Report

VHDL Implementation Code for given logic

MIRIYALA PRANAY KAMAL  
200030033

## Aim of the experiment:

Write clear logic and implement the following codes as per the instructions given.

## Components used:

1. CPLD board
2. Cable to interface it with the computer

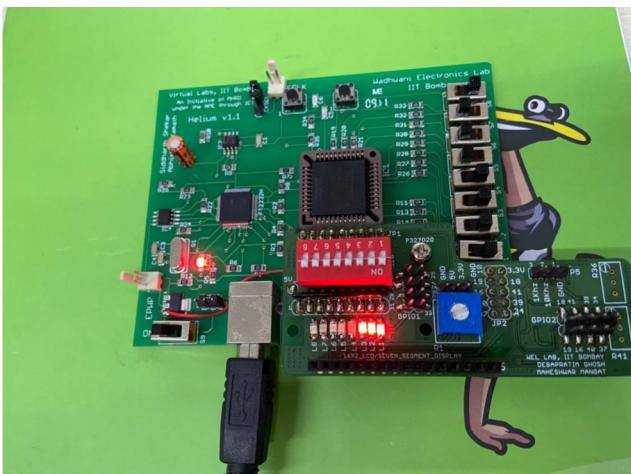
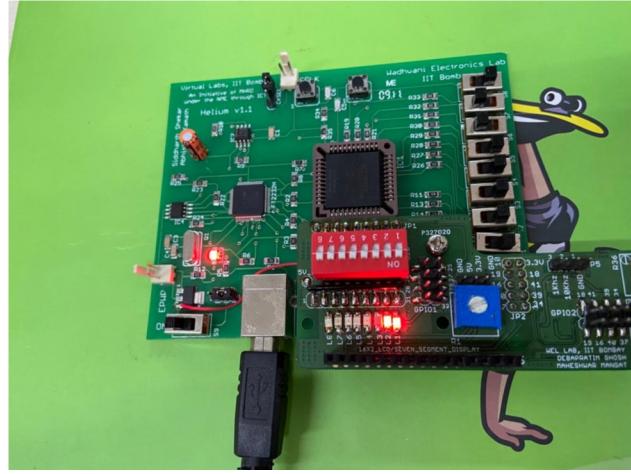
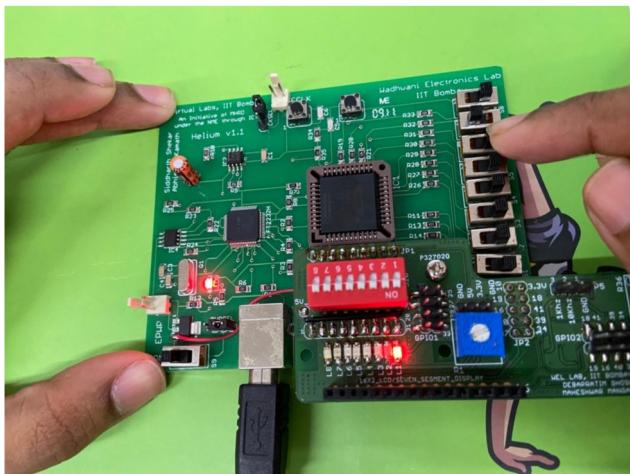
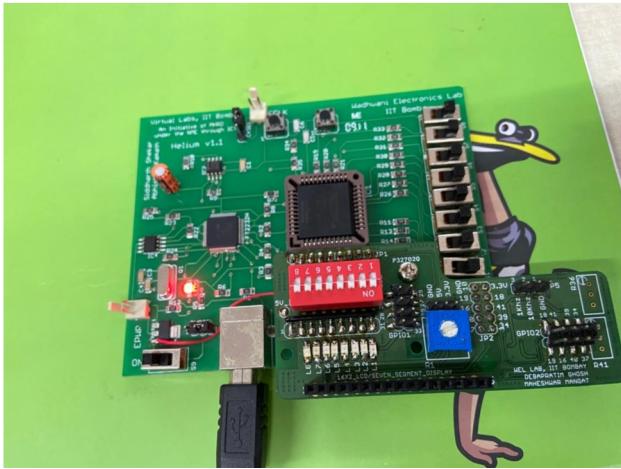
- I. Write VHDL code for 3-bit adder/subtractor in structural modelling style. The adder/subtractor operation is controlled by signal 'm'.

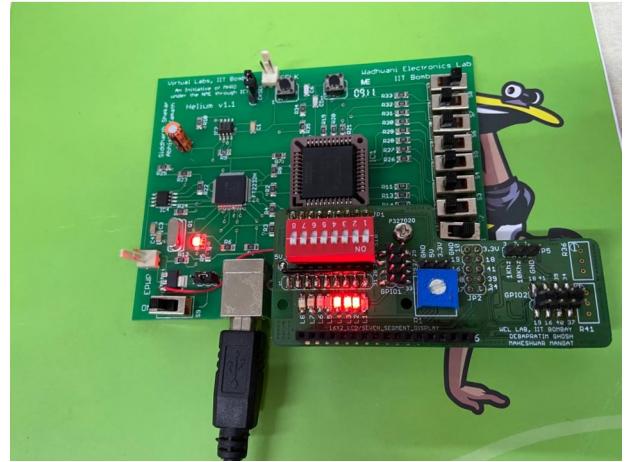
VHDL Code for 3-bit Full-Adder and Subtractor circuit with Control signal

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
--define the entity with ports
entity 3bit is port(
a : in STD_LOGIC_VECTOR(2 downto 0); b : in STD_LOGIC_VECTOR(2 downto 0);
sel : in STD_LOGIC;
sum : out STD_LOGIC_VECTOR(2 downto 0)
);
end 3bit;
architecture rtl of 3bit is
Component fa is
port (a : in STD_LOGIC;
b : in STD_LOGIC;
c : in STD_LOGIC; sum : out STD_LOGIC; carry : out STD_LOGIC );
end component;

signal s : std_logic_vector (2 downto 0);
signal l : std_logic_vector (2 downto 0);
begin
l <= b xor (sel & sel & sel);
u0 : fa port map (a(0),l(0),sel,sum(0),s(0)); u1 : fa port map
(a(1),l(1),s(0),sum(1),s(1)); ue : fa port map (a(2),l(2),s(1),sum(2),open);
FINAL CODE:
end rtl;
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity fa is
port (a : in STD_LOGIC;
b : in STD_LOGIC;
c : in STD_LOGIC; sum : out STD_LOGIC; carry : out STD_LOGIC );
end fa;
architecture fa_arc of fa is begin
sum <= a xor b xor c;
carry <= (a and b) or (b and c) or (c and a);
end fa_arc;
```

## Snap shots of the circuit board:





II. Write VHDL code for ALU which performs following operation depending on selection lines.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use ieee.NUMERIC_STD.all;
entity alu is Port (
    A, B : in STD_LOGIC_VECTOR(2 downto 0);
    ALU_Sel : in STD_LOGIC_VECTOR(1 downto 0);
    ALU_Out : out STD_LOGIC_VECTOR(2 downto 0);
    Carryout : out std_logic
);
end alu;

architecture Behavioral of alu is

signal ALU_Result : std_logic_vector (2 downto 0);
signal tmp: std_logic_vector (3 downto 0);

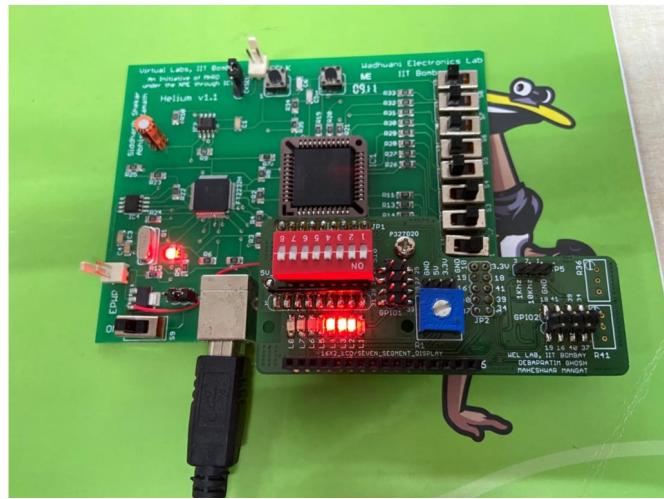
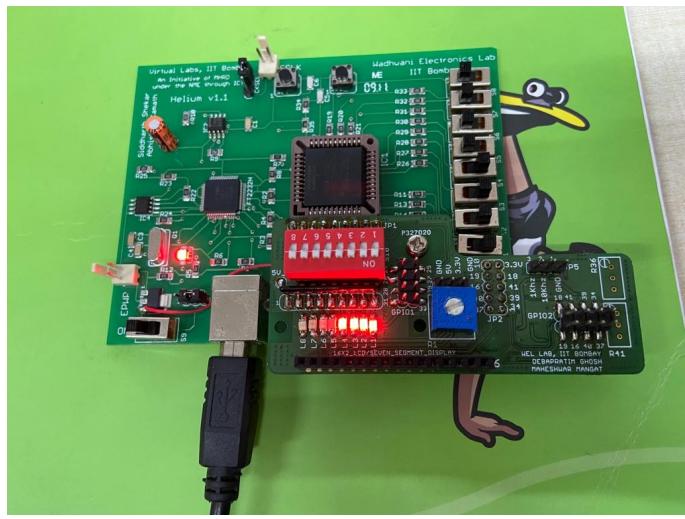
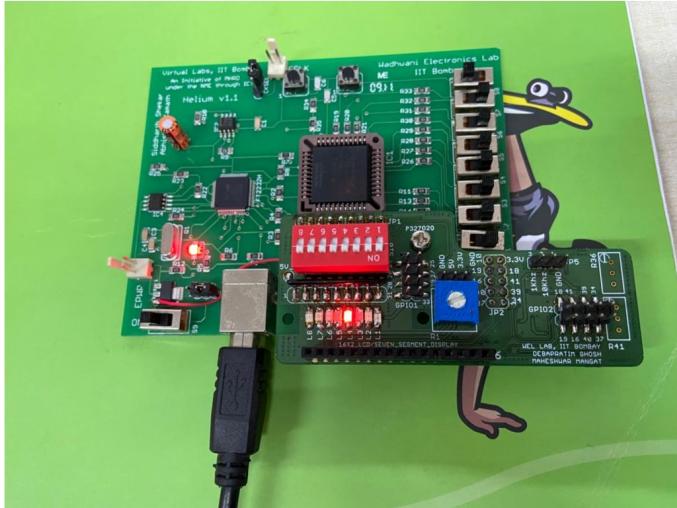
begin process(A,B,ALU_Sel)

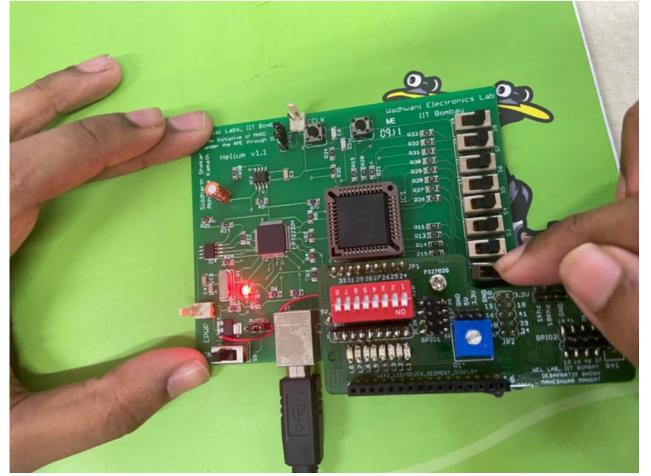
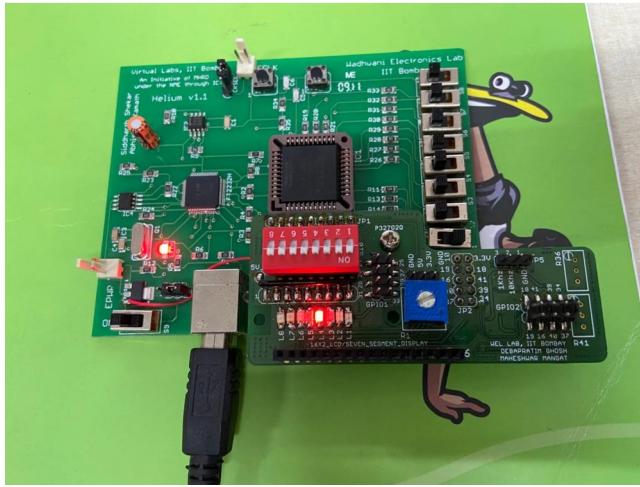
begin
    case(ALU_Sel) is
        when "00" =>
            ALU_Result <= A + B ;
        when "01" =>
            ALU_Result <= A - B ;
        when "10" =>
            ALU_Result <= A and B;
        when "11" =>
            ALU_Result <= A xor B;
    end case;
end process;

ALU_Out <= ALU_Result;
tmp <= ('0' & A) + ('0' & B);
```

```
Carryout <= tmp(3);  
end Behavioral;
```

## Snap shots of the circuit board:





III. Write VHDL code for 4:2 priority encoder with active high enable pin.

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

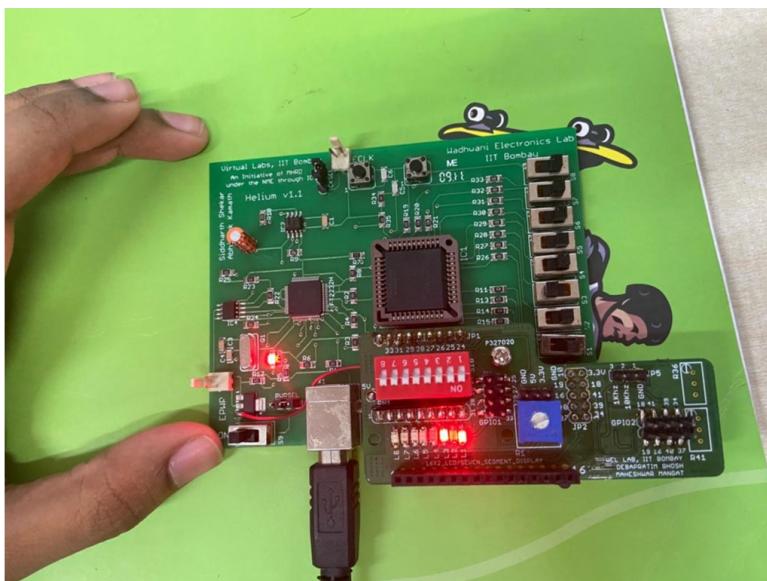
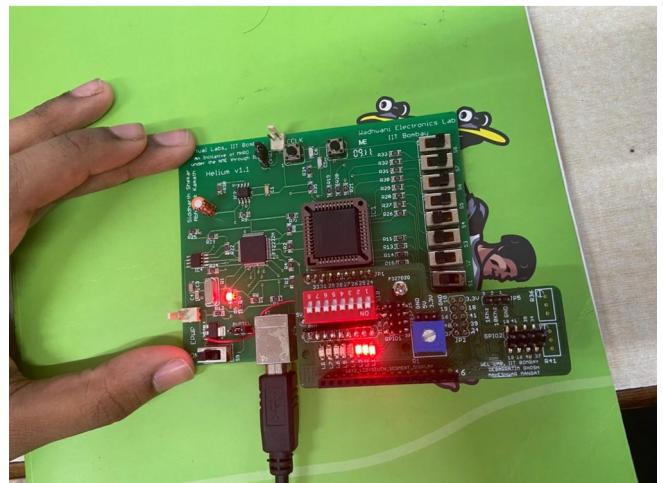
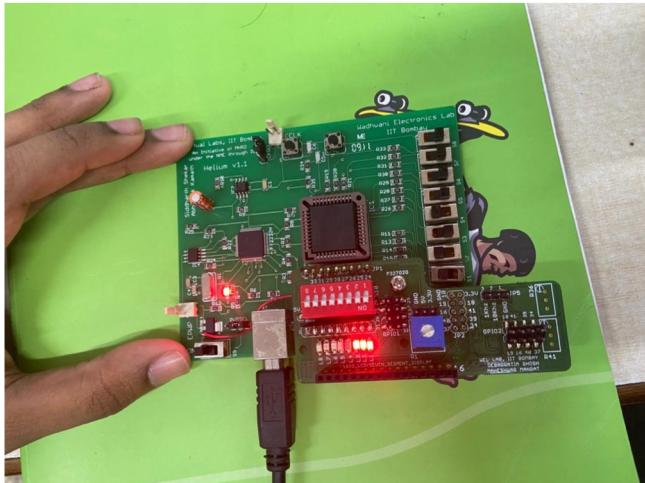
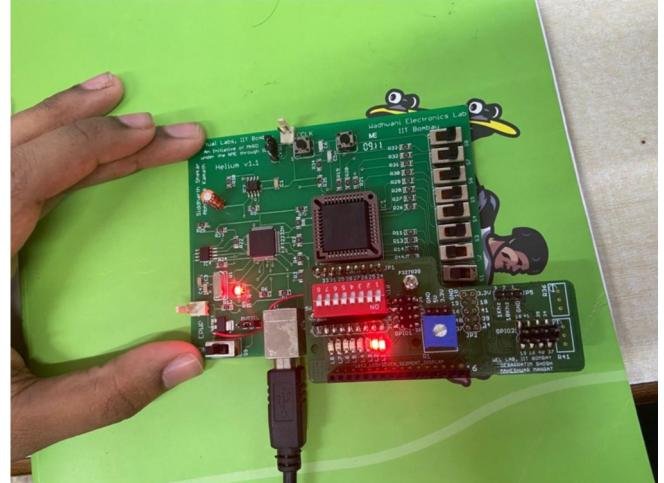
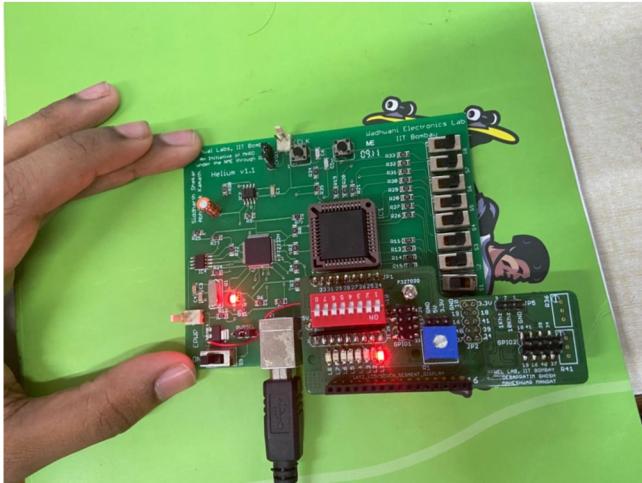
entity PE is
    Port ( INPUT : in STD_LOGIC_VECTOR (3 downto 0);
           OUTPUT : out STD_LOGIC_VECTOR (2 downto 0));
end PE;

architecture Behavioral of PE is

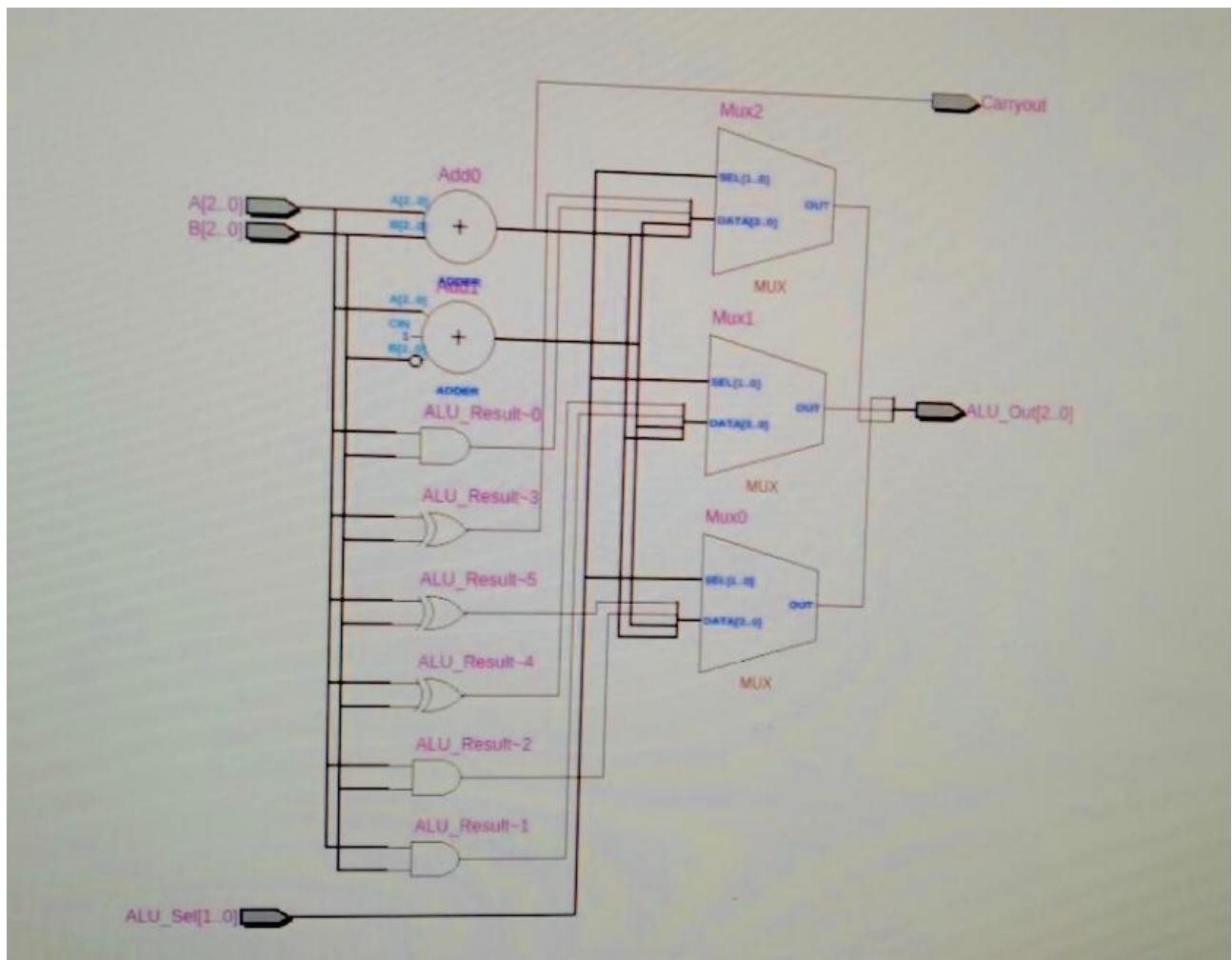
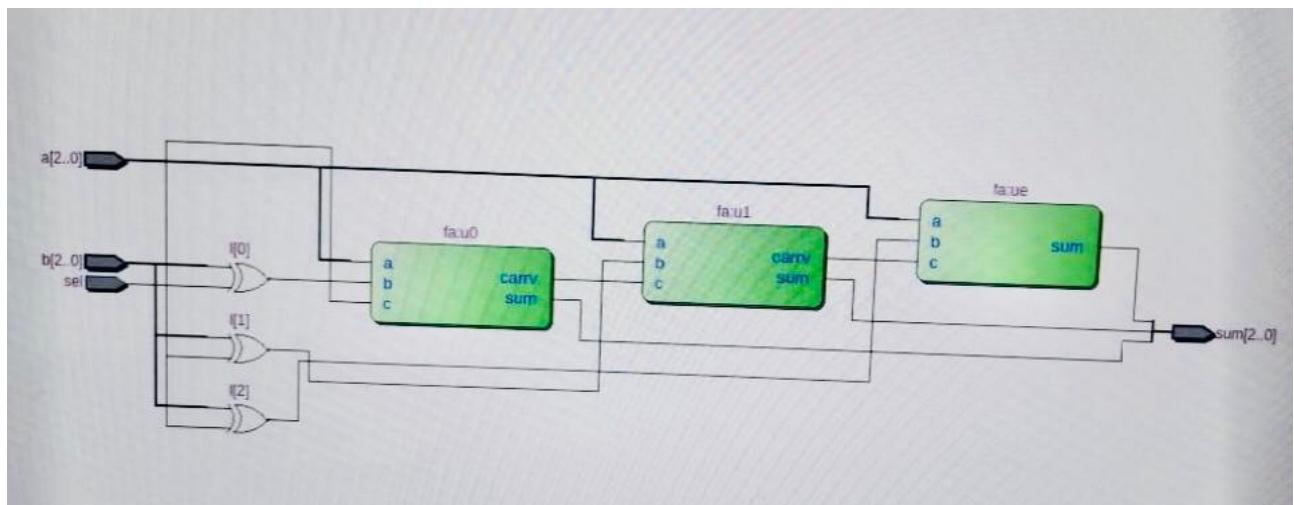
begin
    process(INPUT)
    begin
        if (INPUT(3)='1') then
            OUTPUT <= "111";
        elsif (INPUT(2)='1') then
            OUTPUT <= "101";
        elsif (INPUT(1)='1') then
            OUTPUT <= "011";
        elsif (INPUT(0)='1') then
            OUTPUT <= "001";
        else
            OUTPUT <= "000";
        end if;
    end process;

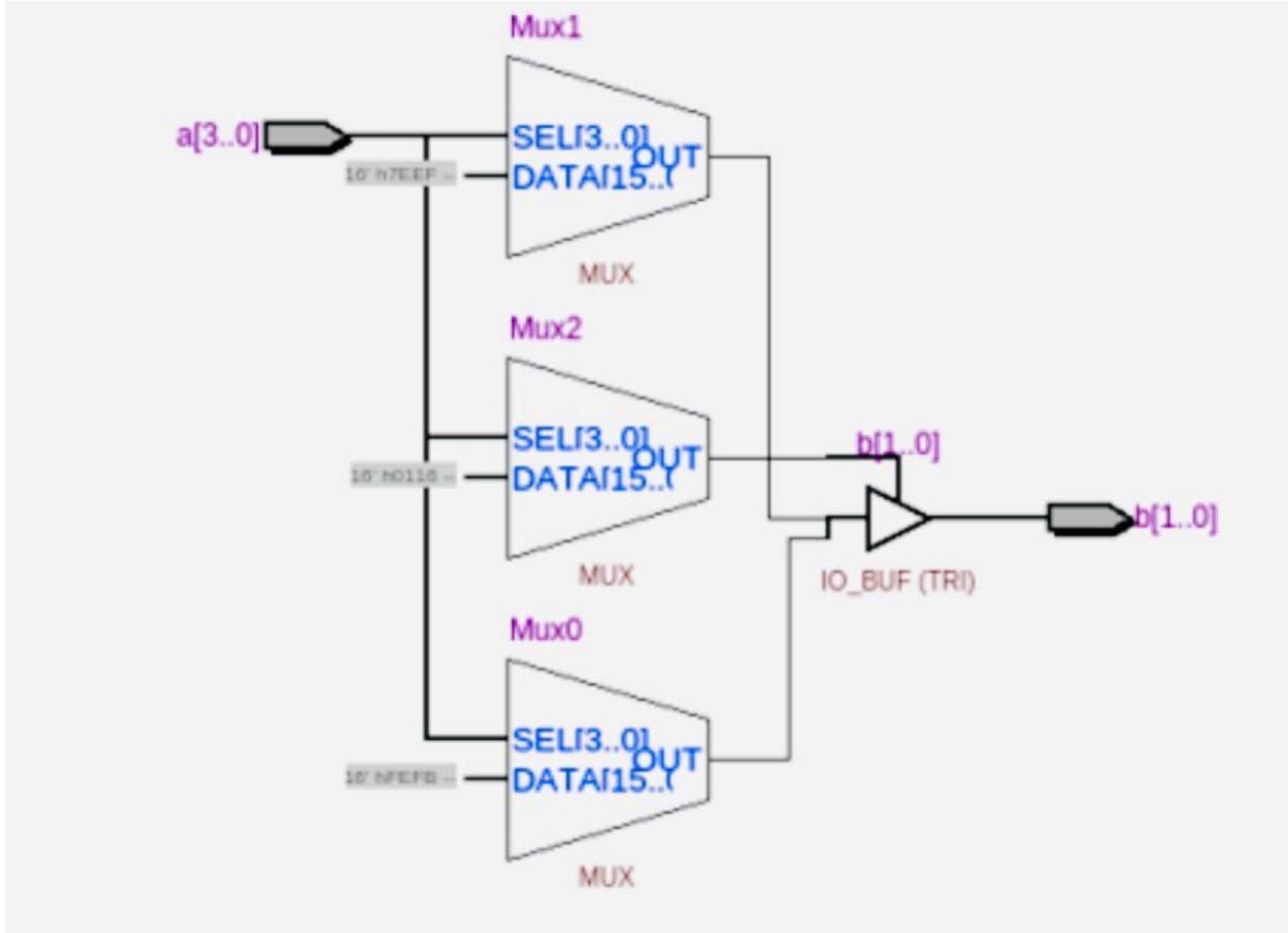
end Behavioral;
```

## Snap shots of the circuit board:



### RTL netlist viewer snapshots:





### Results and Conclusions:

We successfully verified the functionality of 3-bit adder/subtractor, ALU & 4:2 priority encoder circuits by dumping the corresponding VHDL code onto the CPLD board and trying out different combinations of inputs via switches, observing their outputs via LED's on the board.

We also confirmed our code using the RTL net list viewer, by viewing and confirming the logic gates diagram.

Also we dumped the code on to the CPLD board using USB cable and jtag shell in linux terminal and confirmed our observations, whose snapshots are attached in the report.