Lab 7

EE214: Digital Circuits Laboratory

B. Siddharth Prabhu
Roll No. 200010003

11 February 2022

1 Aim

Study of syntax, utility, and execution of VHDL (Very High Speed Integrated Circuit Hardware Description Language), and implementation of half adder using it.

2 Summary of the experiment

Familiarization with VHDL syntax and working using Altera Quartus Simulator, then design and implementation of half adder using different modelling styles, viz., dataflow, behavioral, and structural modelling styles.

3 Components Used

Helium v1.1 (MAX 3000 architecture), Altera Quartus Simulator Software, USB Type-B cable, conducting wires.

4 Design Procedure: Truth Table

X	Y	C (Carry)	S (Sum)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Table 1: Truth Table for a Half Adder

5 VHDL Code

5.1 Half Adder: Dataflow Modelling Style

```
library ieee;
use ieee.std_logic_1164.all;

entity half_adder_dat is
port(
    A, B: in std_logic;
    S, C: out std_logic);
end half_adder_dat;

architecture dataflow of half_adder_dat is
begin
    S <= A xor B;
    C <= A and B;
end dataflow;
```

5.2 Half Adder: Behavioral Modelling Style

```
library ieee;
use ieee.std_logic_1164.all;

entity half_adder_beh is
port(
    A, B: in std_logic;
    S, C: out std_logic);
end half_adder_beh;

architecture behaviour of half_adder_beh is
begin
    c1: process (A,B)
    begin
        if A = '1' then
            S <= not B;
            C <= B;
        else
            S <= B;
            C <= '0';
        end if;
    end process c1;
end behaviour;
```

5.3 Half Adder: Structural Modelling Style

```
-----------------------------------------------
-------------- CODE FOR XOR GATE --------------
library ieee;
use ieee.std_logic_1164.all;

entity xor_gate is
port(
    i1, i2: in std_logic;
    o1: out std_logic);
end xor_gate;

architecture dataflow of xor_gate is
begin
    o1 <= i1 xor i2;
end dataflow;

-----------------------------------------------
-------------- CODE FOR AND GATE --------------
library ieee;
use ieee.std_logic_1164.all;

entity and_gate is
port(
    i3, i4: in std_logic;
    o2: out std_logic);
end and_gate;

architecture dataflow of and_gate is
begin
    o2 <= i3 and i4;
end dataflow;

-----------------------------------------------
------------- CODE FOR HALF ADDER -------------
library ieee;
use ieee.std_logic_1164.all;

entity half_adder_str is
port(
    A, B: in std_logic;
    S, C: out std_logic);
end half_adder_str;
```

```vhdl
architecture structure of half_adder_str is
    component xor_gate is
    port(
        i1, i2: in std_logic;
        o1: out std_logic
    );
    end component;
    component and_gate is
    port(
        i3, i4: in std_logic;
        o2: out std_logic
    );
    end component;
begin
    u1: xor_gate port map (i1 => A, i2 => B, o1 => S);
    u2: and_gate port map (i3 => a, i4 => b, o2 => C);
    -- map the ports of the components to the ports of the overall half adder
end structure;
```

6 Circuit and Simulation Snapshots

(RTL and Circuit snapshots are included in the later pages of this report.)

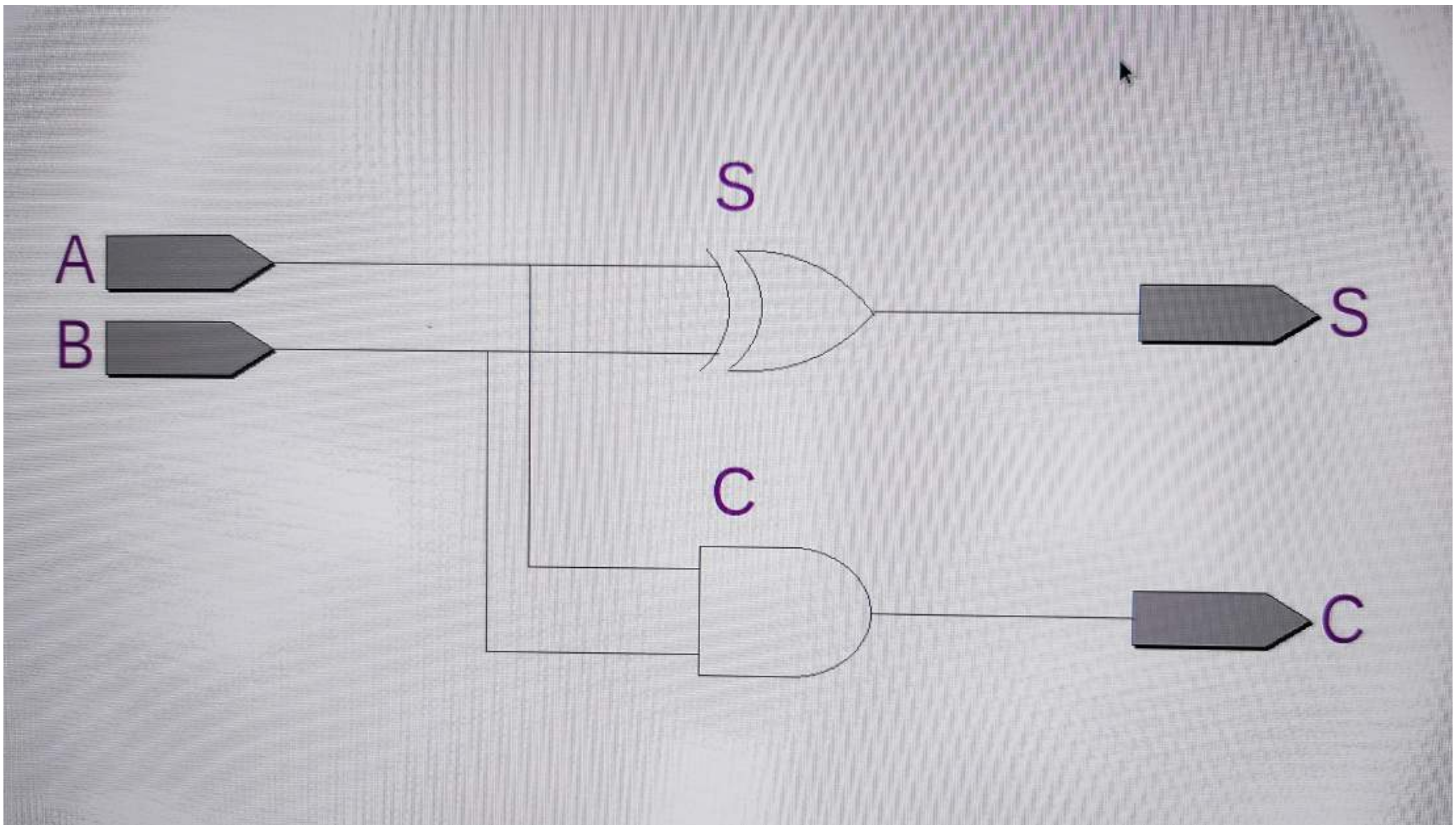
7 Results and Discussions

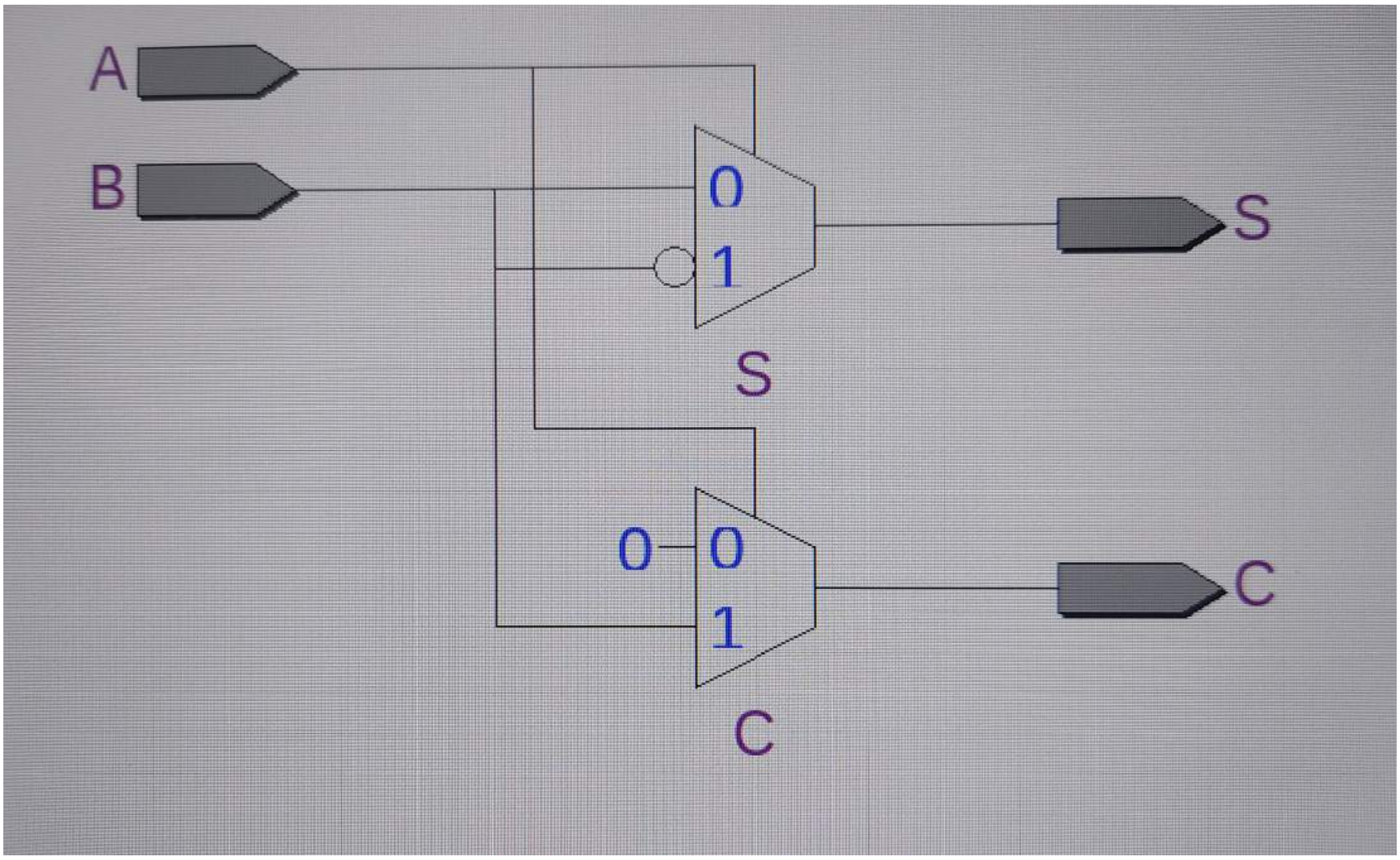
- We studied the syntax, working and utility of VHDL code. We executed the code in the Altera Quartus Software, and used the Helium v1.1 board as a CPLD (Complex Programmable Logic Device).
- The outputs obtained could be verified via truth table of a half adder in our design. It has two inputs (X and Y), and two outputs (Sum and Carry).
- CPLD greatly reduces time and effort required in designing and simulating logic circuits, despite the steep initial learning curve.

8 Conclusion

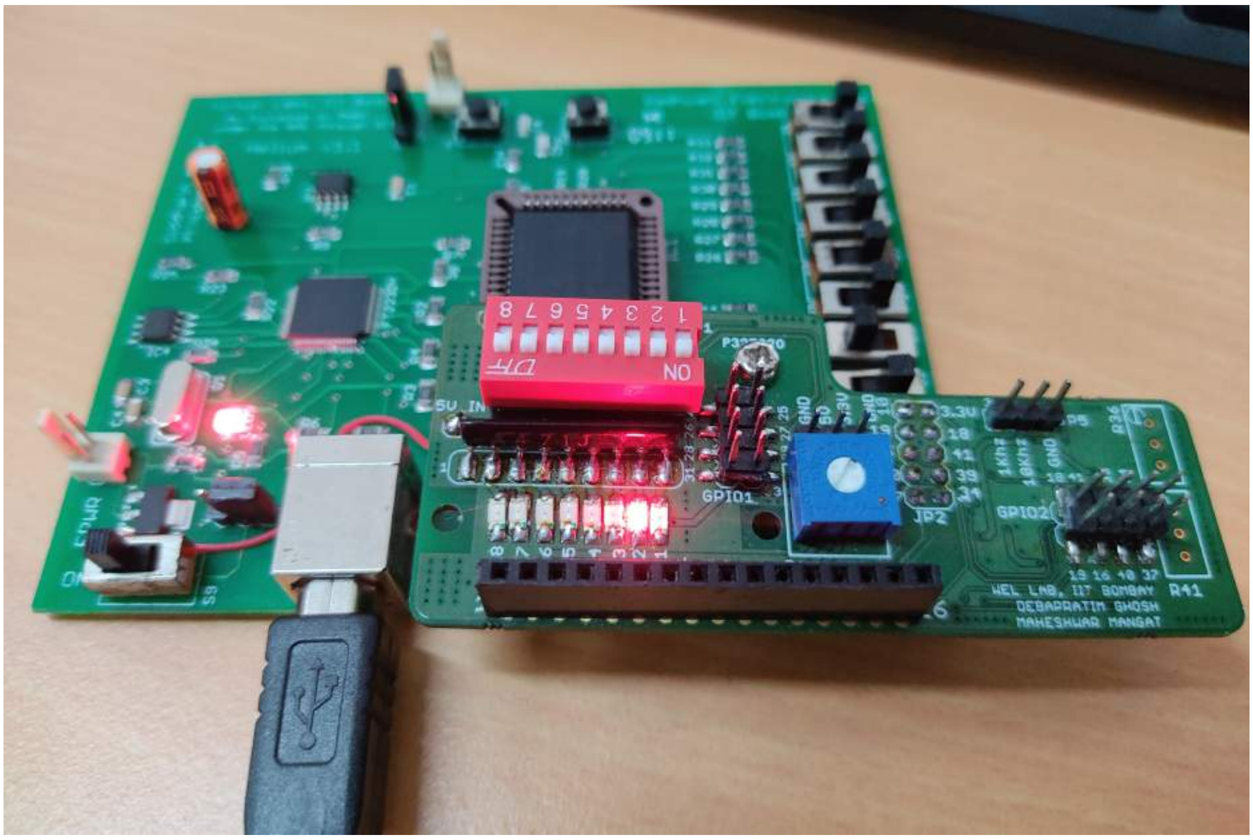
We learnt the working and execution of VHDL code, and its utility in simulating circuits in a simplified manner, and observed this during implementation of half adder circuit.

RTL Images :

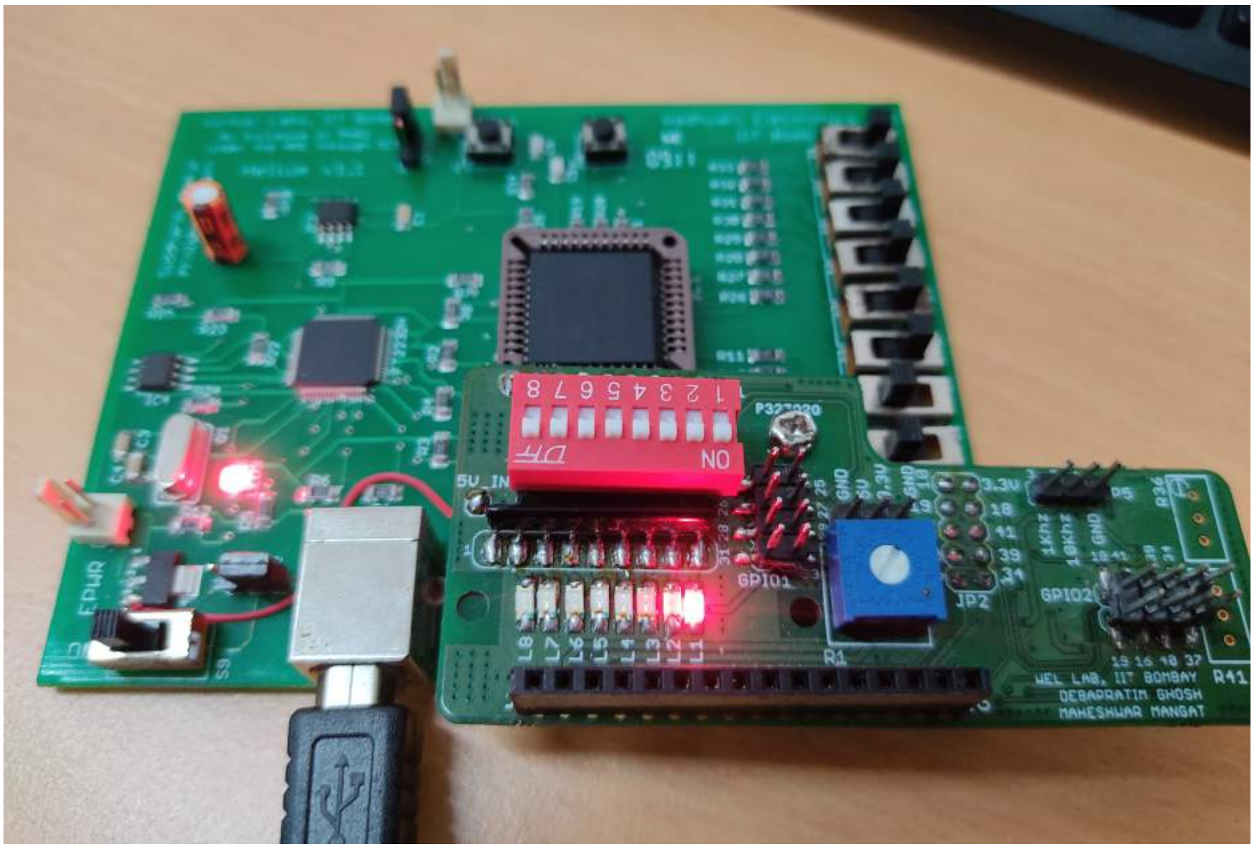




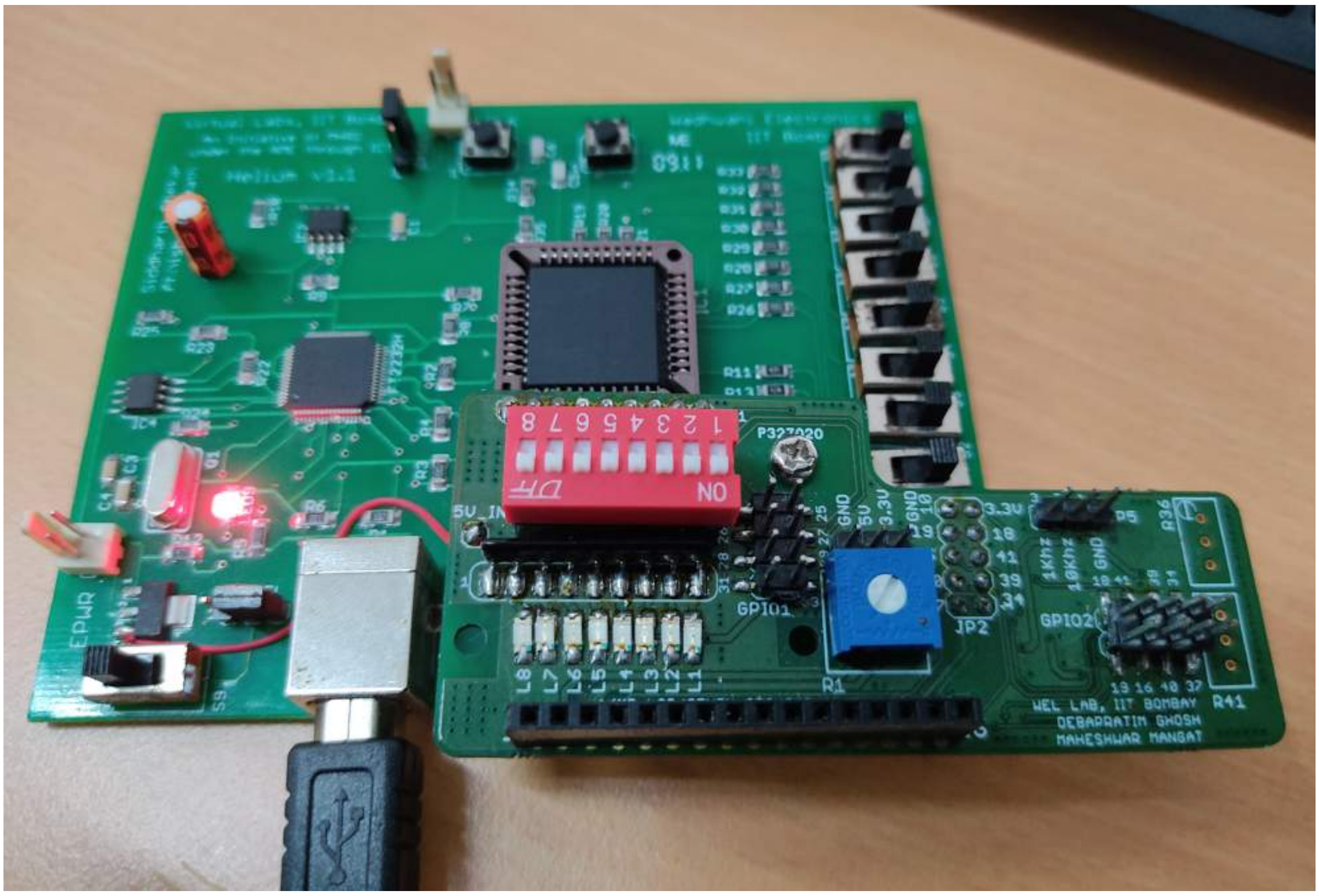
Circuit and Simulation Snapshots :



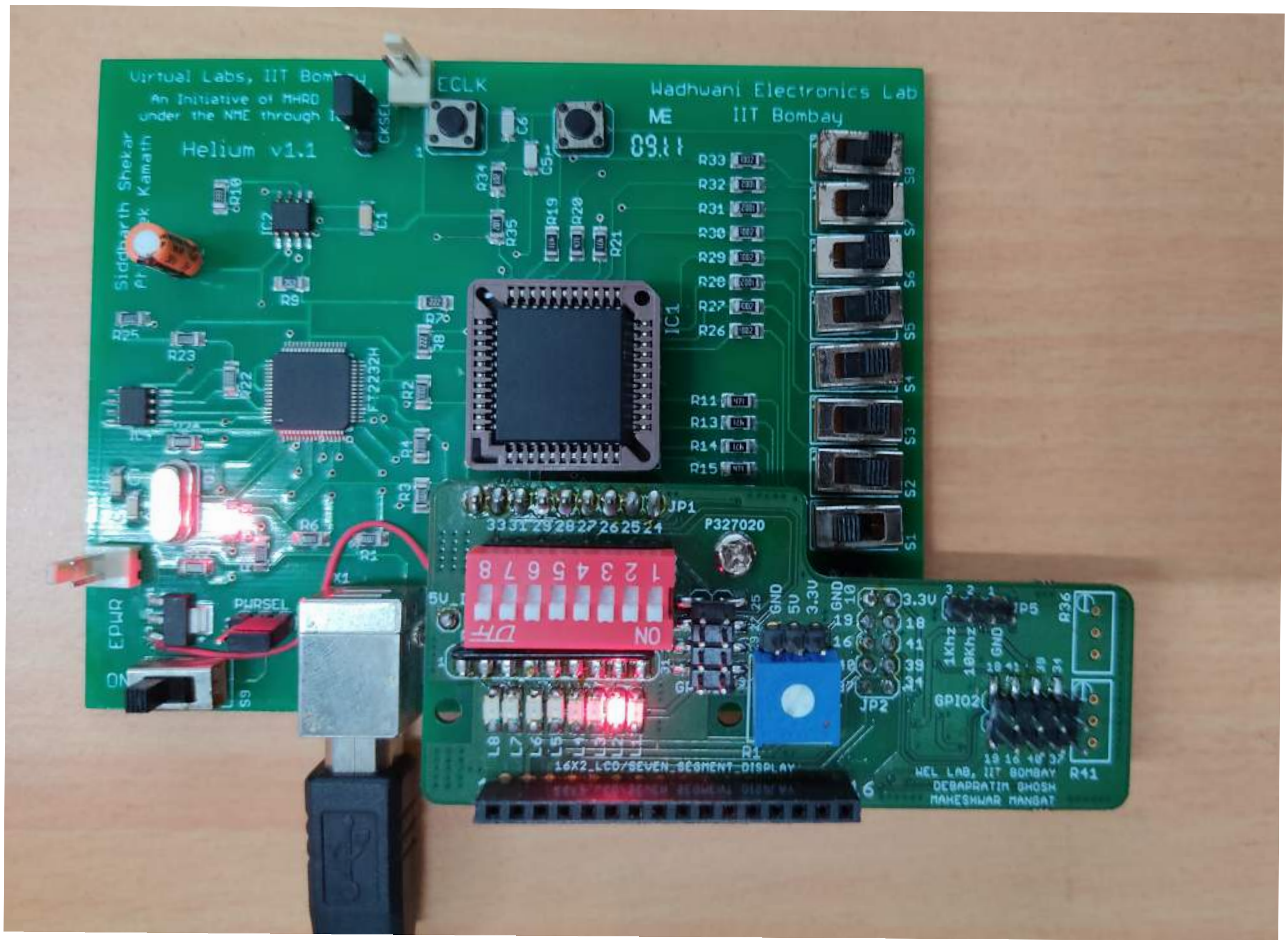
0 + 1 —> S = 1, C = 0



1 + 1 —> S = 0, C = 1



0 + 0 —> S = 0, C = 0



1 + 0 —> S = 1, C = 0