Lab Report 4

Ashish Prasad, 21D180009

August 29, 2022

1 Overview of the experiment

The purpose of this experiment was to make a BCD subtractor using 4-bit adder/subtractor and some gates from the Gates Library.

2 Experiment setup

BCD or Binary Coded Decimal is that number system or code which has binary numbers or digits to represent a decimal number. A decimal number contains 10 digits (0-9). Now the equivalent binary numbers can be found out of these 10 decimal numbers. In the case of BCD the binary number formed by four binary digits will be the equivalent code for the given decimal digits. In BCD we can use the binary number from 0000-1001 only, which are the decimal equivalent from 0-9 respectively.

Decimal	BCD	Binary
0	0000	0000
1	0000	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	0001 0000	1010

Table 1: BCD Representation

2.1 Digital Design for BCD Subtractor

The Pen-paper design for the subtractor(BCD) is shown as follows:

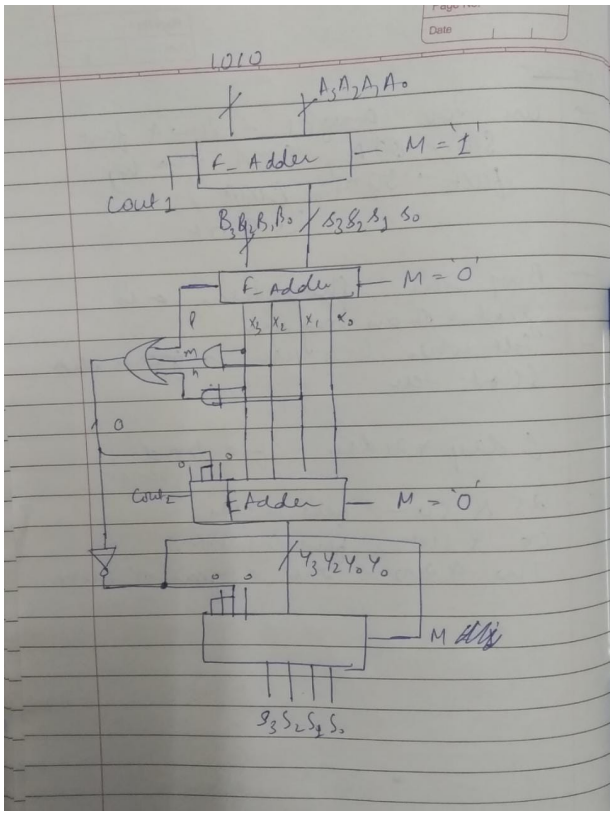

Figure 1: Design

The code for the same BCD Subtractor design is shown :

2.1.1 Entity Declaration

```
entity subtractor is
port(A3,A2,A1,A0,B3,B2,B1,B0: in std_logic;
     sign,S3,S2,S1,S0: out std_logic);
end entity subtractor;
```

2.1.2 Architecture Declaration

```
architecture pqrs of subtractor is
--component
component not_2 is
  port( x: in std_logic; y: out std_logic);
end component not_2;
component adder_subtractor is
port( a3,a2,a1,a0,b3,b2,b1,b0,m: in std_logic;
      s3,s2,s1,s0,cout: out std_logic);
end component adder_subtractor;
component and_2 is
port( a1, b1: in std_logic; c1: out std_logic);
end component and_2;
----------------
```

```
component or_2 is
port( a2, b2: in std_logic; c2: out std_logic);
end component or_2;
----------------
signal ss3,ss2,ss1,ss0,l,m,n,o,ayush,ashish,x3,x2,x1,x0,y3,y2,y1,y0,cout1,cout2,s_sign:

begin
limmk1: adder_subtractor port map ('1','0','1','0',B3,B2,B1,B0,'1',ss3,ss2,ss1,ss0,cout
limmk2: adder_subtractor port map (A3,A2,A1,A0,ss3,ss2,ss1,ss0,'0',x3,x2,x1,x0,l);
limmk3: and_2 port map (x3,x2,m);
limmk4: and_2 port map (x3,x1,n);
limmk5: or_2 port map (l,m,ayush);
limmk6: or_2 port map (ayush,n,o);
limmk7: adder_subtractor port map ('0',o,o,'0',x3,x2,x1,x0,'0',y3,y2,y1,y0,cout2);
limmk8: not_2 port map (o,s_sign);
limmk9: adder_subtractor port map (s_sign,'0',s_sign,'0',y3,y2,y1,y0,s_sign,S3,S2,S1,S0
limmk10: sign<=s_sign;
end architecture pqrs;
```

3 Observations

3.1 Digital Design

The Design passed all the test cases as given in TRACEFILE the RTL Simulation for the same is shown below.

3.2 RTL Simulation

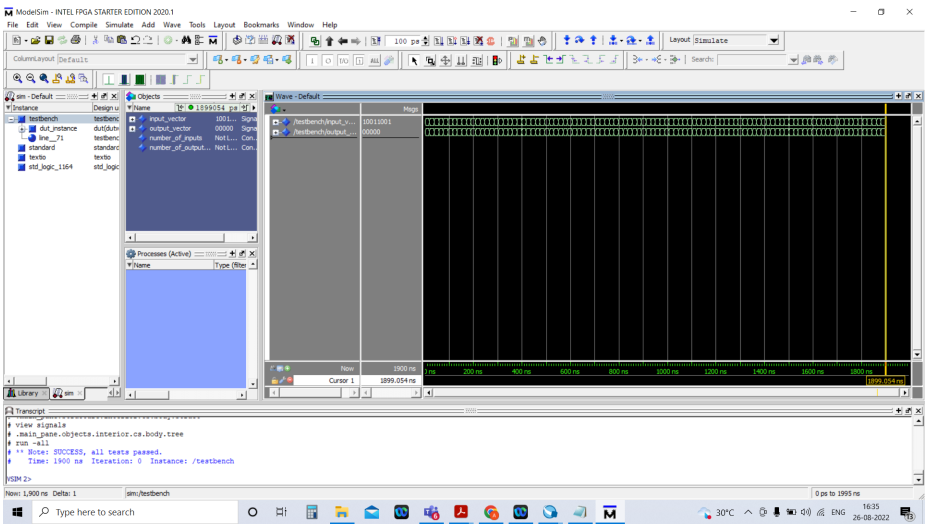

Figure 2: RTL Simulation

3.3 Xenon Board Simulation

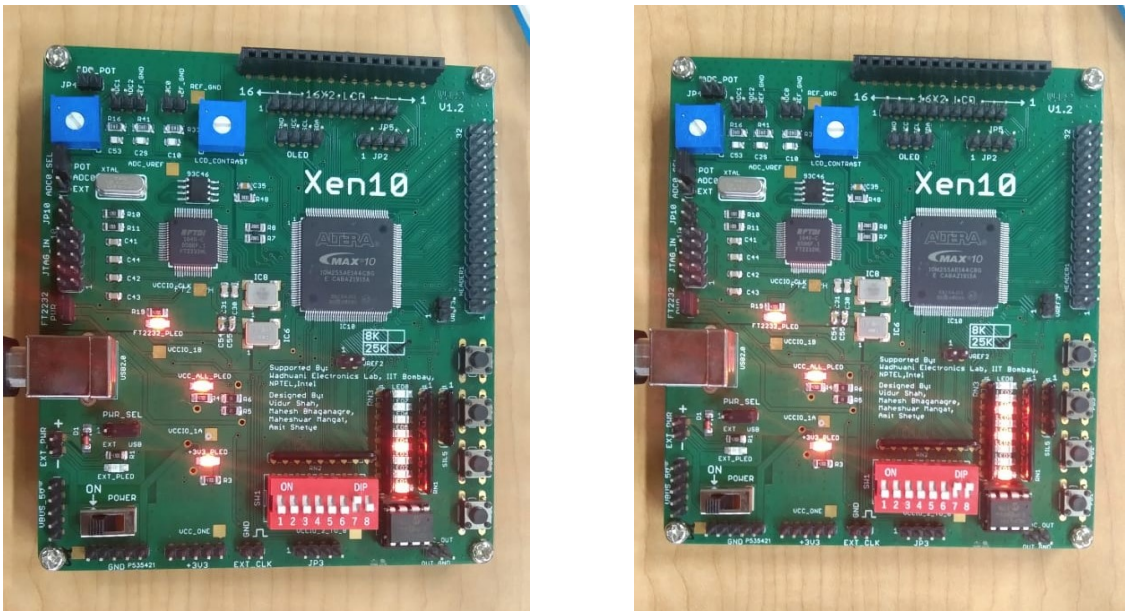

Figure 3: Xenon board simulation