

Lab-7 Report

VHDL Implementation Code for half adder & full adder

MIRIYALA PRANAY KAMAL
200030033

Aim of the experiment:

Study of CPLD board and simulating Full-adder and Half-adder circuit VHDL codes by dumping them on board.

Components used:

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

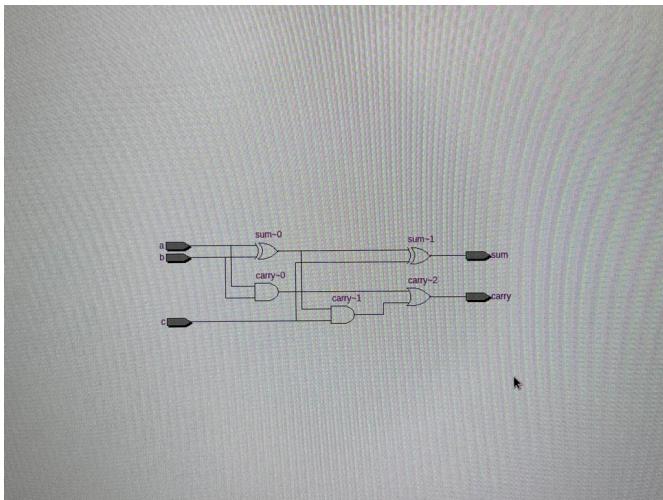
Truth Table and boolean expression for Half-Adder:

Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

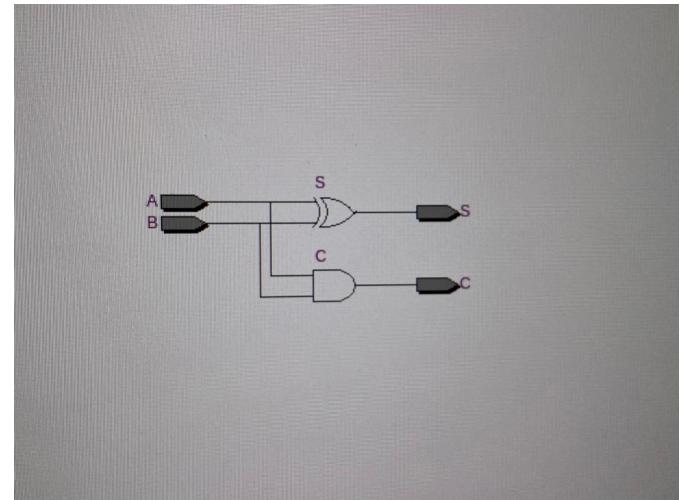
Truth Table and boolean expression for Full-Adder:

Inputs			Outputs	
A	B	C _{in}	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Netlist RTL viewer snapshots:



Full adder circuit



Half adder circuit

VHDL Code for Half-Adder:

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

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

architecture dataflow of half_adder is
Begin
  S <= A xor B;
  C <= a and b;
end dataflow;
```

VHDL Code for Half-Adder:

Data flow modelling:

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

entity full_adder is
port(a, b, c: in bit;
      sum, carry : out bit);
end full_adder;

architecture dataflow of full_adder is
begin
  sum <= a xor b xor c;
  carry <= (a and b) or ((a xor b) and c);
end dataflow;
```

Behavioural modelling:

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

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

architecture behavior of full_adderb is
Begin
c1: process (A,B,Ci)
Begin
if A = '0' and B='0' then
  S <= Ci;
  C <= 0;
if A = '0' and B='1' then
  S <= not Ci;
  C <= Ci;
if A = '1' and B='0' then
  S <= not Ci;
  C <= Ci;
if A = '1' and B='1' then
  S <= Ci;
  C <= 1;
end if;
end process c1;
end behavior;
```

Structural modelling:

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

entity or_gate is
  Port ( X,Y : in STD_LOGIC;
         Z : out STD_LOGIC);
end or_gate;
```

```
architecture dataflow of or_gate is
begin
  Z <= X OR Y;
end dataflow;
```

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

entity FAdder is
  Port ( A, B, Ci : in STD_LOGIC;
         S, C : out STD_LOGIC);
end FAdder;
```

architecture structural of FAdder is

```
component half_adder is
  Port ( A,B : in STD_LOGIC;
         S,C : out STD_LOGIC);
end component;
```

```
component or_gate is
  Port ( X,Y: in STD_LOGIC;
         Z: out STD_LOGIC);
end component;
```

```
SIGNAL S0,S1,S2:STD_LOGIC;
```

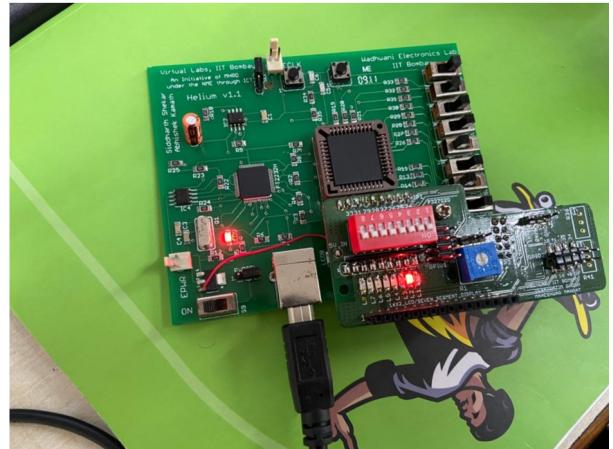
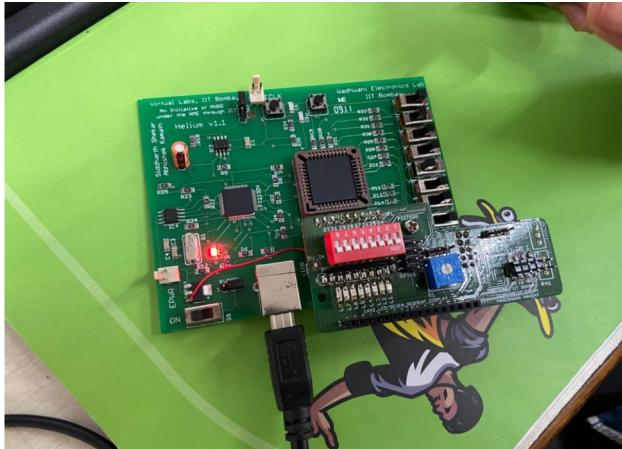
```
begin
```

```
U1:HA PORT MAP(A=>A,B=>B,S=>S0,C=>S1);
U2:HA PORT MAP(A=>S0,B=>Ci,S=>S,C=>S2);
U3:ORGATE PORT MAP(X=>S2,Y=>S1,Z=>C);
```

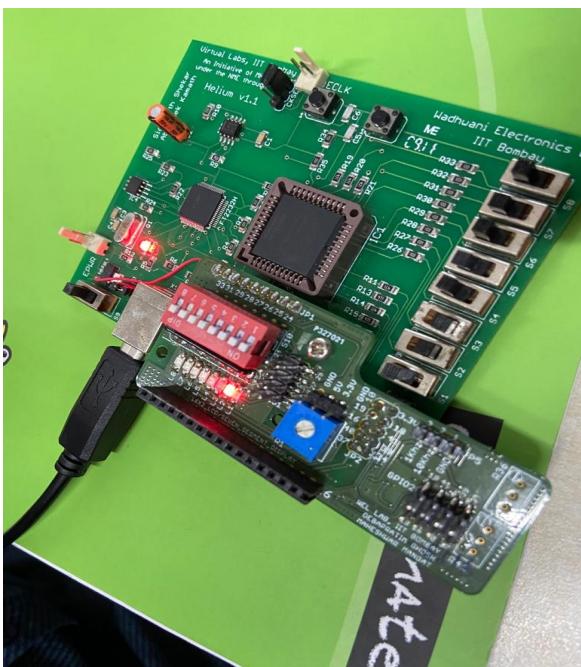
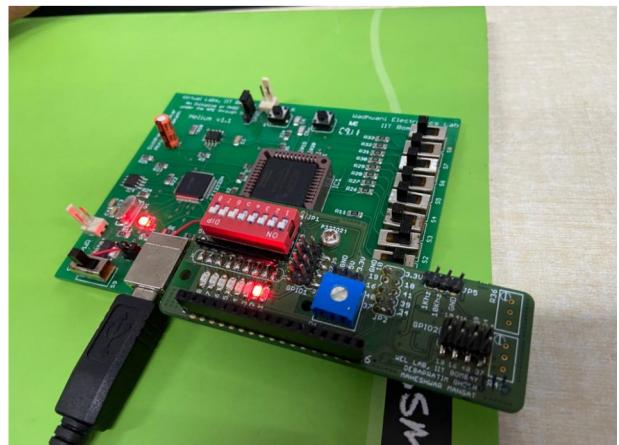
```
end structural;
```

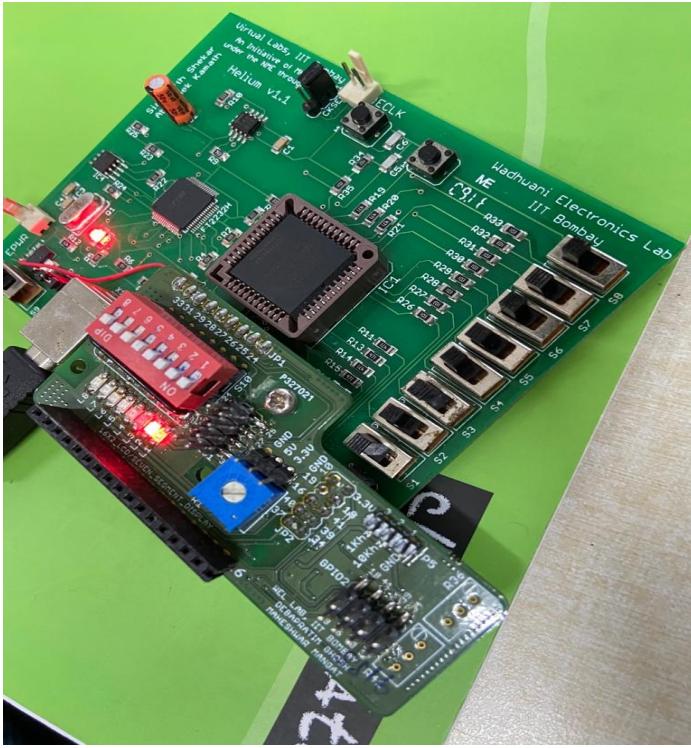
Circuit and Simulation snapshots:

Half-adder



Full adder circuit





Results and Conclusions:

We successfully verified the functionality of **FULL-ADDER** and **HALF-ADDER** 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.