DIGITAL SYSTEM DESIGN
ACTIVITY-I

NAME - Devleena Kumari
Roll No. - 2405051

Q1) **Implement half adder using data flow modeling in verilog HDL. Hence implement a full adder by calling the half adder module in verilog HDL.**

Write the test bench for each of the cases. Show the RTL schematics and output waveforms of half and full adder.

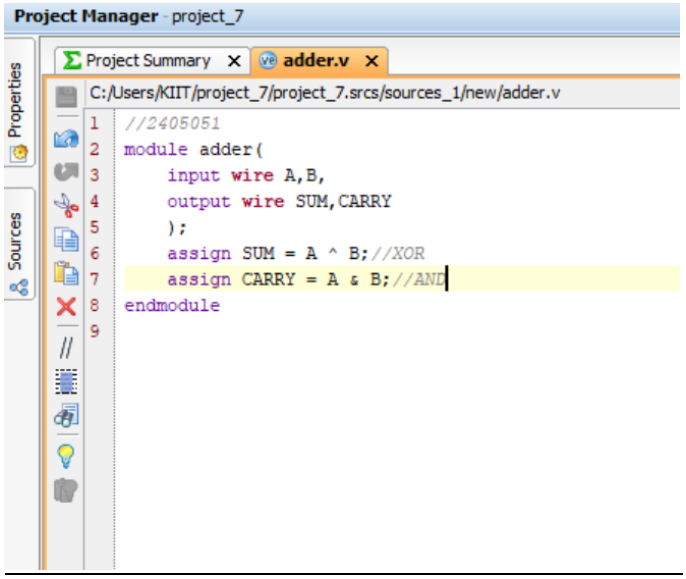
Solution:-

HALF ADDER:-

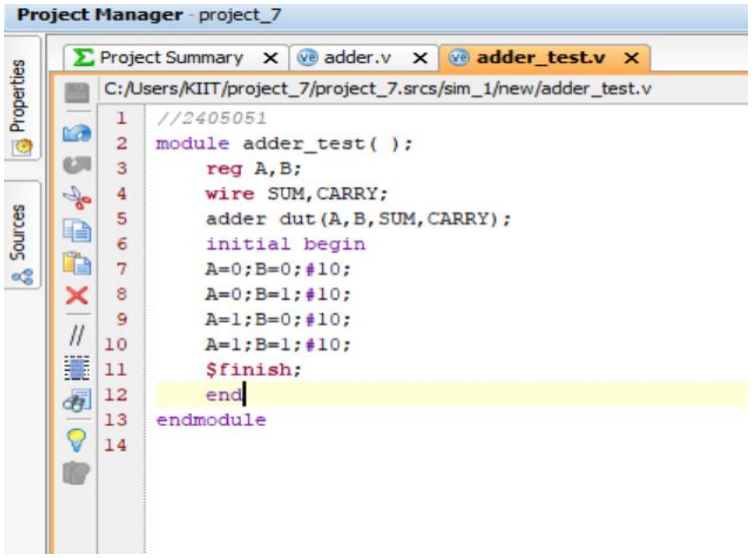
Truth Table:-

A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

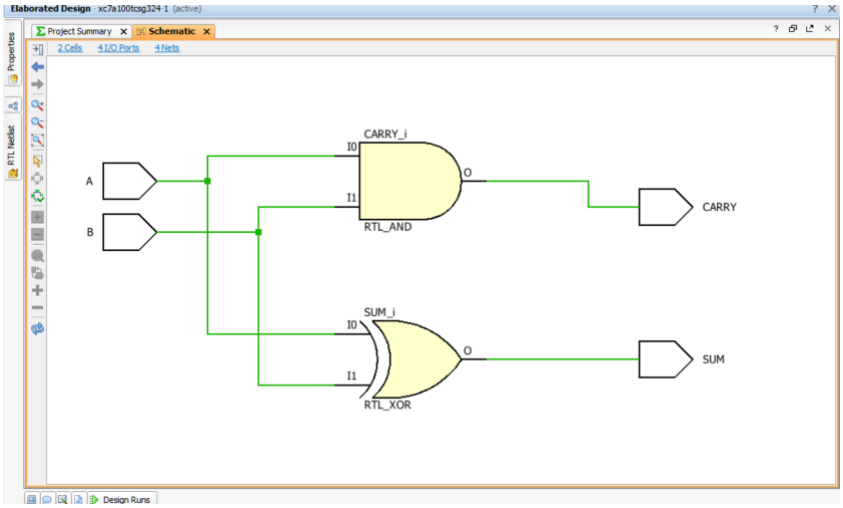
HDL CODE:-



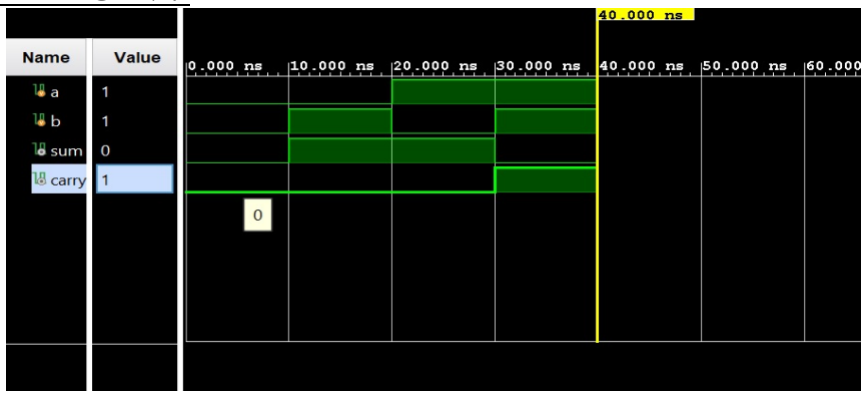
TESTBENCH:-



SCHEMATIC:-



WAVEFORM:-

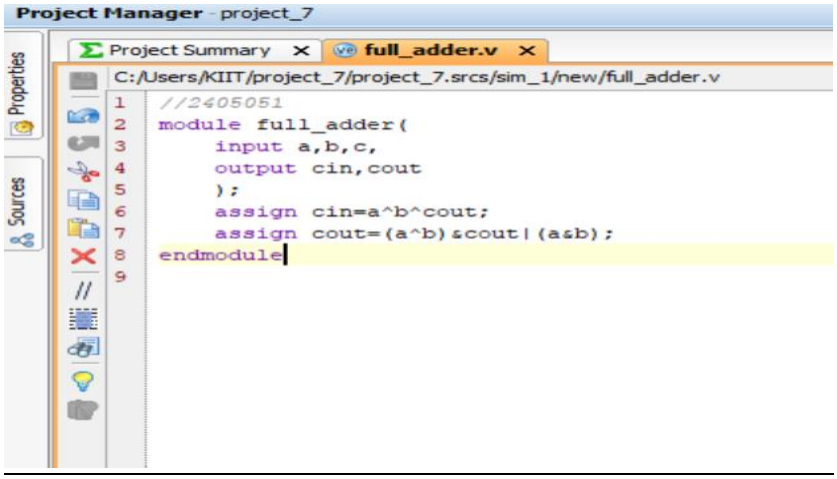


FULL ADDER:-

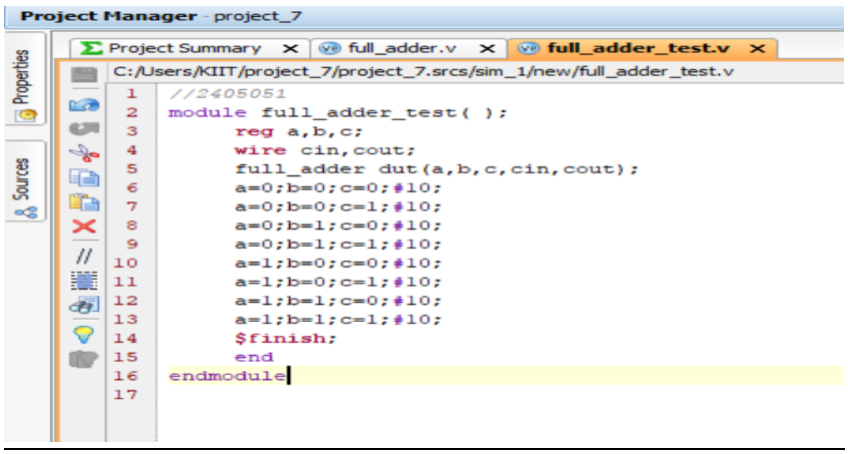
Truth Table:-

a	b	c	cin	cout
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

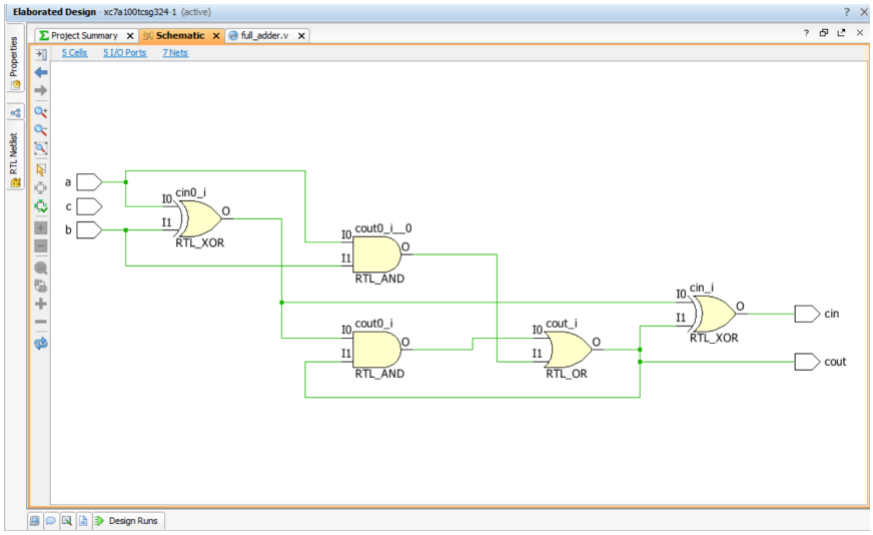
HDL CODE:-



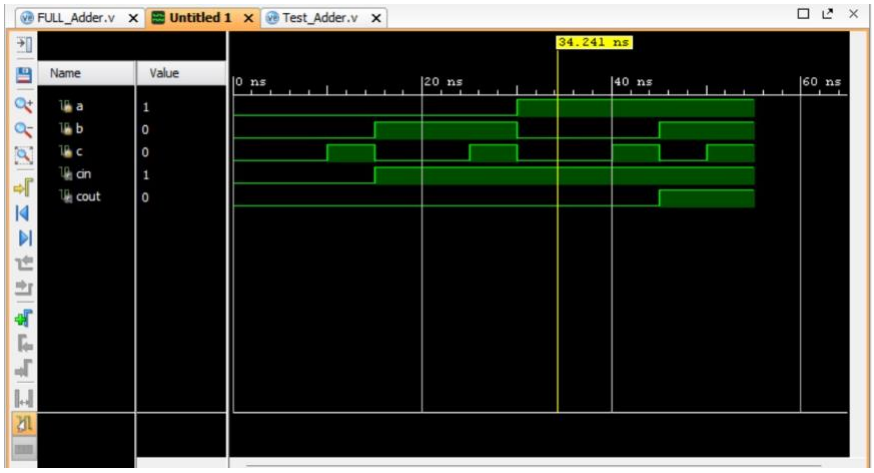
TESTBENCH:-



SCHEMATIC:-



WAVEFORM:-



Q2) **Implement a 3 to 8 line decoder in verilog HDL. Consider F(P,Q,R) = ∑(All digits of your roll number upto digit 7), and ignore the repetition. Implement the function using 3 to 8 line decoder using verilog HDL.**

Solution:-

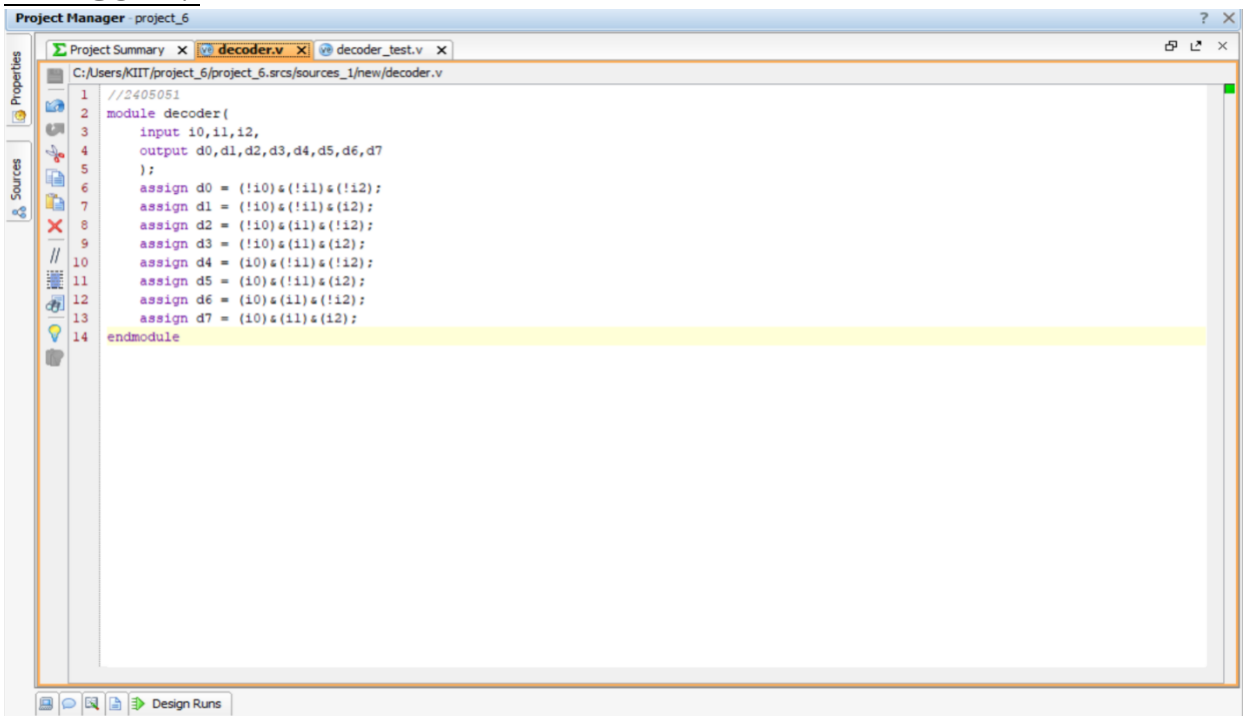
Digits in Roll no.2405051 ----> {2,4,0,5,0,5,1}
----> unique set {0,1,2,4,5}
Therefore: F(P,Q,R) =∑ m(0,1,2,4,5)
(assume order P = MSB , R=LSB)

TRUTH TABLE:-

P	Q	R	minterm	F
0	0	0	0	1
0	0	1	1	1
0	1	0	2	1
0	1	1	3	0
1	0	0	4	1
1	0	1	5	1
1	1	0	6	0
1	1	1	7	0

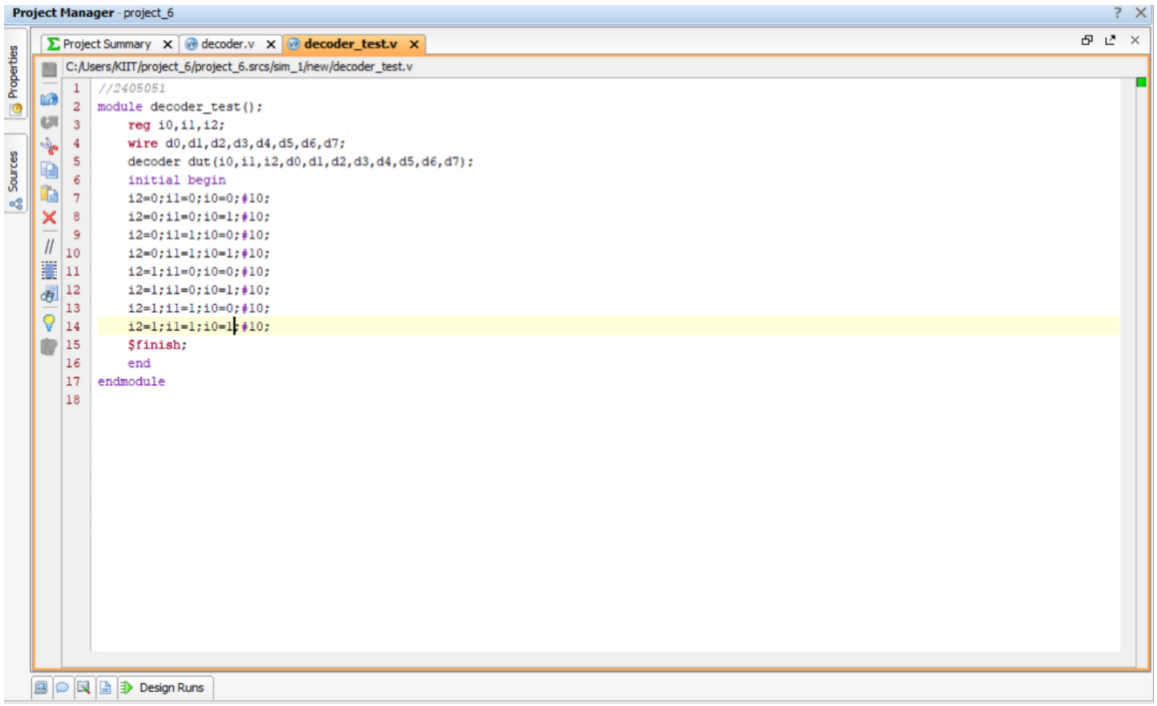
(Checks : F=1 exactly for 0,1,2,4,5.)
F=~Q+(~P&~R)

HDL CODE:-




```
//2405051
module decoder(
    input i0,i1,i2,
    output d0,d1,d2,d3,d4,d5,d6,d7
    );
    assign d0 = (!i0)&(!i1)&(!i2);
    assign d1 = (!i0)&(!i1)&(i2);
    assign d2 = (!i0)&(i1)&(!i2);
    assign d3 = (!i0)&(i1)&(i2);
    assign d4 = (i0)&(!i1)&(!i2);
    assign d5 = (i0)&(!i1)&(i2);
    assign d6 = (i0)&(i1)&(!i2);
    assign d7 = (i0)&(i1)&(i2);
endmodule
```

TESTBENCH:-


Project Manager - project_6
Properties
Sources
Project Summary
decoder.v
decoder_test.v
C:/Users/KIIT/project_6/project_6.srcs/sim_1/new/decoder_test.v
1 //2405051
2 module decoder_test();
3 reg i0,i1,i2;
4 wire d0,d1,d2,d3,d4,d5,d6,d7;
5 decoder dut(i0,i1,i2,d0,d1,d2,d3,d4,d5,d6,d7);
6 initial begin
7 i2=0;i1=0;i0=0;#10;
8 i2=0;i1=0;i0=1;#10;
9 i2=0;i1=1;i0=0;#10;
10 i2=0;i1=1;i0=1;#10;
11 i2=1;i1=0;i0=0;#10;
12 i2=1;i1=0;i0=1;#10;
13 i2=1;i1=1;i0=0;#10;
14 i2=1;i1=1;i0=1;#10;
15 $finish;
16 end
17 endmodule
18
Design Runs