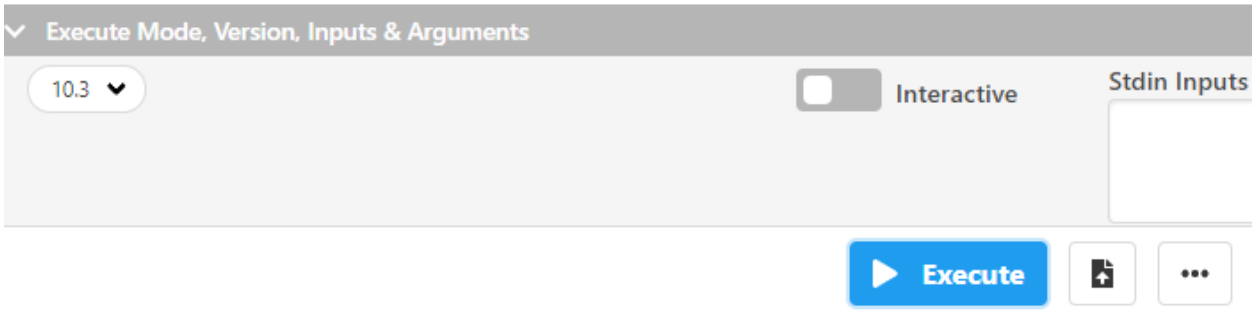
CO LAB WEEK 6

(CED19I027 N Sree Dhyuti)

Implement the Verilog modeling for the Carry Lookahead Adder of 4 bits?

```
//Lab 6: Carry Look Ahead Adder
//Set the measurement and percesion of time
`timescale 1ns / 1ps

//Create a module for Carry Look Ahead Function and define all the inputs , outputs and wires involved

module CLA(a,b,cin,sum,cout);
    input[3:0] a,b;       //a,b are 4-bit number
    input cin;            //Initial carry
    output [3:0] sum;     // Output is a 4-bit binary number
    output cout;          // cout is the final carry
//Define all wires
    wire p0,p1,p2,p3,g0,g1,g2,g3,c1,c2,c3,c4;

// Pi = XOR(Ai,Bi)
    assign p0=(a[0]^b[0]), p1=(a[1]^b[1]), p2=(a[2]^b[2]), p3=(a[3]^b[3]);
// Gi = AND(Ai,Bi)
    assign g0=(a[0]&b[0]), g1=(a[1]&b[1]), g2=(a[2]&b[2]), g3=(a[3]&b[3]);
//C(i) = G(i-1) + P(i)C(i-1)
    assign c0=cin, c1=g0|(p0&cin), c2=g1|(p1&g0)|(p1&p0&cin), c3=g2|(p2&g1)|(p2&p1&g0)|(p1&p1&p0&cin), c4=g3|(p3&g2)|(p3&p2&g1)|(p3&p2&p1&g0)|(p3&p2&p1&p0&cin);
// Sum(i) = XOR(Pi,Ci)
    assign sum[0]=p0^c0, sum[1]=p1^c1, sum[2]=p2^c2, sum[3]=p3^c3;
    assign cout=c4;
endmodule

//Test Bench for CLA
module CLA_tb;
    reg[3:0] a,b;
    reg cin;
    wire[3:0] sum;
    wire cout;

    CLA rpa(a,b,cin,sum,cout);

    initial begin       //Give all input values of a,b

        #0  a=4'b0000;b=4'b0000;cin=0;
        #10  a=4'b0000;b=4'b0000;cin=1;
        #10  a=4'b0000;b=4'b1111;cin=0;
        #10  a=4'b0001;b=4'b1111;cin=0;
    end
```

```
32
33      CLA rpa(a,b,cin,sum,cout);
34
35      initial begin       //Give all input values of a,b
36
37          #0  a=4'b0000;b=4'b0000;cin=0;
38          #10  a=4'b0000;b=4'b0000;cin=1;
39          #10  a=4'b0000;b=4'b1111;cin=0;
40          #10  a=4'b0001;b=4'b1111;cin=0;
41      end
42
43      initial begin
44          $monitor("%d: a=%b b=%b cin=%b sum=%b cout=%b",$time,a,b,cin,sum,cout);
45          //$dumpfile("FA.vcd");
46          //$dumpvars(0,FA_tb);
47      end
48  endmodule
49
```



esult

PU Time: 0.00 sec(s), Memory: 7260 kilobyte(s)

```
 0: a=0000 b=0000 cin=0 sum=0000 cout=0
10: a=0000 b=0000 cin=1 sum=0001 cout=0
20: a=0000 b=1111 cin=0 sum=1111 cout=0
30: a=0001 b=1111 cin=0 sum=0000 cout=1
```

The End
