**Experiment-4 : Carry Look Ahead Adder**

**Objective:**

To design a 4-bit carry look-ahead and test it using 4-bit up counter. The test bench should check the generated output with the expected output and prints pass/fail messages.

**Tool Used:**

Xilinx ISE.

**Theory:**

A regular CLA is implemented with 4 bits and the test pattern in generated in the test bench using always block and a local clock. The 2nd input is derived from the 1st counter input, carry is generated randomly.

**DUT Code:**

module cla(input [3:0]a, b, input cin, output [3:0]sum, output cout);

    wire [3:0]g,p;

    wire [4:1]x;

    assign g = a&b;

    assign p = a^b;

    assign sum = p^{x[3:1],cin};

    assign x[4:1] = g | (p & {x[3:1],cin});

    assign cout = x[4];

endmodule

**TB Code:**

module tb;

    reg [3:0] a,b; reg cin; reg clk = 0;

    wire [3:0] sum; wire cout;

    integer x = 0, y = 0;

    cla uut (a,b,cin,sum,cout);

initial forever #5 clk = !clk;

    always@(posedge clk) begin

        a = y;

        b = y^1;

        cin = $random;

        y = y+1;

        #1;

        if((a+b+cin) != {cout,sum}) x = x+1;

    end

    initial begin

        #100;

        if(!x) $display("success");

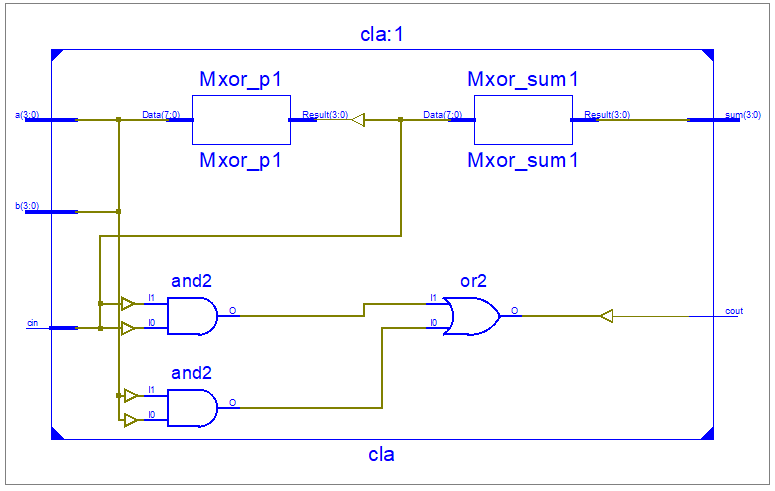
        else $display("%d number of errors",x);

        $finish;

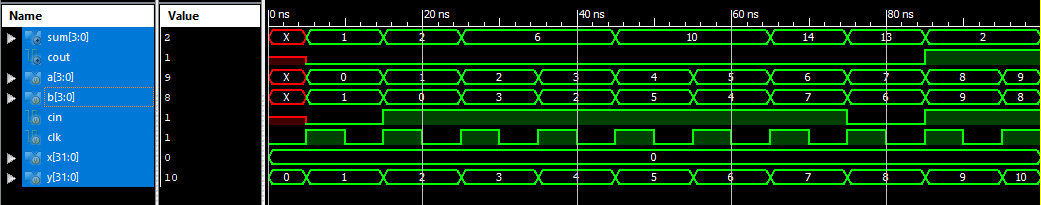
    end

endmodule

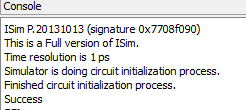
**RTL Diagram:**

****

**Output Waveform:**

****

**Simulation Output:**

****

**Result:**

The simulation output and the RTL diagram is observed and found to be valid.