**Experiment-2: Full Adder**

**Objective:**

To design a Full Adder and write a simple test bench for it. The test bench should generate stimulus to completely verify the functionality of the design under test with delay of 20ns.

**Theory:**

A full adder is a digital circuit that performs addition. Full adders are implemented with logic gates in hardware. A full adder adds three one-bit binary numbers, two operands and a carry bit. The adder outputs two numbers, a sum and a carry bit.

**Design Code:**

module fa(input a, b, c, output s, cy);

    assign s = a ^ b ^ c; // sum expression

    assign cy = a & b | b & c | a & c; //carry expression

endmodule

**Testbench Code:**

module tb;

    // Inputs

    reg a; b; c;

    // Outputs

    wire s; cy;

    // Instantiate the Unit Under Test (UUT)

    fa uut (.a(a), .b(b), .c(c), .s(s), .cy(cy));

    initial begin

        // Initialize Inputs

        a = 0; b = 0;c = 0;

        // Wait 100 ns for global reset to finish

        #20;

        a = 0; b = 0;c = 1;

        // Wait 100 ns for global reset to finish

        #20;

        a = 0; b = 1;c = 0;

        // Wait 100 ns for global reset to finish

        #20;

        a = 0; b = 1;c = 1;

        // Wait 100 ns for global reset to finish

        #20;

      a = 1; b = 0;c = 0;

        // Wait 100 ns for global reset to finish

        #20;

        a = 1; b = 0;c = 1;

        // Wait 100 ns for global reset to finish

        #20;

        a = 1; b = 1;c = 0;

        // Wait 100 ns for global reset to finish

        #20;

        a = 1; b = 1;c = 1;

        // Wait 100 ns for global reset to finish

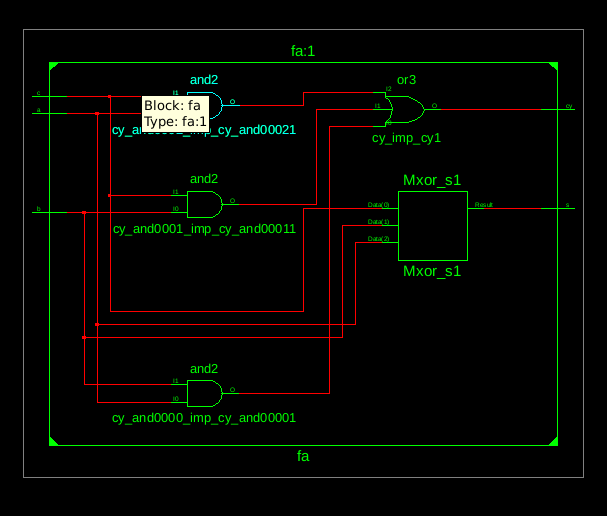
        #20;

    end

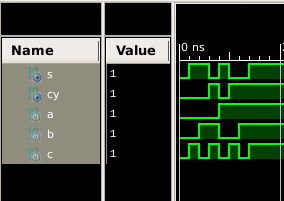
    initial begin $monitor($time, "a=%b, b=%b, c=%b, s=%b, cy=%b", a,b,c,s,cy);end

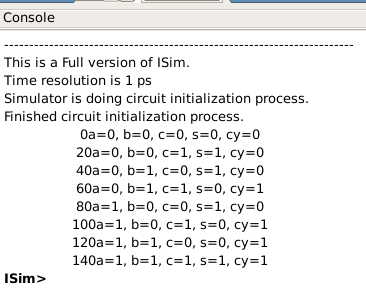
endmodule

**RTL Schematic:**



**Simulation Result:**





**Result:**

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