**Verilog code to implement Sequence detector(1101) using Mealy style**

**Design Code:**

module sequence(input i,clk,rst,output y);

    parameter idle=2'b00,s1=2'b01,s11=2'b10,s110=2'b11;

    reg [1:0] state,next\_state;

    always@(\*)begin

        case(state)

            idle: next\_state = i ? s1   : idle;

            s1  : next\_state = i ? s11  : idle;

            s11 : next\_state = i ? s11  : s110;

            s110: next\_state = i ? s1   : idle;

            default: next\_state = idle;

        endcase

    end

    always@(posedge clk)begin

        if(rst) state <= idle;

        else state <= next\_state;

    end

    assign y = (state==s110)&&i;

endmodule

**Test Bench:**

module tb();

    reg i,clk,rst;

    wire y;

    sequence dut(i,clk,rst,y);

    initial begin

        clk = 1'b0;

        forever #5 clk = !clk;

    end

    initial begin

        $monitor("the state is %d input is %b next state is %d and the output is %b”, dut.state,i,dut.next\_state,y);

        rst = 1'b1;

        #5;

        rst = 1'b0;

        @(negedge clk); i = 0;

        @(negedge clk); i = 1;

        @(negedge clk); i = 1;

        @(negedge clk); i = 0;

        @(negedge clk); i = 1;

        @(negedge clk); i = 1;

        @(negedge clk); i = 0;

        @(negedge clk); i = 0;

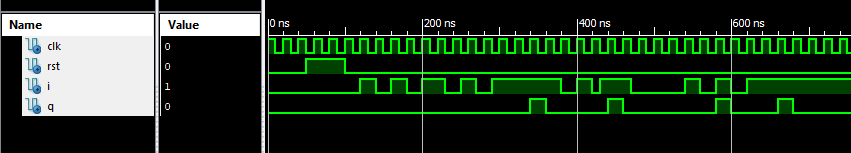
        @(negedge clk); i = 1;

        $finish;

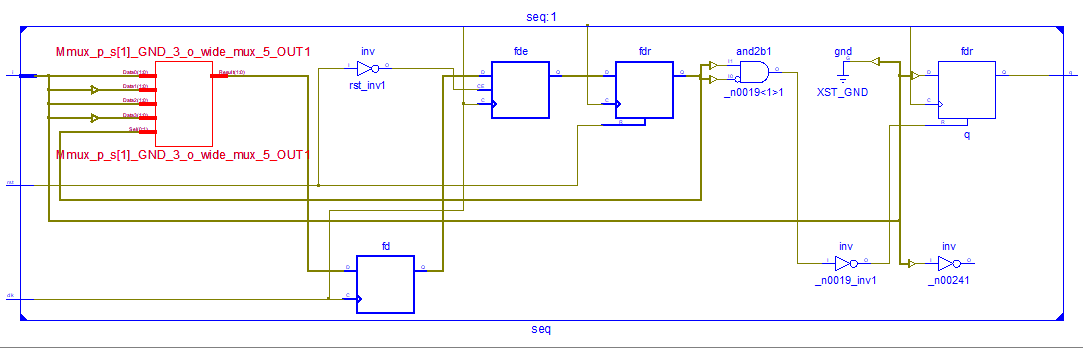
    end

endmodule

**Simulation Result:**

****

**RTL Diagram:**

****