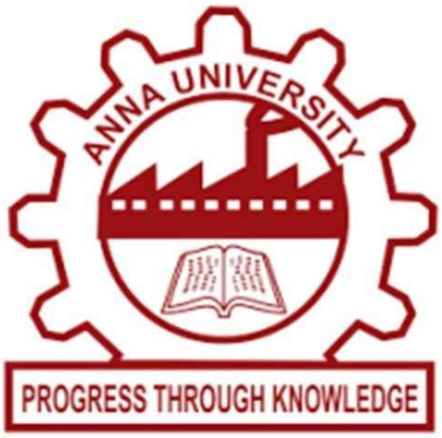
MAVEN SILICON HACKATHON
LEVEL 2

Submitted by

NITHISH REDDY KVS - 2022105065

ELECTRONICS AND COMMUNICATION ENGINEERING



COLLEGE OF ENGINEERING, GUINDY
CHENNAI -600025

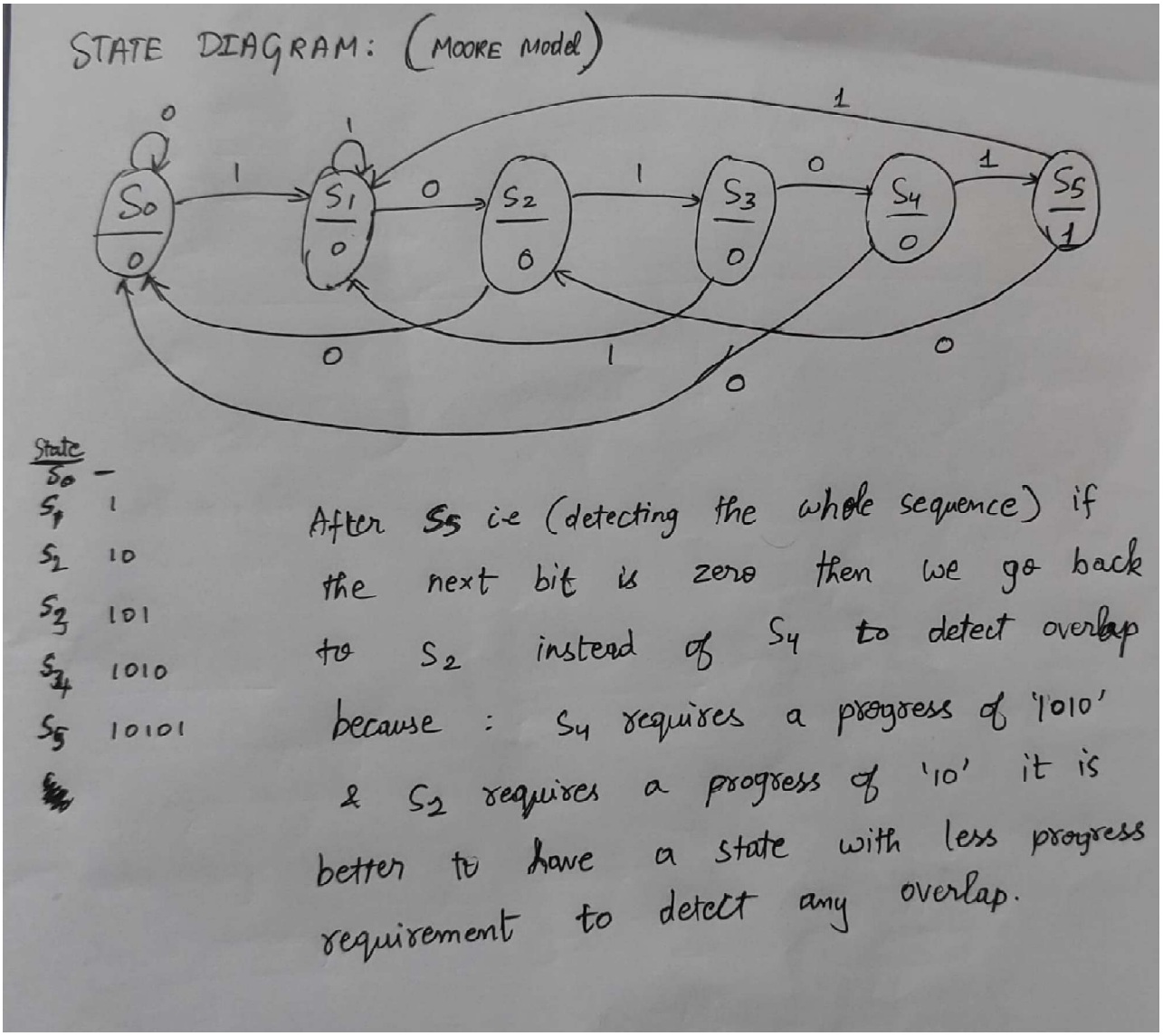
AIM

To Design a sequence detector that detects the sequence “10101” from the input data stream with MSB detected first, Draw the Moore FSM for overlapping sequence.

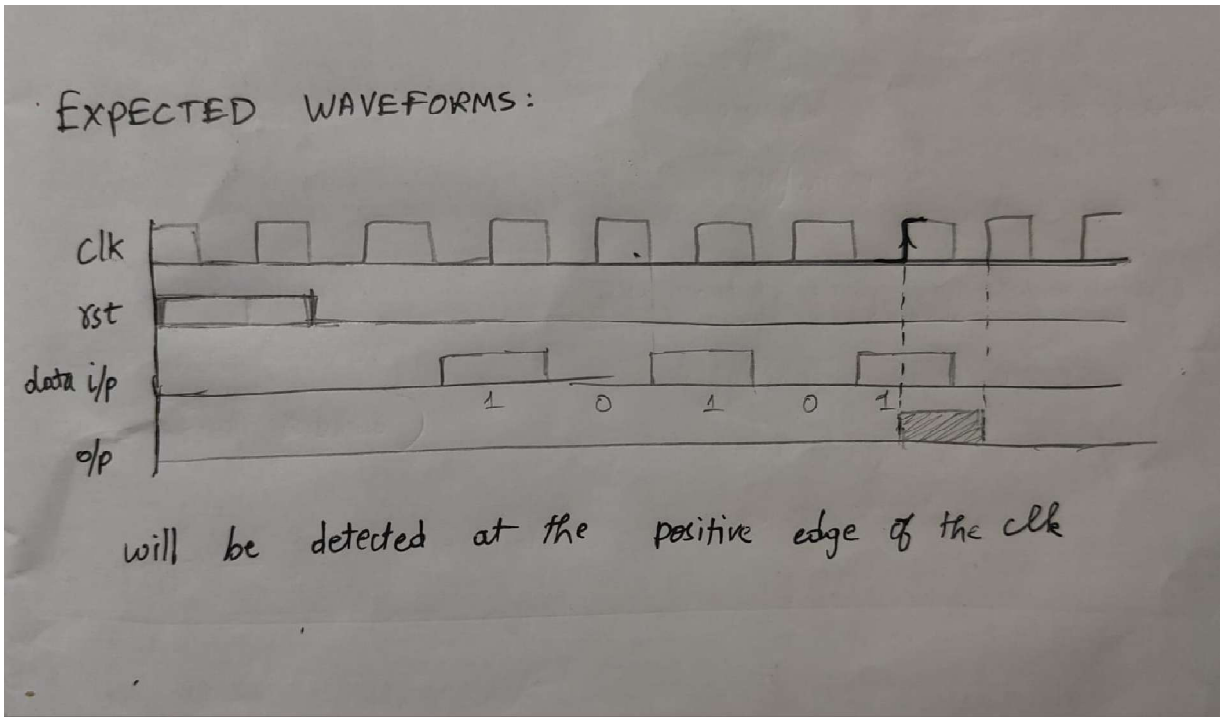
SOFTWARE REQUIRED

Xilinx Vivado

STATE DIAGRAM:



EXPECTED WAVEFORM:



VERILOG CODE:

```
module sequence_detector(
    input clk,
    input reset,
    input data_in,
    output reg detected
);

localparam
    S0 = 3'd0,
    S1 = 3'd1,
    S2 = 3'd2,
```

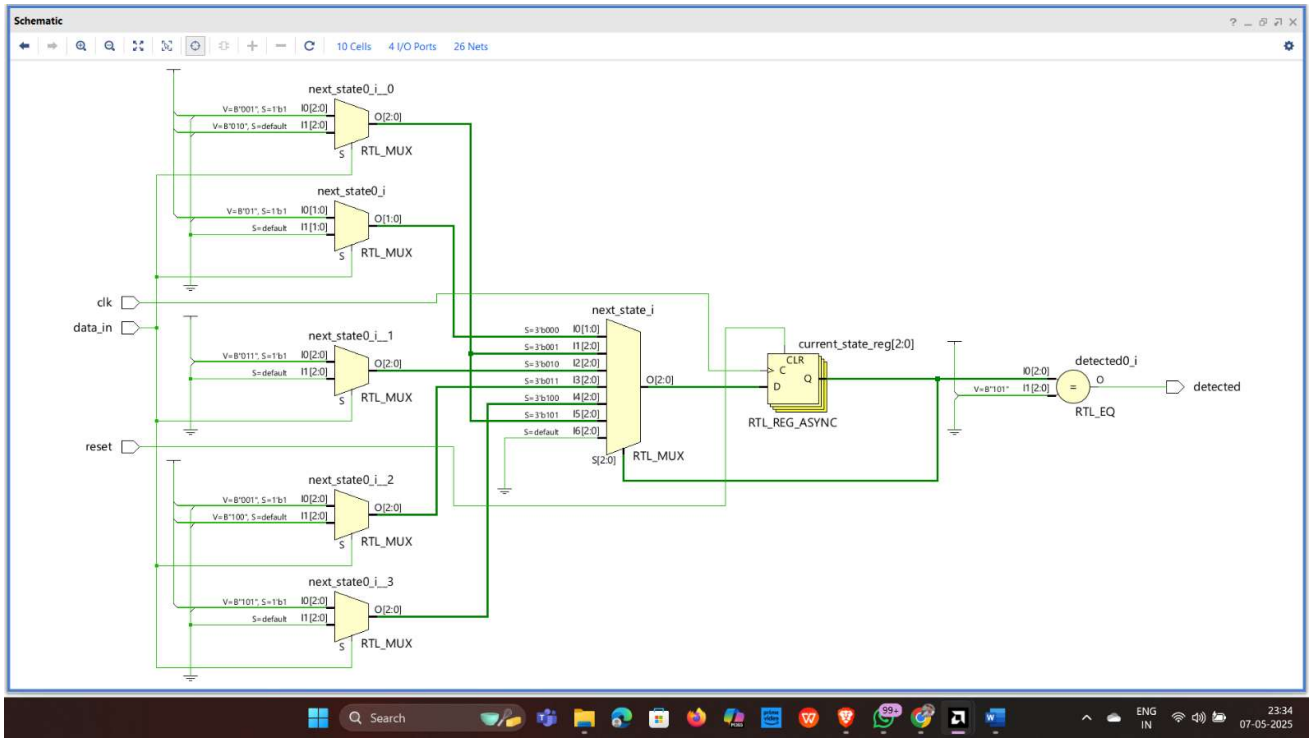
```
    S3 = 3'd3,
    S4 = 3'd4,
    S5 = 3'd5;

reg [2:0] current_state, next_state;

always @(posedge clk or posedge reset) begin
    if(reset) current_state <= S0;
    else current_state <= next_state;
end

always @(*) begin
    case(current_state)
        S0: next_state = data_in ? S1 : S0;
        S1: next_state = data_in ? S1 : S2;
        S2: next_state = data_in ? S3 : S0;
        S3: next_state = data_in ? S1 : S4;
        S4: next_state = data_in ? S5 : S0;
        S5: next_state = data_in ? S1 : S2;
        default: next_state = S0;
    endcase
end

always @(*) begin
    detected = (current_state == S5);
end

endmodule
```

TESTBENCH CODE:

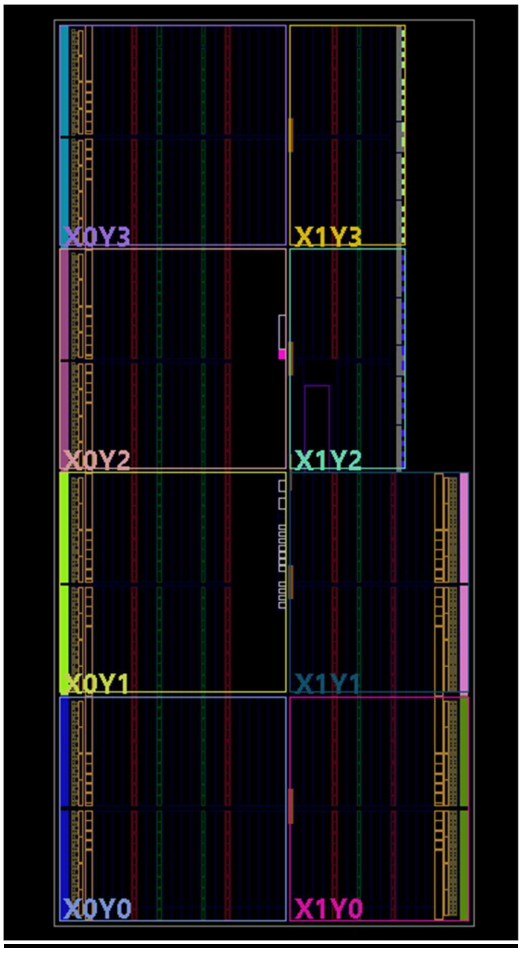
```
`timescale 1ns/1ps

module tb_sequence_detector;
    reg clk, reset, data_in;
    wire detected;

    sequence_detector dut (
        .clk(clk),
        .reset(reset),
        .data_in(data_in),
        .detected(detected));

    always #5 clk = ~clk;

    initial begin
        clk = 0;
        reset = 1;
        data_in = 0;

        #20 reset = 0;

        #10 data_in = 1;
        #10 data_in = 0;
        #10 data_in = 1;
        #10 data_in = 0;
        #10 data_in = 1;
        #10 data_in = 0;
        #10 data_in = 1;
```

```
        #10 data_in = 0;
        #10 data_in = 1;
        #10 data_in = 1;
        #10 data_in = 0;
        #10 data_in = 1;
        #10 data_in = 0;
        #10 data_in = 1;
        #10 data_in = 1;
        #10 data_in = 0;


        #50 $finish;
    end
endmodule
```

RTL SCHEMATIC:



SYNTHESIS DESIGN:



SIMULATION RESULTS:

RESULT:

Thus the sequence detector has been implemented using the moore state logic fsm and the Verilog code and testbench have been written for the same and simulated and verified.