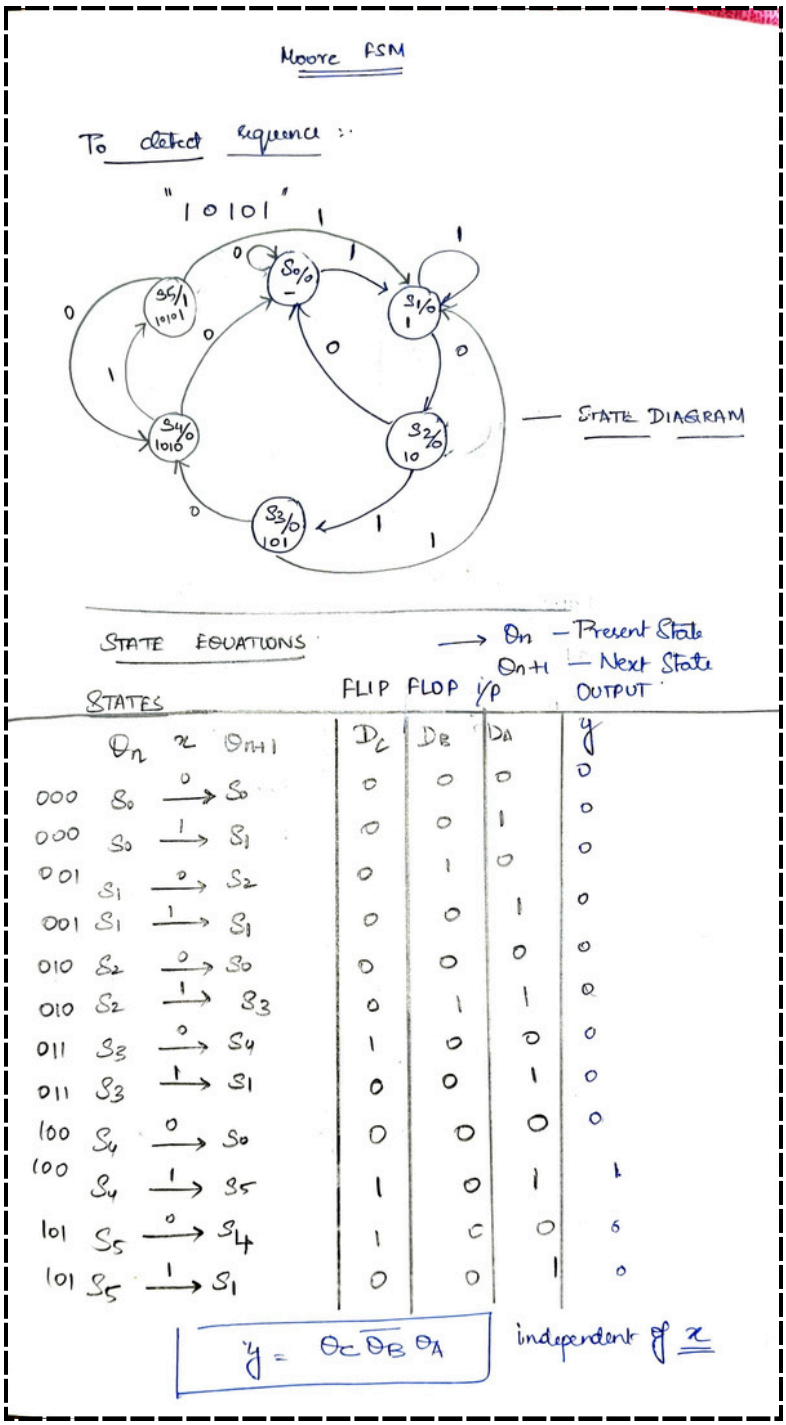
MAVEN SILICON HACKATHON LEVEL 2

Submitted by: SHREYAA VINOD (2022105536)

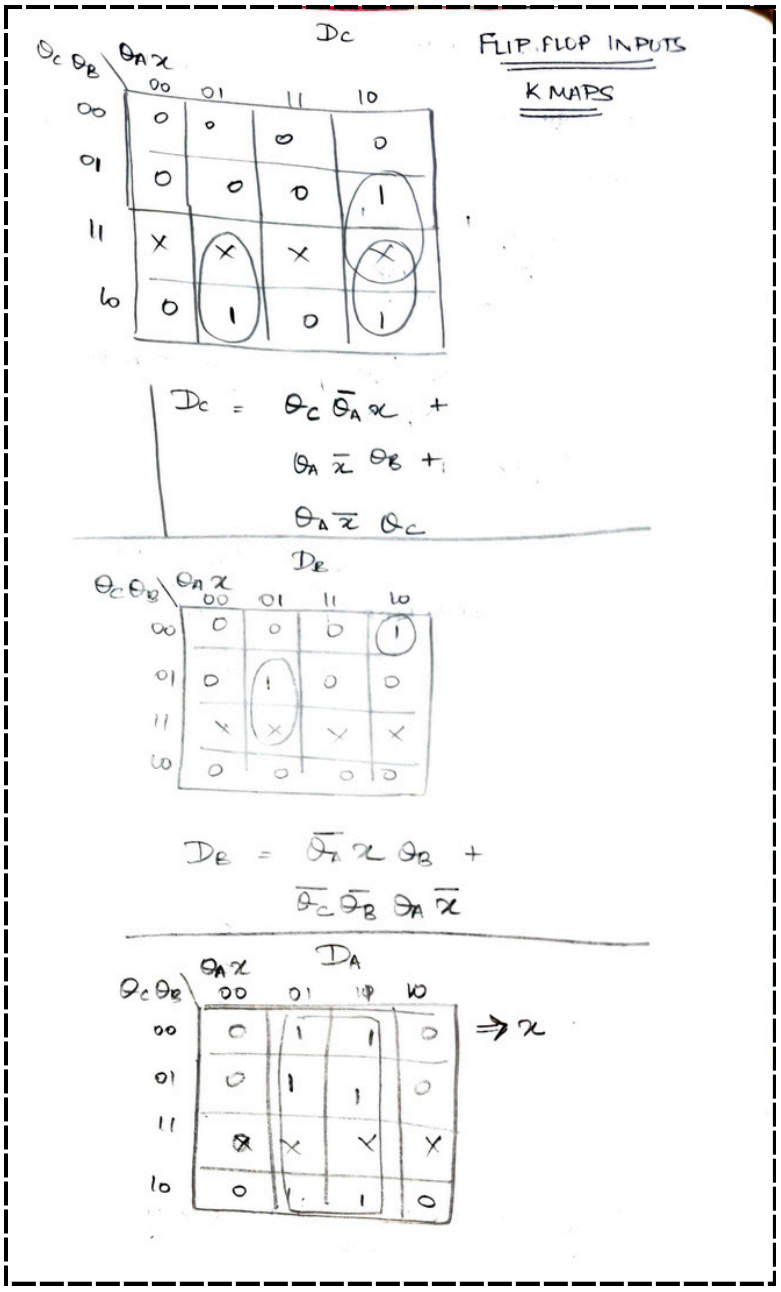
Design a sequence detector that detects the sequence "10101" from the input data stream with MSB detected first. Draw the Moore FSM for overlapping sequence.

[A] Write the RTL code for the design.
[B] Write the TB and verify the RTL.

DESIGN OF MOORE SEQUENCE DETECTOR USING D FLIP FLOPS



D FLIP FLOP INPUTS DETERMINATION:



VERILOG RTL CODE:

TOP MODULE (moore_sd)

```
`timescale 1ns / 1ps

module moore_sd(
        input clk,
        input rst,
        input x,
        output y
```

```
    );
    wire w1,w2,w3,w4,qc,qb,qa,qcb,qbb,qab,xb;
    assign xb=~x;
    assign w1= (qc&qab&x)|(qa&xb&qb)|(qa&xb&qc);
    assign w2=(qab&x&qb)|(qcb&qbb&qa&xb);
    dff dc(clk,rst,w1,qc,qcb);
    dff db(clk,rst,w2,qb,qbb);
    dff da(clk,rst,x,qa,qab);
    assign y=(qc&qbb&qa);

endmodule
```

D FLIP FLOP MODULE (dff)

```
`timescale 1ns / 1ps
module dff(
    input clk,
    input rst,
    input d,
    output reg q,
    output reg qb
    );
    always @(posedge clk or negedge rst)begin
    if (rst==0)begin
    q<=0;
    qb<=1;
```

```
    end
    else if (d==0 || d==1)begin
    q<=d;
    qb<=~d;
    end
    end
endmodule
```

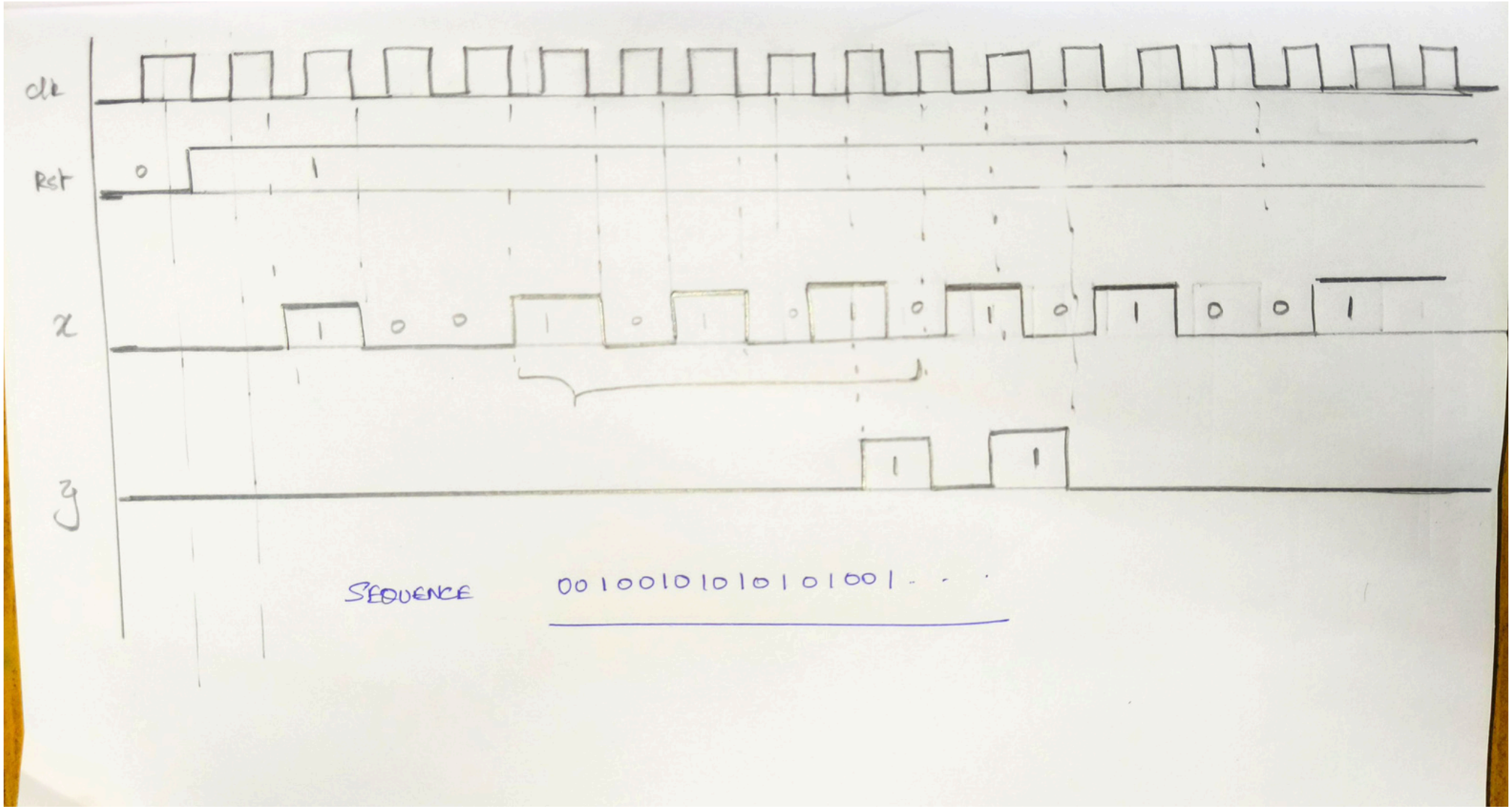
TESTBENCH CODE:

```
`timescale 1ns / 1ps
module moore_sd_tb;

    reg clk, rst, x;
    wire y;

    moore_sd uut (
        .clk(clk),
        .rst(rst),
        .x(x),
        .y(y)
    );
    always #50 clk = ~clk;
    initial begin
    clk=1'b0;
    rst=1'b0;
    x=1'b0; //(s0)
```

```
#100;
 rst=1;
 #100;
 rst=1;
 x=1'b1;
 #100;
 x=1'b0;
 #100;
 x=1'b0;
 #100;
 x=1'b1;
 #100;
 x=1'b0;
 #100;
 x=1'b1;
 #100;
 x=1'b0;
 #100;
 x=1'b1;
 #100;
 x=1'b0;
 #100;
 x=1'b1;
 #100;
 x=1'b0;
 #100;
```

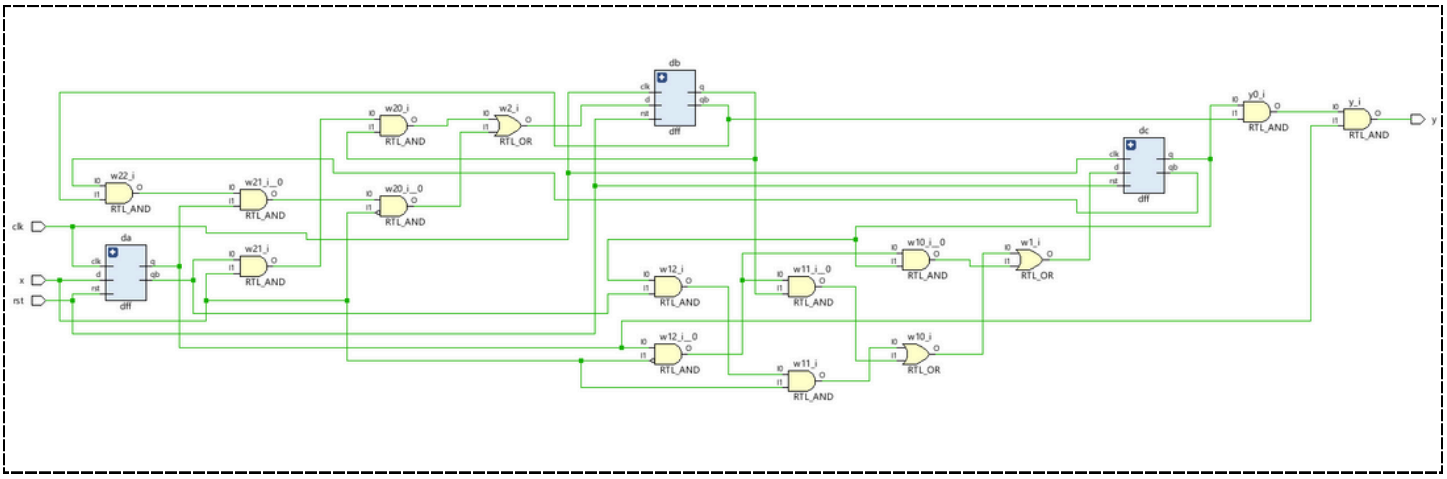
```
  x=1'b0;
  #100;
  x=1'b1;
  #100;
  x=1'b0;
  #100;
  x=1'b1;
  #100;
  x=1'b0;
  #100;
  x=1'b1;
  #300;
   $finish;
   end

endmodule
```

EXPECTED WAVEFORMS:



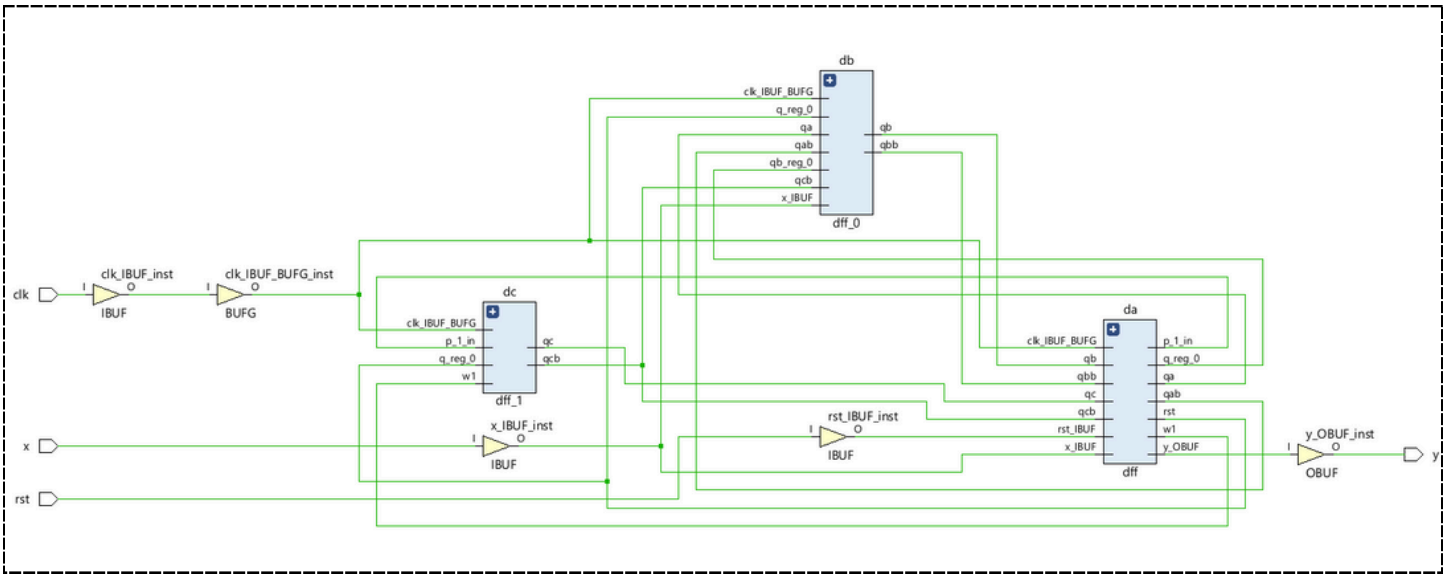
SEQUENCE 0010010101010100 1 . . .

RTL SCHEMATIC

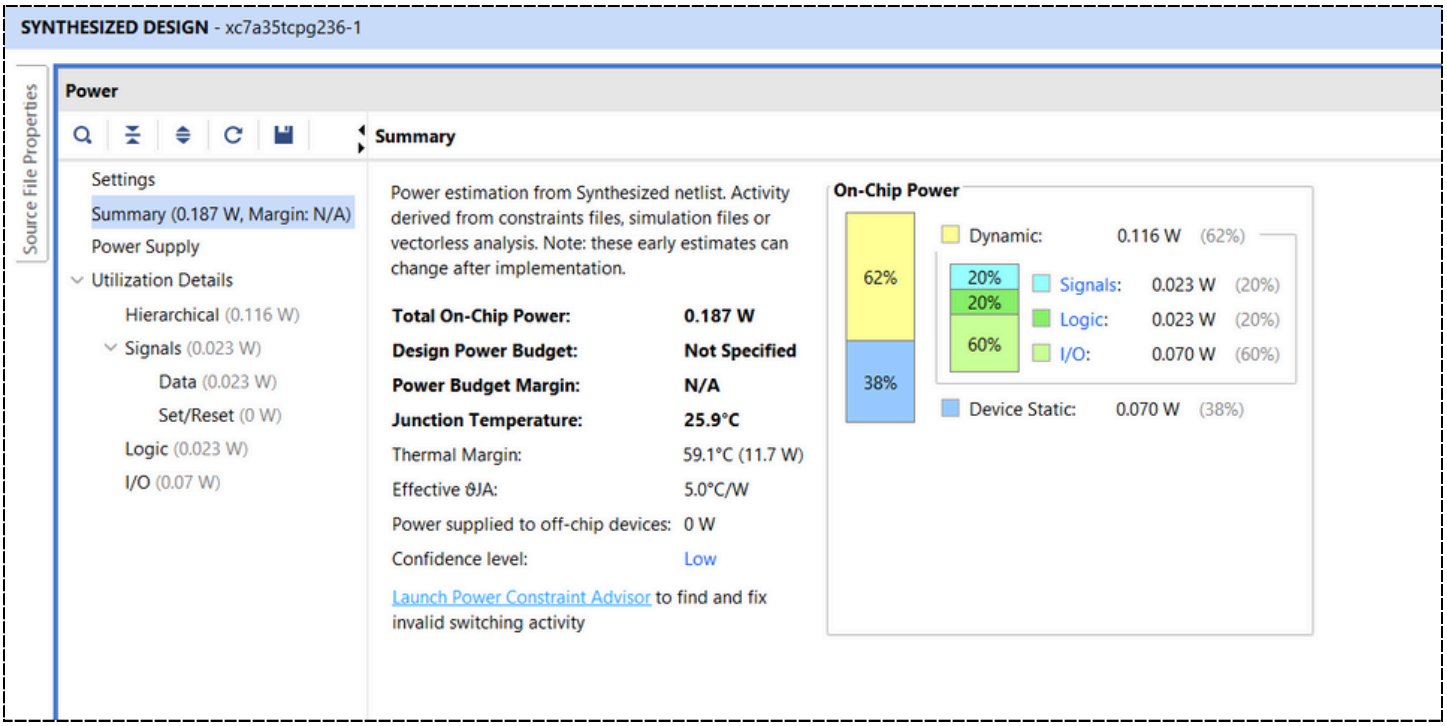


SYNTHESIS:

SCHEMATIC



POWER REPORT:



SIMULATION RESULT:

Given Sequence: 0010010101010010101