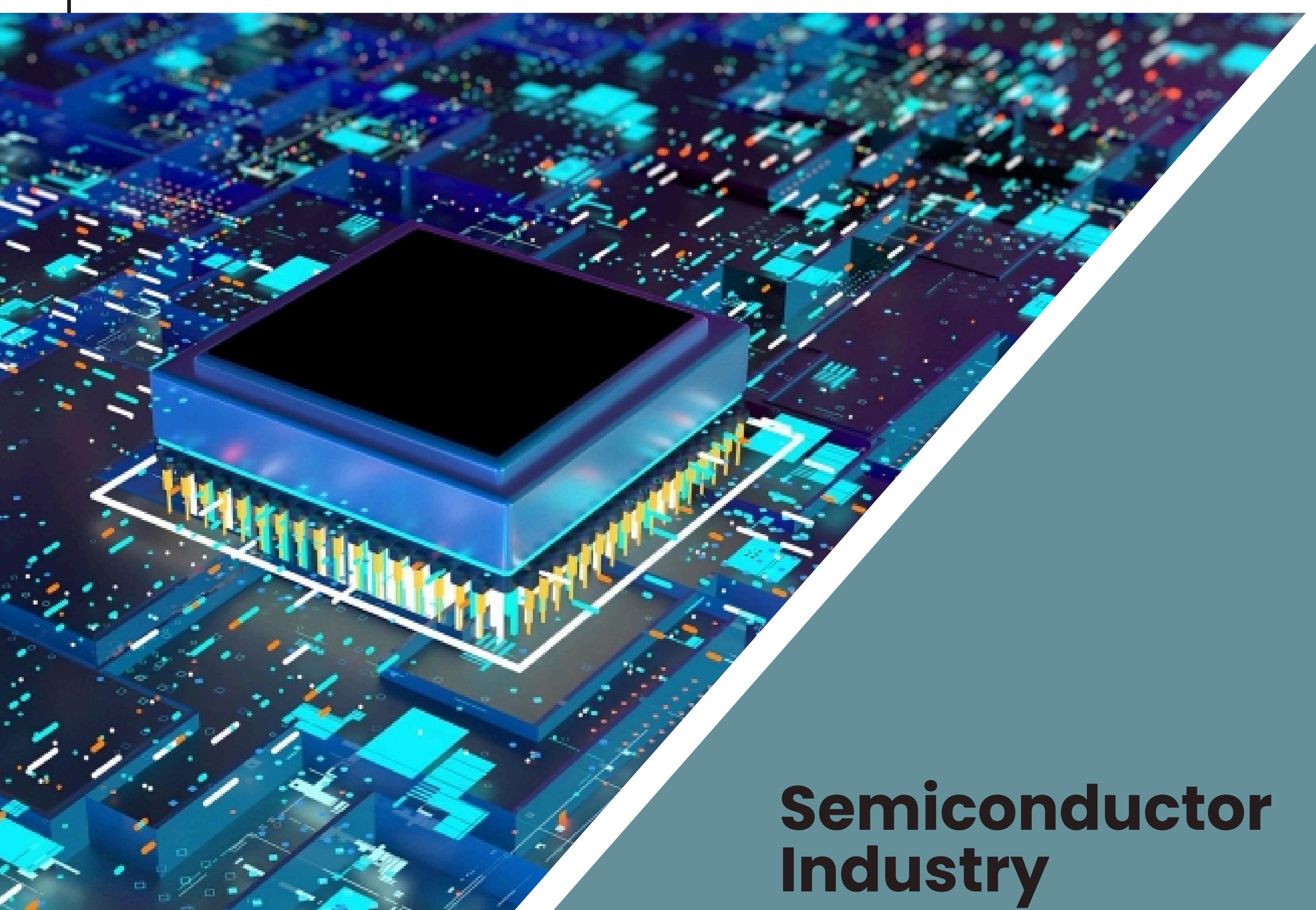


• • •

RTL TO GDS

Article-18

RTL DESIGN Examples



Semiconductor
Industry

Written By-

S. Chinna Venkata Narayana Reddy



DAY-1

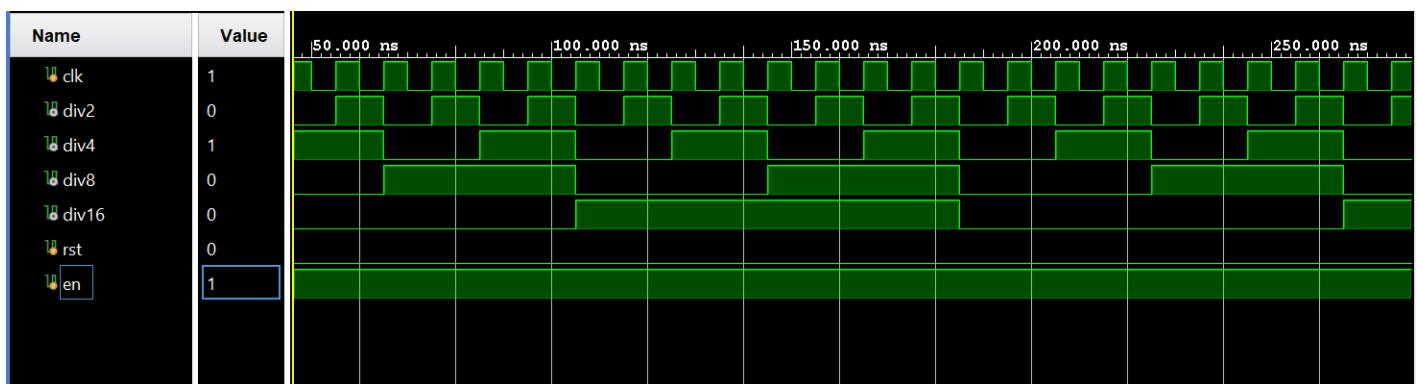
#100DAYSRTL

Aim:- To design the Clock divider (by clk/2^n)

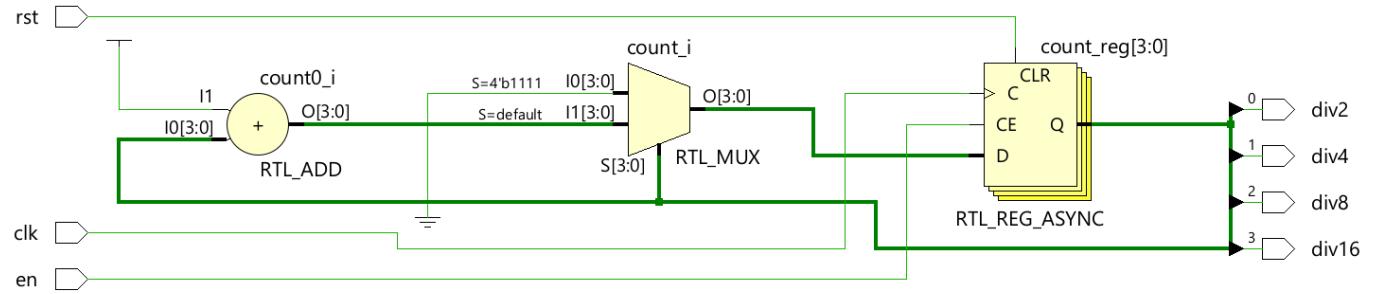
Verilog Code:-

```
Q | S | ← | → | X | ⌂ | F | X | // | ⌚ | ? | 
1 //Author:- Chinna Venkata Narayana Reddy
2 //Date:-28/08/2023
3 module ClockDividerBy2n(
4     input rst, clk, en,
5     output div2, div4, div8, div16
6 );
7     reg [3:0] count;
8
9     always @(posedge clk or posedge rst) begin
10    if (rst)
11        count <= 0;
12    else if (en) begin
13        if (count == 4'd15)
14            count <= 0;
15        else
16            count <= count + 1;
17    end
18 end
19 assign div2 = count[0];
20 assign div4 = count[1];
21 assign div8 = count[2];
22 assign div16 = count[3];
23 endmodule
```

Waveforms:-



Schematics:-





DAY-3

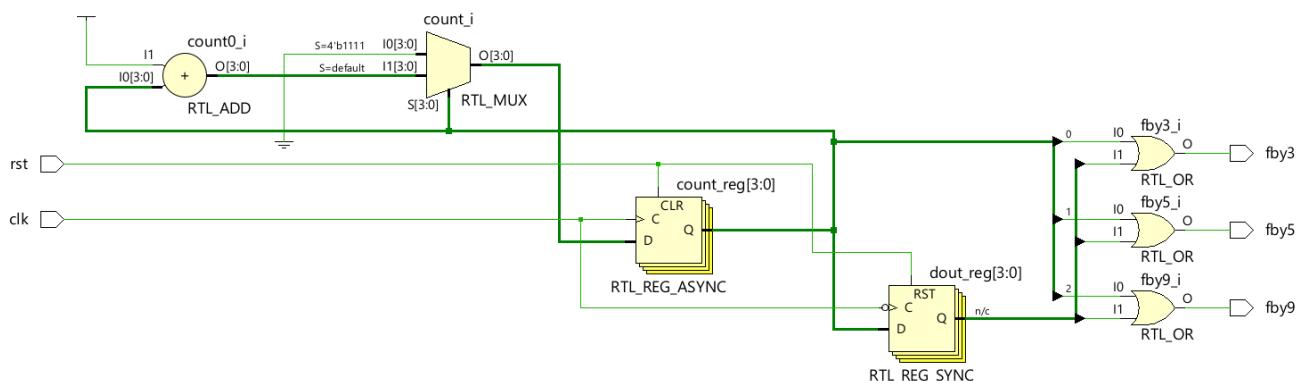
#100DAYSRSL

“Aim”:- To verify the Clock divider (by clk/odd) using Verilog

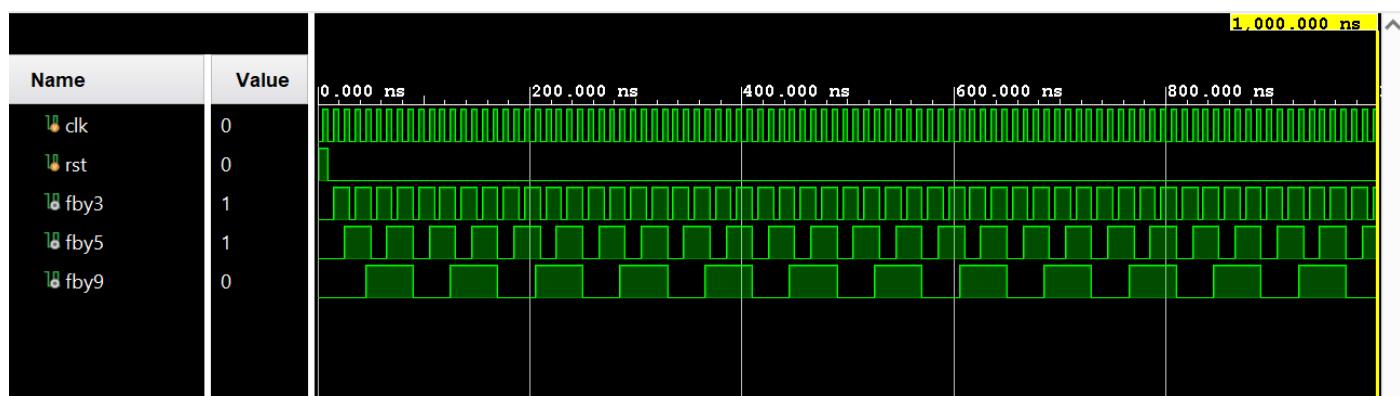
“Verilog Code”:-

```
//Author:- Chinna Venkata Narayana Reddy
//Date:-28/08/2023
module ClockDividerBy3(
    input rst, clk,
    output fby3,fby5,fby9
);
reg [3:0] count,dout;
always @(posedge clk or posedge rst) begin
    if (rst)
        count <= 0;
    else begin
        if (count == 4'd15)
            count <= 0;
        else
            count <= count + 1;
    end
end
always @(negedge clk) begin
if(rst)
    dout<=0;
else
    dout<=count;
end
assign fby3=count[0]||dout[0];
assign fby5=count[1]||dout[1];
assign fby9=count[2]||dout[2];
endmodule
```

Schematics:-



Waveforms:-





DAY-5

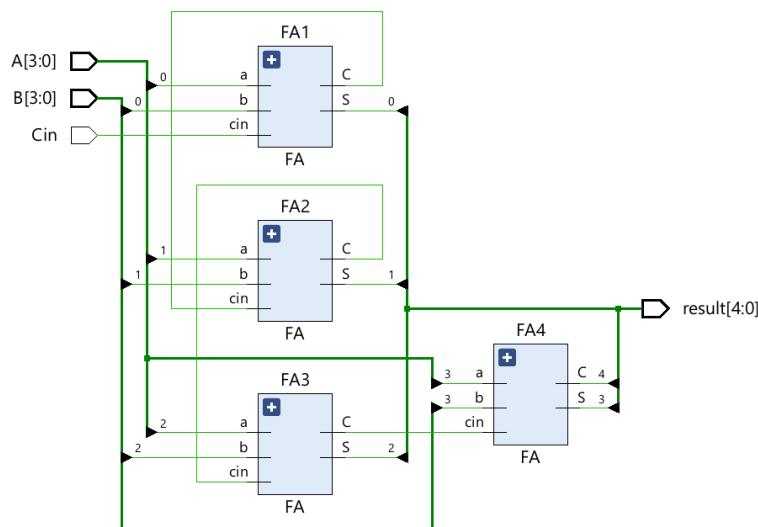
#100DAYSRTL

“Aim”:- To Design the 4 bit ripple carry adder using Verilog

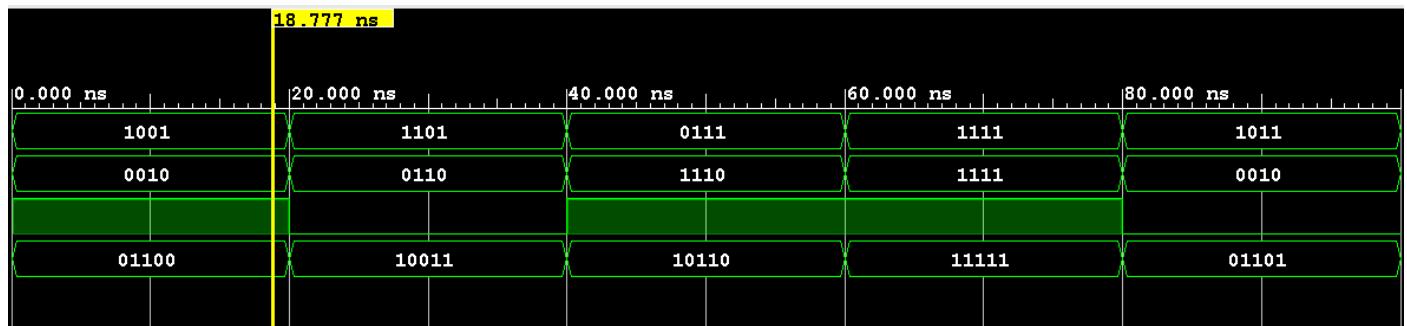
“Verilog Code”:-

```
1 module FA(
2     input a,b,cin,
3     output C,S
4 );
5     assign {C,S}=a+b+cin;
6 endmodule
7 module RCA(
8     input [3:0] A,B,
9     input Cin,
10    output [4:0] result
11 );
12    wire carry;
13    wire [3:0]sum;
14    wire carry1,carry2,carry3,carry4;
15    FA FA1(A[0],B[0],Cin,carry1,sum[0]);
16    FA FA2(A[1],B[1],carry1,carry2,sum[1]);
17    FA FA3(A[2],B[2],carry2,carry3,sum[2]);
18    FA FA4(A[3],B[3],carry3,carry,sum[3]);
19
20    assign result={carry,sum};
21 endmodule
```

“Schematics”:-



“Waveforms”:-



“Console Results”:-

- | A=1001, B=0010, Cin=1, result=01100
- | A=1101, B=0110, Cin=0, result=10011
- | A=0111, B=1110, Cin=1, result=10110
- | A=1111, B=1111, Cin=1, result=11111
- | A=1011, B=0010, Cin=0, result=01101



DAY-7

#100DAYSRTL

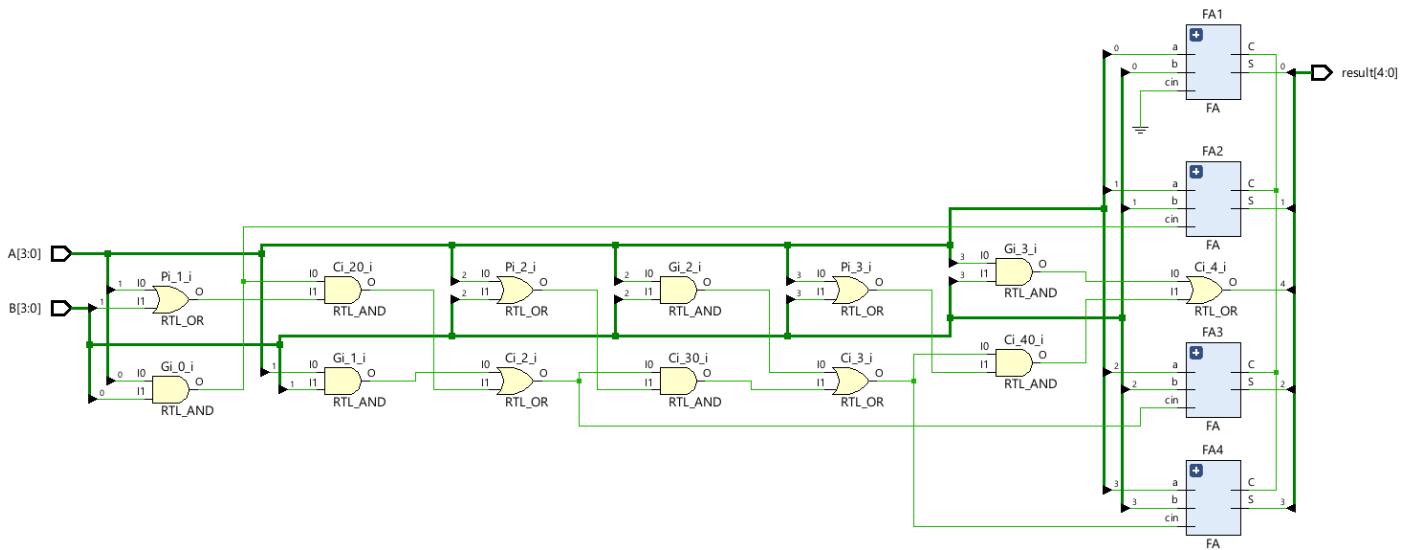
“Aim”:- To Design the 4 bit Carry Look Ahead adder using Verilog

“Verilog Code”:-

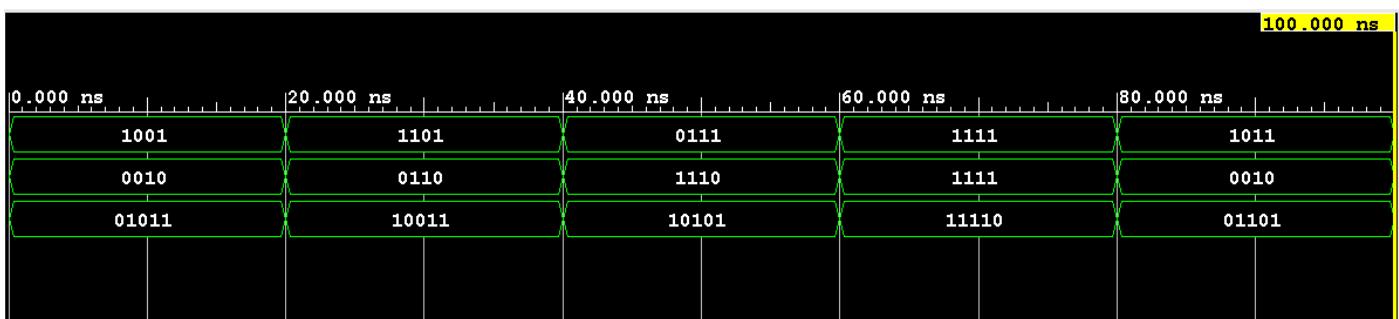
```
module FA(input a,b,cin,output C,S);
    assign {C,S}=a+b+cin;
endmodule

module CLHA(input [3:0] A,B,output [4:0] result);
    wire [4:0] Ci;
    wire [3:0] Gi,Pi;
    wire [3:0] Sum;
    //Full adders
    FA FA1(A[0],B[0],Ci[0],carry,Sum[0]);
    FA FA2(A[1],B[1],Ci[1],carry,Sum[1]);
    FA FA3(A[2],B[2],Ci[2],carry,Sum[2]);
    FA FA4(A[3],B[3],Ci[3],carry,Sum[3]);
    //carry propagate terms
    assign Pi[0]=A[0]|B[0];
    assign Pi[1]=A[1]|B[1];
    assign Pi[2]=A[2]|B[2];
    assign Pi[3]=A[3]|B[3];
    //carry generate terms
    assign Gi[0]=A[0]&B[0];
    assign Gi[1]=A[1]&B[1];
    assign Gi[2]=A[2]&B[2];
    assign Gi[3]=A[3]&B[3];
    //Carry terms
    assign Ci[0]=1'b0;
    assign Ci[1]=Gi[0] | (Ci[0] & Pi[0]);
    assign Ci[2]=Gi[1] | (Ci[1] & Pi[1]);
    assign Ci[3]=Gi[2] | (Ci[2] & Pi[2]);
    assign Ci[4]=Gi[3] | (Ci[3] & Pi[3]);| //Result
    assign result={Ci[4],Sum};
endmodule
```

“Schematics”:-



“Waveforms”:-



“Console Results”:-

```
A=1001,B=0010,result=01011
A=1101,B=0110,result=10011
A=0111,B=1110,result=10101
A=1111,B=1111,result=11110
A=1011,B=0010,result=01101
```

TopModule:-

```
module tb;
  RCA_IF vif();
  RCA dut (vif);
  environment env;
  initial begin
    env = new(vif);
    env.gen.count = 30;
    env.run();
  end
  initial begin
    $dumpfile("dump.vcd");
    $dumpvars;
  end
endmodule
```



DAY-9

#100DAYSRSL

“Aim”:- To Design the 4 bit Adder-Subtractor using Verilog

“Verilog Code”:-

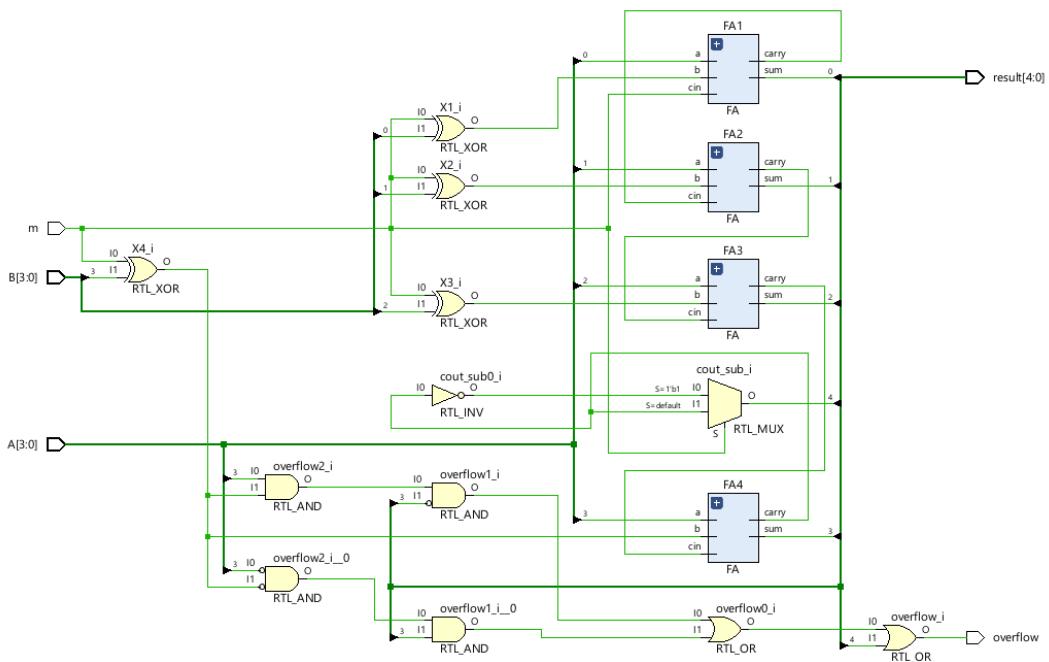
```

module FA(input a, b, cin, output sum, carry);
  assign {carry, sum} = a + b + cin;
endmodule

module A_S(
  input [3:0] A, B,
  input m,
  output [4:0] result,
  output overflow
);
  wire [3:0] Y;
  wire [3:0] C;
  wire [3:0] S;
  wire cout_sub;
  xor X1(Y[0],m,B[0]);
  xor X2(Y[1],m,B[1]);
  xor X3(Y[2],m,B[2]);
  xor X4(Y[3],m,B[3]);
  FA FA1(A[0], Y[0], m, S[0], C[0]);
  FA FA2(A[1], Y[1], C[0], S[1], C[1]);
  FA FA3(A[2], Y[2], C[1], S[2], C[2]);
  FA FA4(A[3], Y[3], C[2], S[3], C[3]);
  assign overflow = ((A[3] & Y[3] & ~S[3]) | (~A[3] & ~Y[3] & S[3])) | cout_sub;
  assign cout_sub = (m == 1'b1) ? ~C[3] : C[3];
  assign result = {cout_sub, S};
endmodule

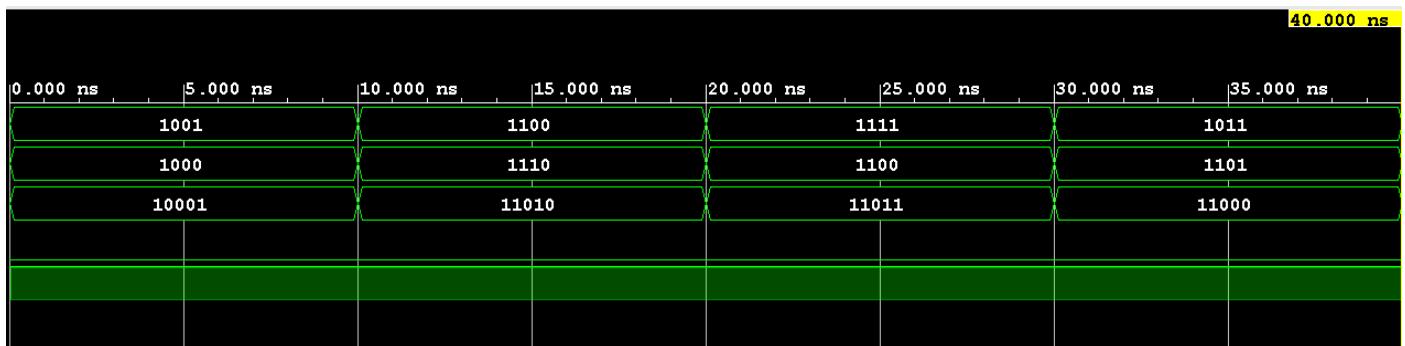
```

“Schematics”:-

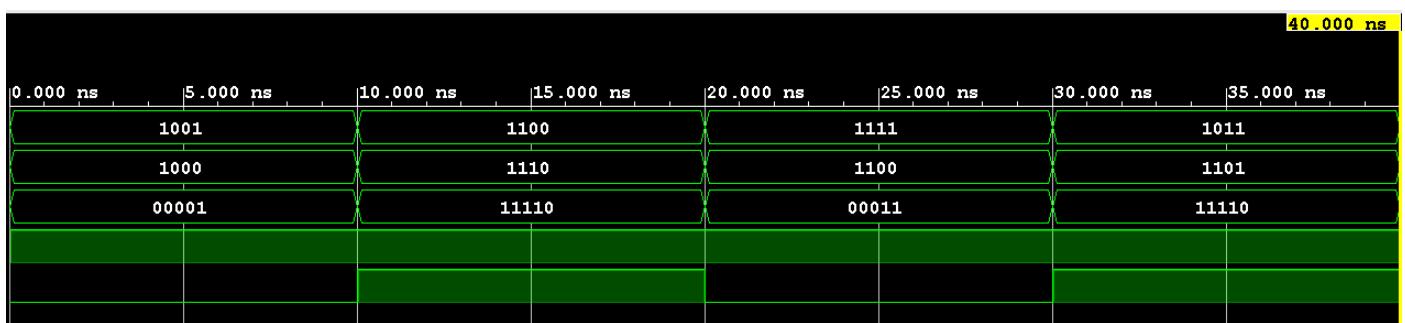


“Waveforms”:-

For Addition



For Subtraction





DAY-11

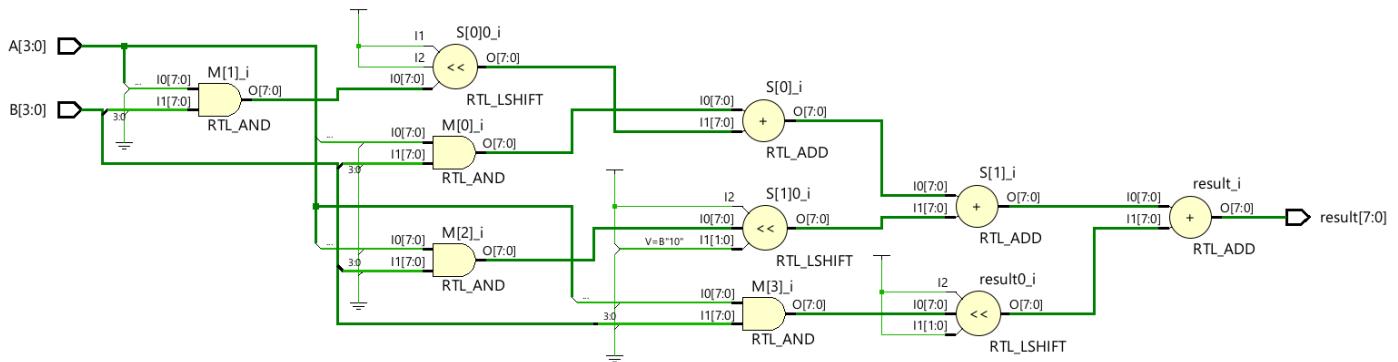
#100DAYSRSL

“Aim”:- To Design the 4 bit multiplier using Verilog

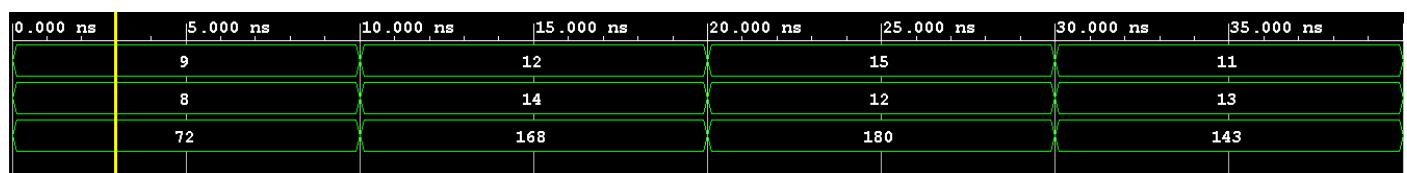
“Verilog Code”:-

```
module Multiplier_4bit(
  input [3:0] A,B,
  output [7:0] result
);
  wire [7:0] M [3:0];
  wire [7:0] S [1:0];
  assign M[0]=({4{A[0]}})&B[3:0];
  assign M[1]=({4{A[1]}})&B[3:0];
  assign M[2]=({4{A[2]}})&B[3:0];
  assign M[3]=({4{A[3]}})&B[3:0];
  assign S[0]=(M[0])+(M[1]<<1);
  assign S[1]=(S[0])+(M[2]<<2);
  assign result=(S[1])+(M[3]<<3);
endmodule
```

“Schematics”:-



“Waveforms”:-





DAY-12

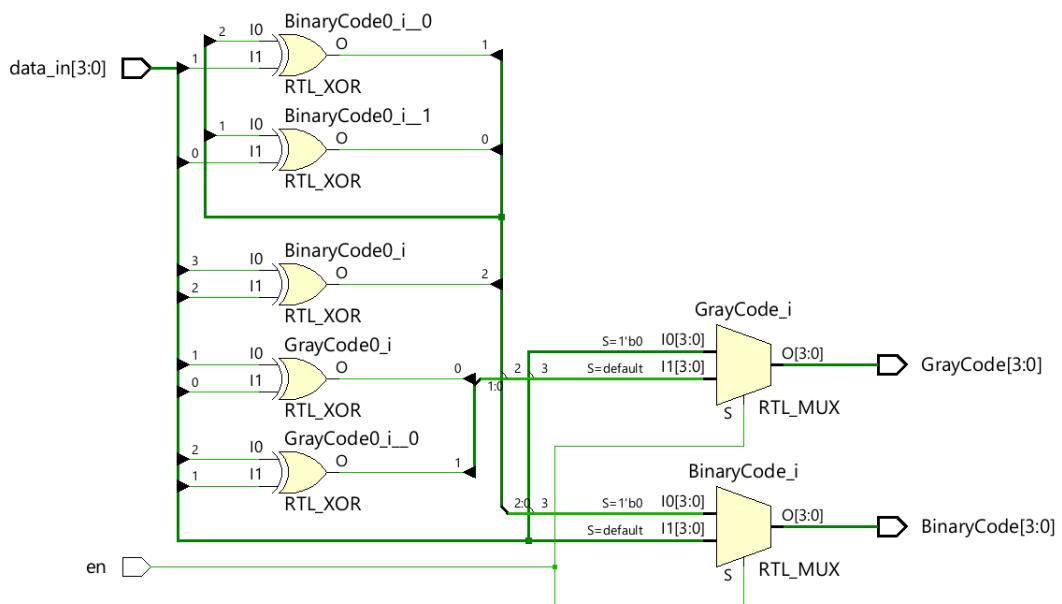
#100DAYSRSL

“Aim”:- To Design the 4bit Gray to Binary converter and Binary to Gray converter Verilog

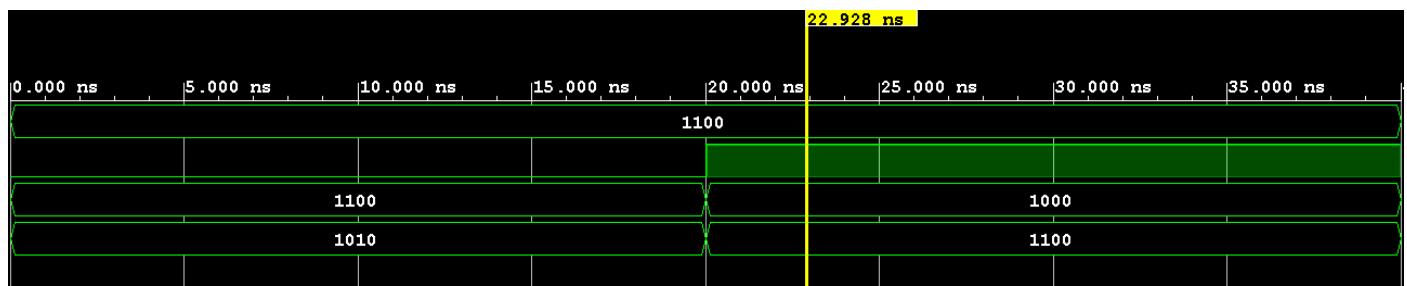
“Verilog Code”:-

```
module B_G(
    input [3:0] data_in,
    input en,
    output reg [3:0] BinaryCode,GrayCode
);
    always @(*) begin
        if(en)begin //Gray to Binary Converter
            BinaryCode[3]=data_in[3];
            BinaryCode[2]=BinaryCode[3]^data_in[2];
            BinaryCode[1]=BinaryCode[2]^data_in[1];
            BinaryCode[0]=BinaryCode[1]^data_in[0];
            GrayCode=data_in;
        end
        else begin //Binary to Gray converter
            GrayCode[3]=data_in[3];
            GrayCode[2]=data_in[3]^data_in[2];
            GrayCode[1]=data_in[2]^data_in[1];
            GrayCode[0]=data_in[1]^data_in[0];
            BinaryCode=data_in;
        end
    end
endmodule
```

“Schematics”:-



“Waveforms”:-





DAY-13

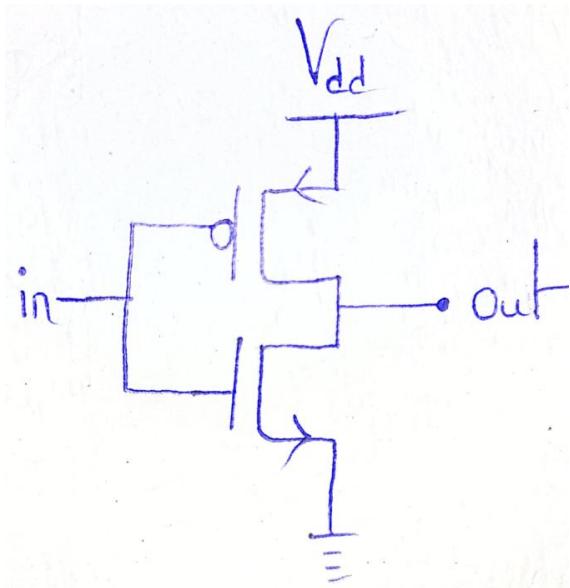
#100DAYSRSL

“Aim”:- To Design the CMOS inverter using Verilog

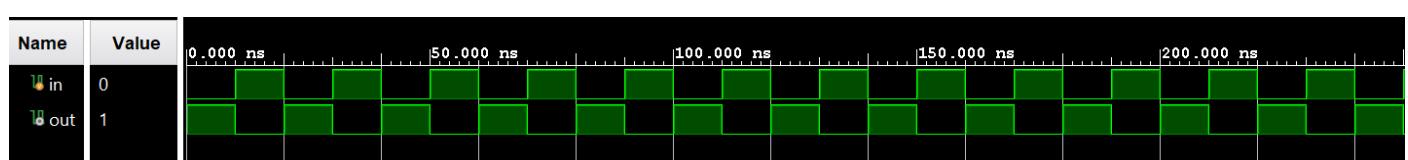
“Verilog Code”:-

```
module cmos_inverter(
  input in,
  output out
);
  supply0 gnd;
  supply1 vdd;
  //pmos (drain,source,gate);
  pmos (out, vdd, in);
  //nmos (drain,source,gate);
  nmos (out, gnd, in);
endmodule
```

“Schematics”:-



“Waveforms”:-





DAY-14

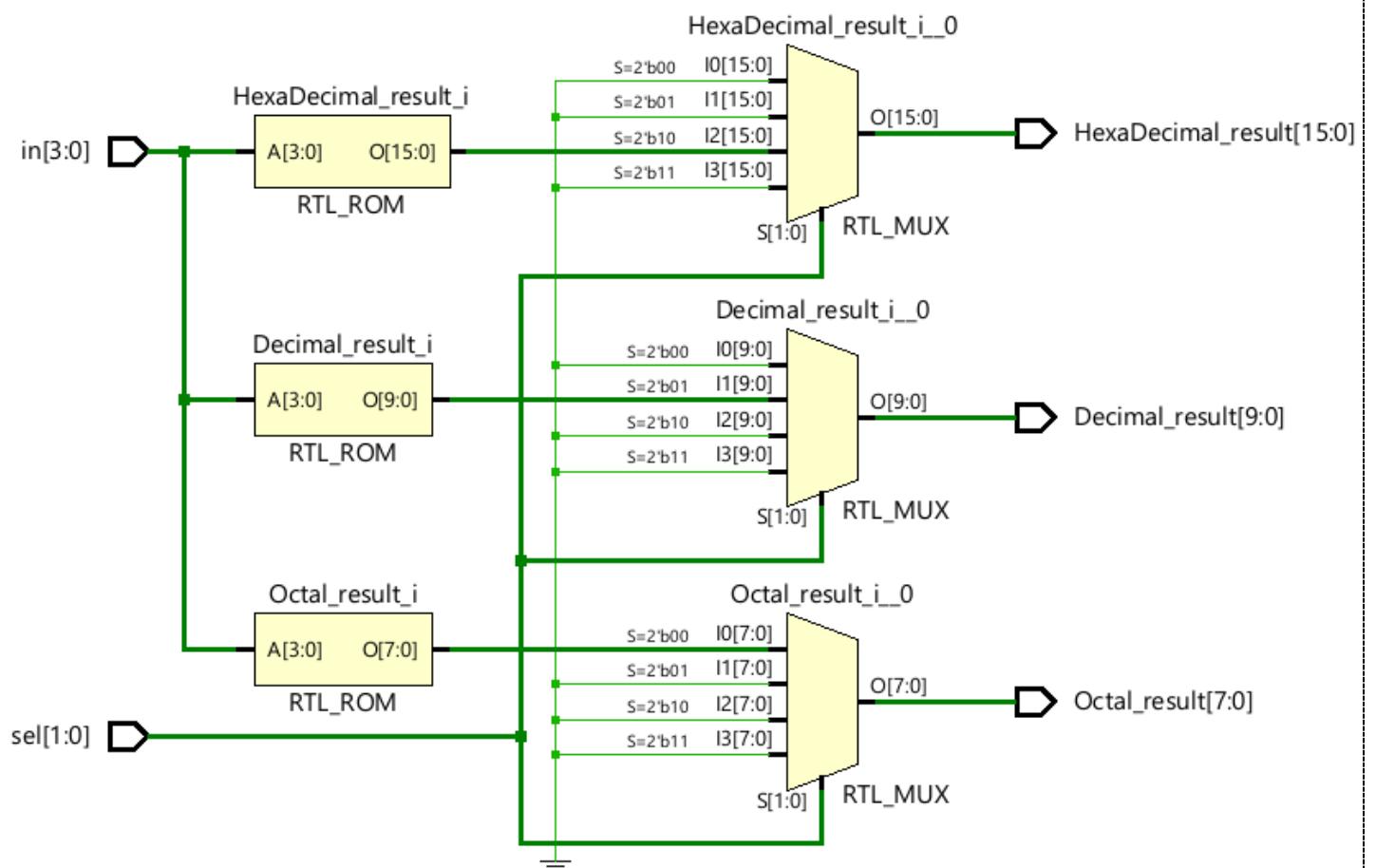
#100DAYSRTL

“Aim”:- To Design the Decoder which converts the binary value to octal, decimal and Hexadecimal

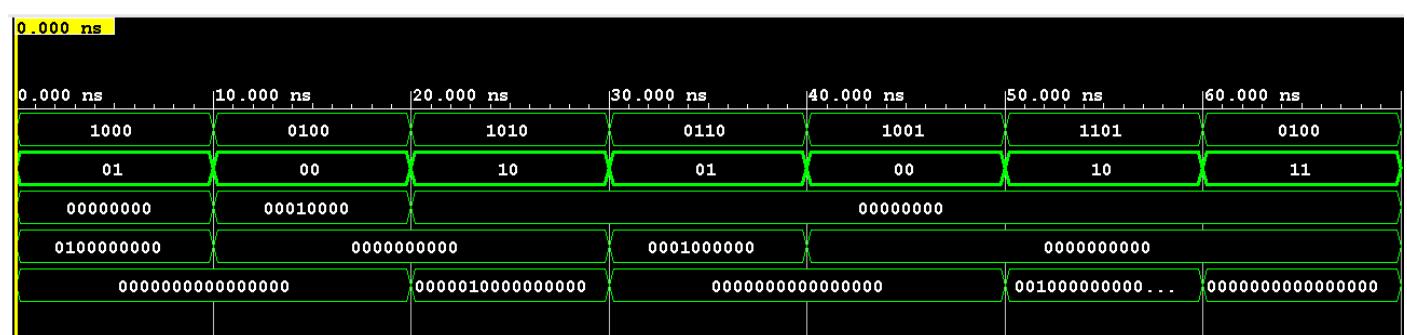
“Verilog Code”:-

```
module Decoder(
    input [3:0] in,
    input [1:0] sel,
    output reg [7:0] Octal_result,
    output reg [9:0] Decimal_result,
    output reg [15:0] HexaDecimal_result);
    always @(*) begin
        case(sel)
            2'b00: begin // Binary to Octal conversion
                Decimal_result=0;
                HexaDecimal_result=0;
                case(in)
                    4'd0: Octal_result = {7'b0, 1};
                    4'd1: Octal_result = {6'b0, 1, 0};
                    4'd2: Octal_result = {5'b0, 1, 2'b0};
                    4'd3: Octal_result = {4'b0, 1, 3'b0};
                    4'd4: Octal_result = {3'b0, 1, 4'b0};
                    4'd5: Octal_result = {2'b0, 1, 5'b0};
                    4'd6: Octal_result = {0,1 ,6'b0};
                    4'd7: Octal_result = {1, 7'b0};
                    default: Octal_result = 8'b0;
                endcase
            end
            2'b01: begin //Binary to decimal
                HexaDecimal_result=0;
                Octal_result=0;
                case(in)
                    4'd0: Decimal_result = {9'b0, 1};
                    4'd1: Decimal_result = {8'b0, 1, 0};
                    4'd2: Decimal_result = {7'b0, 1, 2'b0};
                    4'd3: Decimal_result = {6'b0, 1, 3'b0};
                    4'd4: Decimal_result = {5'b0, 1, 4'b0};
                    4'd5: Decimal_result = {4'b0, 1, 5'b0};
                    4'd6: Decimal_result = {3'b0, 1, 6'b0};
                    4'd7: Decimal_result = {2'b0, 1, 7'b0};
                    4'd8: Decimal_result = {1'b0, 1, 8'b0};
                    4'd9: Decimal_result = {1, 9'b0};
                    default: Decimal_result = 10'b0;
                endcase
            end
            2'b10: begin //binary to hexadecimal
                Decimal_result=0;
                Octal_result=0;
                case(in)
                    4'd0: HexaDecimal_result = {15'b0, 1};
                    4'd1: HexaDecimal_result = {14'b0, 1, 0};
                    4'd2: HexaDecimal_result = {13'b0, 1, 2'b0};
                    4'd3: HexaDecimal_result = {12'b0, 1, 3'b0};
                    4'd4: HexaDecimal_result = {11'b0, 1, 4'b0};
                    4'd5: HexaDecimal_result = {10'b0, 1, 5'b0};
                    4'd6: HexaDecimal_result = {9'b0, 1, 6'b0};
                    4'd7: HexaDecimal_result = {8'b0, 1, 7'b0};
                    4'd8: HexaDecimal_result = {7'b0, 1, 8'b0};
                    4'd9: HexaDecimal_result = {6'b0, 1, 9'b0};
                    4'd10: HexaDecimal_result = {5'b0, 1, 10'b0};
                    4'd11: HexaDecimal_result = {4'b0, 1, 11'b0};
                    4'd12: HexaDecimal_result = {3'b0, 1, 12'b0};
                    4'd13: HexaDecimal_result = {2'b0, 1, 13'b0};
                    4'd14: HexaDecimal_result = {1'b0,1,14'b0};
                    4'd15: HexaDecimal_result = {1, 15'b0};
                    default: HexaDecimal_result = 16'b0;
                endcase
            end
            2'b11: begin
                // No operation
            end
        endcase
    end
endmodule
```

“Schematics”:-



“Waveforms”:-





DAY-15

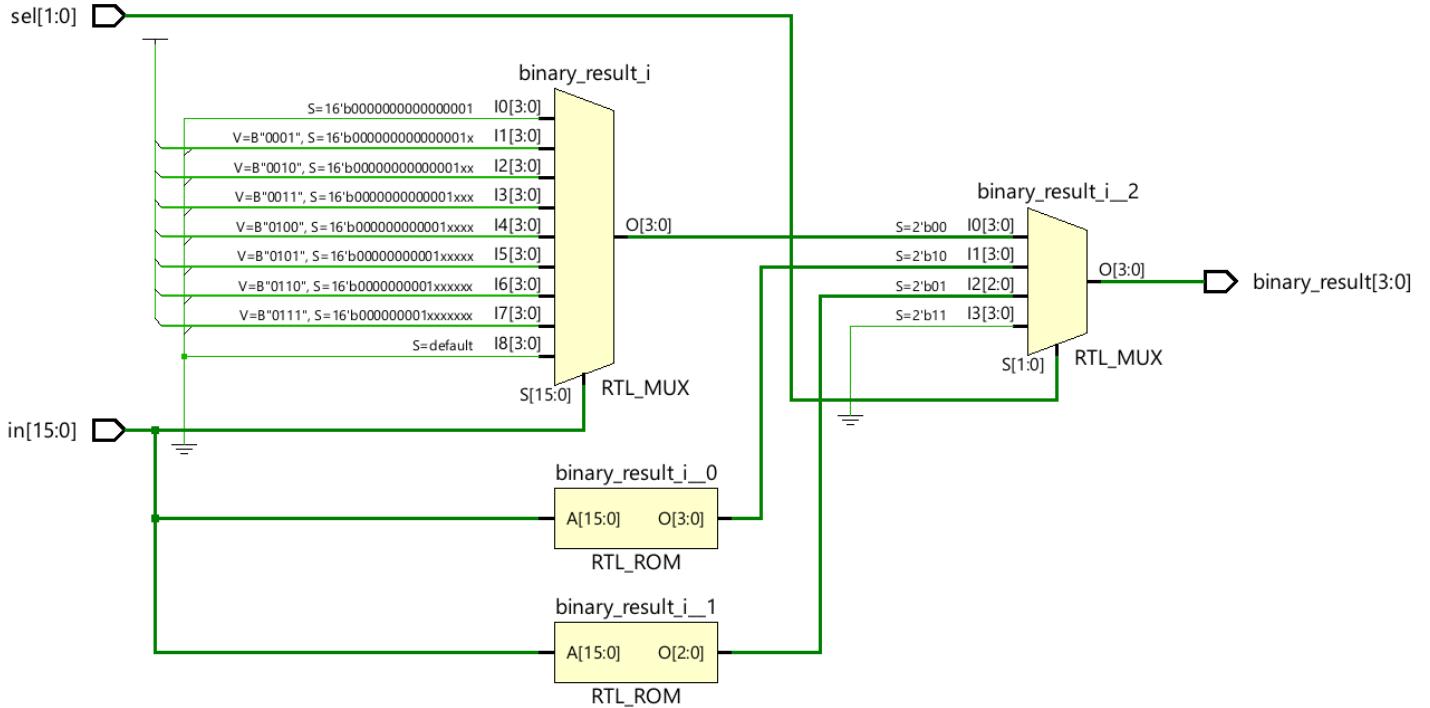
#100DAYSRTL

“Aim”:- To Design the Encoder that converts Decimal, Octal, and Hexadecimal into binary values.

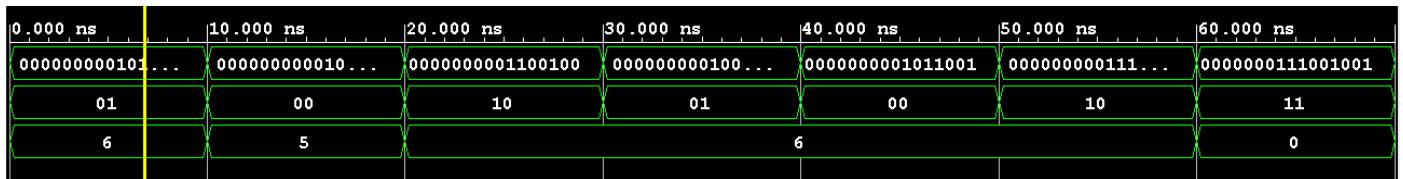
“Verilog Code”:-

```
module Encoder(
    input [15:0] in,
    input [1:0] sel,
    output reg [3:0] binary_result
);
    always @(*) begin
        case(sel)
            2'b00:begin //Octal to binary
                binary_result[3]=0;
                casex(in)
                    16'b0000000000000001:binary_result[2:0]=3'b000;
                    16'b0000000000000001x:binary_result[2:0]=3'b001;
                    16'b0000000000000001xx:binary_result[2:0]=3'b010;
                    16'b0000000000000001xxx:binary_result[2:0]=3'b011;
                    16'b0000000000000001xxxx:binary_result[2:0]=3'b100;
                    16'b0000000000000001xxxxx:binary_result[2:0]=3'b101;
                    16'b0000000000000001xxxxxx:binary_result[2:0]=3'b110;
                    16'b0000000000000001xxxxxxxx:binary_result[2:0]=3'b111;
                    default : binary_result=4'b0000;
                endcase
            end
            2'b10:begin
                casex(in) //HexaDecimal to binary
                    16'b0000000000000001:binary_result=4'b0000;
                    16'b0000000000000001x:binary_result=4'b0001;
                    16'b0000000000000001xx:binary_result=4'b0010;
                    16'b0000000000000001xxx:binary_result=4'b0011;
                    16'b0000000000000001xxxx:binary_result=4'b0100;
                    16'b0000000000000001xxxxx:binary_result=4'b0101;
                    16'b0000000000000001xxxxxx:binary_result=4'b0110;
                    16'b0000000000000001xxxxxxxx:binary_result=4'b0111;
                    16'b000000001xxxxxxxxxx:binary_result=4'b1000;
                    16'b0000001xxxxxxxxxxxx:binary_result=4'b1001;
                    16'b000001xxxxxxxxxxxxx:binary_result=4'b1010;
                    16'b00001xxxxxxxxxxxxxx:binary_result=4'b1011;
                    16'b0001xxxxxxxxxxxxxxx:binary_result=4'b1100;
                    16'b001xxxxxxxxxxxxxxx:binary_result=4'b1101;
                    16'b01xxxxxxxxxxxxxxx:binary_result=4'b1110;
                    16'b1xxxxxxxxxxxxxxx:binary_result=4'b1111;
                    default : binary_result=4'b0000;
                endcase
            end
            2'b01: begin
                casex(in) //Decimal to binary
                    16'b0000000000000001:binary_result=4'b0000;
                    16'b0000000000000001x:binary_result=4'b0001;
                    16'b0000000000000001xx:binary_result=4'b0010;
                    16'b0000000000000001xxx:binary_result=4'b0011;
                    16'b0000000000000001xxxx:binary_result=4'b0100;
                    16'b0000000000000001xxxxx:binary_result=4'b0101;
                    16'b0000000000000001xxxxxx:binary_result=4'b0110;
                    16'b0000000000000001xxxxxxxx:binary_result=4'b0111;
                    16'b0000000000000001xxxxxxxxx:binary_result=4'b1000;
                    16'b0000000000000001xxxxxxxxxx:binary_result=4'b1001;
                    default : binary_result=4'b0000;
                endcase
            end
            2'b11:binary_result=0;
        endcase
    end
endmodule
```

“Schematics”:-



“Waveforms”:-



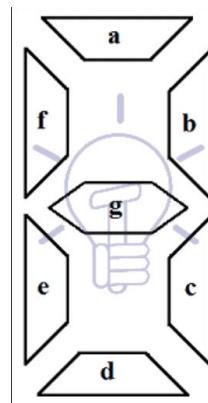


DAY-16

#100DAYSRSL

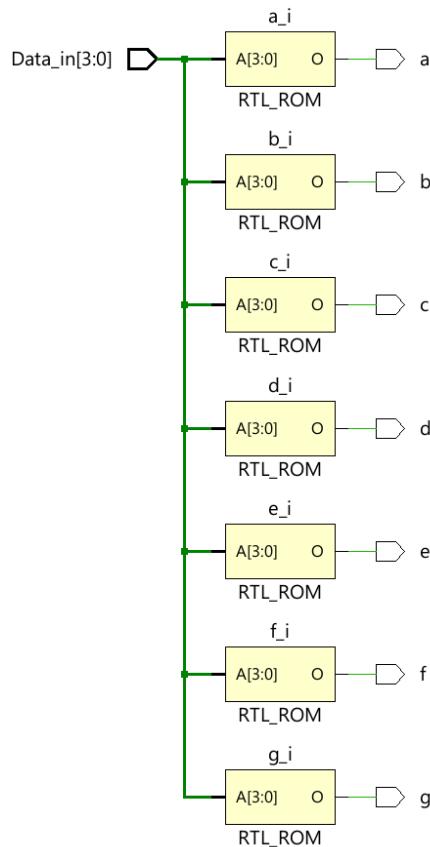
“Aim”:- To Design BCD to a seven-segment display Decoder

“Verilog Code”:-

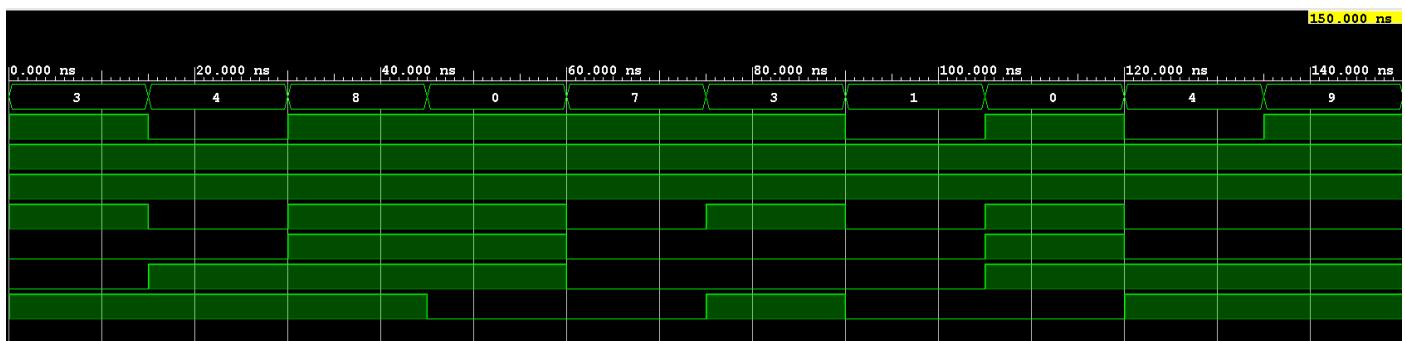


```
module BCD_SevenSegment(input [3:0] Data_in,
output reg a,b,c,d,e,f,g);
always @(*) begin
case(Data_in)
4'd0:begin a=1;b=1;c=1;d=1;e=1;f=1;g=0;end
4'd1:begin a=0;b=1;c=1;d=0;e=0;f=0;g=0;end
4'd2:begin a=1;b=1;c=0;d=1;e=1;f=0;g=1;end
4'd3:begin a=1;b=1;c=1;d=1;e=0;f=0;g=1;end
4'd4:begin a=0;b=1;c=1;d=0;e=0;f=1;g=1;end
4'd5:begin a=1;b=0;c=1;d=1;e=0;f=1;g=1;end
4'd6:begin a=0;b=0;c=1;d=1;e=1;f=1;g=1;end
4'd7:begin a=1;b=1;c=1;d=0;e=0;f=0;g=0;end
4'd8:begin a=1;b=1;c=1;d=1;e=1;f=1;g=1;end
4'd9:begin a=1;b=1;c=1;d=0;e=0;f=1;g=1;end
default:begin a=0;b=0;c=0;d=0;e=0;f=0;g=0;
end
endcase
end
endmodule
```

“Schematics”:-



“Waveforms”:-



“Console Results”:-

```
data_in= 3,a=1,b=1,c=1,d=1,e=0,f=0,g=1
data_in= 4,a=0,b=1,c=1,d=0,e=0,f=1,g=1
data_in= 8,a=1,b=1,c=1,d=1,e=1,f=1,g=1
data_in= 0,a=1,b=1,c=1,d=1,e=1,f=1,g=0
data_in= 7,a=1,b=1,c=1,d=0,e=0,f=0,g=0
data_in= 3,a=1,b=1,c=1,d=1,e=0,f=0,g=1
data_in= 1,a=0,b=1,c=1,d=0,e=0,f=0,g=0
data_in= 0,a=1,b=1,c=1,d=1,e=1,f=1,g=0
data_in= 4,a=0,b=1,c=1,d=0,e=0,f=1,g=1
data_in= 9,a=1,b=1,c=1,d=0,e=0,f=1,g=1
```

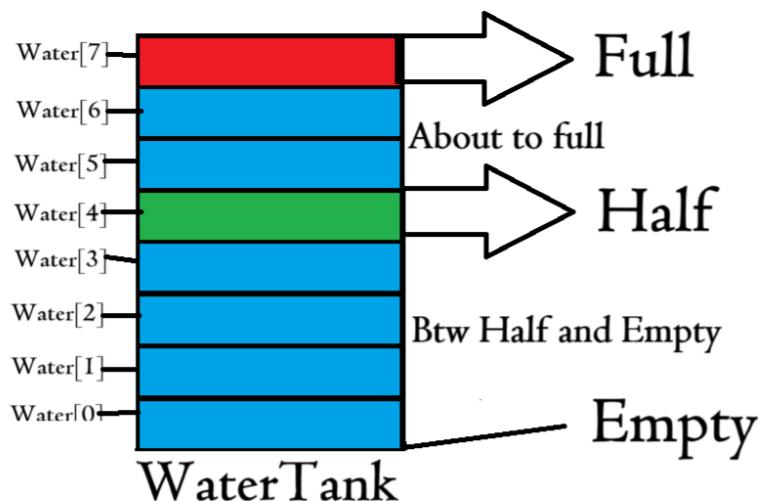


DAY-17

#100DAYSRSL

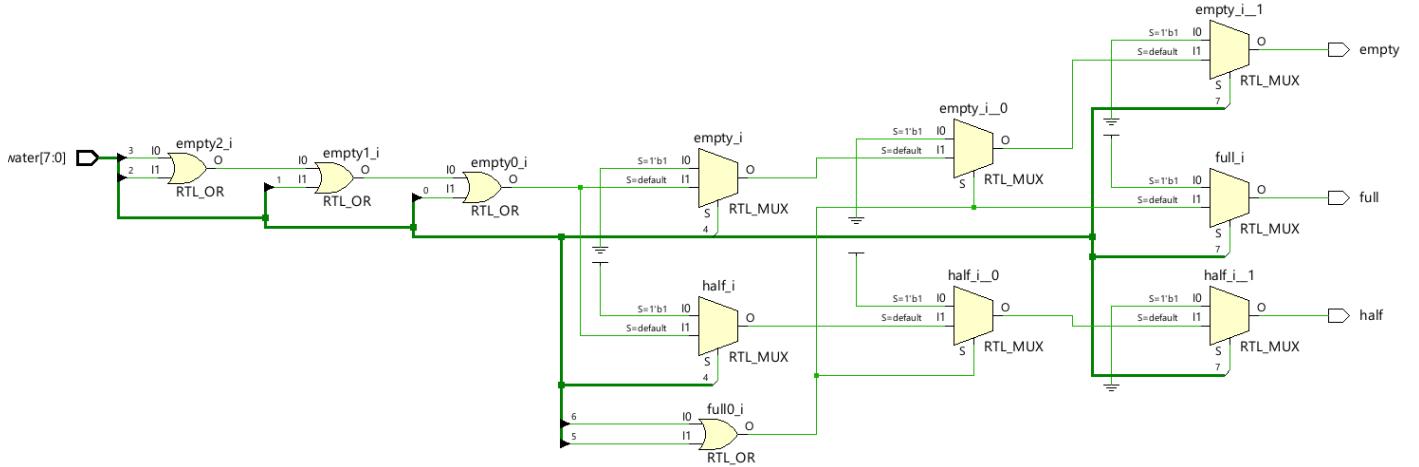
“Aim”:- To Design the water level indicator using 8X3 encoder

“Verilog Code”:-

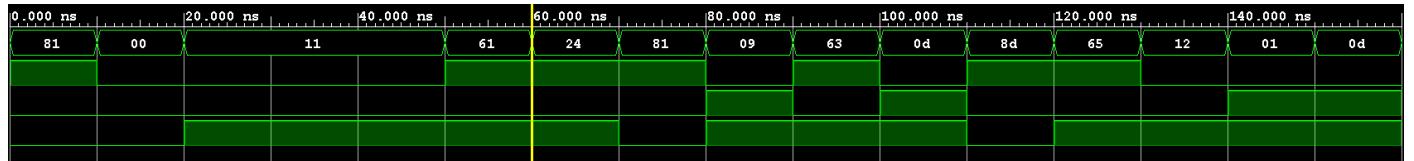


```
| module TankLevelIndicator(input [7:0] water,
|   output reg full,empty,half);
|   always @(*) begin
|     if(water[7]) begin // Tank level is Full
|       full=1;empty=0;half=0;
|     end
|     else if (water[6]|water[5]) begin // Tank level is about to full
|       full=1;empty=0;half=1;
|     end
|     else if (water[4]) begin // Tank is half
|       full=0;empty=0;half=1;
|     end
|     else if (water[3]|water[2]|water[1]|water[0]) begin // Tank level is between Half or Empty
|       full=0;empty=1;half=1;
|     end
|     else begin //Tank level is empty
|       full=0;empty=0;half=0;
|     end
|   end
| endmodule
```

“Schematics”:-



“Waveforms”:-



“Console Results”:-

```
water=10000001,full=1,empty=0,half=0
water=00000000,full=0,empty=0,half=0
water=00010001,full=0,empty=0,half=1
water=01100001,full=1,empty=0,half=1
water=00100100,full=1,empty=0,half=1
water=10000001,full=1,empty=0,half=0
water=00001001,full=0,empty=1,half=1
water=01100011,full=1,empty=0,half=1
water=00001101,full=0,empty=1,half=1
water=10001101,full=1,empty=0,half=0
water=01100101,full=1,empty=0,half=1
water=00010010,full=0,empty=0,half=1
water=00000001,full=0,empty=1,half=1
water=00001101,full=0,empty=1,half=1
```



DAY-18

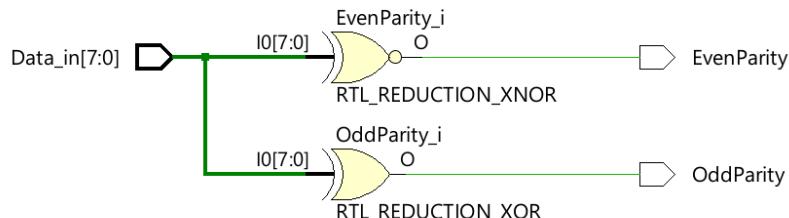
#100DAYSRSL

“Aim”:- To Design the Parity Checker

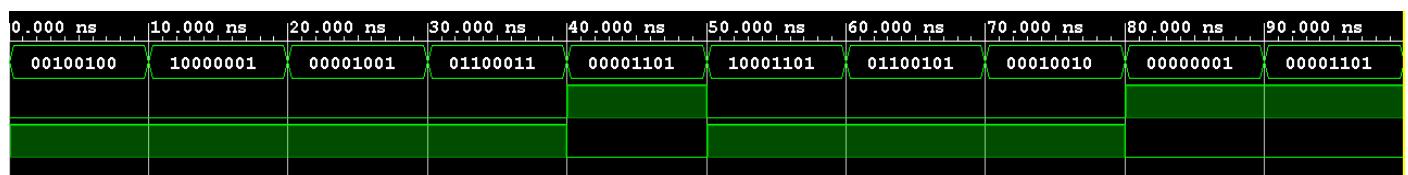
“Verilog Code”:-

```
module ParityChecker(
    input [7:0] Data_in,
    output EvenParity, OddParity
);
    assign OddParity=~(Data_in);
    assign EvenParity=~^(Data_in);
endmodule
```

“Schematics”:-



“Waveforms”:-



“Console Results”:-

```
Data_in=00100100,EvenParity=1,OddParity=0
Data_in=10000001,EvenParity=1,OddParity=0
Data_in=00001001,EvenParity=1,OddParity=0
Data_in=01100011,EvenParity=1,OddParity=0
Data_in=00001101,EvenParity=0,OddParity=1
Data_in=10001101,EvenParity=1,OddParity=0
Data_in=01100101,EvenParity=1,OddParity=0
Data_in=00010010,EvenParity=1,OddParity=0
Data_in=00000001,EvenParity=0,OddParity=1
Data_in=00001101,EvenParity=0,OddParity=1
```



DAY-19

#100DAYSRDL

“Aim”:- To Design the Binary Digit Counter Which Counts the occurrence of ones and zeros.

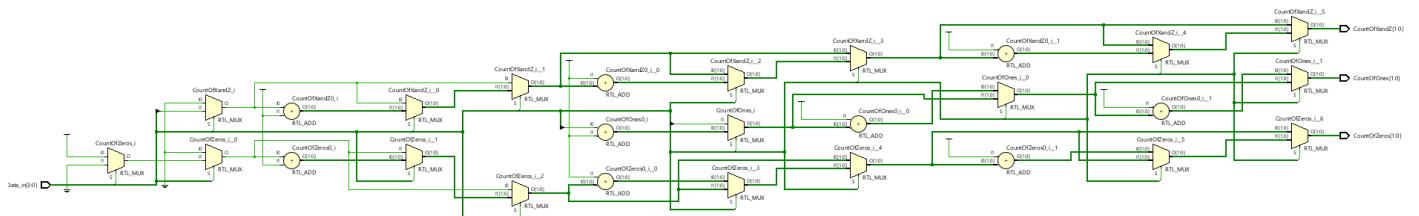
“Design Code”:-

```

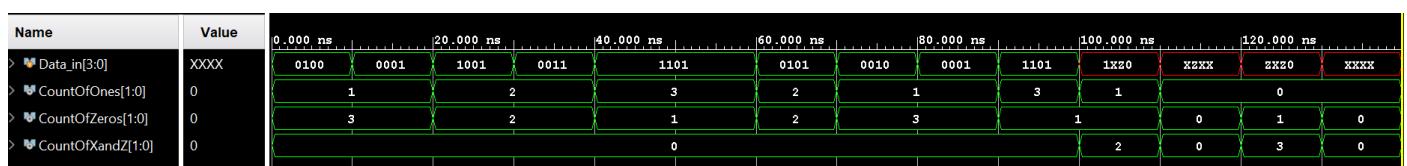
module BinaryDigitCounter(
    input [3:0] Data_in,
    output reg [1:0] Countofones,
    output reg [1:0] CountOfZeros,
    output reg [1:0] CountOfXandZ
);
    always @(*) begin
        CountOfOnes = 0;
        CountOfZeros = 0;
        CountOfXandZ=0;
        for (int i = 0; i < 4; i = i + 1) begin
            if (Data_in[i]) // To count the occurrence of ones
                CountOfOnes = CountOfOnes + 1;
            else if (~Data_in[i]) // To count the occurrence of zeros
                CountOfZeros = CountOfZeros + 1;
            else if (Data_in[i]==1'bX) // To count the occurrence of Unknowns
                CountOfXandZ=CountOfXandZ+1;
        end
    end
endmodule

```

“Schematics”:-



“Waveforms”:-





DAY-20

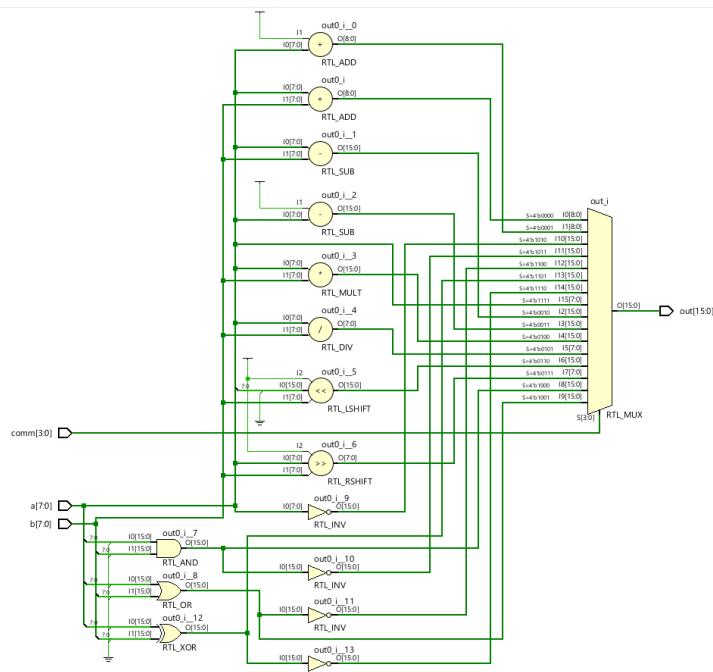
#100DAYSRSL

“Aim”:- To Design the Arithmetic Logic unit (ALU) which performs 16 operations

“Design Code”:-

```
module ALU(input [7:0] a,b,input [3:0] comm,output reg [15:0] out);
parameter ADD=4'b0000,INC=4'b0001,SUB=4'b0010, DEC = 4'b0011, MUL = 4'b0100, DIV = 4'b0101, SHL = 4'b0110,SHR = 4'b0111, AND = 4'b1000;
parameter OROperation=4'b1001,INV= 4'b1010,NANDOperation=4'b1011,NOROperation=4'b1100,XOROperation=4'b1101,XNOROperation=4'b1110,BUFOperation=4'b1111;
always @(*) begin
  case(comm)
    4'b0000 : out = a + b;
    4'b0001 : out = a + 1;
    4'b0010 : out = a - b;
    4'b0011 : out = a - 1;
    4'b0100 : out = a * b;
    4'b0101 : out = a / b;
    4'b0110 : out = a << b;
    4'b0111 : out = a >> b;
    4'b1000 : out = a & b;
    4'b1001 : out = a | b;
    4'b1010 : out = ~a;
    4'b1011 : out = ~(a & b);
    4'b1100 : out = ~(a | b);
    4'b1101 : out = a ^ b;
    4'b1110 : out = ~(a ^ b);
    4'b1111 : out = a;
    default: out = 16'hzzzz;
  endcase
end
endmodule
```

“Schematics”:-



“Waveforms”:-

> a[7:0]	143	36	99	101	13	237	198	229	143	232	189
> b[7:0]	242	129	13	18	118	140	197	119	242	197	45
> comm[3:0]	14	9	13	1	13	9	10	2	14	12	5
> out[15:0]	65410	165	110	102	123	237	65337	110	65410	65298	4



DAY-21

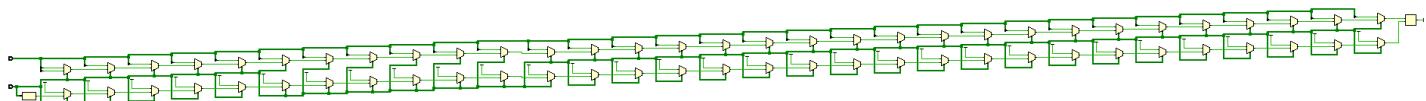
#100DAYSRTL

“Aim”:-To Design a MUX that acts as an Asynchronous PISO

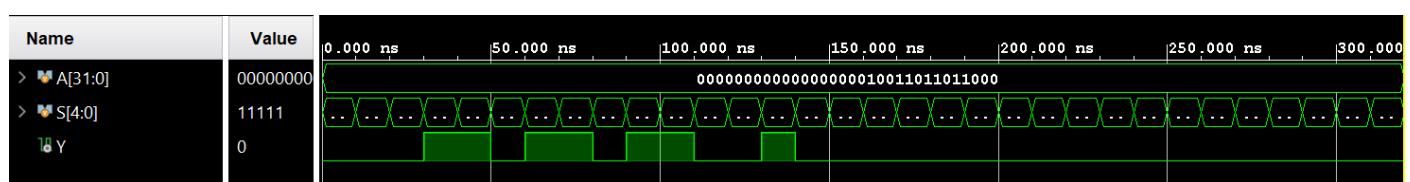
“Design Code”:-

```
module Mux #(parameter N = 32, M = 5) ( //M=log2 (M)
    input [N-1:0] A, //Parallel Data as input
    input [M-1:0] S,
    output reg Y // Serial Data as output
);
always @(*) begin
for(int i=0;i<N;i++) begin
case (S)
i:Y=A[i];
endcase
end
end
endmodule
```

“Schematics”:-



“Waveforms”:-



“Console”:-

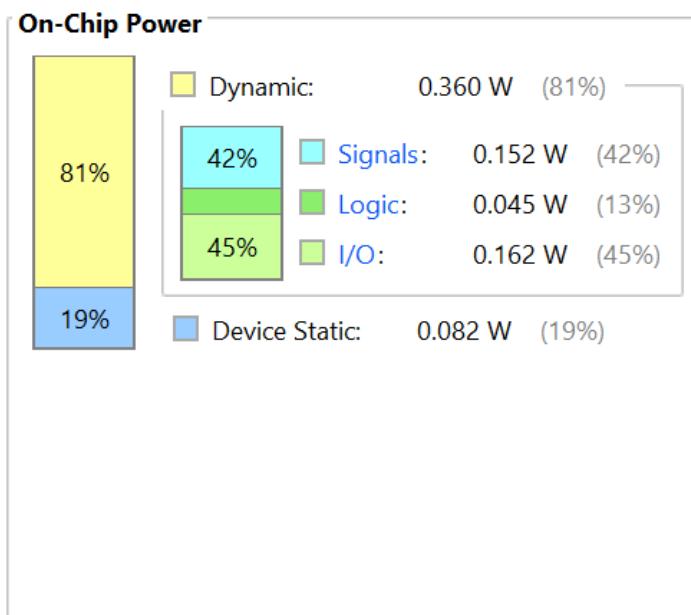
```
A=0000000000000000000000000000000010011011011000,S=00000,Y=0
A=0000000000000000000000000000000010011011011000,S=00001,Y=0
A=0000000000000000000000000000000010011011011000,S=00010,Y=0
A=0000000000000000000000000000000010011011011000,S=00011,Y=1
A=0000000000000000000000000000000010011011011000,S=00100,Y=1
A=0000000000000000000000000000000010011011011000,S=00101,Y=0
A=0000000000000000000000000000000010011011011000,S=00110,Y=1
A=0000000000000000000000000000000010011011011000,S=00111,Y=1
A=0000000000000000000000000000000010011011011000,S=01000,Y=0
A=0000000000000000000000000000000010011011011000,S=01001,Y=1
A=0000000000000000000000000000000010011011011000,S=01010,Y=1
A=0000000000000000000000000000000010011011011000,S=01011,Y=0
A=0000000000000000000000000000000010011011011000,S=01100,Y=0
A=0000000000000000000000000000000010011011011000,S=01101,Y=1
A=0000000000000000000000000000000010011011011000,S=01110,Y=0
A=0000000000000000000000000000000010011011011000,S=01111,Y=0
A=0000000000000000000000000000000010011011011000,S=10000,Y=0
A=0000000000000000000000000000000010011011011000,S=10001,Y=0
A=0000000000000000000000000000000010011011011000,S=10010,Y=0
A=0000000000000000000000000000000010011011011000,S=10011,Y=0
A=0000000000000000000000000000000010011011011000,S=10100,Y=0
A=0000000000000000000000000000000010011011011000,S=10101,Y=0
A=0000000000000000000000000000000010011011011000,S=10110,Y=0
A=0000000000000000000000000000000010011011011000,S=10111,Y=0
A=0000000000000000000000000000000010011011011000,S=11000,Y=0
A=0000000000000000000000000000000010011011011000,S=11001,Y=0
A=0000000000000000000000000000000010011011011000,S=11010,Y=0
A=0000000000000000000000000000000010011011011000,S=11011,Y=0
A=0000000000000000000000000000000010011011011000,S=11100,Y=0
A=0000000000000000000000000000000010011011011000,S=11101,Y=0
A=0000000000000000000000000000000010011011011000,S=11110,Y=0
A=0000000000000000000000000000000010011011011000,S=11111,Y=0
Completed
```

Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power: 0.441 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 25.8°C
Thermal Margin: 59.2°C (31.2 W)
Effective θ_{JA}: 1.9°C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity





DAY-22

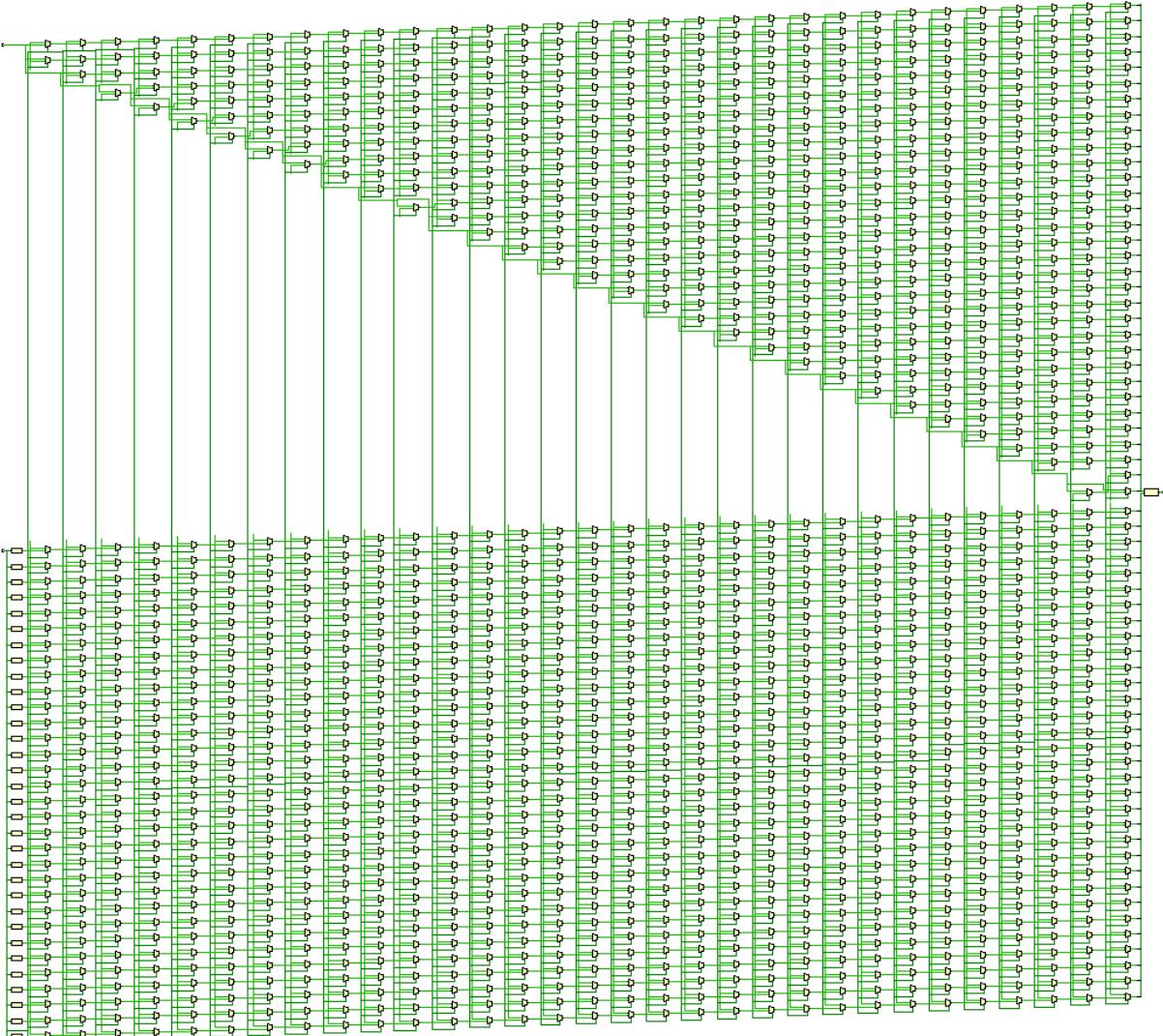
#100DAYSRSL

“Aim”:-To Design a DEMUX that acts as an Asynchronous SIPO using System Verilog Design style

“Design Code”:-

```
module Mux #(parameter N = 32, M = 5) ( //M=log2 (M)
    input A, //Parallel Data as input
    input [M-1:0] S,
    output reg [N-1:0] Y // Serial Data as output
);
always @(*) begin
for(int i=0;i<N;i++) begin
for (int j=0;j<N;j++) begin
case(S)
i:begin
    Y[j]=0;
    if(i==j) Y[i]=A;
end
endcase
end
end
end
endmodule
```

“Schematics”:-



“Console”:-

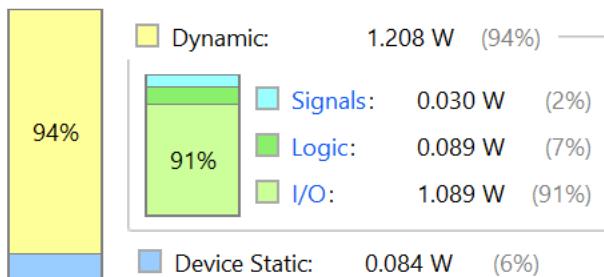
Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	1.292 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	27.4°C
Thermal Margin:	57.6°C (30.4 W)
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-23

#100DAYSRSL

“Aim”:- To Design the input majority circuit in Mixed modeling (Gate & Data Flow).

“Design Code”:-

A0	A1	A2	A3	A4	A5	Y
0	0	0	0	0	0	0
0	0	0	0	0	1	1
0	0	0	0	1	0	1
0	0	0	0	1	1	1
0	0	0	1	0	0	1
0	0	0	1	0	1	1
0	0	0	1	1	0	1
0	0	0	1	1	1	1
0	0	1	0	0	0	1
0	0	1	0	0	1	1
0	0	1	0	1	0	1
0	0	1	0	1	1	1
0	0	1	1	0	0	1
0	0	1	1	0	1	1
0	0	1	1	1	0	1
0	0	1	1	1	1	1
0	1	0	0	1	0	1
0	1	0	0	1	1	1
0	1	0	1	0	1	1
0	1	0	1	1	0	1
0	1	0	1	1	1	1
0	1	1	0	0	0	1
0	1	1	0	0	1	1
0	1	1	0	1	0	1
0	1	1	0	1	1	1
0	1	1	1	0	0	1
0	1	1	1	0	1	1
0	1	1	1	1	0	1
0	1	1	1	1	1	1

A0	A1	A2	A3	A4	A5	Y
0	0	1	0	0	0	1
0	0	1	0	0	1	1
0	0	1	0	1	0	1
0	0	1	0	1	1	1
0	0	1	1	0	0	1
0	0	1	1	0	1	1
0	0	1	1	1	0	1
0	0	1	1	1	1	1
0	1	0	1	0	1	1
0	1	0	1	1	0	1
0	1	0	1	1	1	1
0	1	1	0	0	0	1
0	1	1	0	0	1	1
0	1	1	0	1	0	1
0	1	1	0	1	1	1
0	1	1	1	0	0	1
0	1	1	1	0	1	1
0	1	1	1	1	0	1
0	1	1	1	1	1	1

A0	A1	A2	A3	A4	A5	Y
0	1	1	1	0	0	1
0	1	1	1	0	1	1
0	1	1	1	1	0	1
0	1	1	1	1	1	1
0	1	0	0	0	0	1
0	1	0	0	0	1	1
0	1	0	0	1	0	1
0	1	0	0	1	1	1
0	1	1	0	0	0	1
0	1	1	0	0	1	1
0	1	1	0	1	0	1
0	1	1	0	1	1	1
0	1	1	1	0	0	1
0	1	1	1	0	1	1
0	1	1	1	1	0	1
0	1	1	1	1	1	1

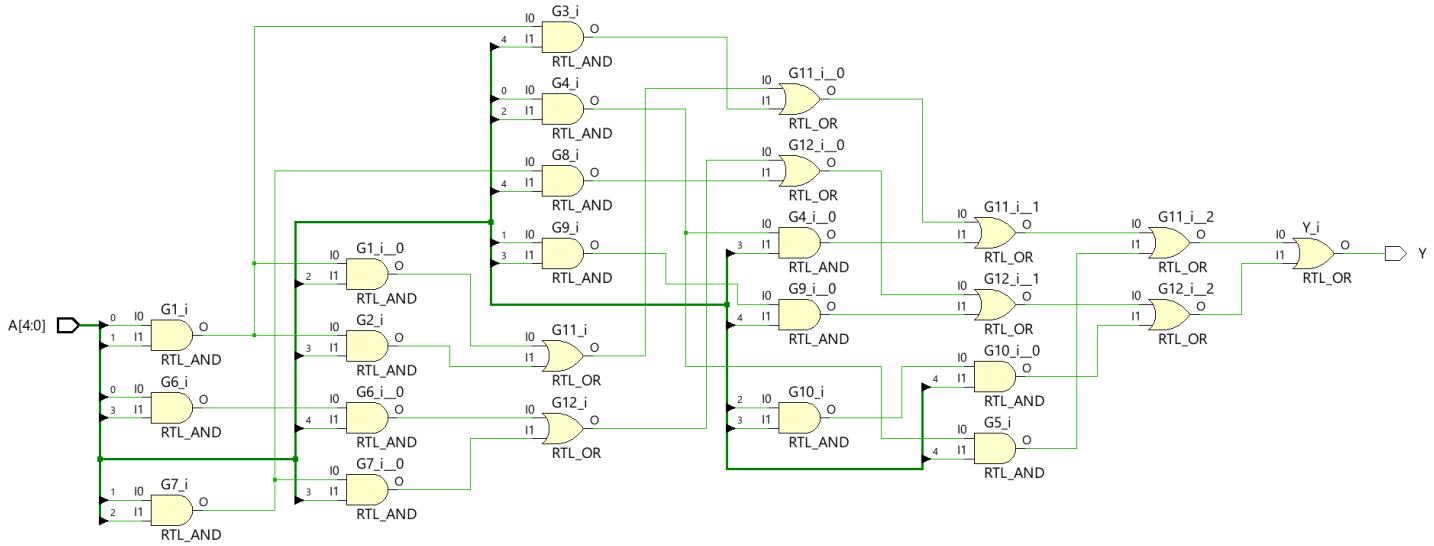
A0	A1	A2	A3	A4	A5	Y
0	1	1	1	0	0	1
0	1	1	1	0	1	1
0	1	1	1	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	0	1
1	0	0	0	0	1	1
1	0	0	0	1	0	1
1	0	0	0	1	1	1
1	0	0	1	0	0	1
1	0	0	1	0	1	1
1	0	0	1	1	0	1
1	0	0	1	1	1	1
1	0	1	0	0	0	1
1	0	1	0	0	1	1
1	0	1	0	1	0	1
1	0	1	0	1	1	1
1	0	1	1	0	0	1
1	0	1	1	0	1	1
1	0	1	1	1	0	1
1	0	1	1	1	1	1

The min-terms notation for the 5-input majority function is:

$\Sigma(1, 3, 5, 7, 9, 11, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31)$

```
module MajorityInputCircuit(
  input [4:0] A,
  output Y
);
  wire [11:0]W;
  and G1(W[0],A[0],A[1],A[2]);
  and G2(W[1],A[0],A[1],A[3]);
  and G3(W[2],A[0],A[1],A[4]);
  and G4(W[3],A[0],A[2],A[3]);
  and G5(W[4],A[0],A[2],A[4]);
  and G6(W[5],A[0],A[3],A[4]);
  and G7(W[6],A[1],A[2],A[3]);
  and G8(W[7],A[1],A[2],A[4]);
  and G9(W[8],A[1],A[3],A[4]);
  and G10(W[9],A[2],A[3],A[4]);
  or G11(W[10],W[0],W[1],W[2],W[3],W[4]);
  or G12(W[11],W[5],W[6],W[7],W[8],W[9]);
  assign Y=W[10] | W[11];
endmodule
```

“Schematics”:-



“Console”:-

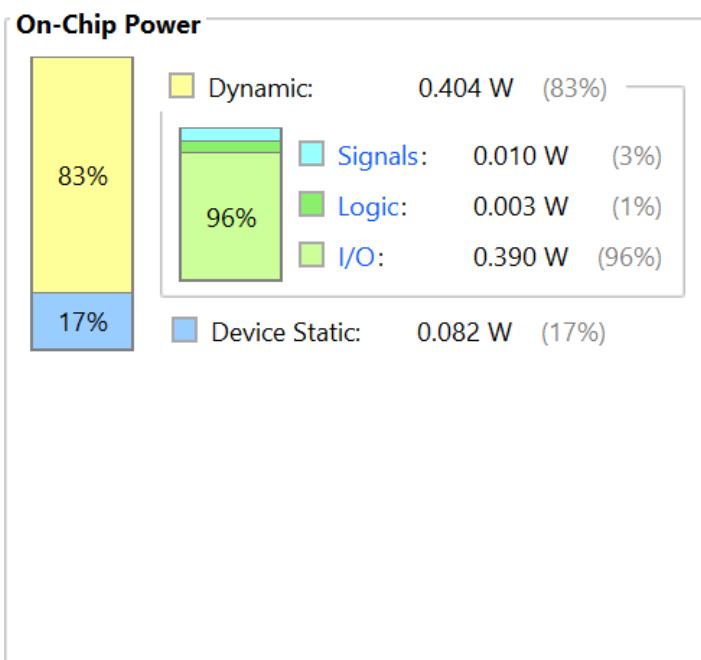
```
A=00100, Y=0
A=00001, Y=0
A=01001, Y=0
A=00011, Y=0
A=01101, Y=1
A=01101, Y=1
A=00101, Y=0
A=10010, Y=0
A=00001, Y=0
```

Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	0.486 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	25.9°C
Thermal Margin:	59.1°C (31.2 W)
Effective θ _{JA} :	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity



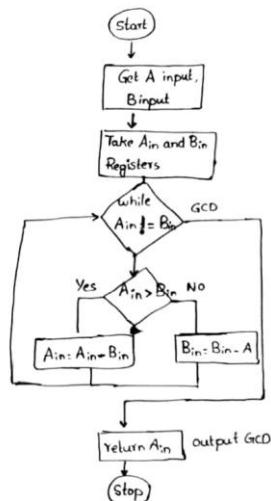


DAY-24

#100DAYSRTL

“Aim”:- To Design the Circuit which gives HCF(Highest common factor) of two numbers using System Verilog.

“Design Code”:-



```
module HCF(
    input [3:0] A,B,
    output [3:0] Y
);
    reg [3:0] Ain,Bin;
    always @(*) begin
        Ain=A;Bin=B;
        while (A!=B) begin
            if(Ain<Bin) Bin=Bin-Ain;
            else Ain=Ain-Bin;
        end
    end
    assign Y=Ain;
endmodule
```

“Waveforms”:-

Name	Value	0.000 ns	10.000 ns	20.000 ns	30.000 ns	40.000 ns	50.000 ns	60.000 ns	70.000 ns	80.000 ns	90.000 ns
> A[3:0]	5	4	9	d	5	1	6	d	9		5
> B[3:0]	7	1	3	d	2	d	c	6	a	7	
> Y[3:0]	1	1	3	d		1		3	5	1	

“Console”:-

```
A= 4,B= 1,Y= 1
A= 9,B= 3,Y= 3
A=13,B=13,Y=13
A= 5,B= 2,Y= 1
A= 1,B=13,Y= 1
A= 6,B=13,Y= 1
A=13,B=12,Y= 1
A= 9,B= 6,Y= 3
A= 5,B=10,Y= 5
A= 5|B= 7,Y= 1
```



DAY-25

#100DAYSRSL

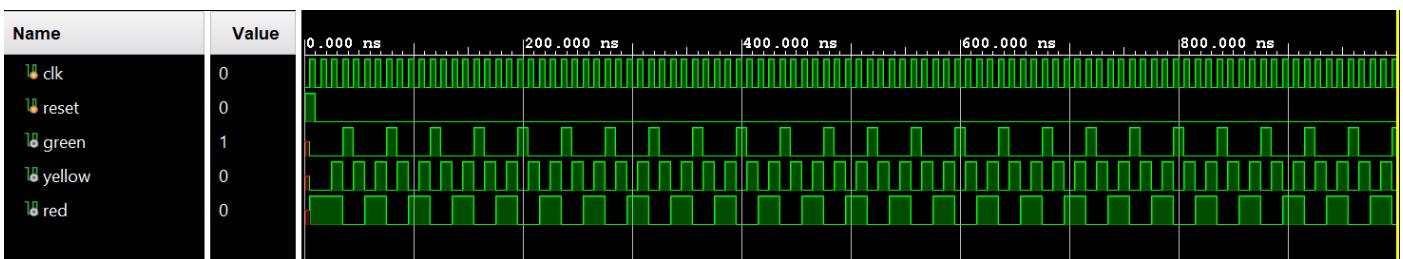
“Aim”:- To Design a synchronous Traffic Light Controller circuit using Verilog

“Design Code”:-

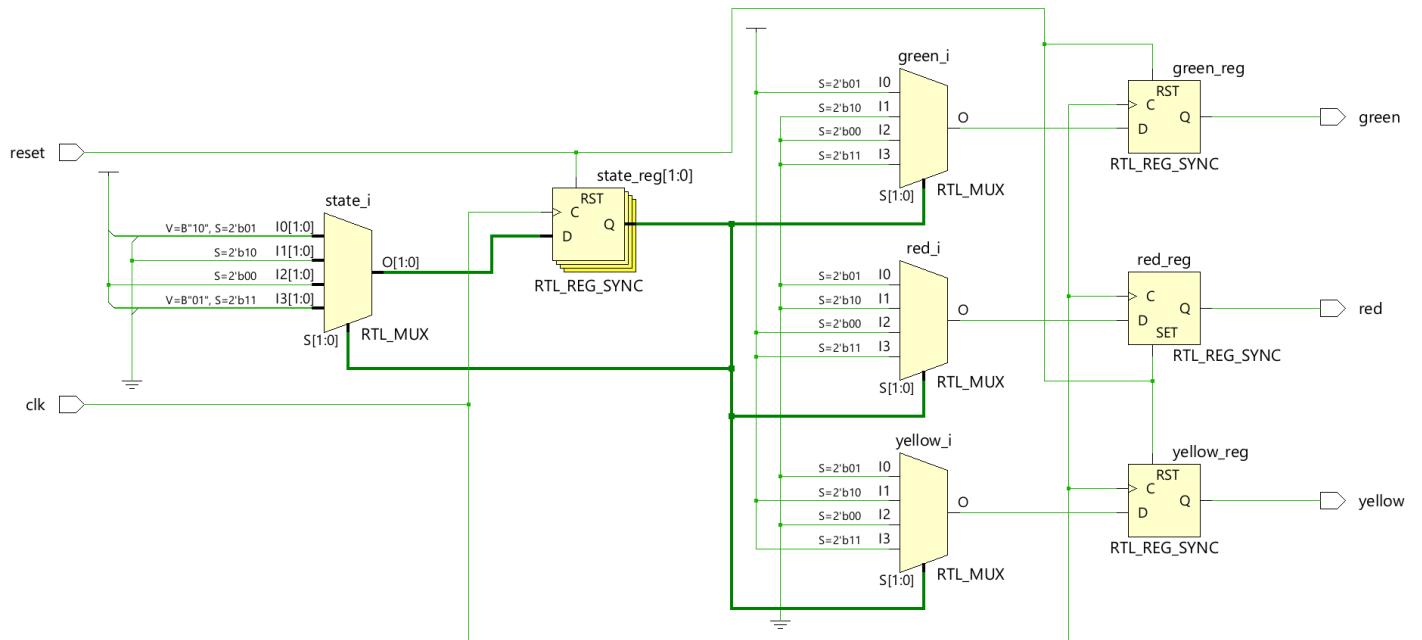


```
`timescale 1ns/1ps
module traffic_light_controller(input clk,reset,output reg green,yellow,red);
reg [1:0] state;
parameter STATE_GREEN = 2'b01,STATE_YELLOW = 2'b10,STATE_RED = 2'b00,STATE_RED_YELLOW = 2'b11;
always @ (posedge clk) begin
if(reset) begin
green <= 0;yellow <= 0;red <= 1;state <= STATE_RED;
end
else begin
case(state)
STATE_GREEN: begin
green <= 1;yellow <= 0;red <= 0;state <= STATE_YELLOW;
end
STATE_YELLOW: begin
green <= 0;yellow <= 1;red <= 0;state <= STATE_RED;
end
STATE_RED: begin
green <= 0;yellow <= 0;red <= 1;state <= STATE_RED_YELLOW;
end
STATE_RED_YELLOW: begin
green <= 0;yellow <= 1;red <= 1;state <= STATE_GREEN;
end
default : begin ; end
endcase
end
end
endmodule
```

“Waveforms”:-



“Schematics”:-



“Console”:-

```

clk=1,reset=0,green=0,yellow=0,red=1
clk=0,reset=0,green=0,yellow=0,red=1
clk=1,reset=0,green=0,yellow=1,red=1
clk=0,reset=0,green=0,yellow=1,red=1
clk=1,reset=0,green=1,yellow=0,red=0
clk=0,reset=0,green=1,yellow=0,red=0
clk=1,reset=0,green=0,yellow=1,red=0
clk=0,reset=0,green=0,yellow=1,red=0
clk=1,reset=0,green=0,yellow=0,red=1
clk=0,reset=0,green=0,yellow=1,red=1
clk=1,reset=0,green=1,yellow=0,red=0
clk=0,reset=0,green=1,yellow=0,red=0
clk=1,reset=0,green=0,yellow=1,red=1
clk=0,reset=0,green=0,yellow=1,red=1
clk=1,reset=0,green=1,yellow=0,red=0
clk=0,reset=0,green=1,yellow=0,red=0
clk=1,reset=0,green=0,yellow=1,red=0
clk=0,reset=0,green=0,yellow=1,red=0
clk=1,reset=0,green=0,yellow=0,red=1
clk=0,reset=0,green=0,yellow=1,red=1
clk=1,reset=0,green=0,yellow=1,red=1
clk=0,reset=0,green=0,yellow=1,red=0
clk=1,reset=0,green=1,yellow=0,red=0
clk=0,reset=0,green=1,yellow=0,red=0
clk=1,reset=0,green=0,yellow=1,red=0
clk=0,reset=0,green=0,yellow=1,red=0

```

Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:

0.819 W

Design Power Budget:

Not Specified

Power Budget Margin:

N/A

Junction Temperature:

26.5°C

Thermal Margin:

58.5°C (30.8 W)

Effective θJA:

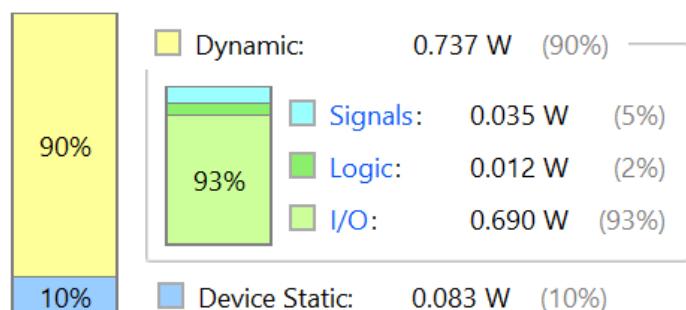
1.9°C/W

Power supplied to off-chip devices: 0 W

Confidence level: **Low**

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-26

#100DAYSRSL

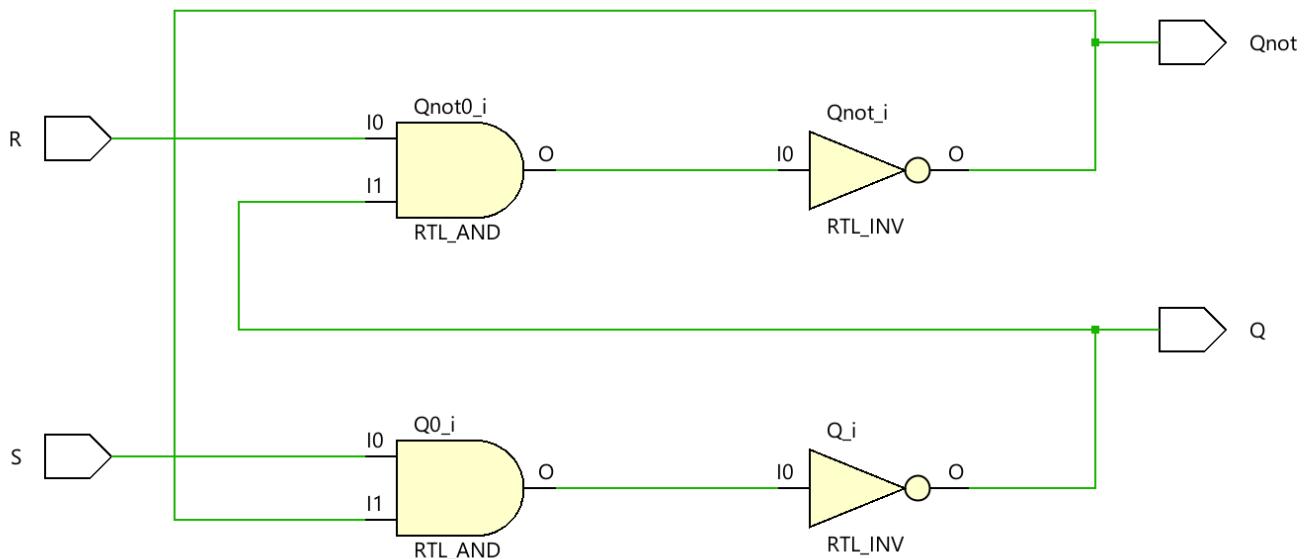
“Aim”:-To design SR Latch using Nand Gates or Nor Gates

“SR latch Using Nand Gates” :-

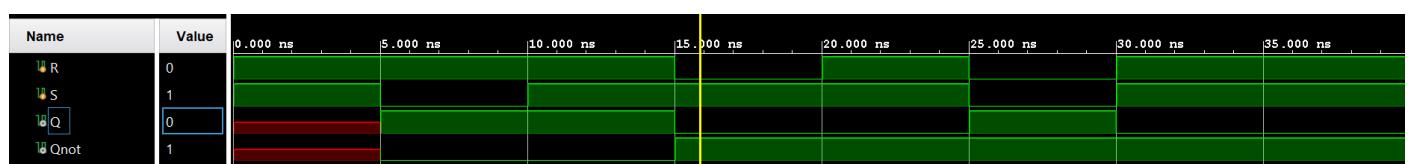
“Design Code”:-

```
module NANDSR_latch(
    input R, S,
    output Q, Qnot);
    nand(Qnot, R, Q);
    nand(Q, S, Qnot);
endmodule
```

“Schematics”:-



“Waveforms”:-



“Console”:-

```
S R !Q Q
1 1 x x
0 1 0 1
1 1 0 1
1 0 1 0
1 1 1 0
0 0 1 1
1 1 1 0
.
```

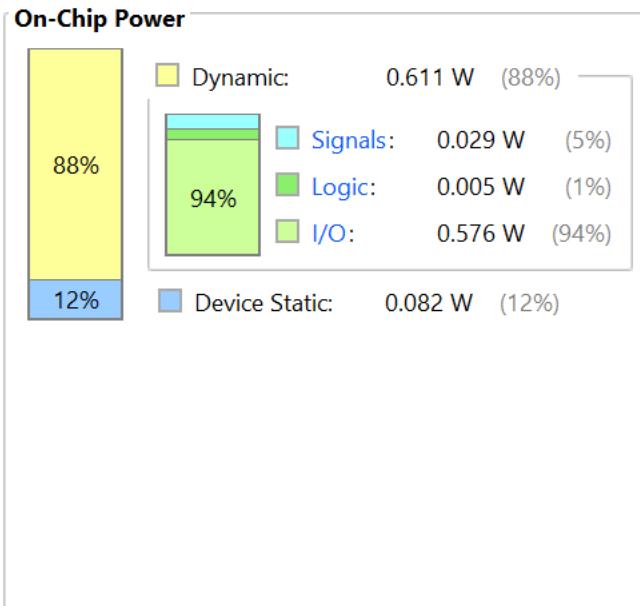
When $\text{!Q} = \text{Q}$, Then it is invalid condition

Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: 0.693 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 26.3°C
Thermal Margin: 58.7°C (31.0 W)
Effective θ_{JA} : 1.9°C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity



“SR Latch Using Nor Gates” :-

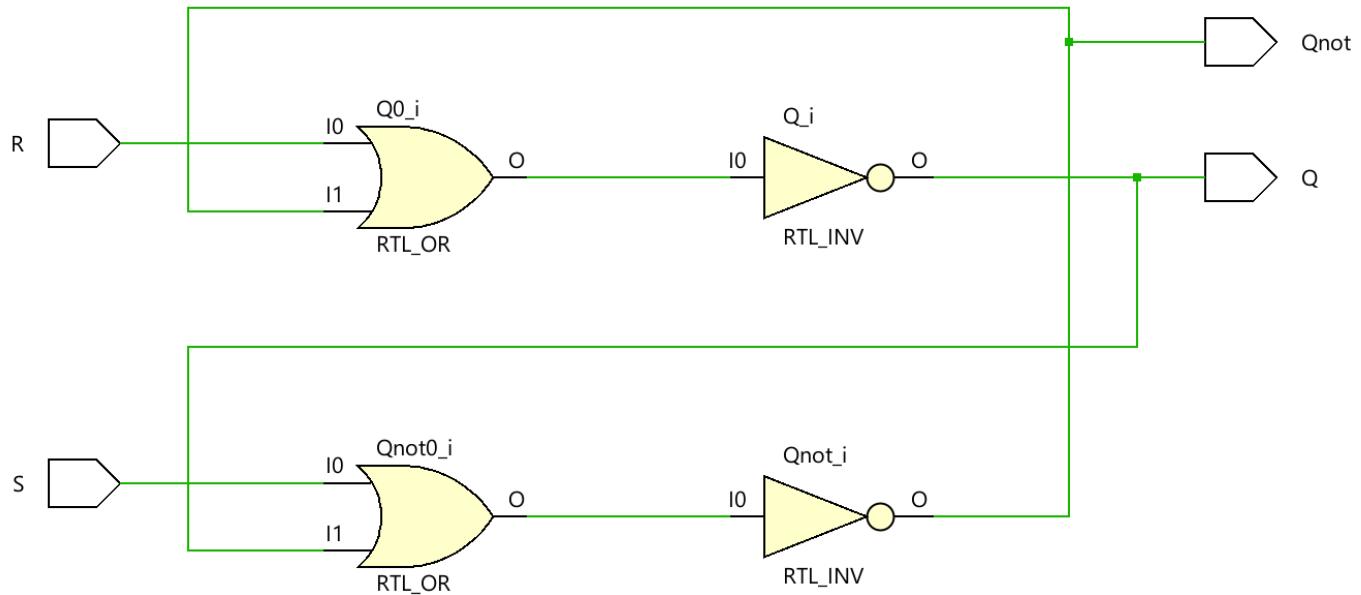
“Design Code”:-

```

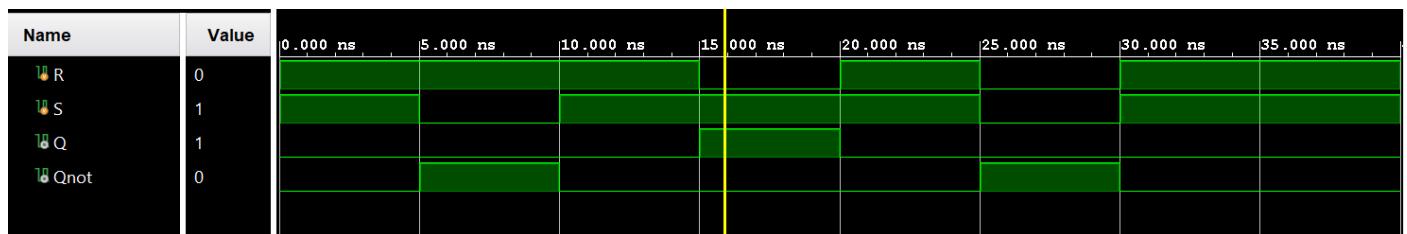
module NORSR_latch(
  input  R, S,
  output Q, Qnot);
  nor(Qnot, S, Q);
  nor(Q, R, Qnot);
endmodule

```

“Schematics”:-



“Waveforms”:-



“Console”:-

```

S R !Q Q
1 1 0 0
0 1 1 0
1 1 0 0
1 0 0 1
1 1 0 0
0 0 1 0
1 1 0 0

```

When $\neg Q = Q$, Then it is invalid condition

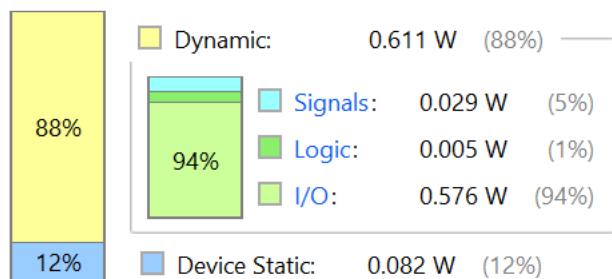
Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	0.693 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	26.3°C
Thermal Margin:	58.7°C (31.0 W)
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-27

#100DAYSRSL

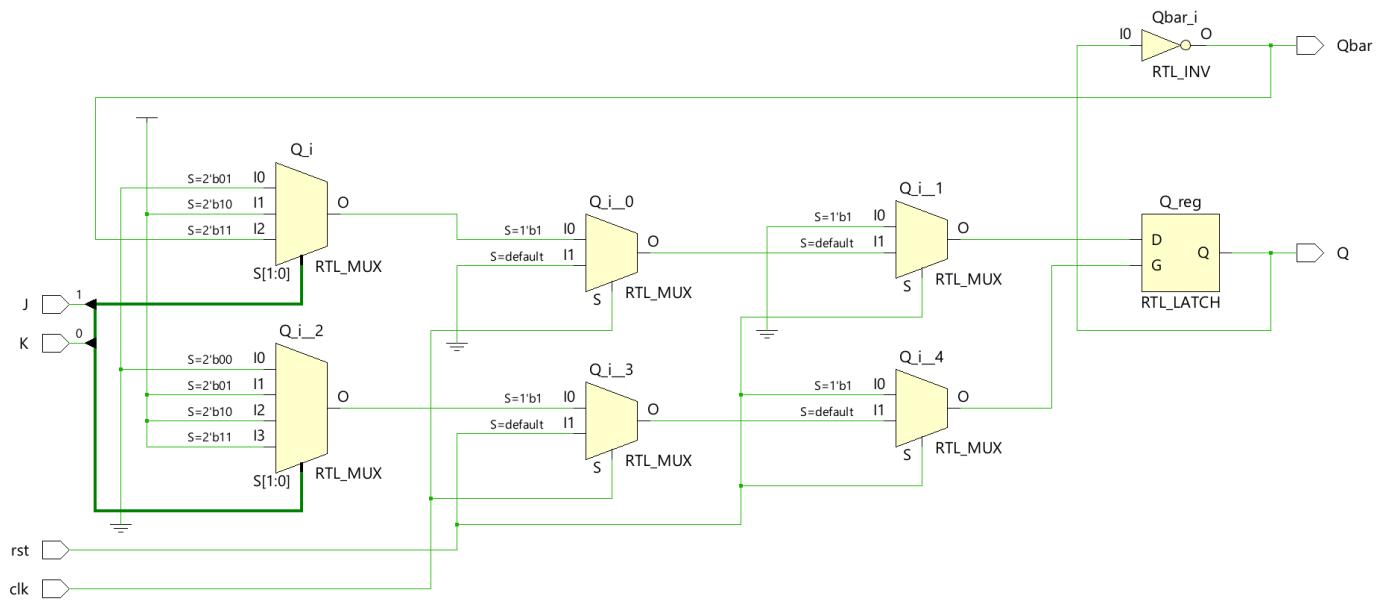
“Aim”:-To design JK Latch and D latch

“JK latch” :-

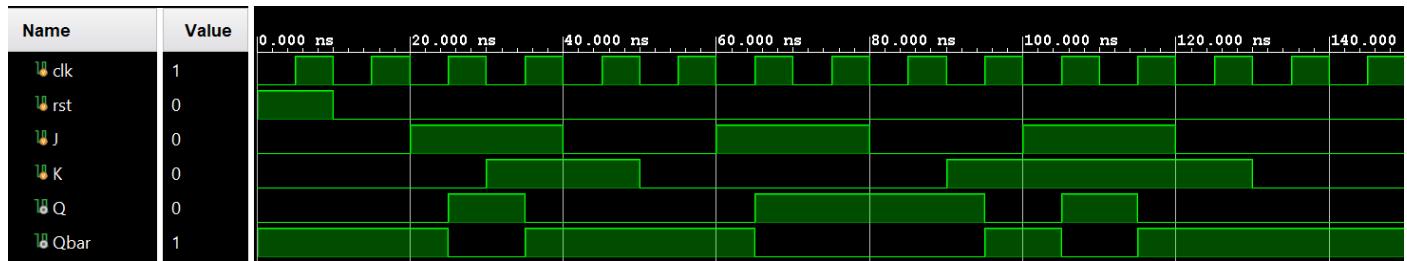
“Design Code”:-

```
`timescale 1ns/1ps
module JK_Latch (
    input clk, rst,
    input J, K,
    output reg Q,
    output Qbar
);
    always_latch begin
        if (rst) Q=0;
        else if(clk) begin
            case ({J, K})
                2'b00: Q <= Q;      // No change
                2'b01: Q <= 1'b0;  // Reset
                2'b10: Q <= 1'b1;  // Set
                2'b11: Q <= ~Q;   // Toggle
            default : begin ; end
            endcase
        end
    end
    assign Qbar = ~Q;
endmodule
```

“Schematics”:-



“Waveforms”:-



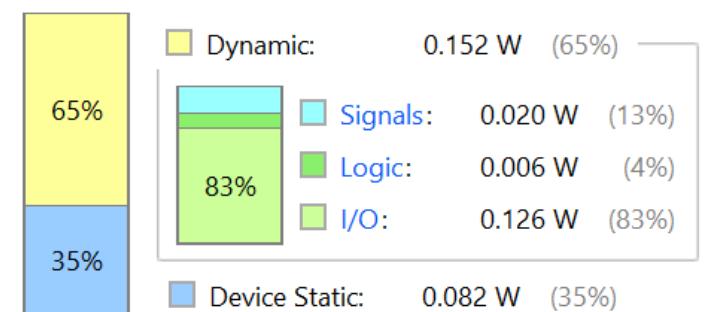
Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	0.233 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	25.4°C

Thermal Margin: 59.6°C (31.4 W)
 Effective θJA: 1.9°C/W
 Power supplied to off-chip devices: 0 W
 Confidence level: Low
[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

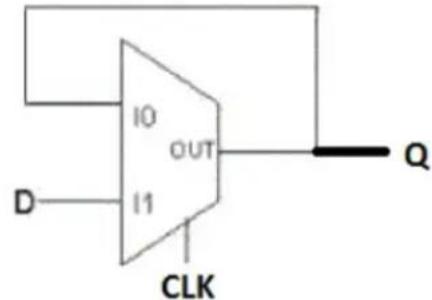
On-Chip Power



“D Latch” :-

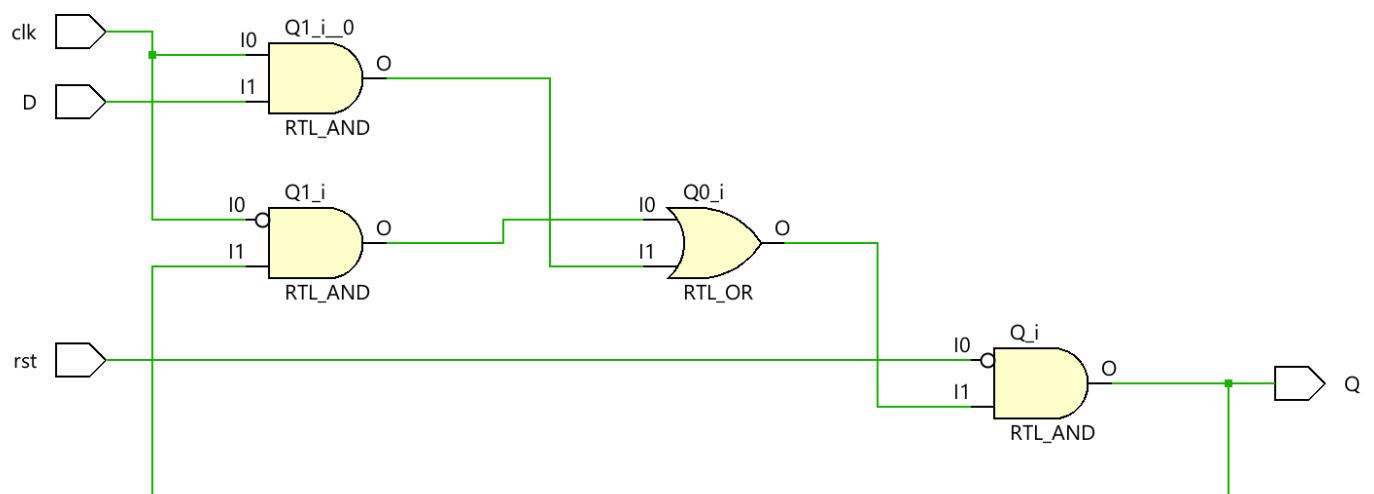
“Design Code”:-

D LATCH USING MUX

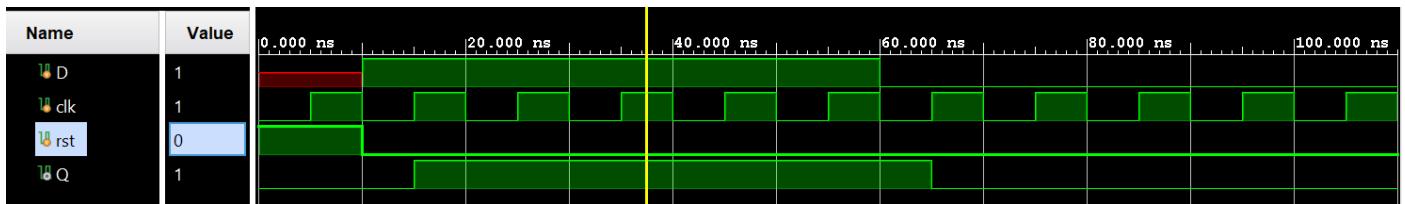


```
'timescale 1ns/1ps
) module D_Latch (
    input clk,rst,
    input D,
    output Q
);
    assign Q=(~rst)&(((~clk)&Q) | (clk & D));
) endmodule
```

“Schematics”:-



“Waveforms”:-



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: 0.275 W

Design Power Budget: Not Specified

Power Budget Margin: N/A

Junction Temperature: 25.5°C

Thermal Margin: 59.5°C (31.4 W)

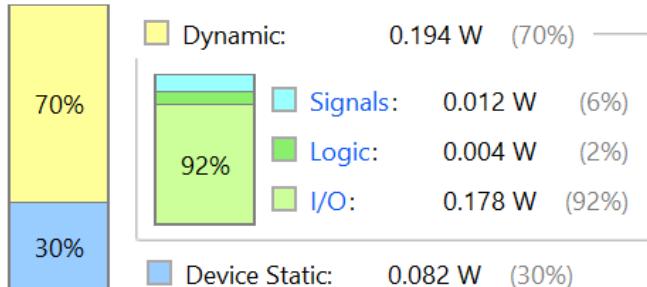
Effective θJA: 1.9°C/W

Power supplied to off-chip devices: 0 W

Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-28

#100DAYSRSL

“Aim”:-To design all single bit flipflops (SR,JK,T,D)

“SR Flipflop” :-

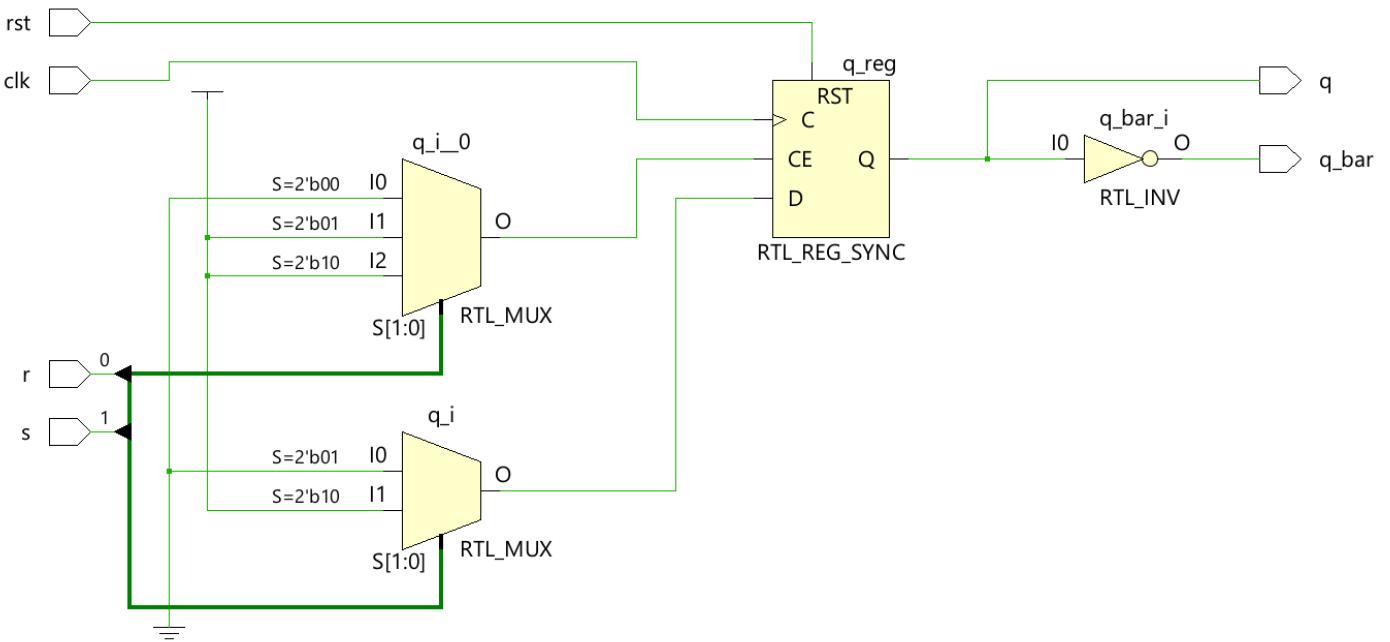
“Design Code”:-

```
module SR_flipflop (
    input clk, rst,
    input s,r,
    output reg q,
    output q_bar
);
    always@(posedge clk) begin
        if(rst) q <= 0;
        else begin
            case({s,r})
                2'b00: q <= q;      // No change
                2'b01: q <= 1'b0; // reset
                2'b10: q <= 1'b1; // set
                2'b11: q <= 1'bX; // Invalid inputs
            endcase
        end
    end
    assign q_bar = ~q;
endmodule
```

“Waveforms”:-



“Schematics”:-



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: 0.488 W

Design Power Budget: Not Specified

Power Budget Margin: N/A

Junction Temperature: 25.9°C

Thermal Margin: 59.1°C (31.2 W)

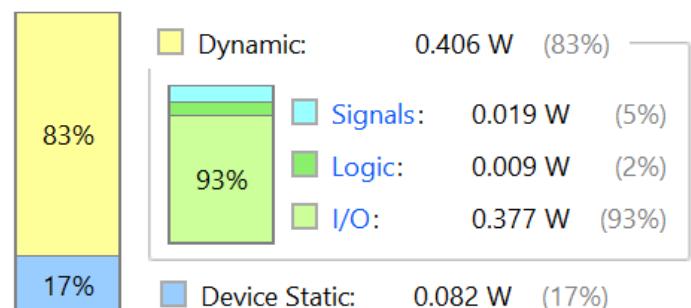
Effective θ_{JA} : 1.9°C/W

Power supplied to off-chip devices: 0 W

Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power

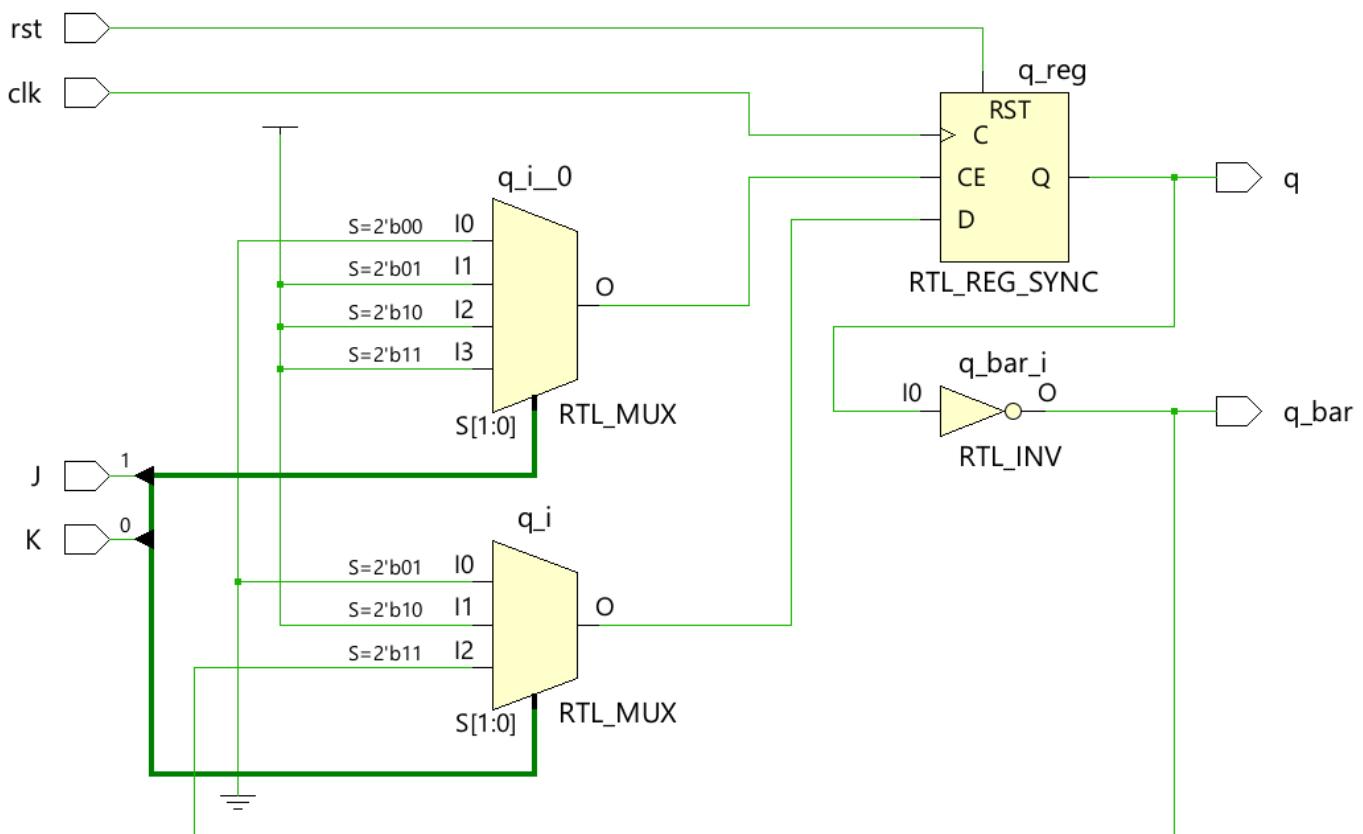


“JK Flipflop” :-

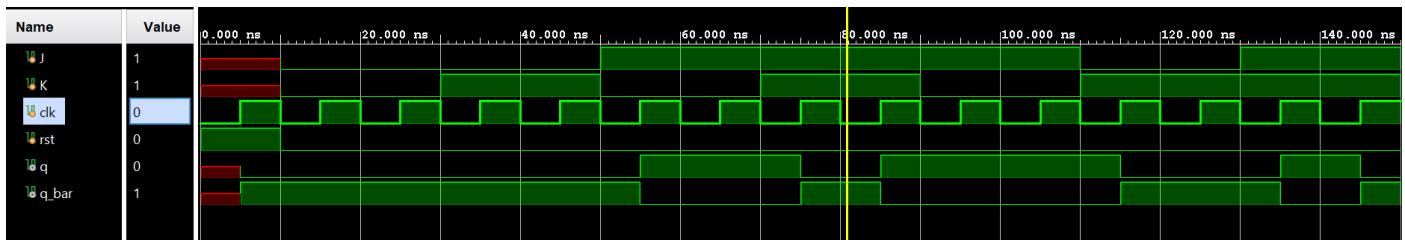
“Design Code”:-

```
module JK_flipflop (
    input clk, rst,
    input J,K,
    output reg q,
    output q_bar
);
    always@(posedge clk) begin
        if(rst) q <= 0;
        else begin
            case({J,K})
                2'b00: q <= q;      // No change
                2'b01: q <= 1'b0; // reset
                2'b10: q <= 1'b1; // set
                2'b11: q <= ~q; // Invalid inputs
            endcase
        end
    end
    assign q_bar = ~q;
endmodule
```

“Schematics”:-



“Waveforms”:-



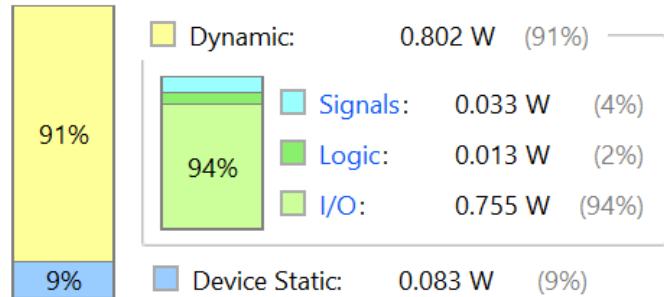
Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	0.884 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	26.7°C
Thermal Margin:	58.3°C (30.8 W)
Effective ϑ_{JA} :	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power

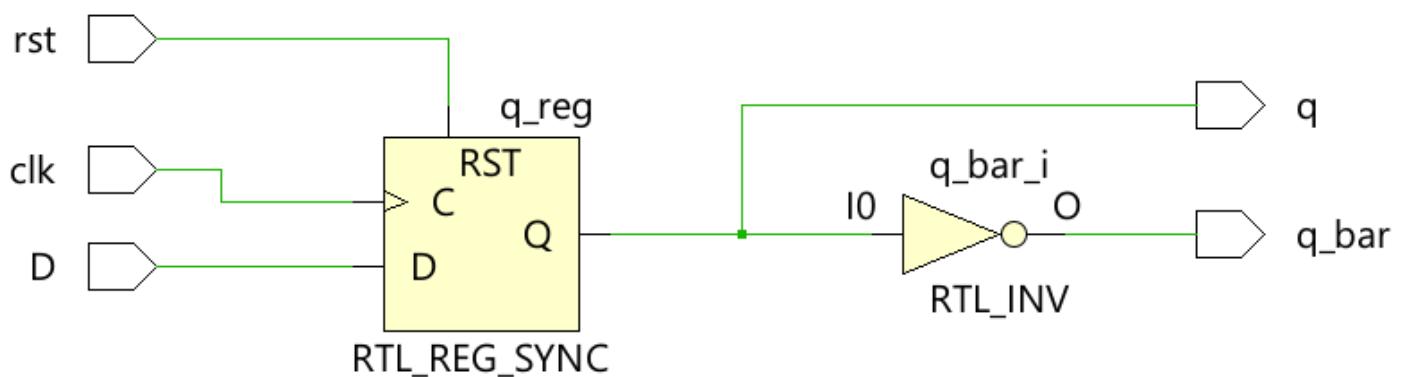


“D Flipflop” :-

“Design Code”:-

```
module D_flipflop (
    input clk, rst,
    input D,
    output reg q,
    output q_bar
);
    always@(posedge clk) begin
        if(rst) q <= 0;
        else begin
            case(D)
                0: q <= 0;      // No change
                1: q <= D;
            endcase
        end
    end
    assign q_bar = ~q;
endmodule
```

“Schematics”:-



“Waveforms”:-

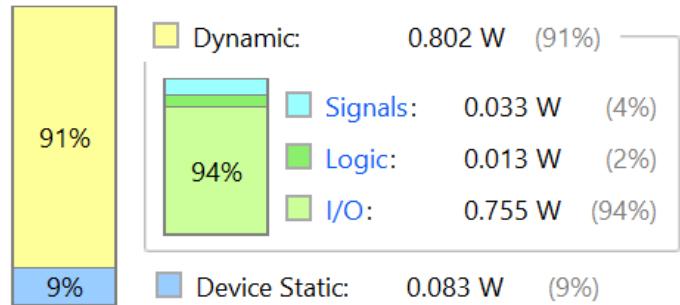


Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	0.884 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	26.7°C
Thermal Margin:	58.3°C (30.8 W)

On-Chip Power

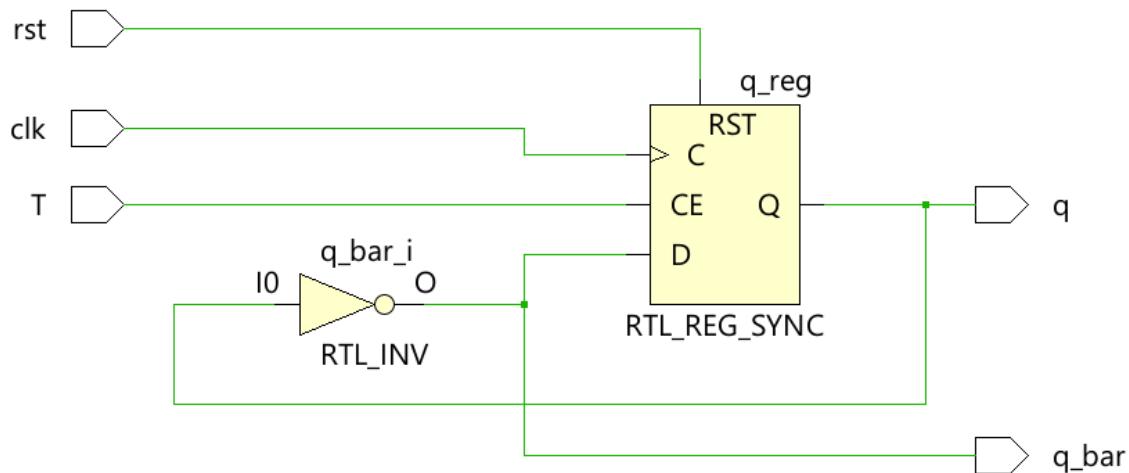


“T Flipflop” :-

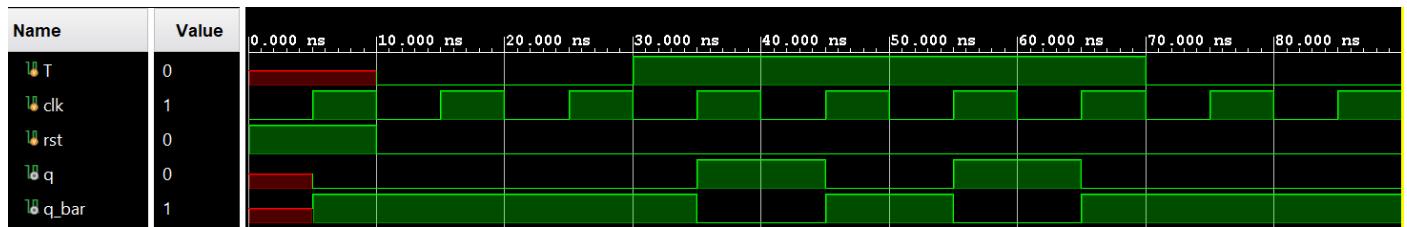
“Design Code”:-

```
module T_flipflop (
    input clk, rst,
    input T,
    output reg q,
    output q_bar
);
    always@(posedge clk) begin
        if(rst) q <= 0;
        else begin
            case(T)
                0: q <= q;      // No change
                1: q <= ~q;
            endcase
        end
    end
    assign q_bar = ~q;
endmodule
```

“Schematics”:-



“Waveforms”:-



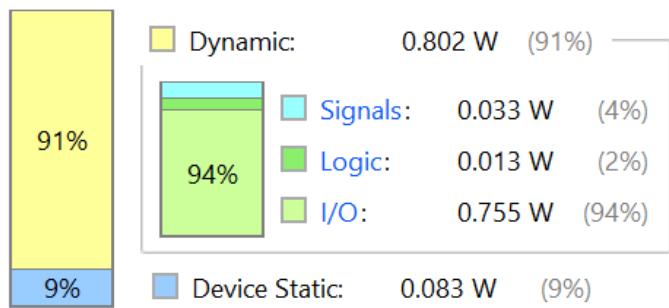
Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	0.884 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	26.7°C
Thermal Margin:	58.3°C (30.8 W)
Effective ϑ_{JA} :	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-29

#100DAYSRSL

“Aim”:-To Specify the differences between a Multi bit flip flop and a register.

Multi-bit Flipflop vs Register :-

Differences between Multi-bit flip-flops and registers:

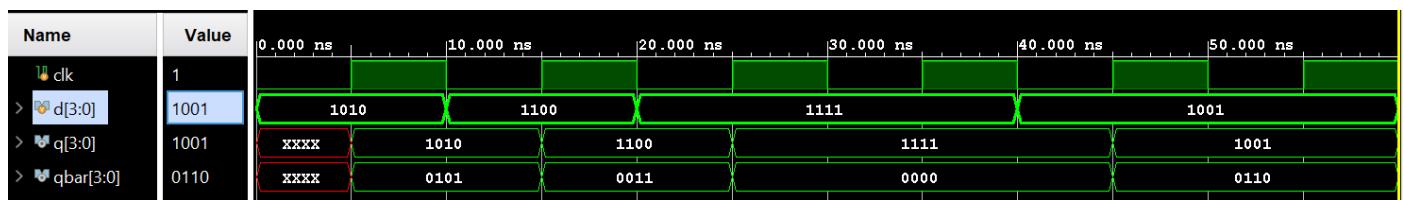
Feature	Multi-bit flip-flop	Register
Definition	A multi-bit flip-flop is a single flip-flop that can store multiple bits of data.	A register is a group of flip-flops that are connected together to store multiple bits of data.
Common uses	Multi-bit flip-flops are typically used to reduce the area and power consumption of digital circuits.	Registers are typically used to store data temporarily, such as the data that is being processed by a microprocessor or the data that is being transmitted between two devices.
Reset signal	Multi-bit flip-flops may or may not have a reset signal.	Registers typically have a reset signal.
Verilog coding	The Verilog code for a multi-bit flip-flop is similar to the Verilog code for a single-bit flip-flop, except that it has multiple D and Q pins.	The Verilog code for a register is simply a group of multi-bit flip-flops that are connected together. The D and Q pins of the flip-flops are connected together in a chain.

D-4bitFlipFlop

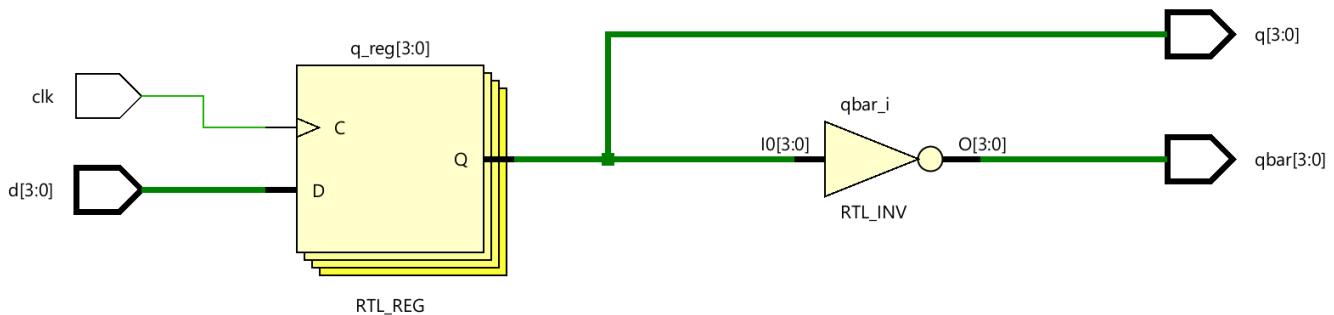
“Design Code”:-

```
| module d_flip_flop_4bit (
|   input clk,
|   input [3:0] d,
|   output reg [3:0] q
| );
|   always @ (posedge clk) begin
|     q <= d;
|   end
| endmodule
```

“Waveforms”:-



“Schematics”:-



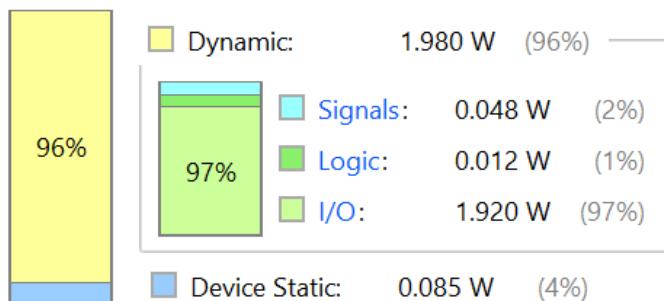
Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	2.065 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	28.9°C
Thermal Margin:	56.1°C (29.6 W)
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



“4bitRegister” :-

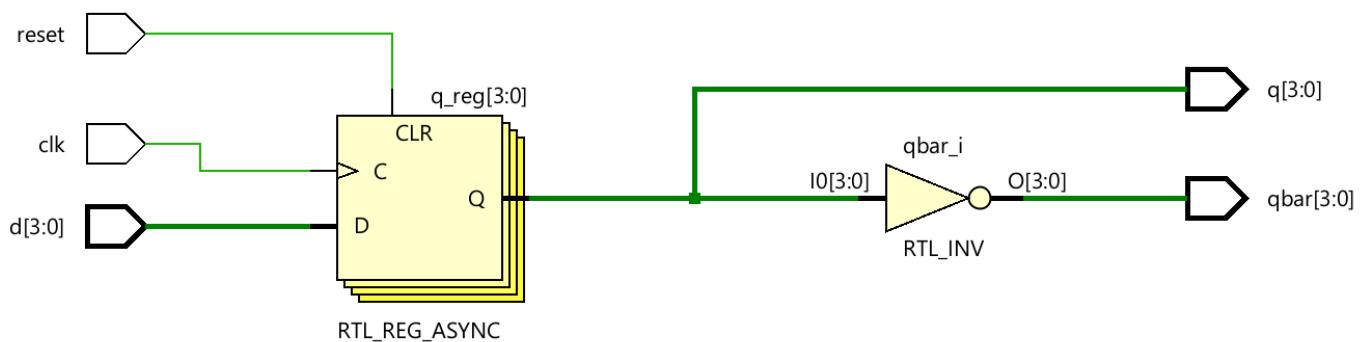
“Design Code”:-

```

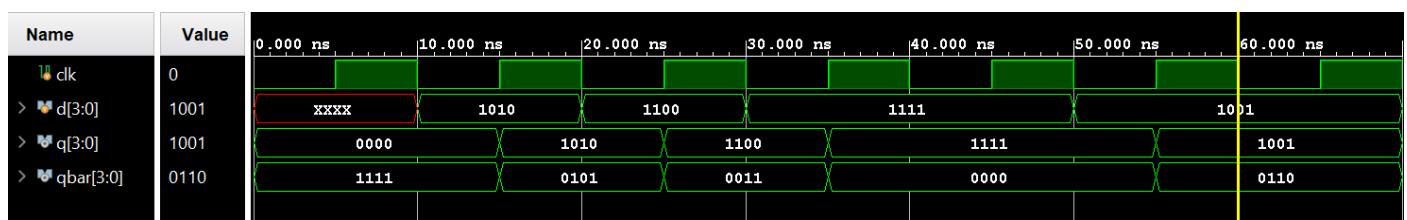
| module register_4bit (
|   input clk,
|   input [3:0] d,
|   input reset,
|   output reg [3:0] q,
|   output [3:0] qbar
| );
|   always @(posedge clk or posedge reset) begin
|     if (reset) begin
|       q <= 4'b0;
|     end else begin
|       q <= d;
|     end
|   end
|   assign qbar=~q;
| endmodule

```

“Schematics”:-



“Waveforms”:-



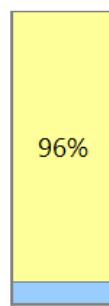
Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	2.111 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	29.0°C
Thermal Margin:	56.0°C (29.6 W)
Effective θ_{JA} :	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



Dynamic: 2.026 W (96%)

Signals: 0.077 W (4%)

Logic: 0.014 W (1%)

I/O: 1.935 W (95%)

Device Static: 0.085 W (4%)



DAY-30

#100DAYSRSL

“Aim”:-To Design Serial Input and Serial output & Serial input and Parallel output

“SISO”

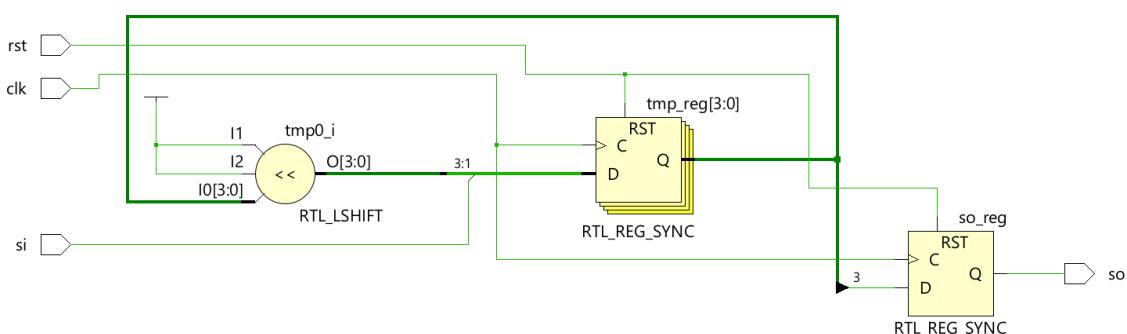
“Design Code”:-

```
module sisomod(clk,rst,si,so);
input clk,si,rst;
output reg so;
reg [3:0] tmp;
always @(posedge clk)
begin
if (rst) begin
tmp <= 4'b0000;
so<=0;
end
else begin
tmp <= tmp << 1;
tmp[0] <= si;
so <= tmp[3];
end
end
endmodule
```

“Waveforms”:-



“Schematics”:-



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:

0.356 W

Design Power Budget:

Not Specified

Power Budget Margin:

N/A

Junction Temperature:

25.7°C

Thermal Margin:

59.3°C (31.3 W)

Effective θ_{JA} :

1.9°C/W

Power supplied to off-chip devices:

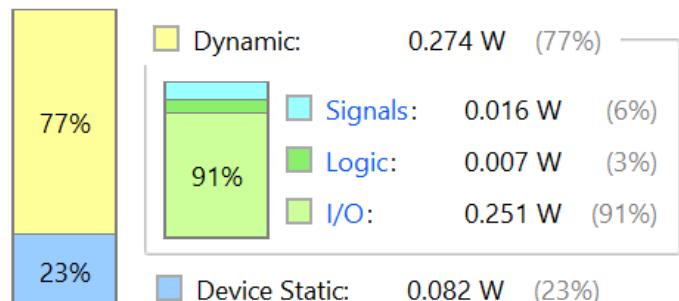
0 W

Confidence level:

Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power

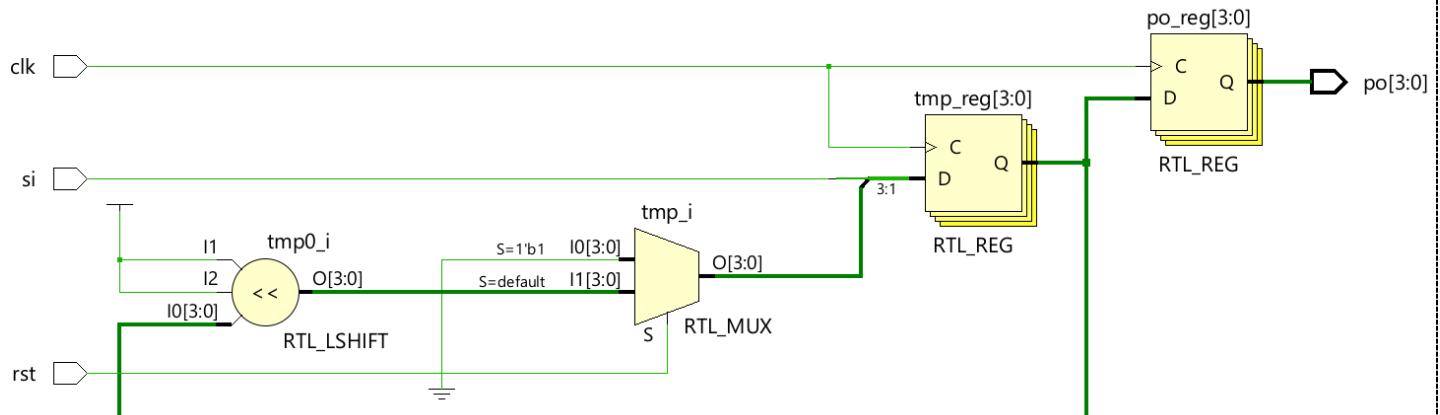


“SIPo”

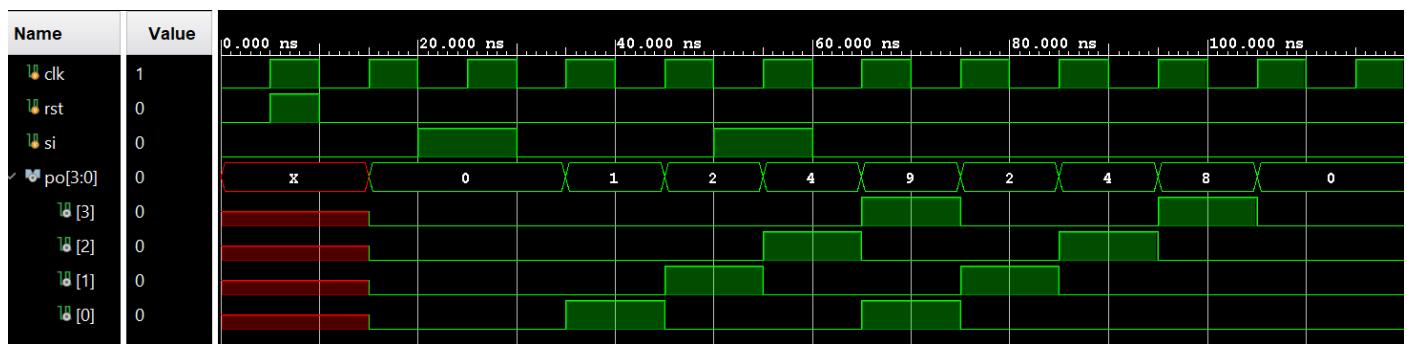
“Design Code”:-

```
module sipomod(clk,rst, si, po);
  input clk, si,rst;
  output reg [3:0] po;
  reg [3:0] tmp;
  always @(posedge clk) begin
    if (rst)
      tmp <= 4'b0000;
    else
      tmp <= tmp << 1;
    tmp[0] <= si;
    po = tmp;
  end
endmodule
```

“Schematics”:-



“Waveforms”:-



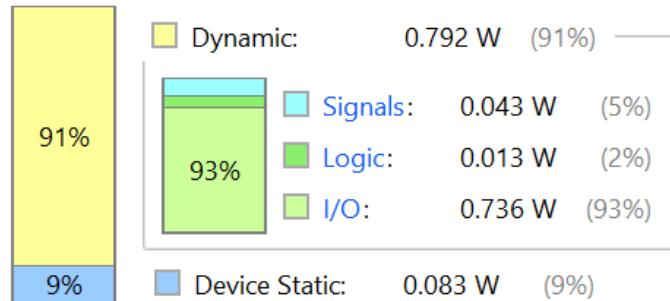
Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	0.875 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	26.6°C
Thermal Margin:	58.4°C (30.8 W)
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-31

#100DAYSRSL

“Aim”:-To Design Parallel Input and Serial output & Parallel input and Parallel output

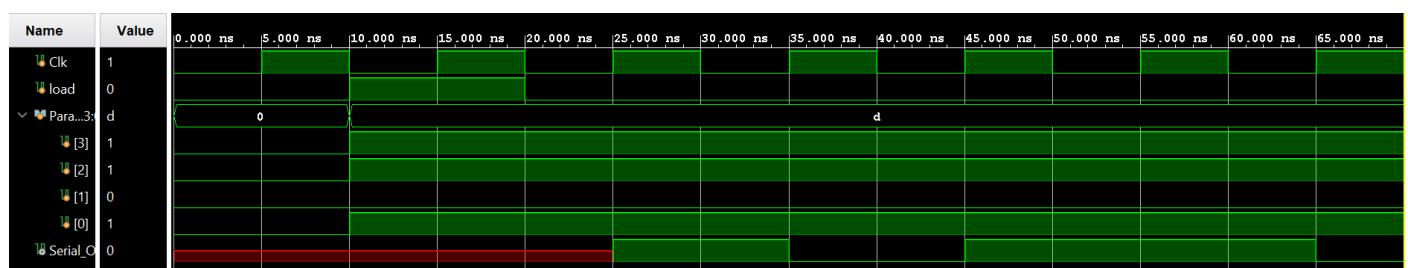
“PISO”

“Design Code”:-

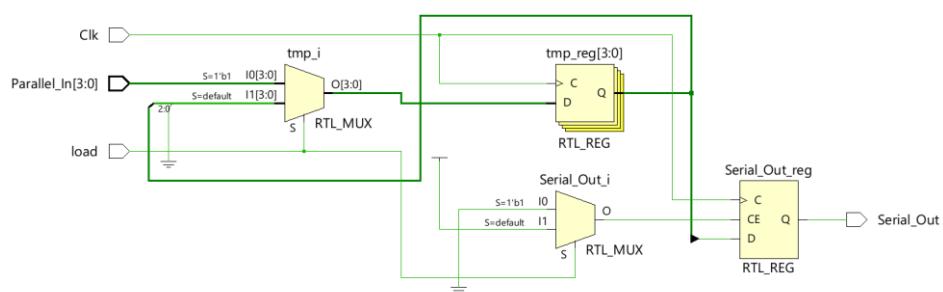
```

module Shiftregister_PISO(Clk, Parallel_In,load, Serial_Out);
  input Clk,load;
  input [3:0]Parallel_In;
  output reg Serial_Out;
  reg [3:0]tmp;
  always @(posedge Clk)
  begin
    if(load)
      tmp<=Parallel_In;
    else
      begin
        Serial_Out<=tmp[0];
        tmp<={1'b0,tmp[3:1]};
      end
    end
  endmodule
  
```

“Waveforms”:-



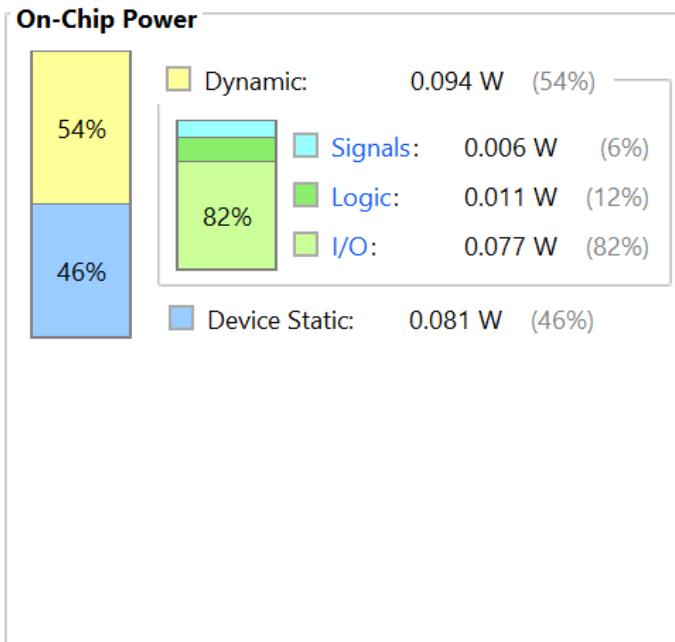
“Schematics”:-



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	0.176 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	25.3°C
Thermal Margin:	59.7°C (31.5 W)
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low
Launch Power Constraint Advisor to find and fix invalid switching activity	

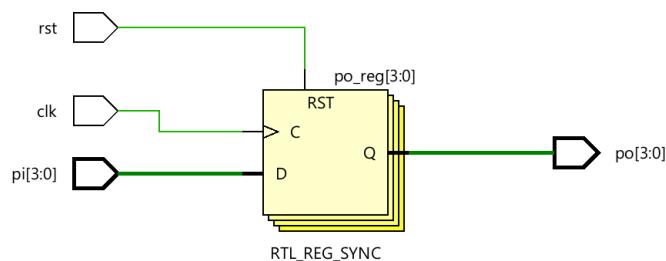


“PIPO”

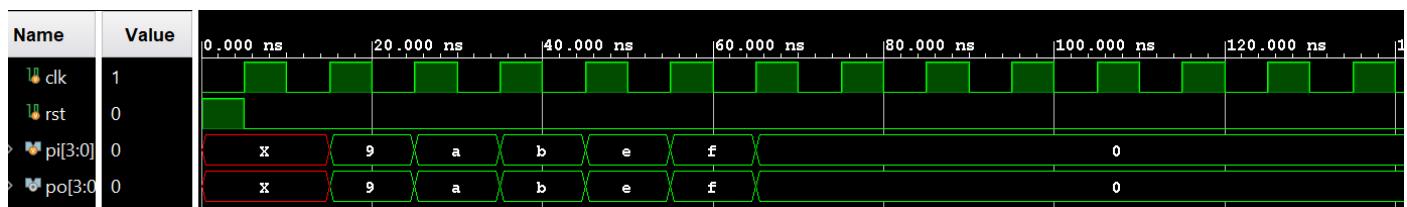
“Design Code”:-

```
module pipomod(clk,rst, pi, po);
input clk,rst;
input [3:0] pi;
output reg [3:0] po;
always @ (posedge clk)
begin
if (rst)
po<= 4'b0000;
else
po <= pi;
end
endmodule
```

“Schematics”:-



“Waveforms”:-



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: 1.106 W

Design Power Budget: Not Specified

Power Budget Margin: N/A

Junction Temperature: 27.1°C

Thermal Margin: 57.9°C (30.6 W)

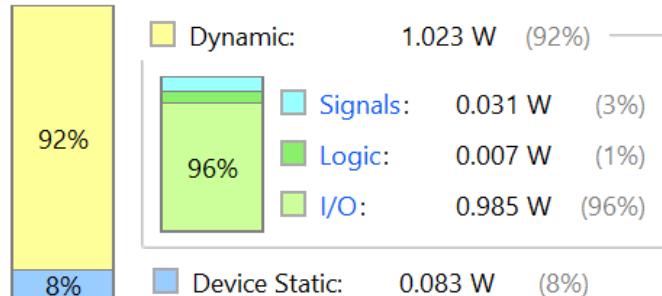
Effective ϑ_{JA} : 1.9°C/W

Power supplied to off-chip devices: 0 W

Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-32

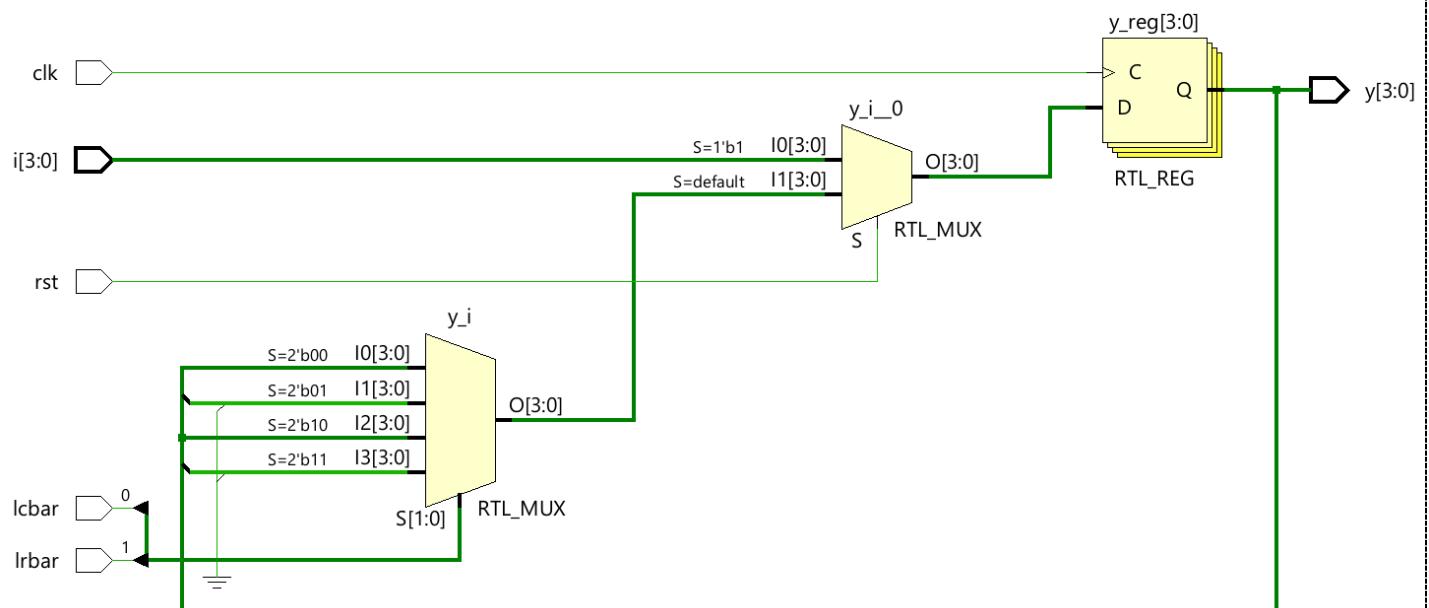
#100DAYSRTL

“Aim”:-To Design Universal Shift register which performs linear circular left and right shift operations

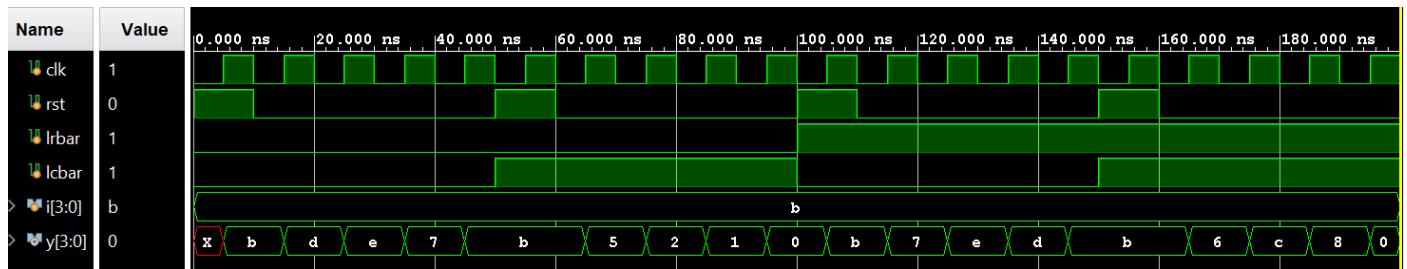
“Design Code”:-

```
module usr_shift_reg.bl(input clk,rst,lrbar,lcbar,input [3:0]i,output reg [3:0]y);
always @ (posedge clk)begin
if(rst) y<=i;
else
begin
case({lrbar,lcbar})
2'b00:begin y<={y[0],y[3:1]}; end
2'b01: begin y<={1'b0,y[3:1]}; end
2'b10: begin y<={y[2:0],y[3]}; end
2'b11: begin y<={y[2:0],1'b0}; end
default: begin end
endcase
end
end
endmodule|
```

“Schematics”:-



“Waveforms”:-



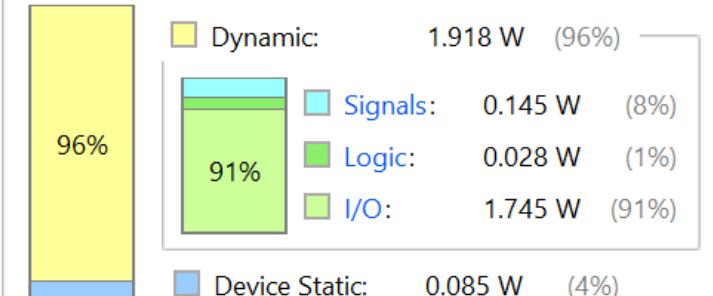
Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	2.003 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	28.8°C
Thermal Margin:	56.2°C (29.7 W)
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-33

#100DAYSRSL

“Aim”:-To Design Binary Ripple counter or Asynchronous (series) counter

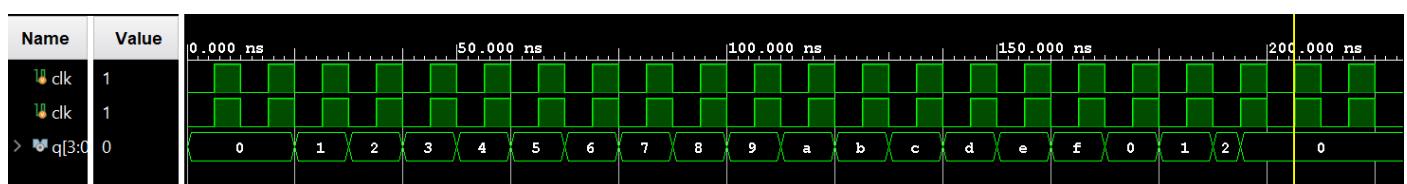
“Design Code”:-

```
module ripplecounter(clk,rst,q); //Ripple counter using T flipflop
  input clk,rst;
  output [3:0]q;
  tff tf1(q[0],clk,rst);
  tff tf2(q[1],q[0],rst);
  tff tf3(q[2],q[1],rst);
  tff tf4(q[3],q[2],rst);
endmodule

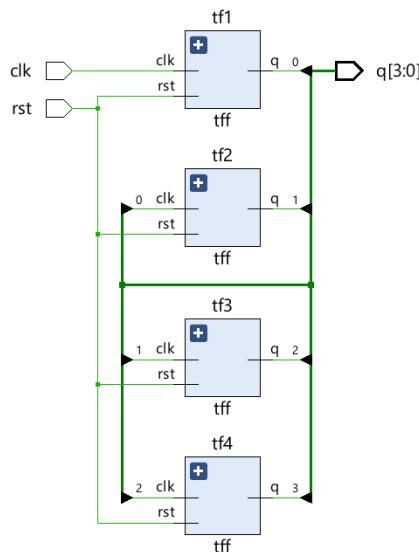
module tff(q,clk,rst); //T flipflop using D-flipflop always in toggling mode
  input clk,rst;
  output q;
  wire d;
  dff df1(q,d,clk,rst);
  not n1(d,q);
endmodule

module dff(q,d,clk,rst); //D-flipflop
  input d,clk,rst;
  output q;
  reg q;
  always @ (negedge clk or posedge rst)
    begin
      if(rst) q=1'b0;
      else q=d;
    end
endmodule
```

“Waveforms”:-



“Schematics”:-



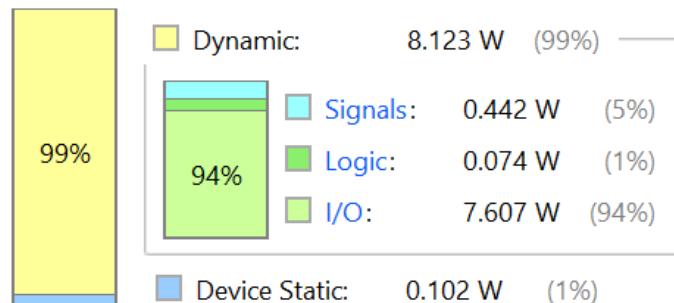
Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	8.225 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	40.5°C
Thermal Margin:	44.5°C (23.5 W)
Effective ϑ_{JA} :	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-34

#100DAYSRSL

“Aim”:-To Design Non-binary Ripple counter (Decade counter or Mod10 counter).

“Design Code”:-

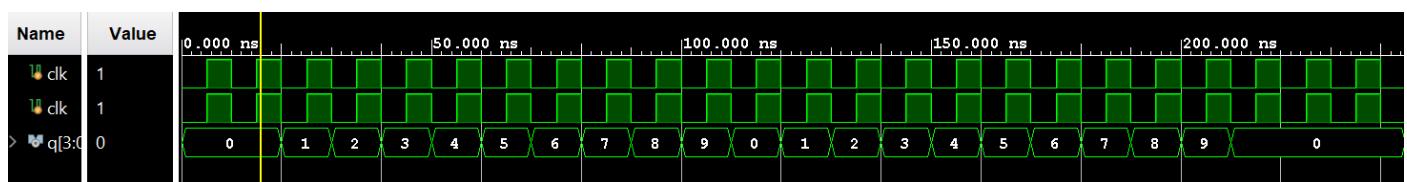
```
module decade_counter(clk, rst, q);
    input clk, rst;
    output [3:0] q;
    wire q31_condition;
    assign q31_condition = (q[3] == 1'b1) && (q[1] == 1'b1); // Detect when q[3] and q[1] are both 1
    ripplecounter ripple_inst(.clk(clk), .rst(rst || q31_condition), .q(q));
endmodule

module ripplecounter(clk,rst,q); //Ripple counter using T flipflop
    input clk,rst;
    output [3:0]q;
    tff tf1(q[0],clk,rst);
    tff tf2(q[1],q[0],rst);
    tff tf3(q[2],q[1],rst);
    tff tf4(q[3],q[2],rst);
endmodule

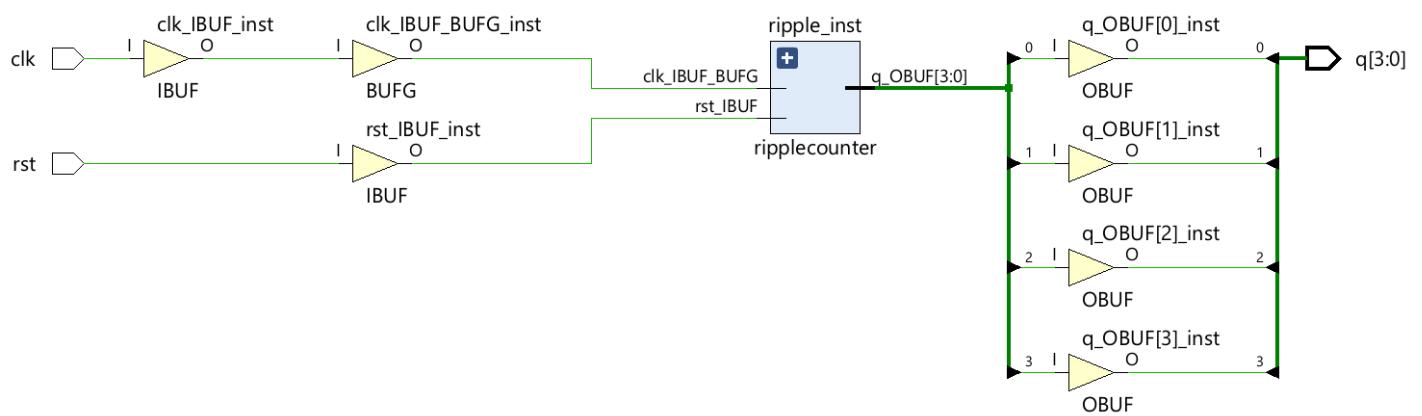
module tff(q,clk,rst); //T flipflop using D-flipflop always in toggling mode
    input clk,rst;
    output q;
    wire d;
    dff df1(q,d,clk,rst);
    not n1(d,q);
endmodule

module dff(q,d,clk,rst); //D-flipflop
    input d,clk,rst;
    output q;
    reg q;
    always @ (negedge clk or posedge rst)
    begin
        if(rst) q=1'b0;
        else q=d;
    end
endmodule
```

“Waveforms”:-



“Schematics”:-

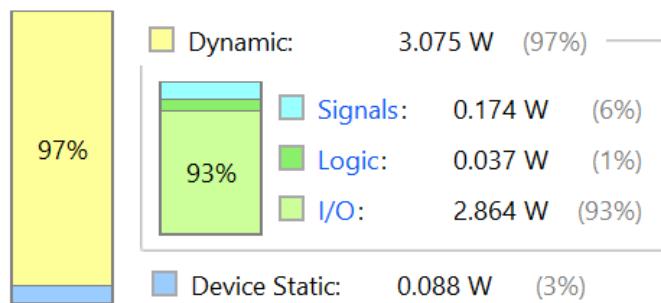


Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	3.162 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	31.0°C
Thermal Margin:	54.0°C (28.5 W)
Effective θ_{JA} :	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low
Launch Power Constraint Advisor to find and fix invalid switching activity	

On-Chip Power





DAY-35

#100DAYSRSL

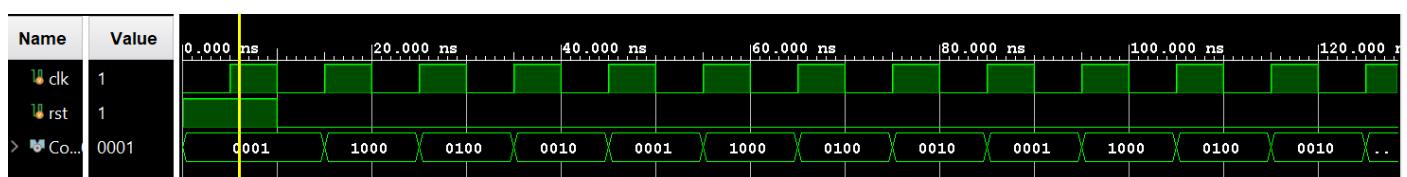
“Aim”:-To Design Ring Counter (non self starting counter)

“Design Code”:-

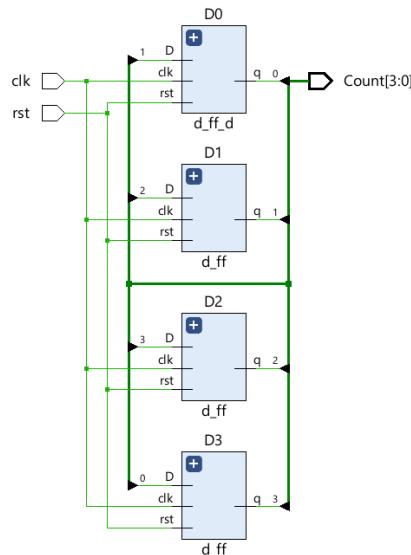
```
module Ringcounter(Count,clk,rst);
  input clk,rst;
  output [3:0] Count;
  wire [3:0] temp ;
  d_ff D3(temp[3],clk,temp[0],rst);
  d_ff D2(temp[2],clk,temp[3],rst);
  d_ff D1(temp[1],clk,temp[2],rst);
  d_ff_d D0(temp[0],clk,temp[1],rst);
  assign Count=temp;
endmodule
module d_ff(q,clk,D,rst);
  input clk,rst,D;
  output reg q;
  always @ (posedge clk or posedge rst) begin
    if (rst) q<=0;
    else q<=D ;
  end
endmodule
module d_ff_d(q,clk,D,rst);
  input clk,rst,D;
  output reg q;
  always @ (posedge clk or posedge rst) begin
    if (rst) q<=1;
    else q<=D ;
  end
endmodule

```

“Waveforms”:-



“Schematics”:-



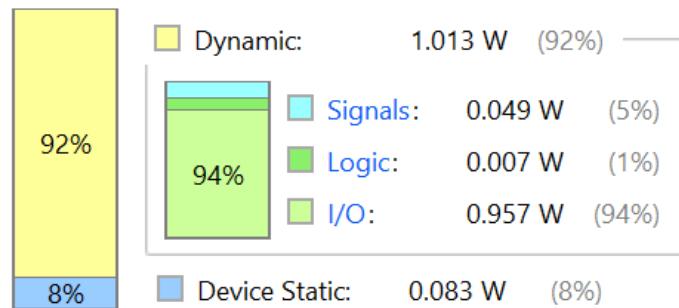
Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	1.096 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	27.1°C
Thermal Margin:	57.9°C (30.6 W)
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-36

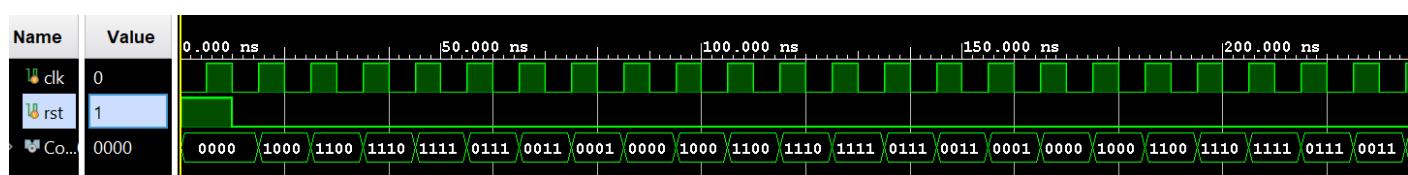
#100DAYSRSL

“Aim”:-To Design JhonSonCounter (SwitchTailRing counter)

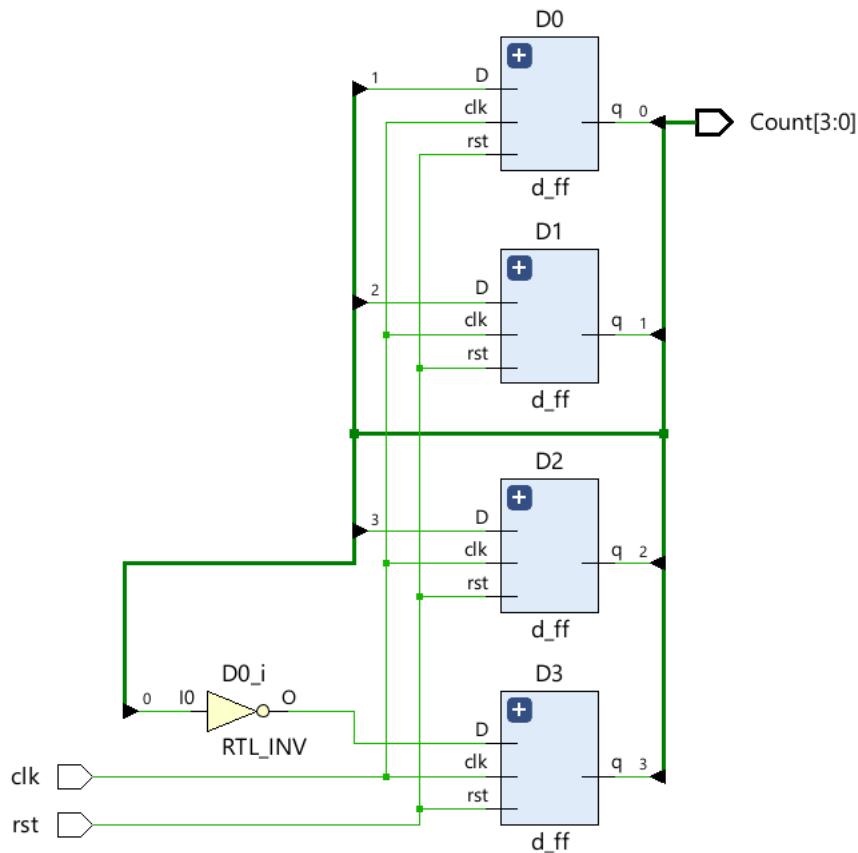
“Design Code”:-

```
module JhonSoncounter(Count,clk,rst);
    input clk,rst;
    output [3:0] Count;
    wire [3:0] temp ;
    d_ff D3(temp[3],clk,~temp[0],rst);
    d_ff D2(temp[2],clk,temp[3],rst);
    d_ff D1(temp[1],clk,temp[2],rst);
    d_ff D0(temp[0],clk,temp[1],rst);
    assign Count=temp;
endmodule
module d_ff(q,clk,D,rst);
    input clk,rst,D;
    output reg q;
    always @ (posedge clk or posedge rst) begin
        if (rst) q<=0;
        else q<=D ;
    end
endmodule
```

“Waveforms”:-



“Schematics”:-



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	1.099 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	27.1°C

Thermal Margin: 57.9°C (30.6 W)

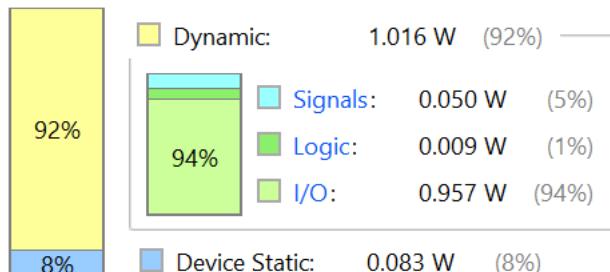
Effective QJA: 1.9°C/W

Power supplied to off-chip devices: 0 W

Confidence level: **Low**

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-37

#100DAYSRSL

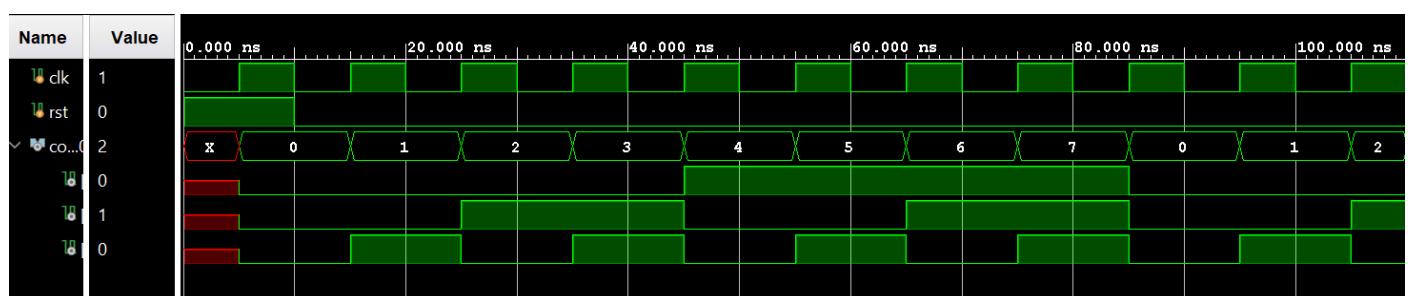
“Aim”:-To Design a Synchronous series counter

“Design Code”:-

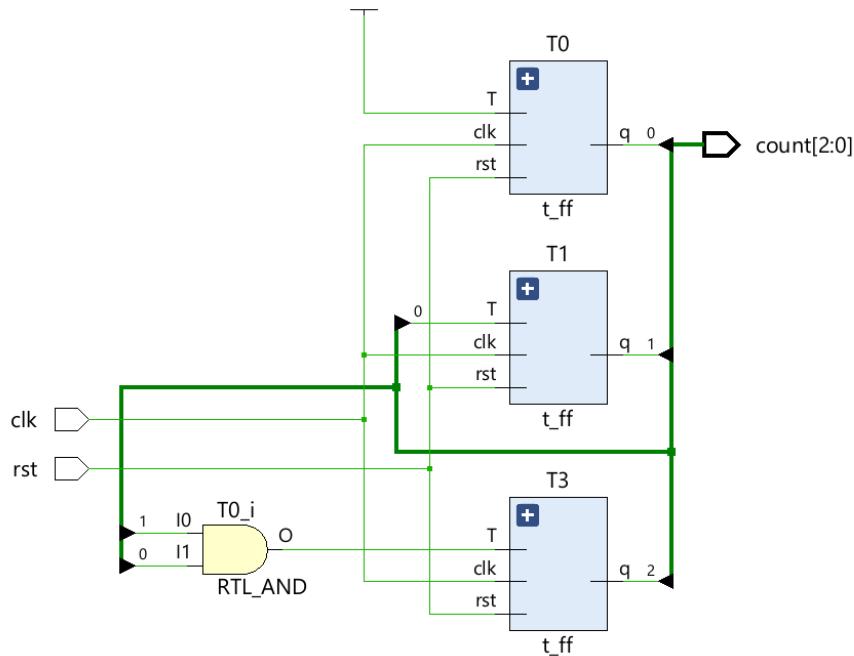
```
module SSC(input clk,rst,output [2:0] count);
  wire [2:0] q;
  t_ff T0(1'b1,rst,clk,q[0]);
  t_ff T1(q[0],rst,clk,q[1]);
  t_ff T3((q[1]&q[0]),rst,clk,q[2]);
  assign count=q;
endmodule

module t_ff(input T,rst,clk,output reg q);
  always @ (posedge clk) begin
    if(rst) q<=1'b0;
    else begin
      case (T)
        1'b0:q<=q;
        1'b1:q<=~q;
        default : begin ; end
      endcase
    end
  end
endmodule
```

“Waveforms”:-



“Schematics”:-



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: 1.581 W

Design Power Budget: Not Specified

Power Budget Margin: N/A

Junction Temperature: 28.0°C

Thermal Margin: 57.0°C (30.1 W)

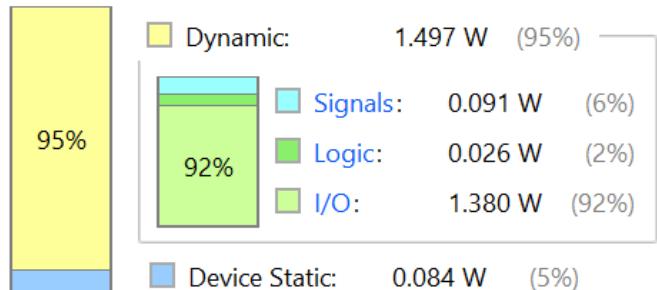
Effective θ_{JA}: 1.9°C/W

Power supplied to off-chip devices: 0 W

Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-38

#100DAYSRSL

“Aim”:-To Design a Universal Counter (Down & Up)

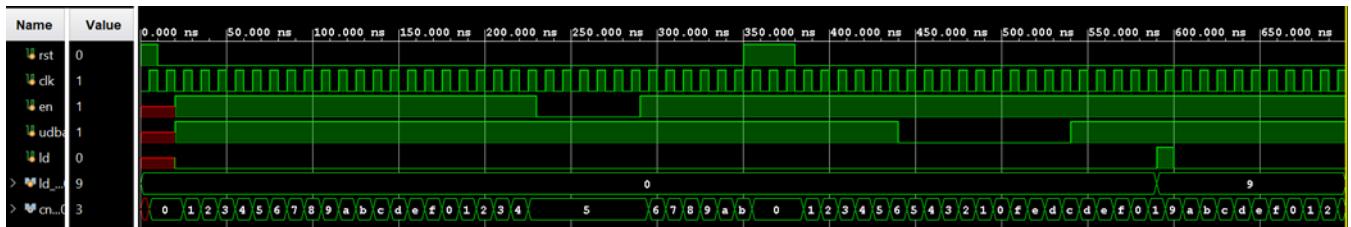
“Design Code”:-

```

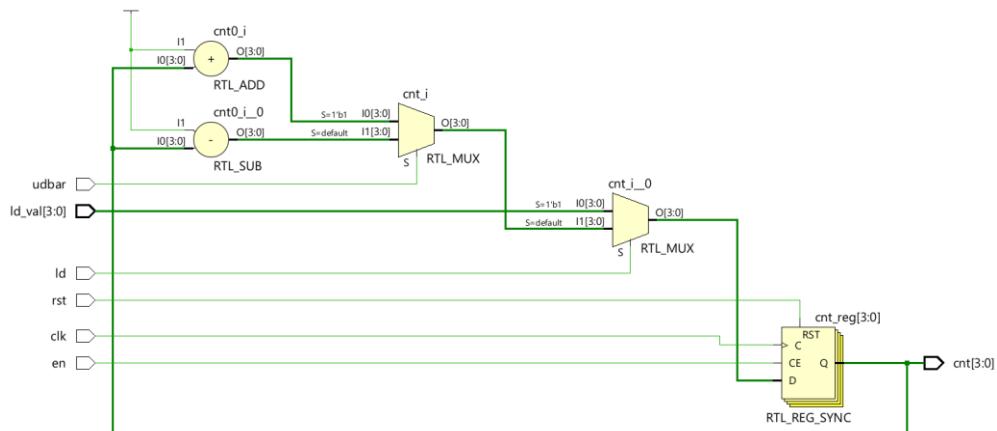
module Counter(input rst,clk,en,udbar,ld,input [3:0] ld_val,output reg [3:0] cnt);
  always @(posedge clk)
    begin
      if (rst) cnt<=0;
      else
        begin
          if(en) begin
            if (ld) cnt<=ld_val;
            else begin
              if (udbar) cnt<=cnt+1;
              else cnt<=cnt-1;
            end
          end
        end
      end
    end
endmodule

```

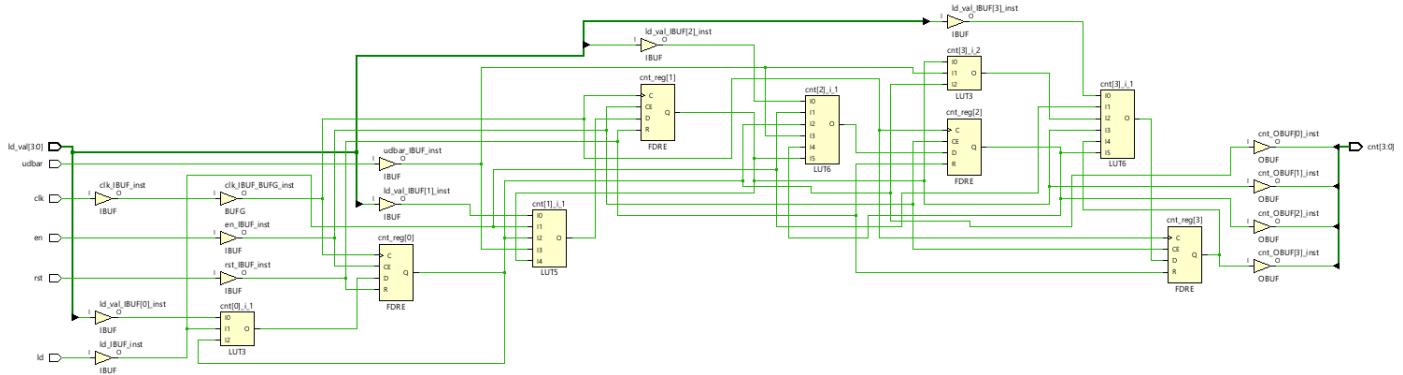
“Waveforms”:-



“Elaborated Design”:-



“Implemented Design”:-



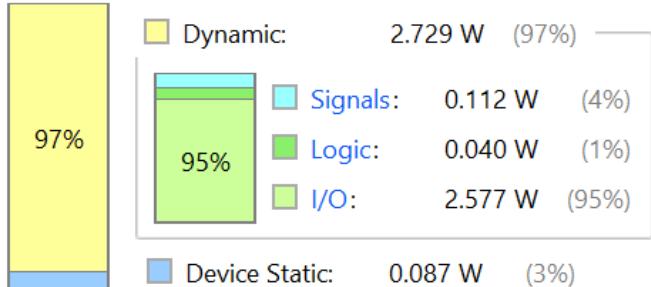
Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	2.816 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	30.3°C
Thermal Margin:	54.7°C (28.8 W)
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-39

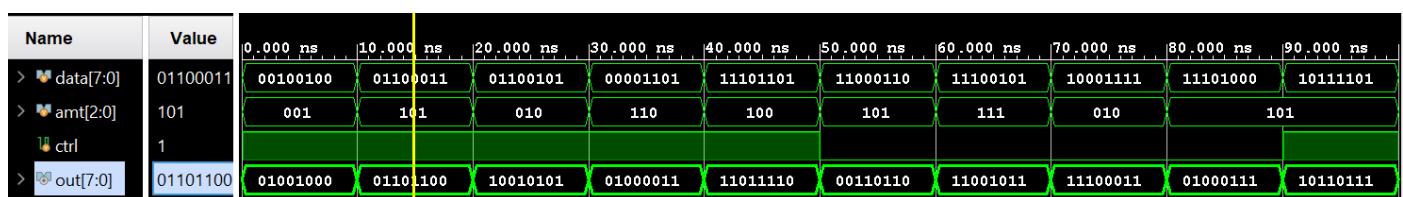
#100DAYSRSL

“Aim”:-To design a Multi-Functional barrel shifter (rotates left or right)

“Design Code”:-

```
module barrel_shifter_multi(input [7:0] data,input [2:0] amt,input ctrl,output reg [7:0] out);
always @(*) begin
  if(ctrl) begin
    case (amt)
      3'd0: out = data;
      3'd1: out = {data[0], data[7:1]};
      3'd2: out = {data[1:0], data[7:2]};
      3'd3: out = {data[2:0], data[7:3]};
      3'd4: out = {data[3:0], data[7:4]};
      3'd5: out = {data[4:0], data[7:5]};
      3'd6: out = {data[5:0], data[7:6]};
      default out = {data[6:0], data[7]};
    endcase
  end
  else begin
    case (amt)
      3'd0: out = data;
      3'd1: out = {data[6:0], data[7]};
      3'd2: out = {data[5:0], data[7:6]};
      3'd3: out = {data[4:0], data[7:5]};
      3'd4: out = {data[3:0], data[7:4]};
      3'd5: out = {data[2:0], data[7:3]};
      3'd6: out = {data[1:0], data[7:2]};
      default out = {data[0], data[7:1]};
    endcase
  end
end
end
endmodule
```

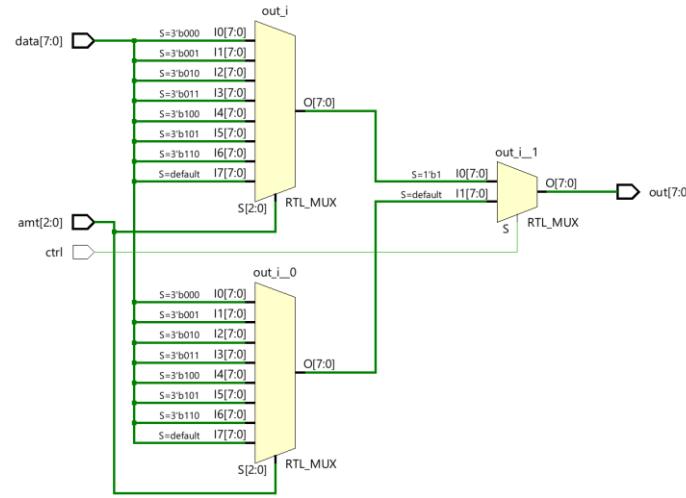
“Waveforms”:-



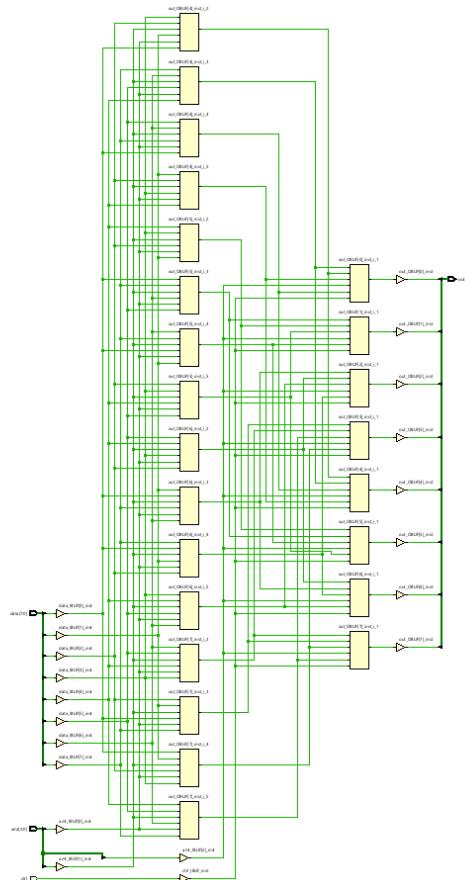
“Console”:-

```
Type=Left,data=00100100,amt=001,ctrl=1,out=01001000
Type=Left,data=01100011,amt=101,ctrl=1,out=01101100
Type=Left,data=01100101,amt=010,ctrl=1,out=10010101
Type=Left,data=00001101,amt=110,ctrl=1,out=01000011
Type=Left,data=11101101,amt=100,ctrl=1,out=11011110
Type=right,data=11000110,amt=101,ctrl=0,out=00110110
Type=right,data=11100101,amt=111,ctrl=0,out=11001011
Type=right,data=10001111,amt=010,ctrl=0,out=11100011
Type=right,data=11101000,amt=101,ctrl=0,out=01000111
Type=Left,data=10111101,amt=101,ctrl=1,out=10110111
```

“Elaborated Design”:-



“Implemented Design”:-



Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:

4.704 W

Design Power Budget:

Not Specified

Power Budget Margin:

N/A

Junction Temperature:

33.9°C

Thermal Margin:

51.1°C (27.0 W)

Effective θJA:

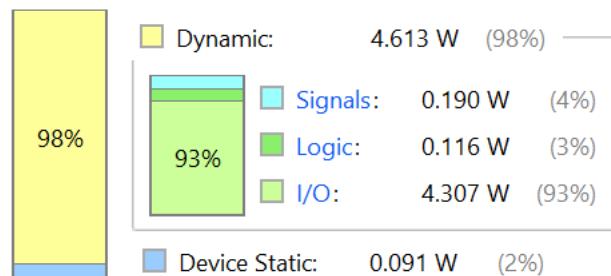
1.9°C/W

Power supplied to off-chip devices: 0 W

Confidence level: **Low**

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-40

#100DAYSRSL

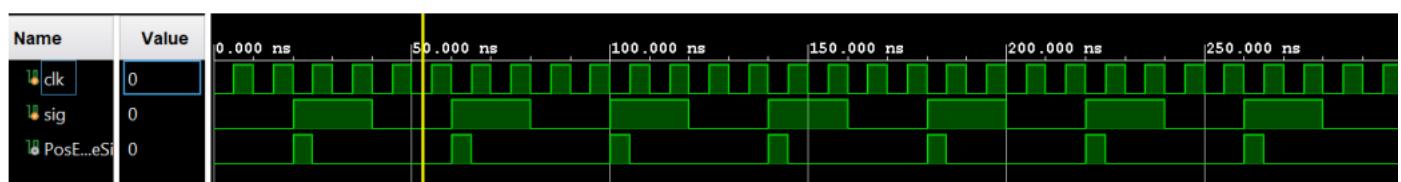
“Aim”:-To design Positive Edge detector Circuit and Negative Edge detector Circuit

“Positive Edge detector Circuit”

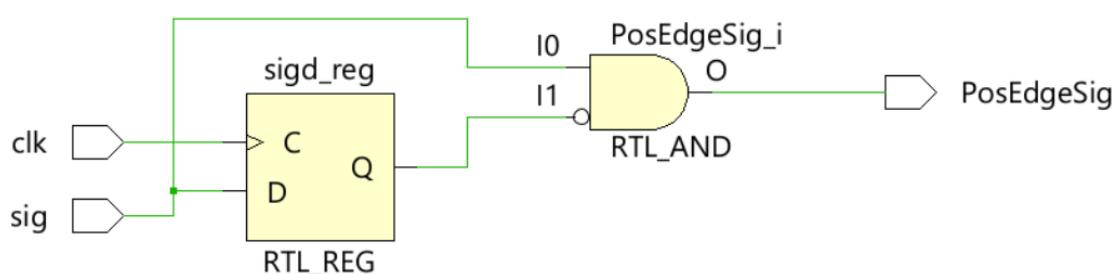
“Design Code”:-

```
module PositiveEdgeDetector(
    input clk,
    input sig, //Sample signal which we want to detect the posedge
    output PosEdgeSig
);
    reg sigd;
    always @ (posedge clk) begin // To delay the signal
        sigd <= sig;
    end
    assign PosEdgeSig = (sig && ~sigd);
endmodule
```

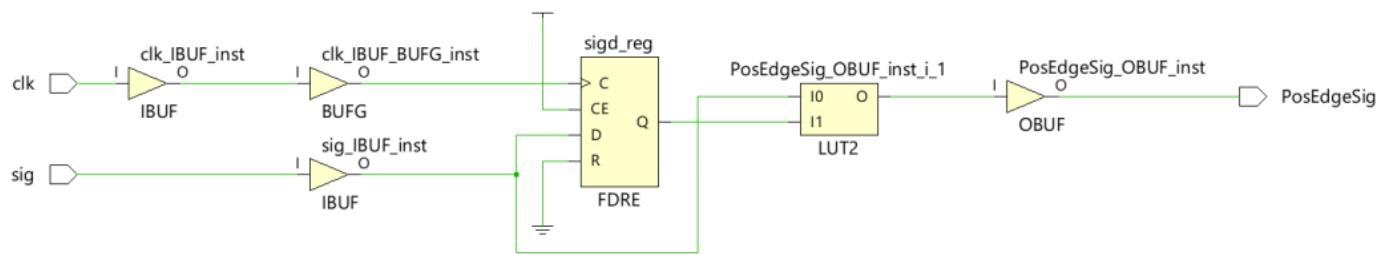
“Waveforms”:-



“Elaborated Design”:-



“Implemented Design”:-



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: 0.339 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 25.6°C

Thermal Margin: 59.4°C (31.3 W)

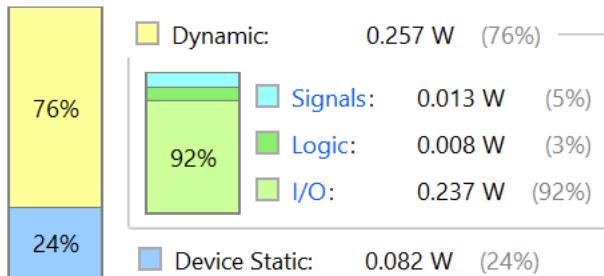
Effective θJA: 1.9°C/W

Power supplied to off-chip devices: 0 W

Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



“Negative Edge detector Circuit”

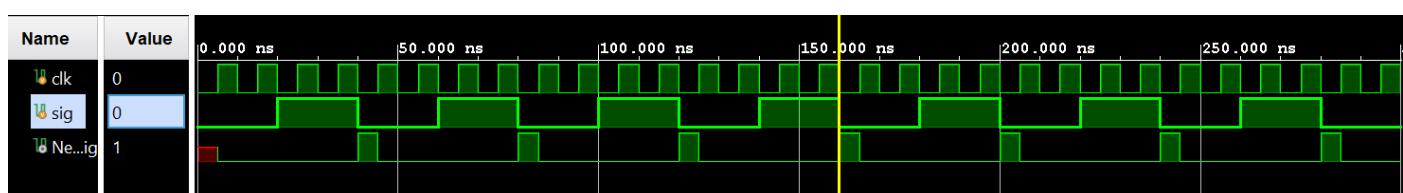
“Design Code”:-

```

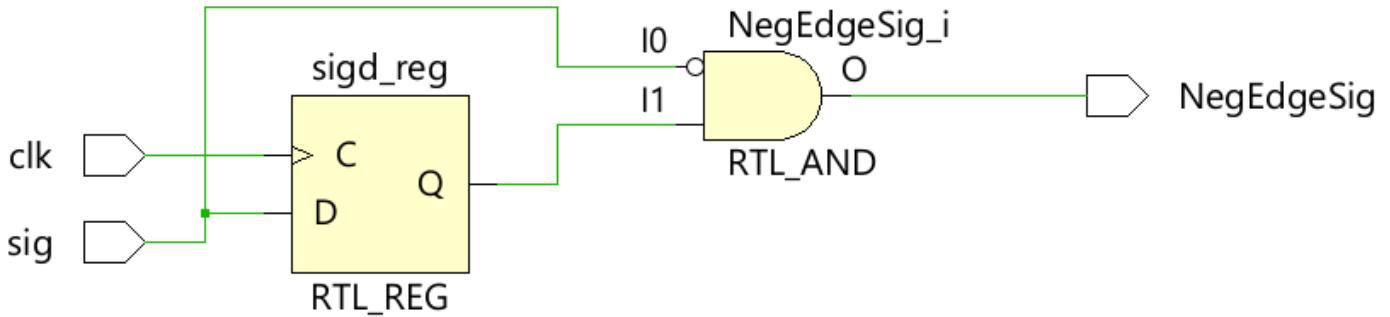
module NegativeEdgeDetector(
  input clk,
  input sig, //Sample signal which we want to detect the negativeedge
  output NegEdgeSig
);
  reg sigd;
  always @ (posedge clk) begin // To delay the signal
    sigd<=sig;
  end
  assign NegEdgeSig=(~sig && sigd);
endmodule

```

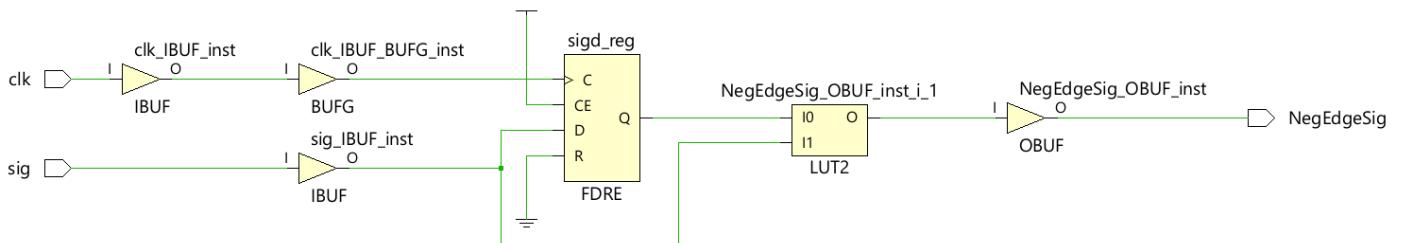
“Waveforms”:-



“Elaborated Design”:-



“Implemented Design”:-



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: **0.339 W**

Design Power Budget: **Not Specified**

Power Budget Margin: **N/A**

Junction Temperature: **25.6°C**

Thermal Margin: **59.4°C (31.3 W)**

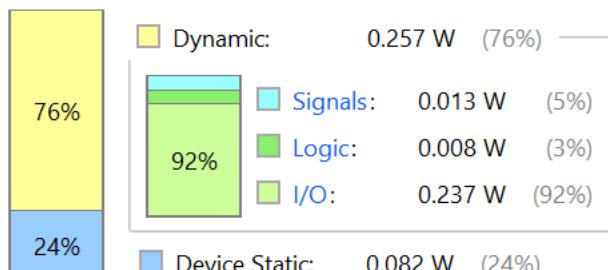
Effective θ_{JA}: **1.9°C/W**

Power supplied to off-chip devices: **0 W**

Confidence level: **Low**

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power





DAY-41

#100DAYSRSL

“Aim”:-To design a Memory of Size 1 MB 32 bit .

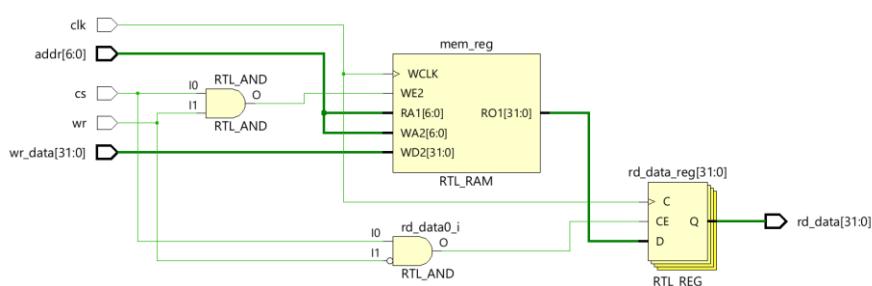
“Design Code”:-

```
module mem_1MB_32bit(
    input clk,wr,cs,
    input [6:0] addr,
    input [31:0] wr_data,
    output reg [31:0] rd_data);
    reg [31:0] mem[0:127];
    always @ (posedge clk) begin
        if(cs&wr)
            mem[addr]<=wr_data;
        end
        always @ (posedge clk)
        begin
            if(cs&(~wr))
                rd_data<=mem[addr];
            end
    endmodule
```

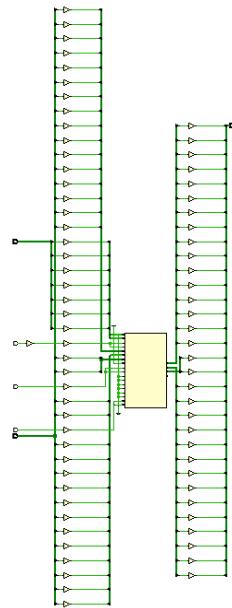
“Waveforms”:-



“Elaborated Design”:-



“Implemented Design”:-

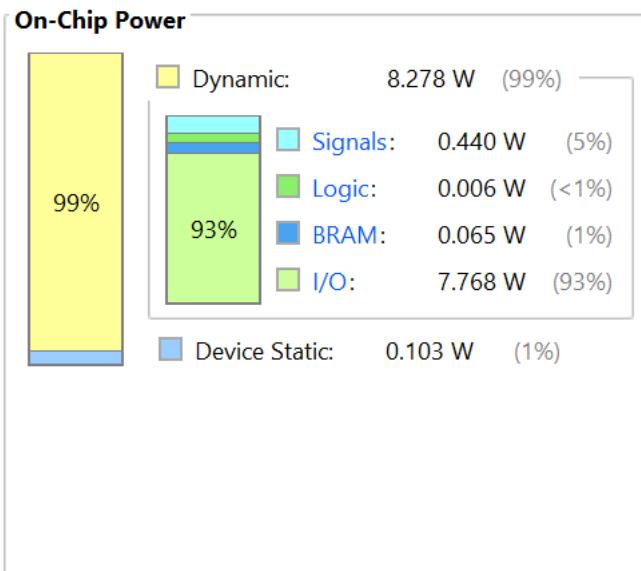


Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	8.381 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	40.8°C
Thermal Margin:	44.2°C (23.3 W)
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity



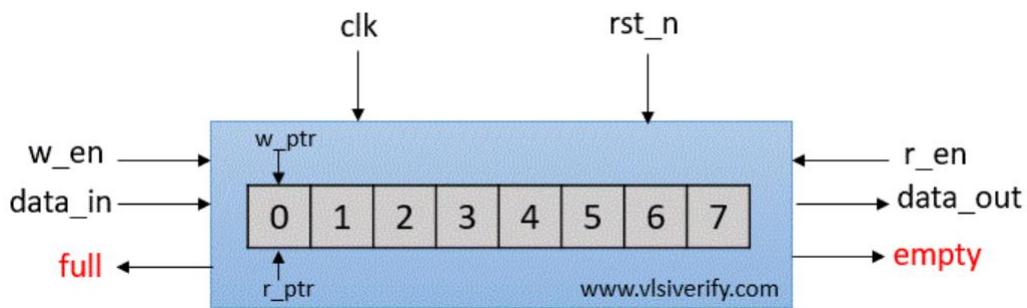


DAY-42

#100DAYSRSL

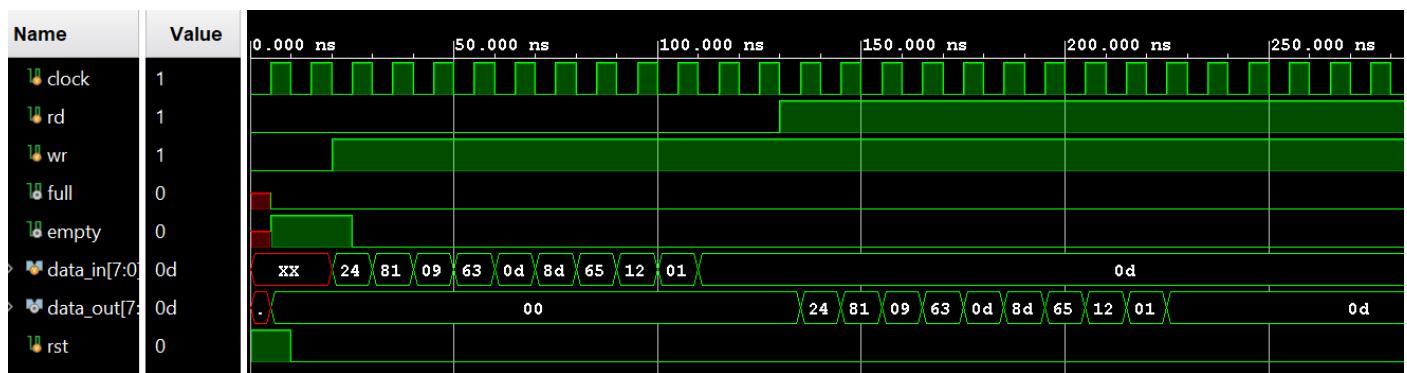
“Aim”:-To design a synchronous 32 bit depth FIFO

“Design Code”:-

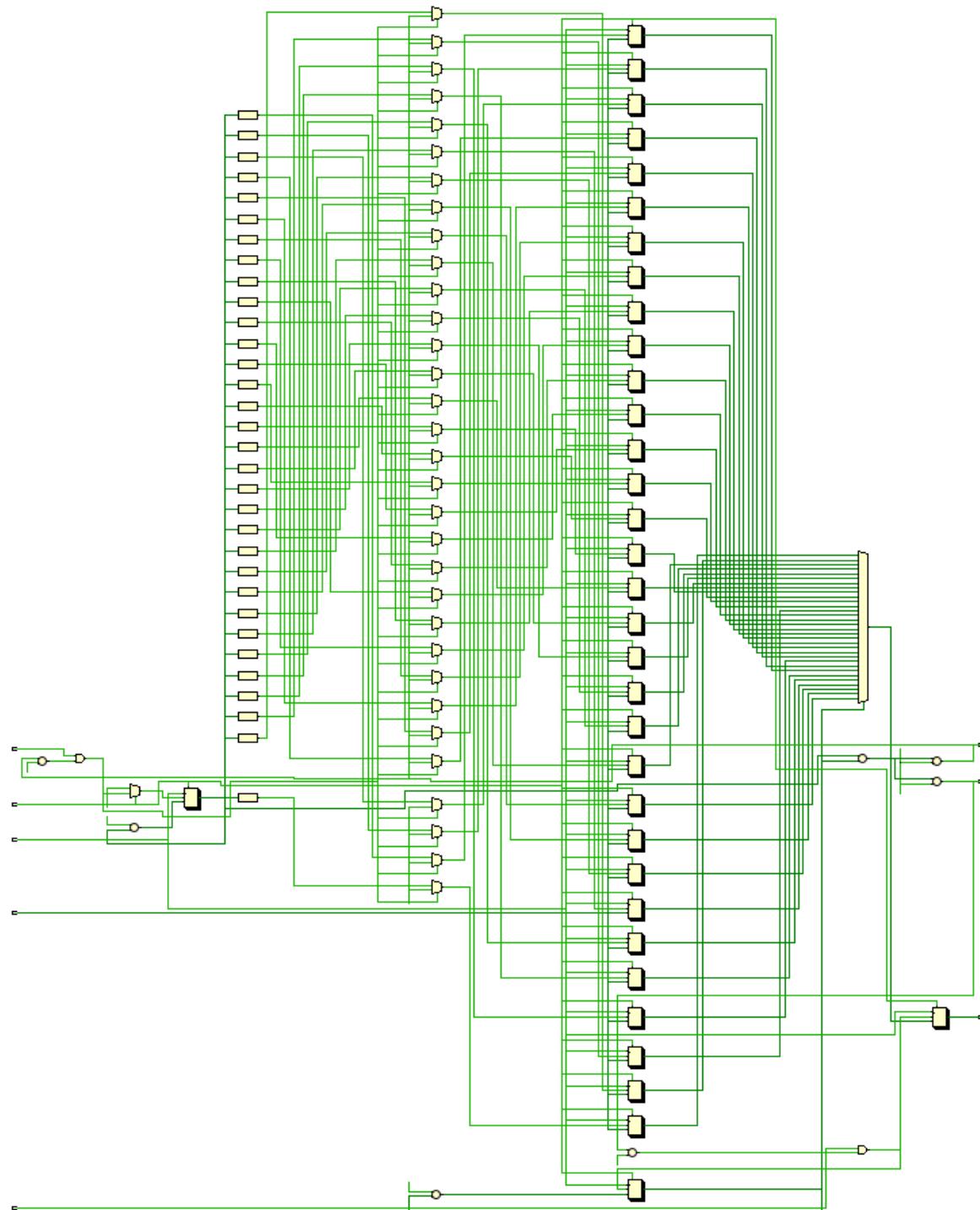


```
module fifo( input clock, rd, wr, output full, empty, input [7:0] data_in, output reg [7:0] data_out, input rst);
reg [7:0] mem [31:0];
reg [4:0] wr_ptr;
reg [4:0] rd_ptr;
always@(posedge clock)
begin
if (rst == 1'b1)
begin
data_out <= 0;
rd_ptr <= 0;
wr_ptr <= 0;
for(int i = 0; i < 32; i++) begin
mem[i] <= 0;
end
end
else
begin
if ((wr == 1'b1) && (full == 1'b0))
begin
mem[wr_ptr] <= data_in;
wr_ptr = wr_ptr + 1;
end
if((rd == 1'b1) && (empty == 1'b0))
begin
data_out <= mem[rd_ptr];
rd_ptr <= rd_ptr + 1;
end
end
end
assign empty = ((wr_ptr - rd_ptr) == 0) ? 1'b1 : 1'b0;
assign full = ((wr_ptr - rd_ptr) == 31) ? 1'b1 : 1'b0;
endmodule
```

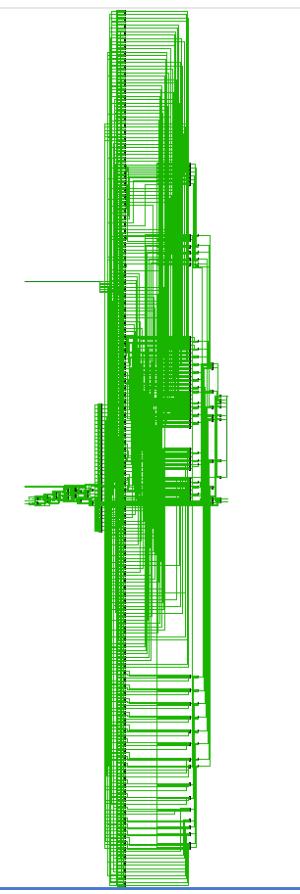
“Waveforms”:-



“Elaborated Design”:-



“Implemented Design”:-



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: 5.298 W

Design Power Budget: Not Specified

Power Budget Margin: N/A

Junction Temperature: 35.0°C

Thermal Margin: 50.0°C (26.4 W)

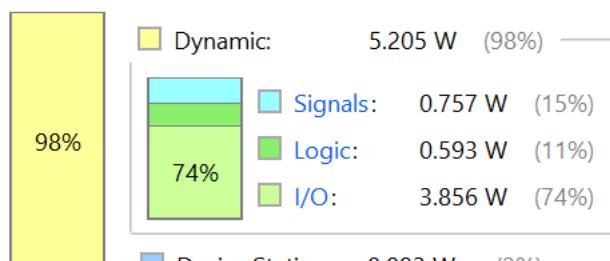
Effective θJA: 1.9°C/W

Power supplied to off-chip devices: 0 W

Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



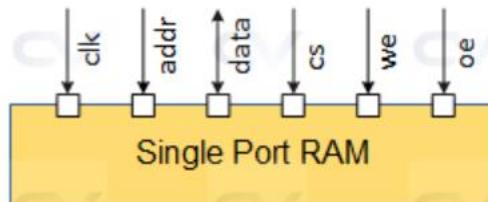


DAY-43

#100DAYSRTL

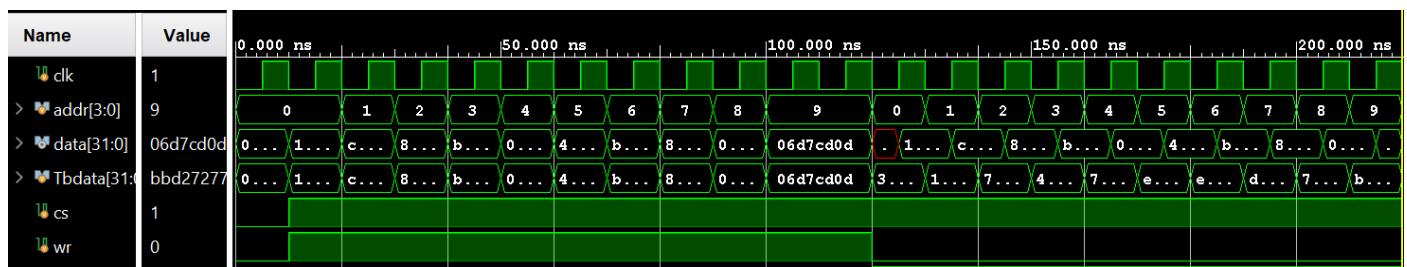
“Aim”:-To design a 32-bit Single Port RAM

“Design Code”:-

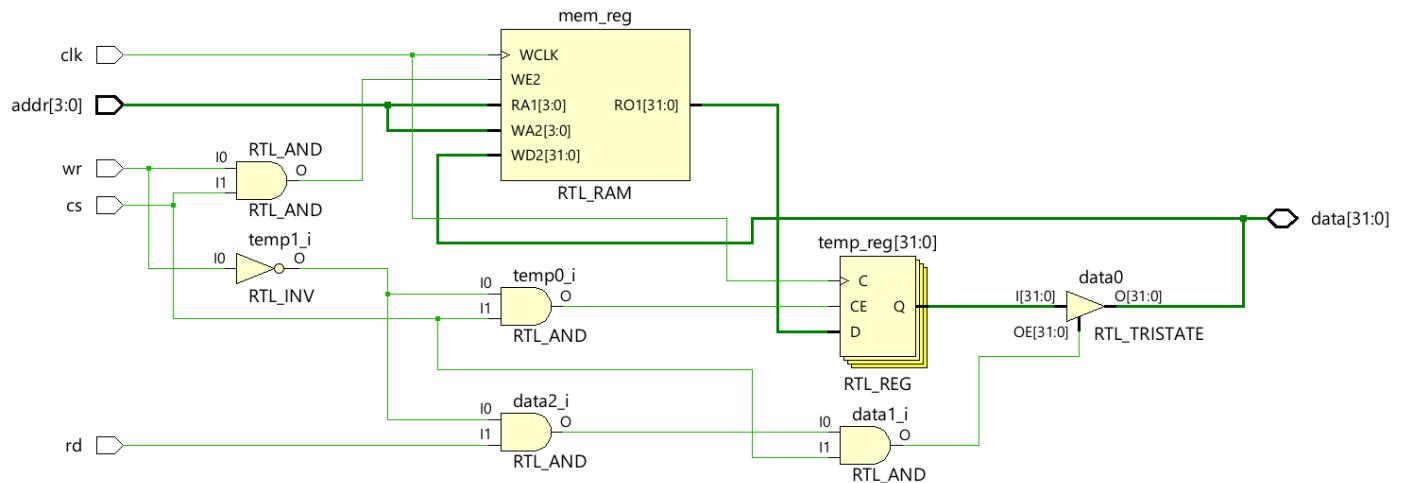


```
module SPRAM #(parameter AddrWidth=4, DataWidth=32, Depth=16) (clk,addr,data,cs,wr,rd);
input clk;
input [AddrWidth-1:0] addr;
inout [DataWidth-1:0] data;
input cs,wr,rd;
reg [DataWidth-1:0] temp;
reg [DataWidth-1:0] mem [Depth];
always @(posedge clk) begin
if(wr&&cs) mem[addr]<=data;
end
always @(posedge clk) begin
if(~wr&&cs) temp<= mem[addr];
end
assign data=((~wr)&&rd&&cs)?temp:'hz;
endmodule
```

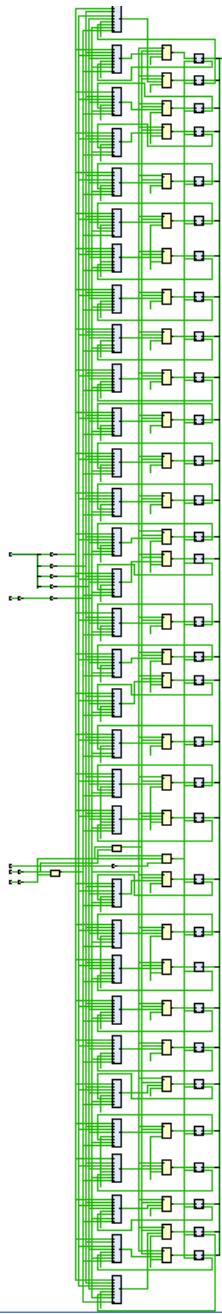
“Waveforms”:-



“Elaborated Design”:-



“Implemented Design”:-



Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power: 1.277 W

Design Power Budget: Not Specified

Power Budget Margin: N/A

Junction Temperature: 27.4°C

Thermal Margin: 57.6°C (30.4 W)

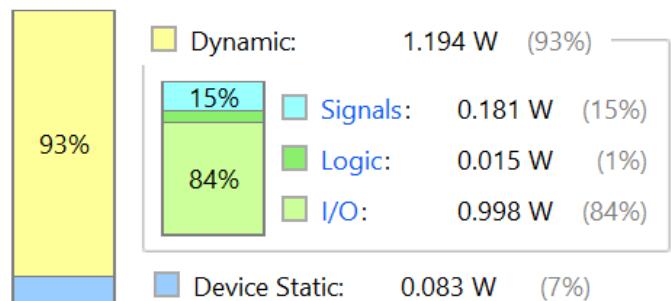
Effective θJA: 1.9°C/W

Power supplied to off-chip devices: 0 W

Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



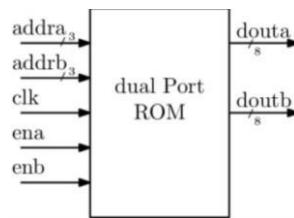


DAY-44

#100DAYSRSL

“Aim”:-To design a 32-bit Dual Port RAM

“Design Code”:-

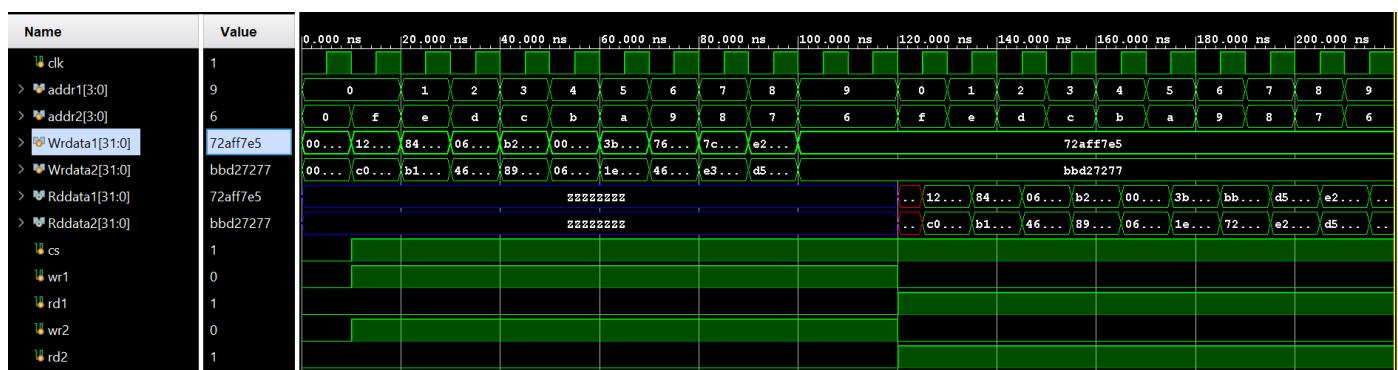


```

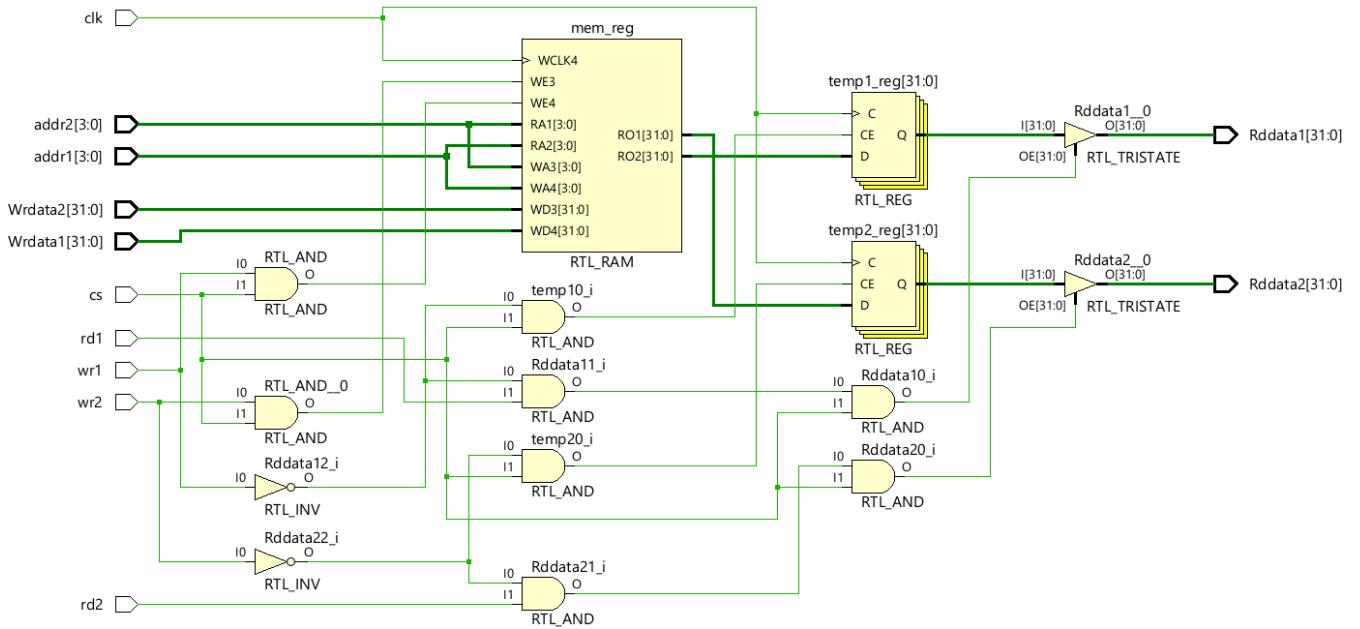
module SPRAM #(parameter AddrWidth=4, DataWidth=32, Depth=16) (clk,addr1,addr2,Wrdata1,Wrdata2,Rddata1,Rddata2,cs,wr1,rd1,wr2,rd2);
    input clk;
    input [AddrWidth-1:0] addr1;
    input [AddrWidth-1:0] addr2;
    input [DataWidth-1:0] Wrdata1;
    input [DataWidth-1:0] Wrdata2;
    output [DataWidth-1:0] Rddata1;
    output [DataWidth-1:0] Rddata2;
    input cs,wr1,rd1,wr2,rd2;
    reg [DataWidth-1:0] temp1;
    reg [DataWidth-1:0] temp2;
    reg [DataWidth-1:0] mem [Depth];
    always @(posedge clk) begin
    if(wr1&&cs) mem[addr1]<=Wrdata1;
    end
    always @(posedge clk) begin
    if(wr2&&cs) mem[addr2]<=Wrdata2;
    end
    always @(posedge clk) begin
    if(~wr1&&cs) temp1<= mem[addr1];
    end
    always @(posedge clk) begin
    if(~wr2&&cs) temp2<= mem[addr2];
    end
    assign Rddata1=((~wr1)&&rd1&&cs)?temp1:'hzz;
    assign Rddata2=((~wr2)&&rd2&&cs)?temp2:'hzz;
endmodule

```

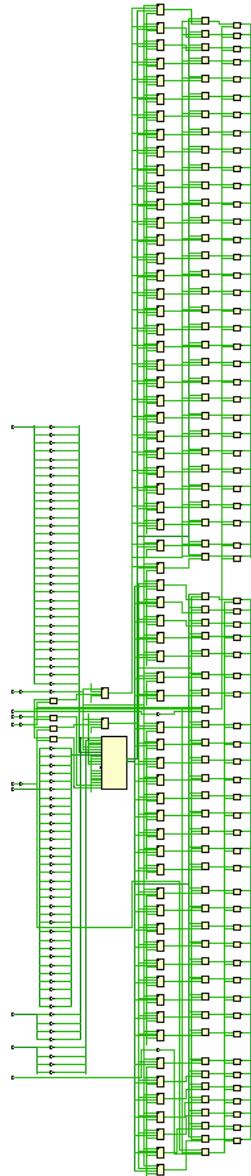
“Waveforms”:-



“Elaborated Design”:-



“Implemented Design”:-



Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:

7.743 W

Design Power Budget:

Not Specified

Power Budget Margin:

N/A

Junction Temperature:

39.6°C

Thermal Margin:

45.4°C (23.9 W)

Effective θJA:

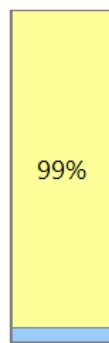
1.9°C/W

Power supplied to off-chip devices: 0 W

Confidence level: [Low](#)

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



Dynamic: 7.642 W (99%)

39%

52%

Device Static:

7.642 W (99%)

2.975 W (39%)

0.396 W (5%)

0.301 W (4%)

3.971 W (52%)

0.100 W (1%)

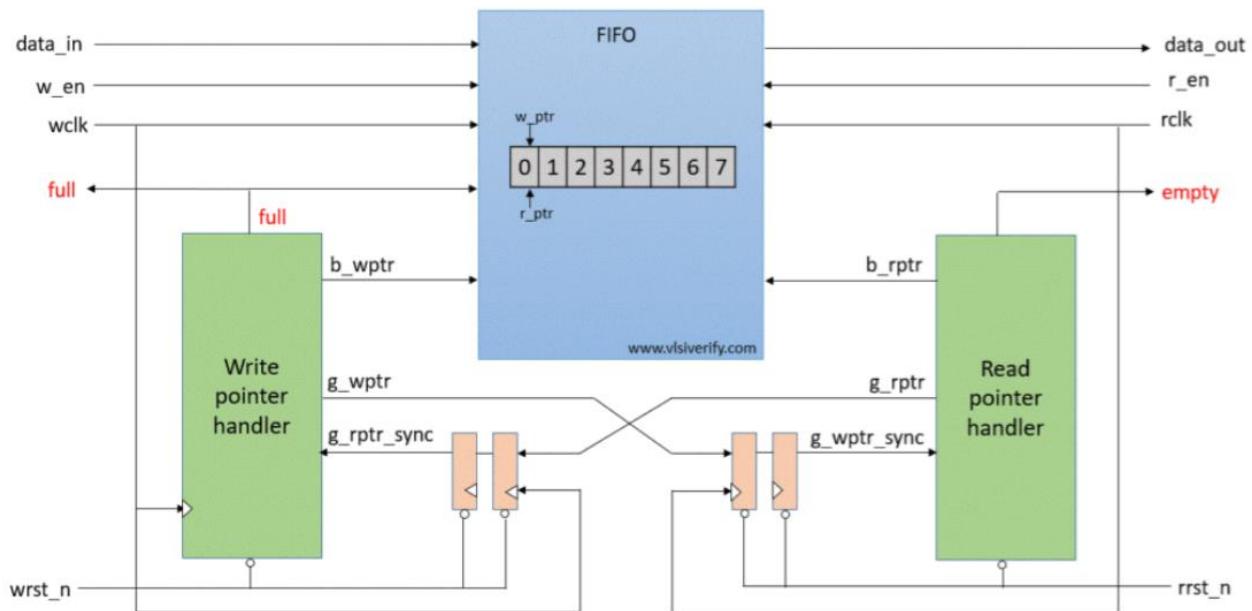


DAY-45

#100DAYSRSL

“Aim”:-To design a 16 bit depth Asynchronous FIFO

“Design Code”:-



```

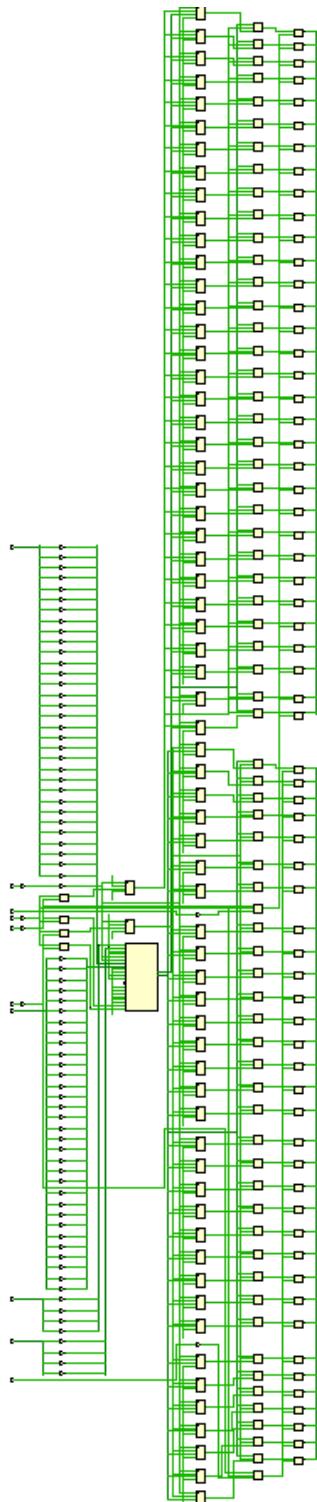
module async_fifo ( input clk_write,clk_read,rst,wr_en,rd_en,input [7:0] data_in,output reg [7:0] data_out,output reg empty,output reg full );
parameter DEPTH = 16; // Depth of the FIFO
reg [7:0] memory [0:DEPTH-1];reg [4:0] wr_ptr = 0; // Write pointer
reg [4:0] rd_ptr = 0; // Read pointer
reg [4:0] count = 0; // Number of elements in the FIFO
// Write process (synchronized to write clock)
always @(posedge clk_write or posedge rst) begin
  if (rst) begin
    wr_ptr <= 0;
    count <= 0;
  end else if (wr_en && !full) begin
    memory[wr_ptr] <= data_in;
    wr_ptr <= (wr_ptr == DEPTH-1) ? 0 : wr_ptr + 1;
    count <= count + 1;
  end
end
// Read process (synchronized to read clock)
always @(posedge clk_read or posedge rst) begin
  if (rst) begin
    rd_ptr <= 0;
    count <= 0;
  end else if (rd_en && !empty) begin
    data_out <= memory[rd_ptr];
    rd_ptr <= (rd_ptr == DEPTH-1) ? 0 : rd_ptr + 1;
    count <= count - 1;
  end
end
// Flags to indicate empty and full conditions
assign empty = (count == 0);
assign full = (count == DEPTH);
endmodule

```

“Waveforms”:-



“Implemented Design”:-



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: **4.795 W**

Design Power Budget: **Not Specified**

Power Budget Margin: **N/A**

Junction Temperature: **34.0°C**

Thermal Margin: 51.0°C (26.9 W)

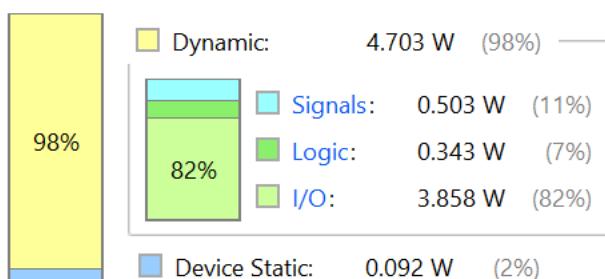
Effective θ_{JA} : 1.9°C/W

Power supplied to off-chip devices: 0 W

Confidence level: [Low](#)

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



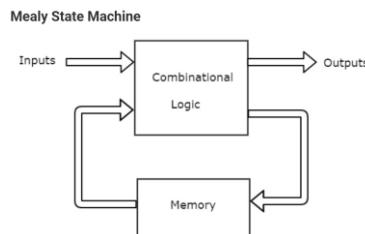


DAY-46

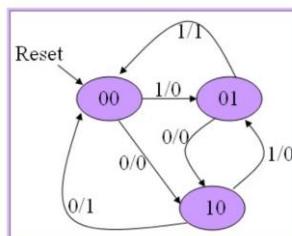
#100DAYSRSL

“Aim”:-To design a Mealy FSM sequence detector to detect the sequence 11 or 00 (Non Overlapping)

“Theory”:-



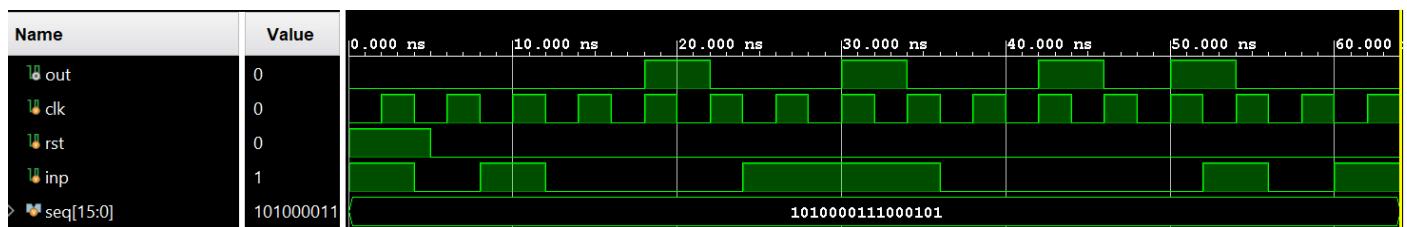
- Mealy machine depends on present state and present input



“Design Code”:-

```
module mealy(clk,rst,inp,out);
  input clk, rst;
  output reg out;
  reg [1:0] state;
  always @(posedge clk, posedge rst) begin
    if(rst) begin
      out <= 0;
      state <= 2'b00;
    end
    else begin
      case (state)
        2'b00:begin if (inp) begin state <=2'b01;out <=0; end
                  else begin state <=2'b10;out <=0; end
                end
        2'b01:begin if(inp) begin state <= 2'b00;out <= 1; end
                  else begin state <= 2'b10;out <= 0; end
                end
        2'b10:begin if(inp) begin state <= 2'b01;out <= 0; end
                  else begin state <= 2'b00;out <= 1; end
                end
        default:begin state <= 2'b00; out <= 0; end
      endcase
    end
  end
endmodule
```

“Waveforms”:-



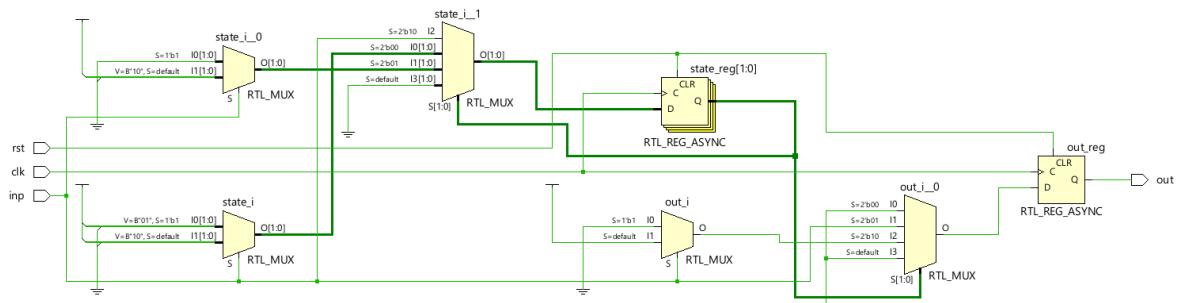
“Console”:-

```

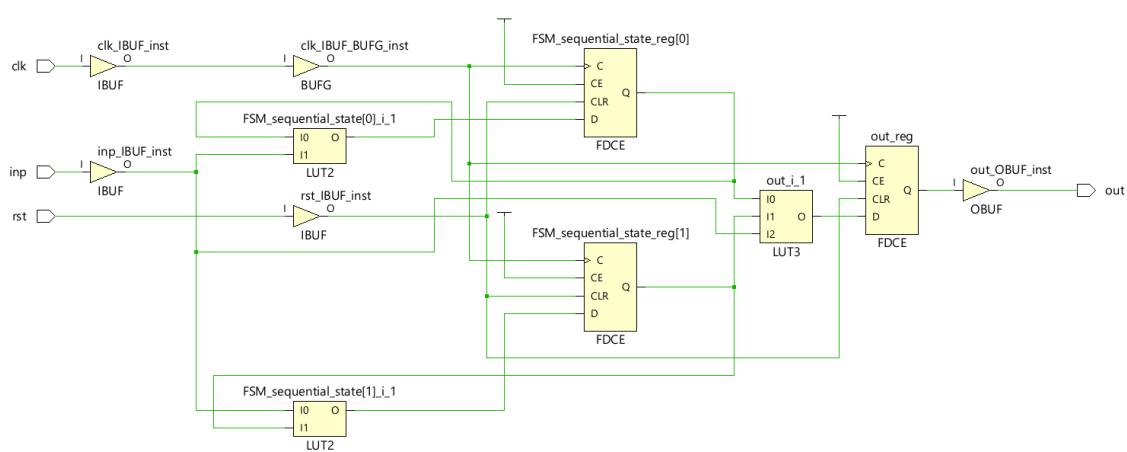
state = 0| input = 1| output = 0
state = 2| input = 0| output = 0
state = 1| input = 1| output = 0
state = 2| input = 0| output = 0
state = 0| input = 0| output = 1
state = 2| input = 0| output = 0
state = 1| input = 1| output = 0
state = 0| input = 1| output = 1
state = 1| input = 1| output = 0
state = 2| input = 0| output = 0
state = 0| input = 0| output = 1
state = 2| input = 0| output = 0
state = 0| input = 0| output = 1
state = 1| input = 1| output = 0
state = 2| input = 0| output = 0
state = 1| input = 1| output = 0

```

“Elaborated Design”:-



“Implemented Design”:-



Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:

0.859 W

Design Power Budget:

Not Specified

Power Budget Margin:

N/A

Junction Temperature:

26.6°C

Thermal Margin:

58.4°C (30.8 W)

Effective θJA:

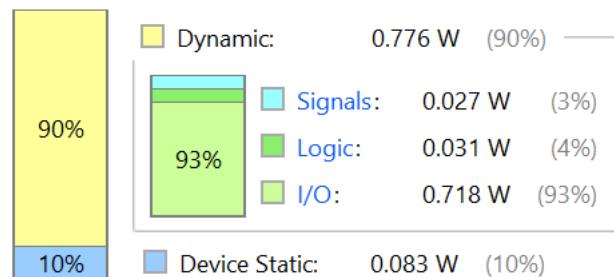
1.9°C/W

Power supplied to off-chip devices: 0 W

Confidence level: [Low](#)

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



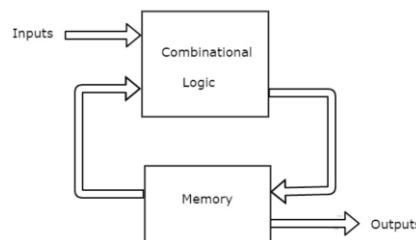


DAY-47

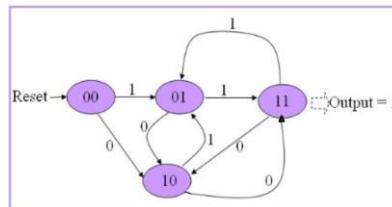
#100DAYSRSL

“Aim”:- To design a Moore FSM sequence detector to detect the sequence 11 or 00 (Non Overlapping)

“Theory”:-



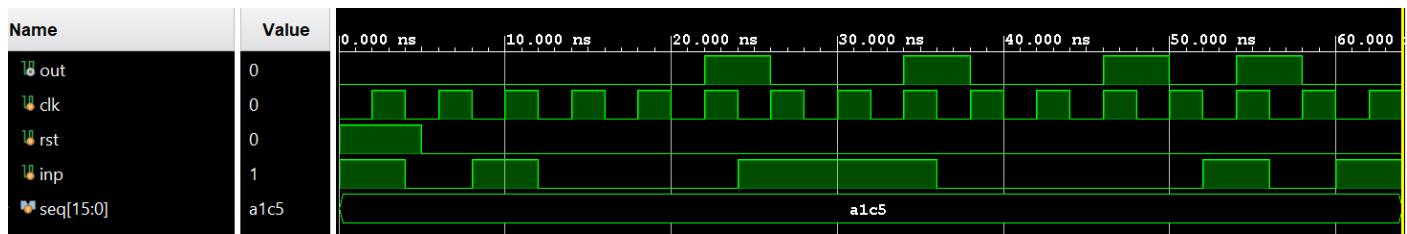
- Moore FSM depends on the present state only



“Design Code”:-

```
module moore(clk, rst, inp, out);
    output out;
    input clk, rst, inp;
    reg out;
    reg [1:0] state;
    always @(posedge clk, posedge rst) begin
        if (rst) begin state <= 2'b00 ; out <= 0; end
        else begin
            case(state)
                2'b00: begin if (inp) state <= 2'b01;
                           else state <= 2'b10;
                           end
                2'b01: begin if (inp) state <= 2'b11;
                           else state <= 2'b10;
                           end
                2'b10: begin if (inp) state <= 2'b01;
                           else state <= 2'b11;
                           end
                2'b11: begin if (inp) state <= 2'b01;
                           else state <= 2'b10;
                           end
                default:state <= 2'b00;
            endcase
        end
    end
    always @(posedge clk) begin
        if (state == 2'b11)  out <= 1;
        else out <= 0;
    end
endmodule
```

“Waveforms”:-



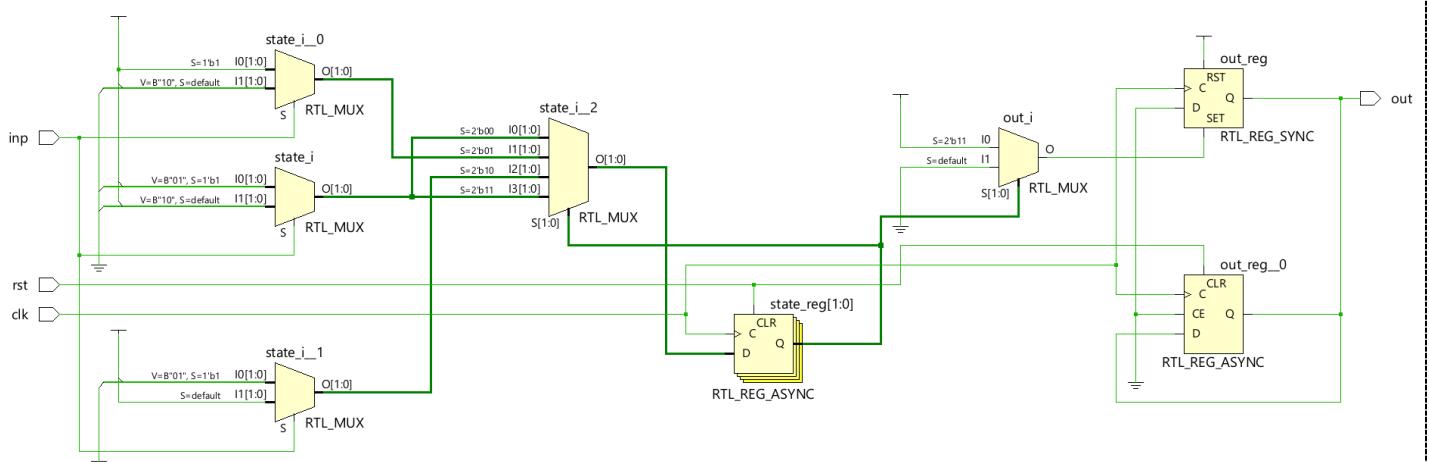
“Console”:-

```

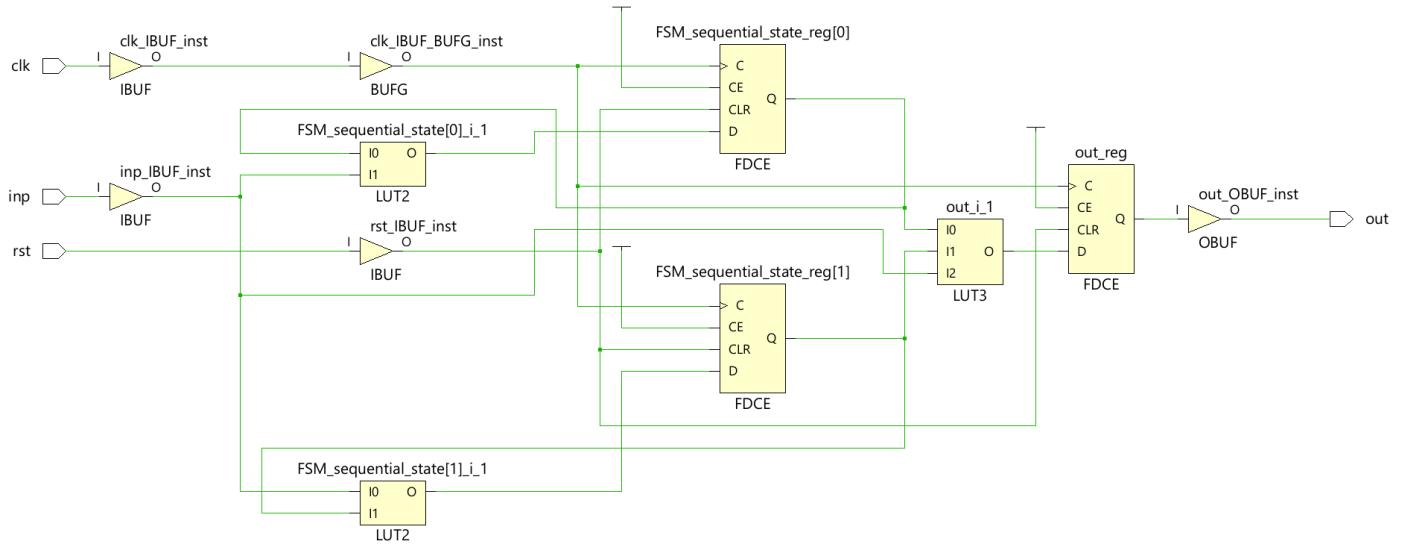
state = 0| input = 1| output = 0
state = 2| input = 0| output = 0
state = 1| input = 1| output = 0
state = 2| input = 0| output = 0
state = 3| input = 0| output = 0
state = 2| input = 0| output = 1
state = 1| input = 1| output = 0
state = 3| input = 1| output = 0
state = 1| input = 1| output = 1
state = 2| input = 0| output = 0
state = 3| input = 0| output = 0
state = 2| input = 0| output = 1
state = 3| input = 0| output = 0
state = 1| input = 1| output = 1
state = 2| input = 0| output = 0
state = 1| input = 1| output = 0

```

“Elaborated design”:-



“Implemented design”:-

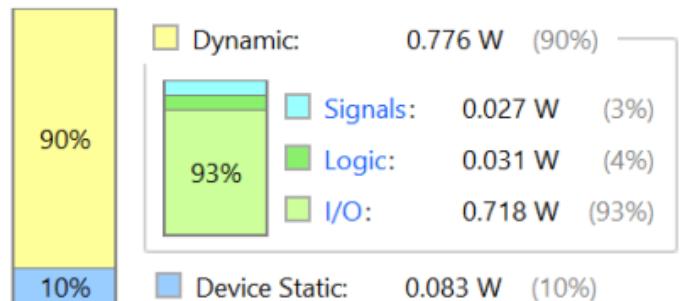


Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	0.859 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	26.6°C
Thermal Margin:	58.4°C (30.8 W)
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low
Launch Power Constraint Advisor to find and fix invalid switching activity	

On-Chip Power



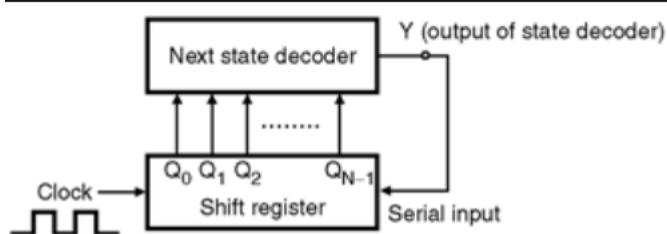


DAY-48

#100DAYSRSL

“Aim”:- To design a Sequence Generator which generates starting five prime numbers (2,3,5,7,11)

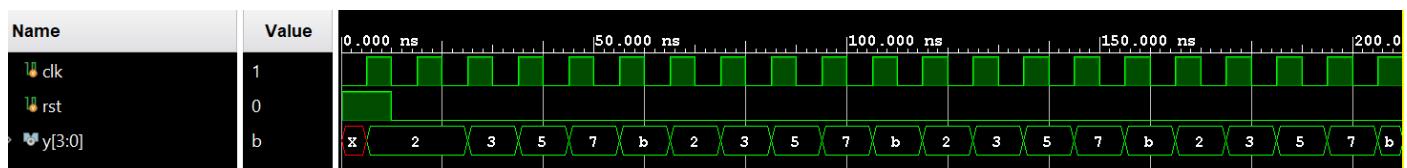
“Theory”:-



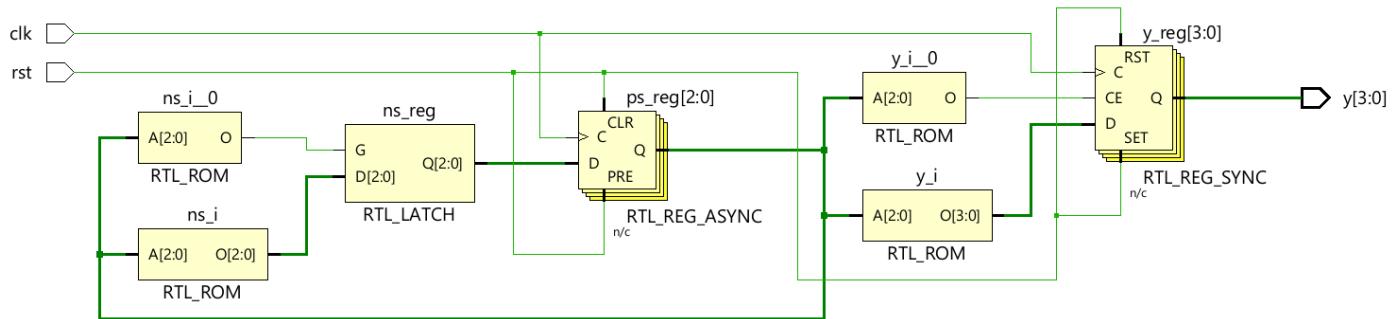
“Design Code”:-

```
module fsm_bl(input clk,rst,output reg [3:0] y;
reg [2:0] ps,ns;
parameter s1=3'd1,s2=3'd2, s3=3'd3, s4=3'd4, s5=3'd5;
always @(*) begin
case(ps)
s1:ns=s2;
s2:ns=s3;
s3:ns=s4;
s4:ns=s5;
s5:ns=s1;
endcase end
always @ (posedge clk or posedge rst) begin
if (rst) ps<=s1;
else
ps<=ns; end
always @ (posedge clk) begin
if(rst) y<=2;
else
begin
case(ps)
s1:y<=2;
s2:y<=3;
s3:y<=5;
s4:y<=7;
s5:y<=11;
endcase
end
end
endmodule
```

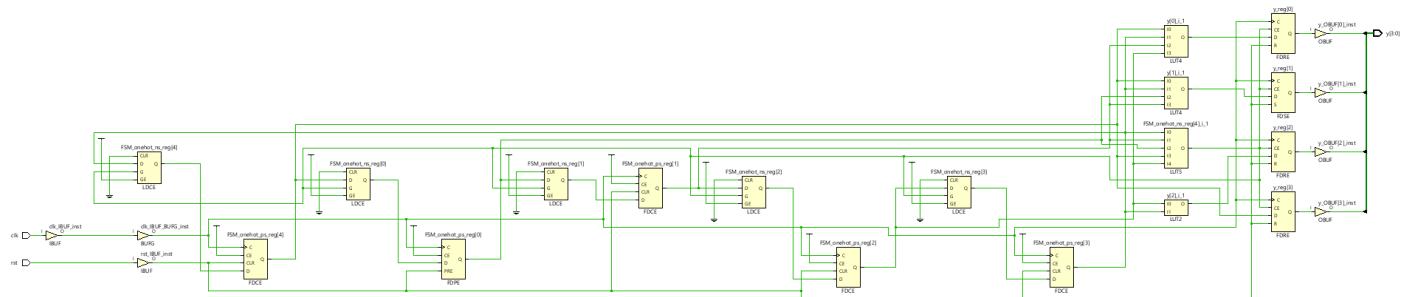
“Waveforms”:-



“Elaborated design”:-



“Implemented design”:-



Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:

0.1 W

Design Power Budget:

Not Specified

Power Budget Margin:

N/A

Junction Temperature:

25.2°C

Thermal Margin:

59.8°C (31.6 W)

Effective θ_{JA}:

1.9°C/W

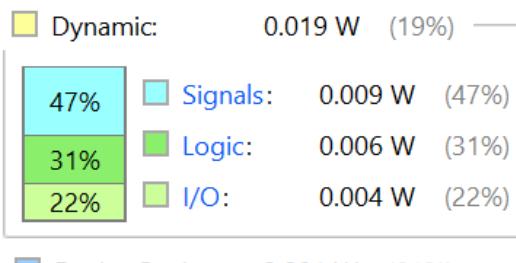
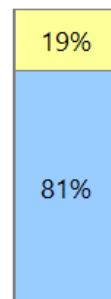
Power supplied to off-chip devices: 0 W

Confidence level:

Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



Device Static: 0.081 W (81%)

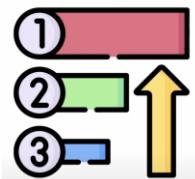


DAY-49

#100DAYSRSL

“Aim”:- To design a Strict or Fixed priority arbiter.

“Theory”:-

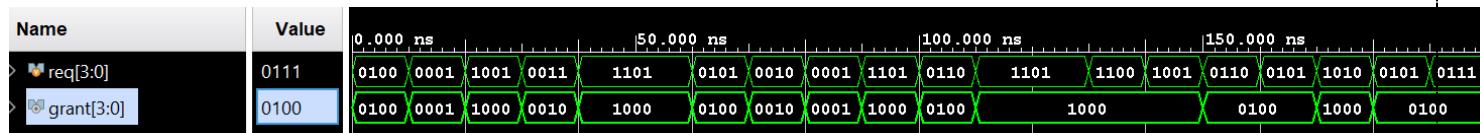


- A fixed priority arbiter selects one of the requesters based on a predefined priority scheme. In a fixed priority arbiter, the highest priority request is granted access first. If multiple requests have the same priority, the arbiter selects one of the requests in a round-robin fashion.

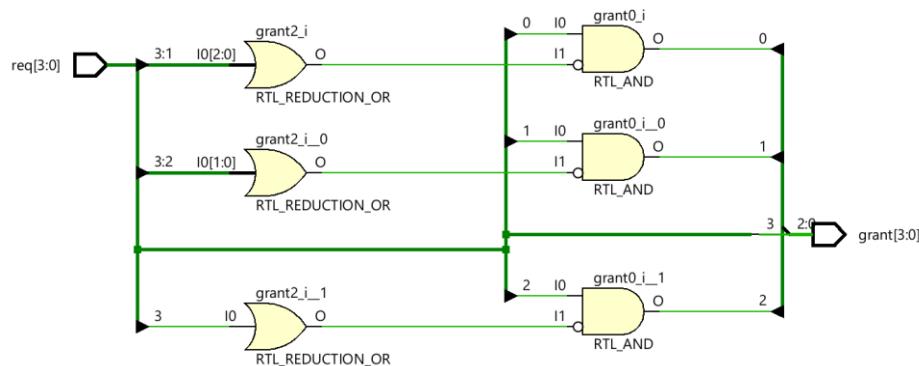
“Design Code”:-

```
module FPA #(parameter NumPorts=4) (req,grant);
    input [NumPorts-1:0] req;
    output [NumPorts-1:0] grant;
    assign grant[3]=req[3];
    genvar i;
    for(i=2; i>=0; i=i-1) begin
        assign grant[i] = req[i] && (~((req[3:i+1])));
    end
endmodule
```

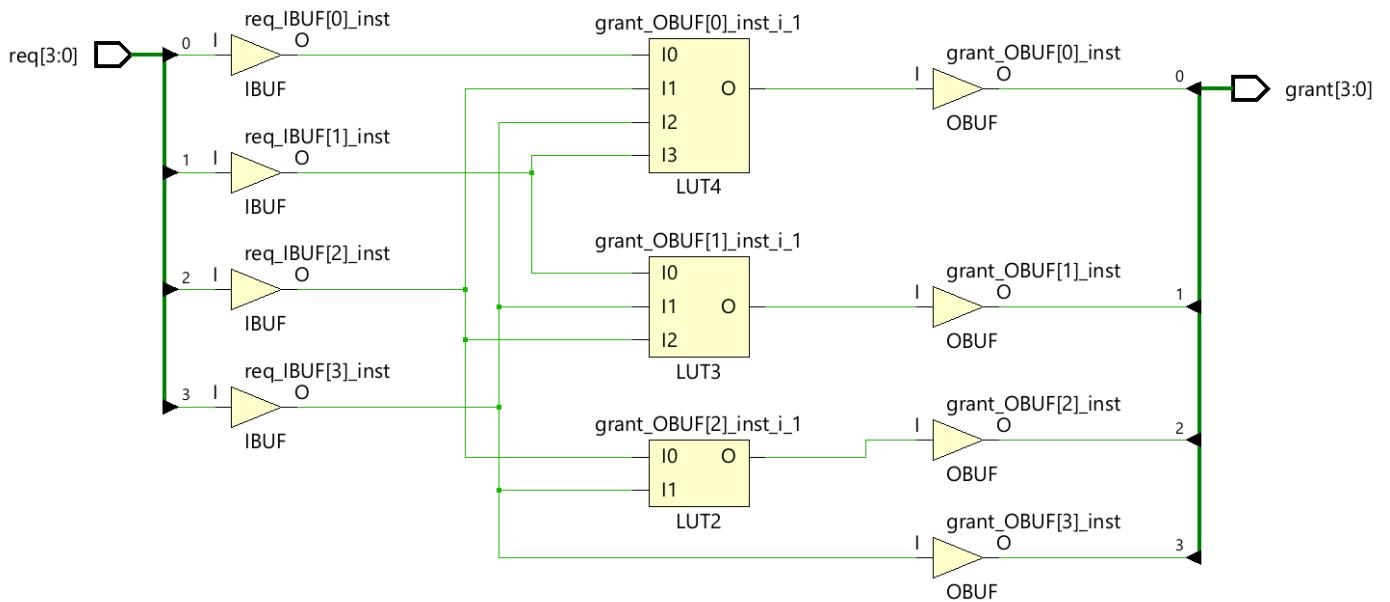
“Waveforms”:-



“Elaborated design”:-



“Implemented design”:-



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: 0.872 W
Design Power Budget: Not Specified

Power Budget Margin: N/A
Junction Temperature: 26.6°C

Thermal Margin: 58.4°C (30.8 W)

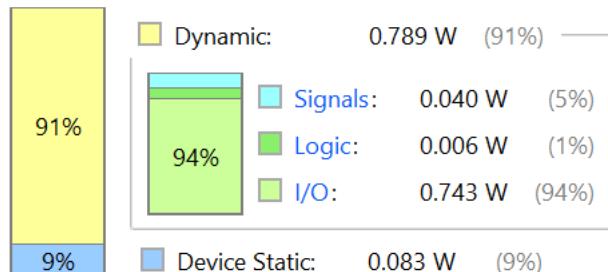
Effective θ_{JA}: 1.9°C/W

Power supplied to off-chip devices: 0 W

Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



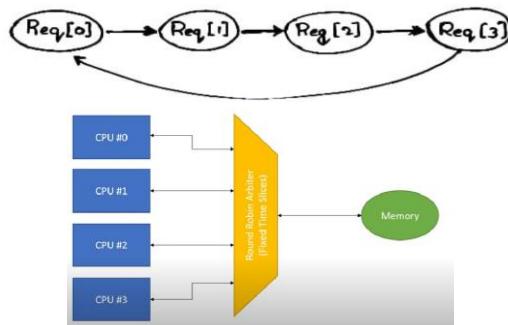


DAY-50

#100DAYSRSL

“Aim”:- To design a Round Robin Arbiter

“Theory”:-



- An arbiter in which the priority of requests is set in which all the requested agents get equal sharing of access is called Round Robin Arbiter

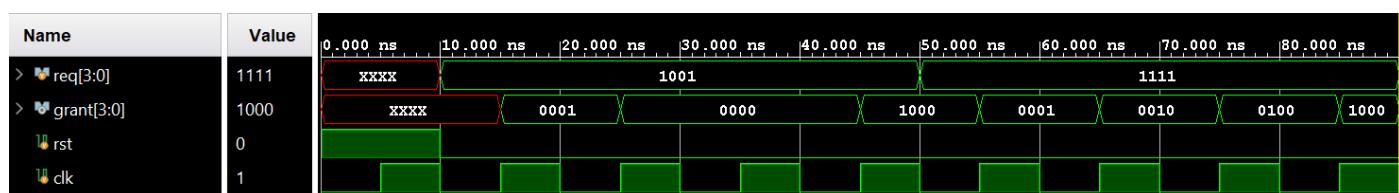
• “Design Code”:-

```

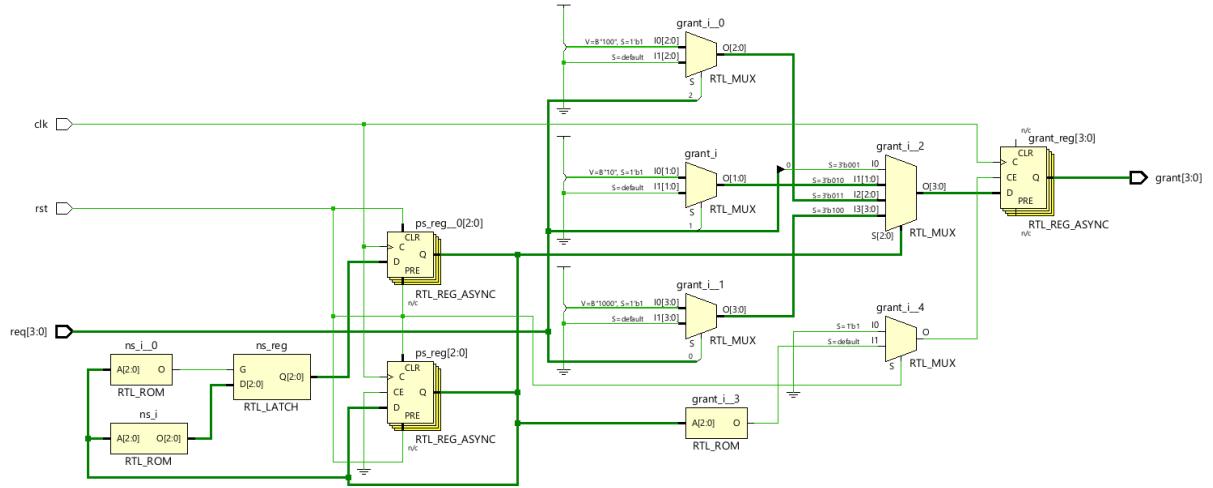
module RRA(clk,rst,req,grant);
input clk,rst;input [3:0] req;output reg [3:0] grant;
reg [2:0] ps,ns;
parameter state1=3'd1,state2=3'd2,state3=3'd3,state4=3'd4;
always @(*) begin
  case (ps)
    state1:ns=state2;
    state2:ns=state3;
    state3:ns=state4;
    state4:ns=state1;
  endcase
end
always @(posedge clk or posedge rst) begin
  if(rst) ps<=state1; else ps<=ns; end
always @(posedge clk or posedge rst) begin
  if (rst) ps<=state1; else begin
    case (ps)
      state1:begin if(req[0])grant<=4'b0001;
        else grant<=4'b0000;
      end
      state2:begin if(req[1])grant<=4'b0010;
        else grant<=4'b0000;
      end
      state3:begin if(req[2])grant<=4'b0100;
        else grant<=4'b0000;
      end
      state4:begin if(req[0])grant<=4'b1000;
        else grant<=4'b0000;
      end
    endcase
  end
end
end
endmodule

```

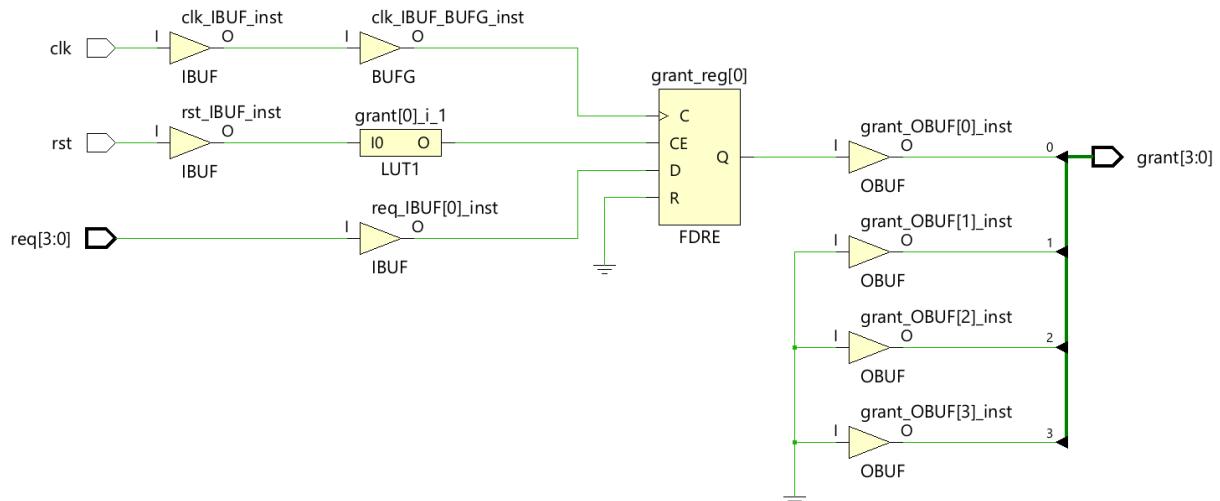
“Waveforms”:-



“Elaborated design”:-



“Implemented design”:-



Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:

0.346 W

Design Power Budget:

Not Specified

Power Budget Margin:

N/A

Junction Temperature:

25.7°C

Thermal Margin:

59.3°C (31.3 W)

Effective θ_{JA} :

1.9°C/W

Power supplied to off-chip devices:

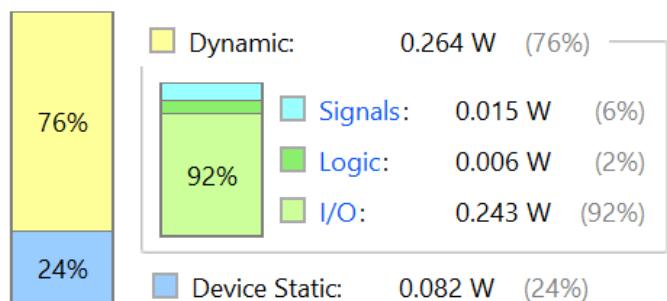
0 W

Confidence level:

Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



THANK YOU !



The VLSI Voyager

chinna venkata narayana reddy seelam
cvnreddyseelam07@gmail.com