



DIGITAL SYSTEM DESIGN - ASSIGNMENT 3 PART I

Name: Rohit Dilip Baruah

ID No.: 21EL015

Division: 4

Year: 2023-24

Subject: Digital System Design (3EL42)

Branch: Electronics (EL)

Code 1: Clock Divider

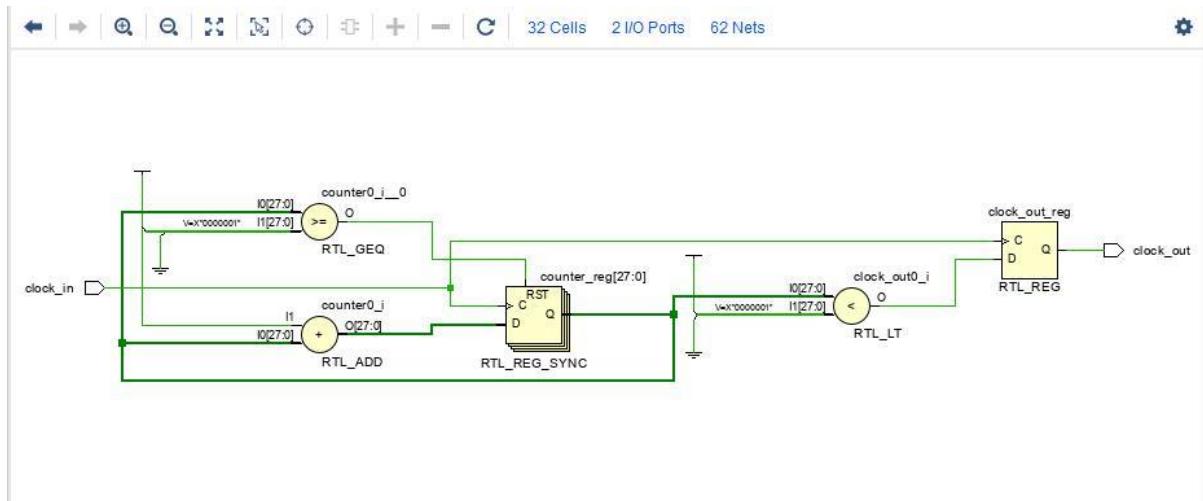
Verilog Code:

```
module Clock_divider(
    input clock_in,
    output reg clock_out
);
    reg[27:0] counter=28'd0;
    parameter DIVISOR = 28'd2;
    always @(posedge clock_in)
    begin
        counter <= counter + 28'd1;
        if(counter>=(DIVISOR-1))
            counter <= 28'd0;
        clock_out <= (counter<DIVISOR/2)?1'b1:1'b0;
    end
endmodule
```

Test Bench:

```
module tb_clock_divider;
    reg clock_in;
    wire clock_out;
    clock_divider uut (
        .clock_in(clock_in),
```

RTL Schematic:



Synthesis Report:

```

source clock_divider.tcl -notrace
Command: synth_design -top clock_divider -part xc7k70tfbv676-1
Starting synth_design
Attempting to get a license for feature 'Synthesis' and/or device 'xc7k70t'
INFO: [Common 17-349] Got license for feature 'Synthesis' and/or device 'xc7k70t'
INFO: [Synth 8-7079] Multithreading enabled for synth_design using a maximum of 2 processes.
INFO: [Synth 8-7078] Launching helper process for spawning children vivado processes
INFO: [Synth 8-7075] Helper process launched with PID 11392
-----
Starting Synthesize : Time (s): cpu = 00:00:04 ; elapsed = 00:00:07 . Memory (MB): peak = 1046.367 ; gain = 0.000
-----
INFO: [Synth 8-6157] synthesizing module 'clock_divider' [E:/projects/dsd/clock_divider/1/clock_divider_1.srcs/sources_1/net
Parameter DIVISOR bound to: 28'b00000000000000000000000000000010
INFO: [Synth 8-6155] done synthesizing module 'clock_divider' (1#1) [E:/projects/dsd/clock_divider/1/clock_divider_1.srcs/s
-----
Finished Synthesize : Time (s): cpu = 00:00:05 ; elapsed = 00:00:08 . Memory (MB): peak = 1046.367 ; gain = 0.000

```

Report BlackBoxes:

BlackBox name	Instances

Report Cell Usage:

Cell	Count
BUFG	1
CARRY4	7
LUT1	1
LUT4	1
LUT5	2
LUT6	8
FDRE	29
TRUF	11

```

Report Instance Areas:
+-----+-----+-----+
|     |Instance |Module |Cells |
+-----+-----+-----+
|1    |top      |      | 51|
+-----+-----+-----+
-----+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:15 ; elapsed = 00:00:29 . Memory (MB): peak = 1046.367 ; gain = 0
+-----+-----+-----+
|4    |LUT4    |      | 1|
|5    |LUT5    |      | 2|
|6    |LUT6    |      | 8|
|7    |FDRE    |      | 29|
|8    |IBUF    |      | 1|
|9    |OBUF    |      | 1|
+-----+-----+-----+
-----+
Report Instance Areas:
+-----+-----+-----+
|     |Instance |Module |Cells |
+-----+-----+-----+
|1    |top      |      | 51|
+-----+-----+-----+
-----+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:15 ; elapsed = 00:00:29 . Memory (MB): peak = 1046.367 ; gain = 0
+-----+

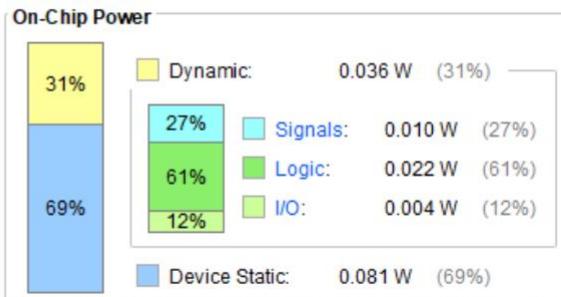
```

Power Report:

Summary

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

Total On-Chip Power:	0.117 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	25.2°C
Thermal Margin:	59.8°C (31.5 W)



Total On-Chip Power:	0.117 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	25.2°C
Thermal Margin:	59.8°C (31.5 W)
Effective 9JA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low



Code 2: Johnson Counter

Verilog Code:

```
`timescale 1ns / 1ps

module johnson_counter(
    input clk,
    input reset,
    output [3:0] out
);
reg [3:0] q;

always @(posedge clk)
begin
if(reset)
    q=4'd0;
else
    begin
        q[3]<=q[2];
        q[2]<=q[1];
        q[1]<=q[0];
        q[0]<=(~q[3]);
    end
end
end

assign out=q;
endmodule
```

Test Bench:

```
`timescale 1ns / 1ps

module jc_tb;
    reg clk,reset;
    wire [3:0] out;

    johnson_counter dut (.out(out), .reset(reset), .clk(clk));

    always
        #5 clk =~clk;

    initial begin
        reset=l'b1; clk=l'b0;
        #20 reset= l'b0;
    end

    initial
        begin

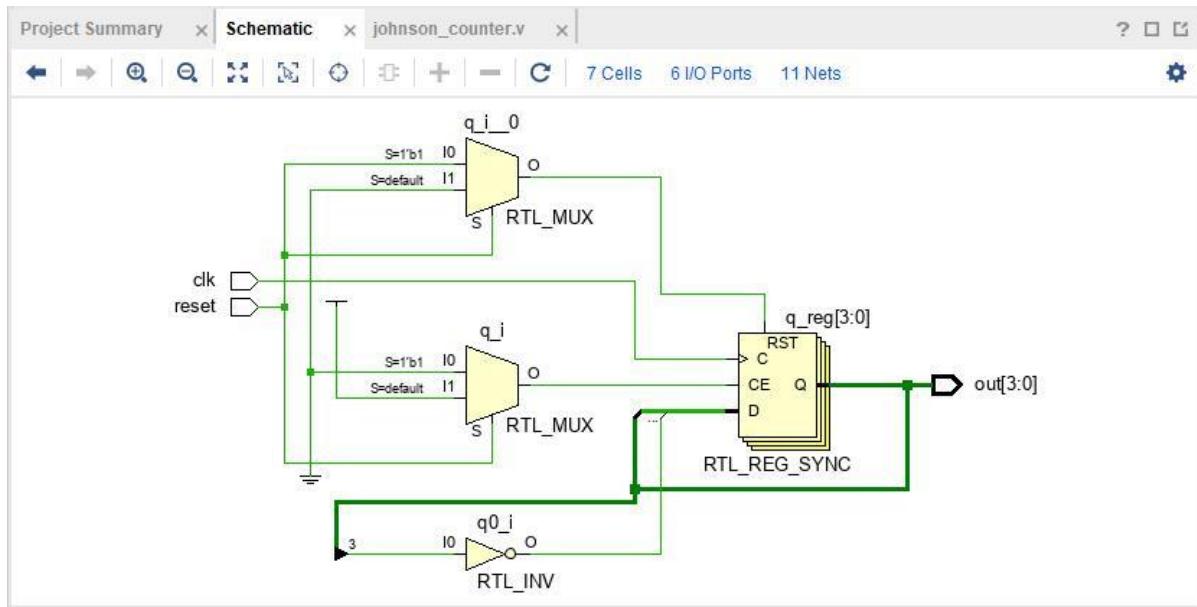
    johnson_counter dut (.out(out), .reset(reset), .clk(clk));

    always
        #5 clk =~clk;

    initial begin
        reset=l'b1; clk=l'b0;
        #20 reset= l'b0;
    end

    initial
        begin
            $monitor( $time, " clk=%b, out= %b, reset=%b", clk,out,reset);
            #105 $stop;
        end
endmodule
```

RTL Schematic:



Synthesis Report:

```
-----
Start Writing Synthesis Report
-----
Report BlackBoxes:
++-----+-----+
| |BlackBox name |Instances |
++-----+-----+
++-----+-----+
Report Cell Usage:
++-----+-----+
| |Cell |Count |
++-----+-----+
|1 |BUFGE | 1|
|2 |LUT1 | 1|
|3 |FDRE | 4|
|4 |IBUF | 2|
|5 |OBUF | 4|
++-----+-----+
Report Instance Areas:
++-----+-----+-----+
| |Instance |Module |Cells |
++-----+-----+-----+
|1 |top | | 12|
++-----+-----+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:16 ; elapsed = 00:00:30 . Memory (MB): peak = 1018.973 ; gain = 0.000
```

Power Report:

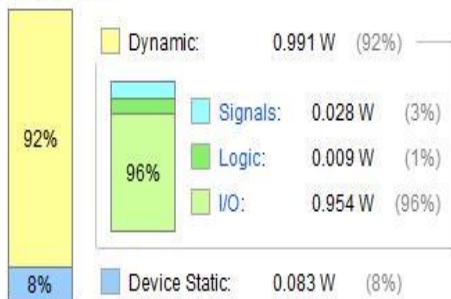
Summary

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

Total On-Chip Power:	1.074 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	27.0°C
Thermal Margin:	58.0°C (30.6 W)
Effective θ _{JA} :	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

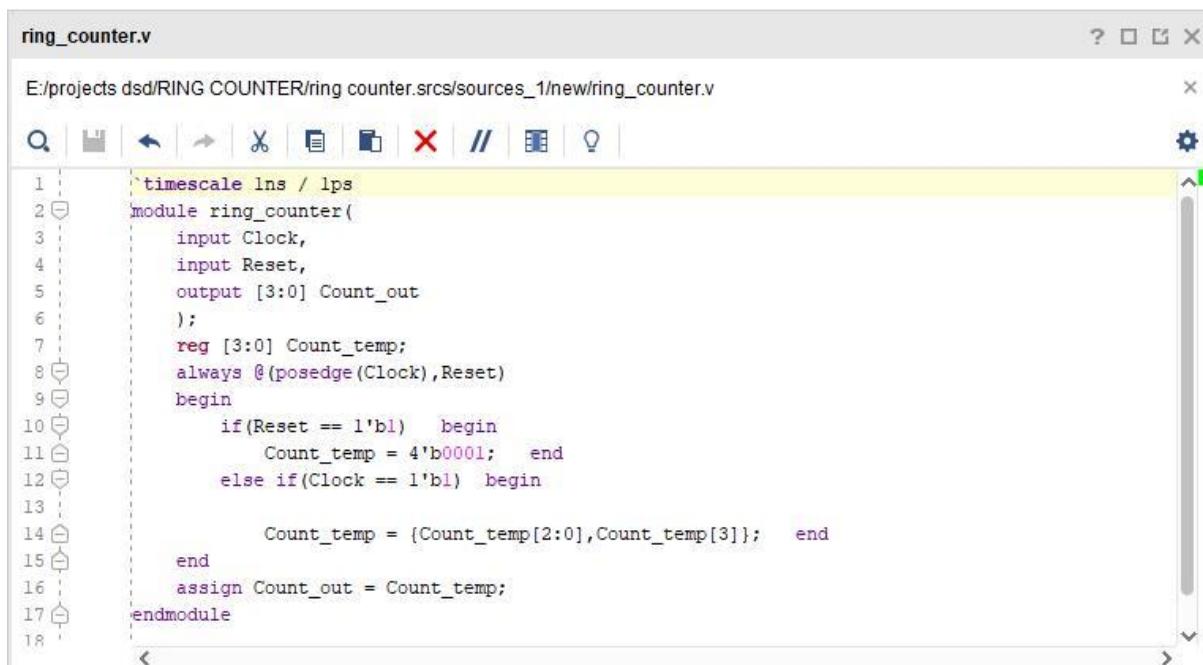
[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



Code 3: Ring Counter

Verilog Code:

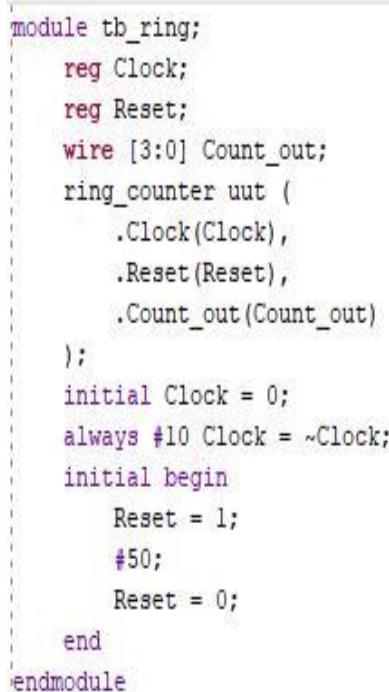


The screenshot shows a Verilog code editor window titled "ring_counter.v". The code is a simple ring counter module. It includes a timescale declaration, a module definition with input Clock and Reset, and output Count_out. Inside the module, there is a reg variable Count_temp and an always block that increments Count_temp on each rising edge of the clock. The code is numbered from 1 to 18.

```
ring_counter.v
E:/projects dsd/RING COUNTER/ring counter.srsc/sources_1/new/ring_counter.v

1  `timescale 1ns / 1ps
2  module ring_counter(
3      input Clock,
4      input Reset,
5      output [3:0] Count_out
6  );
7      reg [3:0] Count_temp;
8      always @(posedge(Clock),Reset)
9      begin
10         if(Reset == 1'b1) begin
11             Count_temp = 4'b0001; end
12         else if(Clock == 1'b1) begin
13             Count_temp = {Count_temp[2:0],Count_temp[3]}; end
14         end
15         assign Count_out = Count_temp;
16     endmodule
17
18 
```

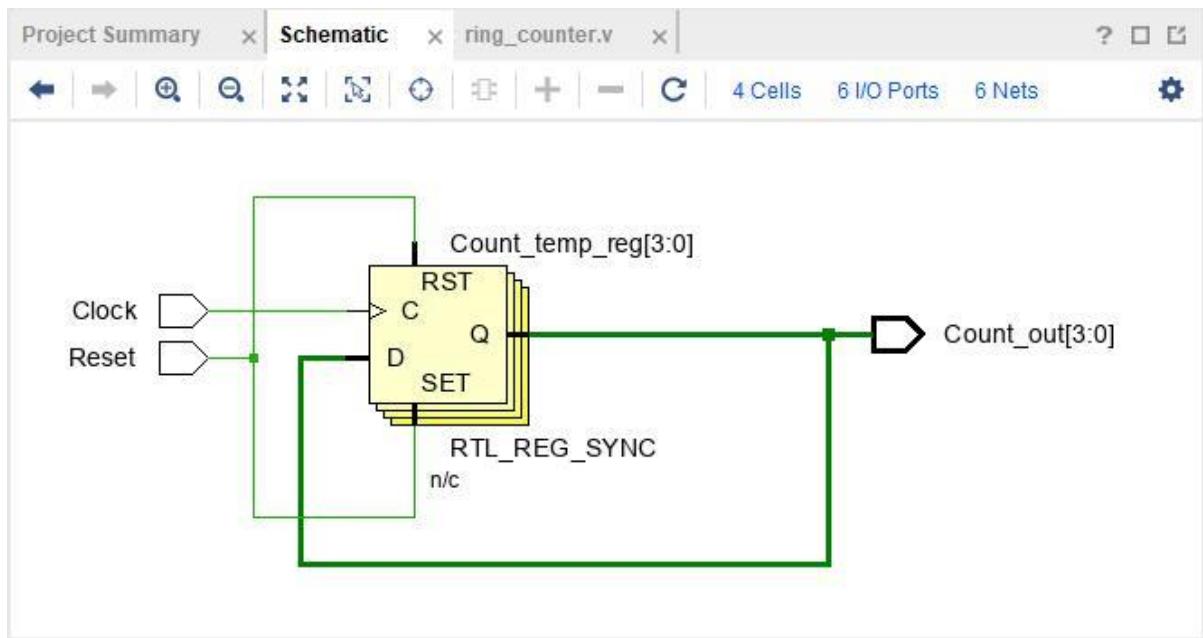
Test Bench:



The screenshot shows a Verilog test bench module named tb_ring. It includes declarations for reg variables Clock and Reset, and a wire variable Count_out. It also includes a ring_counter instance and logic to drive the clock and reset signals. The test bench begins with an initial condition where Clock is 0, followed by an always block that toggles the clock every 10 units of time. It then enters an initial block where it sets Reset to 1, waits 50 units, and then sets Reset to 0, which triggers the counter's reset logic.

```
tb_ring.v
module tb_ring;
    reg Clock;
    reg Reset;
    wire [3:0] Count_out;
    ring_counter uut (
        .Clock(Clock),
        .Reset(Reset),
        .Count_out(Count_out)
    );
    initial Clock = 0;
    always #10 Clock = ~Clock;
    initial begin
        Reset = 1;
        #50;
        Reset = 0;
    end
endmodule
```

RTL Schematic:



Synthesis Report:

```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
+---+---+  
| |BlackBox name |Instances |  
+---+---+  
+---+---+  
+---+---+  
  
Report Cell Usage:  
+---+---+  
| |Cell |Count |  
+---+---+  
|1 |BUFGE | 1|  
|2 |FDRE | 3|  
|3 |FDSE | 1|  
|4 |IBUF | 2|  
|5 |OBUF | 4|  
+---+---+  
  
Report Instance Areas:  
+---+---+---+  
| |Instance |Module |Cells |  
+---+---+---+  
|1 |top | | 11|  
+---+---+---+  
  
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:17 ; elapsed = 00:00:31 . Memory (MB): peak = 1019.336 ; gain = 0.000
```

Power Report:

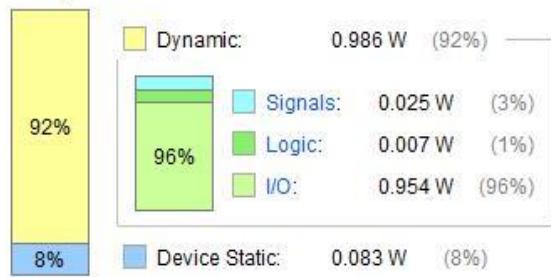
Summary

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

Total On-Chip Power:	1.069 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	27.0°C
Thermal Margin:	58.0°C (30.6 W)
Effective 9JA:	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



Code 4: Five Input Majority Circuit

Verilog Code:

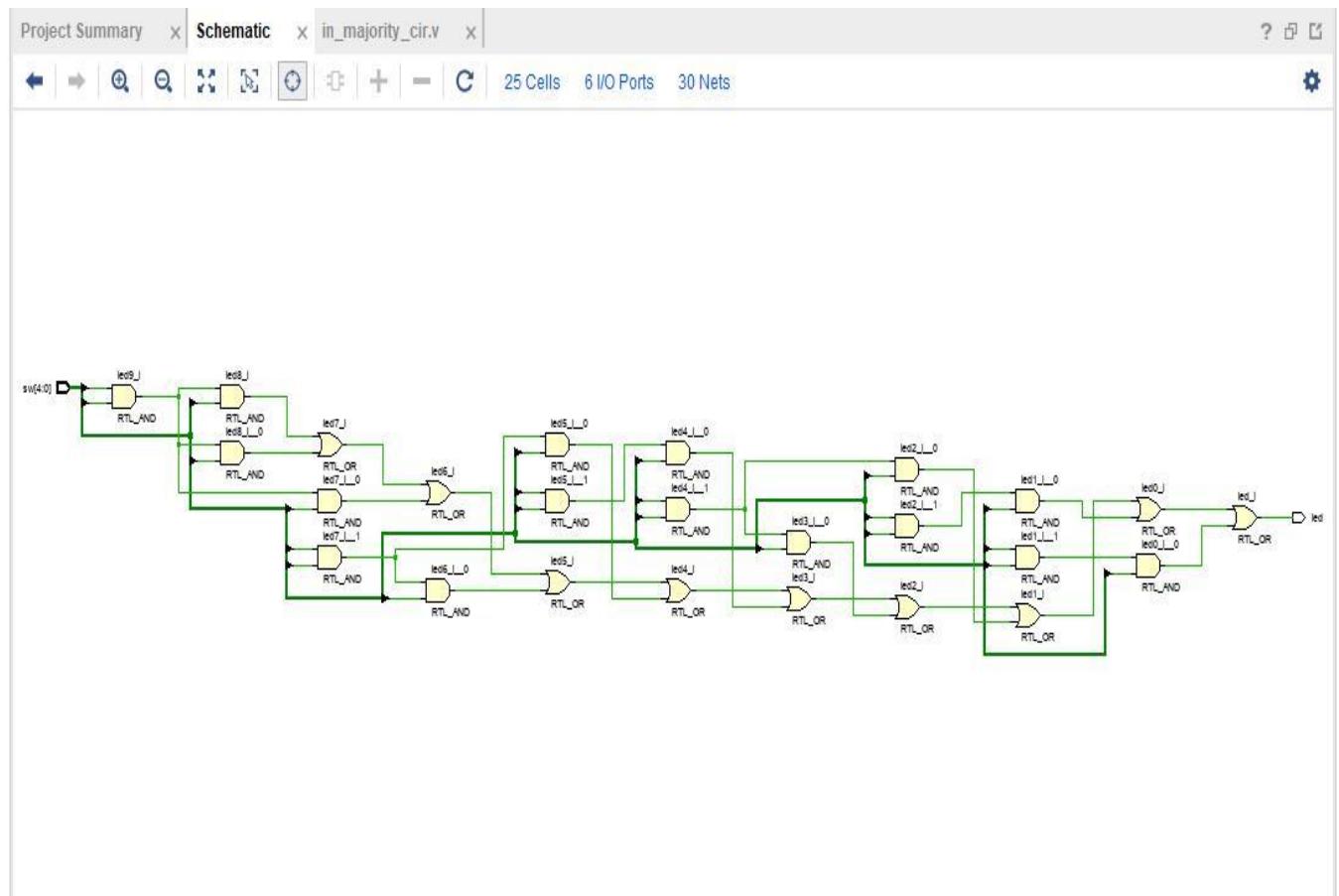
```
module majority_of_five(
    input [4:0] sw,
    output led
);
assign led = (sw[0] & sw[1] & sw[2]) |
            (sw[0] & sw[1] & sw[3]) |
            (sw[0] & sw[1] & sw[4]) |
            (sw[0] & sw[2] & sw[3]) |
            (sw[0] & sw[2] & sw[4]) |
            (sw[0] & sw[3] & sw[4]) |
            (sw[1] & sw[2] & sw[3]) |
            (sw[1] & sw[2] & sw[4]) |
            (sw[1] & sw[3] & sw[4]) |
            (sw[2] & sw[3] & sw[4]);
endmodule
```

Test Bench:

```
module majority_of_five_tb;
reg [4:0] sw;
wire led;
majority_of_five cut (.sw(sw),.led(led));
integer k;

initial
begin
    sw = 0;
    for (k=0; k<32; k=k+1)
        #20 sw = k;
    #20 $finish;
end
endmodule
```

RTL Schematic:



Synthesis Report:

```
-----  
Start Writing Synthesis Report  
-----
```

```
Report BlackBoxes:
```

BlackBox name	Instances

```
Report Cell Usage:
```

Cell	Count
LUT5	1
IBUF	5
OBUF	1

```
Report Instance Areas:
```

Instance	Module	Cells
top		7

```
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:16 ; elapsed = 00:00:27 . Memory (MB): peak = 1018.273 ; gain = 0.000  
-----
```

Power Report:

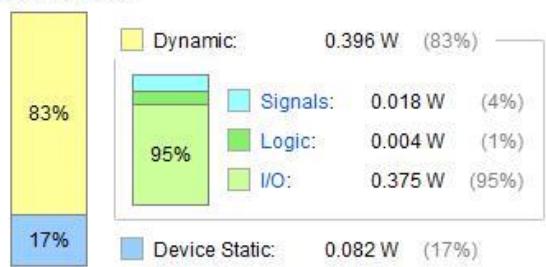
Summary

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

Total On-Chip Power:	0.478 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 9JA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

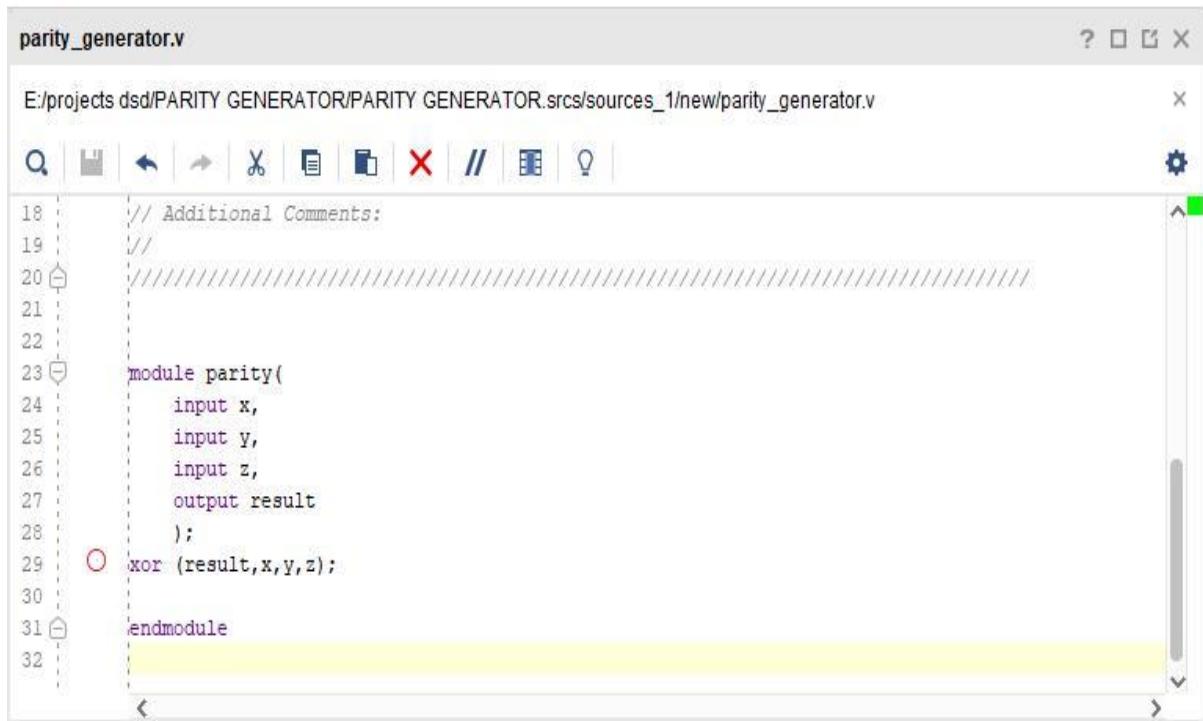
[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



Code 5: Parity Generator

Verilog Code:



The screenshot shows a Verilog code editor window titled "parity_generator.v". The code is as follows:

```
// Additional Comments:  
//  
module parity(  
    input x,  
    input y,  
    input z,  
    output result  
);  
    xor (result,x,y,z);  
endmodule
```

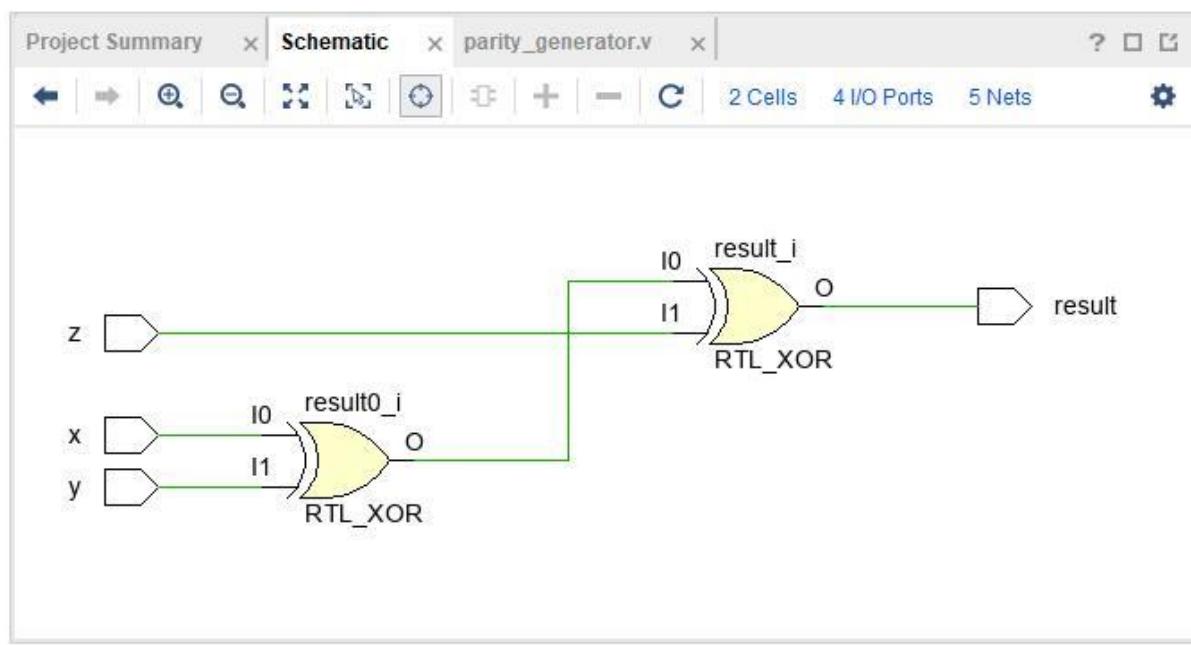
Test Bench:



```
module parity_tb;  
reg x,y,z;  
wire result;  
  
initial begin  
x = 0;  
y = 0;  
z = 0;  
  
#100;  
x = 0;  
y = 0;  
z = 1;  
  
#100;  
x = 0;  
y = 1;  
z = 0;  
  
#100;  
x = 0;  
y = 1;  
z = 1;  
  
#100;  
x = 1;  
y = 0;  
z = 0;  
  
#100;  
x = 1;  
y = 0;  
z = 1;
```

```
#100;  
x = 1;  
y = 0;  
z = 1;  
  
#100;  
x = 1;  
y = 1;  
z = 0;  
  
#100;  
x = 1;  
y = 1;  
z = 1;  
  
#100;  
end  
endmodule
```

RTL Schematic:



Synthesis Report:

```
Start Writing Synthesis Report
```

```
Report BlackBoxes:
```

BlackBox name	Instances

```
Report Cell Usage:
```

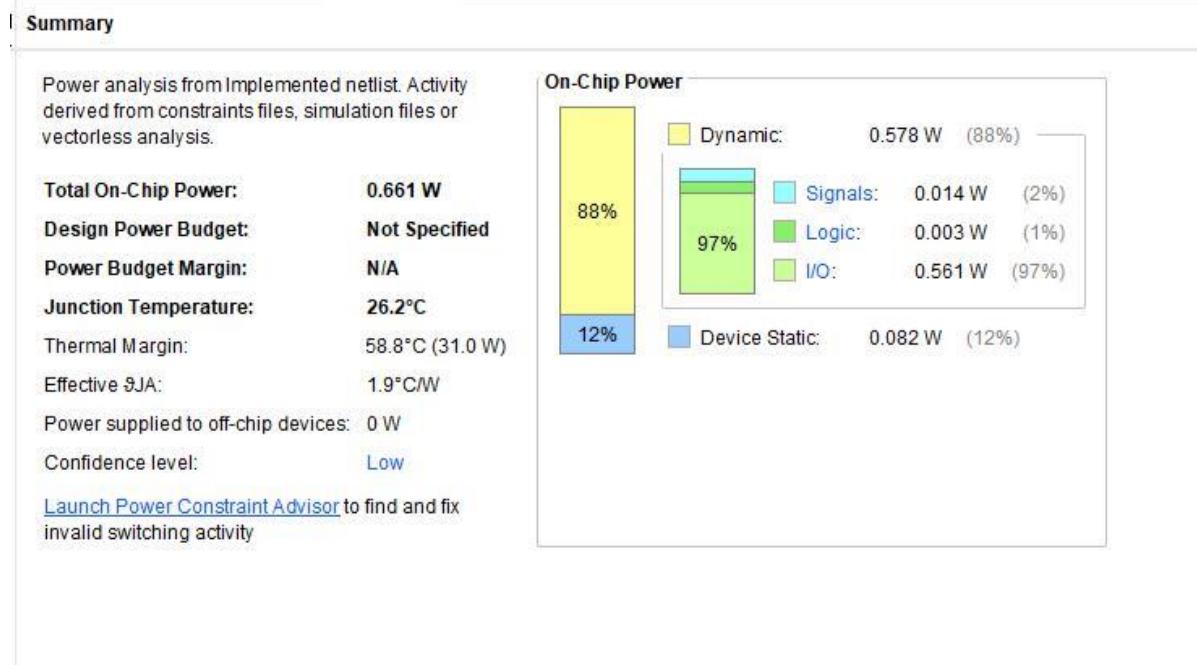
Cell	Count
LUT3	1
IBUF	3
OBUF	1

```
Report Instance Areas:
```

Instance	Module	Cells
top		5

```
Finished Writing Synthesis Report : Time (s): cpu = 00:00:16 ; elapsed = 00:13:45 . Memory (MB): peak = 1014.574 ; gain = 0.000
```

Power Report:



Code 6: Binary to One Hot Code Encoder

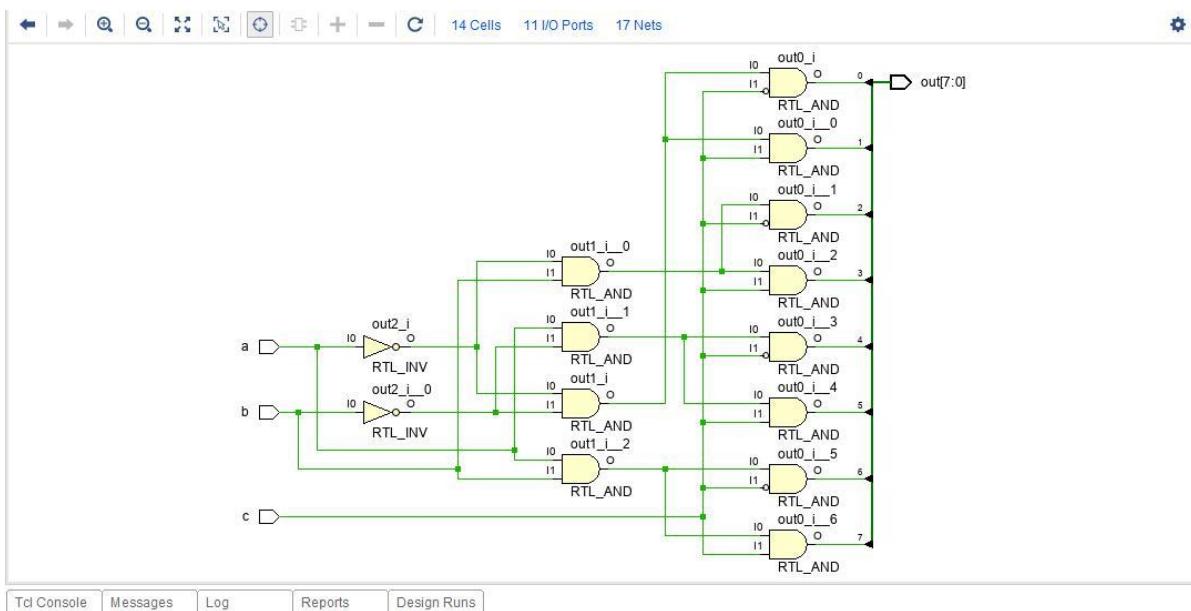
Verilog Code:

```
module decoder_3_8(a, b, c, out);
    input a,b,c;
    output [7:0] out ;
    assign out [0] = (~a&~b&~c) ;
    assign out [1] =(~a&~b&c) ;
    assign out [2] =(~a&b&~c);
    assign out [3] = (~a&b&c);
    assign out [4] = (a&~b&~c);
    assign out [5] = (a&~b&c);
    assign out [6] = (a&b&~c);
    assign out [7] = (a&b&c);
endmodule
```

Test Bench:

```
module test_decoder;
reg a, b,c;
wire [7:0] out;
decoder_3_8 DUT(a,b,c,out);
initial
begin
$monitor($time,"a=%b , b=%b , c=%b , out = %b" , a,b,c,out);
a=0 ; b=0 ;c=0 ;
# 100
a=0 ; b=0 ;c=1 ;
#100
a=0 ; b=1 ;c=0 ;
#100
a=1 ; b=1 ;c=1 ;
#100 $finish;
end
endmodule
```

RTL Schematic:



Synthesis Report:

```
Start Writing Synthesis Report
-----
Report BlackBoxes:
+-----+-----+
| |BlackBox name |Instances |
+-----+-----+
+-----+-----+
Report Cell Usage:
+-----+-----+
| |Cell |Count |
+-----+-----+
|1 |LUT3 |    8|
|2 |IBUF |    3|
|3 |OBUF |    8|
+-----+-----+
Report Instance Areas:
+-----+-----+-----+
| |Instance |Module |Cells |
+-----+-----+-----+
|1 |top      |      |   19|
+-----+-----+-----+
-----
Finished Writing Synthesis Report : Time (s): cpu = 00:00:17 ; elapsed = 00:00:30 . Memory (MB): peak = 1019.449 ; gain = 0.000
```

Power Report:

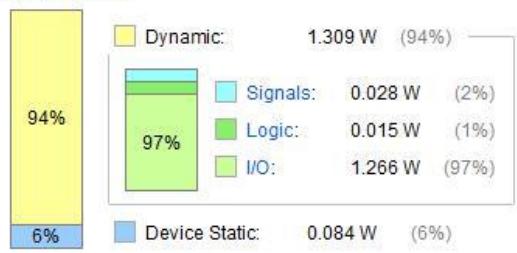
Summary

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

Total On-Chip Power:	1.393 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	27.6°C
Thermal Margin:	57.4°C (30.3 W)
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

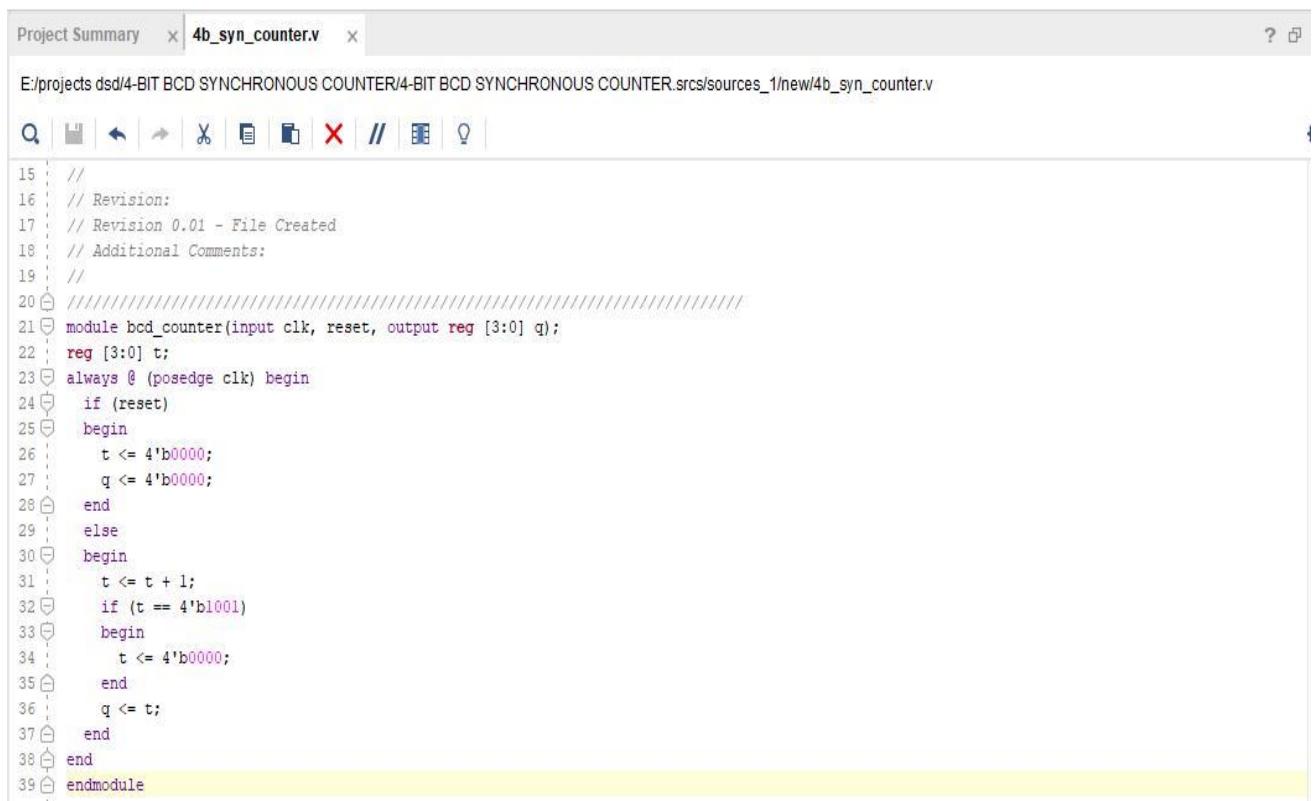
[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



Code 7: 4 Bit BCD Synchronous Counter

Verilog Code:

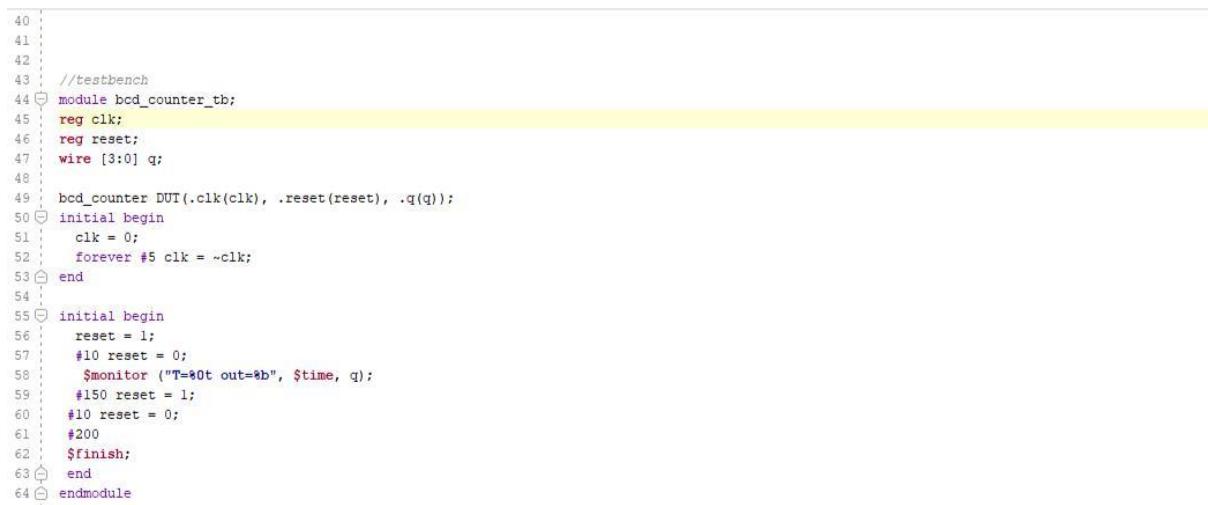


The screenshot shows a Verilog code editor window with the following details:

- Project Summary: E:/projects/dsd/4-BIT BCD SYNCHRONOUS COUNTER/4-BIT BCD SYNCHRONOUS COUNTER.srscs/sources_1/new/4b_syn_counter.v
- File: 4b_syn_counter.v
- Code Content:

```
15 //  
16 // Revision:  
17 // Revision 0.01 - File Created  
18 // Additional Comments:  
19 //  
20 module bcd_counter(input clk, reset, output reg [3:0] q);  
21 reg [3:0] t;  
22 always @ (posedge clk) begin  
23 if (reset)  
24 begin  
25 t <= 4'b0000;  
26 q <= 4'b0000;  
27 end  
28 else  
29 begin  
30 t <= t + 1;  
31 if (t == 4'b1001)  
32 begin  
33 t <= 4'b0000;  
34 end  
35 q <= t;  
36 end  
37 end  
38 endmodule
```

Test Bench:

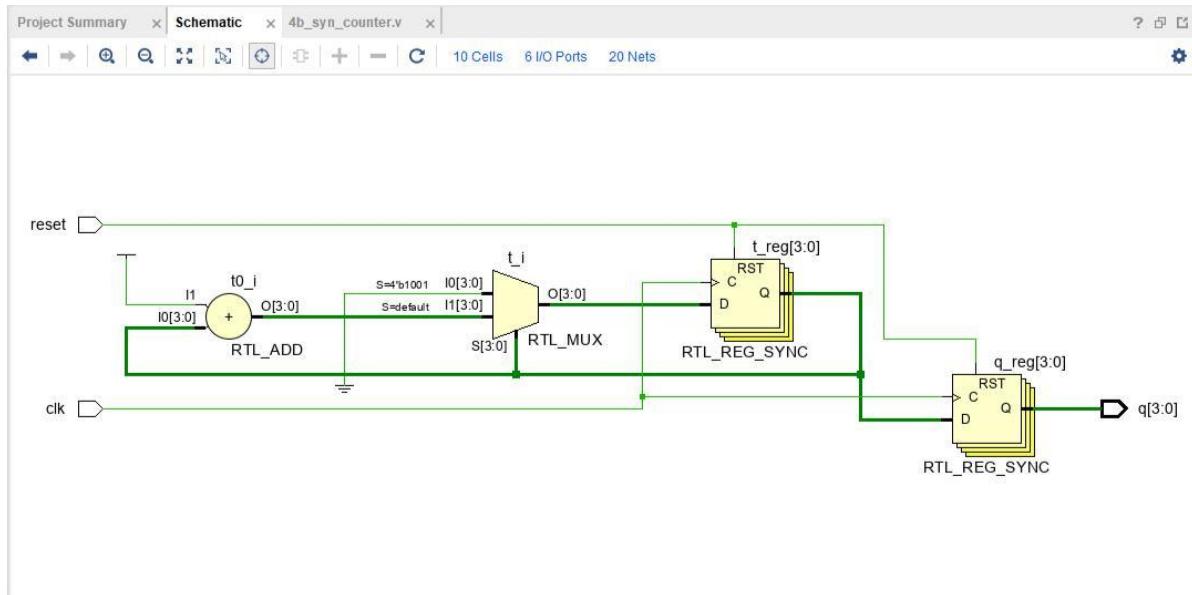


The screenshot shows a Verilog code editor window with the following details:

- File: bcd_counter_tb.v
- Code Content:

```
40  
41  
42  
43 //testbench  
44 module bcd_counter_tb;  
45 reg clk;  
46 reg reset;  
47 wire [3:0] q;  
48  
49 bcd_counter DUT(.clk(clk), .reset(reset), .q(q));  
50 initial begin  
51 clk = 0;  
52 forever #5 clk = ~clk;  
53 end  
54  
55 initial begin  
56 reset = 1;  
57 #10 reset = 0;  
58 $monitor ("T=%0t out=%b", $time, q);  
59 #150 reset = 1;  
60 #10 reset = 0;  
61 #200  
$finish;  
62 end  
63 endmodule
```

RTL Schematic:



Synthesis Report:

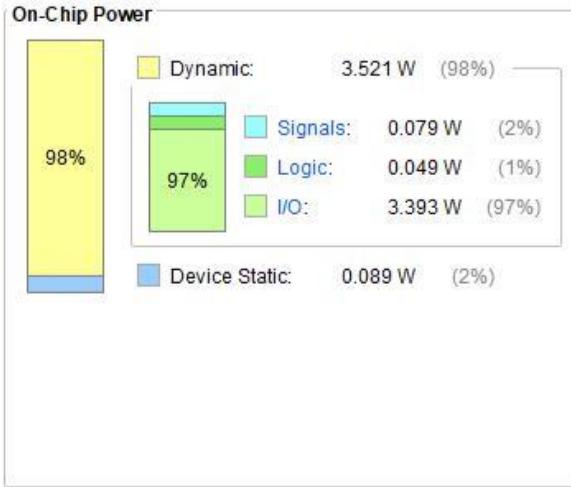
```
Start Writing Synthesis Report
-----
Report BlackBoxes:
+++
| |BlackBox name |Instances |
+++
Report Cell Usage:
+---+---+---+
|     |Cell |Count |
+---+---+---+
|1   |BUFGE |    1|
|2   |LUT1  |    1|
|3   |LUT3  |    1|
|4   |LUT4  |    2|
|5   |FDRE  |    8|
|6   |IBUF  |    2|
|7   |OBUF  |    4|
+---+---+---+
Report Instance Areas:
+---+---+---+---+
|     |Instance |Module |Cells |
+---+---+---+---+
|1   |top      |       | 19|
+---+---+---+---+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:14 ; elapsed = 00:00:26 . Memory (MB): peak = 1018.500 ; gain = 0.000
```

Power Report:

Summary

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

Total On-Chip Power: 3.609 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 31.8°C
Thermal Margin: 53.2°C (28.1 W)
Effective θ_{JA}: 1.9°C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



Code 8: 4 Bit Carry Look-Ahead Adder

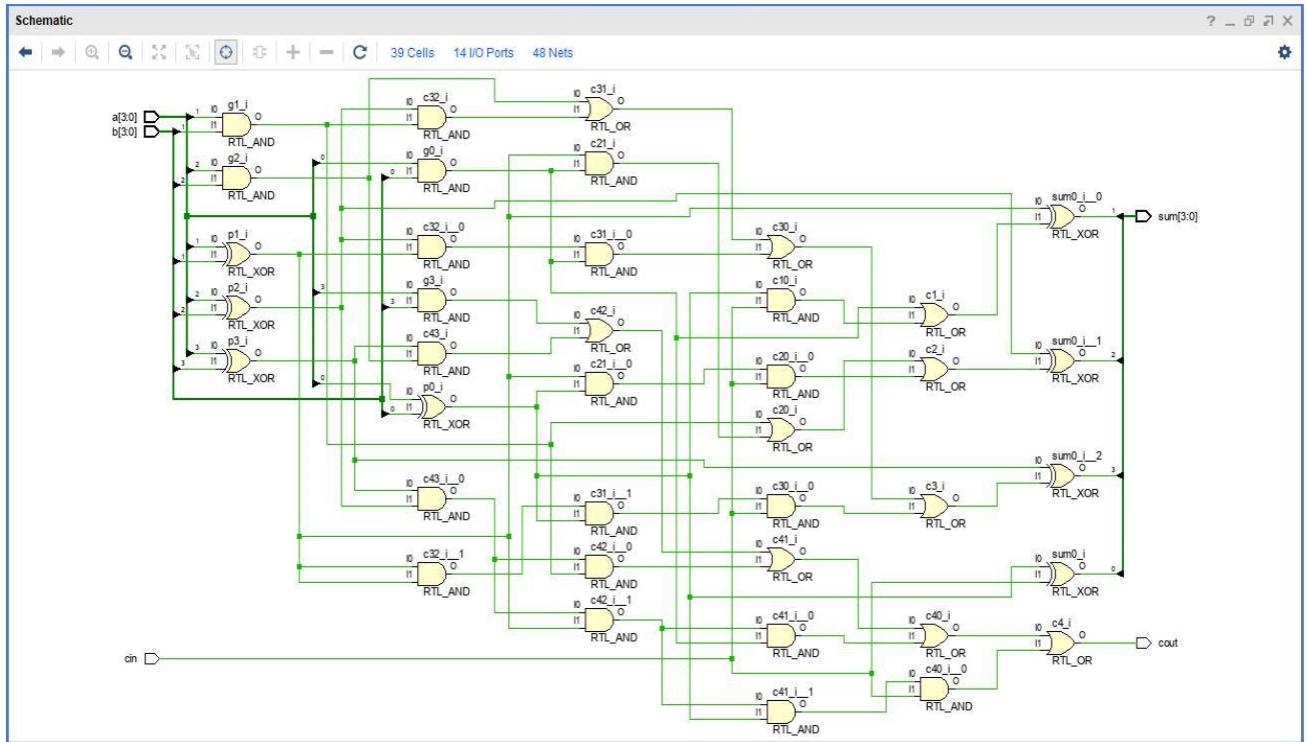
Verilog Code:

```
Q |  |  |  |  |  |  |  | // |  | ? |  
21 module CLA_Adder(a,b,cin,sum,cout);  
22 input [3:0] a,b;  
23 input cin;  
24 output [3:0] sum;  
25 output cout;  
26 wire p0,p1,p2,p3,g0,g1,g2,g3,c1,c2,c3,c4;  
27 assign p0=(a[0]^b[0]),  
28 p1=(a[1]^b[1]),  
29 p2=(a[2]^b[2]),  
30 p3=(a[3]^b[3]);  
31 assign g0=(a[0]&b[0]),  
32 g1=(a[1]&b[1]),  
33 g2=(a[2]&b[2]),  
34 g3=(a[3]&b[3]);  
35 assign c0=cin,  
36 c1=g0|(p0&cin),  
37 c2=g1|(p1&p0&cin),  
38 c3=g2|(p2&g1)|(p2&p1&g0)|(p1&p0&g0),  
39 c4=g3|(p3&g2)|(p3&p2&g1)|(p3&p2&p1&g0)|(p3&p2&p1&p0&cin);  
40 assign sum[0]=p0^c0,  
41 sum[1]=p1^c1,  
42 sum[2]=p2^c2,  
43 sum[3]=p3^c3;  
44 assign cout=c4;  
45 endmodule
```

Test Bench:

```
module TestModule;  
  
reg [3:0] a;  
reg [3:0] b;  
reg cin;  
  
wire [3:0] sum;  
wire cout;  
  
CLA_Adder uut (  
.a(a),  
.b(b),  
.cin(cin),  
.sum(sum),  
.cout(cout)  
);  
initial begin  
a = 0;  
b = 0;  
cin = 0;  
  
#100;  
a = 5;  
b = 6;  
cin = 1;  
#100;  
end  
endmodule
```

RTL Schematic:



Synthesis Report:

Start Writing Synthesis Report

Report BlackBoxes:	
	BlackBox name Instances
+	+
+	+

Report Cell Usage:

+-----+-----+	
Cell	Count
-----+-----+	
1	LUT2 1
2	LUT3 1
3	LUT4 1
4	LUT5 4
5	LUT6 2
6	IBUF 9
7	OBUF 5
-----+-----+	

Report Instance Areas:

+-----+-----+-----+			
	Instance	Module	Cells
-----+-----+-----+			
1	top		23
-----+-----+-----+			

Finished Writing Synthesis Report : Time (s): cpu = 00:00:18 ; elapsed = 00:00:32 . Memory (MB): peak = 1015.535 ; gain = 0.000

Power Report:

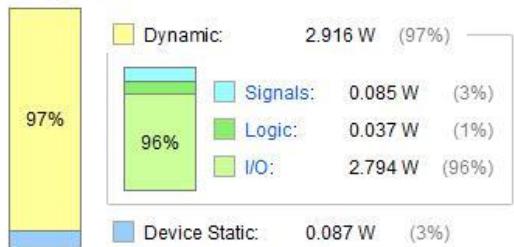
Summary

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

Total On-Chip Power:	3.003 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	30.7°C
Thermal Margin:	54.3°C (28.7 W)
Effective 9JA:	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



Code 9: n-Bit Comparator

Verilog Code:

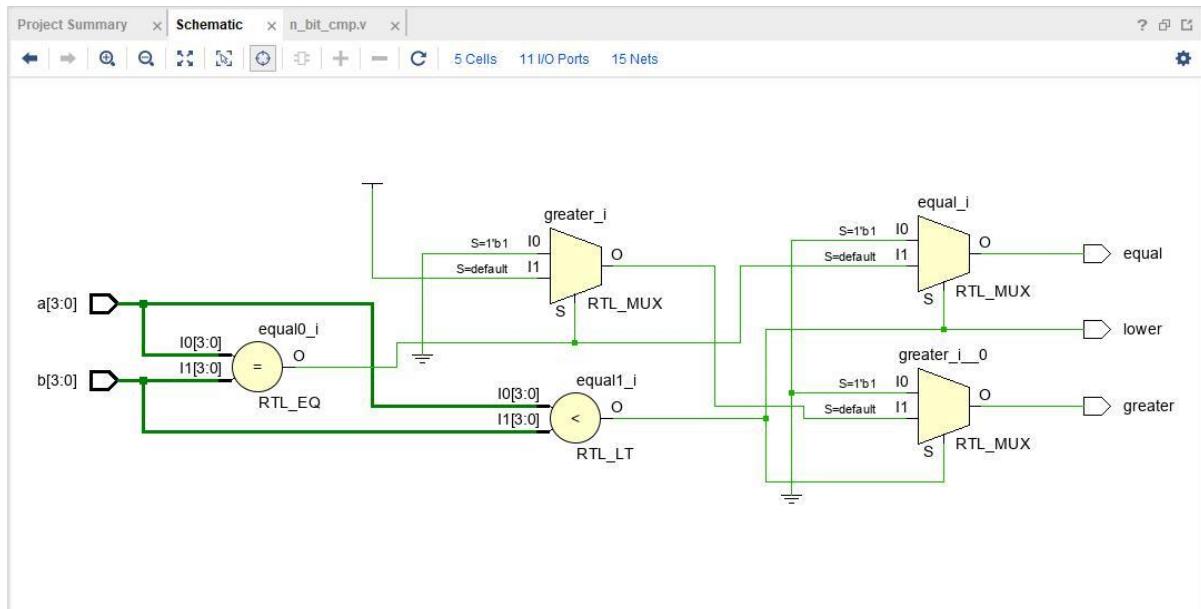
```
3 module comparator (
    input wire [3:0] a,
    input wire [3:0] b,
    output reg equal,
    output reg lower,
    output reg greater
);

    always @* begin
        if (a>b) begin
            equal = 0;
            lower = 1;
            greater = 0;
        end
        else if (a==b) begin
            equal = 1;
            lower = 0;
            greater = 0;
        end
        else begin
            equal = 0;
            lower = 0;
            greater = 1;
        end
    end
endmodule
```

Test Bench:

```
module testcomp;
    reg [3:0] a, b;
    wire eq, lw, gr;
    comparator uut (
        .a(a),
        .b(b),
        .equal(eq),
        .lower(lw),
        .greater(gr)
    );
    initial begin
        a = 0;
        repeat (16) begin
            b = 0;
            repeat (16) begin
                #10;
                $display ("TESTING #d and #d yields eq=%d lw=%d gr=%d", a, b, eq, lw, gr);
                if (a==b && eq!=1'b1 && gr!=1'b0 && lw!=1'b0) begin
                    $display ("ERROR!");
                    $finish;
                end
                if (a>b && eq!=1'b0 && gr!=1'b1 && lw!=1'b0) begin
                    $display ("ERROR!");
                    $finish;
                end
                if (a<b && eq!=1'b1 && gr!=1'b0 && lw!=1'b1) begin
                    $display ("ERROR!");
                    $finish;
                end
                b = b + 1;
            end
            a = a + 1;
        end
        $display ("PASSED!");
        $finish;
    end
endmodule
```

RTL Schematic:



Synthesis Report:

```
Start Writing Synthesis Report
-----
Report BlackBoxes:
++-----+-----+
| |BlackBox name |Instances |
++-----+-----+
++-----+-----+
Report Cell Usage:
+-----+-----+
|     |Cell |Count |
+-----+-----+
|1    |LUT3  |    1|
|2    |LUT4  |    2|
|3    |LUT6  |    2|
|4    |IBUF  |    8|
|5    |OBUF  |    3|
+-----+-----+
Report Instance Areas:
+-----+-----+-----+
|     |Instance |Module |Cells |
+-----+-----+-----+
|1    |ltop    |      |   16|
+-----+-----+-----+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:17 ; elapsed = 00:00:29 . Memory (MB): peak = 1015.203 ; gain = 0.000
```

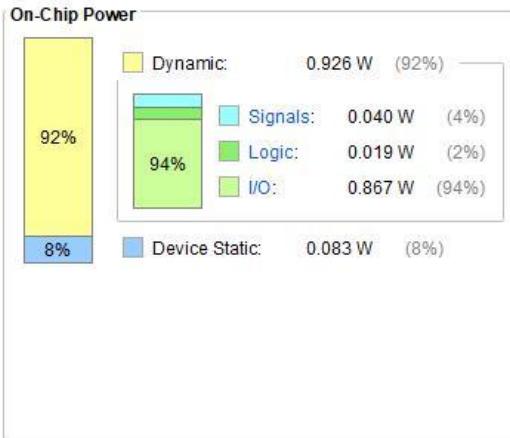
Power Report:

Summary

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

Total On-Chip Power:	1.009 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	26.9°C
Thermal Margin:	58.1°C (30.7 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



Code 10: Serial In Serial Out Shift Register

Verilog Code:

```
module siso_design(input clk,b,output q);
wire w1,w2,w3;

d_ff dut1(.clk(clk),.d(b),.q(w1),.rst());
d_ff dut2(.clk(clk),.d(w1),.q(w2),.rst());
d_ff dut3(.clk(clk),.d(w2),.q(w3),.rst());
d_ff dut4(.clk(clk),.d(w3),.q(q),.rst());

endmodule

// d flip flop
module d_ff (
  input clk,
  input d,
  input rst,
  output reg q);

  always @ (posedge clk)
  begin
    if (rst)
      q <= 1'b0;
    else
      q <= d;
  end

endmodule
```

Test Bench:

```
// testbench
`module siso_tb();

reg clk,b;
wire q;

siso_design uut(.clk(clk),.b(b),.q(q));

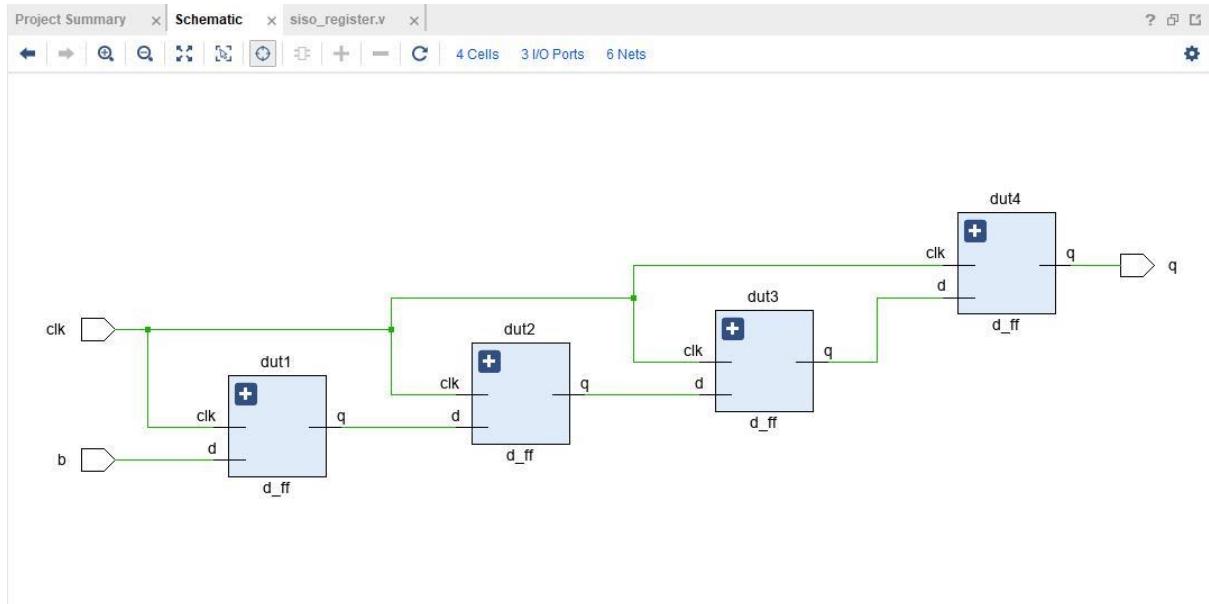
`initial
`begin
  clk=1'b0;
  forever #5clk=~clk;
`end

`initial
`begin
  $monitor("clk=%d,b=%d,q=%d",clk,b,q);
`end

`initial
`begin
  b=1;
  #10;
  b=1;
  #10;
  b=1;
  #10;
  b=0;

  #50;
  $finish;
`end
`endmodule
```

RTL Schematic:



Synthesis Report:

```
Start Writing Synthesis Report
-----
Report BlackBoxes:
+-----+-----+
| |BlackBox name |Instances |
+-----+-----+
+-----+-----+
Report Cell Usage:
+-----+-----+-----+
| |Cell |Count |
+-----+-----+-----+
|1 |BUFG | 1|
|2 |SRL16E | 1|
|3 |FDRE | 2|
|4 |IBUF | 2|
|5 |OBUF | 1|
+-----+-----+-----+
Report Instance Areas:
+-----+-----+-----+
| |Instance |Module |Cells |
+-----+-----+-----+
|1 |top | | 7|
|2 | dut1 | d_ff | 1|
|3 | dut3 | d_ff_0 | 1|
|4 | dut4 | d_ff_1 | 1|
+-----+-----+-----+
-----
Finished Writing Synthesis Report : Time (s): cpu = 00:00:17 ; elapsed = 00:00:31 . Memory (MB): peak = 1018.820 ; gain = 0.000
```

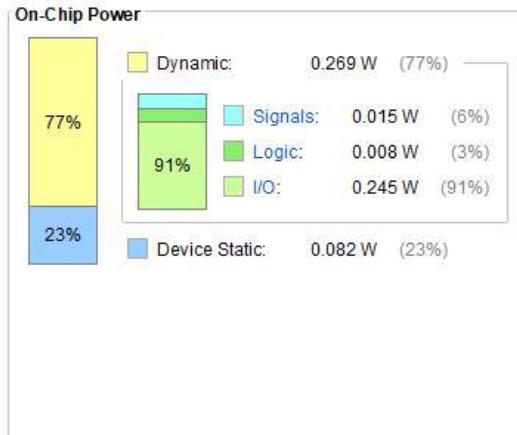
Power Report:

Summary

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

Total On-Chip Power:	0.351 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 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



Code 11" Serial In Parallel Out Shift Register

Verilog Code:

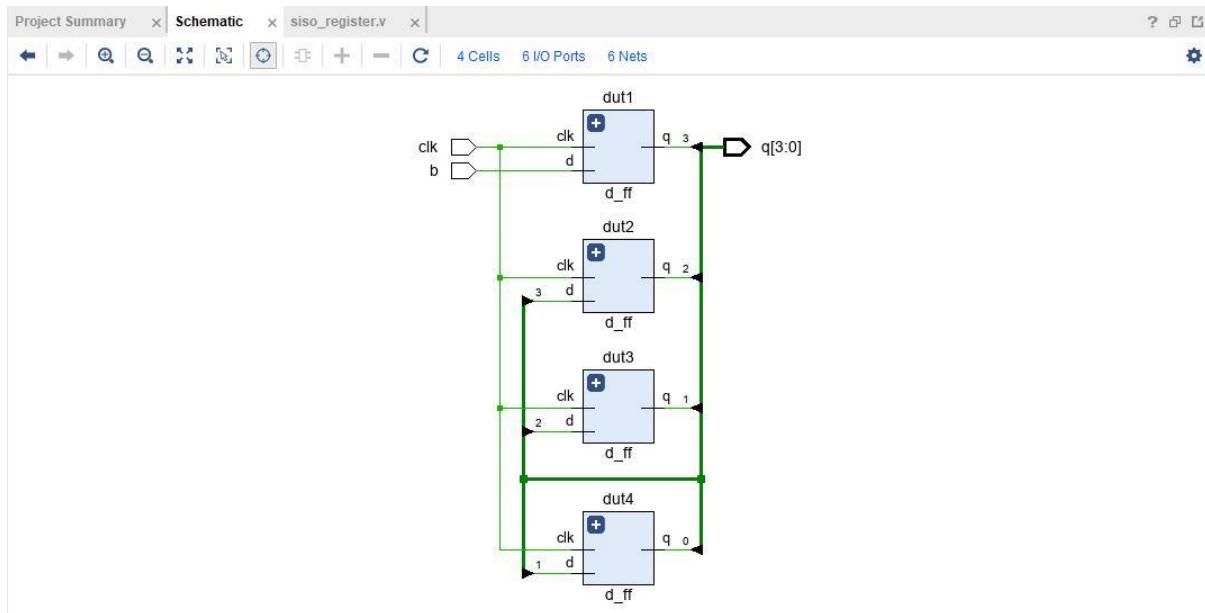
```
E:/projects/dsd/SIPO REGISTER/SIPO REGISTER.srsc/sources_1/new/siso_register.v

1 module sipo_shift_register_design(input clk,b,output[3:0]q);
2
3   d_ff dut1(.clk(clk),.d(b),.q(q[3]),.rst());
4   d_ff dut2(.clk(clk),.d(q[3]),.q(q[2]),.rst());
5   d_ff dut3(.clk(clk),.d(q[2]),.q(q[1]),.rst());
6   d_ff dut4(.clk(clk),.d(q[1]),.q(q[0]),.rst());
7
8 endmodule
// d flip flop
10
11 module d_ff (
12   input clk,
13   input d,
14   input rst,
15   output reg q);
16
17 always @(posedge clk)
18 begin
19   if (rst)
20     q <= 1'b0;
21   else
22     q <= d;
23 end
24
25 endmodule
`z
```

Test Bench:

```
2 // testbench
3 module sipo_tb();
4
5 reg clk,b;
6 wire [3:0]q;
7
8 sipo_shift_register_design uut(.clk(clk),.b(b),.q(q));
9
10 initial
11 begin
12   clk=1'b0;
13   forever #5clk=~clk;
14 end
15
16 initial
17 begin
18   $monitor("clk=%d,b=%d,q=%d",clk,b,q);
19 end
20
21 initial
22 begin
23   b=1;
24   #10;
25   b=0;
26   #10;
27   b=1;
28   #10;
29   b=0;
30
31   #50;
32   $finish;
33
34 end
35
36 endmodule
```

RTL Schematic:



Synthesis Report:

```
Start Writing Synthesis Report
-----
Report BlackBoxes:
+-----+-----+
| |BlackBox name |Instances |
+-----+-----+
+-----+-----+
Report Cell Usage:
+-----+-----+
| |Cell |Count |
+-----+-----+
|1 |BUFGE | 1|
|2 |FDRE | 4|
|3 |IBUF | 2|
|4 |OBUF | 4|
+-----+-----+
Report Instance Areas:
+-----+-----+-----+
| |Instance |Module |Cells |
+-----+-----+-----+
|1 |top | | 11|
|2 | dut1 |d_ff | 1|
|3 | dut2 |d_ff_0 | 1|
|4 | dut3 |d_ff_1 | 1|
|5 | dut4 |d_ff_2 | 1|
+-----+-----+-----+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:18 ; elapsed = 00:00:32 . Memory (MB): peak = 1041.301 ; gain = 0.000
```

Power Report:

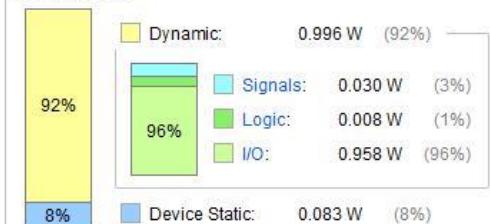
Summary

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

Total On-Chip Power:	1.079 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	27.0°C
Thermal Margin:	58.0°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



Code 12: Parallel In Parallel Out Shift Register

Verilog Code:

```
timescale 1ns / 1ps
module pipo_design(input clk,input [3:0]b,output[3:0]a;
d_ff d1(.clk(clk),.d(b[3]),.q(a[3]),.rst());
d_ff d2(.clk(clk),.d(b[2]),.q(a[2]),.rst());
d_ff d3(.clk(clk),.d(b[1]),.q(a[1]),.rst());
d_ff d4(.clk(clk),.d(b[0]),.q(a[0]),.rst());
endmodule

// d flip flop

module d_ff (
    input clk,      // clock input
    input d,        // data input
    input rst,      // asynchronous reset input
    output reg q   // output
);
    always @ (posedge clk) begin
        if (rst) // asynchronous reset
            q <= 1'b0;
        else // normal operation
            q <= d;
    end
endmodule
```

Test Bench:

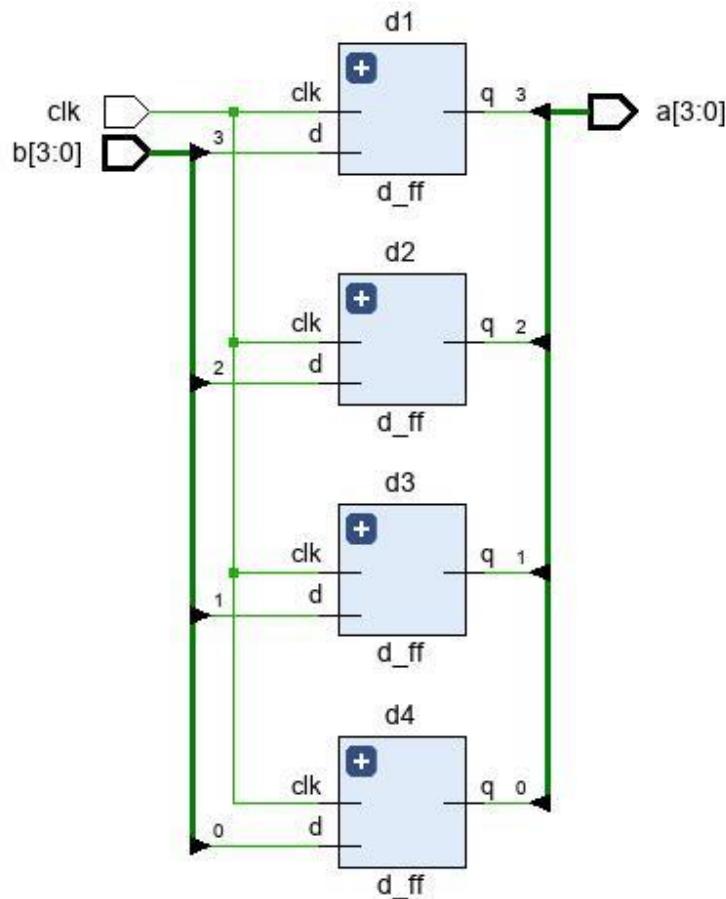
```
// test bench
module pipo_tb();
reg clk;
reg [3:0]b;
wire [3:0]a;

pipo_design uut(.clk(clk),.b(b),.a(a));

initial
begin
clk=0;
forever #10clk=~clk;
end

initial
begin
#10;
b=4'b1000;
#10;
b=4'b0101;
#10;
$display("clk=%d,b=%d,a=%d",clk,b,a);
#100 $finish;
end
endmodule
```

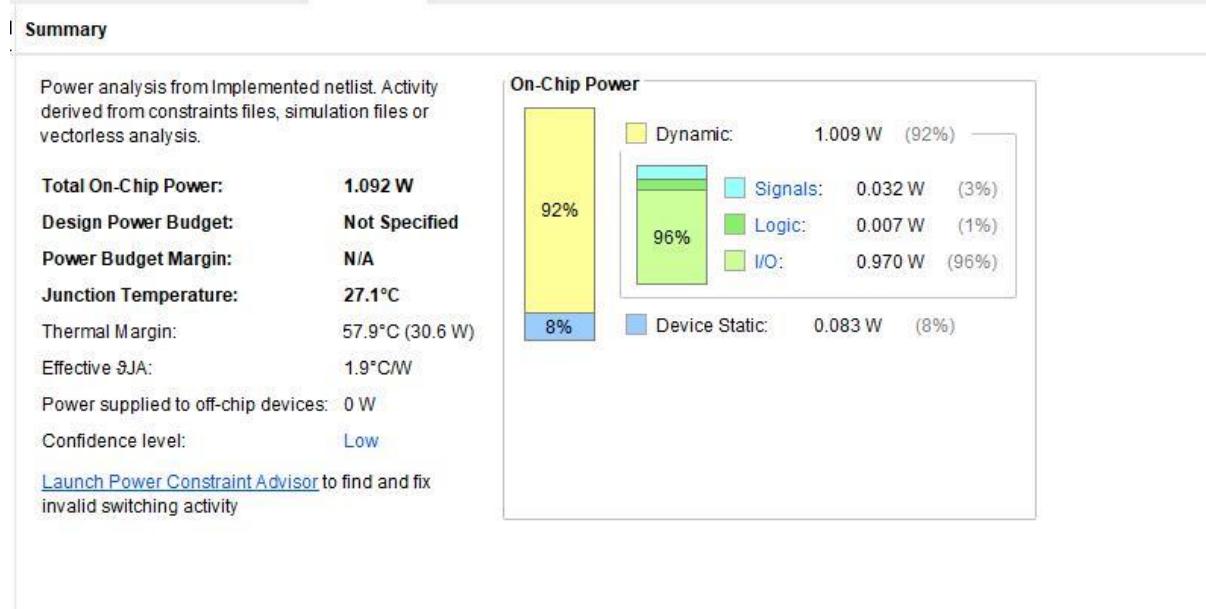
RTL Schematic:



Synthesis Report:

```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
+---+---+  
| |BlackBox name |Instances |  
+---+---+  
+---+---+  
  
Report Cell Usage:  
+---+---+  
| |Cell |Count |  
+---+---+  
|1 |BUFG | 1|  
|2 |FDRE | 4|  
|3 |IBUF | 5|  
|4 |OBUF | 4|  
+---+---+  
  
Report Instance Areas:  
+---+---+---+  
| |Instance |Module |Cells |  
+---+---+---+  
|1 |top | | 14|  
|2 | d1 |d_ff | 1|  
|3 | d2 |d_ff_0 | 1|  
|4 | d3 |d_ff_1 | 1|  
|5 | d4 |d_ff_2 | 1|  
+---+---+---+  
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:09 ; elapsed = 00:00:17 . Memory (MB): peak = 1017.965 ; gain = 0.000
```

Power Report:



Code 13: Parallel In Serial Out Shift Register

Verilog Code:

```
1 module Shiftregister_PISO(Clk, Parallel_In,load, Serial_Out);
2   input Clk,load;
3   input [3:0]Parallel_In;
4   output reg Serial_Out;
5   reg [3:0]tmp;
6   always @(posedge Clk)
7     begin
8       if(load)
9         tmp<=Parallel_In;
10      else
11        begin
12          Serial_Out<=tmp[3];
13          tmp<={tmp[2:0],1'b0};
14        end
15      end
16   endmodule
```

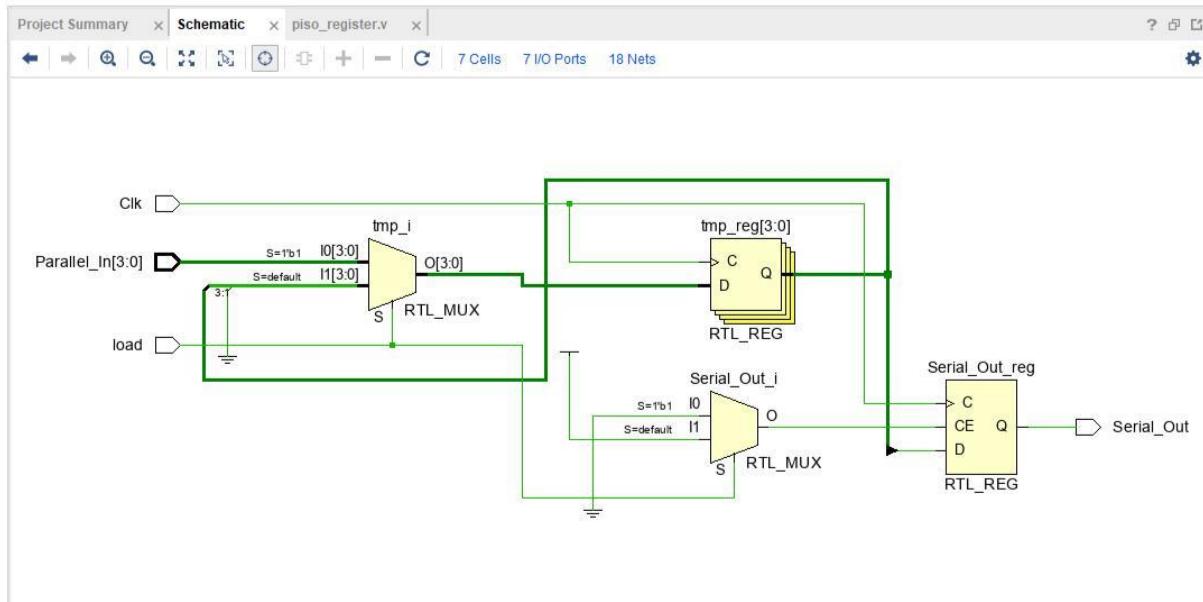
Test Bench:

```

module Shiftregister_PISO_tb();
reg [3:0]Parallel_in;
reg Clk,load;
wire Serial_out;
piso_design dut(Clk,load,Parallel_in,Serial_out);
initial begin
Clk=1'b0;
forever #5 Clk=~Clk;
end
initial begin
load=0;b=4'b0101;
#20 load=1;
#20 load=1;
#10 load=0;
#10 load=0;
#100 $finish;
end
endmodule

```

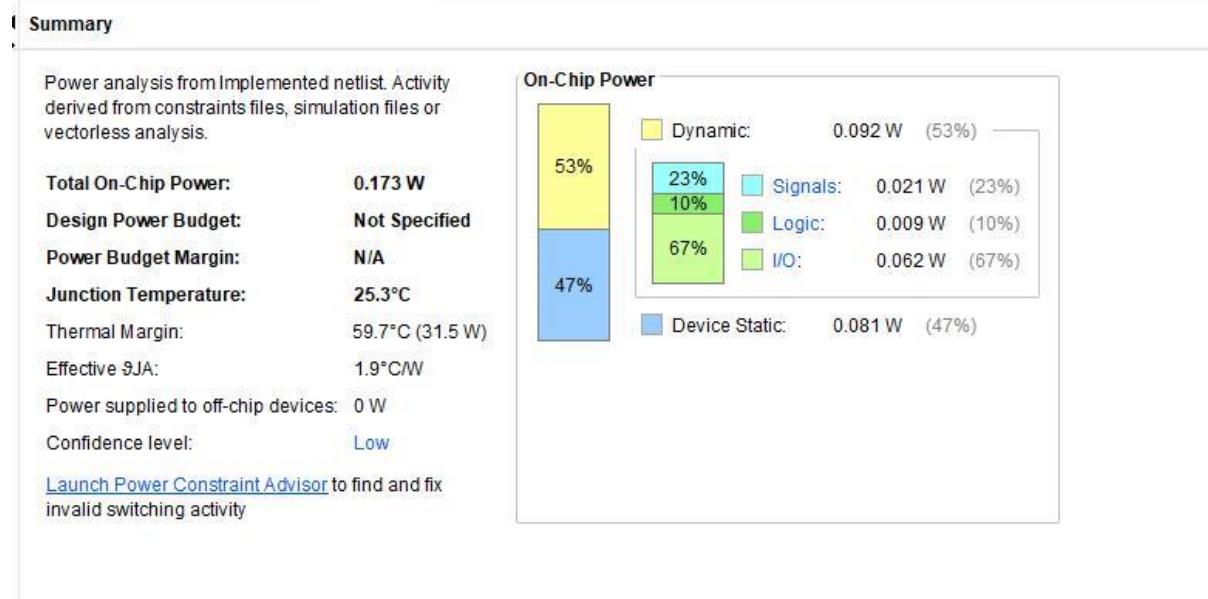
RTL Schematic:



Synthesis Report:

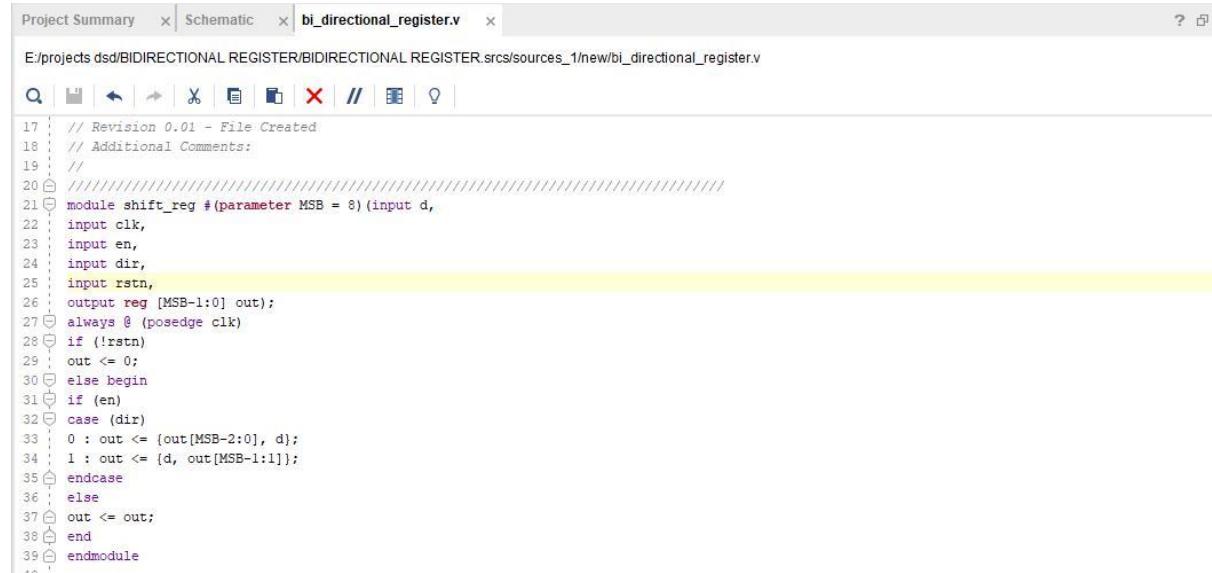
```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
+---+---+  
| |BlackBox name |Instances |  
+---+---+  
+---+---+  
  
Report Cell Usage:  
+---+---+---+  
| |Cell |Count |  
+---+---+---+  
|1 |IBUFG | 1|  
|2 |LUT1 | 1|  
|3 |LUT2 | 1|  
|4 |LUT3 | 3|  
|5 |FDRE | 5|  
|6 |IBUF | 6|  
|7 |OBUF | 1|  
+---+---+---+  
  
Report Instance Areas:  
+---+---+---+  
| |Instance |Module |Cells |  
+---+---+---+  
|1 |top | | 18|  
+---+---+---+  
  
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:15 ; elapsed = 00:00:51 . Memory (MB): peak = 1015.211 ; gain = 0.000
```

Power Report:



Code 14: BiDirectional Shift Register

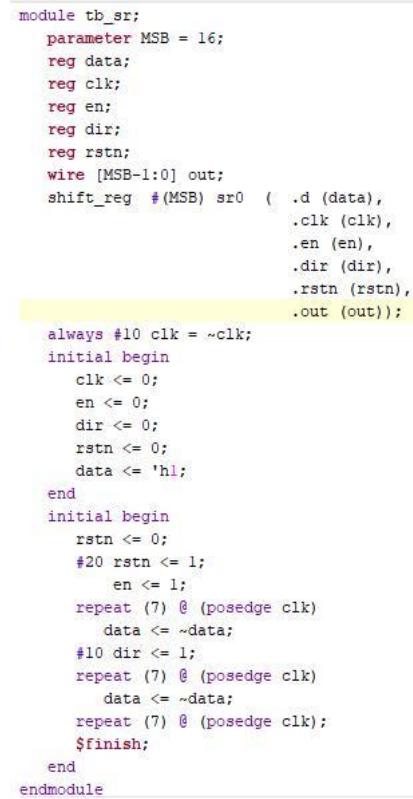
Verilog Code:



The screenshot shows a Verilog code editor window. The title bar says "Project Summary" and "Schematic" with "bi_directional_register.v" selected. The main area displays the Verilog code for a bidirectional shift register. The code includes comments, a module definition, input and output ports, an always block with a case statement for direction, and an endmodule keyword.

```
17: // Revision 0.01 - File Created
18: // Additional Comments:
19:
20: module shift_reg #(parameter MSB = 8) (input d,
21:   input clk,
22:   input en,
23:   input dir,
24:   input rstn,
25:   output reg [MSB-1:0] out);
26:   always @ (posedge clk)
27:     if (!rstn)
28:       if (en)
29:         case (dir)
30:           0 : out <= {out[MSB-2:0], d};
31:           1 : out <= {d, out[MSB-1:1]};
32:         endcase
33:       else
34:         out <= out;
35:     end
36:   end
37: endmodule
```

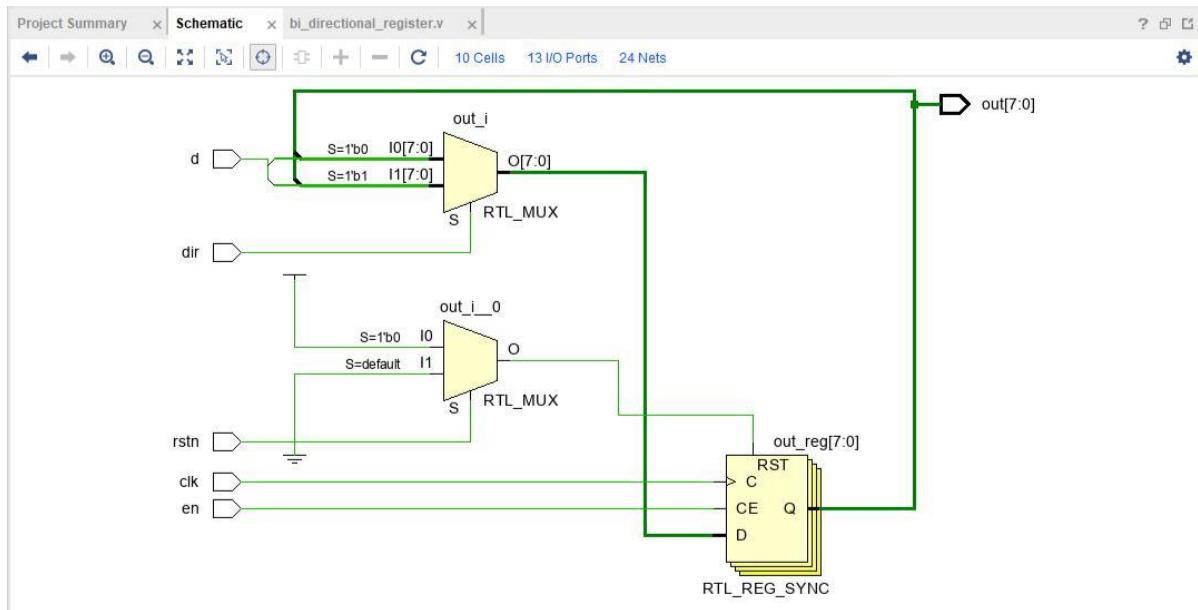
Test Bench:



The screenshot shows a Verilog test bench code editor window. The title bar says "tb_sr". The code defines a module tb_sr with parameters MSB=16, and various input and output ports. It includes an initial block for setting up initial values and a begin block for a test sequence involving a shift register and a stimulus. The code ends with a \$finish command.

```
module tb_sr;
  parameter MSB = 16;
  reg data;
  reg clk;
  reg en;
  reg dir;
  reg rstn;
  wire [MSB-1:0] out;
  shift_reg #(MSB) sr0 (.d (data),
                        .clk (clk),
                        .en (en),
                        .dir (dir),
                        .rstn (rstn),
                        .out (out));
  always #10 clk = ~clk;
  initial begin
    clk <= 0;
    en <= 0;
    dir <= 0;
    rstn <= 0;
    data <= 'h1;
  end
  initial begin
    rstn <= 0;
    #20 rstn <= 1;
    en <= 1;
    repeat (7) @ (posedge clk)
      data <= ~data;
    #10 dir <= 1;
    repeat (7) @ (posedge clk)
      data <= ~data;
    repeat (7) @ (posedge clk);
    $finish;
  end
endmodule
```

RTL Schematic:



Synthesis Report:

```
Start Writing Synthesis Report
-----
Report BlackBoxes:
++-----+-----+
| |BlackBox name |Instances |
++-----+-----+
++-----+-----+
++-----+-----+
Report Cell Usage:
-----+-----+
| |Cell |Count |
-----+-----+
|1 |BUFGE | 1|
|2 |LUT1 | 1|
|3 |LUT3 | 8|
|4 |FDRE | 8|
|5 |IBUF | 5|
|6 |OBUF | 8|
-----+-----+
Report Instance Areas:
-----+-----+
| |Instance |Module |Cells |
-----+-----+
|1 |ltop |  | 31|
-----+-----+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:09 ; elapsed = 00:00:17 . Memory (MB): peak = 1014.953 ; gain = 0.000
```

Power Report:

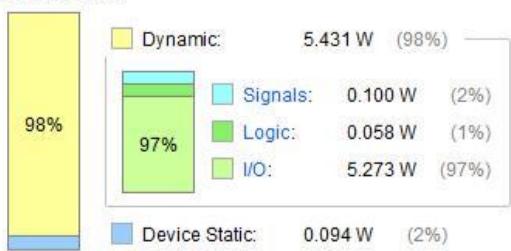
Summary

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

Total On-Chip Power: 5.524 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 35.4°C
Thermal Margin: 49.6°C (26.1 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



Code 15: PRBS Sequence Generator

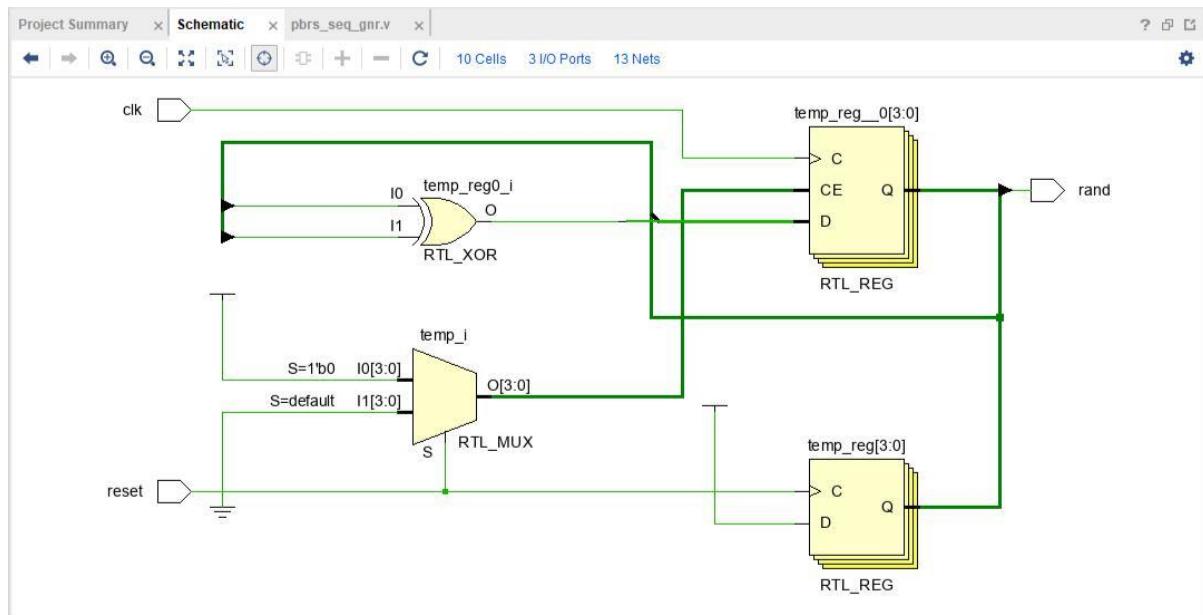
Verilog Code:

```
module prbs (rand, clk, reset);
    input clk, reset;
    output rand;
    wire rand;
    reg [3:0] temp;
    always @ (posedge reset) begin
        temp <= 4'hf;
    end
    always @ (posedge clk) begin
        if (~reset) begin
            temp <= {temp[0]^temp[1],temp[3],temp[2],temp[1]};
        end
    end
    assign rand = temp[0];
endmodule
```

Test Bench:

```
module pbrs_tb;
    reg clk, reset;
    wire rand;
    prbs pr (rand, clk, reset);
    initial begin
        forever begin
            clk <= 0;
            #5
            clk <= 1;
            #5
            clk <= 0;
        end
    end
    initial begin
        reset = 1;
        #12
        reset = 0;
        #90
        reset = 1;
        #12
        reset = 0;
    end
endmodule
```

RTL Schematic:



Synthesis Report:

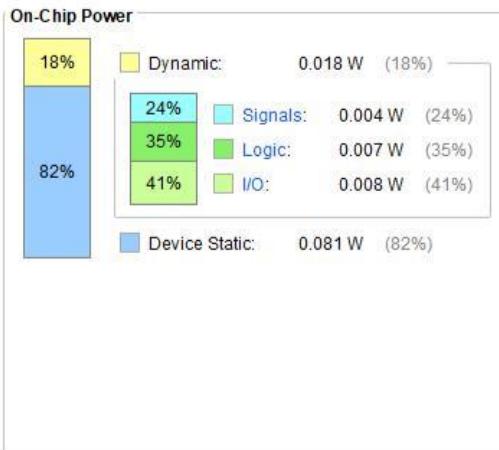
```
-----
Start Writing Synthesis Report
-----
Report BlackBoxes:
++-----+-----+
| |BlackBox name |Instances |
++-----+-----+
++-----+-----+
Report Cell Usage:
++-----+-----+
| |Cell |Count |
++-----+-----+
|1 |IBUFG | 2|
|2 |LUT1 | 2|
|3 |LUT2 | 1|
|4 |FDRE | 8|
|5 |IBUF | 2|
|6 |OBUF | 1|
++-----+-----+
Report Instance Areas:
++-----+-----+-----+
| |Instance |Module |Cells |
++-----+-----+-----+
|1 |top | | 16|
++-----+-----+-----+
-----
Finished Writing Synthesis Report : Time (s): cpu = 00:00:09 ; elapsed = 00:00:16 . Memory (MB): peak = 1018.242 ; gain = 0.000
```

Power Report:

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: **0.1 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 9JA: **1.9°C/W**
Power supplied to off-chip devices: **0 W**
Confidence level: **Low**



Code 16 & 17: 8 Bit Adder & Subtractor

Verilog Code:

```
| module par_sub(a,b,cin,diff,bout);
|   input [7:0] a;
|   input [7:0] b;
|   input cin;
|   output reg [7:0] diff;
|   output reg bout;
|   reg [8:0] c;
|   integer i;
|   always @ (a or b or cin)
|   begin
|     c[0]=cin;
|     if (cin == 0) begin
|       for ( i=0; i<8 ; i=i+1)
|       begin
|         diff[i]= a[i]^b[i]^c[i];
|         c[i+1]= (a[i]&b[i])|(a[i]&c[i])|(b[i]&c[i]);
|       end
|     end
|     else if (cin == 1) begin
|       for ( i=0; i<8 ; i=i+1)
|       begin
|         diff[i]= a[i]^(~ b[i])^c[i];
|         c[i+1]= (a[i]&(~b[i]))|(a[i]&c[i])|((~b[i])&c[i]);
|       end
|     end
|     bout=c[8];
|   end
| endmodule
```

Test Bench:

```
module par_sub_tb
reg [7:0] a;
reg [7:0] b;
reg cin;
wire [7:0] diff;
wire bout;

par_sub_uut (.a(a),.b(b),.cin(cin),.diff(diff),.bout(bout) );

initial begin

#10 a=8'b00000001;b=8'b00000001;cin=l'b0;

#10 a=8'b00000001;b=8'b00000001;cin=l'b1;

#10 a=8'b00000010;b=8'b00000011;cin=l'b0;

#10 a=8'b10000001;b=8'b10000001;cin=l'b0;

#10 a=8'b00011001;b=8'b00110001;cin=l'b0;

#10 a=8'b00000011;b=8'b00000011;cin=l'b1;

#10 a=8'b11111111;b=8'b00000001;cin=l'b0;

#10 a=8'b11111111;b=8'b00000000;cin=l'b1;

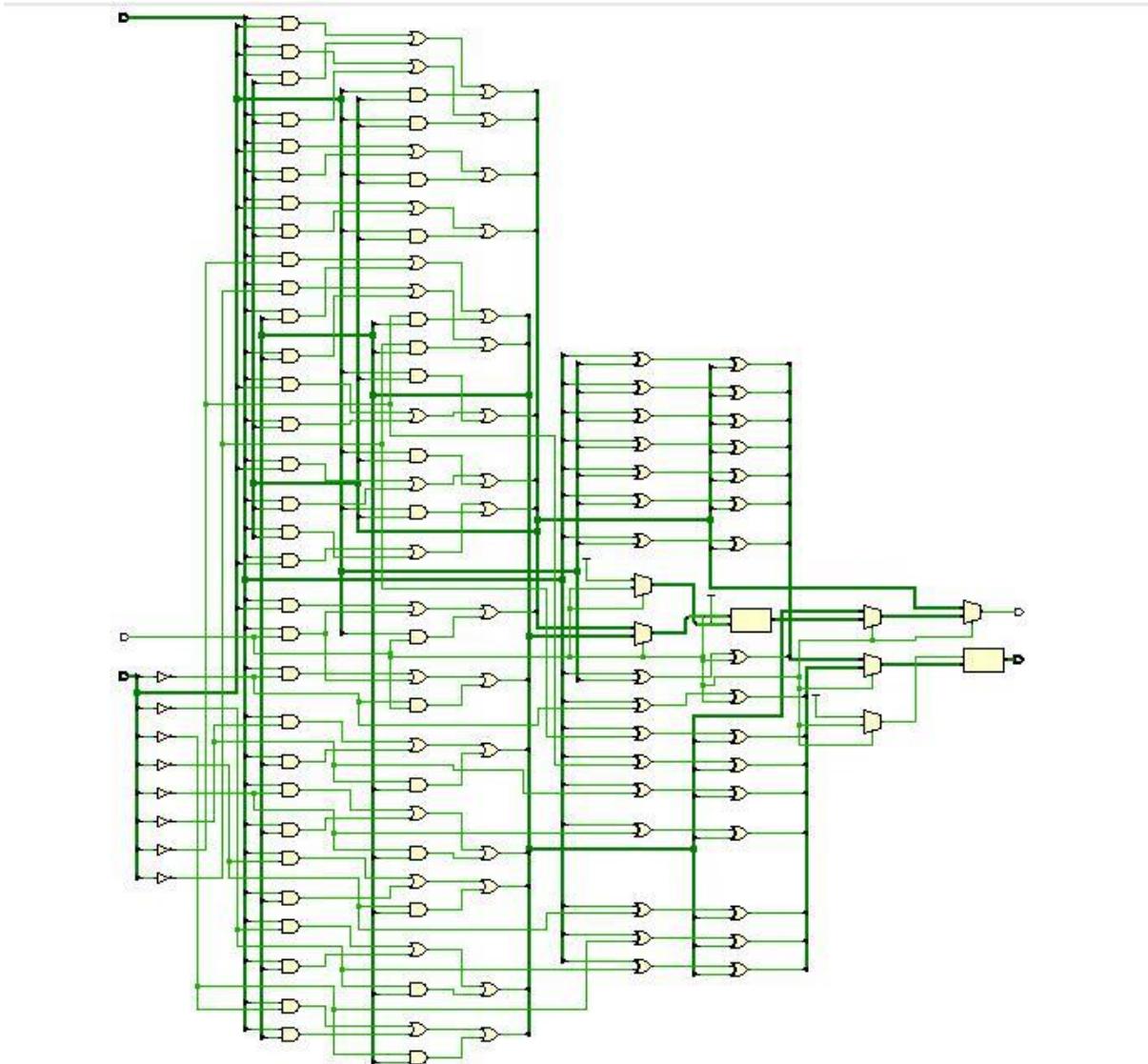
#10 a=8'b11111111;b=8'b11111111;cin=l'b0;

#10 $stop;

end

endmodule
```

RTL Schematic:



Synthesis Report:

```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
+---+---+  
| |BlackBox name |Instances |  
+---+---+  
+---+---+  
  
Report Cell Usage:  
+----+----+  
| |Cell |Count |  
+----+----+  
|1 |LUT2 | 1|  
|2 |LUT3 | 6|  
|3 |LUT5 | 14|  
|4 |IBUF | 17|  
|5 |OBUF | 9|  
+----+----+  
  
Report Instance Areas:  
+----+----+----+  
| |Instance |Module |Cells |  
+----+----+----+  
|1 |top | | 47|  
+----+----+----+  
  
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:09 ; elapsed = 00:00:27 . Memory (MB): peak = 1017.555 ; gain = 0.000
```

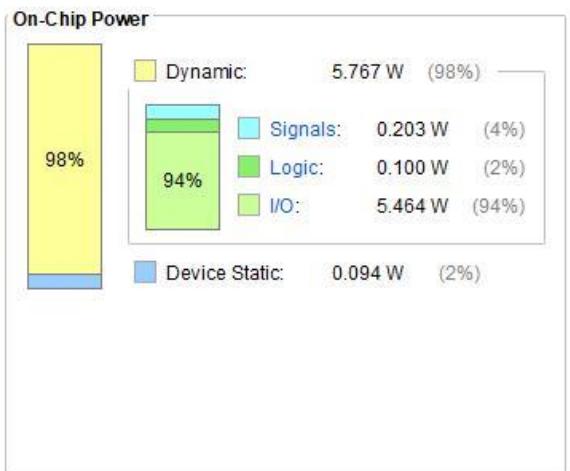
Power Report:

Summary

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

Total On-Chip Power:	5.862 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	36.0°C
Thermal Margin:	49.0°C (25.8 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



Code 18: 4 Bit Multiplier

Verilog Code:

```
timescale 1ns / 1ps
module multiplier_4_x_4(product,inpl,inp2);

    output [7:0]product;
    input [3:0]inpl;
    input [3:0]inp2;

    assign product[0]=(inpl[0]&inp2[0]);

    wire x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16,x17;

    HA HA1(product[1],x1,(inpl[1]&inp2[0]),(inpl[0]&inp2[1]));
    FA FA1(x2,x3,inpl[1]&inp2[1],(inpl[0]&inp2[2]),x1);
    FA FA2(x4,x5,(inpl[1]&inp2[2]),(inpl[0]&inp2[3]),x3);
    HA HA2(x6,x7,(inpl[1]&inp2[3]),x5);

    HA HA3(product[2],x15,x2,(inpl[2]&inp2[0]));
    FA FA5(x14,x16,x4,(inpl[2]&inp2[1]),x15);
    FA FA4(x13,x17,x6,(inpl[2]&inp2[2]),x16);
    FA FA3(x9,x8,x7,(inpl[2]&inp2[3]),x17);

    HA HA4(product[3],x12,x14,(inpl[3]&inp2[0]));
    FA FA8(product[4],x11,x13,(inpl[3]&inp2[1]),x12);
    FA FA7(product[5],x10,x9,(inpl[3]&inp2[2]),x11);
    FA FA6(product[6],product[7],x8,(inpl[3]&inp2[3]),x10);

endmodule

module HA(sout,cout,a,b);
    output sout,cout;
    input a,b;
    assign sout=a^b;
    assign cout=(a&b);
endmodule

module HA(sout,cout,a,b);
    output sout,cout;
    input a,b;
    assign sout=a^b;
    assign cout=(a&b);
endmodule

module FA(sout,cout,a,b,cin);
    output sout,cout;
    input a,b,cin;
    assign sout=(a^b^cin);
    assign cout=((a&b) | (a&cin) | (b&cin));
endmodule
```

Test Bench:

```
) module tb;

    reg [3:0]inpl;
    reg [3:0]inp2;
    wire [7:0]product;

    multiplier_4_x_4 uut(.inpl(inpl),.inp2(inp2),.product(product));

) initial
) begin
    inpl=10;
    inp2=12;
    #30 ;

    inpl=13;
    inp2=12;
    #30 ;

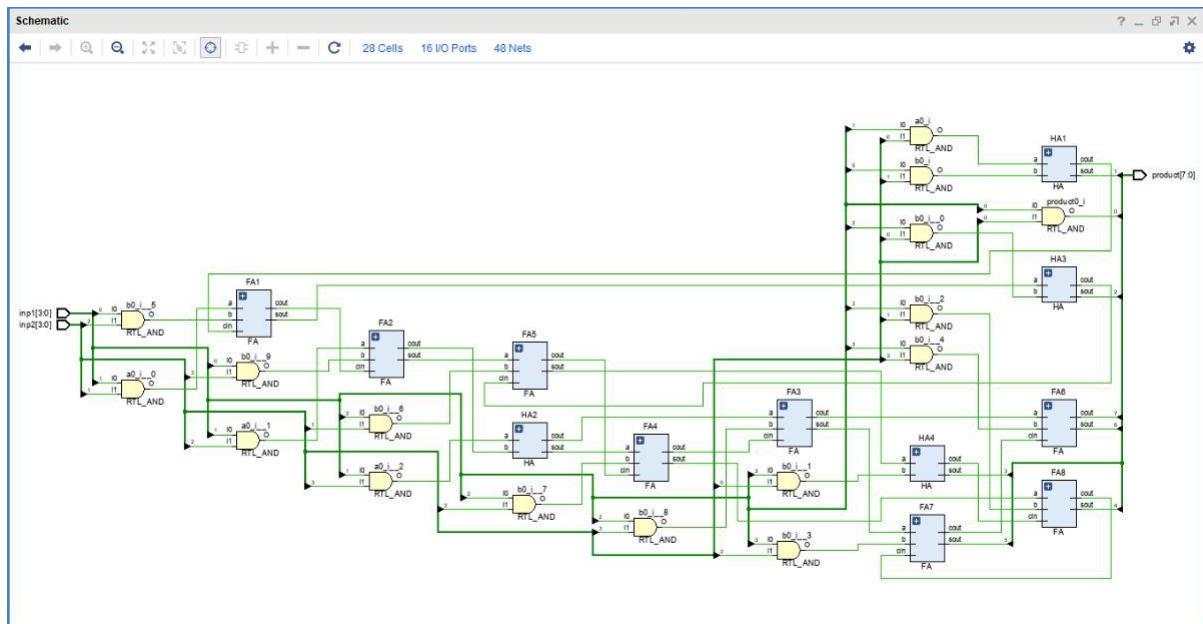
    inpl=10;
    inp2=22;
    #30 ;

    inpl=11;
    inp2=22;
    #30 ;

    inpl=12;
    inp2=15;
    #30 ;

    $finish;
) end
) endmodule
```

RTL Schematic:



Synthesis Report:

```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
++-----+-----+  
| |BlackBox name |Instances |  
++-----+-----+  
++-----+-----+  
  
Report Cell Usage:  
++-----+-----+  
| |Cell |Count |  
++-----+-----+  
|1 |LUT2 | 1|  
|2 |LUT4 | 6|  
|3 |LUT6 | 11|  
|4 |IBUF | 8|  
|5 |OBUF | 8|  
++-----+-----+  
  
Report Instance Areas:  
++-----+-----+-----+  
| |Instance |Module |Cells |  
++-----+-----+-----+  
|1 |top | | 34|  
++-----+-----+-----+  
  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:15 ; elapsed = 00:00:30 . Memory (MB): peak = 1014.992 ; gain = 0.000
```

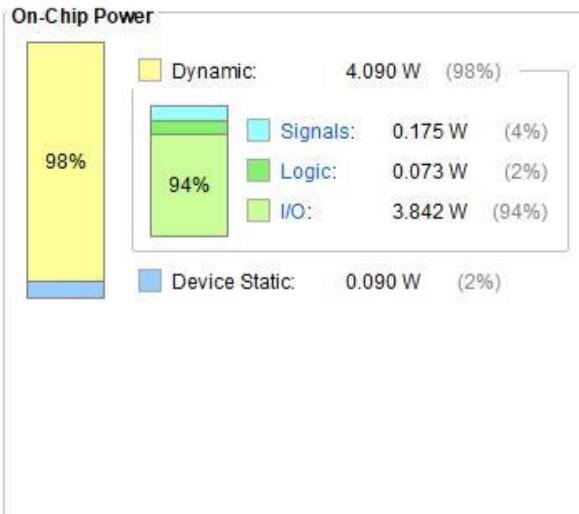
Power Report:

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:	4.18 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	32.9°C
Thermal Margin:	52.1°C (27.5 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



Code 19: Fixed Point Division

Verilog Code:



The screenshot shows a Verilog code editor window with the file "fixed_p_div.v" open. The code implements a fixed-point division module named qdiv. It includes parameter declarations for Q=15 and N=32, input ports for dividend, divisor, start, clock, and output ports for quotient, complete, and overflow. The code uses registers for working quotient, quotient, dividend, divisor, and count, along with done, sign, and overflow flags. It includes initializations for these variables and logic for handling the clock edge and performing the division calculation.

```
fixed_p_div.v
E:/projects/dsd/Fixed point division/Fixed point division.srsc/sources_1/newfixed_p_div.v

Q | I | < | > | X | E | D | X | // | B | Q | 

1 module qdiv #( 
2   parameter O = 15,
3   parameter N = 32
4 )
5 (
6   input  [N-1:0] i_dividend,
7   input  [N-1:0] i_divisor,
8   input  i_start,
9   input  i_clk,
10  output [N-1:0] o_quotient_out,
11  output  o_complete,
12  output  o_overflow
13 );
14
15 reg [2^N+Q-3:0] reg_working_quotient;
16 reg [N-1:0]      reg_quotient;
17 reg [N-2:Q:0]     reg_working_dividend;
18 reg [2^N+Q-3:0] reg_working_divisor;
19
20 reg [N-1:0]      reg_count;
21
22
23 reg           reg_done;
24 reg           reg_sign;
25 reg           reg_overflow;
26
27
28 initial reg_done = 1'b1;
29 initial reg_overflow = 1'b0;
30 initial reg_sign = 1'b0;
31
32 initial reg_working_quotient = 0;
33 initial reg_quotient = 0;
34 initial reg_working_dividend = 0;
35 initial reg_working_divisor = 0;

initial reg_quotient = 0;
initial reg_working_dividend = 0;
initial reg_working_divisor = 0;
initial reg_count = 0;

assign o_quotient_out[N-2:0] = reg_quotient[N-2:0];
assign o_quotient_out[N-1] = reg_sign;
assign o_complete = reg_done;
assign o_overflow = reg_overflow;

always @(posedge i_clk) begin
  if( reg_done && i_start ) begin

    reg_done <= 1'b0;
    reg_count <= N+Q-1;
    reg_working_quotient <= 0;
    reg_working_dividend <= 0;
    reg_working_divisor <= 0;
    reg_overflow <= 1'b0;

    reg_working_dividend[N+Q-2:Q] <= i_dividend[N-2:0];
    reg_working_divisor[2^N+Q-3:N+Q-1] <= i_divisor[N-2:0];

    reg_sign <= i_dividend[N-1] ^ i_divisor[N-1];
    end
  else if(!reg_done) begin
    reg_working_divisor <= reg_working_divisor >> 1;
    reg_count <= reg_count - 1;

    // If the dividend is greater than the divisor
    if(reg_working_dividend >= reg_working_divisor) begin
      reg_working_quotient[reg_count] <= 1'b1;
      reg_working_dividend <= reg_working_dividend - reg_working_divisor;
    end
  end
end
```

```

//stop condition
if(reg_count == 0) begin
    reg_done <= 1'b1;
    reg_quotient <= reg_working_quotient;
    if (reg_working_quotient[2*N+Q-3:N]>0)
        reg_overflow <= 1'bl;
    end
else
    reg_count <= reg_count - 1;
end
end
endmodule

```

Test Bench:

```

module Test_Div;

// Inputs
reg [31:0] i_dividend;
reg [31:0] i_divisor;
reg i_start;
reg i_clk;

// Outputs
wire [31:0] o_quotient_out;
wire o_complete;
wire o_overflow;

// Instantiate the Unit Under Test (UUT)
qdiv uut (
    .i_dividend(i_dividend),
    .i_divisor(i_divisor),
    .i_start(i_start),
    .i_clk(i_clk),
    .o_quotient_out(o_quotient_out),
    .o_complete(o_complete),
    .o_overflow(o_overflow)
);

reg [10:0] count;

initial begin
    // Initialize Inputs
    i_dividend = 1;
    i_divisor = 1;
    i_start = 0;
    i_clk = 0;
    count <= 0;

```

```

// Wait 100 ns for global reset to finish
#100;

// Add stimulus here
forever #2 i_clk = ~i_clk;
end

always @(posedge i_clk) begin
    if (count == 47) begin
        count <= 0;
        i_start <= 1'b1;
        end
    else begin
        count <= count + 1;
        i_start <= 1'b0;
        end
    end

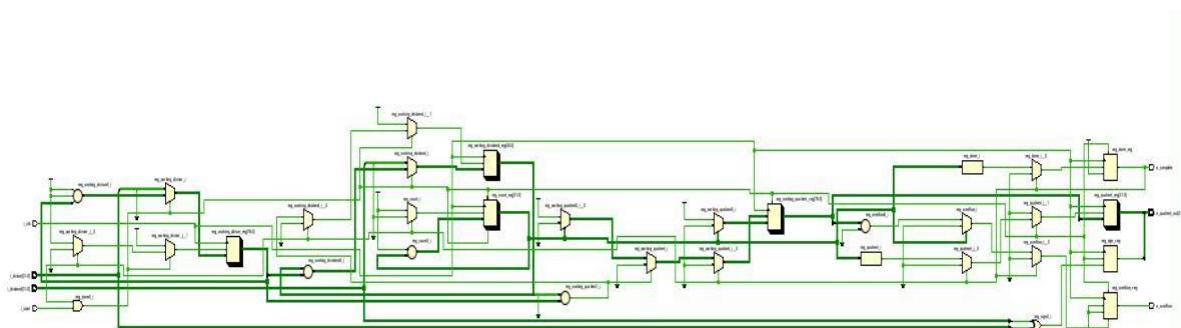
always @ (count) begin
    if (count == 47) begin
        if ( i_divisor > 32'h1FFFFFFF ) begin
            i_divisor <= 1;
            i_dividend = (i_dividend << 1) + 3;
            end
        else
            i_divisor = (i_divisor << 1) + 1;
        end
    end
end

always @(posedge o_complete)
$display ("%b,%b,%b, %b", i_dividend, i_divisor, o_quotient_out, o_overflow);

endmodule

```

RTL Schematic:



Synthesis Report:

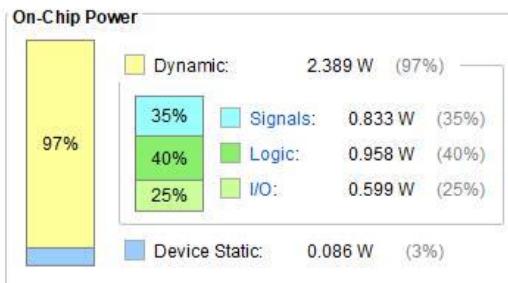
```
Start Writing Synthesis Report
-----
Report BlackBoxes:
++-----+-----+
| |BlackBox name |Instances |
++-----+-----+
++-----+-----+
Report Cell Usage:
++-----+-----+
| |Cell |Count |
++-----+-----+
|1 |BUFG | 1|
|2 |CARRY4 | 30|
|3 |LUT1 | 34|
|4 |LUT2 | 20|
|5 |LUT3 | 121|
|6 |LUT4 | 102|
|7 |LUT5 | 39|
|8 |LUT6 | 27|
|9 |FDRE | 261|
|10 |FDSE | 4|
|11 |IBUF | 66|
|12 |OBUF | 34|
++-----+-----+
Report Instance Areas:
++-----+-----+
| |Instance |Module |Cells |
++-----+-----+
|1 |top | | 739|
++-----+-----+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:17 ; elapsed = 00:00:30 . Memory (MB): peak = 1019.531 ; gain = 0.000
```

Power Report:

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: **2.475 W**
Design Power Budget: **Not Specified**
Power Budget Margin: **N/A**
Junction Temperature: **29.7°C**
Thermal Margin: **55.3°C (29.2 W)**
Effective 9JA: **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



Code 20: Master Slave JK Flip Flop

Verilog Code:

```
module jk_flip_flop_master_slave(Q, Qn, C, J, K, RESETn);
    output Q;
    output Qn;
    input C;
    input J;
    input K;
    input RESETn;

    wire MQ;
    wire MQn;
    wire Cn;
    wire J1;
    wire K1;
    wire J2;
    wire K2;
    assign J2 = !RESETn ? 0 : J1;
    assign K2 = !RESETn ? 1 : K1;

    and(J1, J, Qn);
    and(K1, K, Q);
    not(Cn, C);
    sr_latch_gated master(MQ, MQn, C, J2, K2);
    sr_latch_gated slave(Q, Qn, Cn, MQ, MQn);
endmodule

module sr_latch_gated(Q, Qn, G, S, R);
    output Q;
    output Qn;
    input G;
    input S;
    input R;

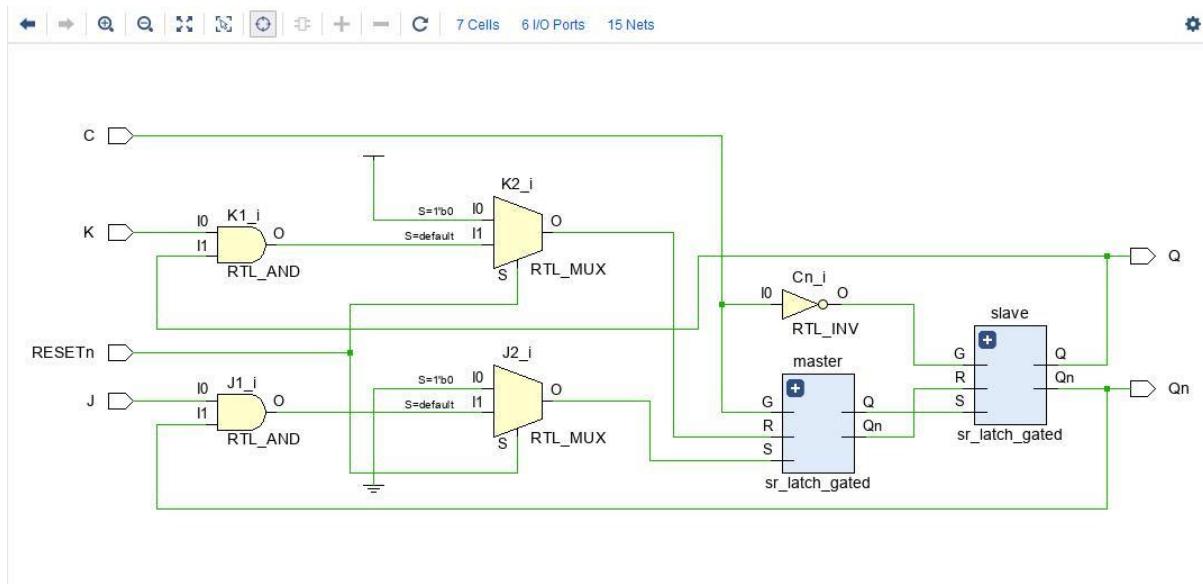
    wire S1;
    wire R1;

    and(S1, G, S);
    and(R1, G, R);
    nor(Qn, S1, Q);
    nor(Q, R1, Qn);
endmodule
```

Test Bench:

```
) module JK_ff_tb;  
  
reg C, J, K, RESETn;  
  
wire Q;  
wire Qn;  
  
jk_flip_flop_master_slave jkflipflop( .C(C), .RESETn(RESETn), .J(J), .K(K), .Q(Q), .Qn(Qn) );  
  
) initial begin  
    $dumpfile("dump.vcd"); $dumpvars;  
    $monitor(C,J,Q,Qn,RESETn);  
  
    J = 1'b0;  
    K = 1'b0;  
    RESETn = 1;  
    C=1;  
  
    #10  
    RESETn=0;  
    J=1'b1;  
    K=1'b0;  
  
    #100  
    RESETn=0;  
    J=1'b0;  
    K=1'b1;  
  
    #100  
    RESETn=0;  
    J=1'b1;  
    K=1'b1;  
  
    #100  
    RESETn=0;  
  
    #100  
    RESETn=0;  
    J=1'b1;  
    K=1'b1;  
  
    #100  
    RESETn=0;  
    J=1'b0;  
    K=1'b1;  
  
    #100  
    RESETn=0;  
    J=1'b0;  
    K=1'b0;  
  
    #100  
    RESETn=1;  
    J=1'b1;  
    K=1'b0;  
  
> end  
always #25 C <= ~C;  
  
> endmodule
```

RTL Schematic:



Synthesis Report:

```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
+---+  
| |BlackBox name |Instances |  
+---+  
+---+  
  
Report Cell Usage:  
+---+  
| |Cell |Count |  
+---+  
|1 |LUT3 | 2|  
|2 |LUT6 | 2|  
|3 |IBUF | 4|  
|4 |OBUF | 2|  
+---+  
  
Report Instance Areas:  
+---+  
| |Instance |Module |Cells |  
+---+  
|1 |top | | 10|  
+---+  
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:15 ; elapsed = 00:00:28 . Memory (MB): peak = 1015.500 ; gain = 0.000
```

Power Report:

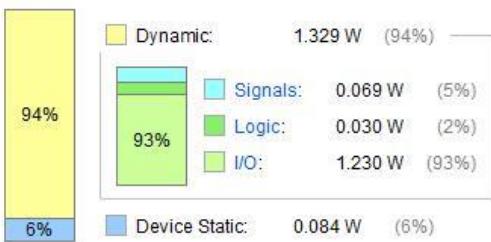
Summary

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:	1.413 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	27.7°C
Thermal Margin:	57.3°C (30.2 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



Code 21: Positive Edge Detector

Verilog Code:

```
module pos_edge_detect(clk,nrst,din,dout);
    input clk;
    input nrst;
    input din;
    output dout;
    reg d_ff;

    always @(posedge clk or negedge nrst)
    begin
        if(!nrst)
            d_ff<=1'b0;
        else
            d_ff<=din;
    end
    assign dout=din&&(d_ff^din);
endmodule

module d_ff(D,C,a);
    input D;
    input C;
    output a;
    reg a;
    always @(posedge C)
    begin
        a <= D;
    end
endmodule
```

Test Bench:

```
module tb;
    reg nrst;
    reg clk;
    reg din;
    wire dout;

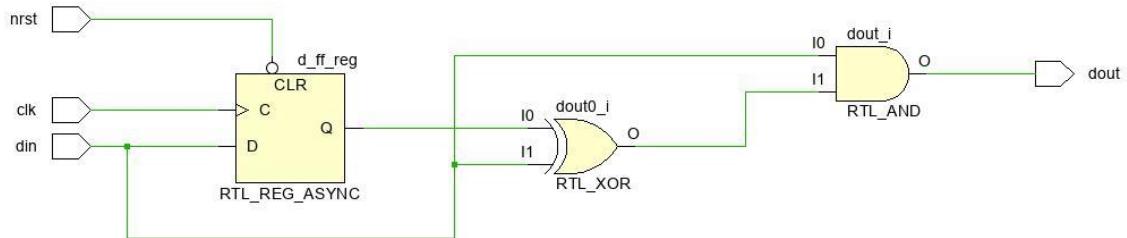
    pos_edge_det ped0 (.nrst(nrst),
                      .clk(clk),
                      .din(din),.dout(dout));

    always #5 clk = ~clk;

initial begin
    clk <= 0;
    nrst <= 0;
    #15 nrst<= 1;
    #20 nrst<= 0;
    #15 nrst<= 1;
    #10 nrst <= 0;
    #20 $finish;
end

initial begin
    $dumpvars;
    $dumpfile("dump.vcd");
end
endmodule
```

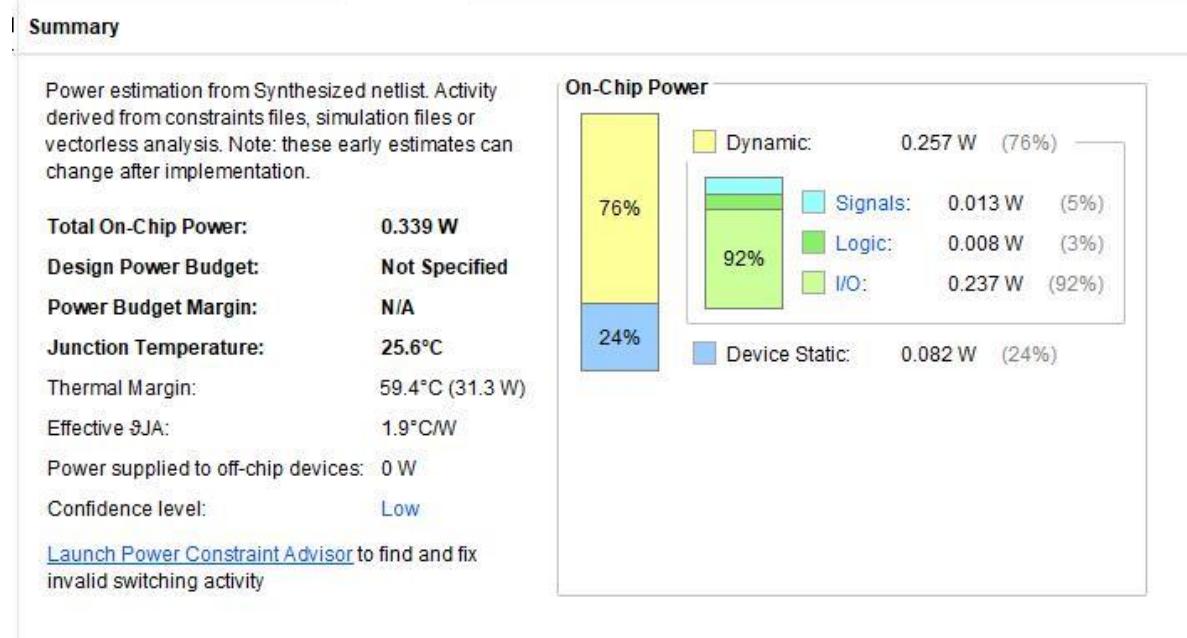
RTL Schematic:



Synthesis Report:

```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
+---+---+  
| |BlackBox name |Instances |  
+---+---+  
+---+---+  
  
Report Cell Usage:  
+---+---+  
| |Cell |Count |  
+---+---+  
|1| BUFG | 1|  
|2| LUT1 | 1|  
|3| LUT2 | 1|  
|4| FDCE | 1|  
|5| IBUF | 3|  
|6| OBUF | 1|  
+---+---+  
  
Report Instance Areas:  
+---+---+---+  
| |Instance |Module |Cells |  
+---+---+---+  
|1| top | | 8|  
+---+---+---+  
  
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:20 ; elapsed = 00:00:41 . Memory (MB): peak = 1018.688 ; gain = 0.000  
-----
```

Power Report:



Code 22: BCD Adder

Verilog Code:

```
) module bcd_adder(a,b,carry_in,sum,carry);

    input [3:0] a,b;
    input carry_in;
    output [3:0] sum;
    output carry;

    reg [4:0] sum_temp;
    reg [3:0] sum;
    reg carry;

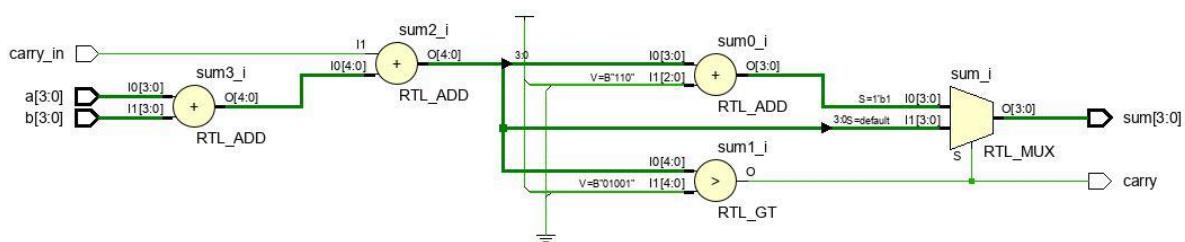
)   always @(a,b,carry_in)
)   begin
    sum_temp = a+b+carry_in;
)   if(sum_temp > 9)      begin
        sum_temp = sum_temp+6;
        carry = 1;
)   sum = sum_temp[3:0];    end
)   else      begin
        carry = 0;
        sum = sum_temp[3:0];
)   end
)   end

) endmodule
```

Test Bench:

```
| module tb_bcdadder;  
  
|   reg [3:0] a;  
|   reg [3:0] b;  
|   reg carry_in;  
  
|   wire [3:0] sum;  
|   wire carry;  
  
|   bcd_adder uut (  
|     .a(a),  
|     .b(b),  
|     .carry_in(carry_in),  
|     .sum(sum),  
|     .carry(carry)  
|   );  
  
|   initial begin  
  
|     a = 0;  b = 0;  carry_in = 0;  #100;  
|     a = 6;  b = 9;  carry_in = 0;  #100;  
|     a = 3;  b = 3;  carry_in = 1;  #100;  
|     a = 4;  b = 5;  carry_in = 0;  #100;  
|     a = 8;  b = 2;  carry_in = 0;  #100;  
|     a = 9;  b = 9;  carry_in = 1;  #100;  
|   end  
  
| endmodule
```

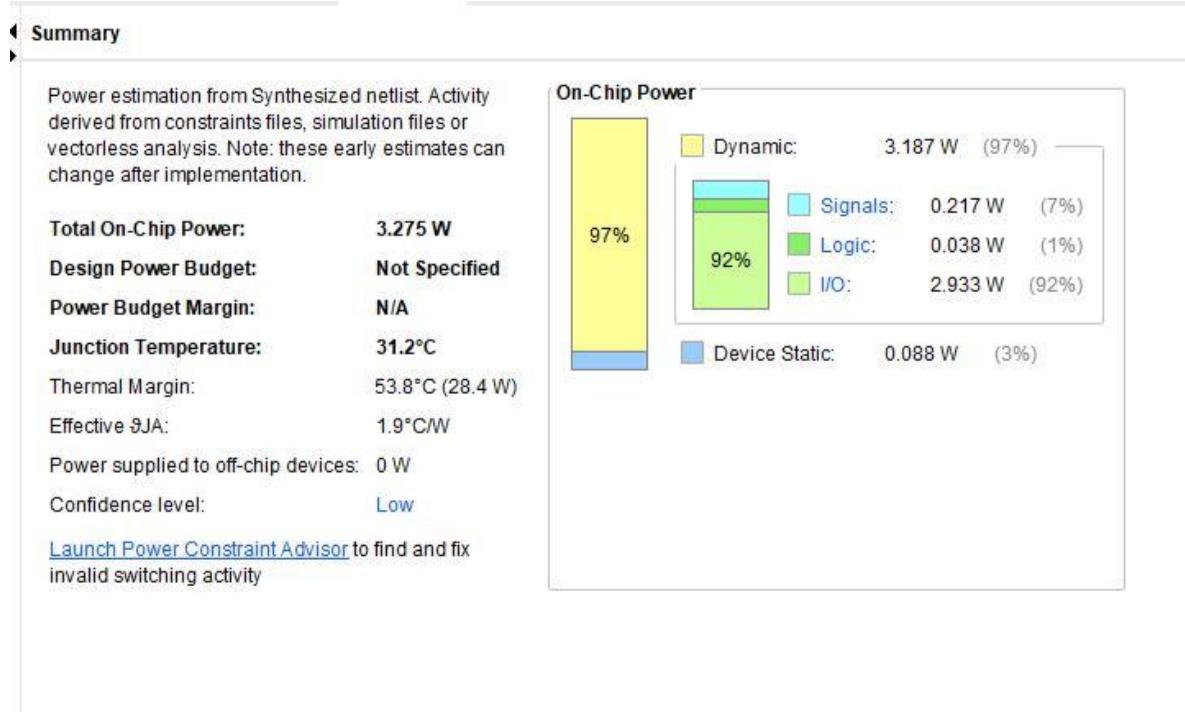
RTL Schematic:



Synthesis Report:

```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
++-----+  
| |BlackBox name | Instances |  
++-----+  
++-----+  
  
Report Cell Usage:  
++-----+  
| |Cell |Count |  
++-----+  
|1 |LUT3 | 1|  
|2 |LUT5 | 2|  
|3 |LUT6 | 4|  
|4 |IBUF | 9|  
|5 |OBUF | 5|  
++-----+  
  
Report Instance Areas:  
++-----+  
| |Instance |Module |Cells |  
++-----+  
|1 |top | | 21|  
++-----+  
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:23 ; elapsed = 00:00:44 . Memory (MB): peak = 1016.285 ; gain = 0.000
```

Power Report:



Code 23: 4 Bit Carry Select Adder

Verilog Code:

```
) module carry_select_adder
  (  input [3:0] A,B,
    input cin,
    output [3:0] S,
    output cout
  );
  wire [3:0] temp0,temp1,carry0,carry1;

  fulladder fa00(A[0],B[0],1'b0,temp0[0],carry0[0]);
  fulladder fa01(A[1],B[1],carry0[0],temp0[1],carry0[1]);
  fulladder fa02(A[2],B[2],carry0[1],temp0[2],carry0[2]);
  fulladder fa03(A[3],B[3],carry0[2],temp0[3],carry0[3]);

  fulladder fa10(A[0],B[0],1'b1,temp1[0],carry1[0]);
  fulladder fa11(A[1],B[1],carry1[0],temp1[1],carry1[1]);
  fulladder fa12(A[2],B[2],carry1[1],temp1[2],carry1[2]);
  fulladder fa13(A[3],B[3],carry1[2],temp1[3],carry1[3]);

  multiplexer2 mux_carry(carry0[3],carry1[3],cin,cout);
  multiplexer2 mux_sum0(temp0[0],temp1[0],cin,S[0]);
  multiplexer2 mux_sum1(temp0[1],temp1[1],cin,S[1]);
  multiplexer2 mux_sum2(temp0[2],temp1[2],cin,S[2]);
  multiplexer2 mux_sum3(temp0[3],temp1[3],cin,S[3]);

) endmodule
```

```
) module fulladder
    (    input a,b,cin,
        output sum,carry
    );
    assign sum = a ^ b ^ cin;
    assign carry = (a & b) | (cin & b) | (a & cin);
) endmodule

) module multiplexer2
    (    input i0,i1,sel,
        output reg bitout
    );
    always@(i0,i1,sel)
    begin
        if(sel == 0)
            bitout = i0;
        else
            bitout = i1;
    end
) endmodule
```

Test Bench:

```
) module tb_adder;

    reg [3:0] A;
    reg [3:0] B;
    reg cin;

    wire [3:0] S;
    wire cout;
    integer i,j,error;

    carry_select_adder uut (
        .A(A),
        .B(B),
        .cin(cin),
        .S(S),
        .cout(cout)
    );

) initial begin

    A = 0;
    B = 0;
    error = 0;

    cin = 0;
)    for(i=0;i<16;i=i+1) begin
)        for(j=0;j<16;j=j+1) begin
            A = i;
            B = j;
            #10;
)            if({cout,S} != (i+j))
)
```

```

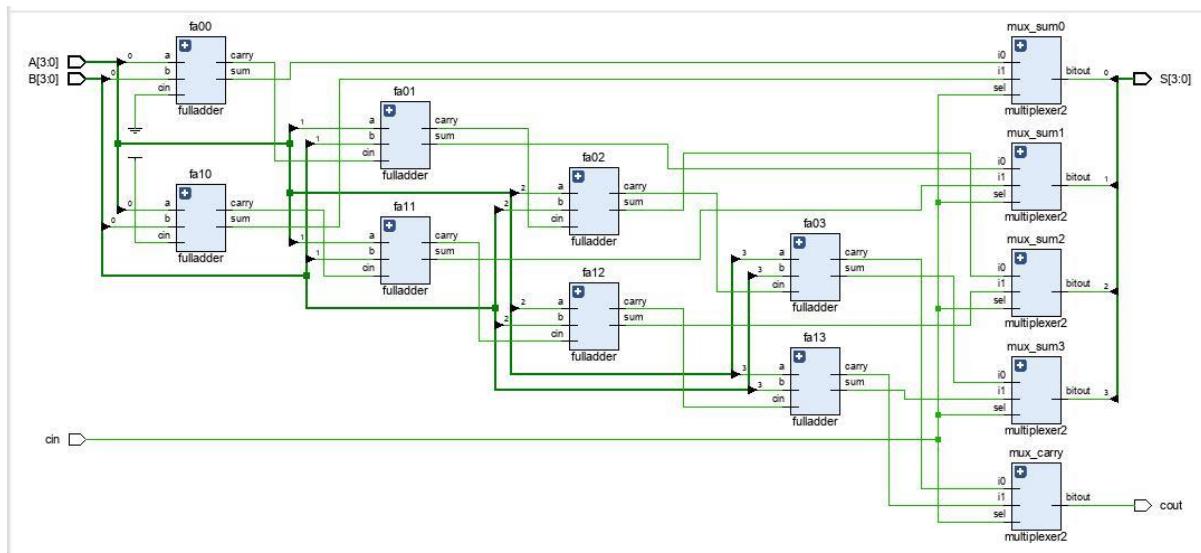
end

cin = 1;
for(i=0;i<16;i=i+1) begin
    for(j=0;j<16;j=j+1) begin
        A = i;
        B = j;
        #10;
        if({cout,S} != (i+j+1))
            error <= error + 1;
    end
end
end

endmodule

```

RTL Schematic:



Synthesis Report:

```
Start Writing Synthesis Report
-----
Report BlackBoxes:
+-----+-----+
| BlackBox name | Instances |
+-----+-----+
+-----+-----+
Report Cell Usage:
+-----+-----+
| Cell | Count |
+-----+-----+
|1 |LUT3 | 2|
|2 |LUT5 | 4|
|3 |IBUF | 9|
|4 |OBUF | 5|
+-----+-----+
Report Instance Areas:
+-----+-----+-----+
| Instance | Module | Cells |
+-----+-----+-----+
|1 |top | 20|
+-----+-----+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:22 ; elapsed = 00:00:43 . Memory (MB): peak = 1017.844 ; gain = 0.000
```

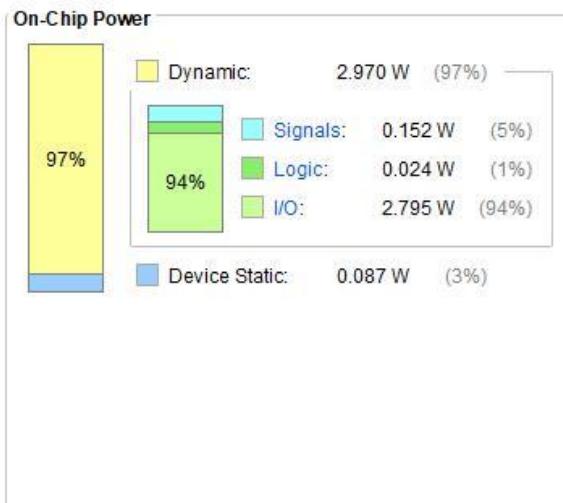
Power Report:

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:	3.058 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	30.8°C
Thermal Margin:	54.2°C (28.6 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



Code 24: Moore FSM 1010 Sequence Detector

Verilog Code:

```
module morfsmolp(din, reset, clk, y);
    input din;
    input clk;
    input reset;
    output reg y;
    reg [2:0] cst, nst;
    parameter S0 = 3'b000,
              S1 = 3'b001,
              S2 = 3'b010,
              S3 = 3'b100,
              S4 = 3'b101;
    always @(cst or din)
        begin
            case (cst)
                S0: if (din == 1'b1)
                    begin
                        nst = S1;
                        y=1'b0;
                    end
                else nst = cst;
                S1: if (din == 1'b0)
                    begin
                        nst = S2;
                        y=1'b0;
                    end
                else
                    begin
                        nst = cst;
                        y=1'b0;
                    end
                S2: if (din == 1'b1)
                    begin
                        nst = S3;
                        y=1'b0;
                    end
            endcase
        end
endmodule
```

```
S3: if (din == 1'b0)
    begin
        nst = S4;
        y=1'b0;
    end
else
begin
    nst = S1;
    y=1'b0;
end
S4: if (din == 1'b0)
begin
    nst = S1;
    y=1'b1;
end
else
begin
    nst = S3;
    y=1'b1;
end
default: nst = S0;
endcase
end
always@(posedge clk)
begin
    if (reset)
        cst <= S0;
    else
        cst <= nst;
end
endmodule
```

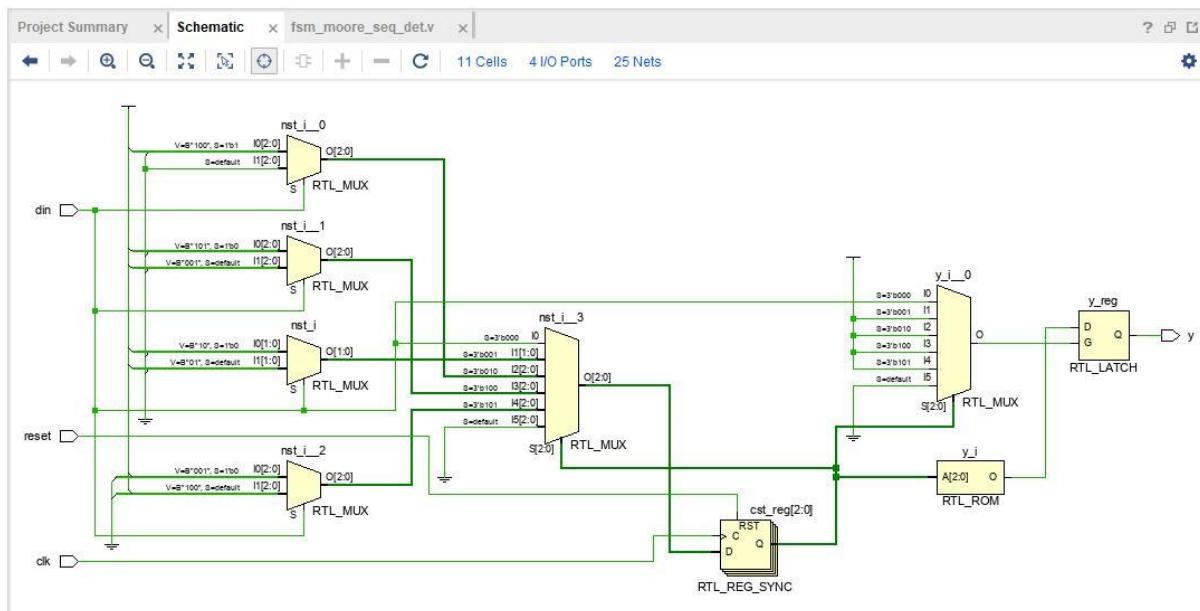
Test Bench:

```

module morfsmolp_tb;
reg din,clk,reset;
wire y;
morfsmolp ml(din, reset, clk, y);
initial
begin
reset=0      ;clk=0;din=0;
$monitor($time, , "c=%b",clk,, "y=%b",y,, "r=%b",reset,, "d=%b",din);
#10 din=1;
#10 din=0;
#10 din=1;
#10 din=0;
end
always
#5 clk=~clk;
initial
#100 $finish ;
endmodule

```

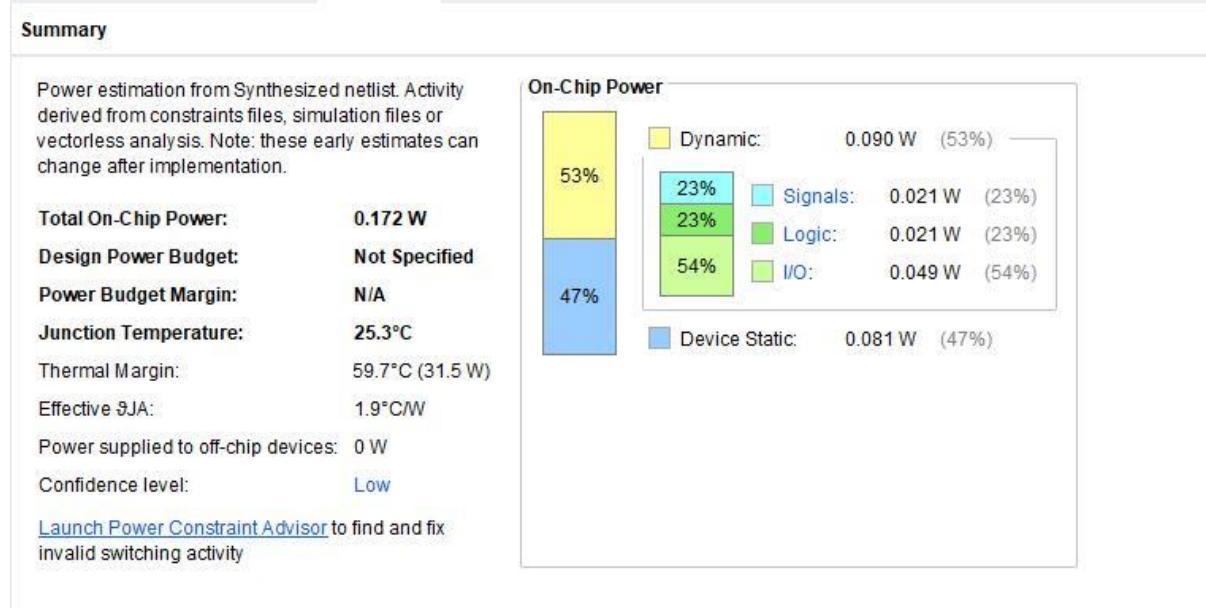
RTL Schematic:



Synthesis Report:

```
Start Writing Synthesis Report
-----
Report BlackBoxes:
+-----+-----+
| |BlackBox name |Instances |
+-----+-----+
+-----+-----+
Report Cell Usage:
+-----+-----+
| |Cell |Count |
+-----+-----+
|1 |BUFGE | 1|
|2 |LUT2 | 2|
|3 |LUT3 | 2|
|4 |LUT5 | 1|
|5 |LUT6 | 1|
|6 |FDRE | 4|
|7 |FDSE | 1|
|8 |LD | 1|
|9 |IBUF | 3|
|10 |OBUF | 1|
+-----+-----+
Report Instance Areas:
+-----+-----+
| |Instance |Module |Cells |
+-----+-----+
|1 |top | | 17|
+-----+-----+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:10 ; elapsed = 00:00:17 . Memory (MB): peak = 1017.680 ; gain = 0.000
```

Power Report:



Code 25: N:1 MUX

Verilog Code:

```
) module mux_4_1(
    input [1:0] sel,
    input i0,i1,i2,i3,
    output reg y);

) always @(*) begin
)   case(sel)
    2'h0: y = i0;
    2'h1: y = i1;
    2'h2: y = i2;
    2'h3: y = i3;
    default: $display("Invalid sel input");
) endcase
) end
) endmodule
```

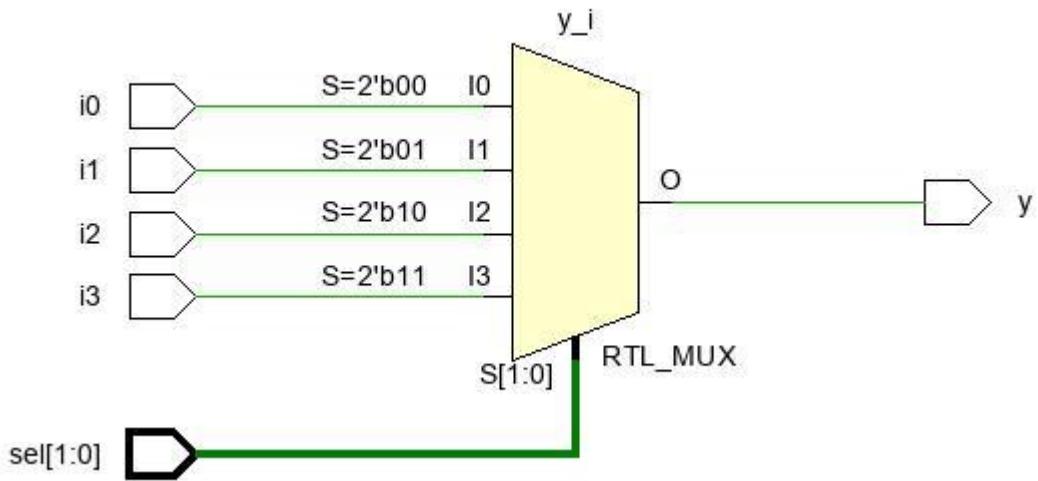
Test Bench:

```
) module tb;
    reg [1:0] sel;
    reg i0,i1,i2,i3;
    wire y;

    mux_example mux(sel, i0, i1, i2, i3, y);

) initial begin
    $monitor("sel = %b -> i3 = %0b, i2 = %0b ,i1 = %0b, i0 = %0b -> y = %0b", sel,i3,i2,i1,i0, y);
    {i3,i2,i1,i0} = 4'h5;
) repeat(6) begin
    sel = $random;
    #5;
) end
) end
) endmodule
```

RTL Schematic:



Synthesis Report:

```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
+---+-----+  
| |BlackBox name |Instances |  
+---+-----+  
+---+-----+  
  
Report Cell Usage:  
+---+-----+  
| |Cell |Count |  
+---+-----+  
|1 |LUT6 | 1|  
|2 |IBUF | 6|  
|3 |OBUF | 1|  
+---+-----+  
  
Report Instance Areas:  
+---+-----+-----+  
| |Instance |Module |Cells |  
+---+-----+-----+  
|1 |top | | 8|  
+---+-----+-----+  
  
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:11 ; elapsed = 00:00:20 . Memory (MB): peak = 1020.262 ; gain = 0.000
```

Power Report:

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:	0.544 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	26.0°C
Thermal Margin:	59.0°C (31.1 W)
Effective gJA:	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

