



ASSIGNMENT 3

Name:- Akshay Singh

ID Number :- 21EL083

Course Code:- 3EL42

Q1) Clock Divider

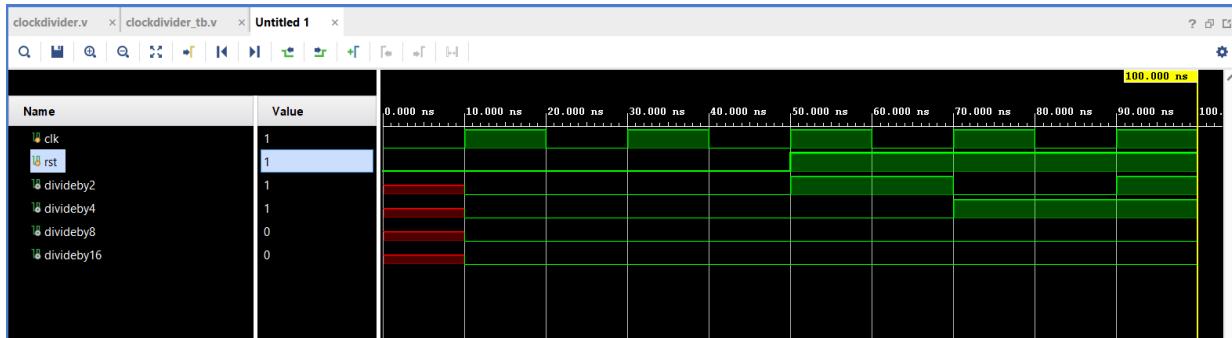
Design: -

```
1 `timescale 1ns / 1ps
2 module clockdivider(clk,divideby2,divideby4,divideby8,divideby16,rst);
3     input clk,rst;
4     reg [3:0]count;
5     output reg divideby2,divideby4,divideby8,divideby16;
6     always@(posedge clk)
7     begin
8         if(rst==0)
9             count=4'b0000;
10        else
11            count=count+1;
12        divideby2=count[0];
13        divideby4=count[1];
14        divideby8=count[2];
15        divideby16=count[3];
16    end
17 endmodule|
```

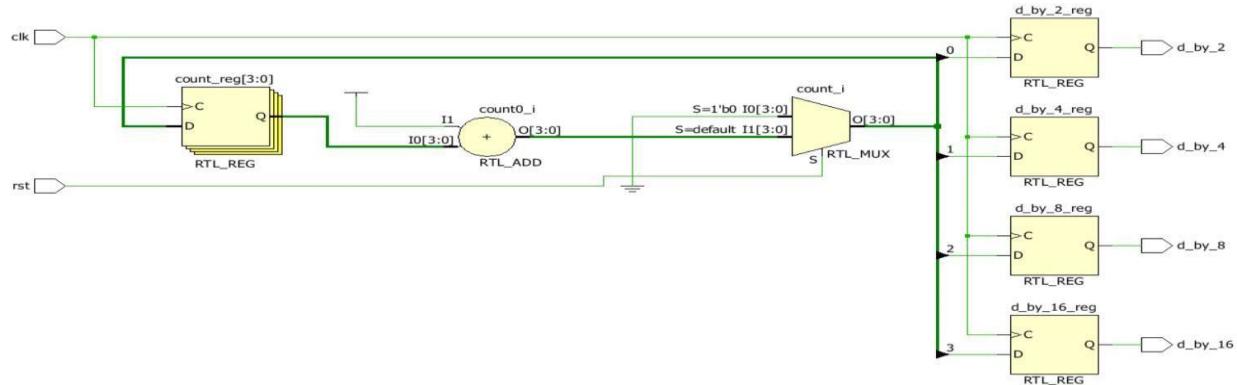
Testbench : -

```
1 `timescale 1ns/1ps;
2 module clockdivider_tb;
3
4     reg clk;
5     reg rst;
6
7     wire divideby2;
8     wire divideby4;
9     wire divideby8;
10    wire divideby16;
11
12    clockdivider uut (
13        .clk(clk),
14        .divideby2(divideby2),
15        .divideby4(divideby4),
16        .divideby8(divideby8),
17        .divideby16(divideby16),
18        .rst(rst)
19    );
20
21
22    initial begin
23        clk = 0;
24        rst = 0;
25        #50 rst=1;
26
27    end
28    always
29        #10 clk=~clk;
30    initial
31        #100 $finish;
32 endmodule
```

Output: -



RTL Schematic:-



Synthesis Report :-

```

Start Writing Synthesis Report
-----
Report BlackBoxes:
++-----+
| |BlackBox name |Instances |
++-----+
+-----+
Report Cell Usage:
+-----+-----+
|     |Cell |Count |
+-----+-----+
|1  |IBUF |    1|
|2  |LUT1 |    2|
|3  |LUT2 |    1|
|4  |LUT3 |    1|
|5  |LUT4 |    1|
|6  |FDRE |    4|
|7  |IBUF |    2|
|8  |OBUF |    4|
+-----+-----+
Report Instance Areas:
+-----+-----+
|     |Instance |Module |Cells |
+-----+-----+
|1  |top   |      |  16|
+-----+-----+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:05 ; elapsed = 00:00:07 . Memory (MB): peak = 412.129 ; gain = 252.445
-----
```

Power Report:-

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: 3.862 W

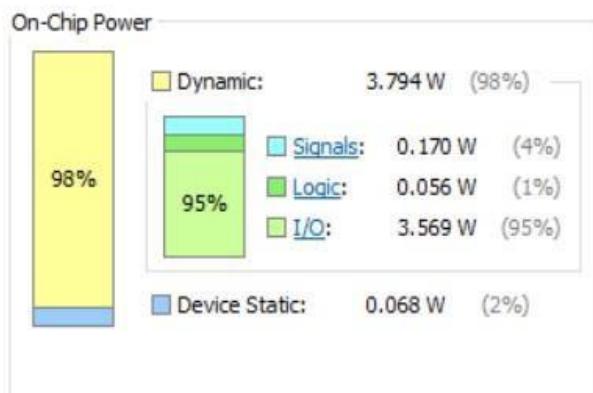
Junction Temperature: 43.5 °C

Thermal Margin: 56.5 °C (11.8 W)

Effective θ_{JA} : 4.8 °C/W

Power supplied to off-chip devices: 0 W

Confidence level: Low



2) Johnson Counter

Design

The screenshot shows a text editor window with three tabs at the top: "johnsncounter.v *", "johnsncounter_tb.v", and "Untitled 2". The current tab is "johnsncounter.v". The code is as follows:

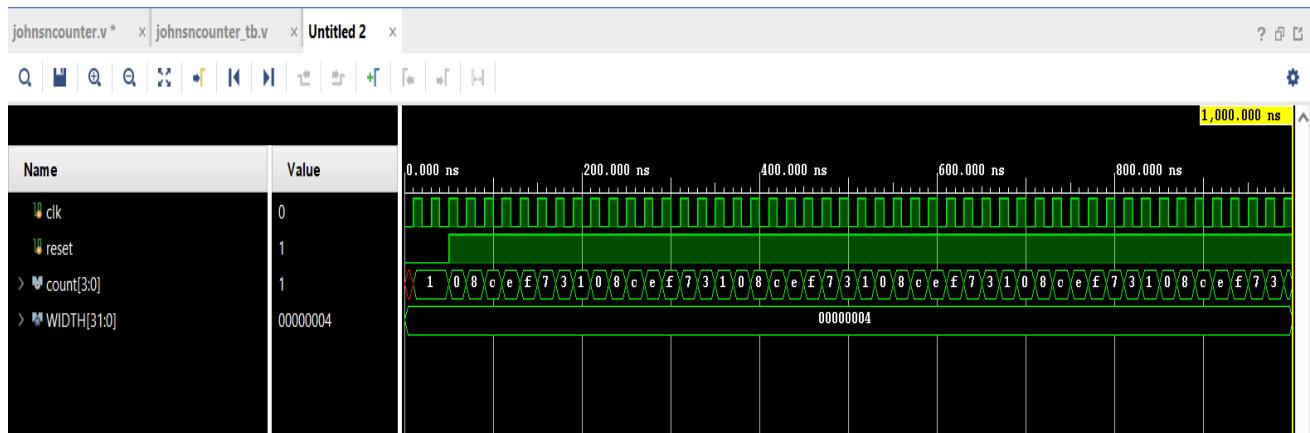
```
1 module johnson_counter(clk,reset,count);
2
3     parameter WIDTH=4;
4
5     input clk,reset;
6
7     output reg [WIDTH-1:0] count;
8
9     always@(posedge clk)
10    begin
11        if(reset)
12            count={~count[0],count[WIDTH-1:1]};
13        else
14            count=1;
15    end
16 endmodule
```

Testbench

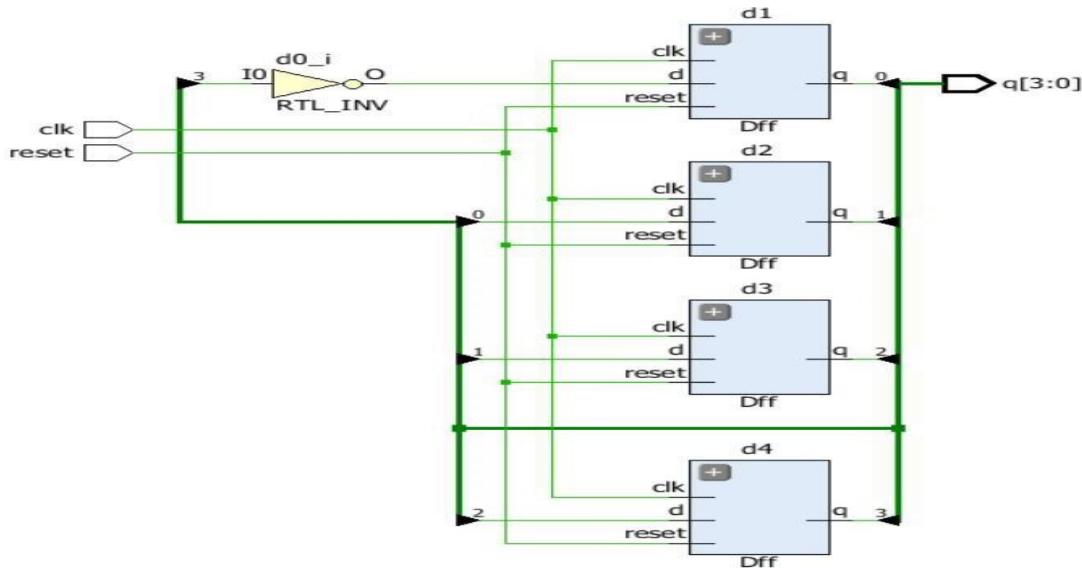
The screenshot shows a text editor window with three tabs at the top: "johnsncounter.v *", "johnsncounter_tb.v", and "Untitled 2". The current tab is "johnsncounter_tb.v". The code is as follows:

```
1 module johnson_counter_tb();
2     parameter WIDTH=4;
3     reg clk,reset;
4     wire [WIDTH-1:0] count;
5     johnson_counter dut(clk,reset,count);
6     always #10 clk=~clk;
7     initial
8     begin
9         reset=0;
10        clk=0;
11        $monitor ("T=%0t out=%b", $time, count);
12        #50 reset=1;
13    end
14 endmodule
```

Output



RTL Schematic:-



SYNTHESIS REPORT :-

Start Writing Synthesis Report

Report BlackBoxes:

	BlackBox name	Instances
+-----+-----+		

Report Cell Usage:

+-----+-----+		
	Cell	Count
+-----+-----+		
1	BUF	1
2	LUT1	1
3	FDRE	4
4	IBUF	2
5	OBUF	4
+-----+-----+		

Report Instance Areas:

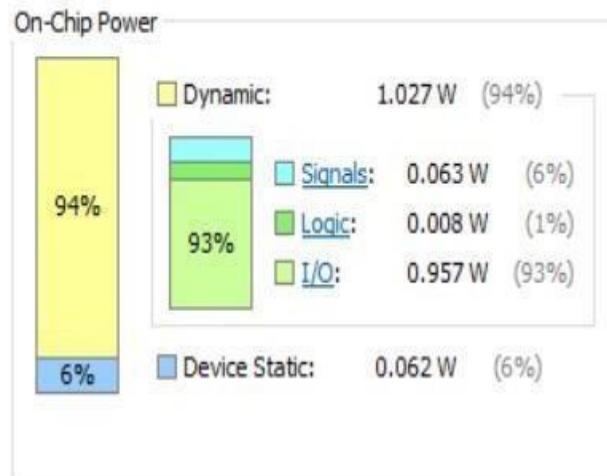
+-----+-----+-----+-----+				
	Instance	Module	Cells	
+-----+-----+-----+-----+				
1	top		12	
2	d1	Dff	1	
3	d2	Dff_0	1	
4	d3	Dff_1	1	
5	d4	Dff_2	2	
+-----+-----+-----+-----+				

Finished Writing Synthesis Report : Time (s): cpu = 00:00:06 ; elapsed = 00:00:07 . Memory (MB): peak = 412.262 ; gain = 252.645

POWER REPORT :-

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: 1.089 W
Junction Temperature: 30.2 °C
Thermal Margin: 69.8 °C (14.5 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



3) Ring Counter

Design

The screenshot shows a text editor window with three tabs at the top: "ringcounter.v", "ringcounter_tb.v", and "Untitled 1". The "ringcounter.v" tab is active, displaying the following Verilog code:

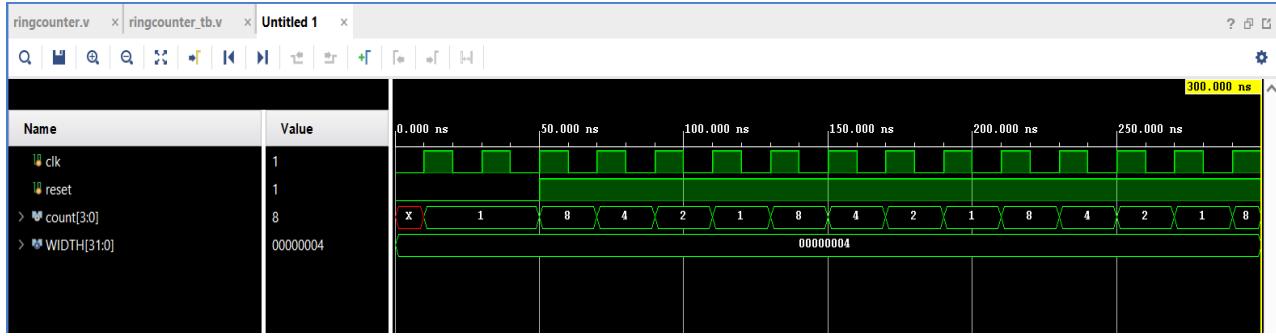
```
1 `timescale 1ns / 1ps
2 module ring_counter( clk,
3   reset, count );
4   parameter WIDTH=4;
5   input clk,reset;
6   output reg [WIDTH-1:0] count;
7   always@(posedge clk)
8 begin
9   if(reset)
10    count={count[0],count[WIDTH-1:1]};
11   else
12    count=4'b0001;
13 end
14 endmodule
```

Testbench :-

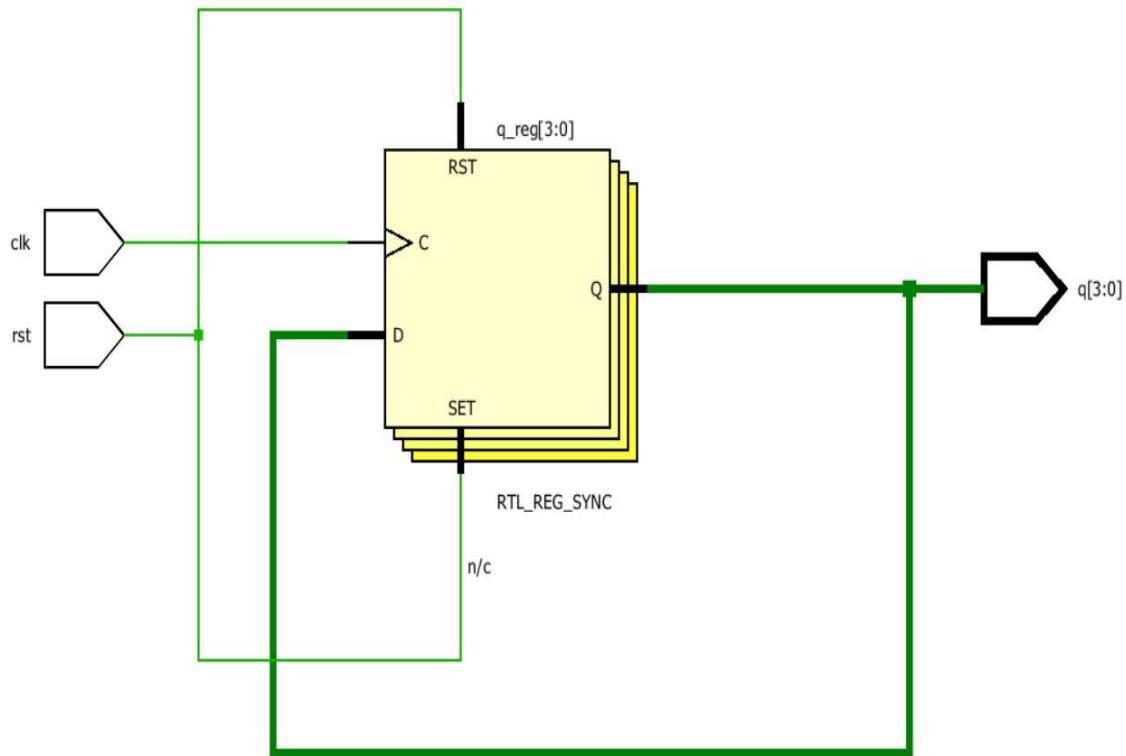
The screenshot shows a text editor window with three tabs at the top: "ringcounter.v", "ringcounter_tb.v", and "Untitled 1". The "ringcounter_tb.v" tab is active, displaying the following Verilog testbench code:

```
1 `timescale 1ns / 1ps
2 module ring_counter_tb;
3   reg clk;
4   reg reset;
5   parameter WIDTH=4;
6
7   wire [WIDTH-1:0] count;
8
9   ring_counter uut (
10     .clk(clk),
11     .reset(reset),
12     .count(count)
13   );
14
15   always #10 clk=~clk;
16
17 initial begin
18   clk = 0;
19   reset = 0;
20   $monitor ("T=%0t out=%b", $time, count);
21   #50 reset=1;
22
23 end
24
25 initial
26   #300 $finish;
27
28 endmodule
```

Output



RTL SCHEMATIC :-



SYNTHESIS REPORT :-

Start Writing Synthesis Report

Report BlackBoxes:

BlackBox name	Instances

Report Cell Usage:

Cell	Count
BUF	1
FDRE	3
FDSE	1
IBUF	2
OBUF	4

Report Instance Areas:

Instance	Module	Cells
top		11

Finished Writing Synthesis Report : Time (s): cpu = 00:00:05 ; elapsed = 00:00:07 . Memory (MB): peak = 403.297 ; gain = 243.977

POWER REPORT :-

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: 1.086 W

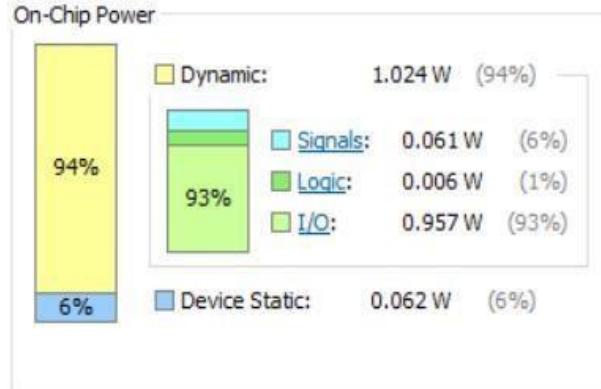
Junction Temperature: 30.2 °C

Thermal Margin: 69.8 °C (14.5 W)

Effective ΔTJA: 4.8 °C/W

Power supplied to off-chip devices: 0 W

Confidence level: Low



4) Input Majority Circuit

Design

The screenshot shows a text editor window with three tabs: "inputmajckt.v", "inputmajckt_tb.v", and "Untitled 3". The "inputmajckt.v" tab is active, displaying Verilog code for a majority circuit. The code defines a module "majority_ckt" with an input vector "x" of width 5:1 and an output "z". It uses 10 AND gates to generate intermediate wires "w" (width 9:0), followed by an OR gate "or1" to produce the final output "z". The code is annotated with comments explaining the instantiation of AND gates.

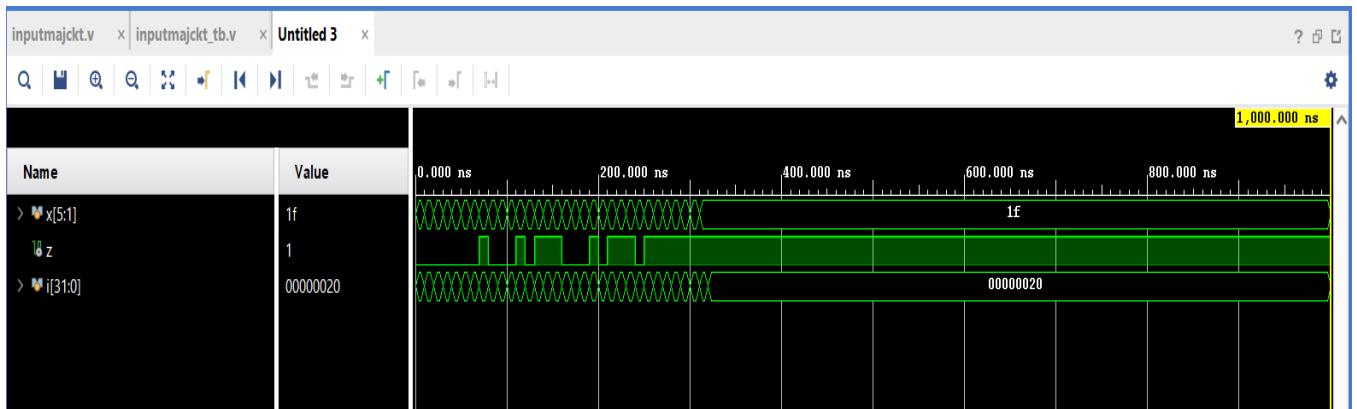
```
1 module majority_ckt(input [5:1] x, output z);
2   wire [9:0] w;
3
4   // instantiate with And Gates
5   and and1(w[0],x[3],x[4],x[5]);
6   and and2(w[1],x[2],x[4],x[5]);
7   and and3(w[2],x[2],x[3],x[5]);
8   and and4(w[3],x[2],x[4],x[3]);
9   and and5(w[4],x[1],x[4],x[5]);
10  and and6(w[5],x[1],x[3],x[5]);
11  and and7(w[6],x[1],x[4],x[3]);
12  and and8(w[7],x[2],x[1],x[5]);
13  and and9(w[8],x[2],x[4],x[1]);
14  and and10(w[9],x[2],x[1],x[3]);
15
16  or or1(z,w[0],w[1],w[2],w[3],w[4],w[5],w[6],w[7],w[8],w[9]);
17
18 endmodule
```

Testbench

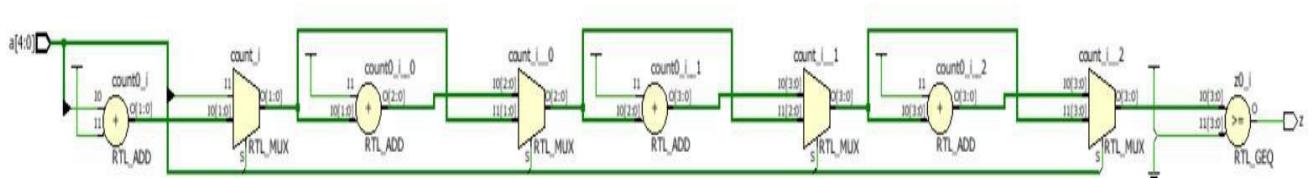
The screenshot shows a text editor window with three tabs: "inputmajckt.v", "inputmajckt_tb.v", and "Untitled 3". The "inputmajckt_tb.v" tab is active, displaying Verilog code for a testbench. The code defines a module "majority_ckt_tb" with a reg "x" of width 5:1 and an output "z". It instantiates the "majority_ckt" module and uses a for loop to generate test cases for all possible 5-bit input combinations, printing them to the console using \$display.

```
1 module majority_ckt_tb();
2   reg [5:1] x;
3   wire z;
4   integer i;
5
6   majority_ckt dut(x,z);
7
8   initial
9   begin
10  for (i=0; i<32; i=i+1)
11    begin
12      {x[5], x[4], x[3], x[2], x[1]} = i;
13      #10 $display ("%1x%2x%3x%4x%5 = %b, Output = %b",
14      {x[1], x[2], x[3], x[4], x[5]}, z);
15    end
16  end
17 endmodule
```

Output



RTL SCHEMATIC :-



SYNTHESIS REPORT :-

```

-----
Start Writing Synthesis Report

Report BlackBoxes:
+---+ +---+
| |BlackBox name |Instances |
+---+ +---+
+---+ +---+
+---+ +---+
Report Cell Usage:
+---+ +---+
| |Cell |Count |
+---+ +---+
|1 |LUT5 | 1|
|2 |IBUF | 5|
|3 |OBUF | 1|
+---+ +---+
Report Instance Areas:
+---+ +---+ +---+
| |Instance |Module |Cells |
+---+ +---+ +---+
|1 |top | 7|
+---+ +---+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:05 ; elapsed = 00:00:07 . Memory (MB): peak = 410.039 ; gain = 250.438
-----
```

POWER REPORT :-

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: **0.518 W**

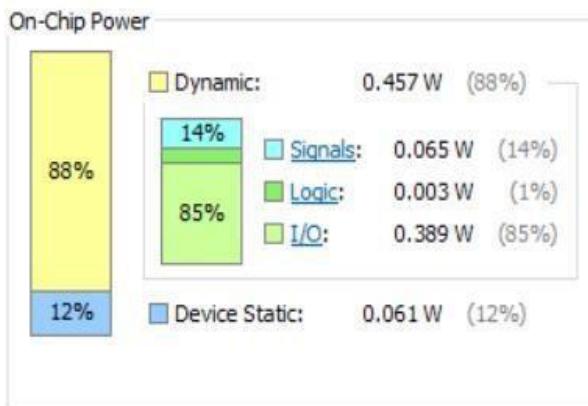
Junction Temperature: **27.5 °C**

Thermal Margin: 72.5 °C (15.1 W)

Effective ΔJA: 4.8 °C/W

Power supplied to off-chip devices: 0 W

Confidence level: [Low](#)



5) Parity Generator

Design

The screenshot shows a text editor window with three tabs at the top: "paritygen.v", "paritygen_tb.v", and "Untitled 1". The "paritygen.v" tab is active, displaying the following Verilog code:

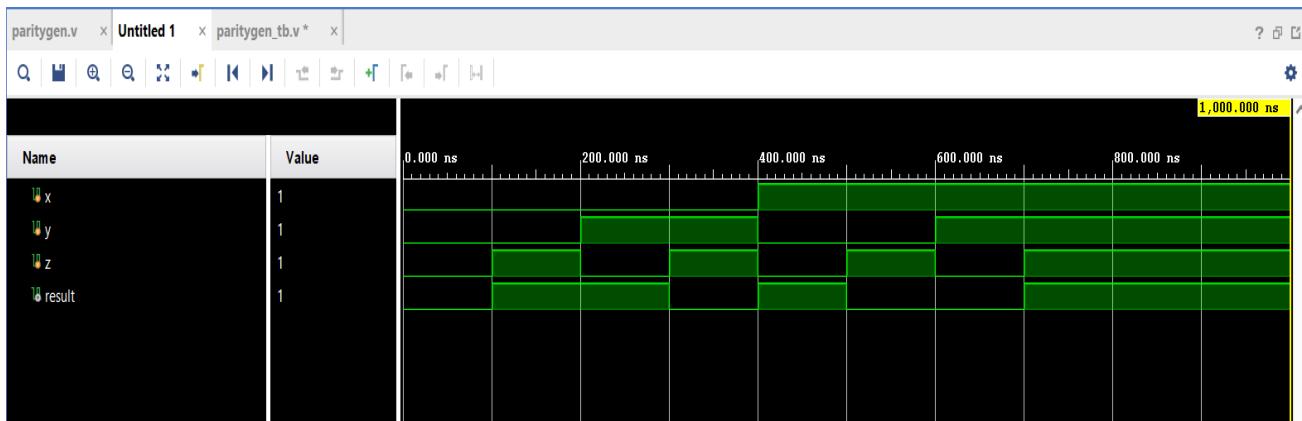
```
1 `timescale 1ns / 1ps
2
3 module par_gen(x,y,z,result);
4   input x,y,z;
5   output result;
6   xor (result,x,y,z);
7 endmodule
```

Testbench

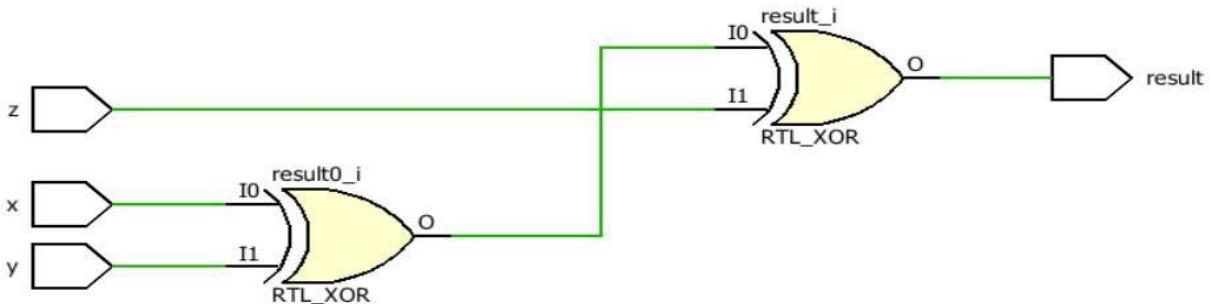
The screenshot shows a text editor window with three tabs at the top: "paritygen.v", "Untitled 1", and "paritygen_tb.v *". The "paritygen_tb.v" tab is active, displaying the following Verilog testbench code:

```
1 `timescale 1ns / 1ps
2
3 module par_gen_tb;
4   reg x;
5   reg y;
6   reg z;
7   wire result;
8   par_gen uut (
9     .x(x),
10    .y(y),
11    .z(z),
12    .result(result)
13  );
14 initial begin
15   x = 0;
16   y = 0;
17   z = 0;
18   #100;
19   x = 0;
20   y = 0;
21   z = 1;
22
23   #100;
24   x = 0;
25   y = 1;
26   z = 0;
27
28   #100;
29   x = 0;
30   y = 1;
31   z = 1;
32 end
```

Output



RTL SCHEMATIC :-



SYNTHESIS REPORT :-

```

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

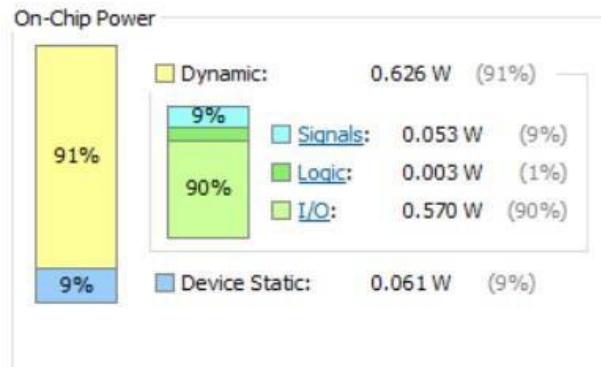
Report BlackBoxes:
+++
| |BlackBox name |Instances |
+++
+++
+-----+
Report Cell Usage:
+-----+-----+
| |Cell |Count |
+-----+-----+
|1 |LUT3 | 1|
|2 |IBUF | 3|
|3 |OBUF | 1|
+-----+-----+

Report Instance Areas:
+-----+-----+-----+
| |Instance |Module |Cells |
+-----+-----+-----+
|1 |top | | 5|
+-----+-----+-----+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:05 ; elapsed = 00:00:06 . Memory (MB): peak = 409.969 ; gain = 250.379
-----
```

POWER REPORT :-

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: 0.687 W
Junction Temperature: 28.3 °C
Thermal Margin: 71.7 °C (14.9 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



6) Binary to one hot encoder

Design

The screenshot shows a text editor window with three tabs at the top: "onehot.v", "onehot_tb.v", and "Untitled 2". The "onehot.v" tab is active. The code is as follows:

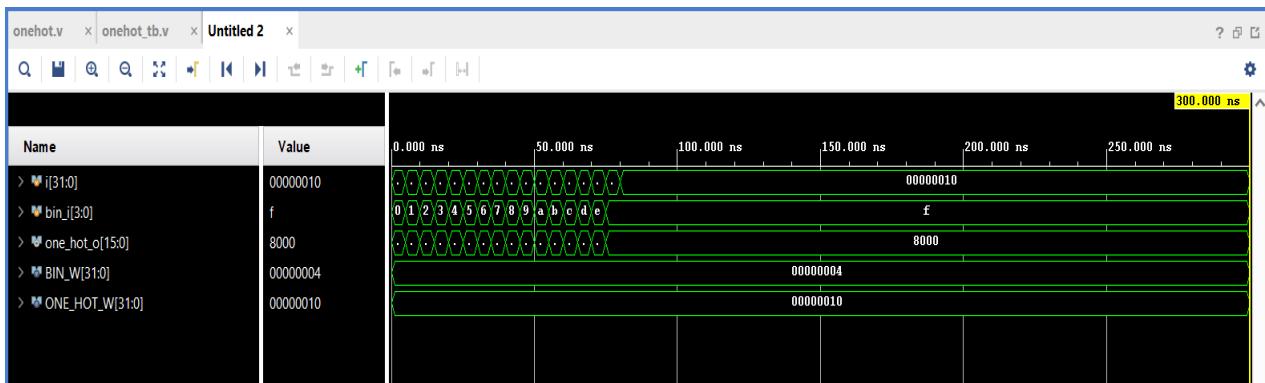
```
1 module one_hot(bin_i,one_hot_o);
2   parameter BIN_W=4;
3   parameter ONE_HOT_W=16;
4   input [BIN_W-1:0] bin_i;
5   output [ONE_HOT_W-1:0] one_hot_o;
6
7   assign one_hot_o = 1'b1<<bin_i;
8
9 endmodule
```

Testbench

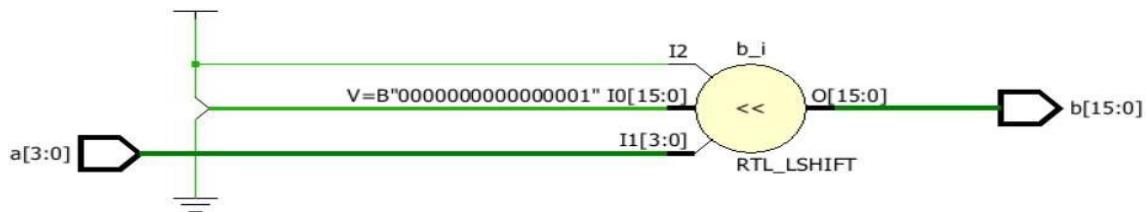
The screenshot shows a text editor window with three tabs at the top: "onehot.v", "onehot_tb.v", and "Untitled 2". The "onehot_tb.v" tab is active. The code is as follows:

```
1 module one_hot_tb();
2
3   localparam BIN_W = 4;
4   localparam ONE_HOT_W = 16;
5   integer i;
6
7   reg [BIN_W-1:0] bin_i;
8   wire [ONE_HOT_W-1:0] one_hot_o;
9
10  one_hot dut(bin_i,one_hot_o);
11
12  initial
13  begin
14    for( i=0; i<16; i=i+1)
15      begin
16        bin_i = i;
17        #5;
18      end
19
20
21  end
22
23  initial
24  begin
25    $monitor("Binary Value: %b | One hot encoded value : %b", bin_i,one_hot_o);
26
27    #300 $finish;
28  end
29
30 endmodule
```

Output



RTL SCHEMATIC: -



SYNTHESIS REPORT :-

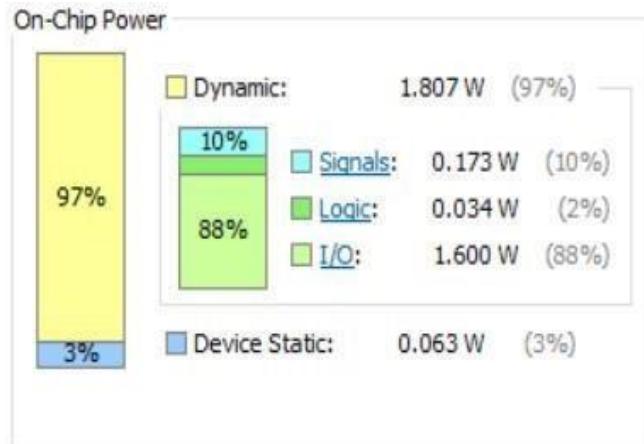
```

Start Writing Synthesis Report
-----
Report BlackBoxes:
+---+---+
| |BlackBox name |Instances |
+---+---+
+---+---+
Report Cell Usage:
+---+---+---+
| |Cell |Count |
+---+---+---+
|1 |LUT4 | 16|
|2 |IBUF | 4|
|3 |OBUF | 16|
+---+---+---+
Report Instance Areas:
+---+---+---+
| |Instance |Module |Cells |
+---+---+---+
|1 |top | | 36|
+---+---+---+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:05 ; elapsed = 00:00:06 . Memory (MB): peak = 410.676 ; gain = 251.180
-----
```

POWER REPORT: -

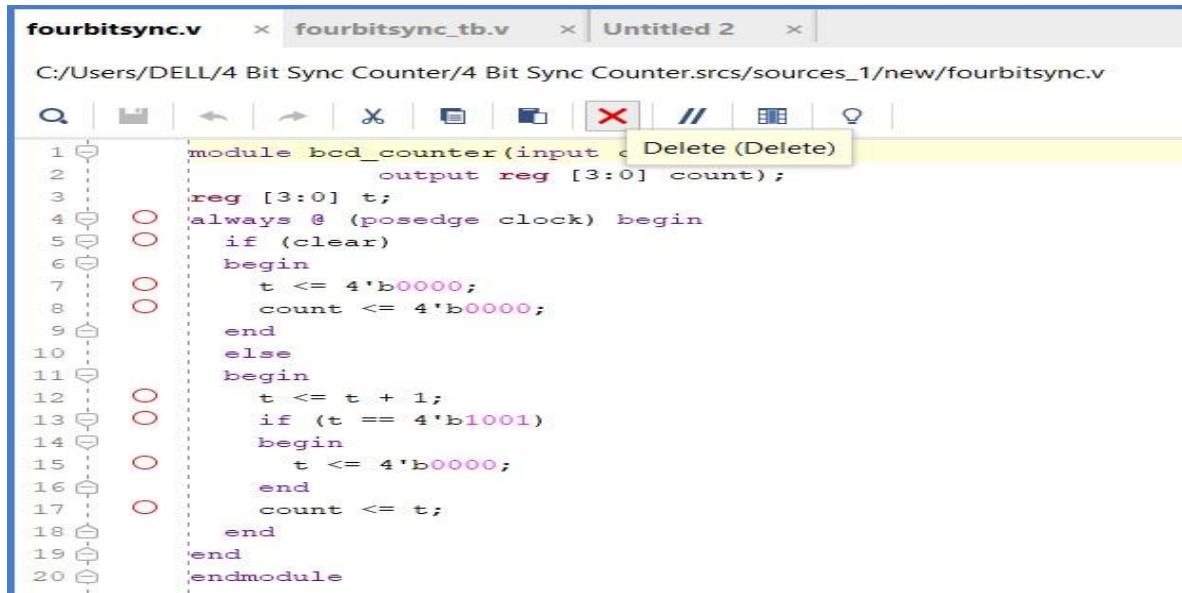
Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: 1.87 W
Junction Temperature: 33.9 °C
Thermal Margin: 66.1 °C (13.7 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



7) 4-Bit Synchronous BCD Counter

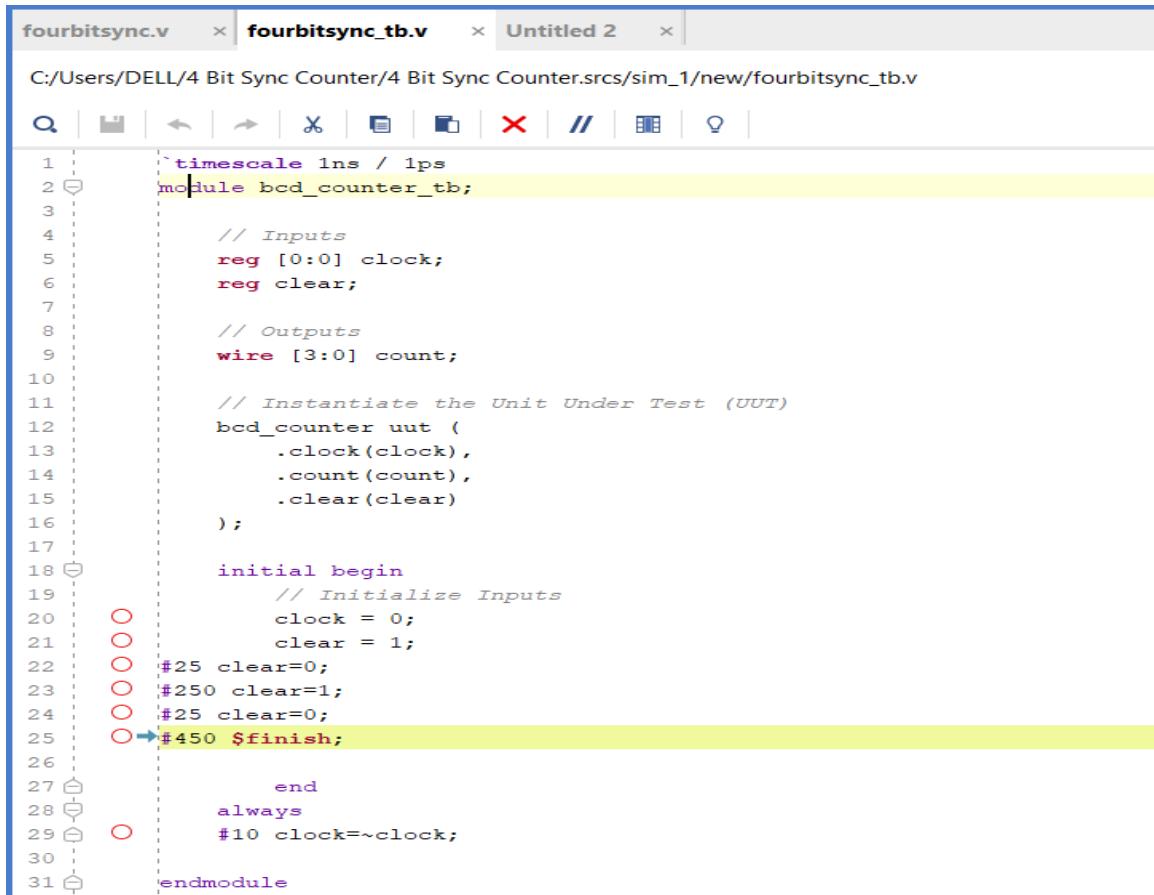
Design



```
fourbitsync.v  × fourbitsync_tb.v  × Untitled 2  ×
C:/Users/DELL/4 Bit Sync Counter/4 Bit Sync Counter.srcts/sources_1/new/fourbitsync.v

1 module bcd_counter(input  < Delete (Delete)
2                      output reg [3:0] count);
3   reg [3:0] t;
4   always @ (posedge clock) begin
5     if (clear)
6       begin
7         t <= 4'b0000;
8         count <= 4'b0000;
9       end
10    else
11      begin
12        t <= t + 1;
13        if (t == 4'b1001)
14          begin
15            t <= 4'b0000;
16          end
17        count <= t;
18      end
19   end
20 endmodule
```

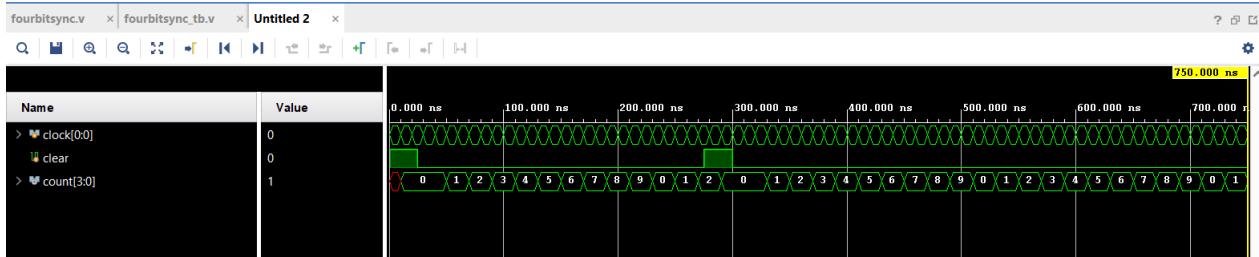
Testbench



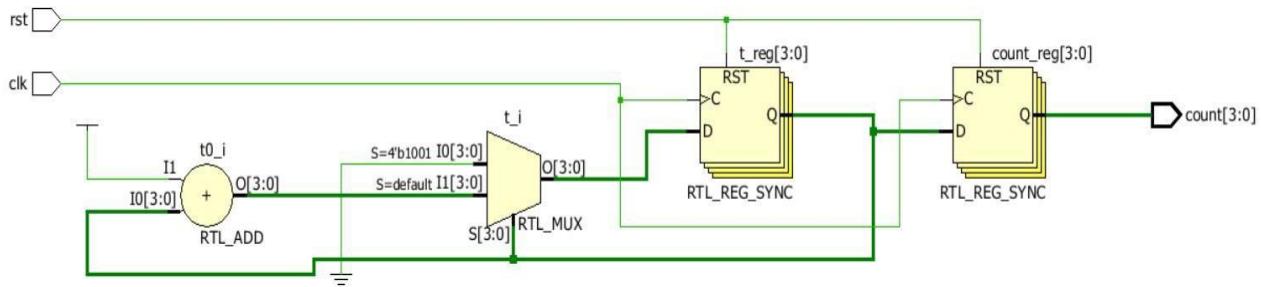
```
fourbitsync.v  × fourbitsync_tb.v  × Untitled 2  ×
C:/Users/DELL/4 Bit Sync Counter/4 Bit Sync Counter.srcts/sim_1/new/fourbitsync_tb.v

1 `timescale 1ns / 1ps
2 module bcd_counter_tb;
3
4   // Inputs
5   reg [0:0] clock;
6   reg clear;
7
8   // Outputs
9   wire [3:0] count;
10
11  // Instantiate the Unit Under Test (UUT)
12  bcd_counter uut (
13    .clock(clock),
14    .count(count),
15    .clear	clear)
16  ;
17
18 initial begin
19   // Initialize Inputs
20   clock = 0;
21   clear = 1;
22   #25 clear=0;
23   #250 clear=1;
24   #25 clear=0;
25   #450 $finish;
26
27   end
28
29   always
30     #10 clock=~clock;
31 endmodule
```

Output



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  |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 = 00:00:05 ; elapsed = 00:00:07 . Memory (MB): peak = 408.035 ; gain = 248.551
-----
```

POWER REPORT :-

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: 5.063 W

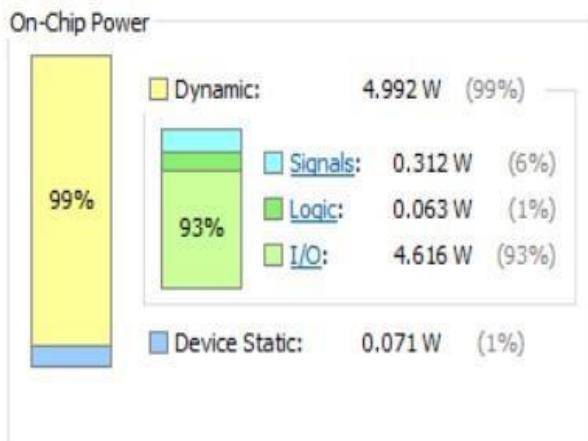
Junction Temperature: 49.2 °C

Thermal Margin: 50.8 °C (10.6 W)

Effective ΔJA: 4.8 °C/W

Power supplied to off-chip devices: 0 W

Confidence level: Low



8) 4-BIT CARRY LOOKAHEAD ADDER

Design

```
lookaheadadder.v  x  lookaheadadder_tb.v  x  Untitled 1  x
C:/Users/DELL/LookAheadAdder/LookAheadAdder.srcts/sources_1/new/lookaheadadder.v

Q | H | ← | → | X | D | F | X | // | B | ? |

1 | module cla_4bit (a, b, cin, sum, cout);
2 |   input [3:0] a, b; //define inputs and outputs
3 |   input cin;
4 |   output [3:0] sum;
5 |   output cout;
6 |   //design the logic for the generate functions
7 |   O and (g0, a[0], b[0]),
8 |     (g1, a[1], b[1]),
9 |     (g2, a[2], b[2]),
10 |    (g3, a[3], b[3]);
11 |   //design the logic for the propagate functions
12 |   O xor (p0, a[0], b[0]),
13 |     (p1, a[1], b[1]),
14 |     (p2, a[2], b[2]),
15 |     (p3, a[3], b[3]);
16 |   //design the logic for the sum equations
17 |   O xor (sum[0], p0, cin),
18 |     (sum[1], p1, c0),
19 |     (sum[2], p2, c1),
20 |     (sum[3], p3, c2); //design the logic for the carry equations
21 |   //using the continuous assign statement
22 |   O assign c0 = g0 | (p0 & cin),
23 |     c1 = g1 | (p1 & g0) | (p1 & p0 & cin),
24 |     c2 = g2 | (p2 & g1) | (p2 & p1 & g0)
25 |     | (p2 & p1 & p0 & cin),
26 |     O c3 = g3 | (p3 & g2) | (p3 & p2 & g1)
27 |       | (p3 & p2 & p1 & g0)
28 |       | (p3 & p2 & p1 & p0 & cin);
29 |   //design the logic for cout using assign
30 |   O assign cout = c3;
31 | endmodule
```

Testbench

lookaheadadder.v lookaheadadder_tb.v * Untitled 1

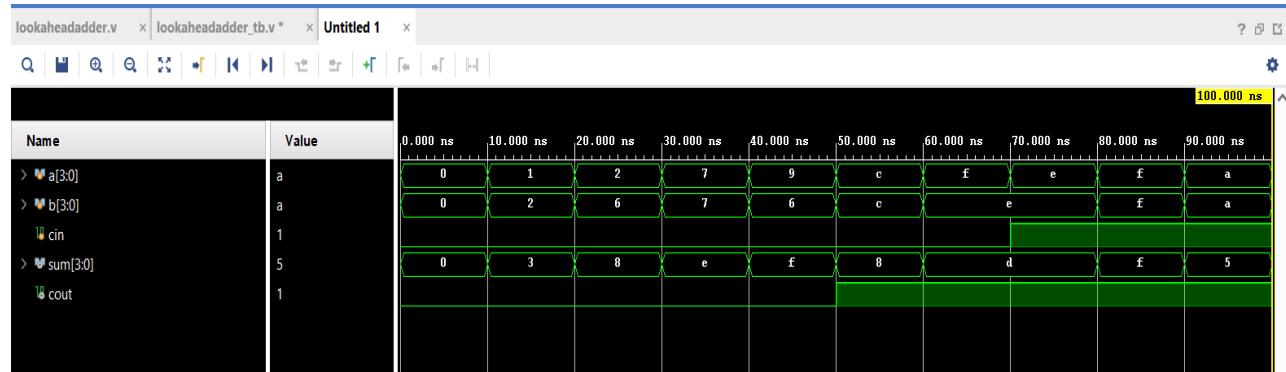
C:/Users/DELL/LookAheadAdder/LookAheadAdder.srccs/sim_1/new/lookaheadadder_tb.v

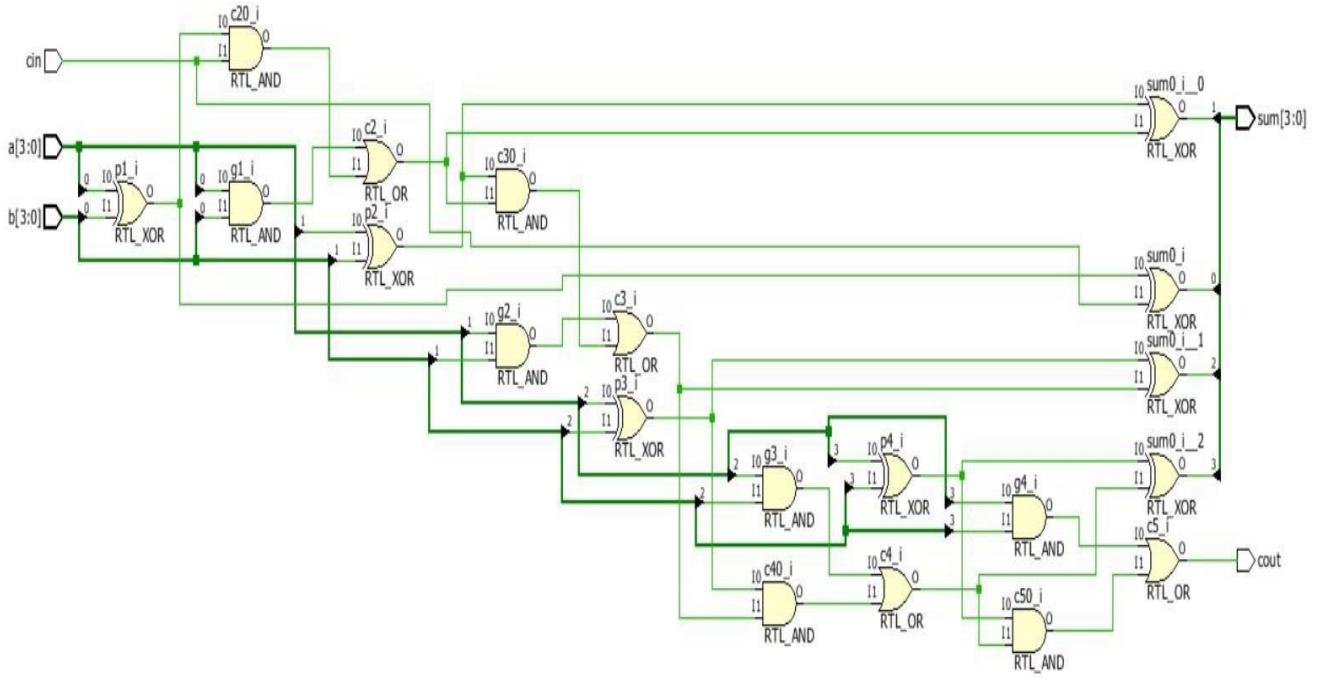
```

1 module cla_4bit_tb;
2 reg [3:0] a, b;
3 reg cin;
4 wire [3:0] sum;
5 wire cout;
6
7 cla_4bit dut(a, b, cin, sum, cout);
8
9 initial
10 begin
11 #0 a = 4'b0000; b = 4'b0000; cin = 1'b0;
12 #10 a = 4'b0001; b = 4'b0010; cin = 1'b0;
13 #10 a = 4'b0010; b = 4'b0110; cin = 1'b0;
14 #10 a = 4'b0111; b = 4'b0111; cin = 1'b0;
15 #10 a = 4'b1001; b = 4'b0110; cin = 1'b0;
16 #10 a = 4'b1100; b = 4'b1100; cin = 1'b0;
17 #10 a = 4'b1111; b = 4'b1110; cin = 1'b0;
18 #10 a = 4'b1110; b = 4'b1110; cin = 1'b1;
19 #10 a = 4'b1111; b = 4'b1111; cin = 1'b1;
20 #10 a = 4'b1010; b = 4'b1010; cin = 1'b1;
21 #10 a = 4'b1000; b = 4'b1000; cin = 1'b0;
22 #10 a = 4'b1101; b = 4'b1000; cin = 1'b1;
23 end
24 initial
25 begin
26 //display variables
27 $monitor("A=%b | B=%b | Cin=%b | Sum=%b | Carry=%b",a,b,cin,sum,cout);
28
29 #100 $finish;
30 end
31 endmodule

```

Output





SYNTHESIS REPORT :-

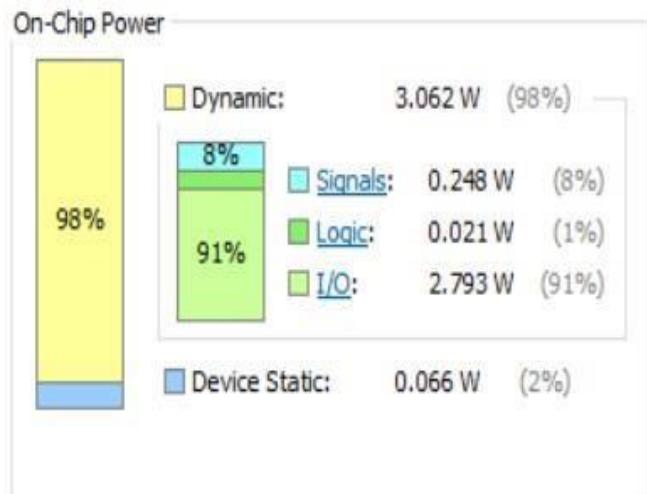
```

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 = 00:00:05 ; elapsed = 00:00:06 . Memory (MB): peak = 420.418 ; gain = 260.879
-----
```

POWER REPORT: -

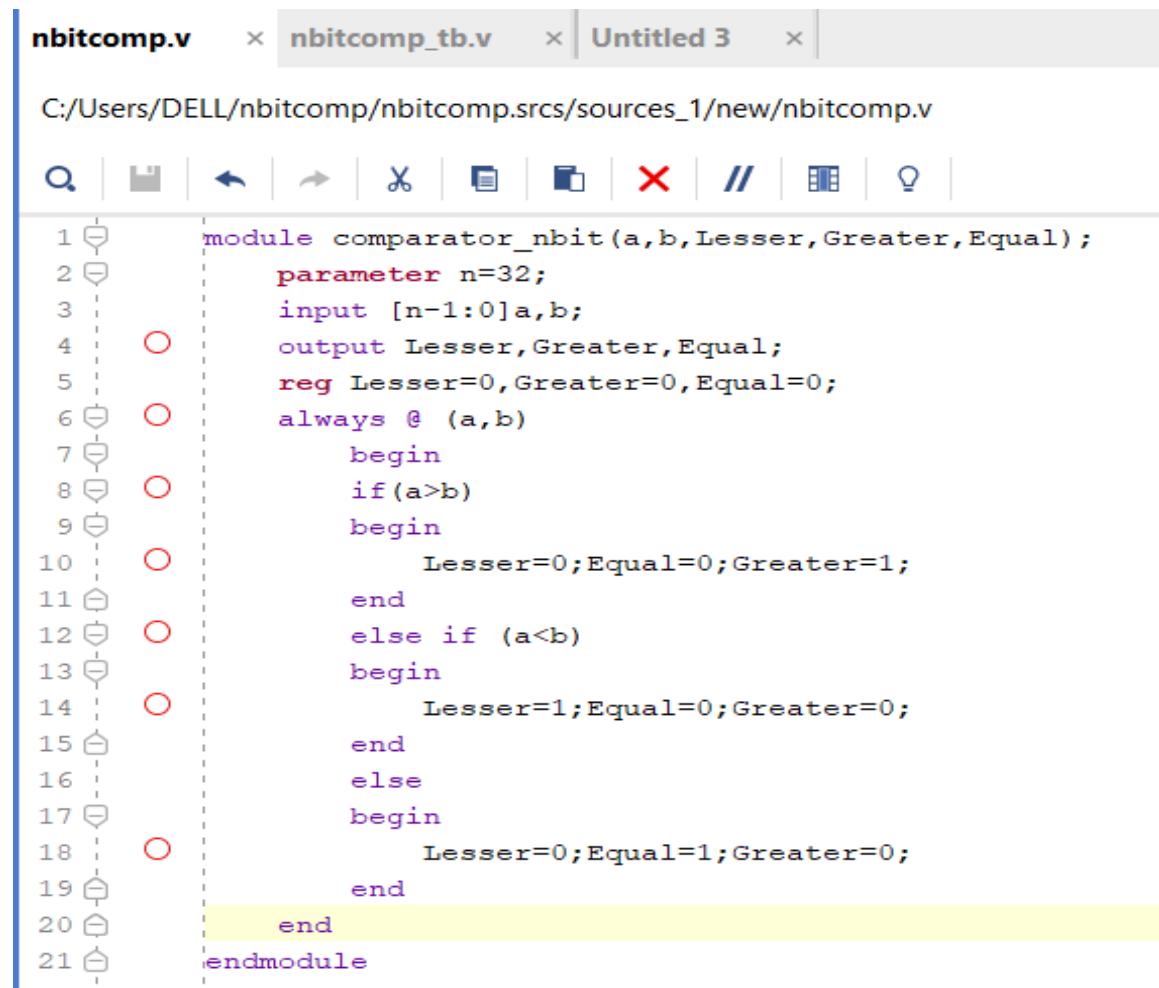
Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: 3.128 W
Junction Temperature: 39.9 °C
Thermal Margin: 60.1 °C (12.5 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



9) N-BIT COMPARATOR

Design



```
1  module comparator_nbit(a,b,Lesser,Greater,Equal);
2  parameter n=32;
3  input [n-1:0]a,b;
4  output Lesser,Greater,Equal;
5  reg Lesser=0,Greater=0,Equal=0;
6  always @ (a,b)
7    begin
8      if(a>b)
9        begin
10         Lesser=0;Equal=0;Greater=1;
11       end
12     else if (a<b)
13       begin
14         Lesser=1;Equal=0;Greater=0;
15       end
16     else
17       begin
18         Lesser=0;Equal=1;Greater=0;
19       end
20     end
21   endmodule
```

Testbench

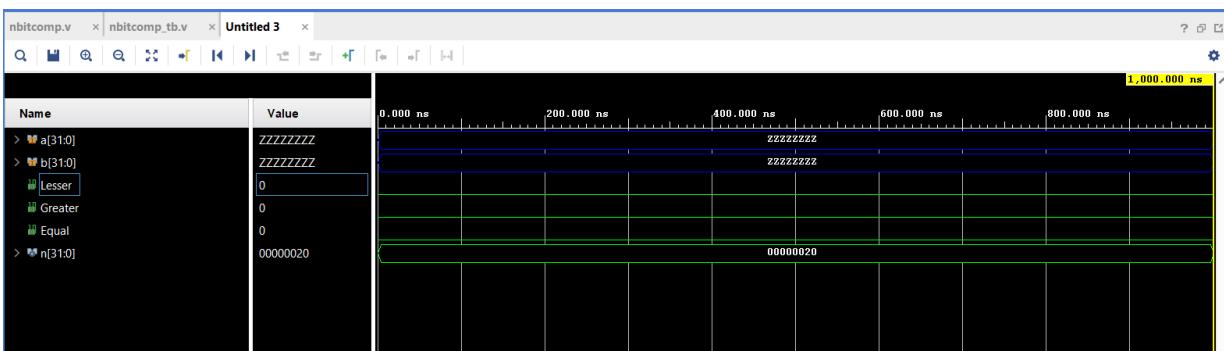
nbitcomp.v x nbitcomp_tb.v x Untitled 3 x

C:/Users/DELL/nbitcomp/nbitcomp.srccs/sim_1/new/nbitcomp_tb.v

Q | H | ← | → | X | D | F | X | // | M | ? |

```
1 module comparator_nbit_tb;
2
3     reg [31:0] a;
4     reg [31:0] b;
5
6     wire Lesser;
7     wire Greater;
8     wire Equal;
9
10    comparator_nbit uut (
11        .a(a),
12        .b(b),
13        .Lesser(Lesser),
14        .Greater(Greater),
15        .Equal(Equal)
16    );
17    initial begin
18        $monitor("      A=%d      B=%d      Lesser=%b      Greater=%b      Equal=%b      ",a,b,Lesser,Greater,Equal);
19    end
20
21    initial begin
22        a = 2;b = 2;
23        #100 a=22;b=444;
24        #100 a=444;b=555;
25        #100 a=777;b=111;
26        #100 a=8888;b=8888;
27        #500 $finish;
28    end
29
30 endmodule
31
```

Output



10) SERIAL IN SERIAL OUT SHIFT REGISTER]

Design

The screenshot shows a text editor window with the following details:

- Tab bar: `siso_reg.v*`, `siso_reg_tb.v`, `Untitled 1`.
- File path: C:/Users/DELL/SISO REG/SISO REG.srcts/sources_1/new/siso_reg.v
- Toolbar icons: magnifying glass, save, back, forward, cut, copy, paste, delete, search, double slash, question mark.
- Code content:

```
1 module siso_reg(clk,clear,si,so);
2
3     input clk,si,clear;
4
5     output reg so;
6
7     reg [3:0] tmp;
8
9     always @(posedge clk)
10    begin
11        if (clear)
12            tmp <= 4'b0000;
13        else
14            tmp <= tmp << 1;
15            tmp[0] <= si;
16            so = tmp[3];
17
18    end
19
20
21
22
23
24 endmodule
```

Testbench

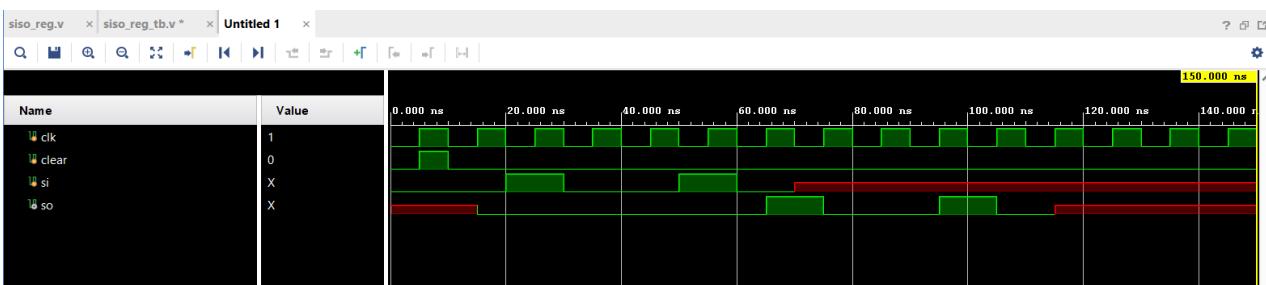
siso_reg.v x siso_reg_tb.v * x Untitled 1 x

C:/Users/DELL/SISO REG/SISO REG.srcts/sim_1/new/siso_reg_tb.v

Q | F | ← | → | X | E | D | X | // | I | ? |

```
1 module siso_reg_tb;
2   reg clk,clear,si;
3   wire so;
4   siso_reg uut (.clk(clk), .clear	clear),.si(si),.so(so));
5
6 initial begin
7
8   clk = 0;
9   clear = 0;
10  si = 0;
11
12  #5 clear=1'b1;
13  #5 clear=1'b0;
14  #10 si=1'b1;
15  #10 si=1'b0;
16  #10 si=1'b0;
17  #10 si=1'b1;
18  #10 si=1'b0;
19  #10 si=1'b0;
20
21 end
22
23 always #5 clk = ~clk;
24
25 initial #150 $stop;
26
27 endmodule
```

Output



11) SERIAL IN PARALLEL OUT SHIFT REGISTER

Design

The screenshot shows a text editor window with the title bar "sipo_reg.v * < sipo_reg_tb.v < Untitled 2". The file path is "C:/Users/DELL/SIPO REG/SIPO REG.srcs/sources_1/new/sipo_reg.v". The code is as follows:

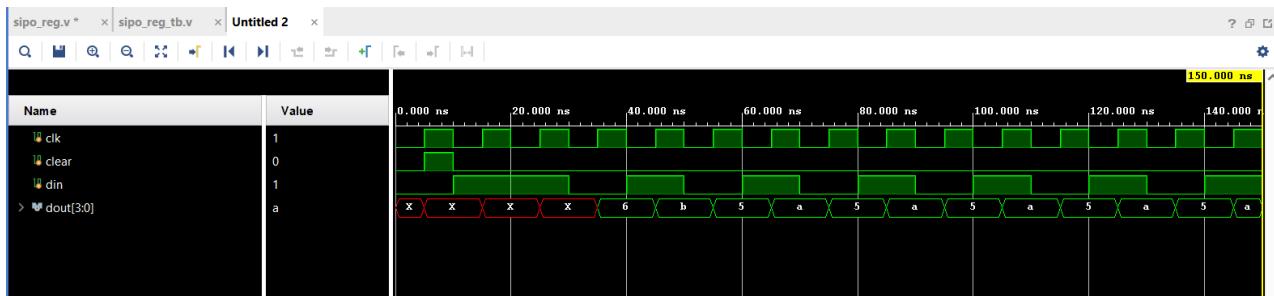
```
1 module sipo_reg(clk,clear,din,dout);
2     input clk,din,clear;
3     output reg [3:0] dout;
4
5     reg [3:0] tmp;
6
7     always @(posedge clk)
8     begin
9         if (clear)
10             dout = 4'b0000;
11         else
12             tmp = dout >>1 ;
13             dout = {din,tmp[2:0]};
14     end
15
16 endmodule
```

Testbench

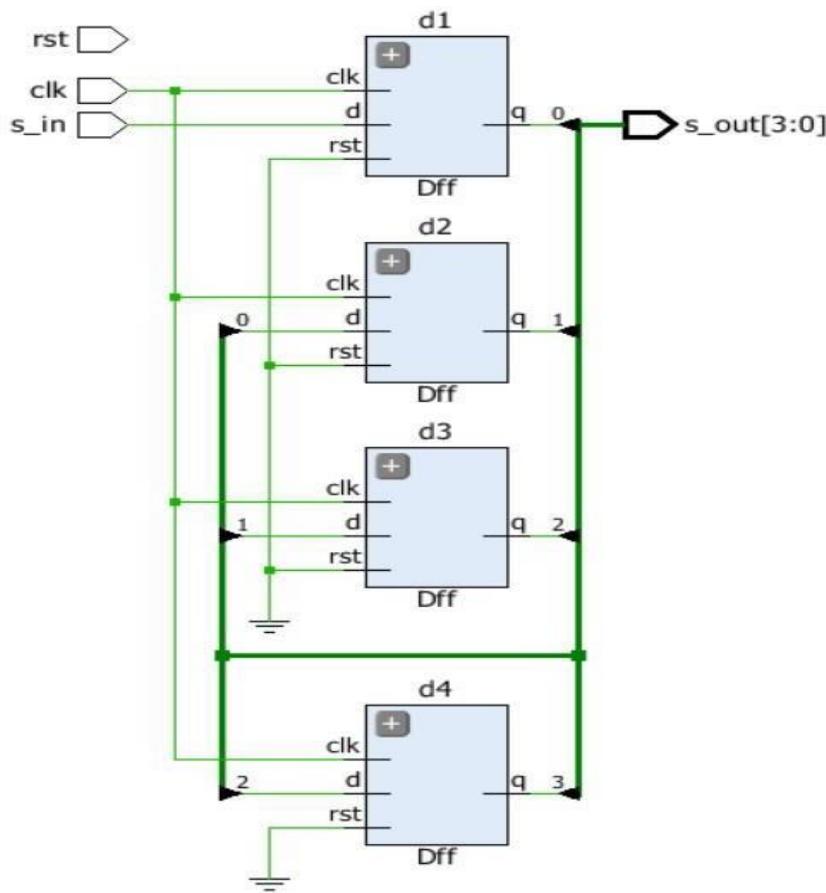
The screenshot shows a text editor window with the title bar "sipo_reg.v * < sipo_reg_tb.v < Untitled 2". The file path is "C:/Users/DELL/SIPO REG/SIPO REG.srcs/sim_1/new/sipo_reg_tb.v". The code is as follows:

```
1 module sipo_reg_tb;
2
3     reg clk,clear,din;
4     wire [3:0] dout;
5     sipo_reg uut (.clk(clk), .clear	clear),.din(din),.dout(dout));
6     initial begin
7         clk = 0;
8         clear = 0;
9         din = 0;
10        #5 clear=1'b1;
11        #5 clear=1'b0;
12        #10 din=1'b1;
13    end
14    always #5 clk = ~clk;
15    always #10 din=~din;
16    initial
17        begin
18            $monitor(" Data Input = %b | Data output =%b",din,dout);
19            #150 $finish;
20        end
21    endmodule
```

Output



RTL SCHEMATIC :-



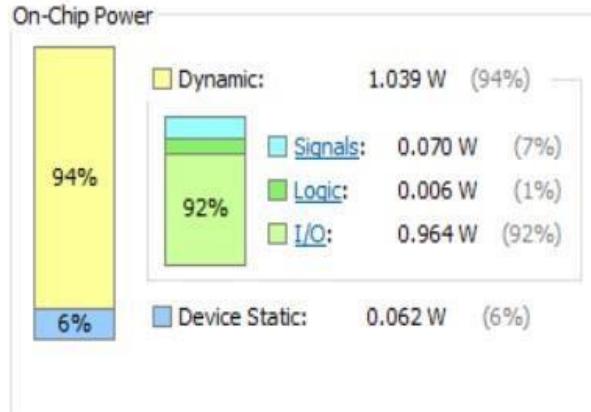
SYNTHESIS REPORT :-

```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
+---+---+  
| |BlackBox name |Instances |  
+---+---+  
+---+---+  
  
Report Cell Usage:  
+---+---+  
| |Cell |Count |  
+---+---+  
|1|BUFG | 1|  
|2|FDRE | 4|  
|3|IBUF | 2|  
|4|OBUF | 4|  
+---+---+  
  
Report Instance Areas:  
+---+---+---+  
| |Instance |Module |Cells |  
+---+---+---+  
|1|top | | 11|  
|2| d1 |Dff | 1|  
|3| d2 |Dff_0 | 1|  
|4| d3 |Dff_1 | 1|  
|5| d4 |Dff_2 | 1|  
+---+---+  
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:05 ; elapsed = 00:00:07 . Memory (MB): peak = 402.305 ; gain = 242.840  
-----
```

POWER REPORT :-

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: 1.101 W
Junction Temperature: 30.3 °C
Thermal Margin: 69.7 °C (14.5 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



12) PARALLEL IN PARALLEL OUT REGISTER :-

Design

The screenshot shows a Verilog code editor window titled "pipo_reg.v". The code defines a module "pipo_reg" with three ports: clk, clear, and d, and one output port q. It contains an always block that updates q based on the value of d when clear is low.

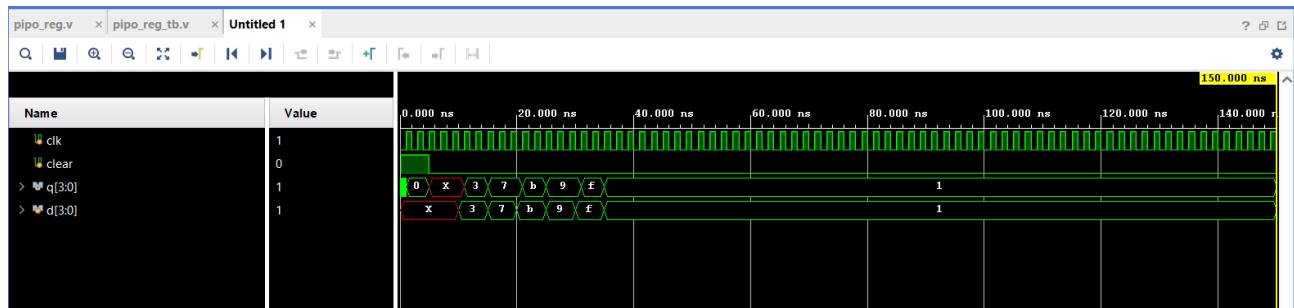
```
1 module pipo_reg(clk,clear,d,q);
2     input clk,clear;
3     output reg [3:0] q;
4     input [3:0] d;
5
6
7     always @(posedge clk)
8         begin
9             if (clear)
10                 q = 4'b0000;
11             else
12                 q = d;
13         end
14
15     endmodule
```

Testbench

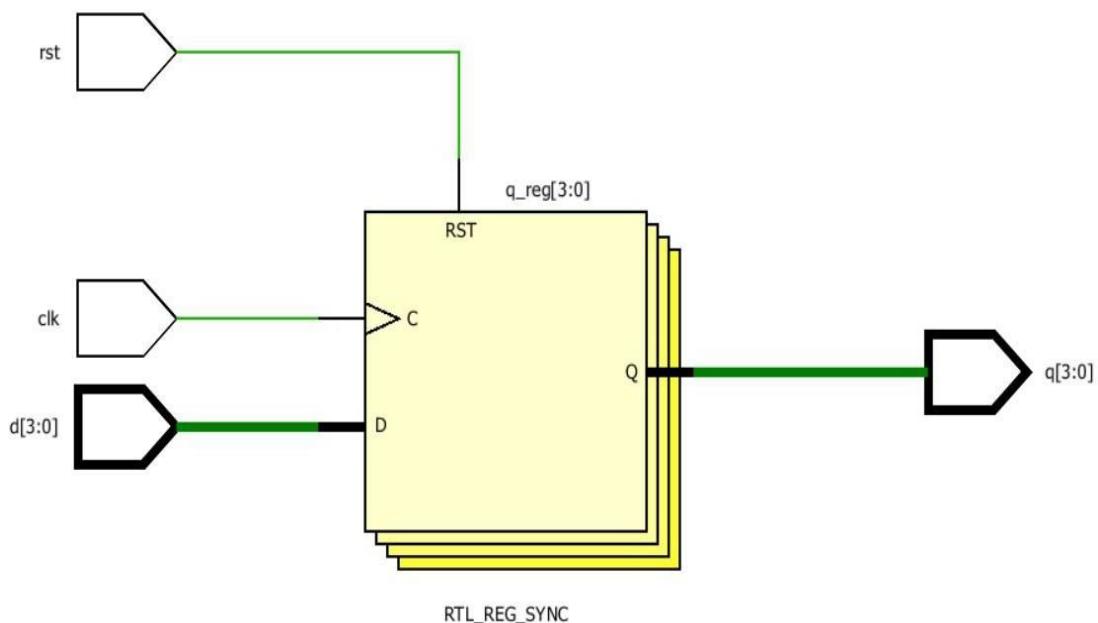
The screenshot shows a Verilog code editor window titled "pipo_reg_tb.v". The code defines a testbench module "pipo_reg_tb" with three ports: clk, clear, and d, and one output port q. It includes an initial block that generates various data patterns for d and monitors the output q. A \$stop statement is present at the end of the initial block.

```
1 module pipo_reg_tb;
2
3     reg clk,clear;
4     wire [3:0] q;
5     reg [3:0] d;
6
7     pipo_reg uut (.clk(clk), .clear	clear), .d(d), .q(q));
8
9     initial begin
10
11         clk = 0;
12         clear = 1;
13
14         #5 clear=1'b0;
15         #5 d=4'b0011;
16         #5 d=4'b0111;
17         #5 d=4'b1011;
18         #5 d=4'b1001;
19         #5 d=4'b1111;
20         #5 d=4'b0001;
21
22     end
23
24     always #1 clk = ~clk;
25
26     initial
27         begin
28             $monitor(" Data In = %b | Data out=%b",d,q);
29             #150 $stop;
30         end
31
32     endmodule
```

Output



RTL SCHEMATIC :-



SYNTHESIS REPORT :-

Start Writing Synthesis Report

Report BlackBoxes:

BlackBox name	Instances

Report Cell Usage:

Cell	Count
BUF	1
FDRE	4
IBUF	6
OBUF	4

Report Instance Areas:

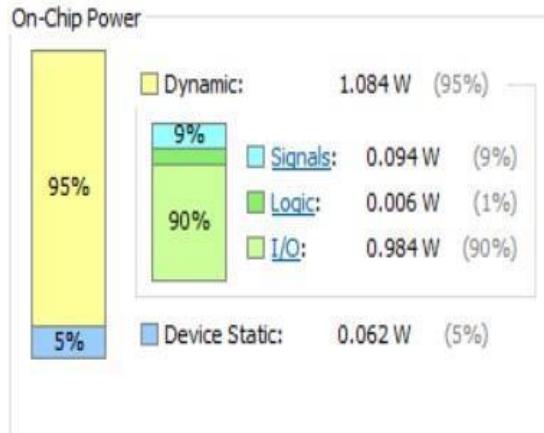
Instance	Module	Cells
top		15

Finished Writing Synthesis Report : Time (s): cpu = 00:00:06 ; elapsed = 00:00:06 . Memory (MB): peak = 393.809 ; gain = 234.270

POWER REPORT :-

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: 1.146 W
Junction Temperature: 30.5 °C
Thermal Margin: 69.5 °C (14.5 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



13) PARALLEL IN SERIAL OUT REGISTER

Design

Schematic x pipo_shift.v x

C:/Users/DELL/PIPO SHIFT/PIPO SHIFT.srcts/sources_1/new/pipo_shift.v

Q | | | | | | | |

```
1 module s1(a, b, sl ,q);
2     input a,b,sl;
3     output q;
4     assign q=(~sl&b) | (sl&a);
5 endmodule
6 module dff(d,clk,q);
7     input d,clk;
8     output q;
9     reg q=0;
10    always @ (posedge clk)
11    begin
12        q<=d;
13    end
14 endmodule|
15
16 module pipo_shift (d, clk, sl, q);
17     input [3:0]d;
18     input clk,sl;
19     output q;
20     wire q1,q2,q3,d1,d2,d3;
21     dff a(d[3],clk,q1);
22     s1 a1(q1,d[2],sl,d1);
23     dff b(d1,clk,q2);
24     s1 b1(q2,d[1],sl,d2);
25     dff c(d2,clk,q3);
26     s1 c1(q3,d[0],sl,d3);
27     dff dd(d3,clk,q);
28 endmodule
```

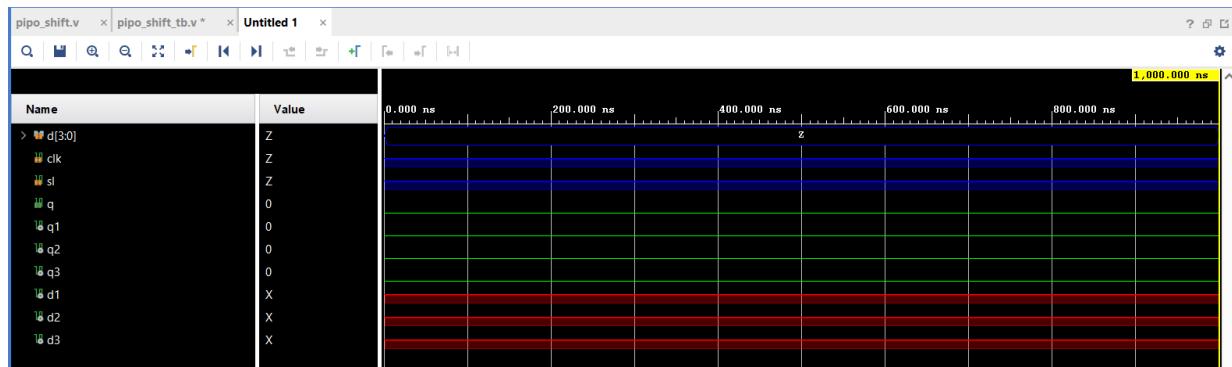
Testbench

C:/Users/DELL/PIPO SHIFT/PIPO SHIFT.srcs/sim_1/new/pipo_shift_tb.v

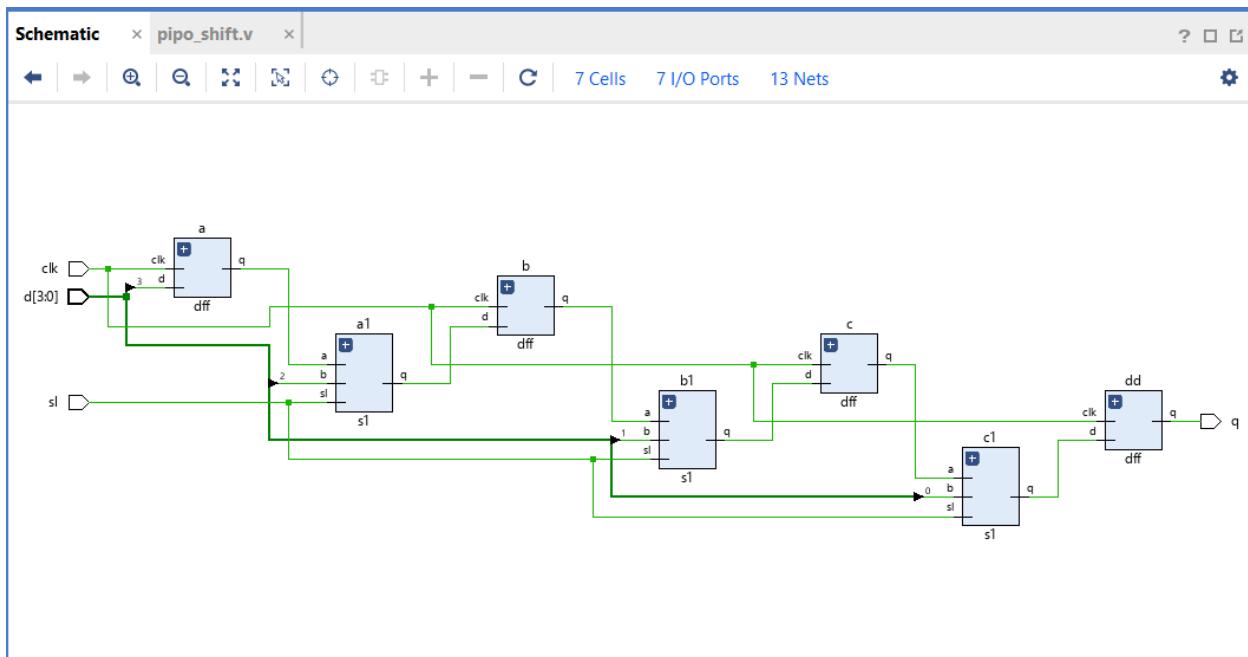
```
Q | F | < | > | X | D | C | X | // | E | ? |
```

```
1 module master_slave_tb;
2     reg s;
3     reg r;
4     reg clk;
5     wire qn;
6     wire qn_bar;
7     master_slave uut (
8         .s(s),
9         .r(r),
10        .clk(clk),
11        .qn(qn),
12        .qn_bar(qn_bar)
13    );
14 initial begin
15     clk=0;
16     #10 s = 0;r = 0;
17     #10 s = 0;r = 1;
18     #10 s = 1;r = 0;
19     #10 s = 1;r = 1;
20     #10 s = 0;r = 0;
21     #10 s = 0;r = 1;
22 end
23 always #5 clk=~clk;
24
25 initial
26 begin
27     $monitor("S=%b, R=%b, Q=%b, Q_bar=%b",s,r,qn,qn_bar);
28     #200 $finish;
29 end
30 endmodule
31
```

Output



RTL Schematic



SYNTHESIS REPORT :-

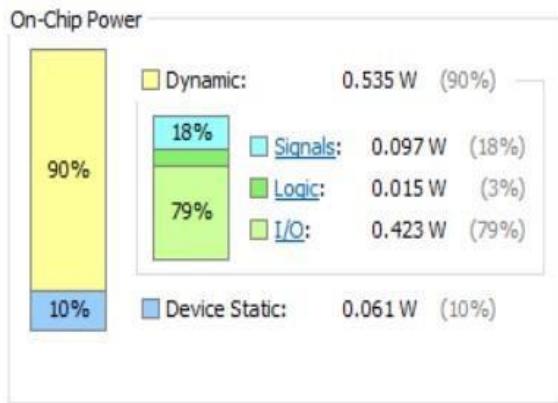
```
-----
Start Writing Synthesis Report
-----

Report BlackBoxes:
+-----+-----+
| |BlackBox name |Instances |
+-----+-----+
+-----+-----+
+-----+-----+
Report Cell Usage:
+-----+-----+
| |Cell |Count |
+-----+-----+
|1 |BUFG | 1|
|2 |LUT3 | 3|
|3 |FDRE | 4|
|4 |IBUF | 7|
|5 |OBUF | 1|
+-----+-----+
Report Instance Areas:
+-----+-----+-----+
| |Instance |Module |Cells |
+-----+-----+-----+
|1 |top | | 16|
|2 | d1 | Dff | 2|
|3 | d2 | Dff_0 | 2|
|4 | d3 | Dff_1 | 2|
|5 | d4 | Dff_2 | 1|
+-----+-----+-----+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:09 ; elapsed = 00:00:21 . Memory (MB): peak = 409.051 ; gain = 249.641
-----
```

POWER REPORT : -

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: 0.596 W
Junction Temperature: 27.8 °C
Thermal Margin: 72.2 °C (15.0 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



14) Bidirectional Shift Registers

Design

```
module bishift_reg #(parameter MSB=4) ( input d,           // Declare input for data
to the first flop in the shift register
    input clk,           // Declare input for clock to all flops in the
shift register
    input en,            // Declare input for enable to switch the shift
register on/off
    input dir,           // Declare input to shift in either left or right
direction
    input rstn,          // Declare input to reset the register to a
default value
    output reg [MSB-1:0] out); // Declare output to read out the
current value of all flops in this register
```

```
always @ (posedge clk)
if (!rstn)
  out <= 0;
else begin
  if (en)
    case (dir)
      0 : out <= {out[MSB-2:0], d};
      1 : out <= {d, out[MSB-1:1]};
    endcase
  else
    out <= out;
end
endmodule
```

Testbench

```
module bishift_reg_tb;
    parameter MSB = 4;      // [Optional] Declare a parameter to represent number of bits
in shift register

    reg data;            // Declare a variable to drive d-input of design
    reg clk;             // Declare a variable to drive clock to the design
    reg en;              // Declare a variable to drive enable to the design
    reg dir;             // Declare a variable to drive direction of shift register
    reg rstn;            // Declare a variable to drive reset to the design
    wire [MSB-1:0] out; // Declare a wire to capture output from the design

    // Instantiate design (16-bit shift register) by passing MSB and connect with TB signals
    bishift_reg #(MSB) sr0 (.d(data),
                           .clk(clk),
                           .en(en),
                           .dir(dir),
                           .rstn(rstn),
                           .out(out));

    // Generate clock time period = 20ns, freq => 50MHz
    always #10 clk = ~clk;

    // Initialize variables to default values at time 0
    initial begin
        clk <= 0;
        en <= 0;
        dir <= 0;
        rstn <= 0;
        data <= 'h1;
    end

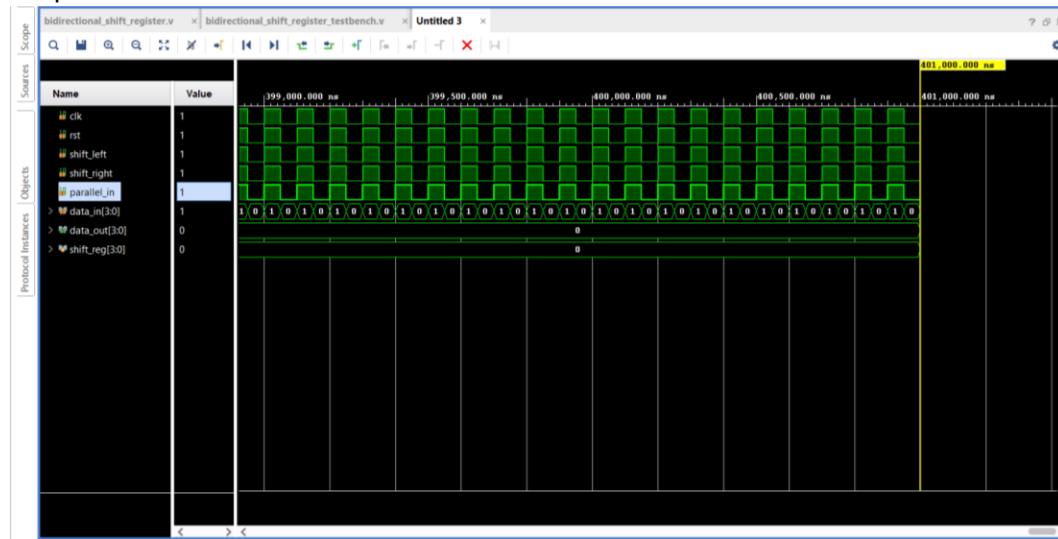
    // Drive main stimulus to the design to verify if this works
    initial begin

        // 1. Apply reset and deassert reset after some time
        rstn <= 0;
        #20 rstn <= 1;
        en <= 1;

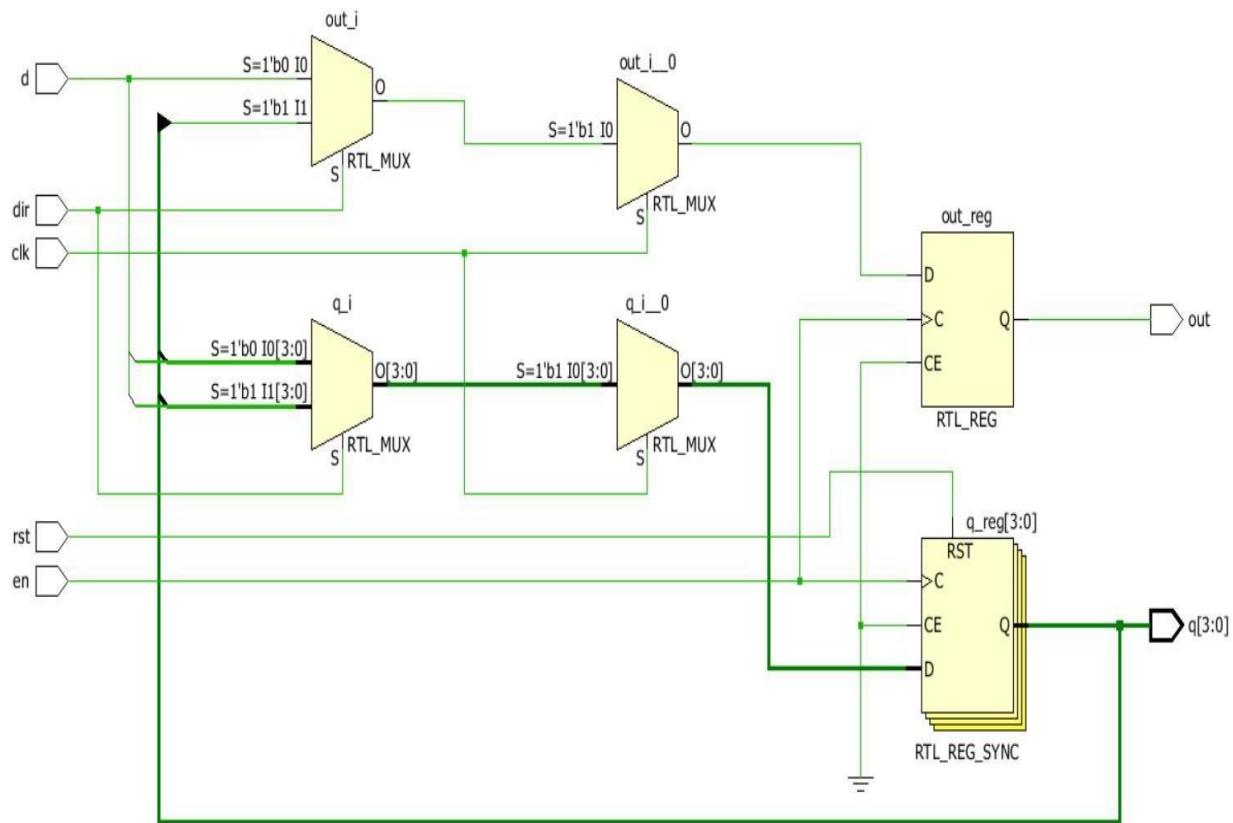
        // 2. For 7 clocks, drive alternate values to data pin
        repeat (7) @ (posedge clk)
            data <= ~data;
    end
```

```
// 4. Shift direction and drive alternate value to data pin for another 7 clocks  
#10 dir <= 1;  
repeat (7) @ (posedge clk)  
    data <= ~data;  
  
// 5. Drive nothing for next 7 clocks, allow shift register to simply shift based on dir  
repeat (7) @ (posedge clk);  
  
// 6. Finish the simulation  
#300 $finish;  
end  
  
// Monitor values of these variables and print them into the logfile for debug  
initial  
    $monitor ("rstn=%0b data=%b, en=%0b, dir=%0b, out=%b", rstn, data, en, dir, out);  
endmodule
```

Output:-



RTL SCHEMATIC :-



SYNTHESIS REPORT :-

```

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

Report BlackBoxes:
+++
| BlackBox name | Instances |
+++
+++
+-----+
Report Cell Usage:
+-----+-----+
| Cell | Count |
+-----+-----+
| IOBUF | 5 |
+-----+-----+

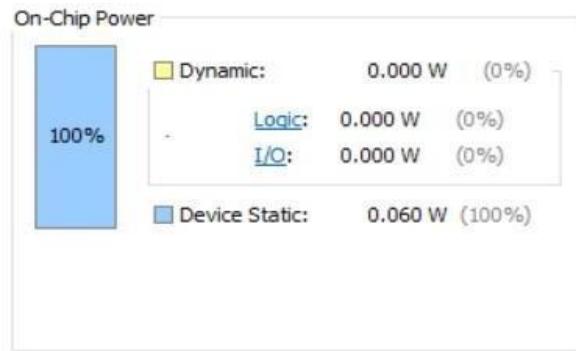
Report Instance Areas:
+-----+-----+-----+
| Instance | Module | Cells |
+-----+-----+-----+
| top |  | 5 |
+-----+-----+-----+
-----+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:09 ; elapsed = 00:00:11 . Memory (MB): peak = 403.184 ; gain = 243.828
-----+

```

POWER REPORT :-

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: 0.06 W
Junction Temperature: 25.3 °C
Thermal Margin: 74.7 °C (15.5 W)
Effective ΔT : 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



15) Psuedo Bit Random Sequence

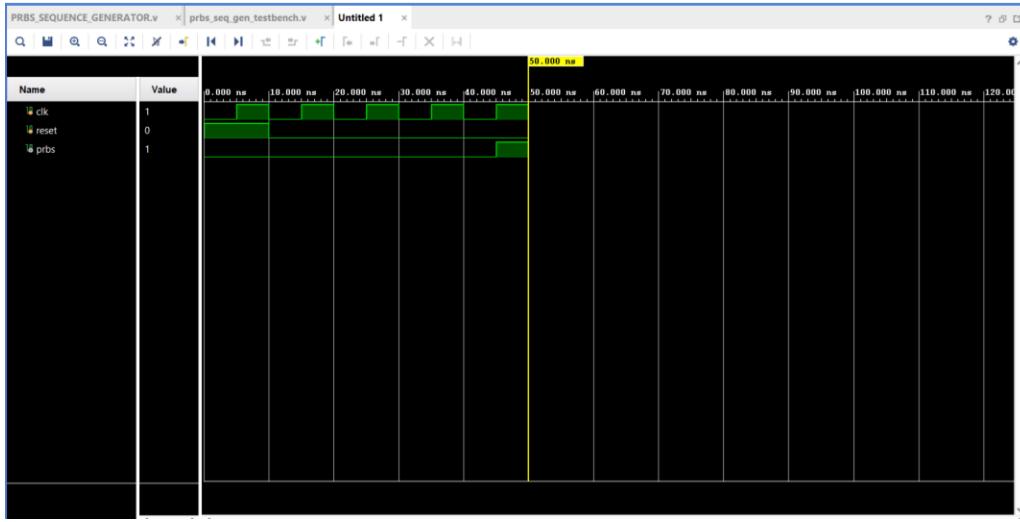
Design

```
module prbs_gen (rand, clk, reset);
    //Inputs
    input clk, reset;
    //output
    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
```

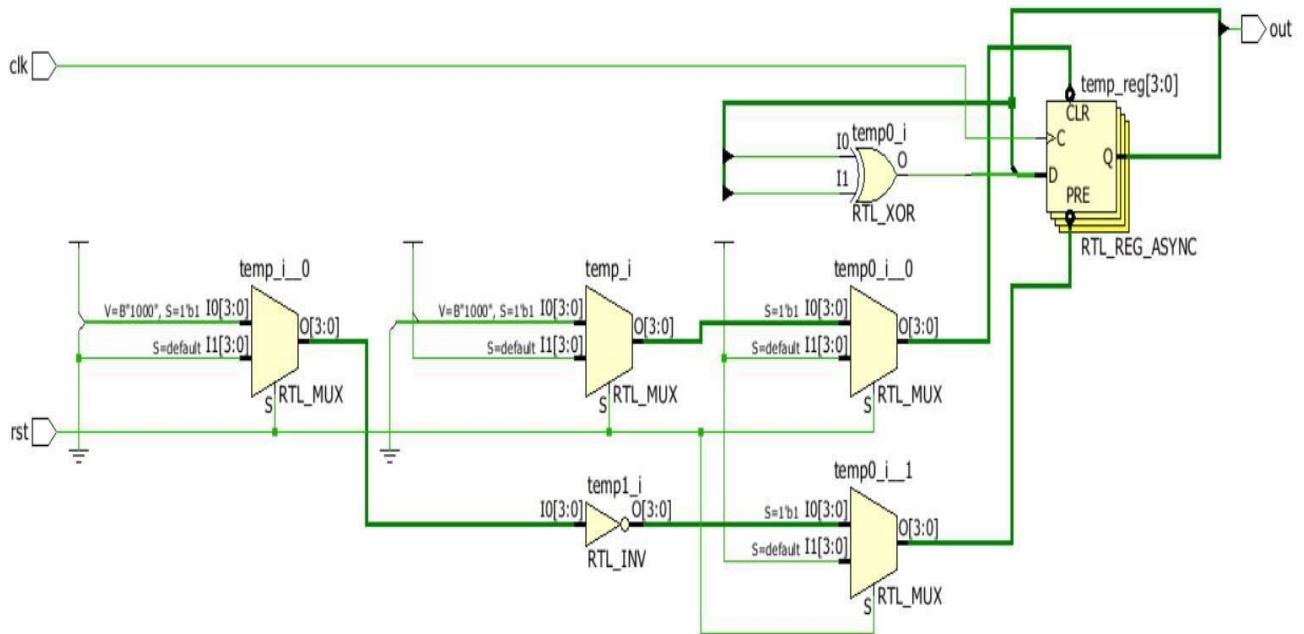
Testbench:-

```
module prbs_gen_tb();
reg clk, reset;
wire rand;
prbs_gen 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
initial
begin
$monitor(" Random bit : %b | Reset=%b | Clock=%b", rand,reset,clk);
#1000 $finish;
end
endmodule
```

Output:-



RTL SCHEMATIC :-



SYNTHESIS REPORT :-

Start Writing Synthesis Report

Report BlackBoxes:

+	-----	-----	+
	BlackBox name	Instances	
+	-----	-----	+
+	-----	-----	+

Report Cell Usage:

+	-----	-----	+
	Cell	Count	
+	-----	-----	+
1	IOBUF	1	
+	-----	-----	+

Report Instance Areas:

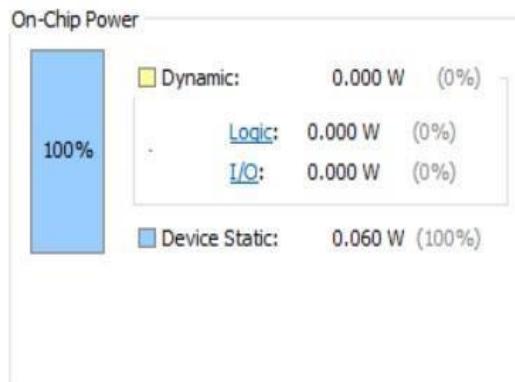
+	-----	-----	-----	-----	+
	Instance	Module	Cells		
+	-----	-----	-----	-----	+
1	top			1	
+	-----	-----	-----	-----	+

Finished Writing Synthesis Report : Time (s): cpu = 00:00:06 ; elapsed = 00:00:07 . Memory (MB): peak = 410.449 ; gain = 251.328

POWER REPORT :-

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: 0.06 W
Junction Temperature: 25.3 °C
Thermal Margin: 74.7 °C (15.5 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



16) 8 Bit Substractor

Design

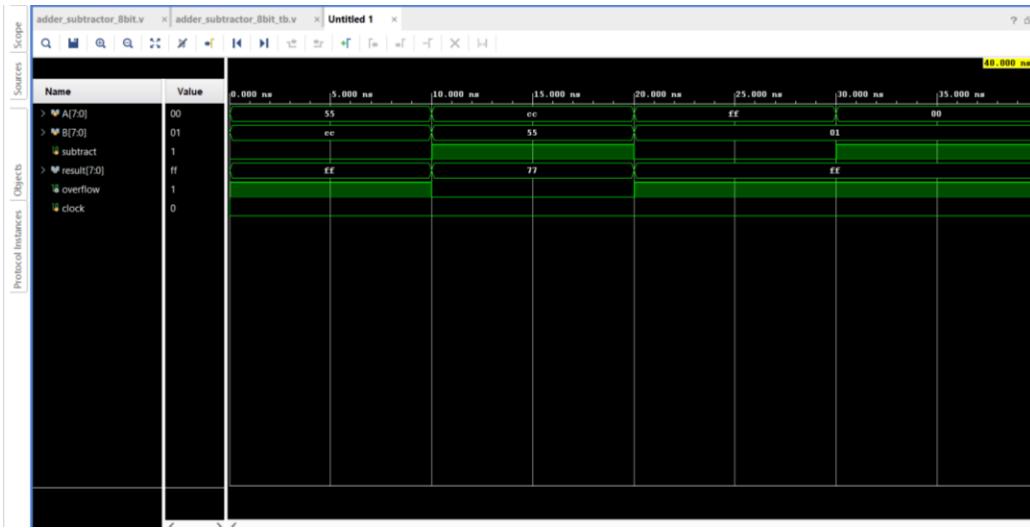
```
module subtract_8bit (a, b, result);
//define inputs and outputs
input [7:0] a, b;
output [7:0] result;
//variables used in always are declared as reg
reg [7:0] result;
//neg_b is used in the subtract operation
reg [7:0] neg_b;
always @ (a or b)
begin
    neg_b = ~b + 1;
    result = a + neg_b;
end
endmodule
```

Testbench

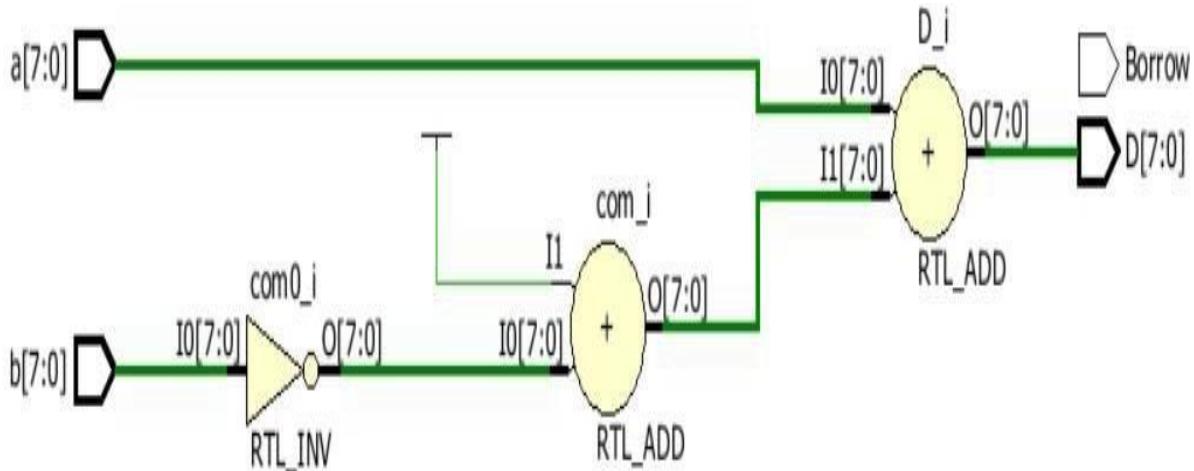
```
module sub_8bit_bh_tb;
reg [7:0] a, b; //inputs are reg for test bench
wire [7:0] result; //outputs are wire for test bench

//instantiate the module into the test bench
subtract_8bit dut(a,b,result);
//apply input vectors
initial
begin
#0 a = 8'b0000_0011; b = 8'b0000_0001; //3-1 = 2
#10 a = 8'b0000_0100; b = 8'b0000_0011; //4-3 = 1
#10 a = 8'b0000_0110; b = 8'b0000_0011; //6-3 = 3
#10 a = 8'b0000_1110; b = 8'b0000_0111; //14-7 = 7
#10 a = 8'b0000_1100; b = 8'b0000_0101; //12-5 = 7
#10 a = 8'b0100_1100; b = 8'b0001_0101; //76-21 = 55
#10 a = 8'b0011_0001; b = 8'b0001_1000; //49-24 = 25
#10 a = 8'b0111_0001; b = 8'b0011_1001; //113-57 = 56
#10 a = 8'b1000_0001; b = 8'b1000_0001; //-127+127=0
#10 a = 8'b0110_0001; b = 8'b0010_0001; //97-33 = 64
#10 a = 8'b1100_0110; b = 8'b1000_0101; //-58+123=65
#10 a = 8'b0101_0101; b = 8'b0000_1111; //85-15 = 70
#10 a = 8'b1111_1000; b = 8'b0000_0010; //-8-2 = -10
end
initial
begin
//display variables
$monitor ("a = %b, b = %b, rslt = %b", a, b, result);
#200 $finish;
end
endmodule
```

Output



RTL SCHEMATIC :-



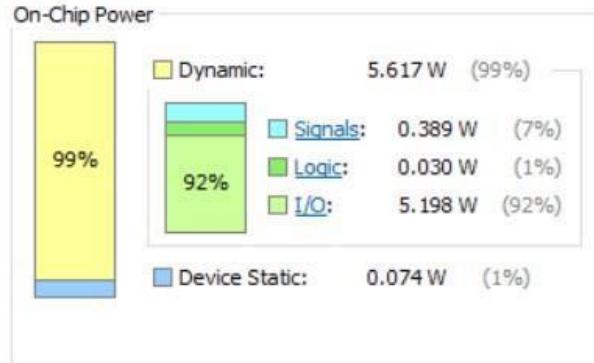
SYNTHESIS REPORT :-

```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
+---+---+  
| BlackBox name | Instances |  
+---+---+  
+---+---+  
  
Report Cell Usage:  
+---+---+  
| Cell | Count |  
+---+---+  
|1 | CARRY4 | 2|  
|2 | LUT2 | 8|  
|3 | IBUF | 16|  
|4 | OBUF | 8|  
|5 | OBUFT | 1|  
+---+---+  
  
Report Instance Areas:  
+---+---+  
| Instance | Module | Cells |  
+---+---+  
|1 | top | | 35|  
+---+---+  
  
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:06 ; elapsed = 00:00:07 . Memory (MB): peak = 410.813 ; gain = 251.418  
-----
```

POWER REPORT :-

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: 5.691 W
Junction Temperature: 52.2 °C
Thermal Margin: 47.8 °C (9.9 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: [Low](#)



17) 8-BIT ADDER/SUBTRACTOR

Design

```
module adder_subtractor (a, b, mode, result, ovfl);
input [7:0] a, b; //define inputs and outputs
input mode;
output [7:0] result;
output ovfl;
//variables rslt and ovfl are left-hand side targets
//in the always block and are declared as type reg
reg [7:0] result;
reg ovfl;
wire [7:0] a, b; //since inputs default to wire
wire mode; //the type wire is not required
//neg_b = ~b + 1 specifies an internal register
reg [7:0] neg_b ;
always @ (a or b or mode)
begin
if (mode == 0) //add
begin
result = a + b;
ovfl =(a[7] & b[7] & ~result[7]) |
(~a[7] & ~b[7] & result[7]);
end
else //subtract
begin
neg_b=~b+1;
result = a + neg_b;
ovfl =(a[7] & neg_b[7] & ~result[7]) |
(~a[7] & ~neg_b[7] & result[7]);
end
endmodule
```

Testbench

```
module adder_subtractor_tb();
//inputs are reg for test bench
//outputs are wire for test bench
reg [7:0] a, b;
reg mode;
wire [7:0] result;
wire ovfl;
//instantiate the module into the test bench
adder_subtractor inst1 (a, b, mode, result, ovfl);

initial //apply input vectors
begin
#0 a = 8'b0000_0000; b = 8'b0000_0001; mode = 1'b0;
#10 a = 8'b0000_0000; b = 8'b0000_0001; mode = 1'b1;
#10 a = 8'b0000_0001; b = 8'b1111_1001; mode = 1'b0;
#10 a = 8'b0000_0001; b = 8'b1111_1001; mode = 1'b1;
#10 a = 8'b0000_0001; b = 8'b1000_0001; mode = 1'b0;
#10 a = 8'b0000_0001; b = 8'b1000_0001; mode = 1'b1;
//ovfl = 1
#10 a = 8'b1111_0000; b = 8'b0000_0001; mode = 1'b0;
#10 a = 8'b1111_0000; b = 8'b0000_0001; mode = 1'b1;
#10 a = 8'b0110_1101; b = 8'b0100_0101; mode = 1'b0;
//ovfl = 1
#10 a = 8'b0010_1101; b = 8'b0000_0101; mode = 1'b1;
#10 a = 8'b0000_0110; b = 8'b0000_0001; mode = 1'b0;
#10 a = 8'b0000_0110; b = 8'b0000_0001; mode = 1'b1;
#10 a = 8'b0001_0101; b = 8'b0011_0001; mode = 1'b0;
#10 a = 8'b0001_0101; b = 8'b0011_0001; mode = 1'b1;
#10 a = 8'b1000_0000; b = 8'b1001_1100; mode = 1'b0;
//ovfl = 1
#10 a = 8'b1000_0000; b = 8'b1001_1100; mode = 1'b1;
#10 a = 8'b1000_0101; b = 8'b0010_0001; mode = 1'b0;
#10 a = 8'b1000_0101; b = 8'b0010_0001; mode = 1'b1;
//ovfl = 1
end

initial
begin
//display variables
$monitor ("a=%b| b=%b| mode=%b| result=%b| ovfl=%b",

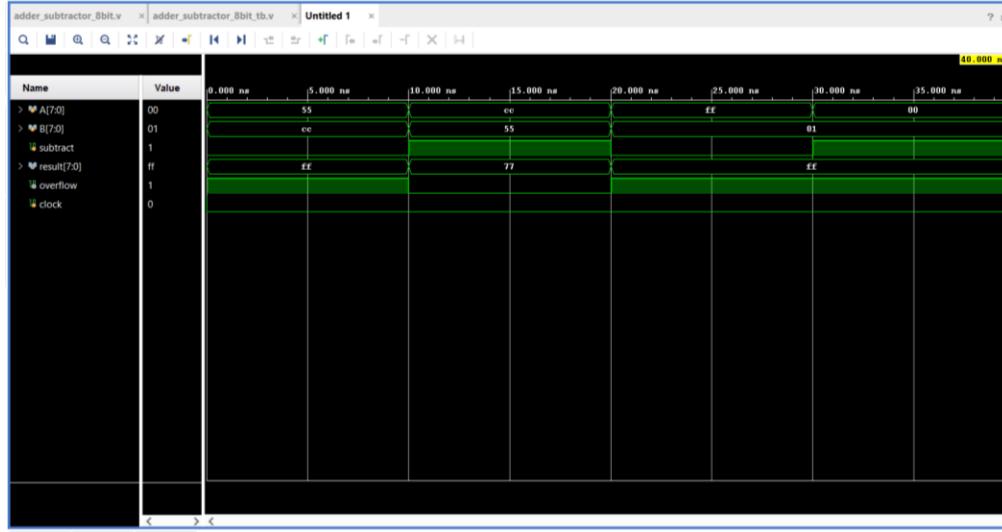
```

```

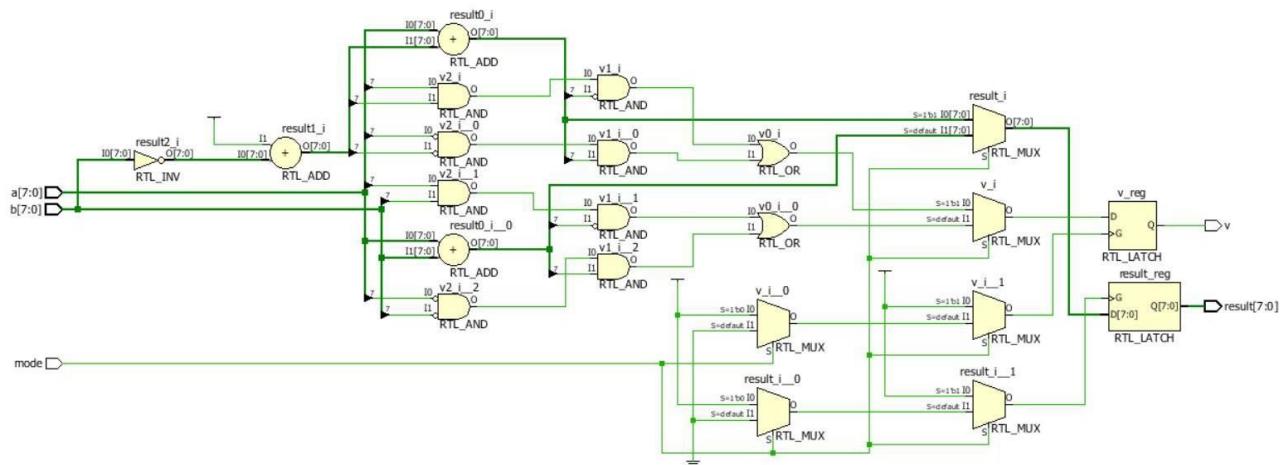
a, b, mode, result, ovfl);
#200 $finish;
end
endmodule

```

Output



RTL SCHEMATIC :-



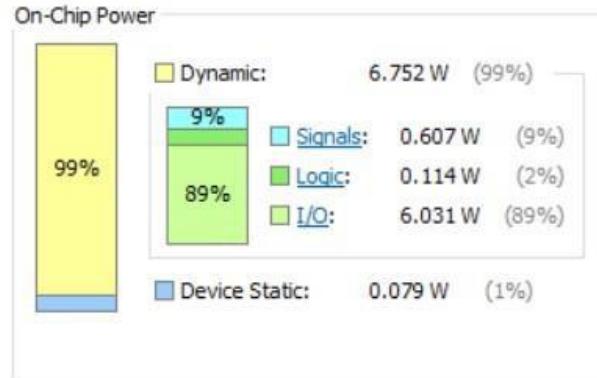
SYNTHESIS REPORT :-

```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
+---+---+  
| |BlackBox name |Instances |  
+---+---+  
+---+---+  
  
Report Cell Usage:  
+---+---+  
| |Cell |Count |  
+---+---+  
|1 |CARRY4 | 4|  
|2 |LUT2 | 16|  
|3 |LUT3 | 9|  
|4 |LUT6 | 2|  
|5 |IBUF | 17|  
|6 |OBUF | 9|  
+---+---+  
  
Report Instance Areas:  
+---+---+---+  
| |Instance |Module |Cells |  
+---+---+---+  
|1 |top | - | 57|  
+---+---+---+  
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:07 ; elapsed = 00:00:15 . Memory (MB): peak = 421.660 ; gain = 262.137  
-----
```

POWER REPORT :-

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: 6.831 W
Junction Temperature: 57.6 °C
Thermal Margin: 42.4 °C (8.8 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: [Low](#)



18) 4 Bit Multiplier

Design

```
module multiplier_4bit(a,b,product);
input [3:0] a,b;
output [7:0] product;
wire [3:0] m0;
wire [4:0] m1;
wire [5:0] m2;
wire [6:0] m3;

//middle terms
wire [7:0] s1,s2,s3;

assign m0={4{a[0]}}&b[3:0];
assign m1={4{a[1]}}&b[3:0];
assign m2={4{a[2]}}&b[3:0];
assign m3={4{a[3]}}&b[3:0];

assign s1= m0+(m1<<1);
assign s2= s1+(m2<<1);
assign s3= s2+(m3<<1);
assign product=s3;
endmodule
```

Testbench :-

```
module multiplier_4bit_tb();
reg [3:0] a,b;
wire [7:0] product;

// Instantiate the design module
multiplier_4bit dut(a,b,product);

initial
begin
#0 a = 4'b0110; b = 4'b0110;
#10 a = 4'b0010; b = 4'b0110;
#10 a = 4'b0111; b = 4'b0101;
#10 a = 4'b0111; b = 4'b0111;
#10 a = 4'b0101; b = 4'b0101;
#10 a = 4'b0111; b = 4'b0011;
#10 a = 4'b0100; b = 4'b0110;
```

```

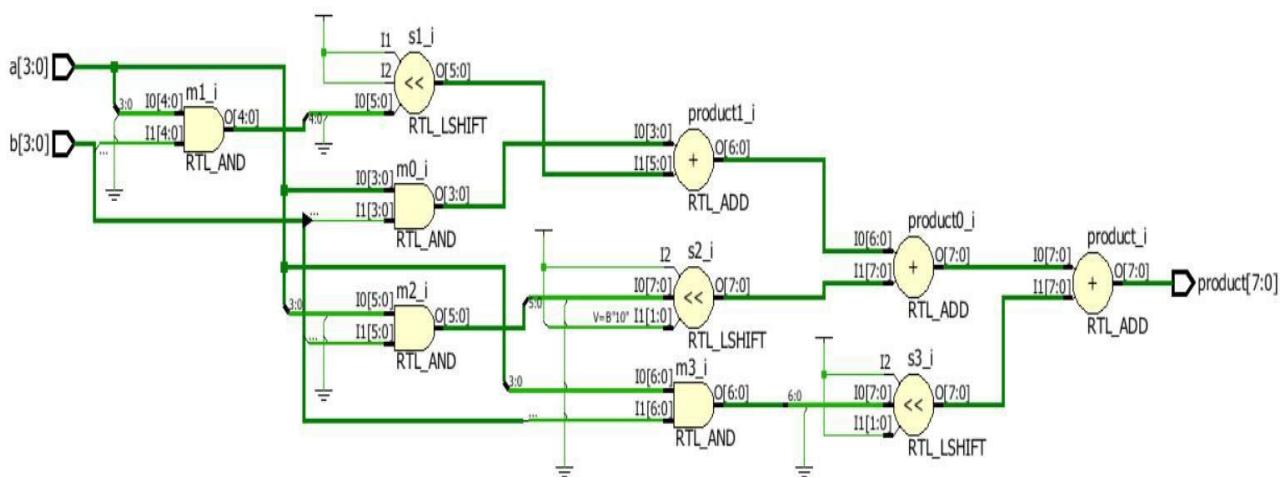
end

//display variables
initial
begin
$monitor ("A = %b, B = %b, Product = %b", a, b, product);
#100 $finish;
end

endmodule

```

RTL SCHEMATIC :-



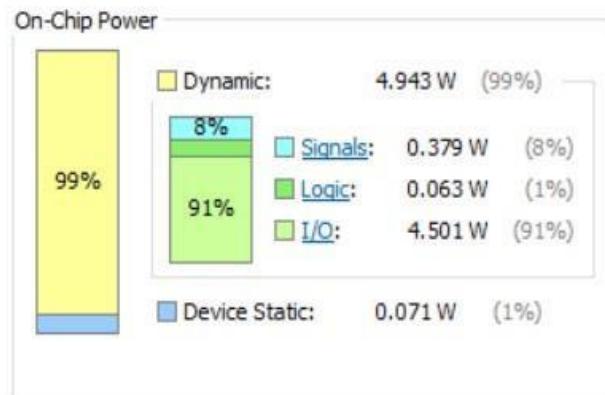
SYNTHESIS REPORT :-

```
)-----  
. Start Writing Synthesis Report  
|  
|  
| Report BlackBoxes:  
| +---+ +---+  
| | BlackBox name |Instances |  
| +---+ +---+  
| +---+ +---+  
|  
| Report Cell Usage:  
| +---+ +---+  
| | Cell |Count |  
| +---+ +---+  
| |1 |CARRY4 | 2|  
| |2 |LUT2 | 6|  
| |3 |LUT3 | 2|  
| |4 |LUT4 | 1|  
| |5 |LUT6 | 9|  
| |6 |IBUF | 8|  
| |7 |OBUF | 8|  
| +---+ +---+  
|  
| Report Instance Areas:  
| +---+ +---+ +---+  
| | Instance |Module |Cells |  
| +---+ +---+ +---+  
| |1 |top | | 36|  
| +---+ +---+ +---+  
|  
| Finished Writing Synthesis Report : Time (s): cpu = 00:00:06 ; elapsed = 00:00:07 . Memory (MB): peak = 421.020 ; gain = 261.523  
|-----
```

POWER REPORT :-

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: 5.014 W
Junction Temperature: 49.0 °C
Thermal Margin: 51.0 °C (10.6 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



19) Restoring Division

Design

```
module restoring_division(a,b,result,start);

//Defining inputs and outputs
input [7:0] a;
input [3:0] b;
output [7:0] result;
input start;

//Define internal net
wire [3:0] b_bar;

//Define internal registers
//Variables used in always block should be declared as reg
reg [3:0] b_neg;
reg [7:0] result;
reg [3:0] count;

assign b_bar=~b;

//always block to find 2's complement
always @(b_bar)
b_neg=b_bar+1;

always @(posedge start)
begin
result=a;
count =4'b0100;

if ((a!=0) && (b!=0))
while(count)
begin
result=result<<1;
result={{(result[7:4]+b_neg),result[3:0]};
if(result[7]==1)
begin
result= {(result[7:4] + b), result[3:1], 1'b0};
count=count-1;
end
else
begin
```

```

result={result[7:1],1'b1};
count=count-1;
end
end
end
endmodule

```

Testbench:-

```

module restoring_division_tb;

//inputs are reg for test bench
//outputs are wire for test bench
reg [7:0] a;
reg [3:0] b;
reg start;
wire [7:0] result;

//Instantiate the design module
restoring_division dut(a,b,result,start);

initial //apply input vectors
begin
#0 start = 1'b0;
a = 8'b0000_1101; b = 4'b0101;
#10 start = 1'b1;
#10 start = 1'b0;
#10 a = 8'b00011000; b = 4'b0111;
#10 start = 1'b1;
#10 start = 1'b0;
#10 a = 8'b0101_0010; b = 4'b0110;
#10 start = 1'b1;
#10 start = 1'b0;
#10 a = 8'b0011_1000; b = 4'b0111;
#10 start = 1'b1;
#10 start = 1'b0;
#10 a = 8'b0110_0100; b = 4'b0111;
#10 start = 1'b1;
#10 start = 1'b0;
#10 a = 8'b0110_1110; b = 4'b0111;
#10 start = 1'b1;
#10 start = 1'b0;
#10 a = 8'b0010_0101; b = 4'b0011;

```

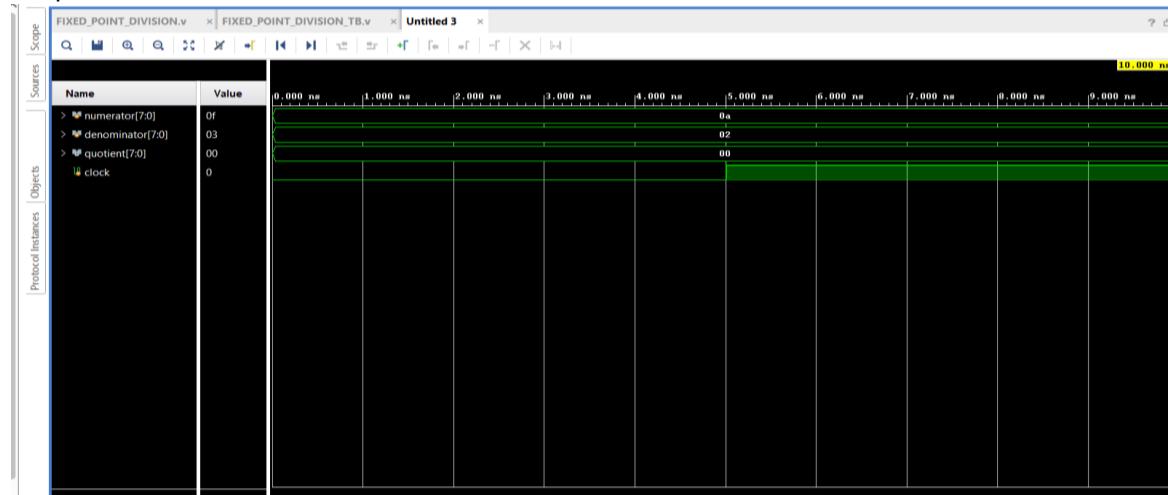
```

#10 start = 1'b1;
#10 start = 1'b0;
#10 a = 8'b0100_1000; b = 4'b0111;
#10 start = 1'b1;
#10 start = 1'b0;
#10 a = 8'b0101_0100; b = 4'b0110;
#10 start = 1'b1;
#10 start = 1'b0;
end

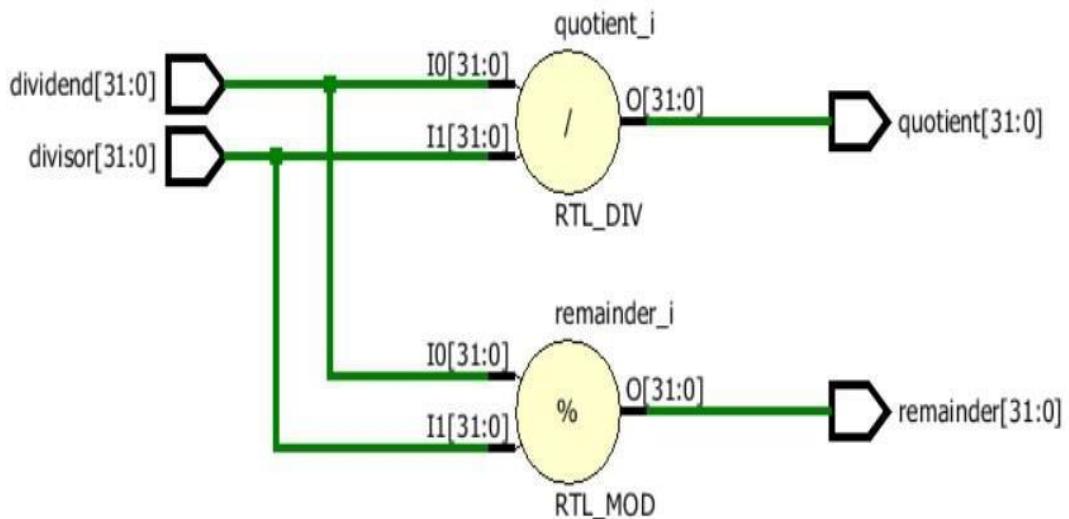
//Display Variables
initial
begin
$monitor ("Dividend = %b, Divisor = %b, Quotient = %b, Remainder = %b",
a, b, result[3:0], result[7:4]);
#300 $finish;
end
endmodule

```

Output:-



RTL SCHEMATIC :-



SYNTHESIS REPORT :-

```
Start Writing Synthesis Report
-----
Report BlackBoxes:
+-----+-----+
| |BlackBox name |Instances |
+-----+-----+
+-----+-----+

Report Cell Usage:
+-----+-----+
|     |Cell    |Count   |
+-----+-----+
|1    |CARRY4 | 576|
|2    |LUT1    | 124|
|3    |LUT2    | 101|
|4    |LUT3    | 1991|
|5    |LUT4    | 29|
|6    |LUT5    | 9|
|7    |LUT6    | 50|
|8    |IBUF    | 64|
|9    |OBUF    | 64|
+-----+-----+

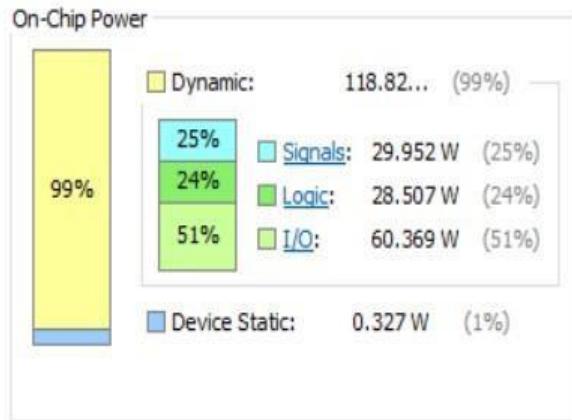

Report Instance Areas:
+-----+-----+-----+
|     |Instance |Module |Cells   |
+-----+-----+-----+
|1    |top      |        | 3008|
+-----+-----+-----+


Finished Writing Synthesis Report : Time (s): cpu = 00:00:08 ; elapsed = 00:00:08 . Memory (MB): peak = 417.785 ; gain = 258.359
```

POWER REPORT :-

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: **119.155 W**
Junction Temperature: **125.0 °C**
Thermal Margin: -494.4 °C (-103.3 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



20) Master Slave JK Flip Flop

Design

The screenshot shows a software interface for editing Verilog code. The tabs at the top are "master_slave.v" (selected), "master_slave_tb.v", and "Schematic". The code editor displays the following Verilog code:

```
1 module jk_ff(j,k,clk,q,q_bar);
2   input j,k,clk;
3   output q,q_bar;
4   reg q;
5   assign q_bar= ~q;
6   always @(posedge clk)
7   begin
8
9   case({j,k})
10  2'b00: q<=q;
11  2'b01: q<=0;
12  2'b10: q<=1;
13  2'b11: q<=~q;
14 endcase
15 end
16 endmodule
17
18 module master_slave(s,r,clk,qn,qn_bar,);
19   input s,r,clk;
20   output qn,qn_bar;
21
22   wire mq;
23   wire mq_bar;
24   wire mclk;
25   assign mclk= ~clk;
26
27   jk_ff Master(s,r,clk,mq,mq_bar);
28   jk_ff Slave(mq,mq_bar,mclk,qn,qn_bar);
29
30 endmodule
```

Testbench

master_slave.v x master_slave_tb.v * x Schematic x

C:/Users/DELL/Master Slave Flipflop/Master Slave Flipflop.srsc/sim_1/new/master_slave_tb.v

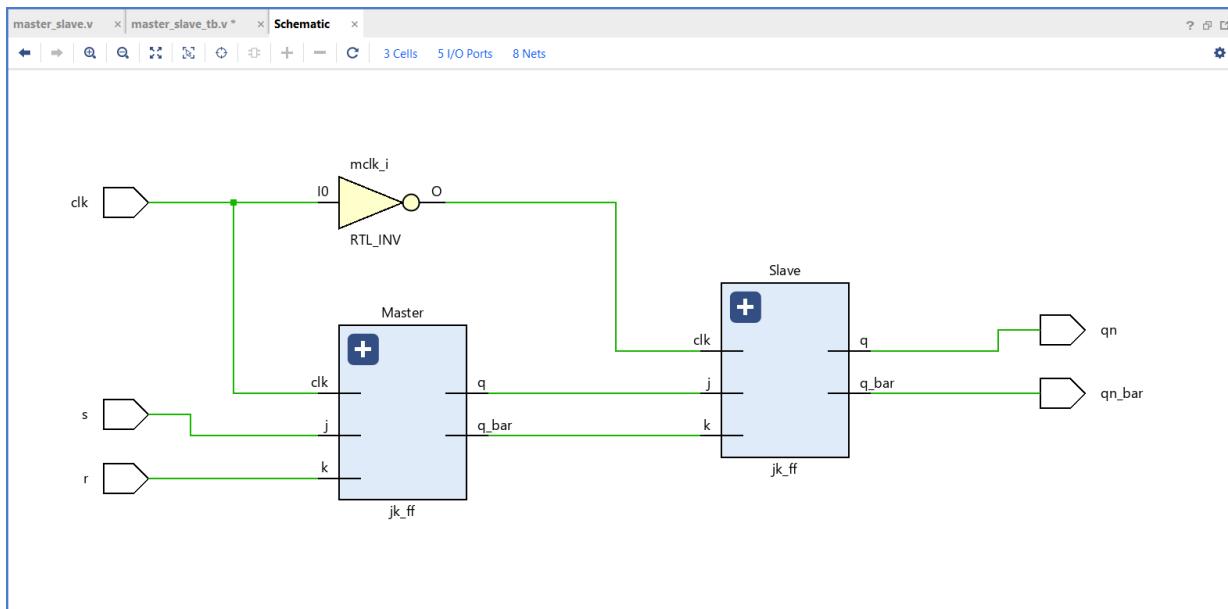
Q | B | ← | → | X | D | C | X | // | ☰ | ? |

```
1 module master_slave_tb;
2     reg s;
3     reg r;
4     reg clk;
5     wire qn;
6     wire qn_bar;
7
8     master_slave uut (
9         .s(s),
10        .r(r),
11        .clk(clk),
12        .qn(qn),
13        .qn_bar(qn_bar)
14    );
15 initial begin
16     clk=0;
17     #10 s = 0;r = 0;
18     #10 s = 0;r = 1;
19     #10 s = 1;r = 0;
20     #10 s = 1;r = 1;
21     #10 s = 0;r = 0;
22     #10 s = 0;r = 1;
23 end
24 always #5 clk=~clk;
25 initial
26 begin
27     $monitor("S=%b, R=%b, Q=%b, Q_bar=%b",s,r,qn,qn_bar);
28     #200 $finish;
29 end
30 endmodule
```

Output



RTL Schematic



SYNTHESIS REPORT :-

```

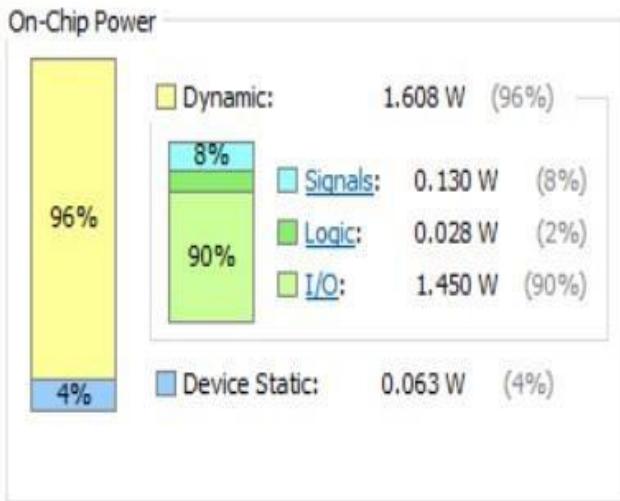
-----
Start Writing Synthesis Report
-----

Report BlackBoxes:
+-----+
| |BlackBox name |Instances |
+-----+
+-----+
+-----+
Report Cell Usage:
+-----+
| |Cell |Count |
+-----+
|1 |BUFGE | 1|
|2 |INV | 1|
|3 |LUT3 | 4|
|4 |FDRE | 4|
|5 |IBUF | 3|
|6 |OBUF | 2|
+-----+
Report Instance Areas:
+-----+
| |Instance |Module |Cells |
+-----+
|1 |top | 15| |
|2 | master |jk_flipflop | 6|
|3 | slave |jk_flipflop_0 | 21|
+-----+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:05 ; elapsed = 00:00:08 . Memory (MB): peak = 409.574 ; gain = 249.809
-----
```

POWER REPORT :-

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: 1.67 W
Junction Temperature: 33.0 °C
Thermal Margin: 67.0 °C (13.9 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



21) POSITIVE EDGE DETECTOR:-

Design

The screenshot shows a text editor window with the title bar "posedge_detct.v" and "posedge_tb.v". The main content is a Verilog module definition:

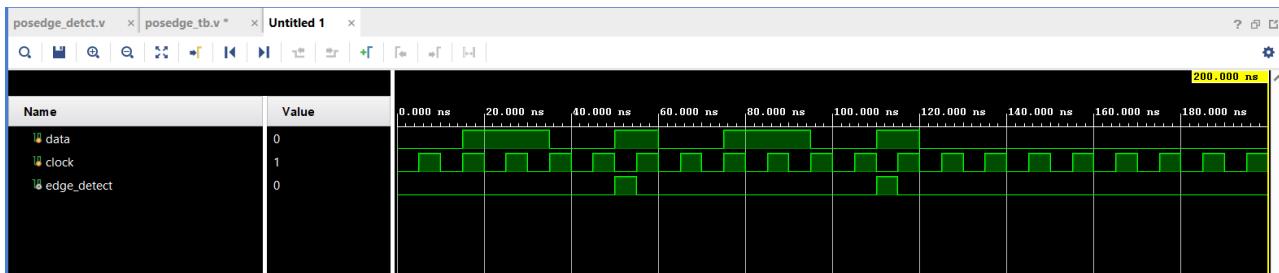
```
1 module edge_detector(
2     input  data,
3     input  clock,
4     output edge_detect
5 );
6     reg data_d;
7
8     always @ (posedge clock) begin
9         data_d <= data;
10    end
11    assign edge_detect = data & ~data_d;
12 endmodule
```

Testbench

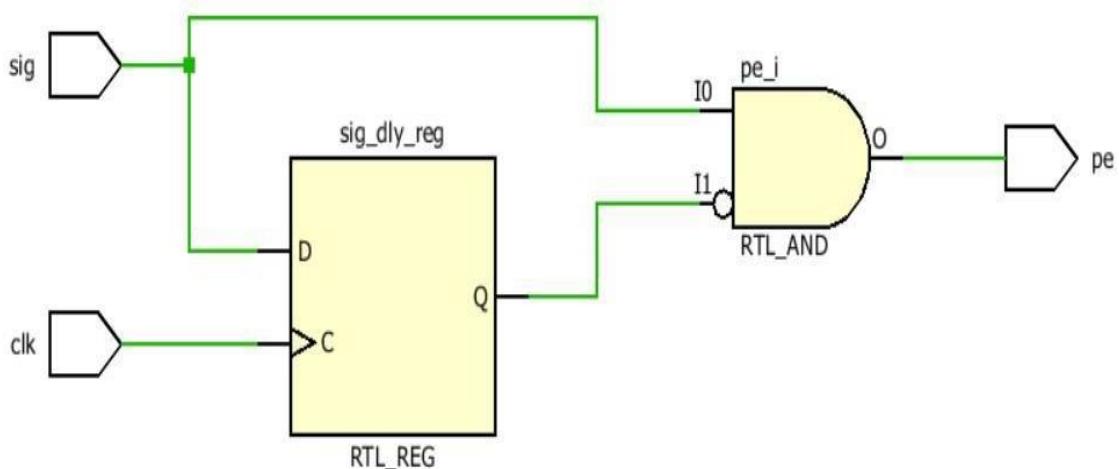
The screenshot shows a text editor window with the title bar "posedge_detct.v" and "posedge_tb.v". The main content is a Verilog testbench module:

```
1 module edge_tb;
2
3     reg data;
4     reg clock;
5     wire edge_detect;
6     edge_detector uut (
7         .data(data),
8         .clock(clock),
9         .edge_detect(edge_detect)
10    );
11    initial begin
12        data = 0;
13        clock = 0;
14        #15 data = 1;
15        #20 data= 0;
16        #15 data = 1;
17        #10 data = 0;
18        #15 data = 1;
19        #20 data= 0;
20        #15 data = 1;
21        #10 data = 0;
22    end
23
24    always #5 clock=~clock;
25    initial begin
26        $monitor("Data =%b, Edge_detect=%b ", data,edge_detect);
27        #200 $finish;
28    end
29
30 endmodule
```

Output



RTL SCHEMATIC :-



SYNTHESIS REPORT :-

Start Writing Synthesis Report

Report BlackBoxes:

BlackBox name	Instances
---------------	-----------

Report Cell Usage:

	Cell	Count
1	BUFG	1
2	LUT2	1
3	FDRE	1
4	IBUF	2
5	OBUF	1

Report Instance Areas:

	Instance	Module	Cells	
				+
	1	top		6
+	+	+	+	+

Finished Writing Synthesis Report : Time (s): cpu = 00:00:06 ; elapsed = 00:00:07 . Memory (MB): peak = 410.668 ; gain = 250.883

POWER REPORT :-

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

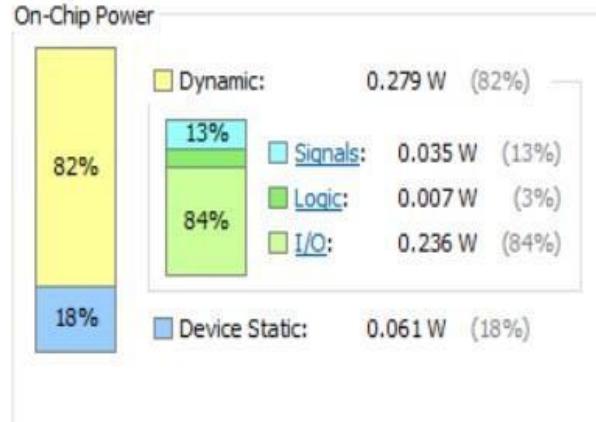
Total On-Chip Power:

Junction Temperature: 26.6 °C

Thermal Margin: 73.4 °C (15.3 W)

Effective ΔT : $4.8 \text{ }^{\circ}\text{C}/\text{W}$

Power supplied to off-chip devices: 0 W



22) BCD ADDER

Design

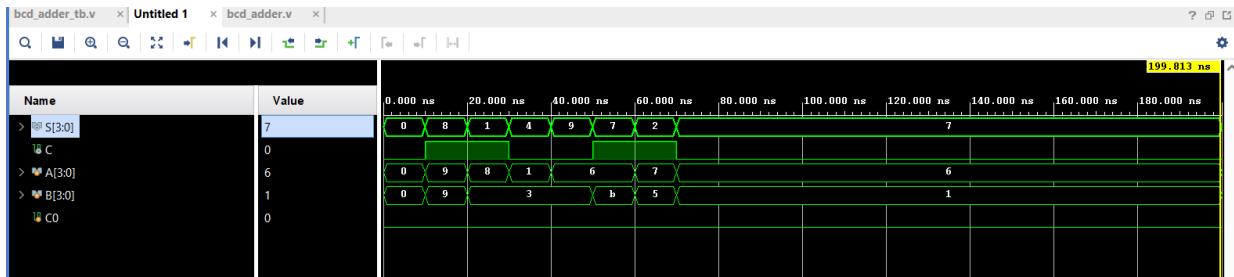
```
module four_bit_adder(x,y,cin,sum,cout);
input [3:0] x,y;
input cin;
output [3:0] sum;
output cout;
wire c0,c1,c2;
full_adder fa0(.a(x[0]),.b(y[0]),.c(cin),.s(sum[0]),.co(c0));
full_adder fa1(.a(x[1]),.b(y[1]),.c(c0),.s(sum[1]),.co(c1));
full_adder fa2(.a(x[2]),.b(y[2]),.c(c1),.s(sum[2]),.co(c2));
full_adder fa3(.a(x[3]),.b(y[3]),.c(c2),.s(sum[3]),.co(cout));
endmodule

module BCD_adder(S, C, A, B, C0);
input [3:0] A, B;
input C0;
output [3:0] S;
output C;
wire C1, C2, C3, C4, C5;
wire [3:0] X, Z;
and (C1, Z[3], Z[2]);
and (C2, Z[3], Z[1]);
or (C, C3, C1,C2);
xor (C5, C, C);
assign X[2] = C;
assign X[1] = C;
assign X[3] = C5;
assign X[0] = C5;
four_bit_adder F_1 ( A, B, C0,Z,C3);
four_bit_adder F_2 (X, Z, C0,S,C4);
endmodule
```

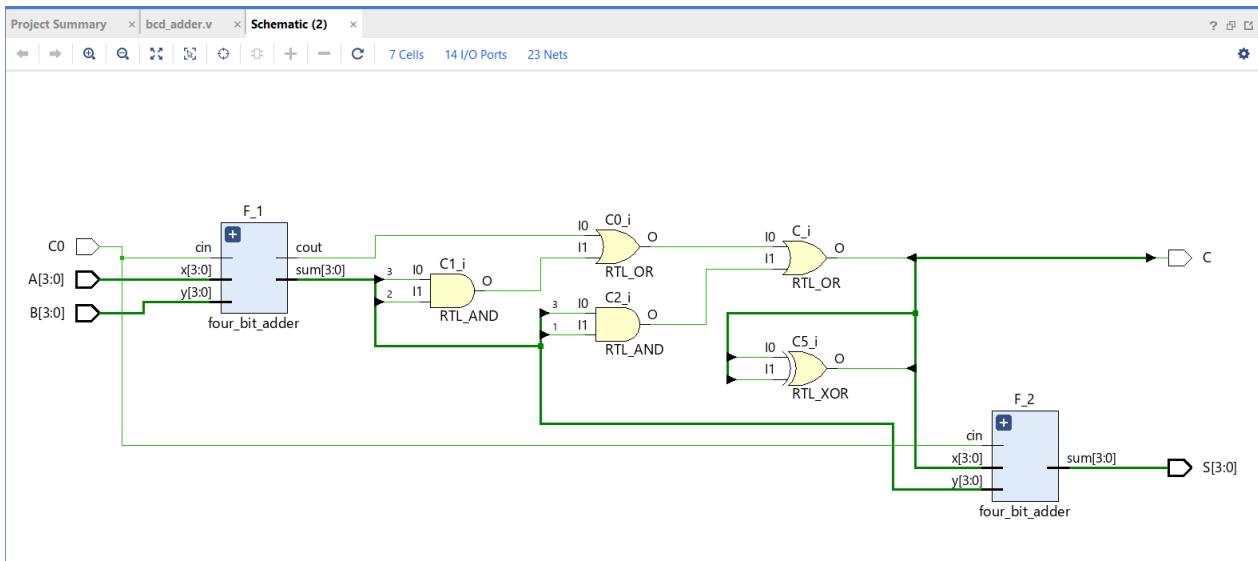
Testbench

```
module BCD_adder_tb;
wire [3:0] S;
wire C;
reg [3:0]A, B;
reg C0;
BCD_adder F1(S, C, A, B, C0);|
initial
begin
A[3:0] = 4'b0000; B = 4'b0000; C0 = 1'b0;
#10 A[3:0] = 4'b1001; B = 4'b1001; C0 = 1'b0;
#10 A[3:0] = 4'b1000; B = 4'b0011; C0 = 1'b0;
#10 A[3:0] = 4'b0001; B = 4'b0011; C0 = 1'b0;
#10 A[3:0] = 4'b0110; B = 4'b0011; C0 = 1'b0;
#10 A[3:0] = 4'b0110; B = 4'b1011; C0 = 1'b0;
#10 A[3:0] = 4'b0111; B = 4'b0101; C0 = 1'b0;
#10 A[3:0] = 4'b0110; B = 4'b0001; C0 = 1'b0;
end
initial
begin
$monitor("A=%b| B=%b| Cin=%b| BCD Sum=%b| Cout=%b",A,B,C0,S,C);
#200 $finish;
end
endmodule
```

Output



RTL Schematic



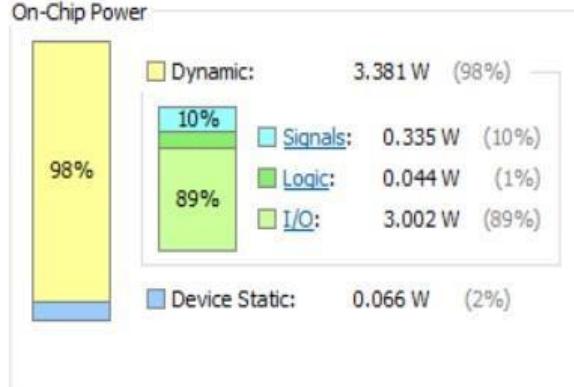
SYNTHESIS REPORT :-

```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
+-----+-----+  
| |BlackBox name |Instances |  
+-----+-----+  
+-----+-----+  
  
Report Cell Usage:  
+-----+-----+  
| |Cell |Count |  
+-----+-----+  
|1 |LUT2 | 1|  
|2 |LUT3 | 3|  
|3 |LUT5 | 2|  
|4 |LUT6 | 4|  
|5 |IBUF | 9|  
|6 |OBUF | 5|  
+-----+-----+  
  
Report Instance Areas:  
+-----+-----+-----+  
| |Instance |Module |Cells |  
+-----+-----+-----+  
|1 |top | | 24|  
+-----+-----+  
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:05 ; elapsed = 00:00:07 . Memory (MB): peak = 421.852 ; gain = 262.324
```

POWER REPORT :-

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: 3.448 W
Junction Temperature: 41.5 °C
Thermal Margin: 58.5 °C (12.2 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



23) 4-BIT CARRY SELECT ADDER:-

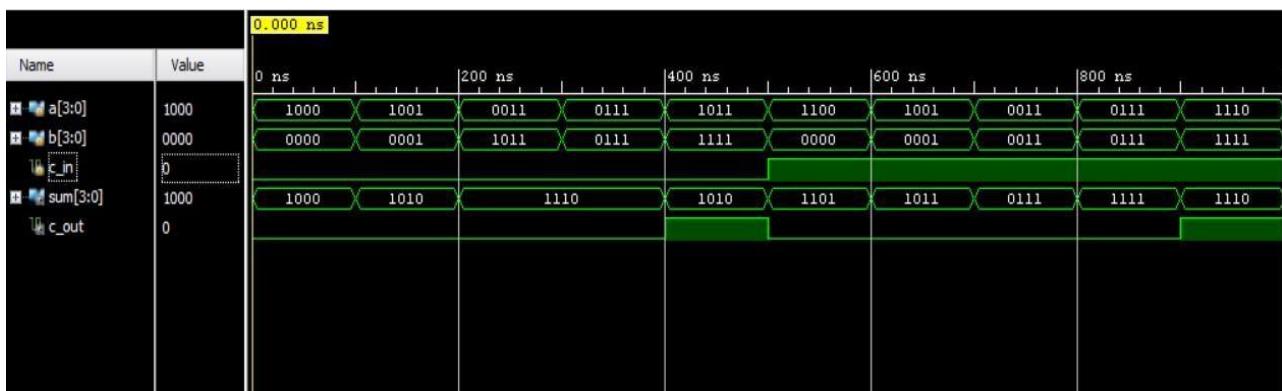
Design:-

```
module full_adder( A, B, Cin, S, Cout);
  input wire A, B, Cin;
  output reg S, Cout;
  always @(A or B or Cin)
  begin
    S = A ^ B ^ Cin;
    Cout = A&B | B&Cin | Cin&A ;
  end
endmodule
module mux(A,B,S,Y);
  input A,B,S;
  output reg Y;
  always@(A,B,S)
  begin
    Y=~S&A | S&B;
  end
endmodule
module carry_select (x,y,carry,s,cout);
  input [3:0]x,y;
  input carry;
  output [3:0]s;
  output cout;
  wire w1,w2,w3,w4,w5,w6,w7,w8,w9,w10,w11,w12,w13,w14,w15,w16;
  full_adder fa0 (x[0],y[0],1'b0,w1,w2);
  full_adder fa1 (x[1],y[1],w2,w3,w4);
  full_adder fa2 (x[2],y[2],w4,w5,w6);
  full_adder fa3 (x[3],y[3],w6,w7,w8);
  full_adder fa4 (x[0],y[0],1'b1,w9,w10);
  full_adder fa5 (x[1],y[1],w10,w11,w12);
  full_adder fa6 (x[2],y[2],w12,w13,w14);
  full_adder fa7 (x[3],y[3],w14,w15,w16);
  mux mu0(w1,w9,carry,s[0]);
  mux mu1(w3,w11,carry,s[1]);
  mux mu2(w5,w13,carry,s[2]);
  mux mu3(w7,w15,carry,s[3]);
  mux mu4(w8,w16,carry,cout);
endmodule
```

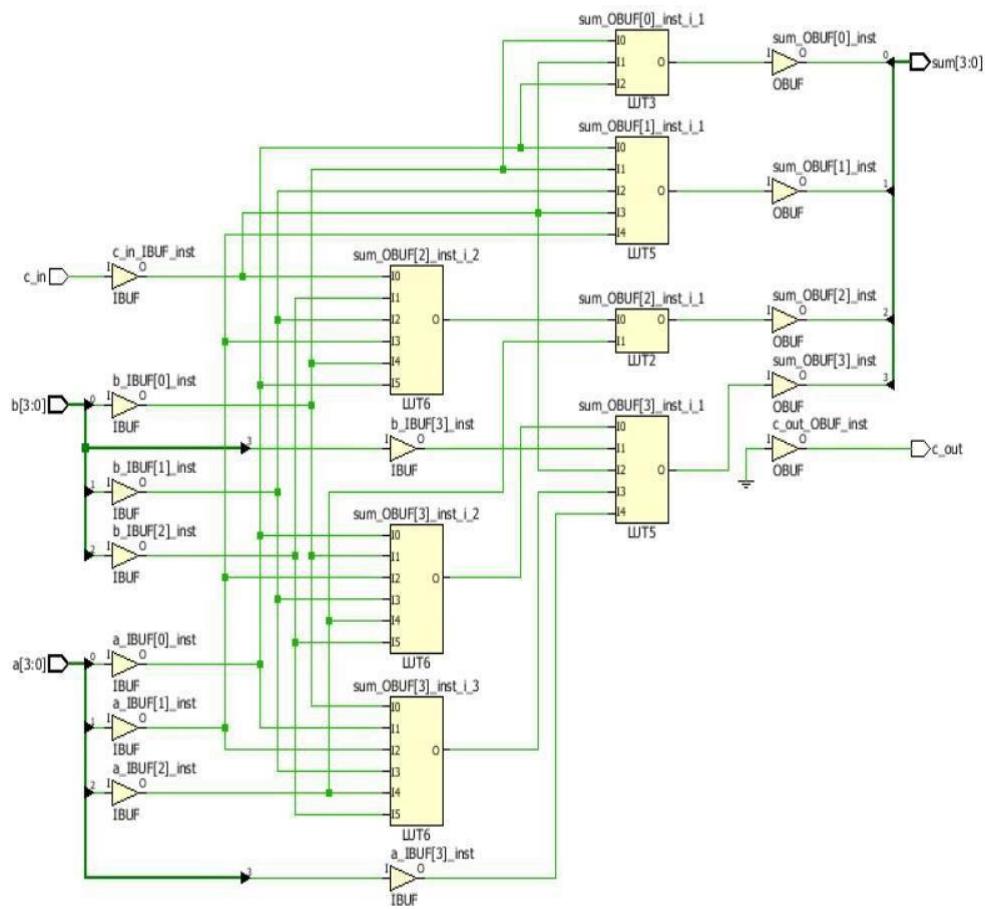
Testbench:-

```
module carry_select_tb();
reg [3:0]x,y;
reg carry;
wire[3:0]s;
wire cout ;
carry_select dut(.x(x),.y(y),.carry(carry),.s(s),.cout(cout));
initial
begin
#0 carry=1'b0;x=4'b0001;y=4'b0110;
#10 carry=1'b1;x=4'b1101;y=4'b1010;
#10 carry=1'b1;x=4'b0111;y=4'b0010;
#10 carry=1'b1;x=4'b0101;y=4'b0010;
end
initial
begin
$monitor("x=%b,y=%b,carry=%b,s=%b,cout=%b",x,y,carry,s,cout);
#1000 $finish;
end
endmodule
```

Output :-



RTL Schematic :-



SYNTHESIS REPORT :-

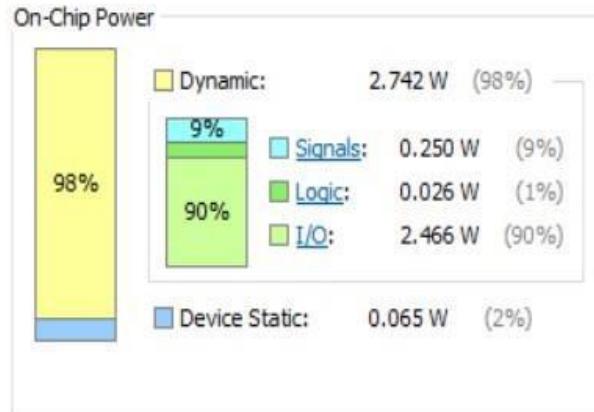
```
Start Writing Synthesis Report

Report BlackBoxes:
+---+---+
| |BlackBox name |Instances |
+---+---+
+---+---+
+---+---+
Report Cell Usage:
+---+---+
| |Cell |Count |
+---+---+
|1 |LUT2 | 1|
|2 |LUT3 | 1|
|3 |LUT5 | 2|
|4 |LUT6 | 3|
|5 |IBUF | 9|
|6 |OBUF | 5|
+---+---+
Report Instance Areas:
+---+---+
| |Instance |Module |Cells |
+---+---+
|1 |top | 21|
+---+---+
Finished Writing Synthesis Report : Time (s): cpu = 00:00:06 ; elapsed = 00:00:07 . Memory (MB): peak = 420.801 ; gain = 261.102
```

POWER REPORT :-

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: 2.807 W
Junction Temperature: 38.4 °C
Thermal Margin: 61.6 °C (12.8 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



24) Moore FSM

Design

```
module moore_fsm(din, reset, clk, y);
input din;
input clk;
input reset;
output reg y;
reg [2:0] cst, nst;
localparam 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
      else
      begin
        nst = s0;
        y=1'b0;
      end
    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
endmodule
```

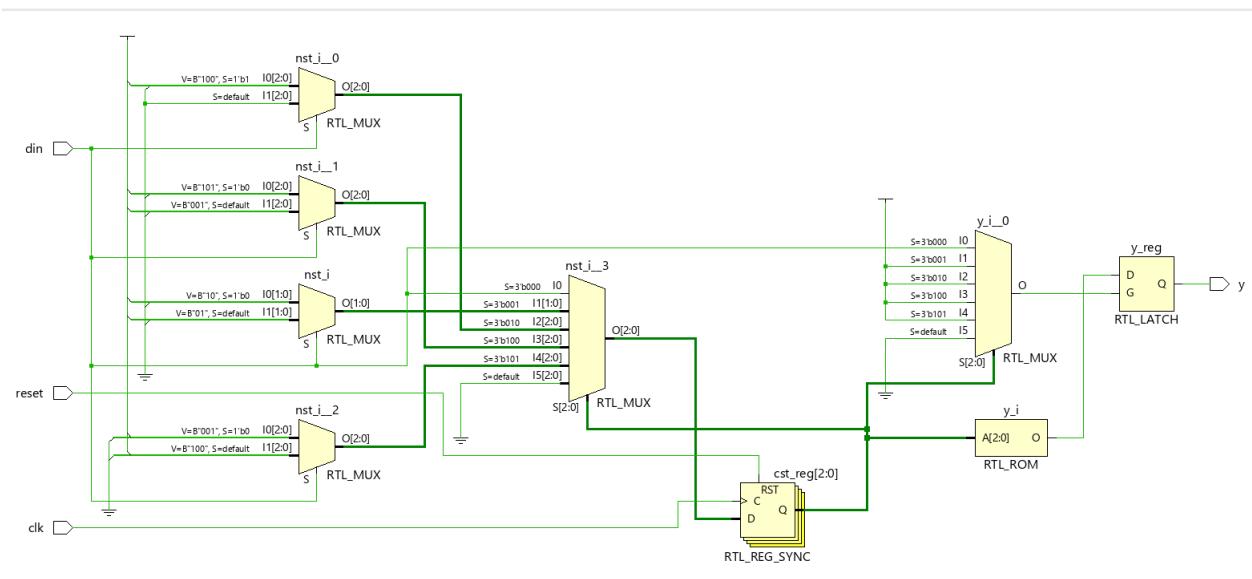
```
end
always@ (posedge clk)
begin
    if (reset)
        cst <= S0;
    else
        cst <= nst;
end
endmodule
```

Testbench

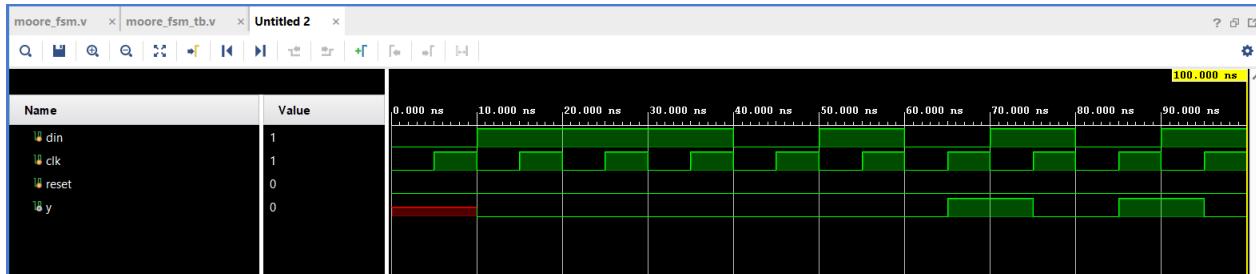
```
module moore_fsm_tb;
reg din,clk,reset;
wire y;
moore_fsm m1(din, reset, clk, y);
initial
begin
reset=0;clk=0;din=0;
$monitor($time, , , "c=%b",clk,, "y=%b",y,, "r=%b",reset,, "d=%b",din);
#10 din=1;
#10 din=1;
#10 din=1;
#10 din=0;
#10 din=1;
#10 din=0;
#10 din=1;
#10 din=0;
#10 din=1;
#10 din=1;

end
always
#5 clk=~clk;
initial
#100 $finish ;
endmodule
```

RTL Schematic



Output



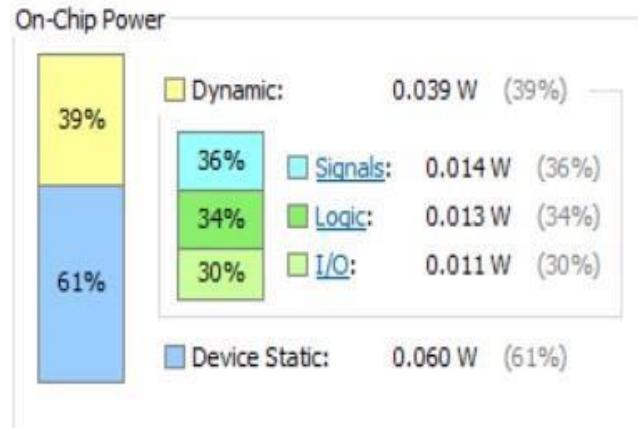
SYNTHESIS REPORT :-

```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
+-----+  
| |BlackBox name |Instances |  
+-----+  
+-----+  
  
Report Cell Usage:  
+-----+  
| |Cell |Count |  
+-----+  
|1 |BUFG | 1|  
|2 |LUT3 | 2|  
|3 |LUT5 | 3|  
|4 |FDRE | 3|  
|5 |LD | 1|  
|6 |IBUF | 3|  
|7 |OBUF | 1|  
+-----+  
  
Report Instance Areas:  
+-----+  
| |Instance |Module |Cells |  
+-----+  
|1 |top | | 14|  
+-----+  
  
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:04 ; elapsed = 00:00:09 . Memory (MB): peak = 407.980 ; gain = 248.656  
-----
```

POWER REPORT :-

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

Total On-Chip Power: 0.099 W
Junction Temperature: 25.5 °C
Thermal Margin: 74.5 °C (15.5 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low



25) N:1 Multiplexer

Design

```
n1_mux.v  x  n1_mux_tb.v  x  Untitled 3  x
C:/Users/DELL/n1 mux/n1 mux.srcc/sources_1/new/n1_mux.v

Q | H | ← | → | X | E | F | X | // | ■ | ? |

1 | module n_1mux(in,sel,y);
2 | parameter n=16;
3 | input [n-1:0] in;
4 | input [3:0] sel;
5 | output reg y;
6 |
7 | always @(in or sel)
8 | begin
9 |   case(sel)
10 |     4'b0000: y= in[0];
11 |     4'b0001: y= in[1];
12 |     4'b0010: y= in[2];
13 |     4'b0011: y= in[3];
14 |     4'b0100: y= in[4];
15 |     4'b0101: y= in[5];
16 |     4'b0110: y= in[6];
17 |     4'b0111: y= in[7];
18 |     4'b1000: y= in[8];
19 |     4'b1001: y= in[9];
20 |     4'b1010: y= in[10];
21 |     4'b1011: y= in[11];
22 |     4'b1100: y= in[12];
23 |     4'b1101: y= in[13];
24 |     4'b1110: y= in[14];
25 |     4'b1111: y= in[n-1];
26 |   default: y=4'b0000;
27 | endcase
28 | end
29 | endmodule
```

Testbench

The screenshot shows a text editor window with three tabs at the top: "n1_mux.v", "n1_mux_tb.v", and "Untitled 3". The "n1_mux_tb.v" tab is active, displaying Verilog testbench code. The code defines a module n_1mux_tb with an initial block containing multiple stimulus cases for a 1-to-16 multiplexer (n_1mux). The testbench monitors the inputs and output.

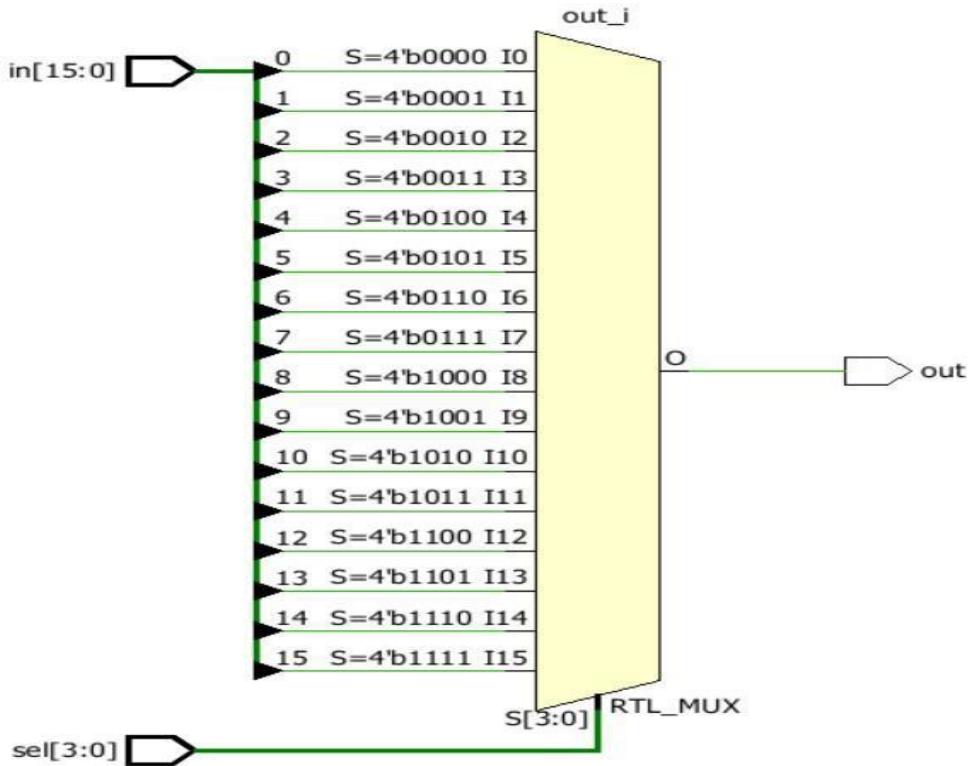
```
1 module n_1mux_tb;
2     reg [15:0] in;
3     reg [3:0] sel;
4     wire y;
5     n_1mux uut (
6         .in(in),
7         .sel(sel),
8         .y(y)
9     );
10    initial begin
11        in=16'b1000000000000000; sel=4'b0000;
12        #10 in=16'b0100000000000010; sel=4'b0001;
13        #10 in=16'b0010000000000100; sel=4'b0010;
14        #10 in=16'b0001000000000000; sel=4'b0011;
15        #10 in=16'b0000100000011000; sel=4'b0100;
16        #10 in=16'b0000010000000000; sel=4'b0101;
17        #10 in=16'b0000001000000000; sel=4'b0110;
18        #10 in=16'b0000000111100000; sel=4'b0111;
19        #10 in=16'b0000000010000000; sel=4'b1000;
20        #10 in=16'b0000000001000000; sel=4'b1001;
21        #10 in=16'b0000011000100000; sel=4'b1010;
22        #10 in=16'b000000000010000; sel=4'b1011;
23        #10 in=16'b000000000001000; sel=4'b1100;
24        #10 in=16'b000000000000100; sel=4'b1101;
25        #10 in=16'b000000000000010; sel=4'b1110;
26        #10 in=16'b1000000000000001; sel=4'b1111;
27    end
28
29    initial
30    begin
31        $monitor("in=%b | sel=%b | out=%b",in,sel,y);
32    end

```

Output :-



RTL SCHEMATIC :-



SYNTHESIS REPORT :-

```
-----  
Start Writing Synthesis Report  
-----  
  
Report BlackBoxes:  
+---+---+  
| |BlackBox name |Instances |  
+---+---+  
+---+---+  
  
Report Cell Usage:  
+---+---+  
| |Cell |Count |  
+---+---+  
|1 |LUT6 | 4|  
|2 |MUXF7 | 2|  
|3 |MUXF8 | 1|  
|4 |IBUF | 20|  
|5 |OBUF | 1|  
+---+---+  
  
Report Instance Areas:  
+---+---+  
| |Instance |Module |Cells |  
+---+---+  
|1 |top | | 28|  
+---+---+  
  
-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:06 ; elapsed = 00:00:07 . Memory (MB): peak = 413.047 ; gain = 253.766
```

POWER REPORT :-

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: 0.982 W
Junction Temperature: 29.7 °C
Thermal Margin: 70.3 °C (14.6 W)
Effective ΔJA: 4.8 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low

