SWITCHES AND LEDS

Verilog code:

```
module swiled (
 input wire s1, s2, // input signals
output reg L1, // individual LED outputs
output reg L2,
output reg L3,
output reg L4 );

// Define parameters for count values
parameter L1_COUNT = 5;
parameter L2_COUNT = 10;
parameter L3_COUNT = 15;
parameter L4_COUNT = 2

reg [3:0] count1; // 4-bit counter for s1
reg [3:0] count2; // 4-bit counter for s2

// Always block to handle LED control based on switch inputs
always @(*) begin
// Initialize all LEDs to OFF
```

```
    L1 = 0;
    L2 = 0;
    L3 = 0;
    L4 = 0;

// Update LED states based on switch inputs
    if (s1 == 0) begin
    L1 = 1; // Turn ON L1 when s1 is 0
    end else begin
    L2 = 1; // Turn ON L2 when s1 is 1
    end
    if (s2 == 0) begin
    L3 = 1; // Turn ON L3 when s2 is 0
    end else begin
    L4 = 1; // Turn ON L4 when s2 is 1
    end
    end
    endmodule
```

Testbench Code:

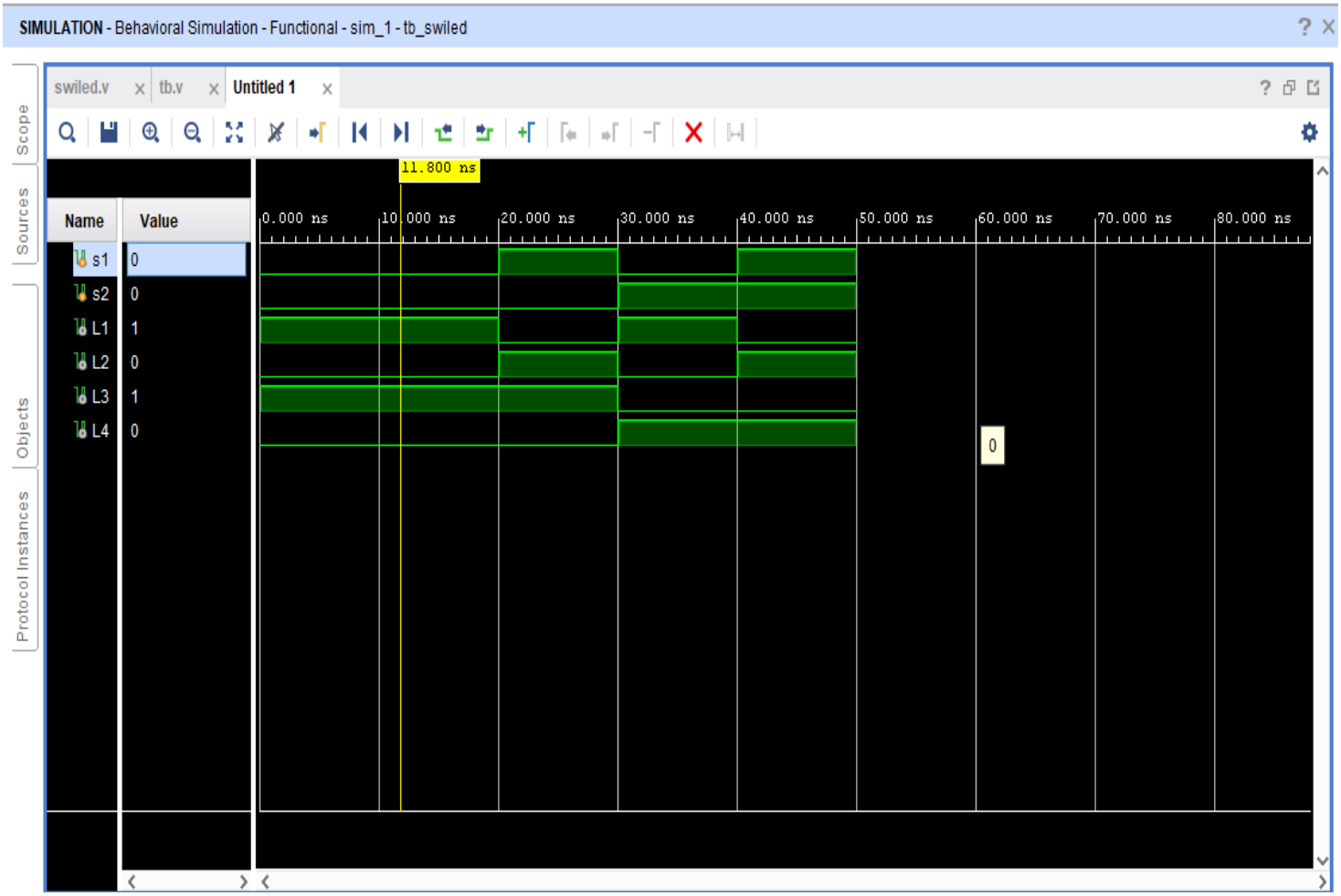
```
    module tb_swiled;
    // Inputs
    reg s1;
    reg s2;
```

```
    // Outputs
      wire L1;
      wire L2;
      wire L3;
      wire L4;

   // Instantiate the Unit Under Test (UUT)
    swiled uut (
                    .s1(s1), .s2(s2), .L1(L1), .L2(L2), .L3(L3), .L4(L4));
     initial begin
 // Initialize Inputs
      s1 = 0;
      s2 = 0;
// Wait for global reset to finish
      #10;

// Test Case 1: s1 = 0, s2 = 0
    s1 = 0; s2 = 0;
#10; // Wait for 10 time units
$display("Test Case 1: s1 = 0, s2 = 0 -> L1 = %b, L2 = %b, L3 =
        %b, L4 = %b", L1, L2, L3, L4);

// Test Case 2: s1 = 1, s2 = 0
    s1 = 1; s2 = 0;
#10; // Wait for 10 time units
```

```
 $display("Test Case 2: s1 = 1, s2 = 0 -> L1 = %b, L2 = %b, L3 =
        %b, L4 = %b", L1, L2, L3, L4);


// Test Case 3: s1 = 0, s2 = 1

   s1 = 0; s2 = 1;

#10; // Wait for 10 time units

 $display("Test Case 3: s1 = 0, s2 = 1 -> L1 = %b, L2 = %b, L3 =
        %b, L4 = %b", L1, L2, L3, L4);


 // Test Case 4: s1 = 1, s2 = 1

    s1 = 1; s2 = 1;

 #10; // Wait for 10 time units

 $display("Test Case 4: s1 = 1, s2 = 1 -> L1 = %b, L2 = %b, L3 =
        %b, L4 = %b", L1, L2, L3, L4);


 $finish; // End simulation

 end


 initial begin

$monitor("Time: %0d, s1: %b, s2: %b, L1: %b, L2: %b, L3: %b,
   L4: %b", $time, s1, s2, L1, L2, L3, L4);

 end

 endmodule
```

OUTPUT WAVEFORM :

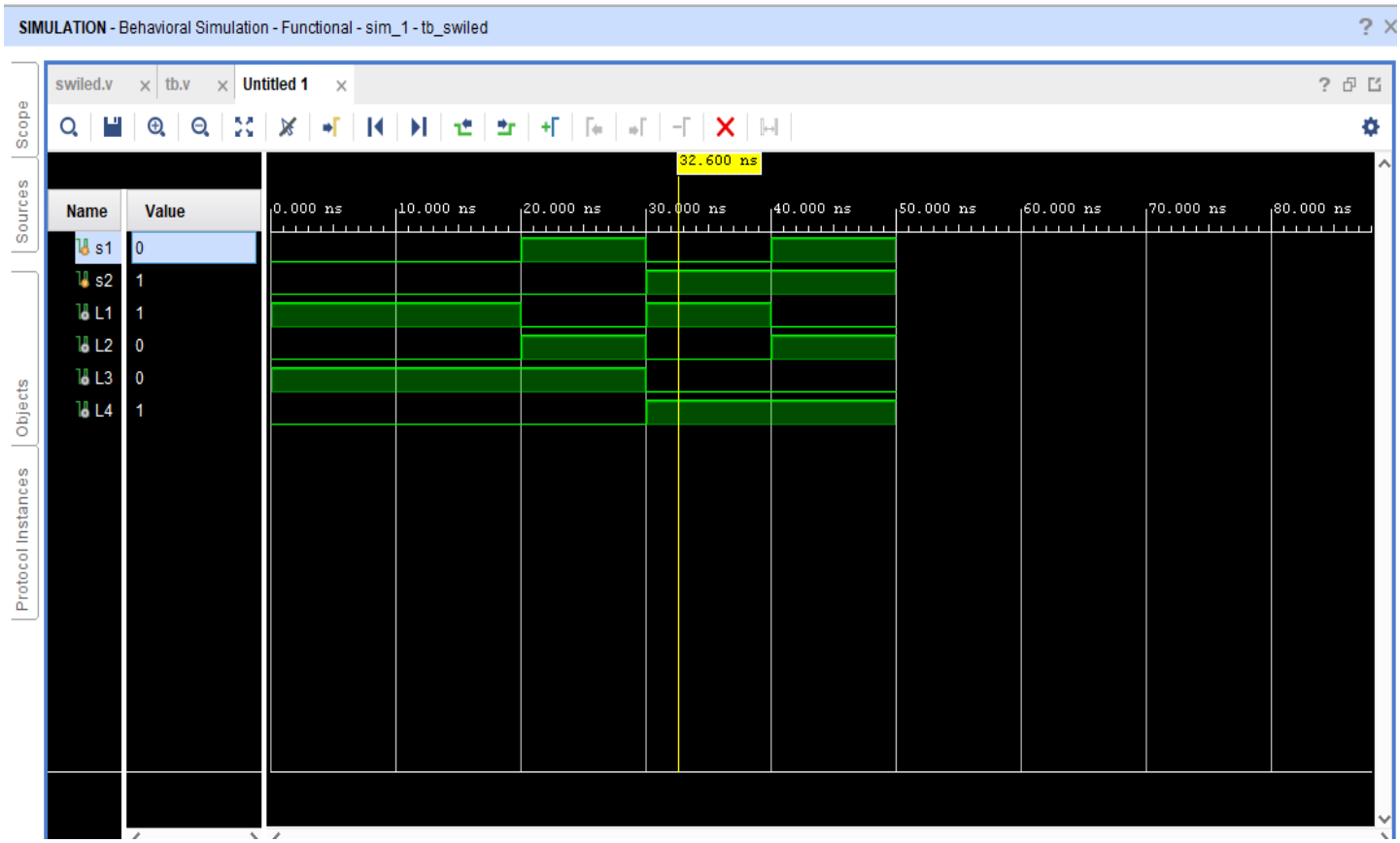
1) For S1=0, S2=0;
   L1 gets turns ON
   L3 gets turns ON (since S2=0)



2) For S1=0, S2=1;

L1 gets turns ON

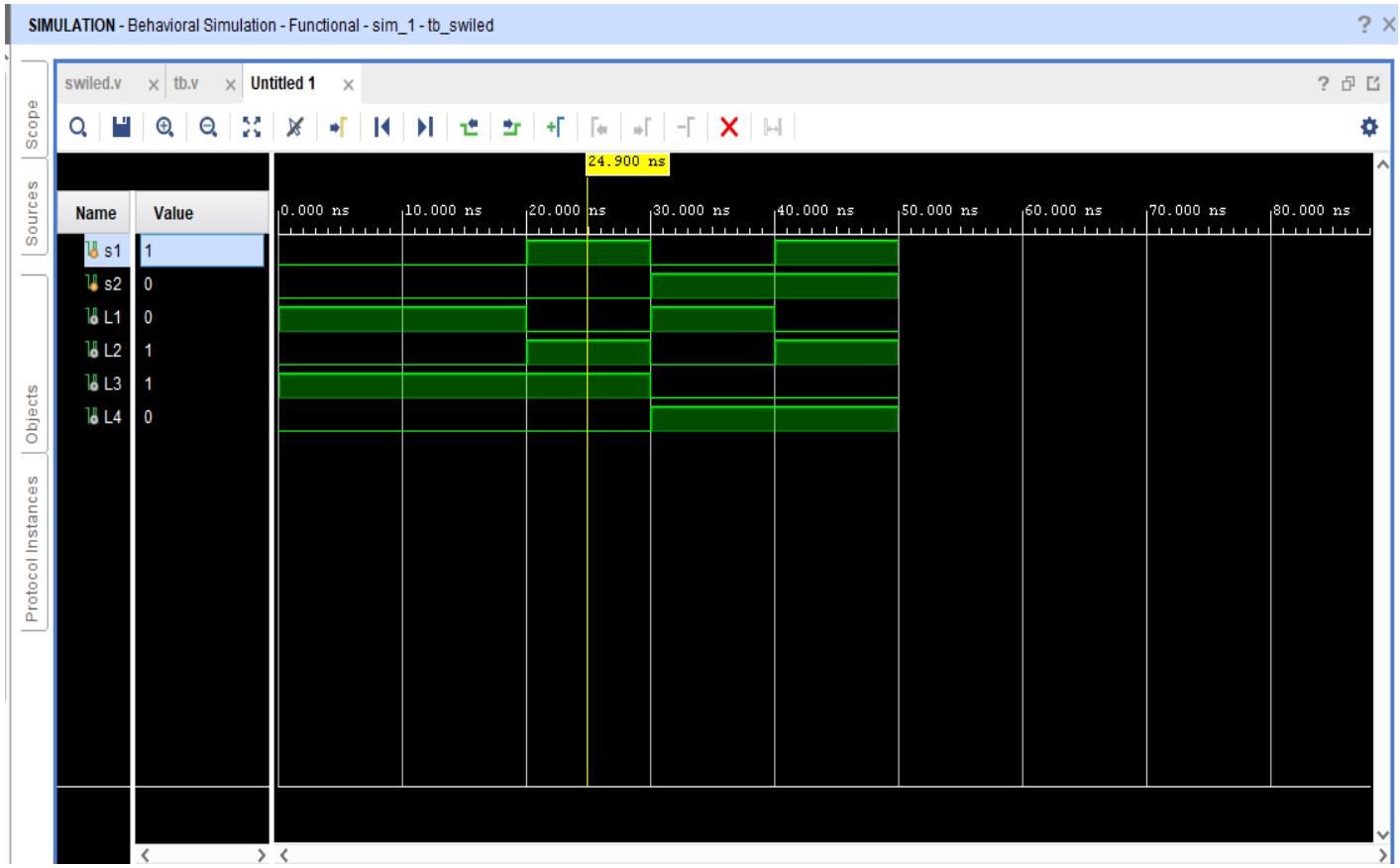
L4 gets turns ON (since S2=1)



3) For S1=1, S2=0;

L2 gets turns ON (since S1=1)

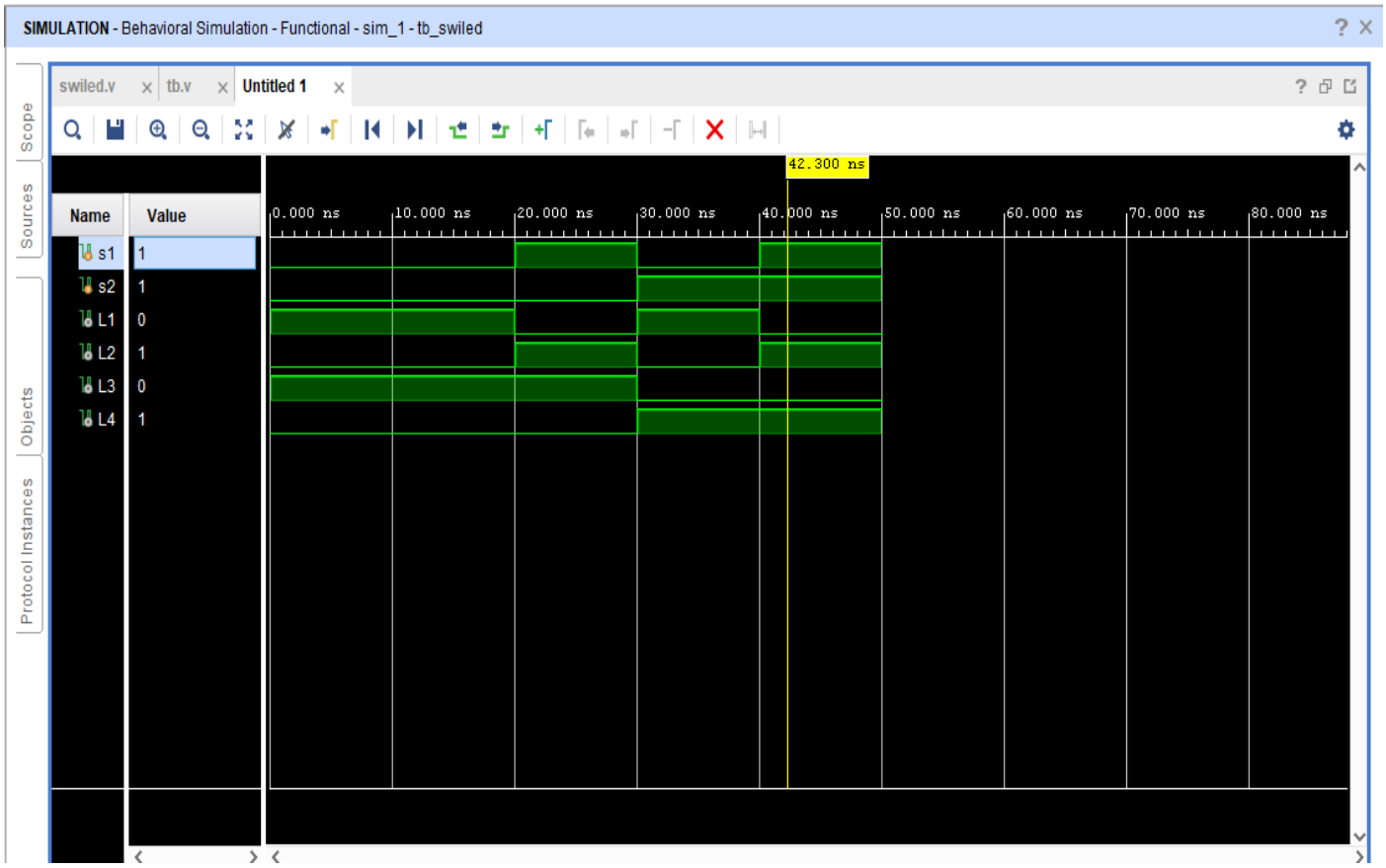
L3 gets turns ON



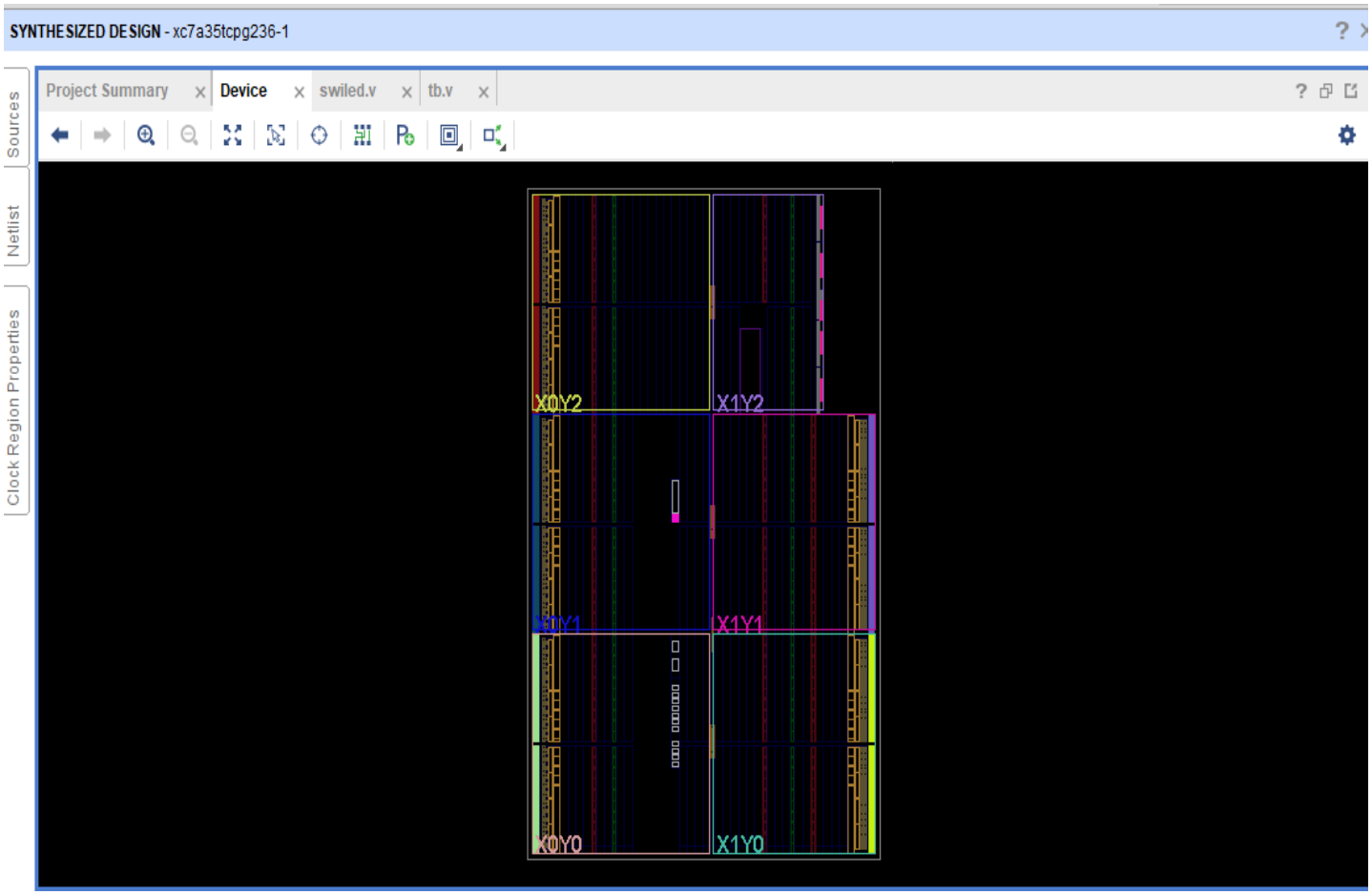
4) For S1=1, S2=1;

L2 gets turns ON (since S1=1)

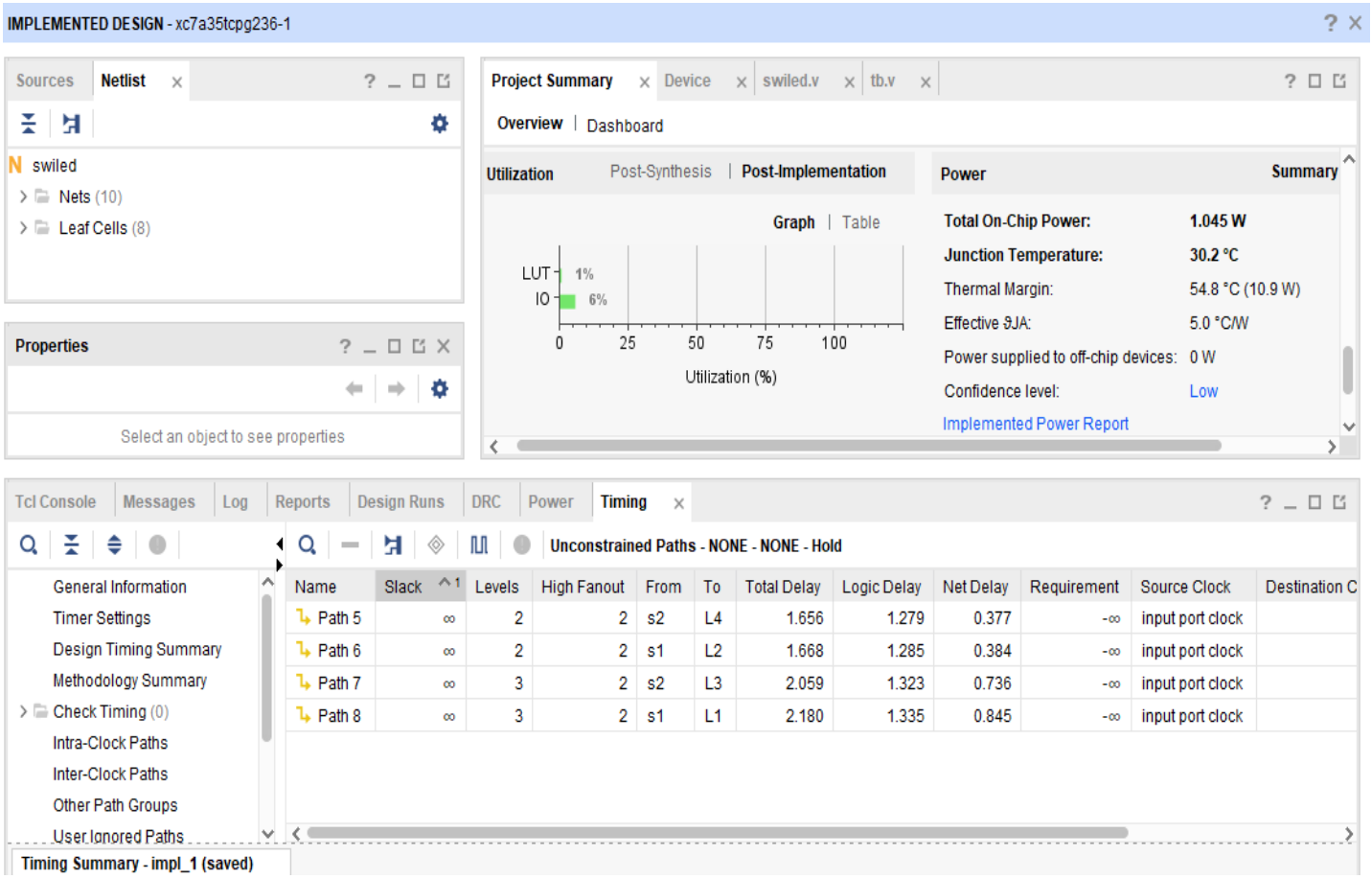
L4 gets turns ON



Synthesis Layout:



Power and Utilization:



Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination C
Path 5	∞	2	2	s2	L4	1.656	1.279	0.377	-∞	input port clock	
Path 6	∞	2	2	s1	L2	1.668	1.285	0.384	-∞	input port clock	
Path 7	∞	3	2	s2	L3	2.059	1.323	0.736	-∞	input port clock	
Path 8	∞	3	2	s1	L1	2.180	1.335	0.845	-∞	input port clock	

Schematic View: