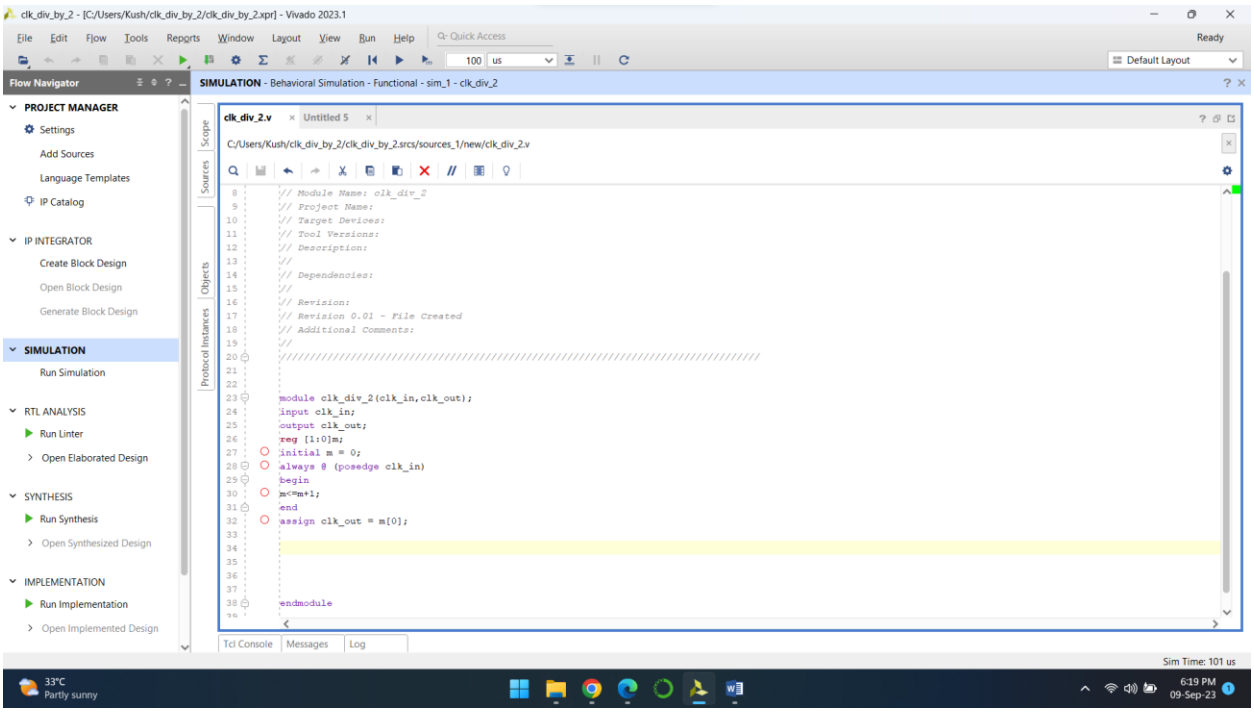
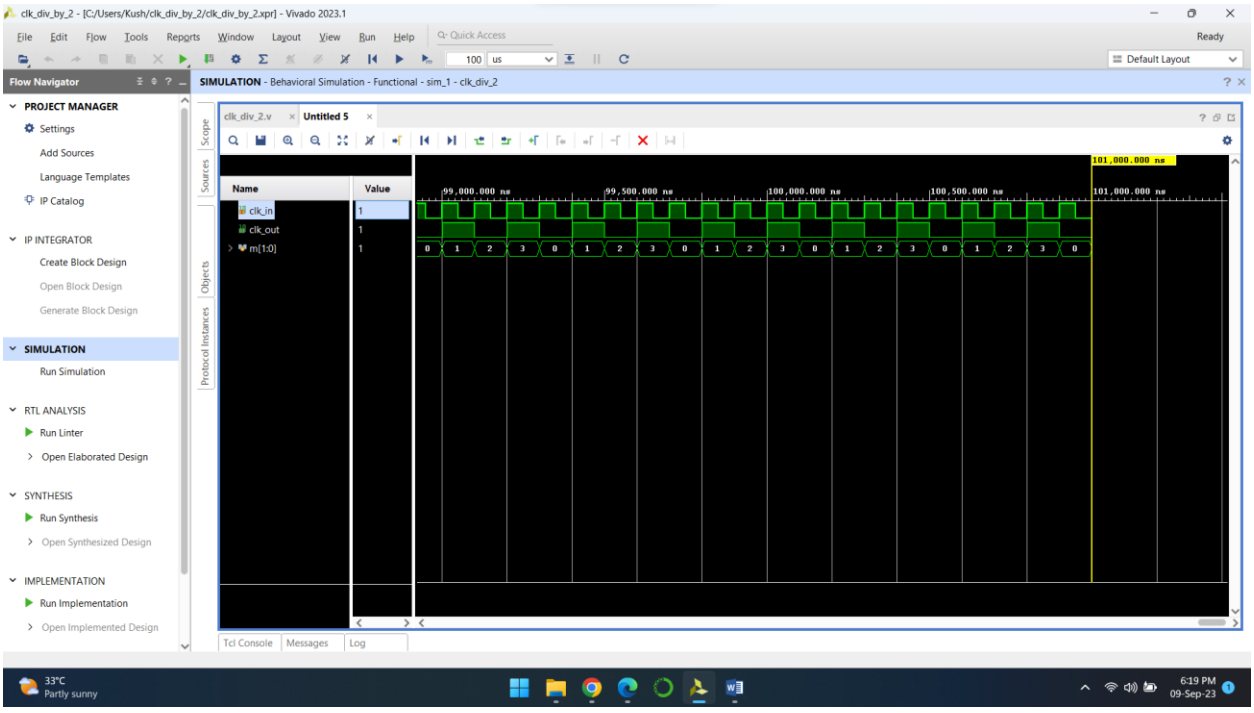
1. CLOCK DIVIDER.

CLOCK DIVIDER BY 2:

CODE:




WAVEFORM:



CLOCK DIVIDER 4:

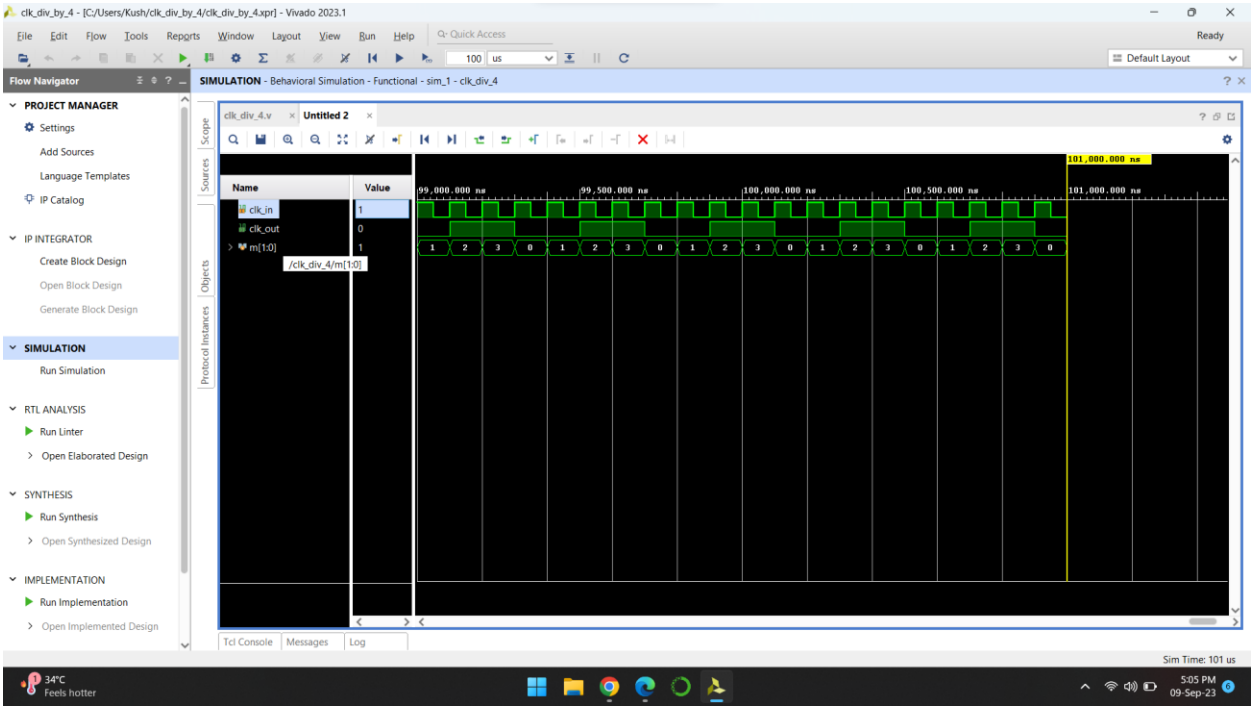
CODE:

clk_div_4.v | Untitled 2

C:/Users/Kush/clk_div_by_4/clk_div_by_4.srcs/sources_1/new/clk_div_4.v

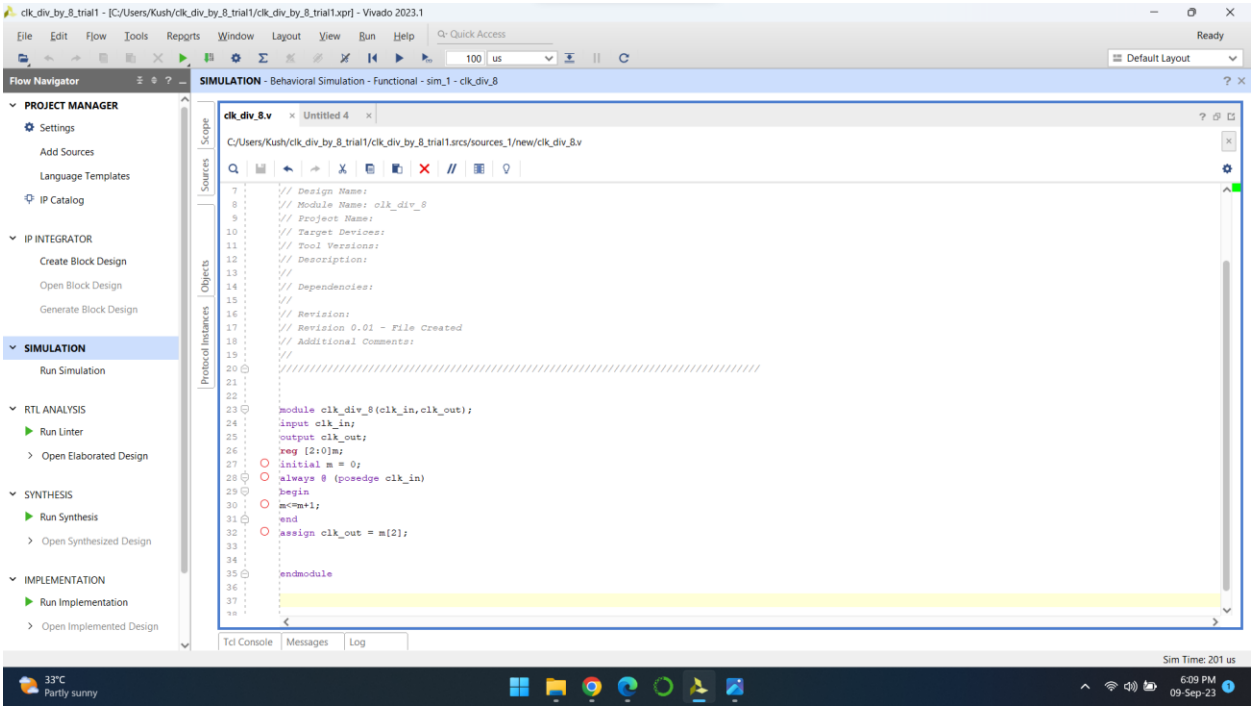
```
5   //
6   // Create Date: 09/09/2023 04:55:35 PM
7   // Design Name:
8   // Module Name: clk_div_4
9   // Project Name:
10  // Target Devices:
11  // Tool Versions:
12  // Description:
13  //
14  // Dependencies:
15  //
16  // Revision:
17  // Revision 0.01 - File Created
18  // Additional Comments:
19  //
20  //////////////////////////////////////////////////////////////////////////////////


21
22
23  module clk_div_4(clk_in,clk_out);
24  input clk_in;
25  output clk_out;
26  reg [1:0]m;
27  initial m = 0;
28  always @ (posedge clk_in)
29  begin
30  m<=m+1;
31  end
32  assign clk_out = m[1];
33
34
35  endmodule
```

WAVEFORM:

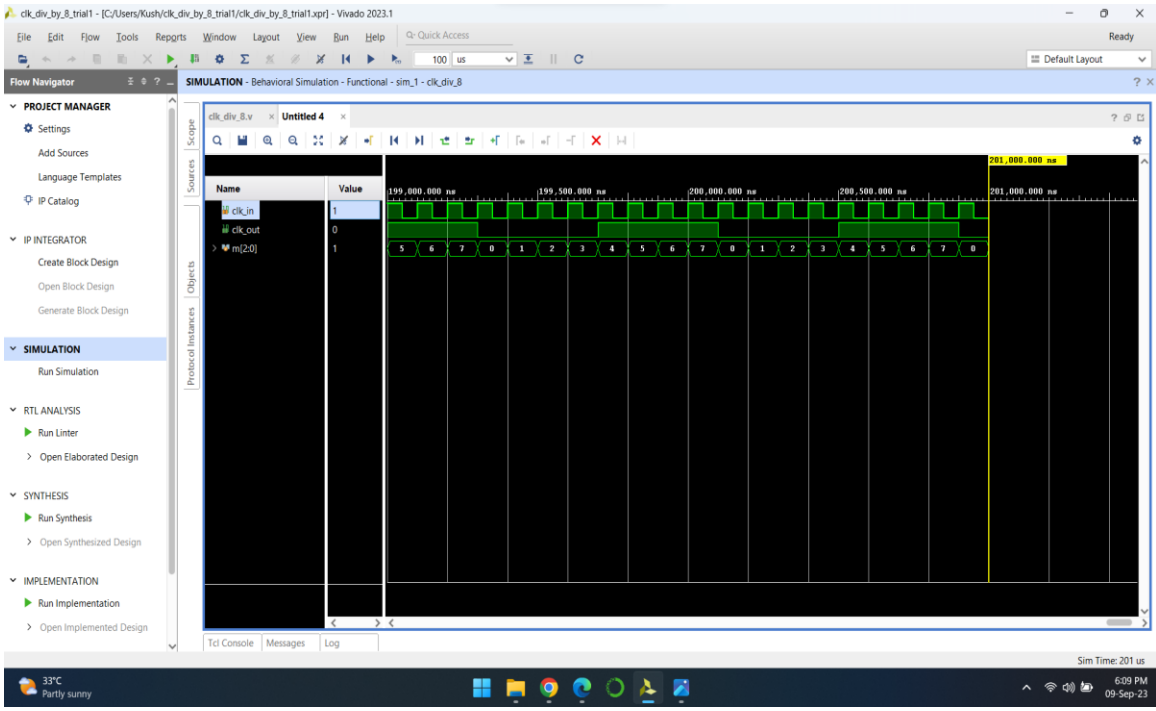


CLOCK DIVIDER BY 8:

CODE:

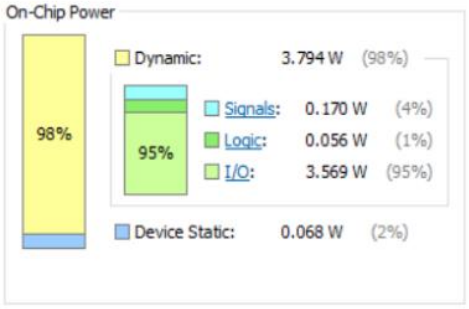



WAVEFORM:



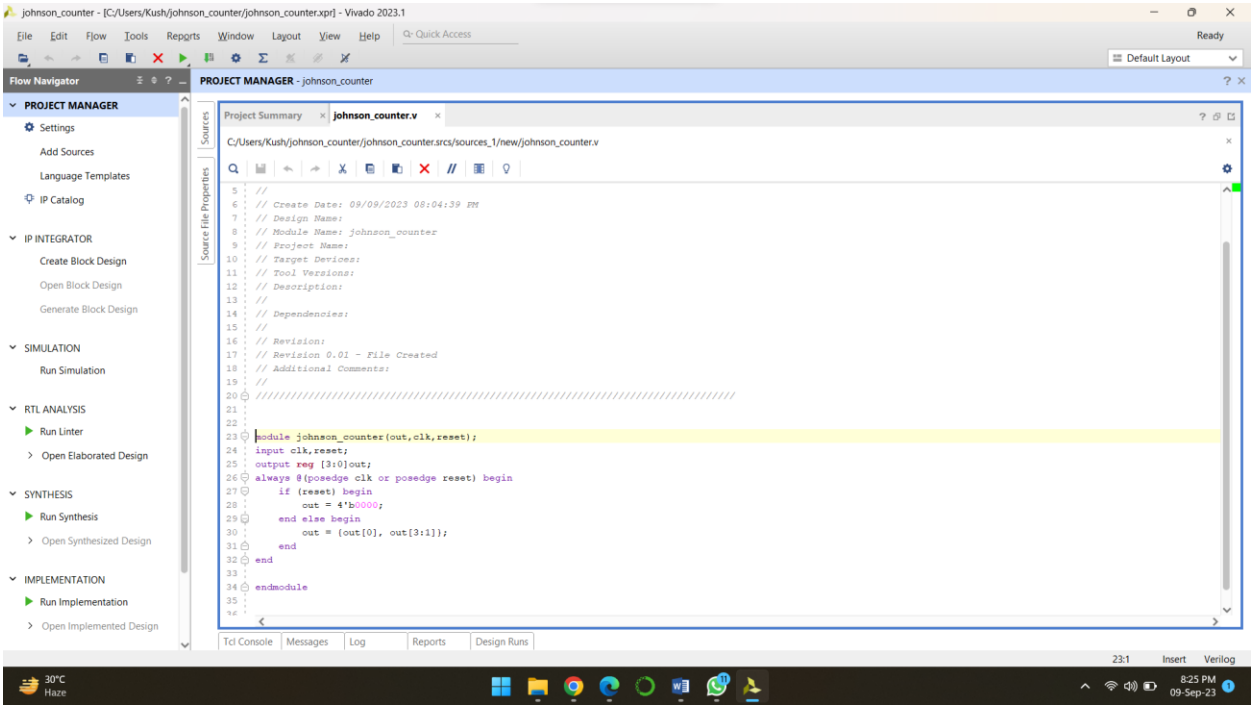
Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

| | |
|---|---|
| **Total On-Chip Power:** | **3.862 W** |
| **Junction Temperature:** | **43.5 °C** |
| Thermal Margin: | 56.5 °C (11.8 W) |
| Effective ϑJA: | 4.8 °C/W |
| Power supplied to off-chip devices: | 0 W |
| Confidence level: | Low |

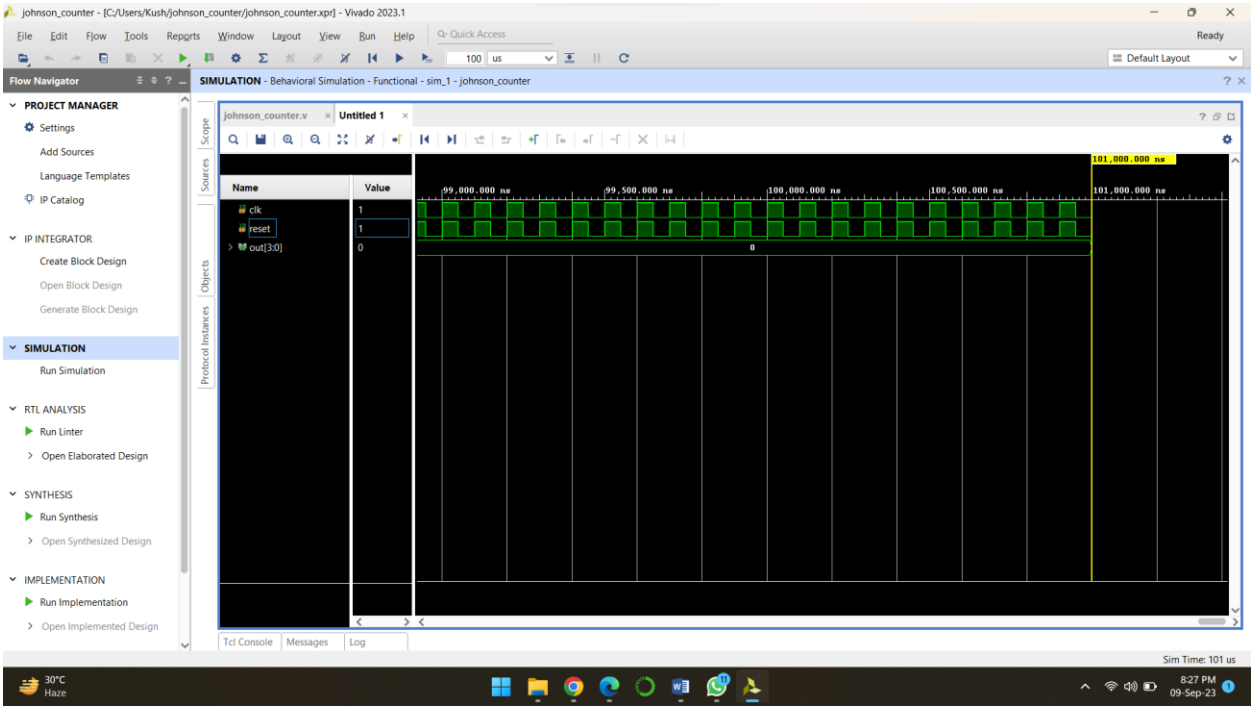



2. JOHNSON COUNTER:

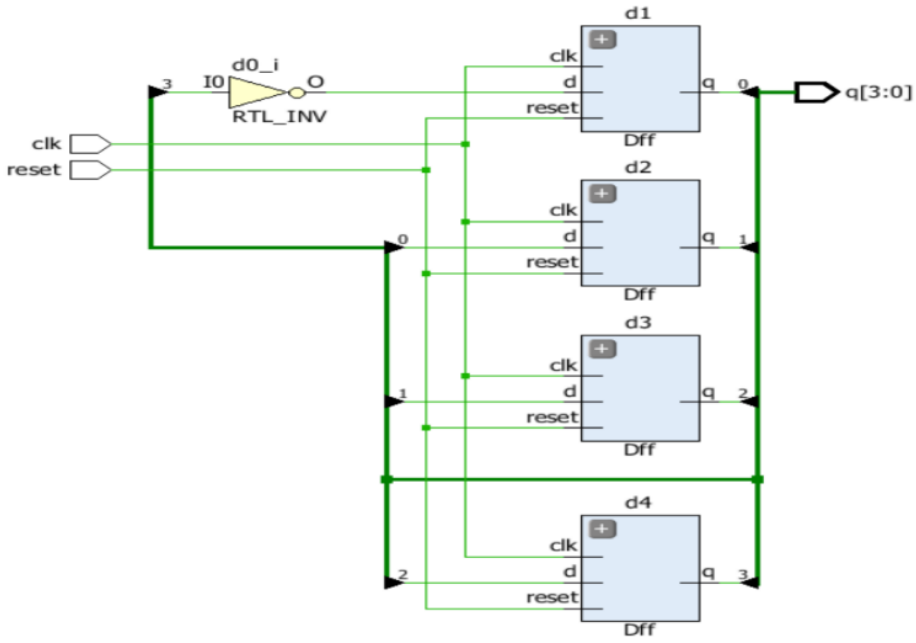
CODE:



WAVEFORM:



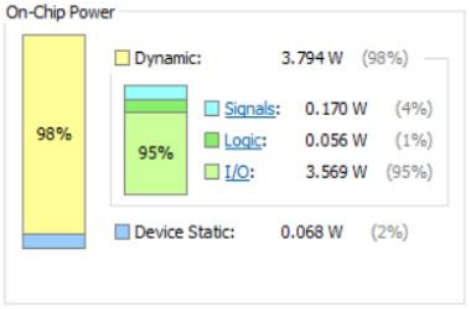

RTL REPORT:




POWER REPORT:

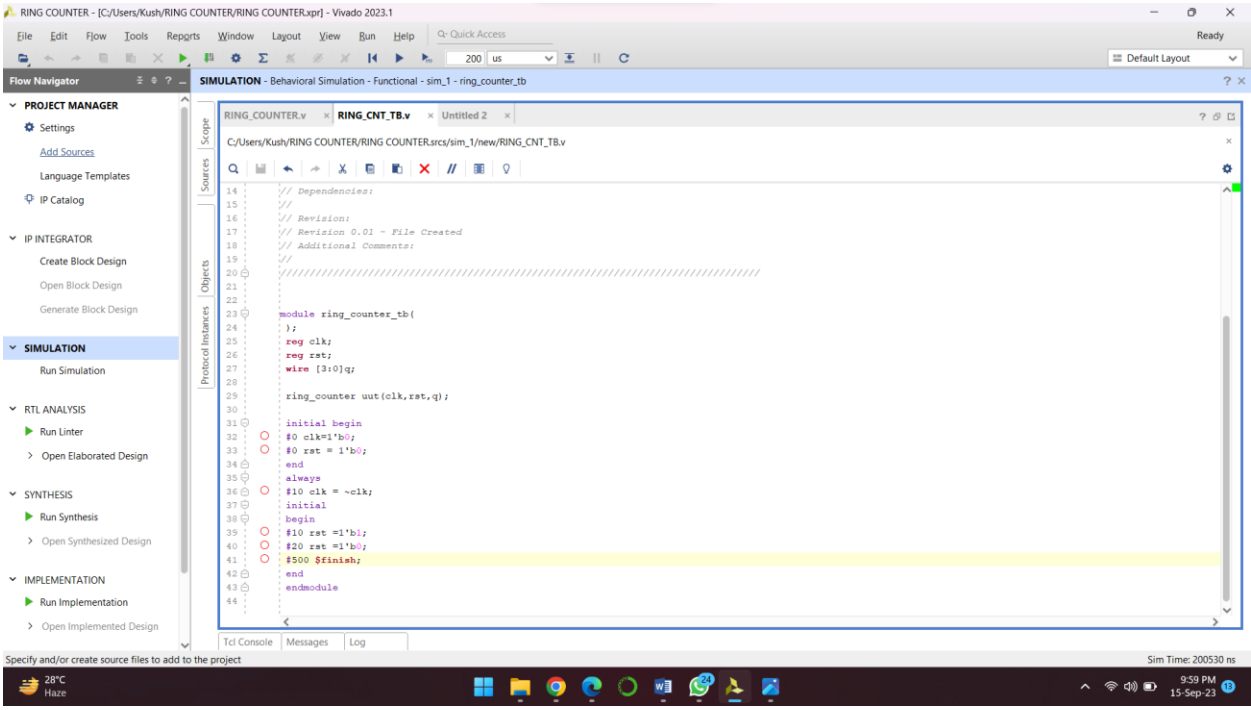
Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

| | |
|---|---|
| **Total On-Chip Power:** | **3.862 W** |
| **Junction Temperature:** | **43.5 °C** |
| Thermal Margin: | 56.5 °C (11.8 W) |
| Effective ϑJA: | 4.8 °C/W |
| Power supplied to off-chip devices: | 0 W |
| Confidence level: | Low |

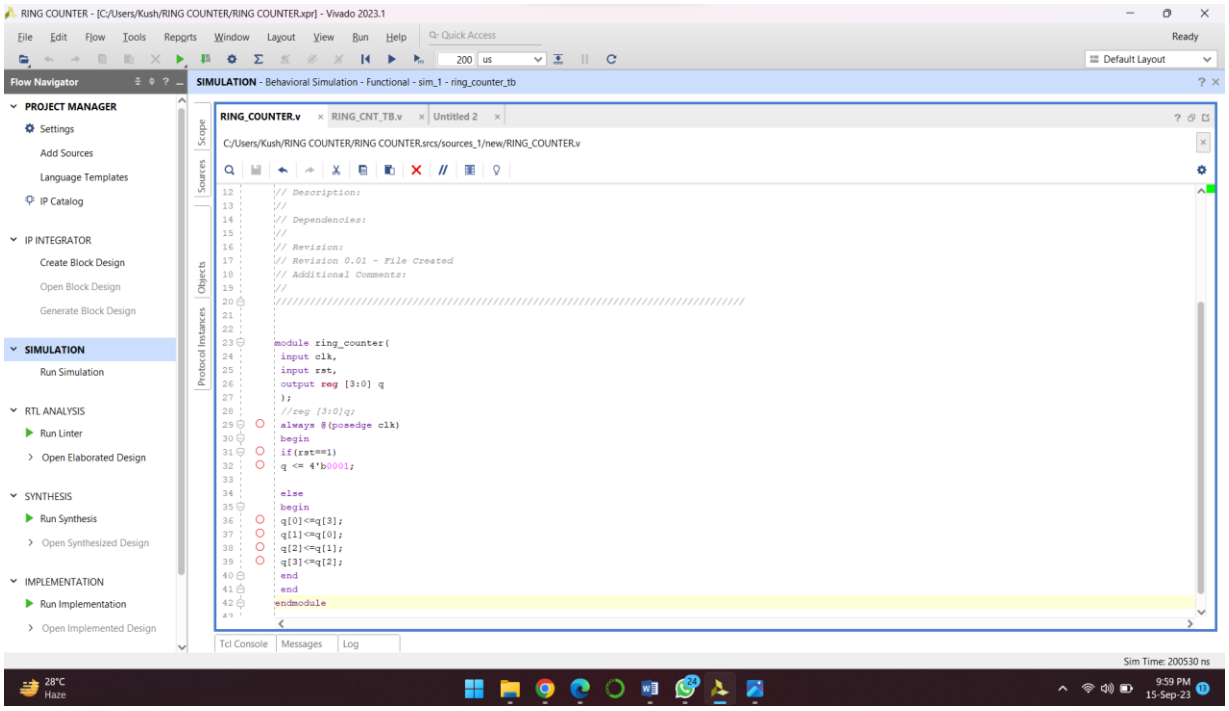



3. RING COUNTER:

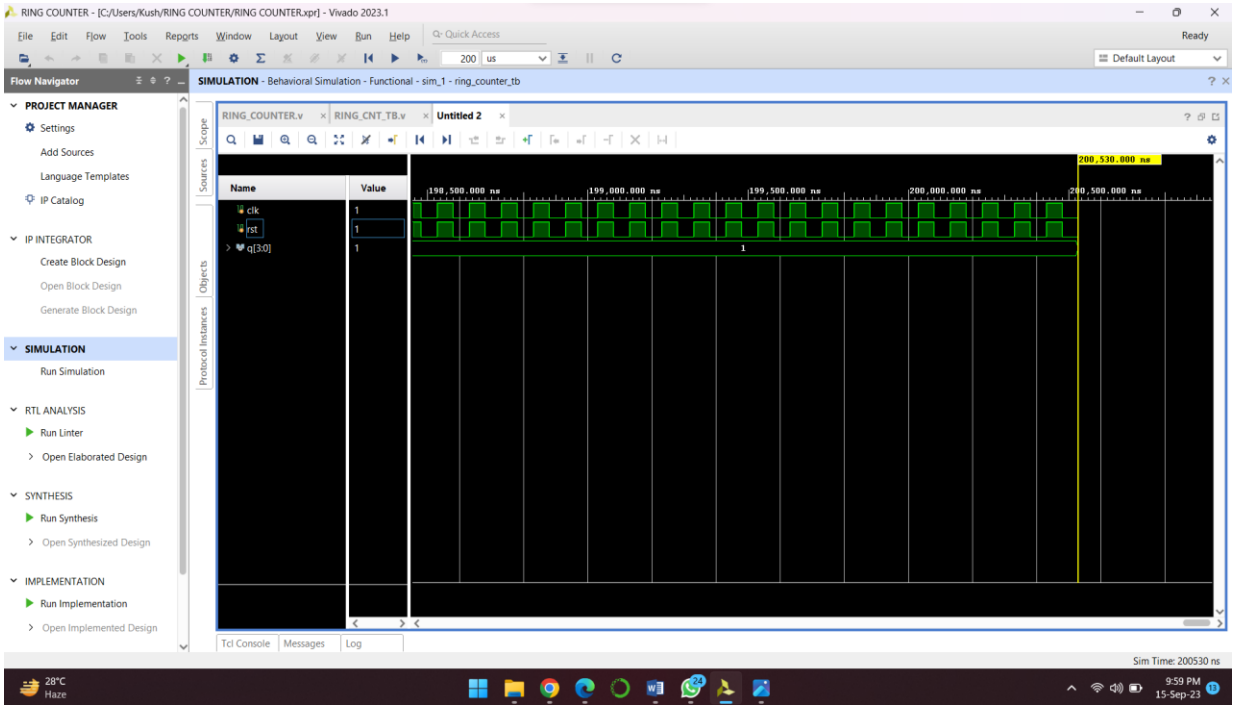
CODE:



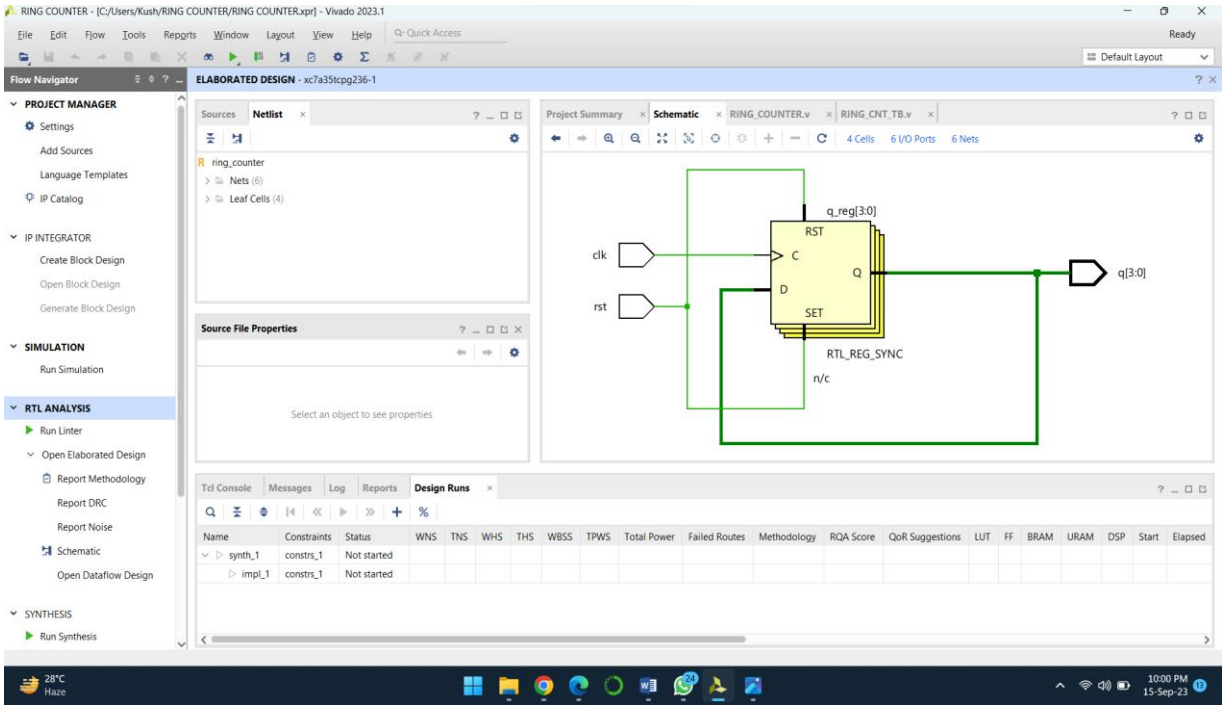

TESTBENCH:


RING COUNTER - [C:/Users/Kush/RING COUNTER/RING COUNTER.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - ring_counter_tb
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
RING_COUNTER.v
RING_CNT_TB.v
Untitled 2
C:/Users/Kush/RING COUNTER/RING COUNTER.srcs/sources_1/new/RING_COUNTER.v
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
module ring_counter(
input clk,
input rst,
output reg [3:0] q
);
//reg [3:0]q;
always @(posedge clk)
begin
if(rst==1)
q <= 4'b0001;
else
begin
q[0]<=q[3];
q[1]<=q[0];
q[2]<=q[1];
q[3]<=q[2];
end
end
endmodule
Tcl Console
Messages
Log
Sim Time: 200530 ns
28°C
Haze
9:59 PM
15-Sep-23

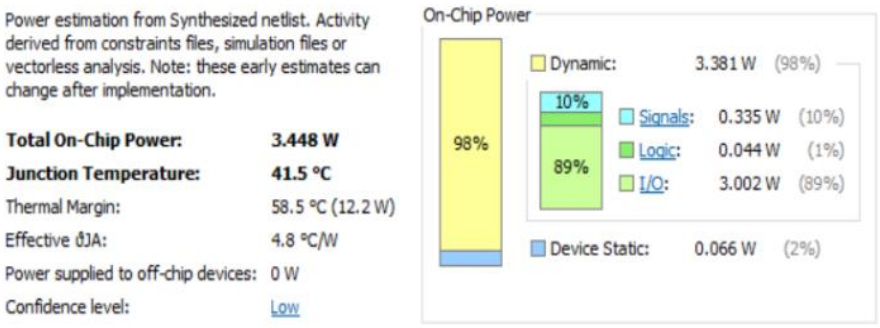
WAVEFORM:


RING COUNTER - [C:/Users/Kush/RING COUNTER/RING COUNTER.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - ring_counter_tb
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
RING_COUNTER.v
RING_CNT_TB.v
Untitled 2
Name
Value
clk
rst
q[3:0]
200,530.000 ns
199,500.000 ns
199,000.000 ns
199,500.000 ns
200,000.000 ns
200,500.000 ns
Tcl Console
Messages
Log
Sim Time: 200530 ns
28°C
Haze
9:59 PM
15-Sep-23

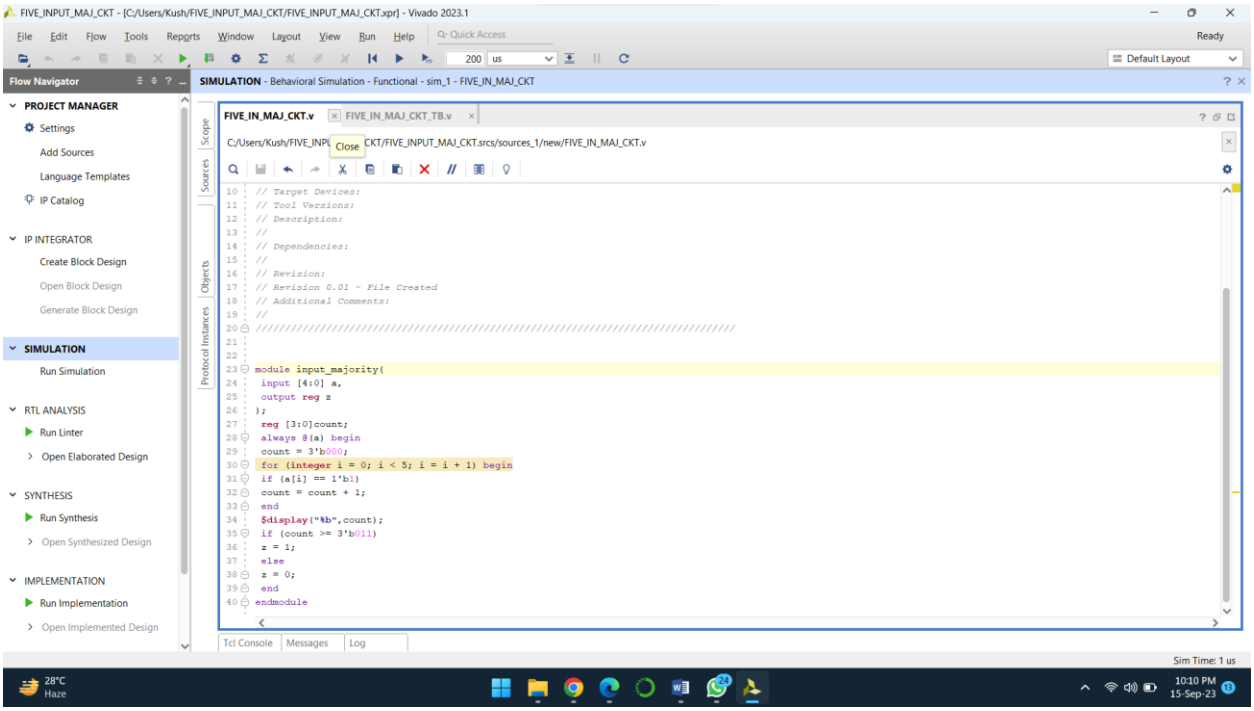
RTL SCHEMATIC:



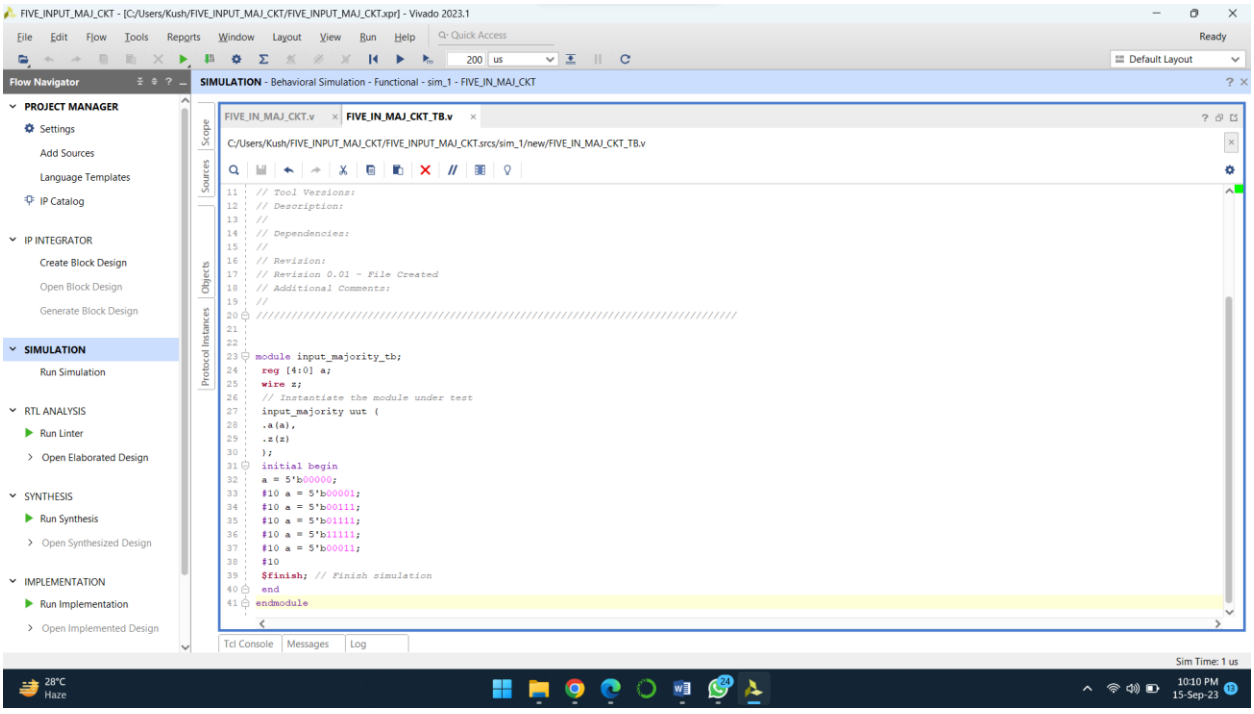

POWER REPORT:



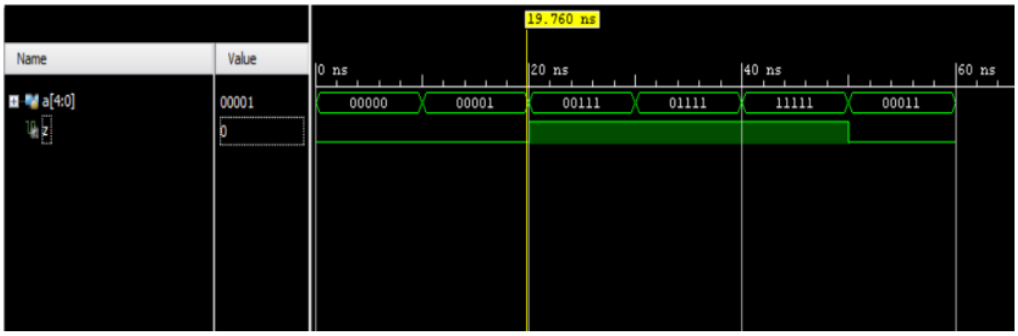

4. 5 INPUT MAJORITY CIRCUIT
   CODE:



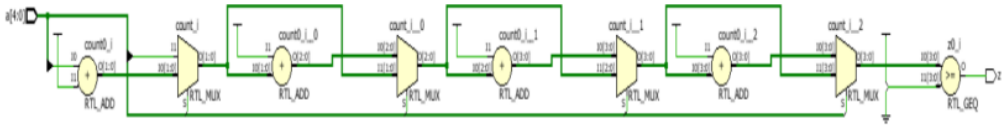

TESTBENCH:




WAVEFORM:



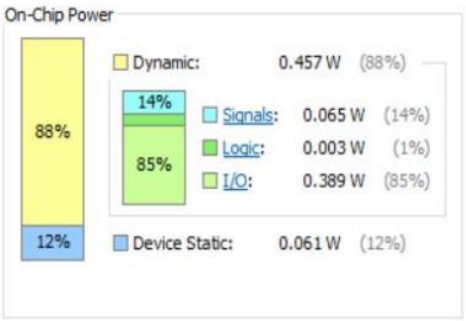

RTL SCHEMATIC:



POWER REPORT:

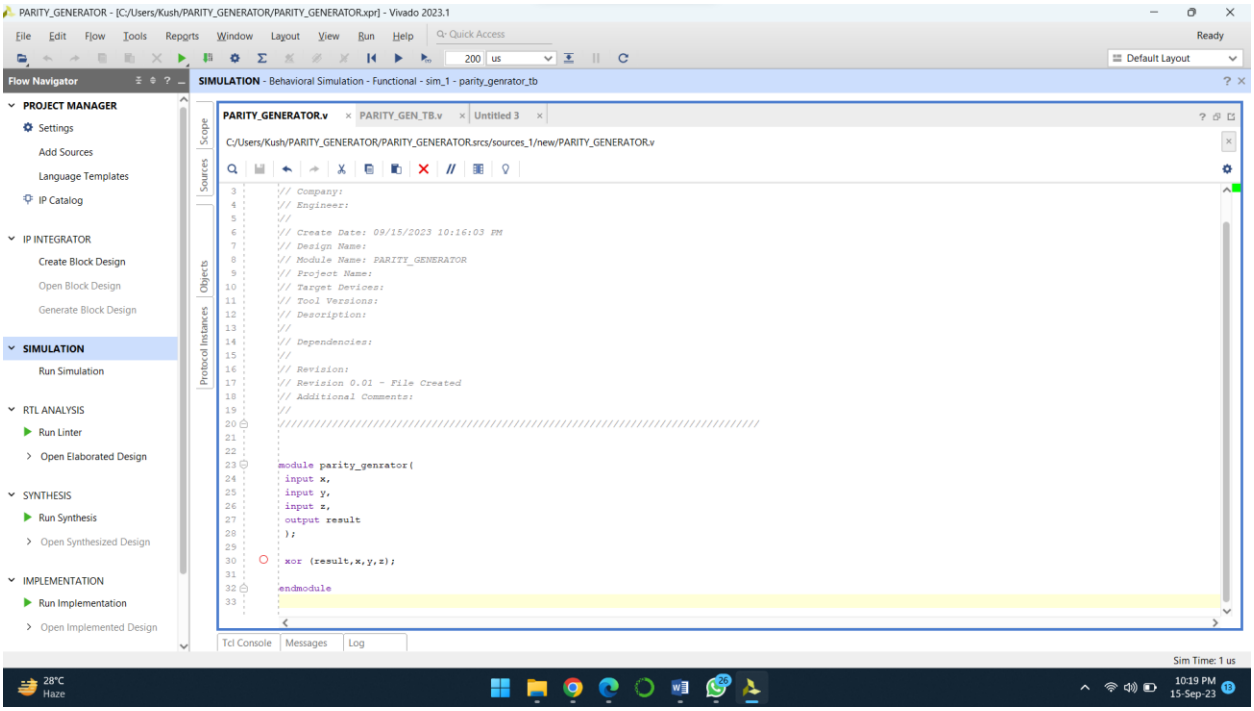
Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

| | |
|---|---|
| **Total On-Chip Power:** | **0.518 W** |
| **Junction Temperature:** | **27.5 °C** |
| Thermal Margin: | 72.5 °C (15.1 W) |
| Effective ϑJA: | 4.8 °C/W |
| Power supplied to off-chip devices: | 0 W |
| Confidence level: | Low |

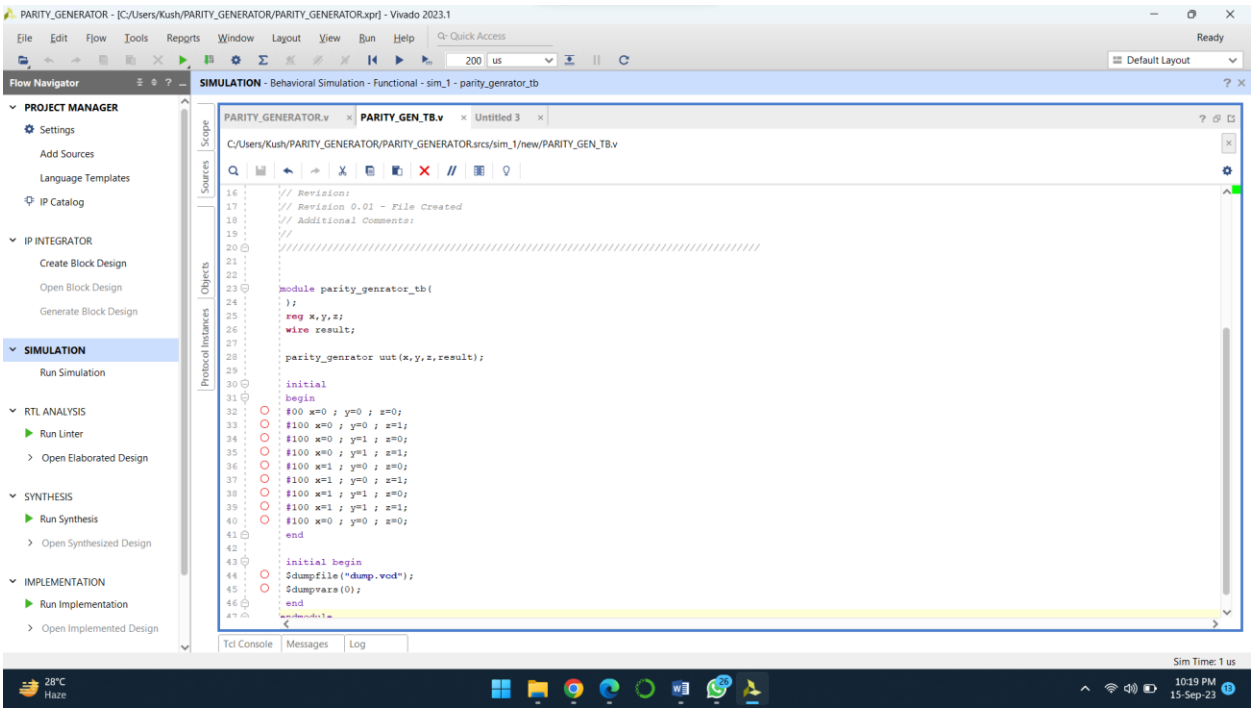


5. PARITY GENERATOR:

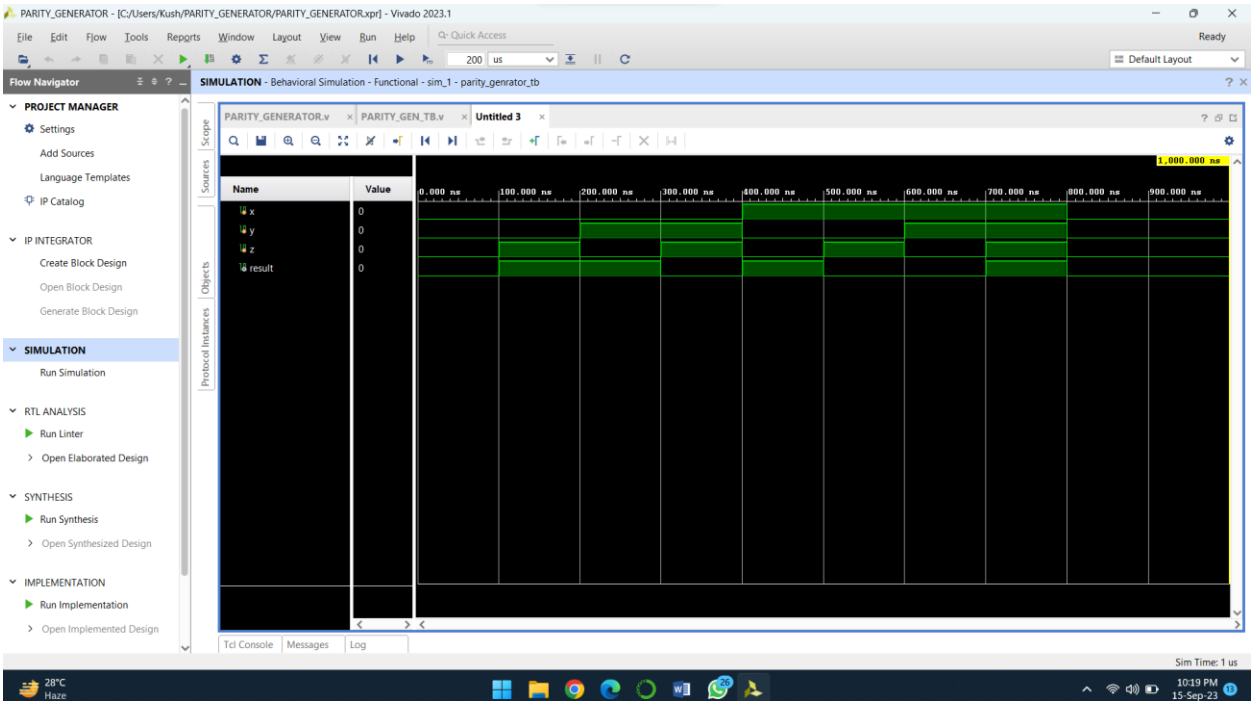
CODE:



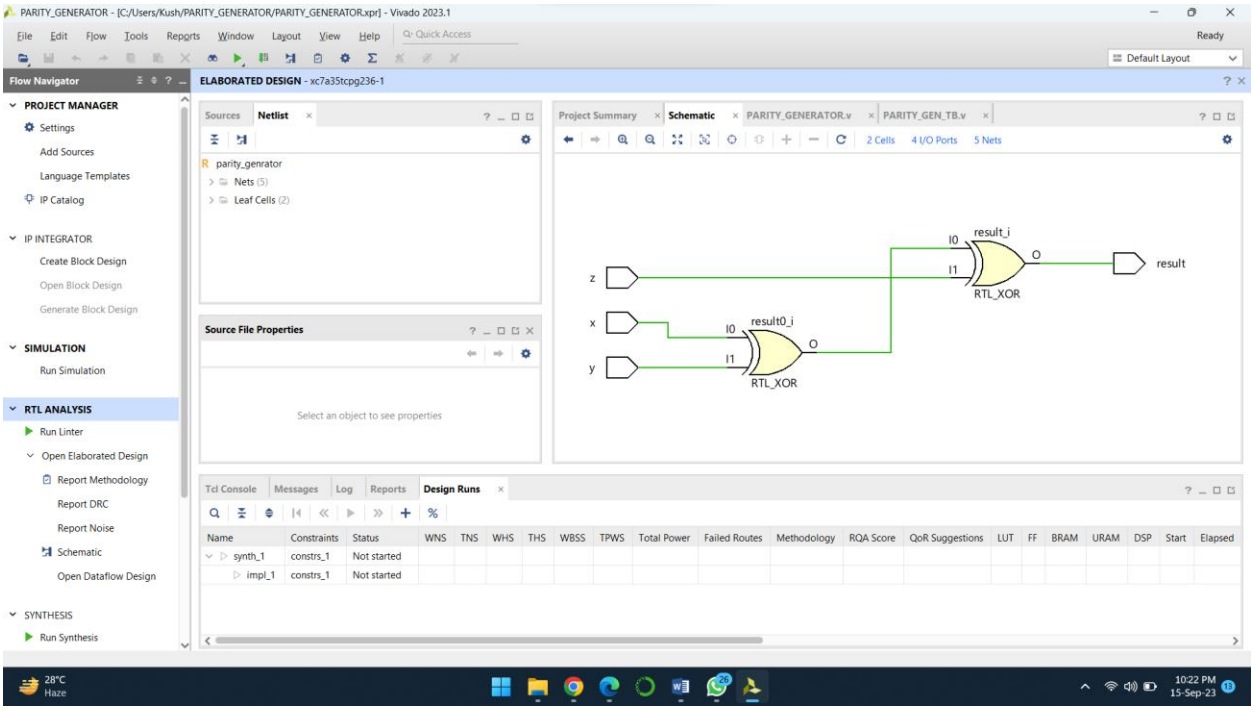

TESTBENCH:




WAVEFORM:



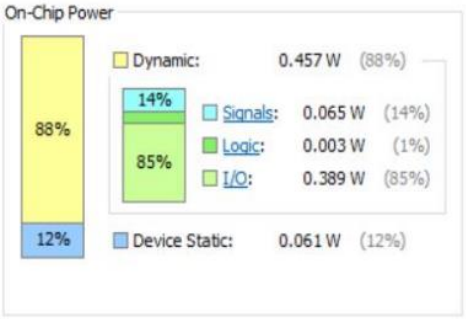

RTL SCHEMATIC:



POWER REPORT:

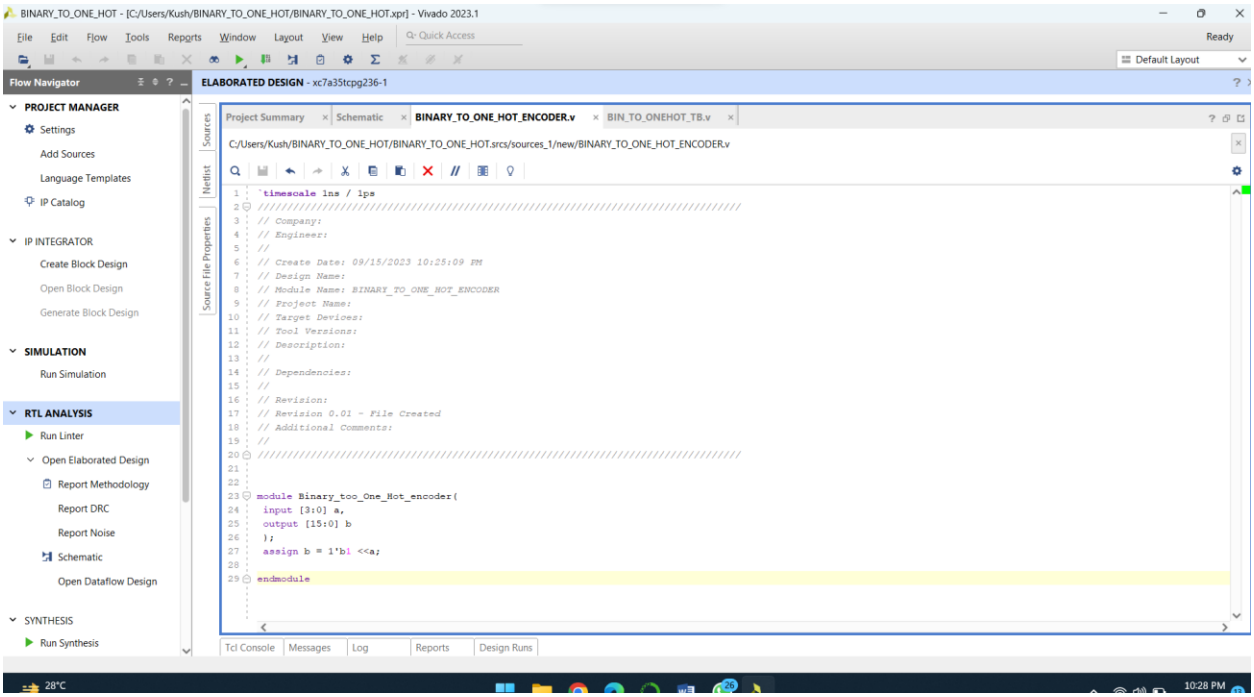
Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

| | |
|---|---|
| **Total On-Chip Power:** | **0.518 W** |
| **Junction Temperature:** | **27.5 °C** |
| Thermal Margin: | 72.5 °C (15.1 W) |
| Effective ϑJA: | 4.8 °C/W |
| Power supplied to off-chip devices: | 0 W |
| Confidence level: | Low |

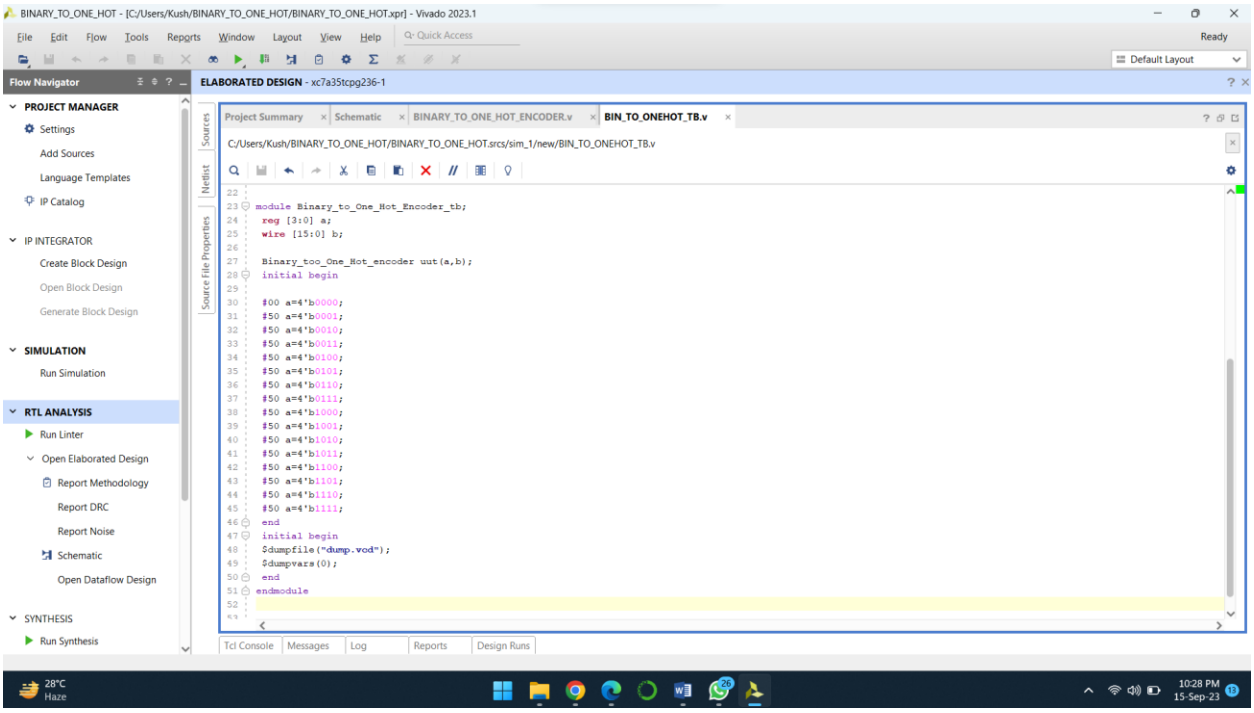



6. BINARY TO ONE HOT ENCODER:

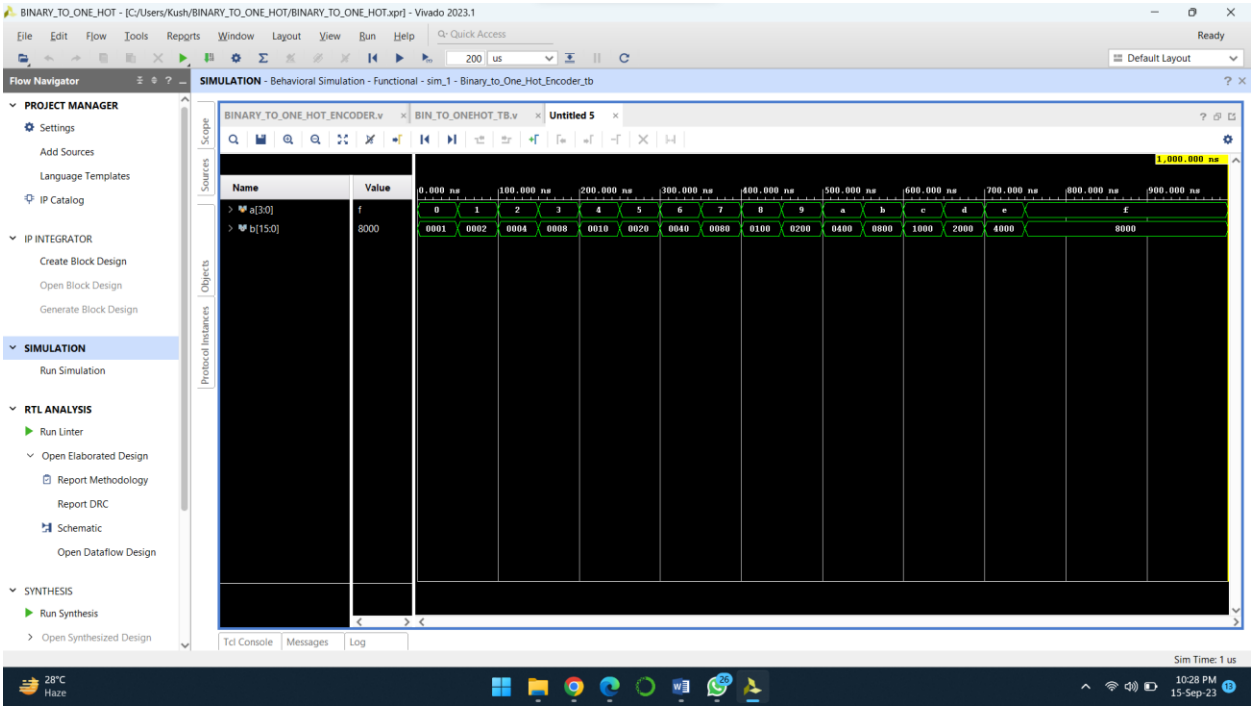
CODE:



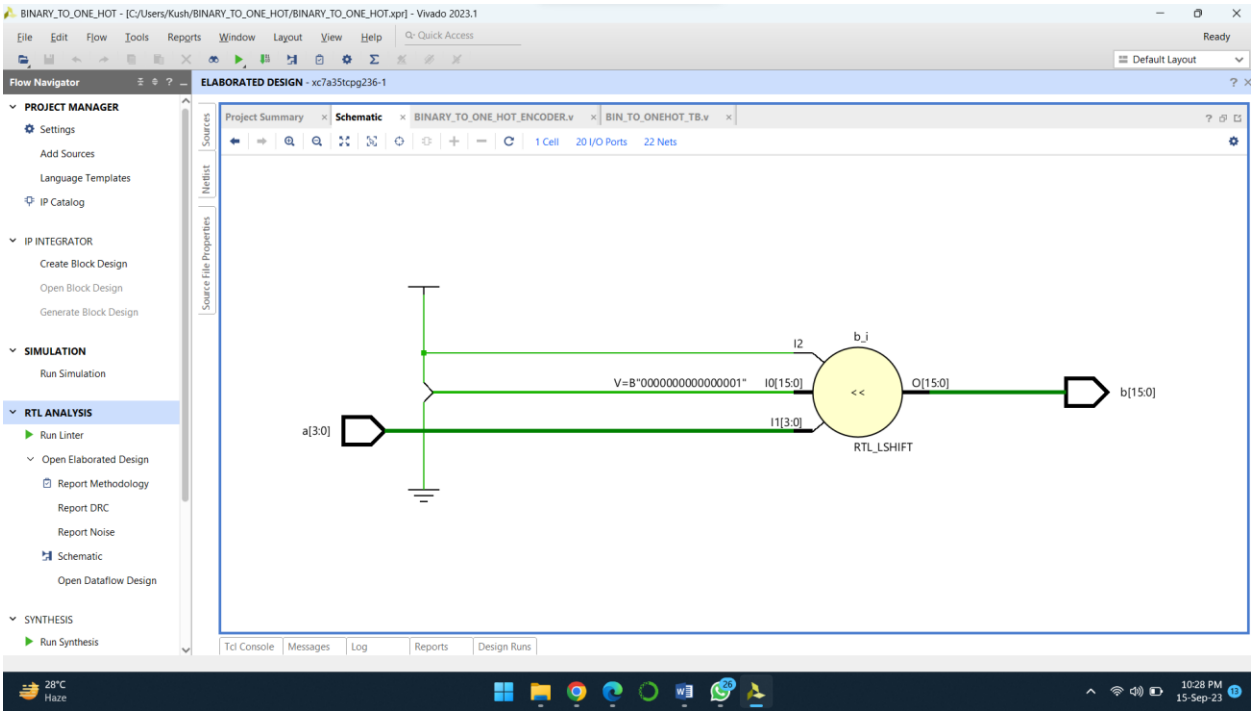

TESTBENCH:



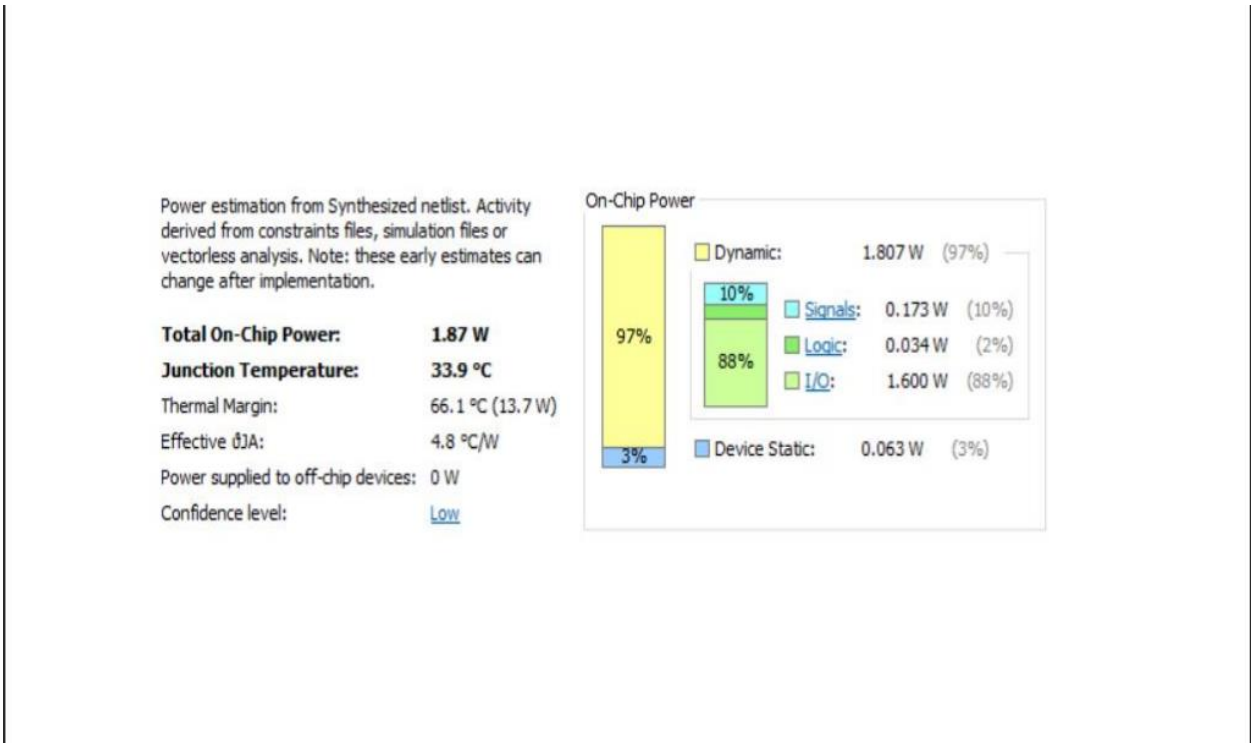

WAVEFORM:




RTL REPORT:

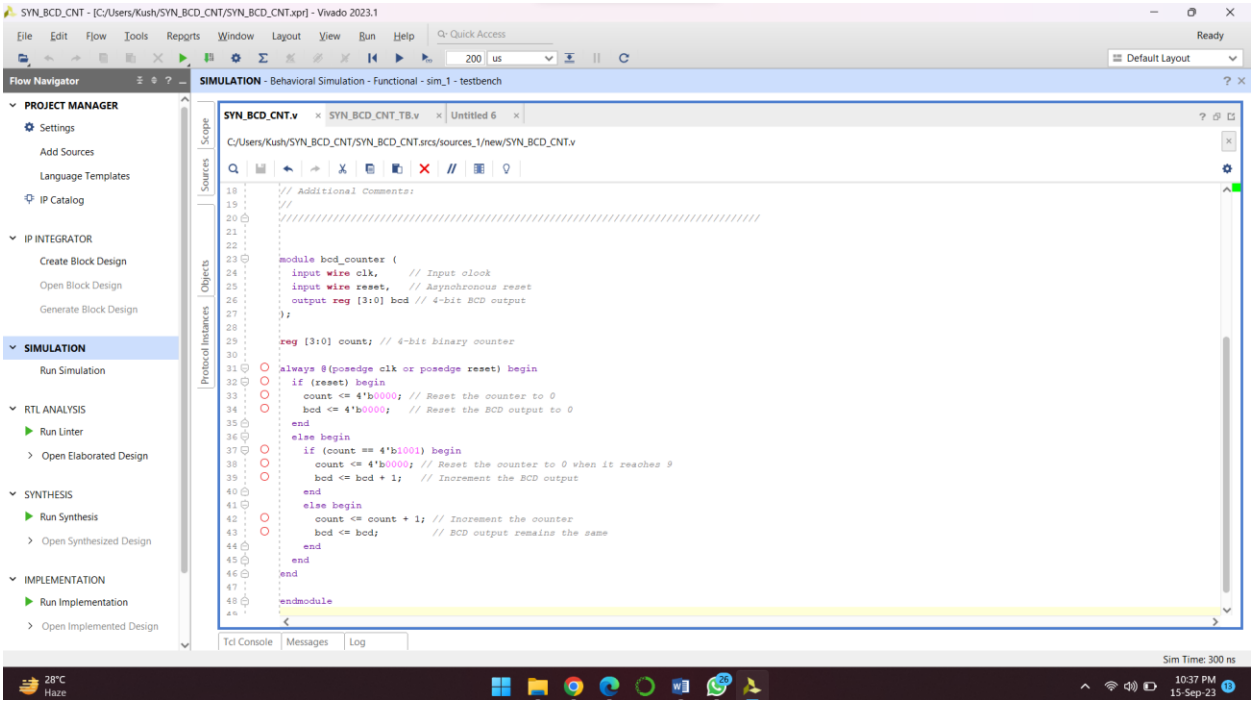



POWER REPORT:

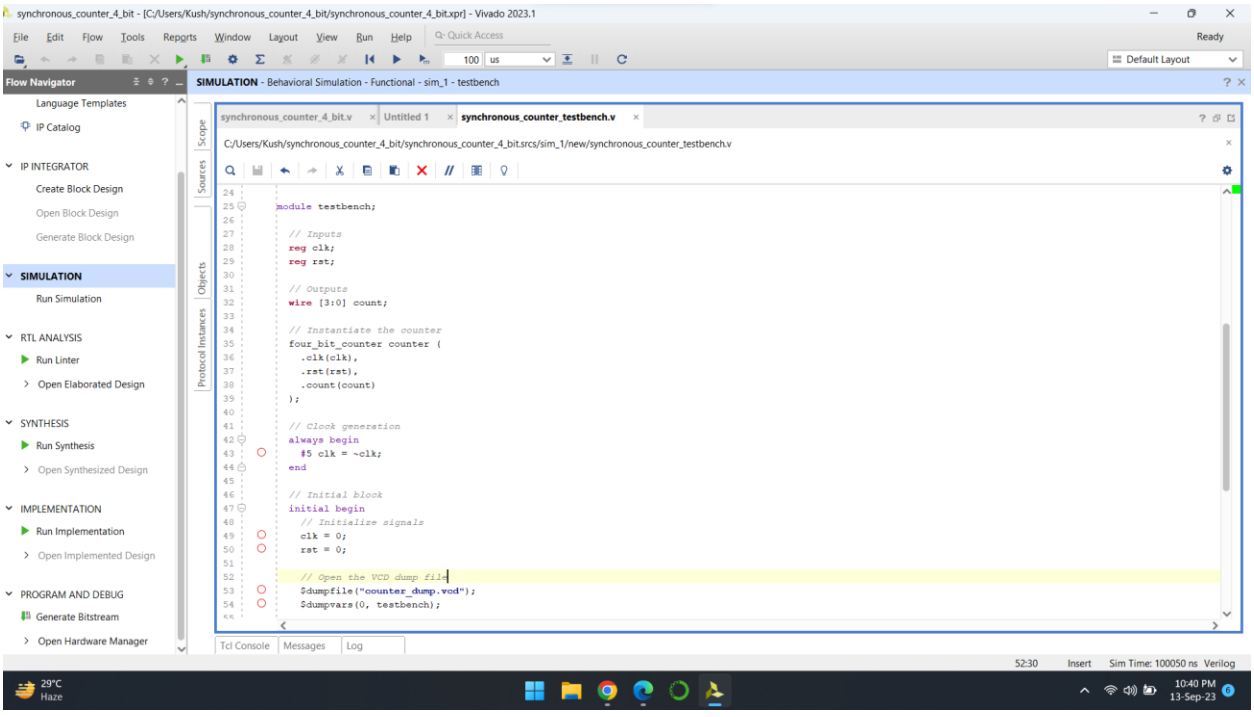


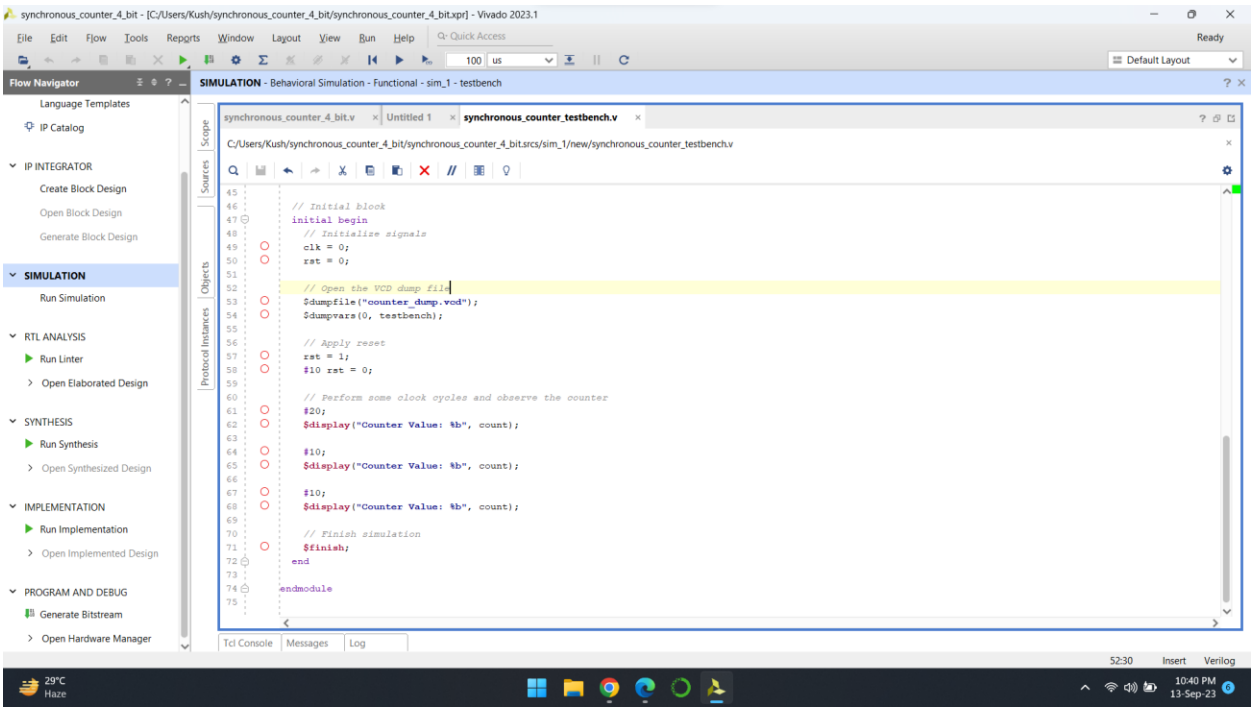
7. 4-Bit Synchronous BCD Counter:

CODE:

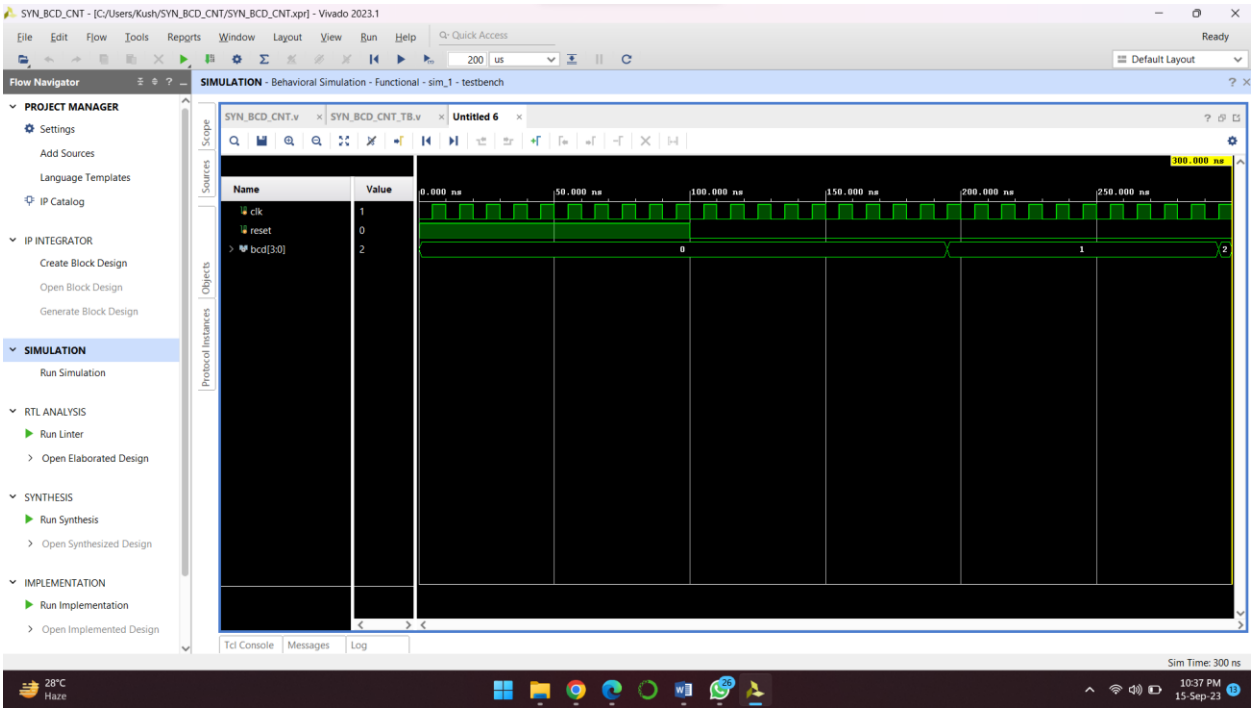



TESTBENCH:

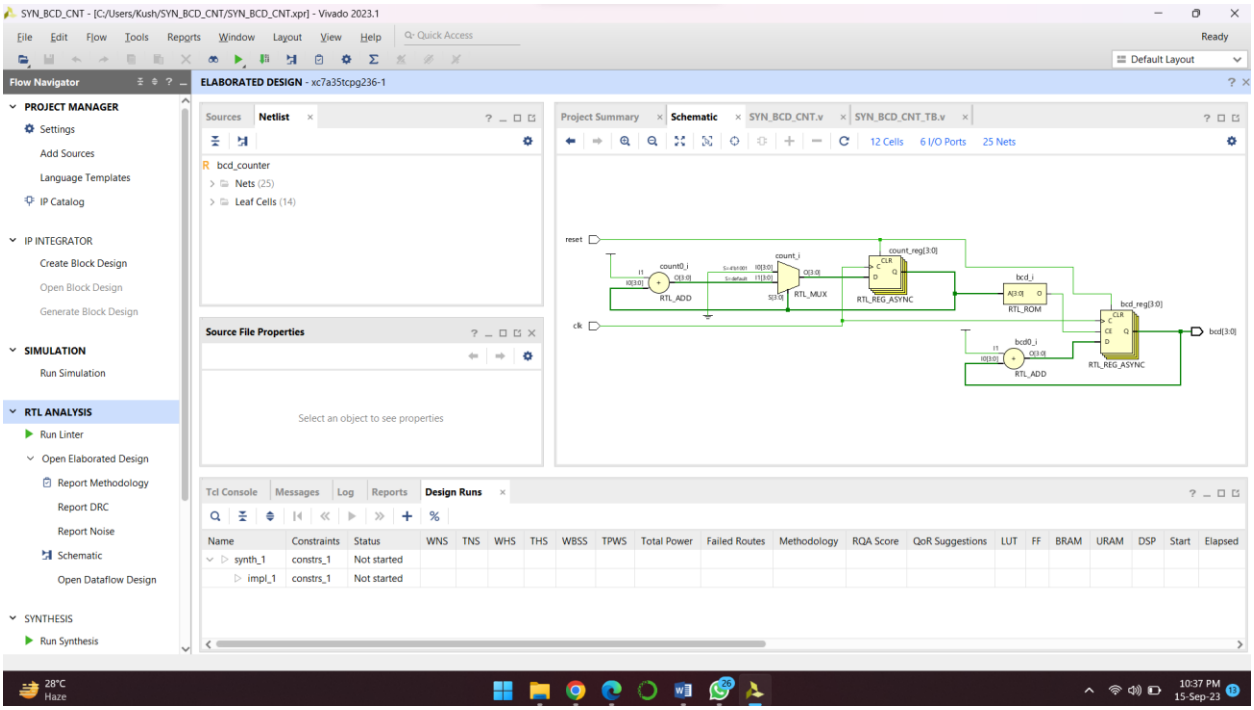



synchronous_counter_4_bit - [C:/Users/Kush/synchronous_counter_4_bit/synchronous_counter_4_bit.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
Open Hardware Manager
synchronous_counter_4_bit.v
Untitled 1
synchronous_counter_testbench.v
C:/Users/Kush/synchronous_counter_4_bit/synchronous_counter_4_bit.srcs/sim_1/new/synchronous_counter_testbench.v
// Initial block
initial begin
// Initialize signals
clk = 0;
rst = 0;
// Open the VCD dump file
$dumpfile("counter_dump.vcd");
$dumpvars(0, testbench);
// Apply reset
rst = 1;
#10 rst = 0;
// Perform some clock cycles and observe the counter
#20;
$display("Counter Value: %b", count);
#10;
$display("Counter Value: %b", count);
#10;
$display("Counter Value: %b", count);
// Finish simulation
$finish;
end
endmodule
Tcl Console
Messages
Log
52:30
Insert
Verilog
29°C
Haze
10:40 PM
13-Sep-23

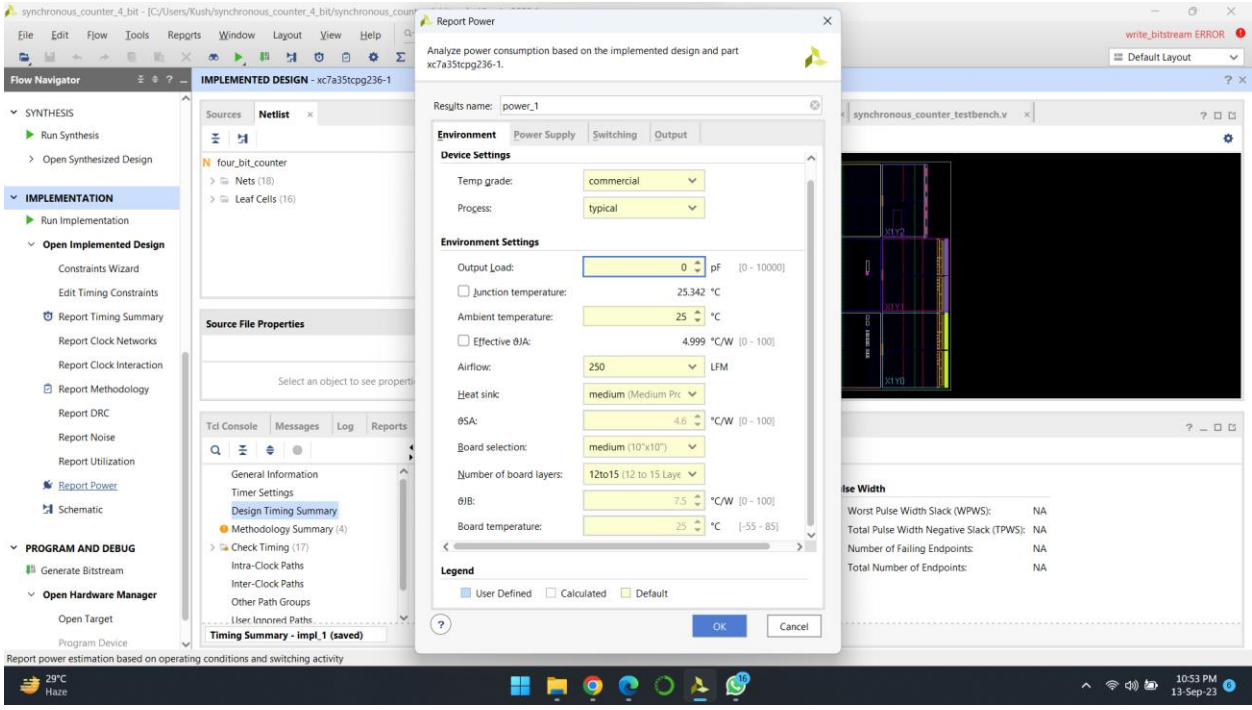
WAVEFORM:

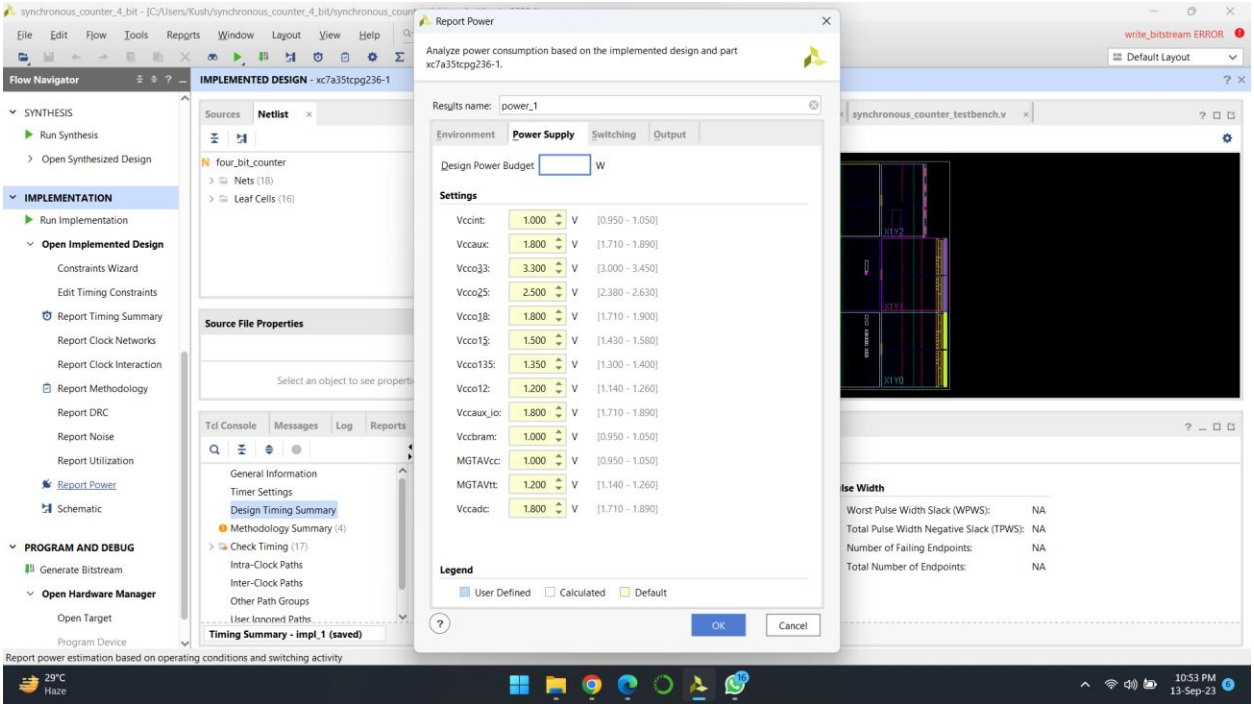

SYN_BCD_CNT - [C:/Users/Kush/SYN_BCD_CNT/SYN_BCD_CNT.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
SYN_BCD_CNT.v
SYN_BCD_CNT_TB.v
Untitled 6
Name
Value
clk
1
reset
0
bcd[3:0]
2
0.000 ns
50.000 ns
100.000 ns
150.000 ns
200.000 ns
250.000 ns
300.000 ns
Tcl Console
Messages
Log
Sim Time: 300 ns
28°C
Haze
10:37 PM
15-Sep-23

RTL SCHEMATIC:



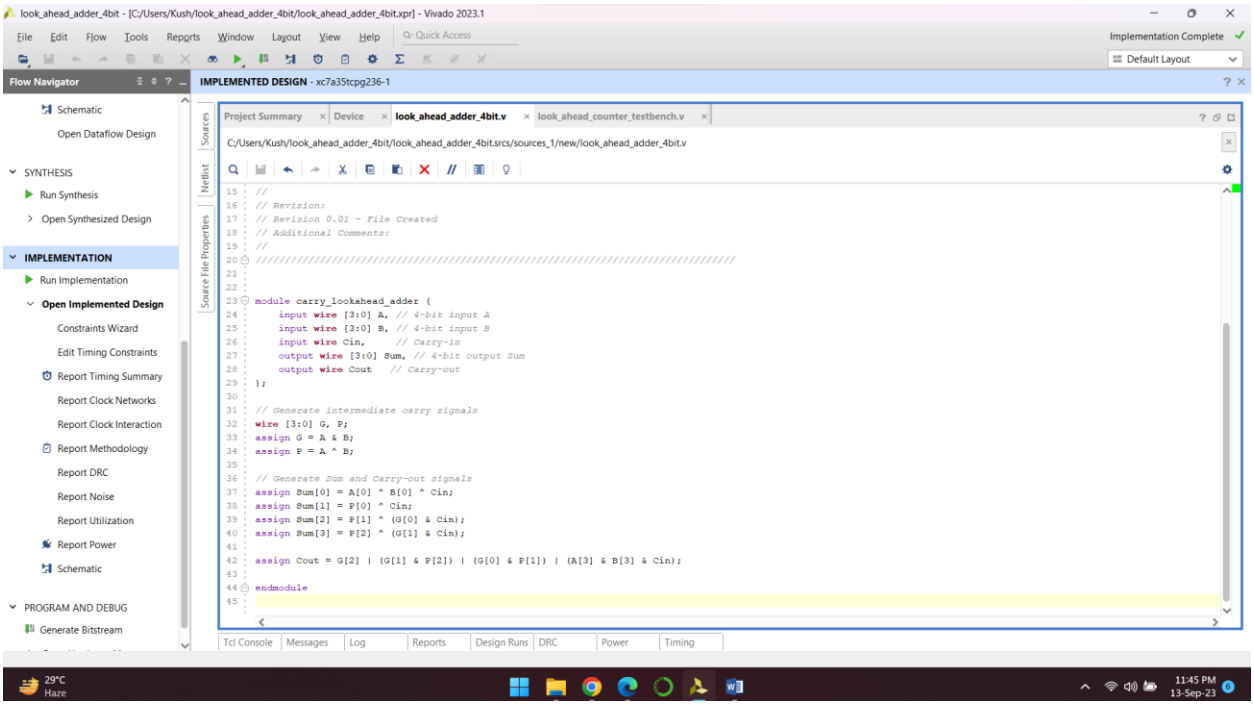

POWER REPORT:



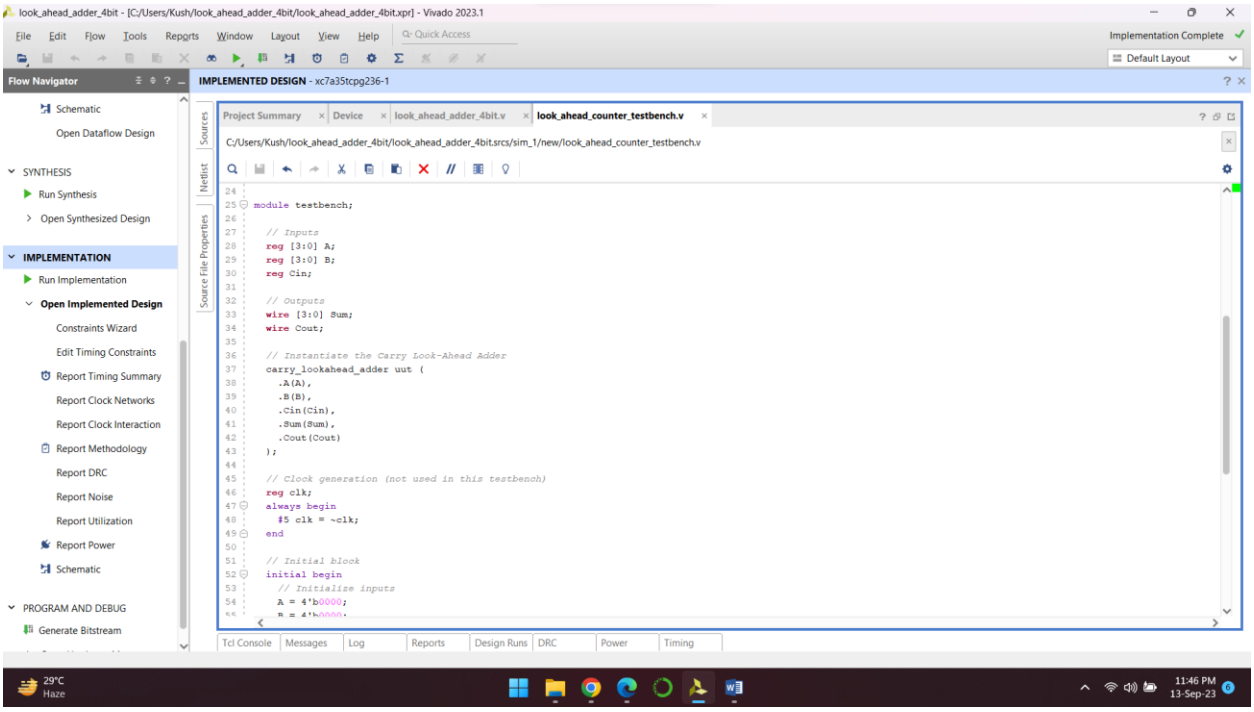


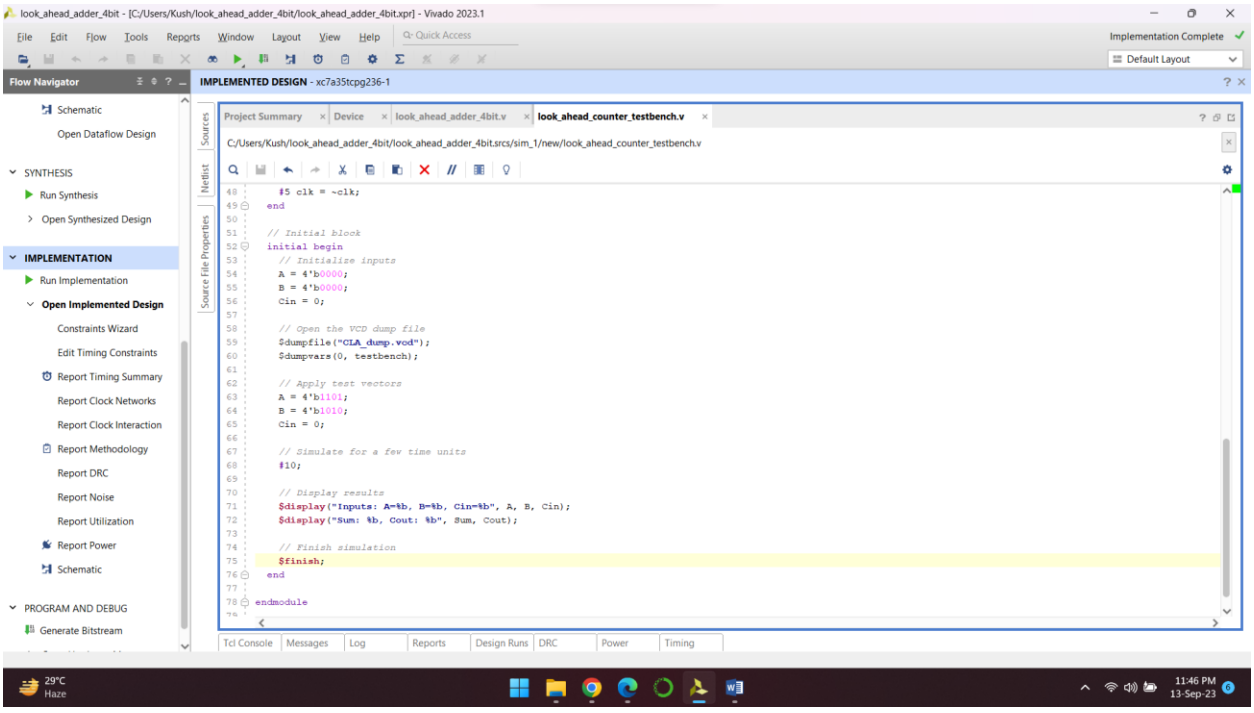

## 8. 4- BIT LOOK AHEAD ADDER:

CODE:

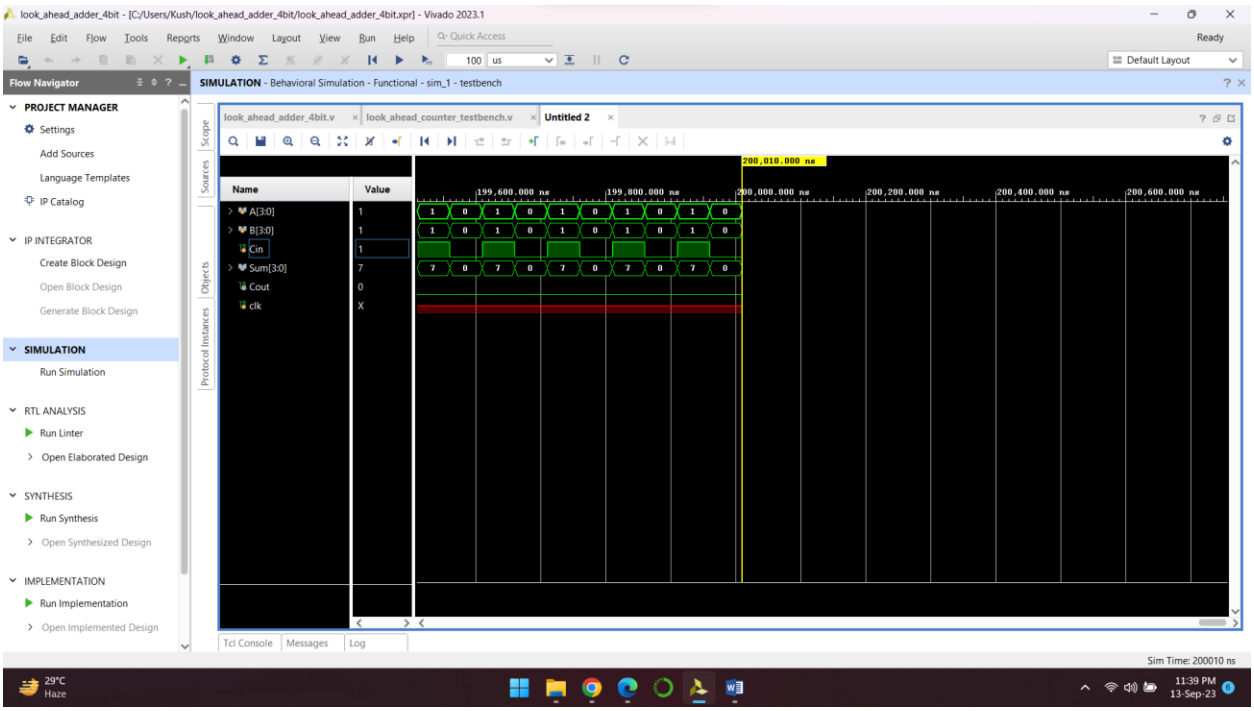


TESTBENCH:

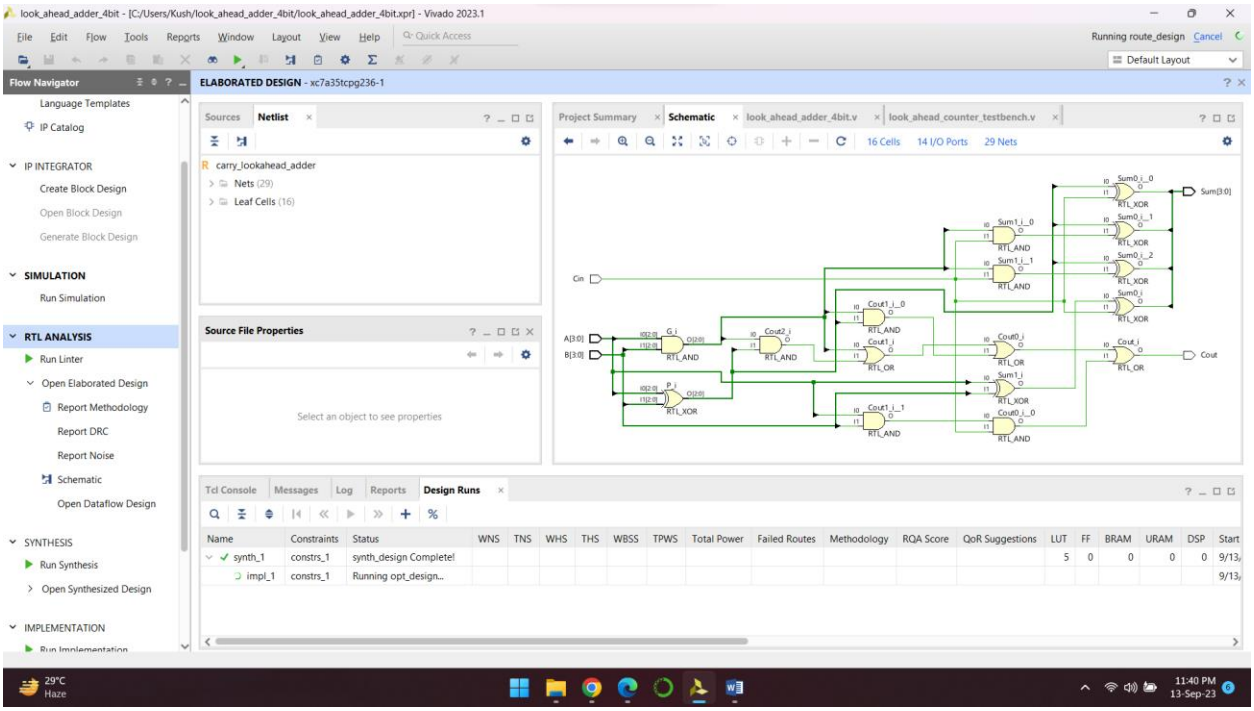


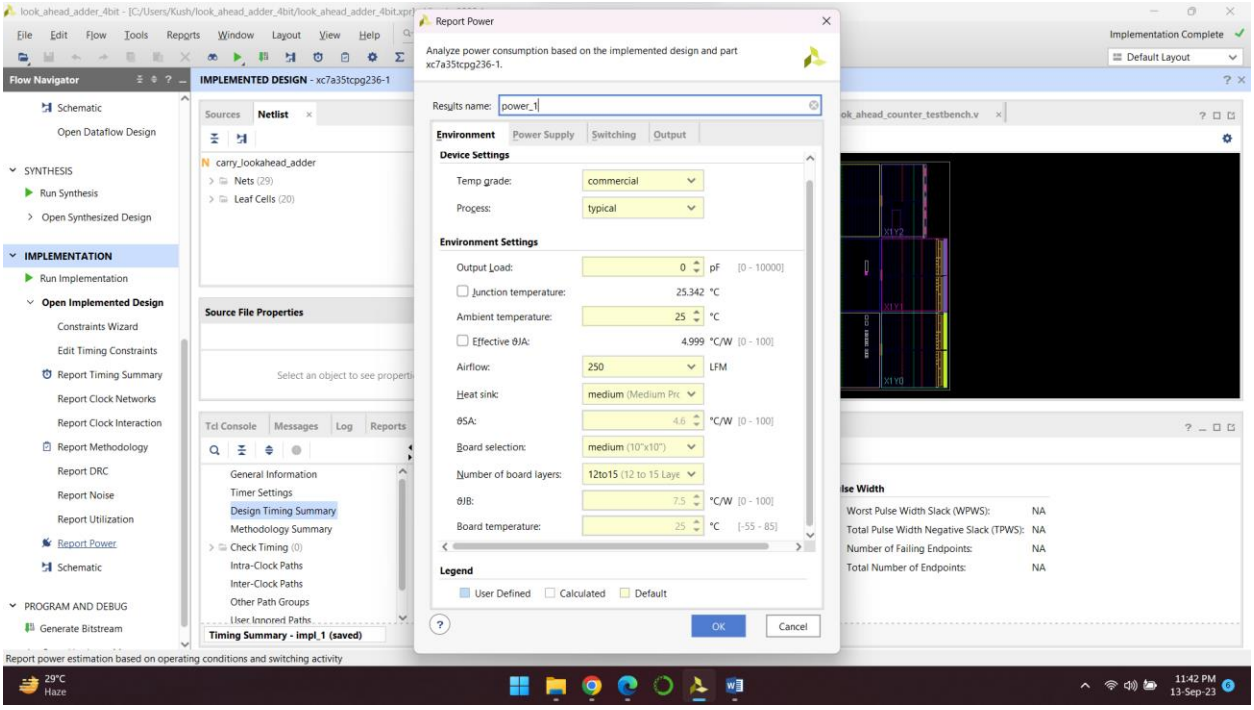

WAVEFORM:

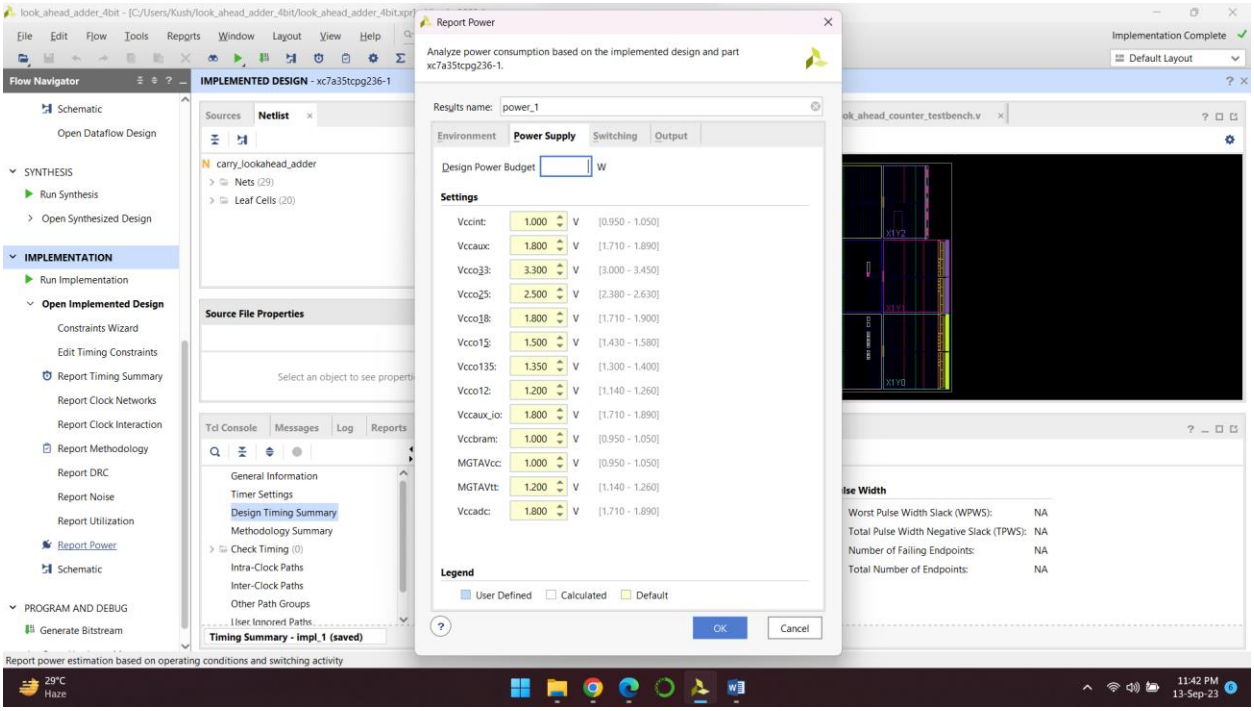



RTL SYNTHESIS:



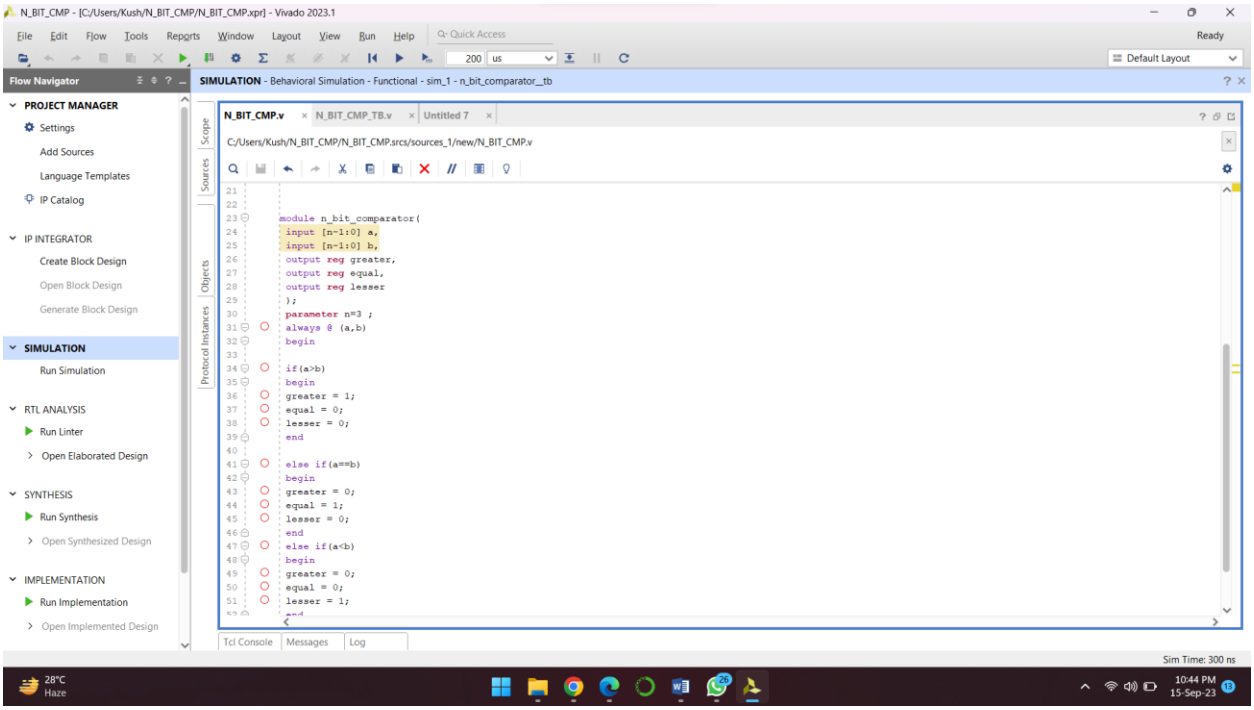
POWER REPORT:

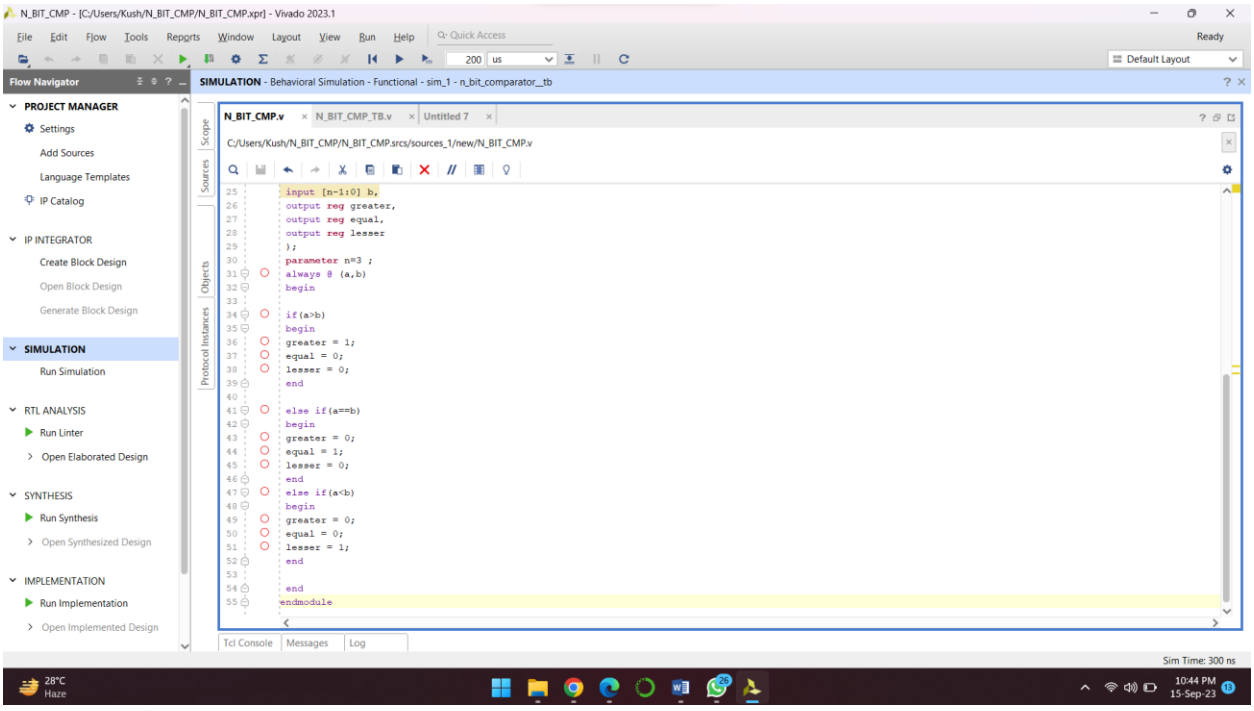


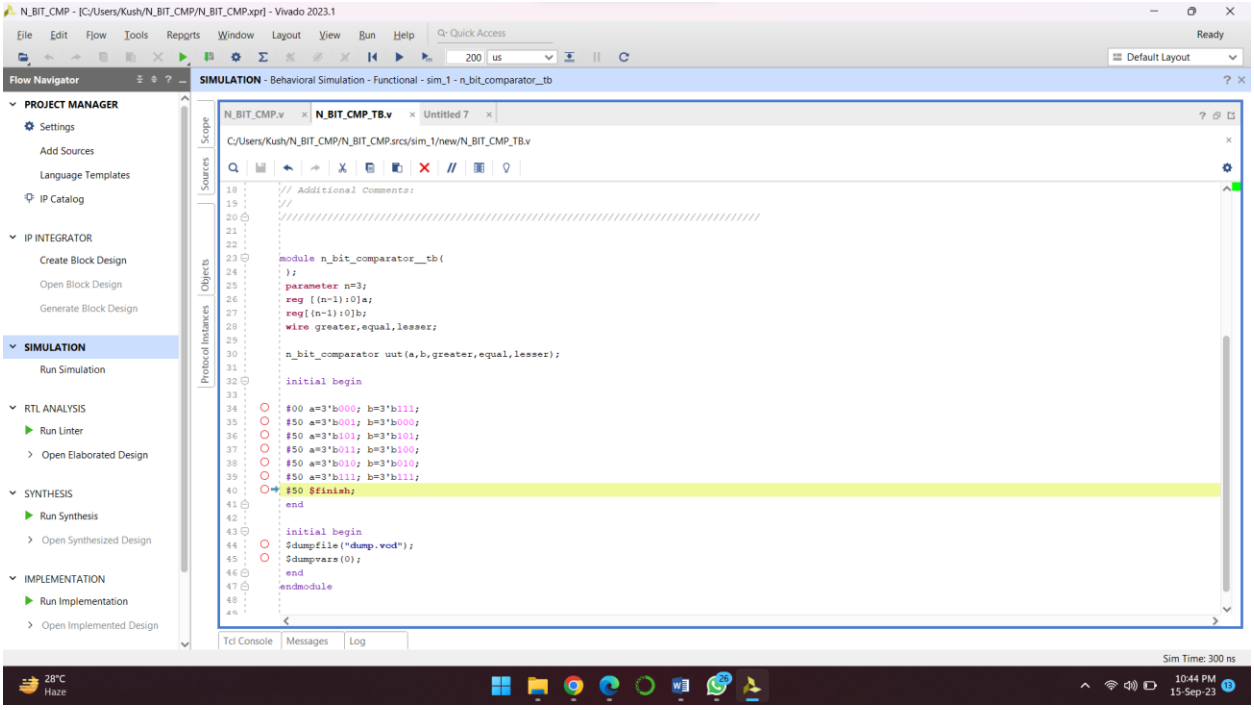
9. N-BIT COMPARATOR:

CODE:

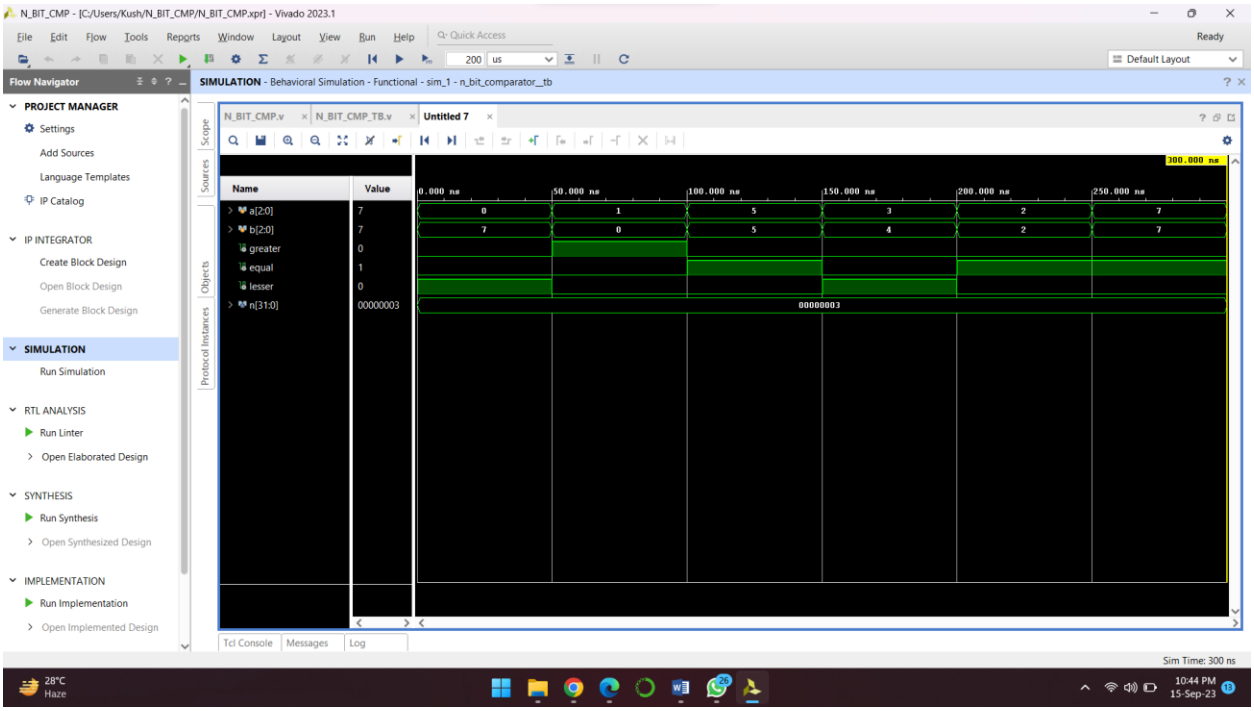


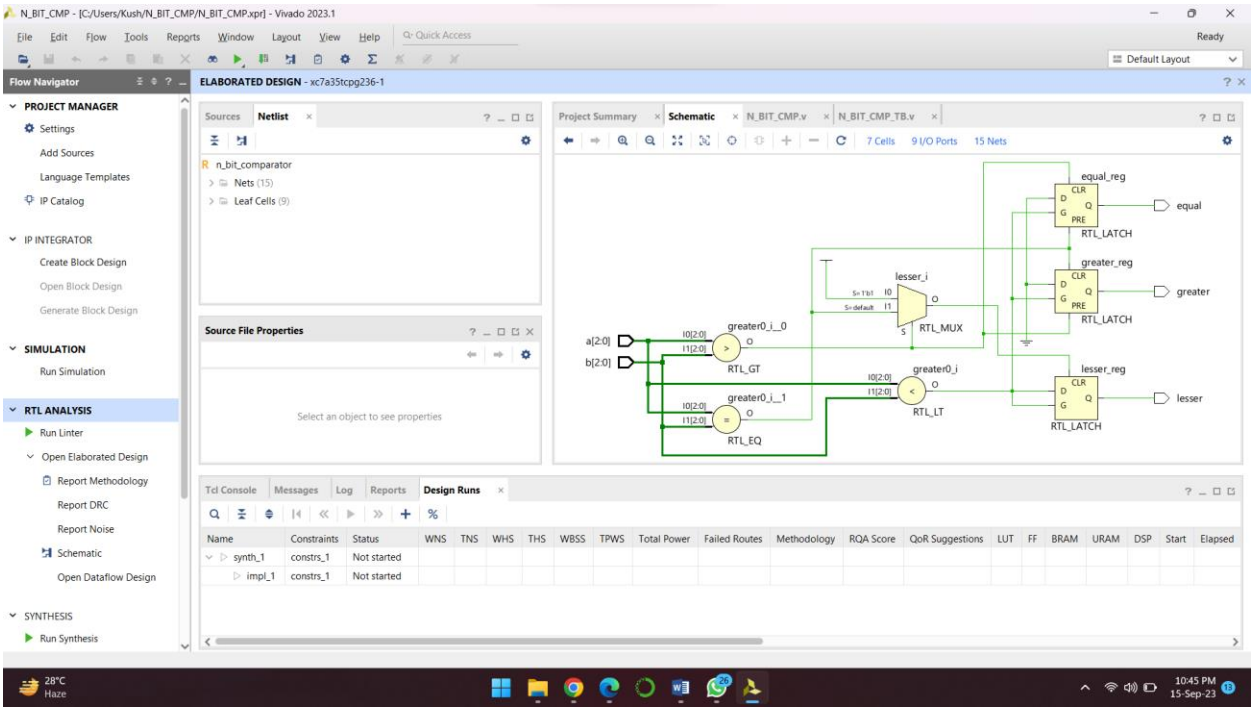
TESTBENCH:



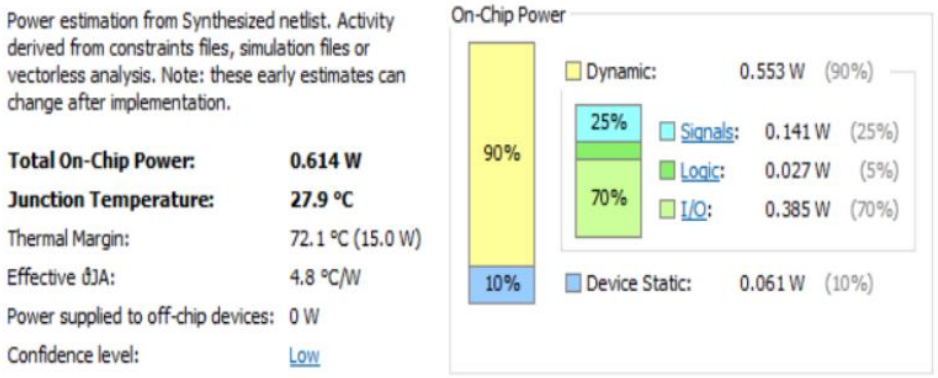

WAVEFORM:



RTL SCHEMATIC:

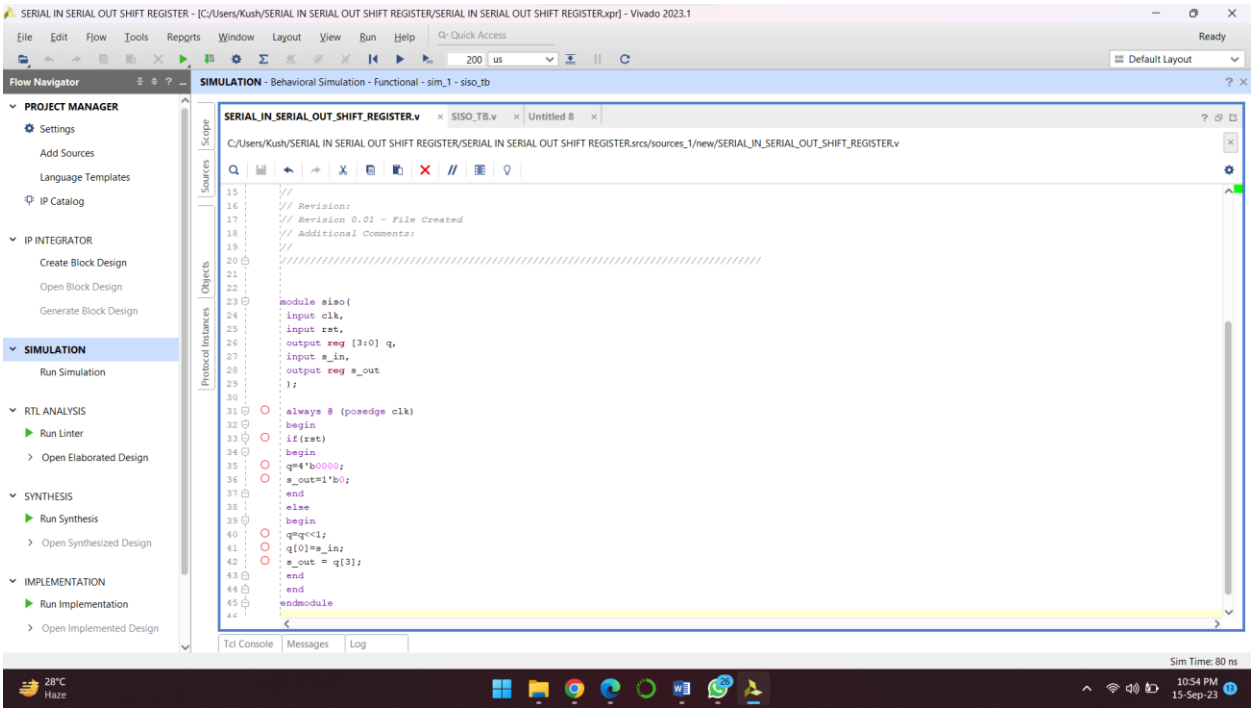



POWER REPORT:

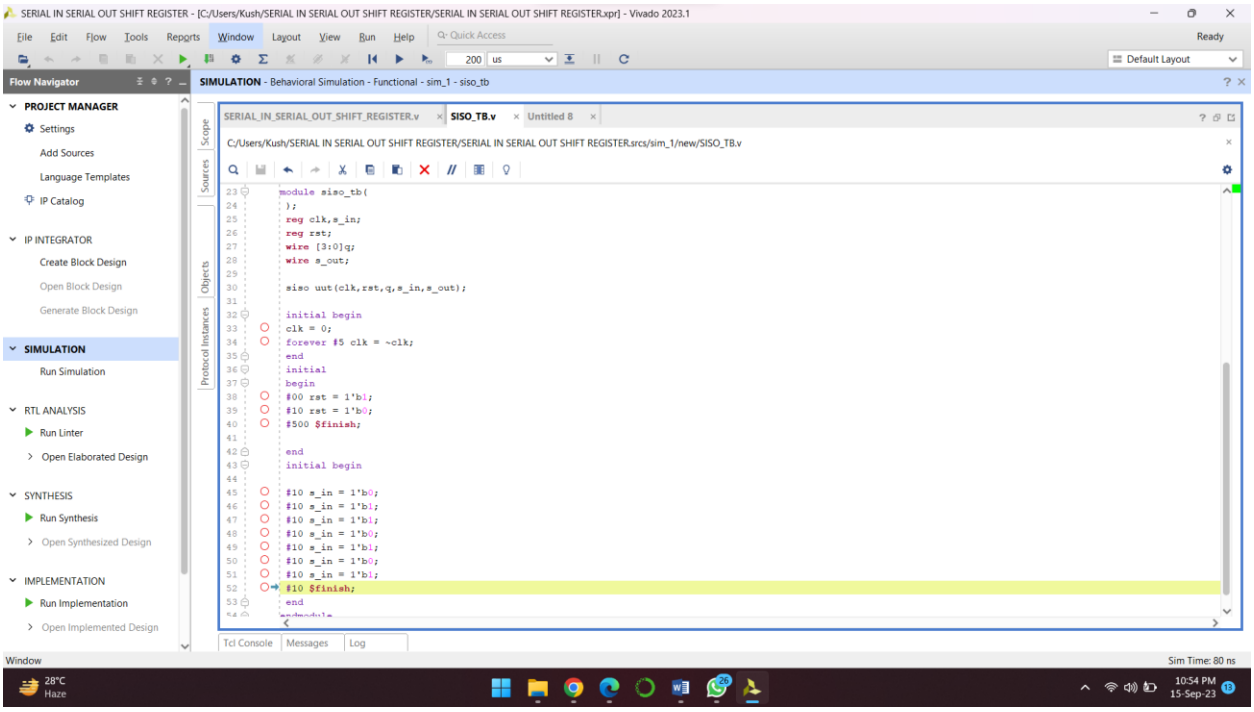


10. SERIAL IN SERIAL OUT SHIFT REGISTER:

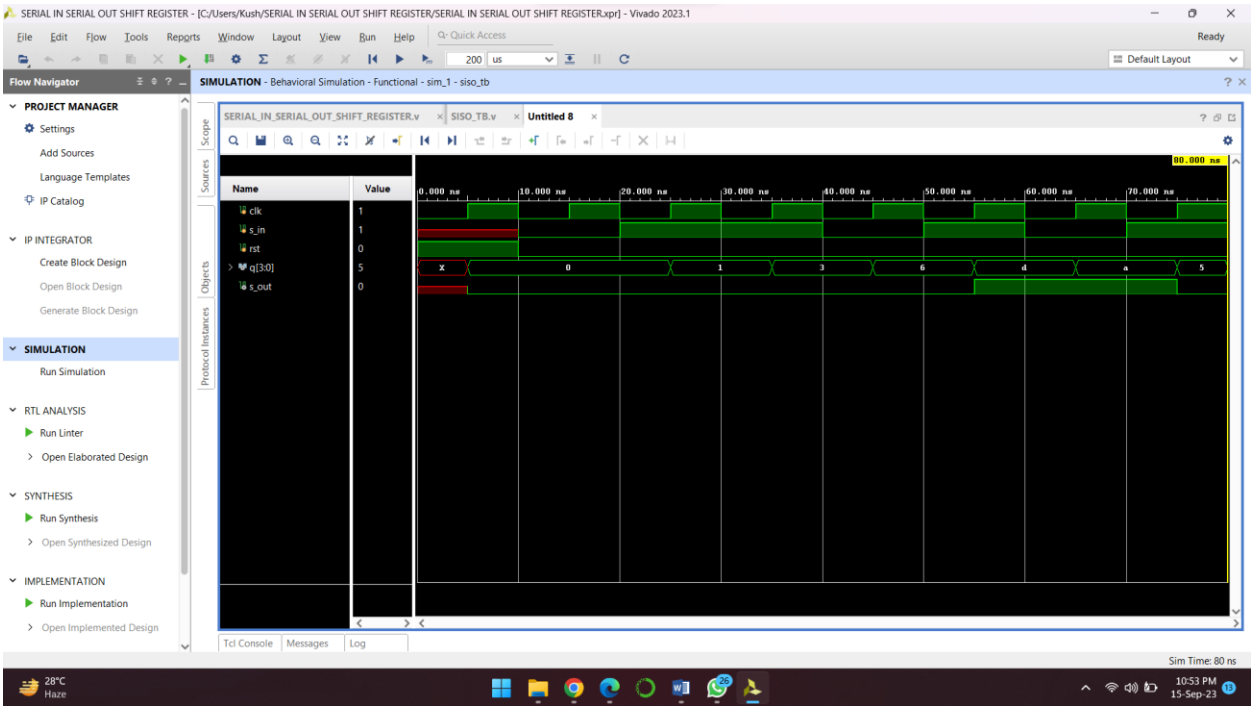
CODE:



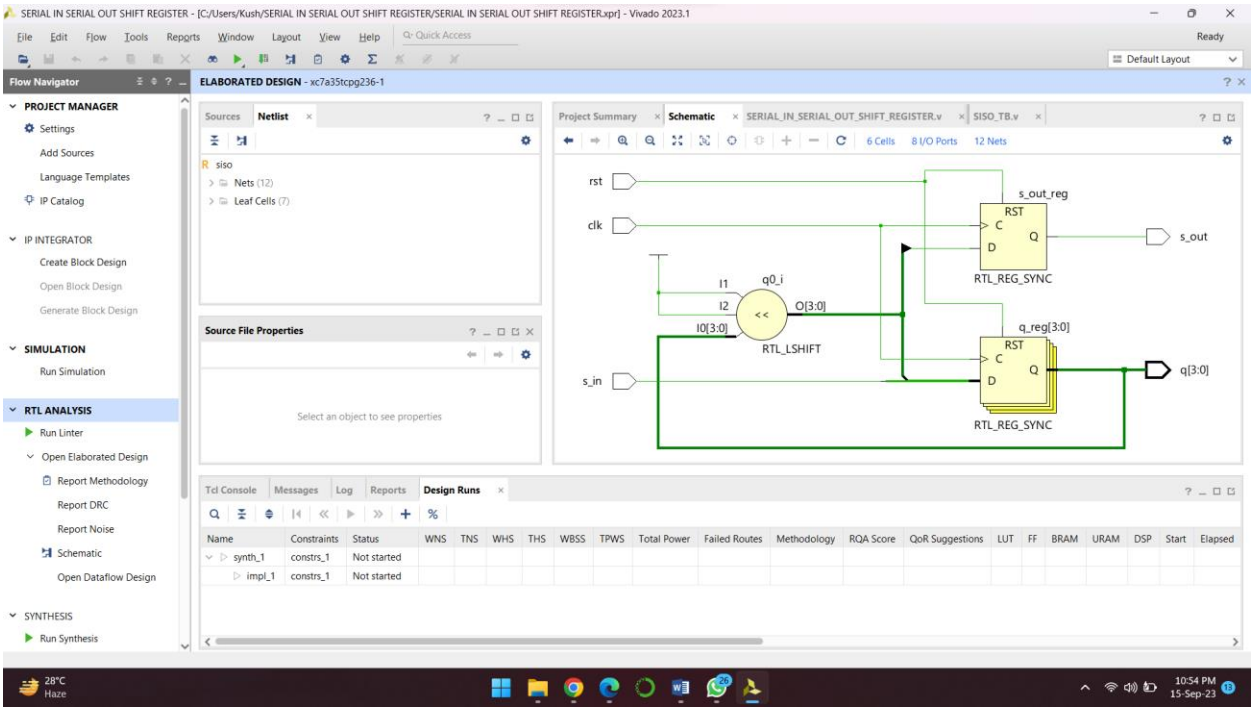
TESTBENCH:



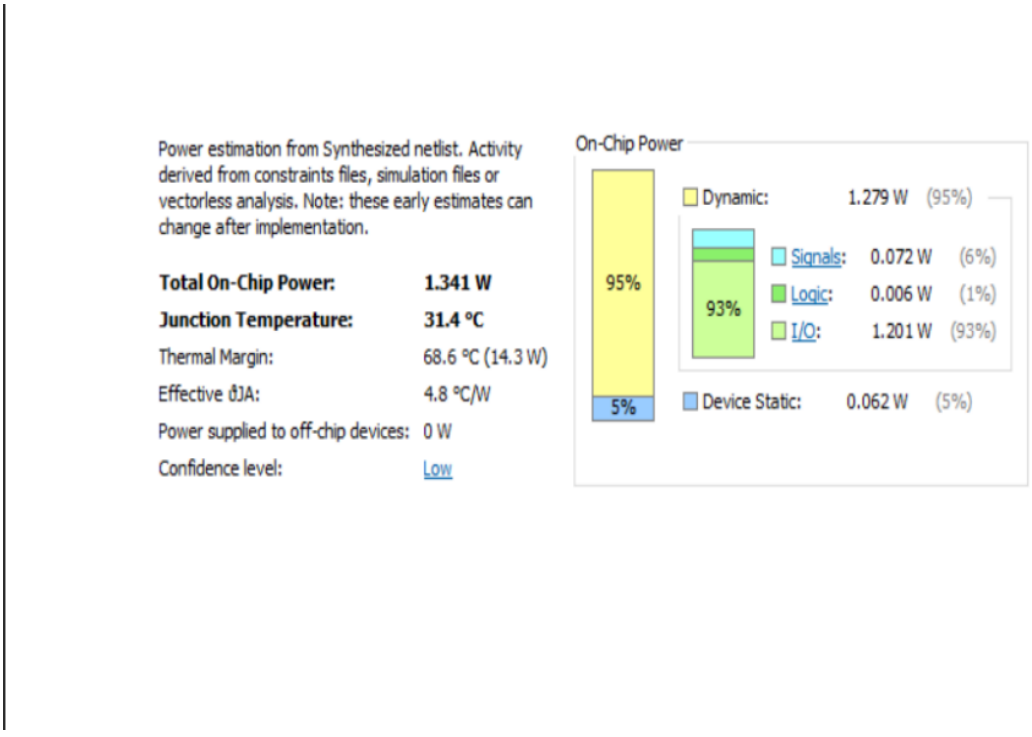

WAVEFORM:



RTL SCHEMATIC:

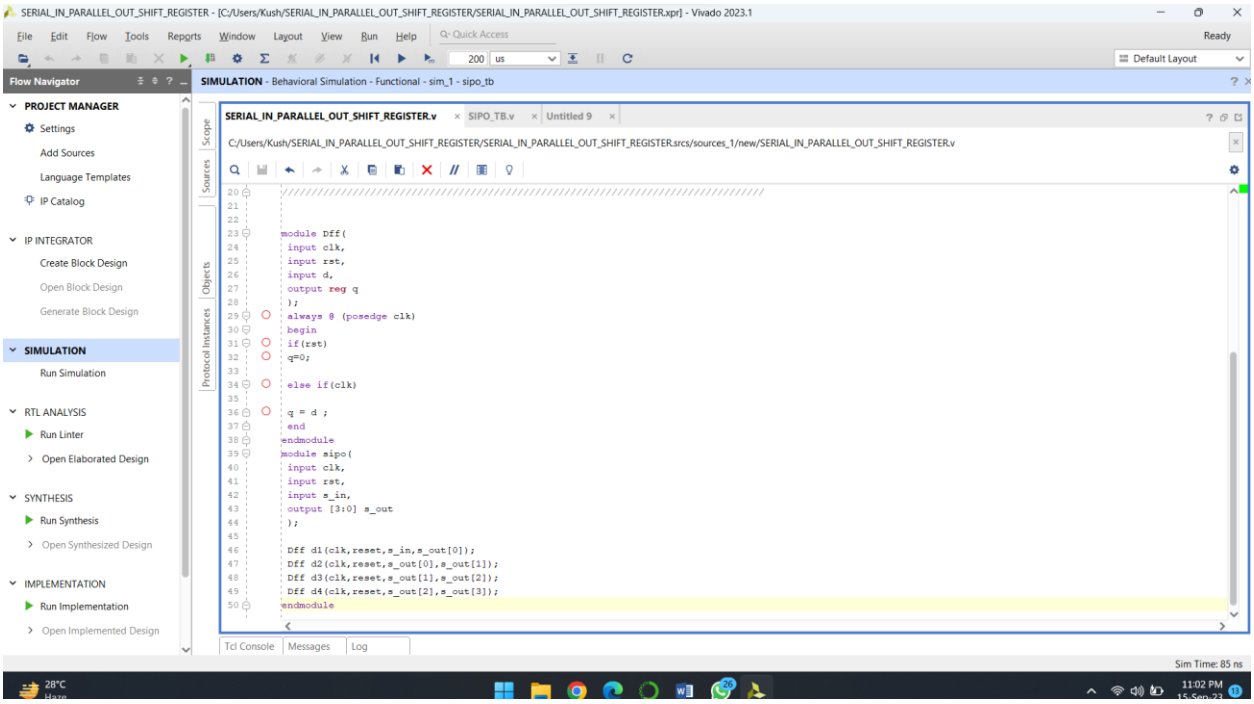



POWER REPORT:

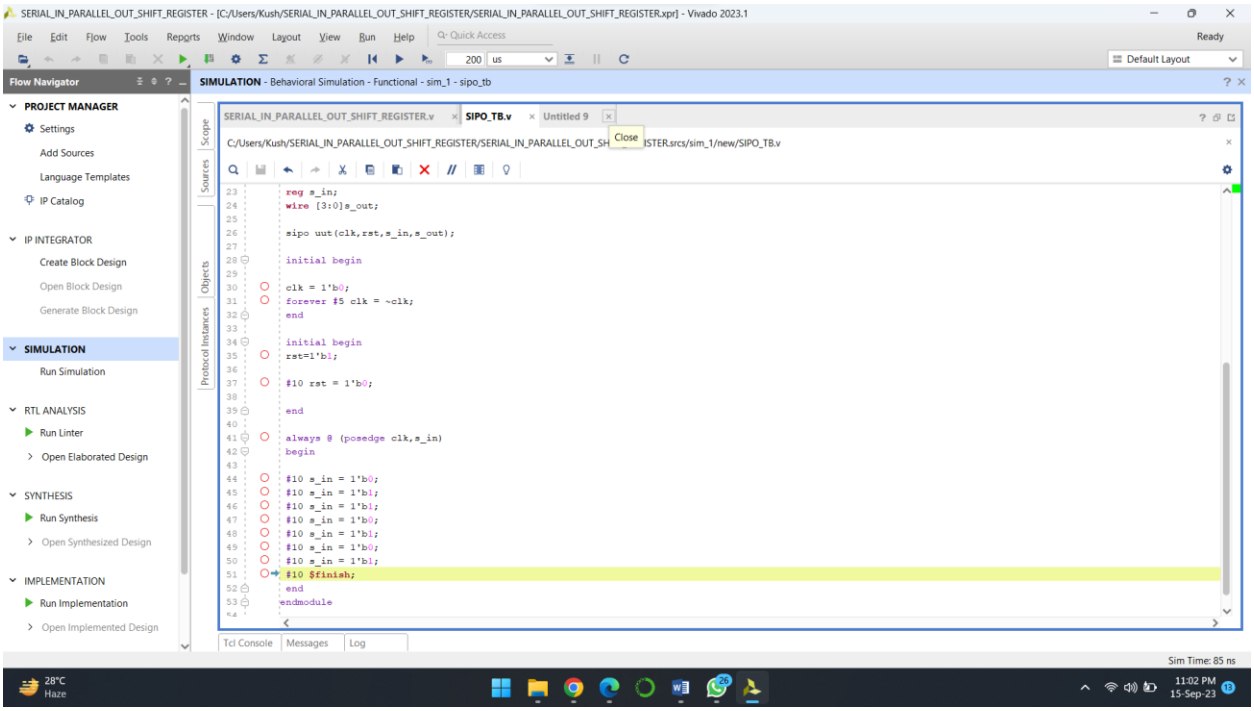


11. SERIAL IN PARALLEL OUT SHIFT REGISTER:

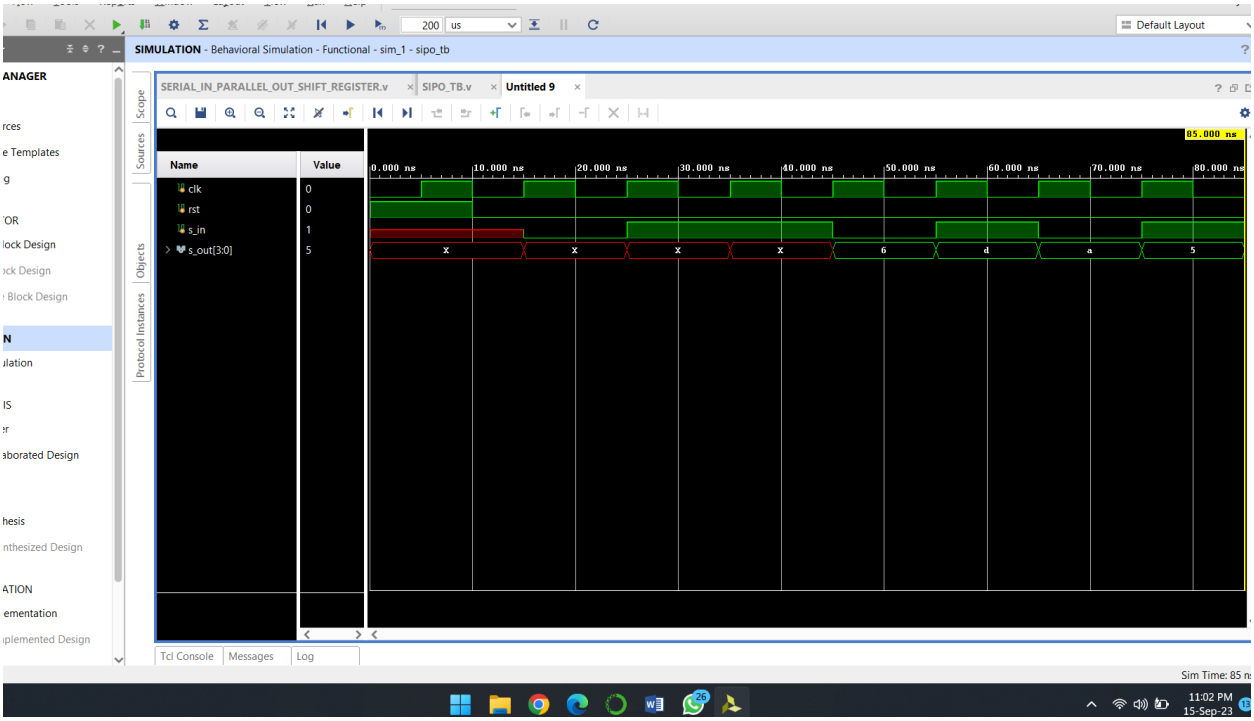
CODE:



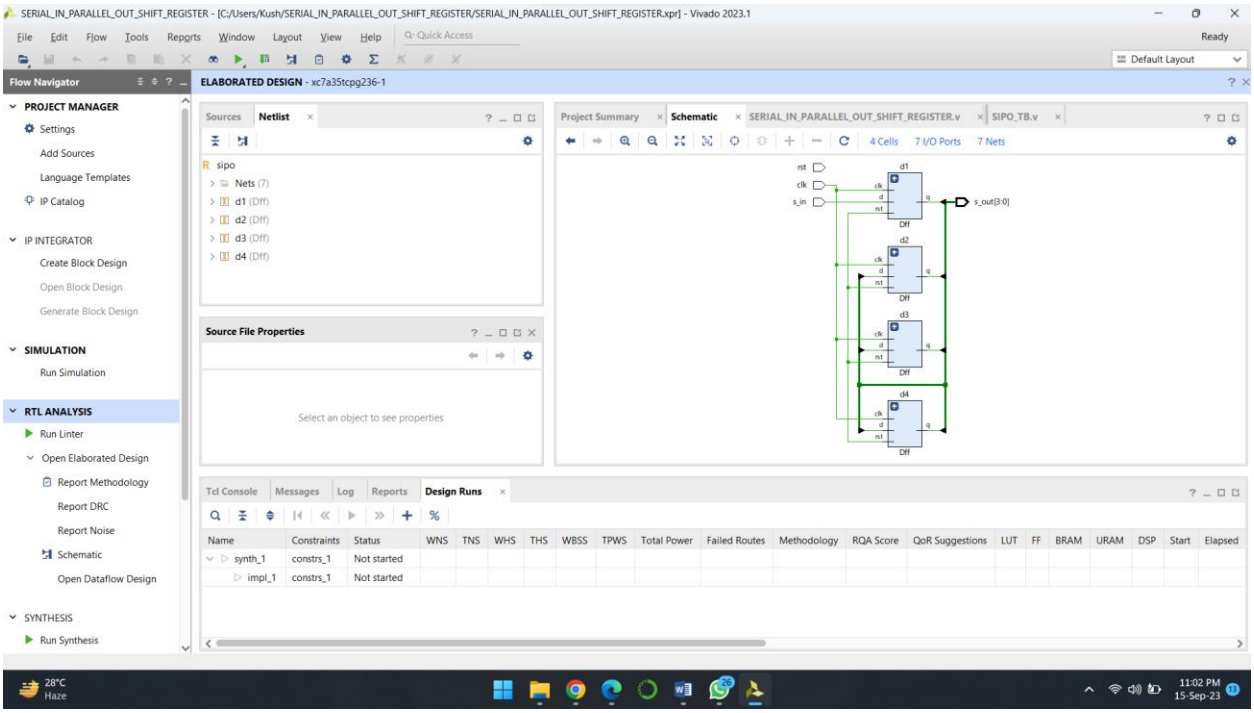

TESTBENCH:



## WAVEFORM:



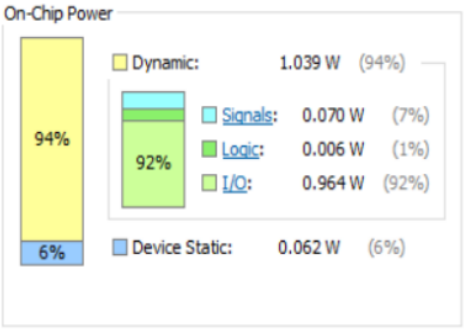

## RTL SCHEMATIC:



POWER REPORT:

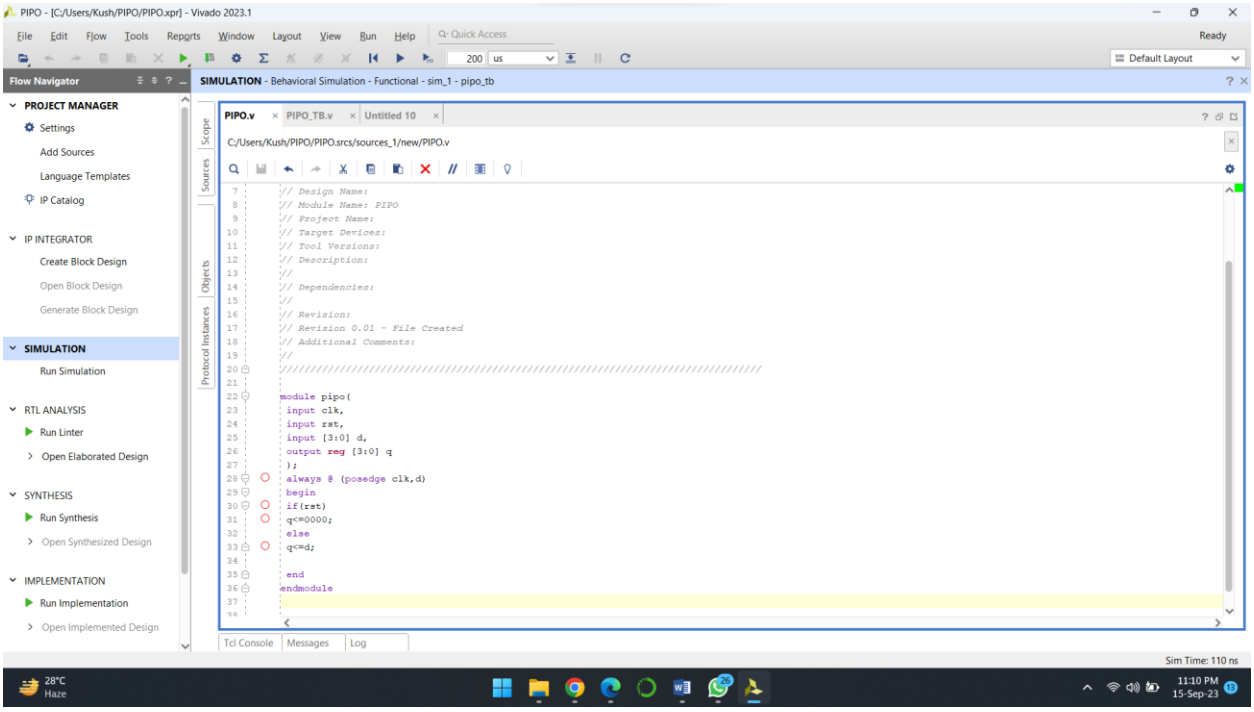
Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

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

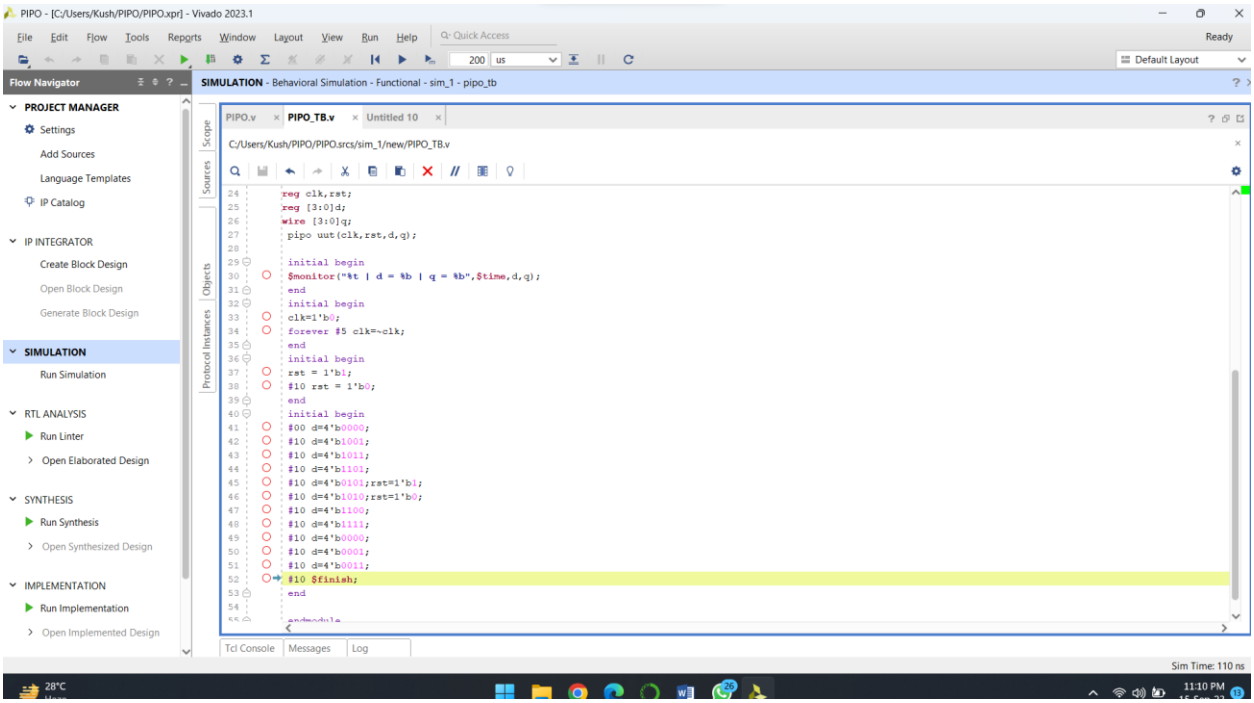



12. PARALLEL IN PARALLEL OUT REGISTER:

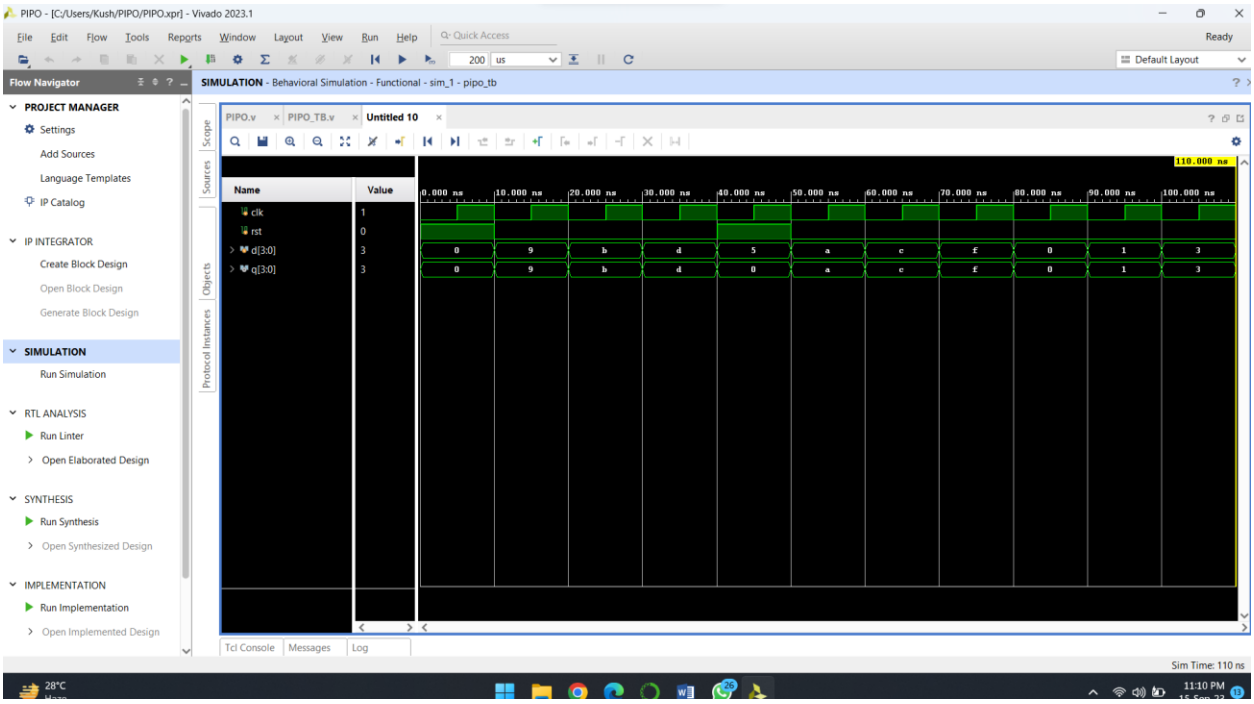
CODE:



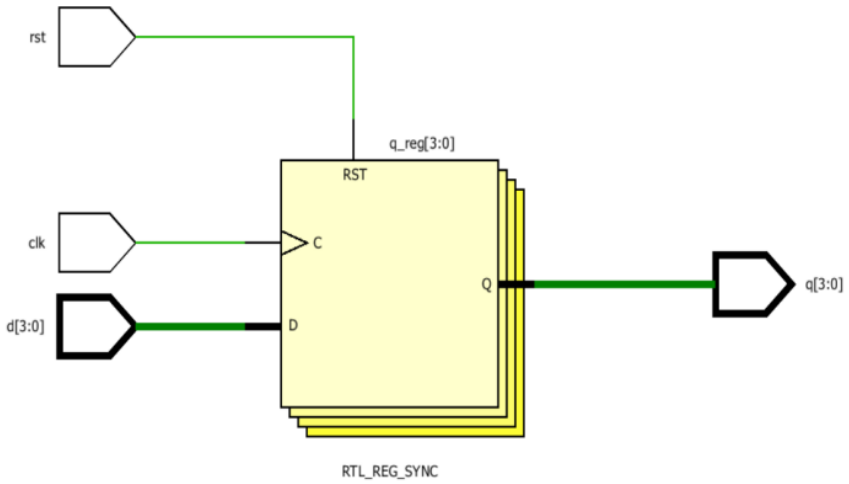
TESTBENCH:




WAVEFORM:



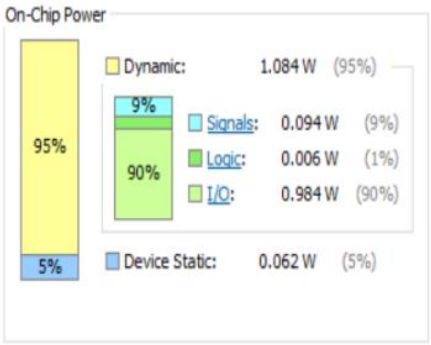
RTL SCHEMATIC:




POWER REPORT:

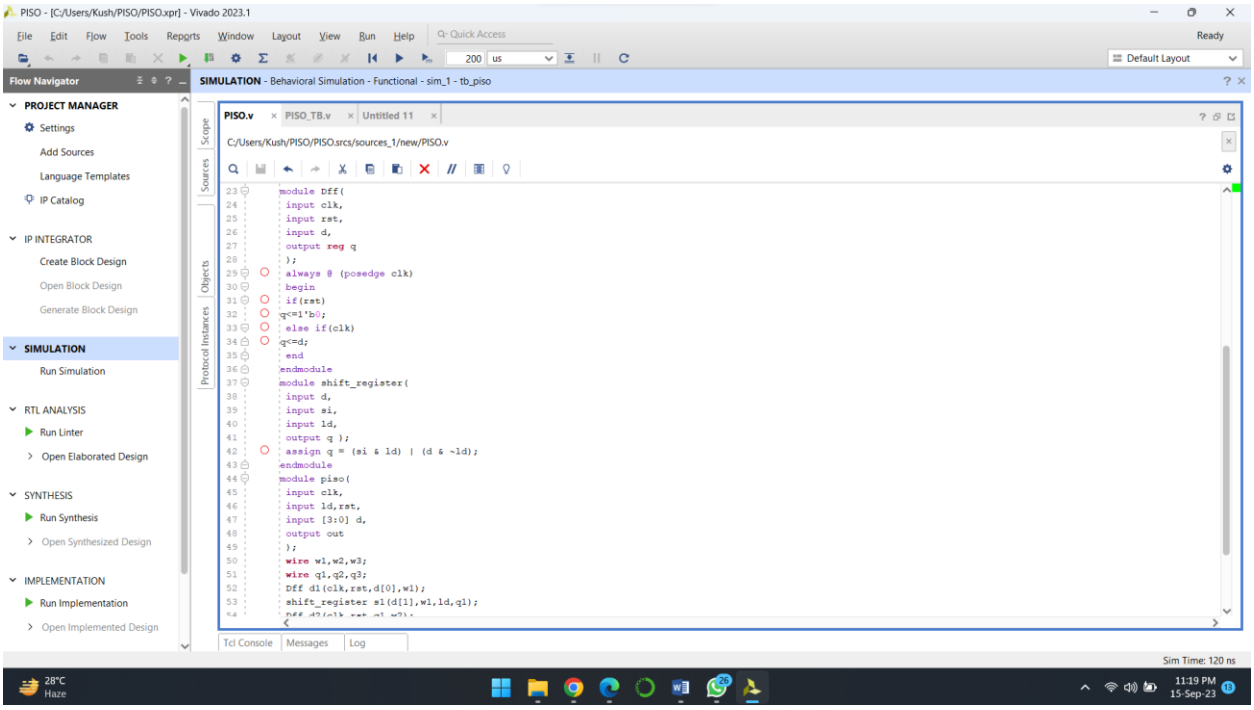
Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

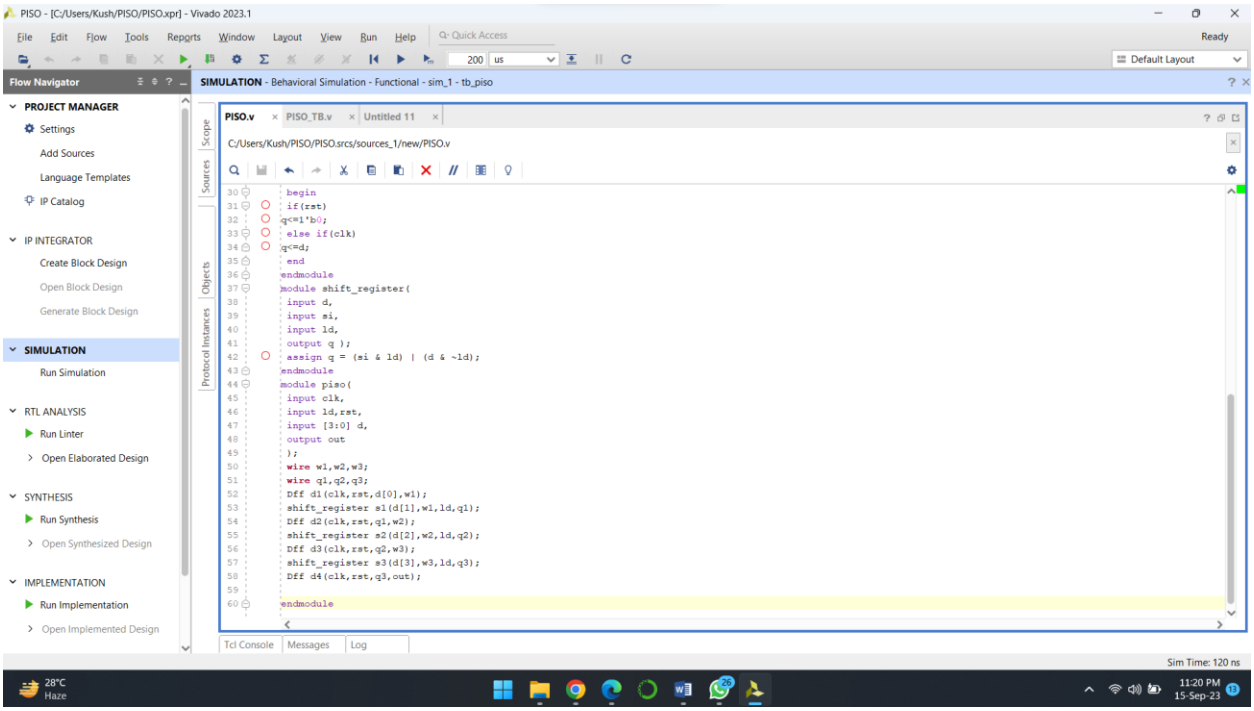
| | |
|---|---|
| **Total On-Chip Power:** | **1.146 W** |
| **Junction Temperature:** | **30.5 °C** |
| Thermal Margin: | 69.5 °C (14.5 W) |
| Effective ϑJA: | 4.8 °C/W |
| Power supplied to off-chip devices: | 0 W |
| Confidence level: | Low |



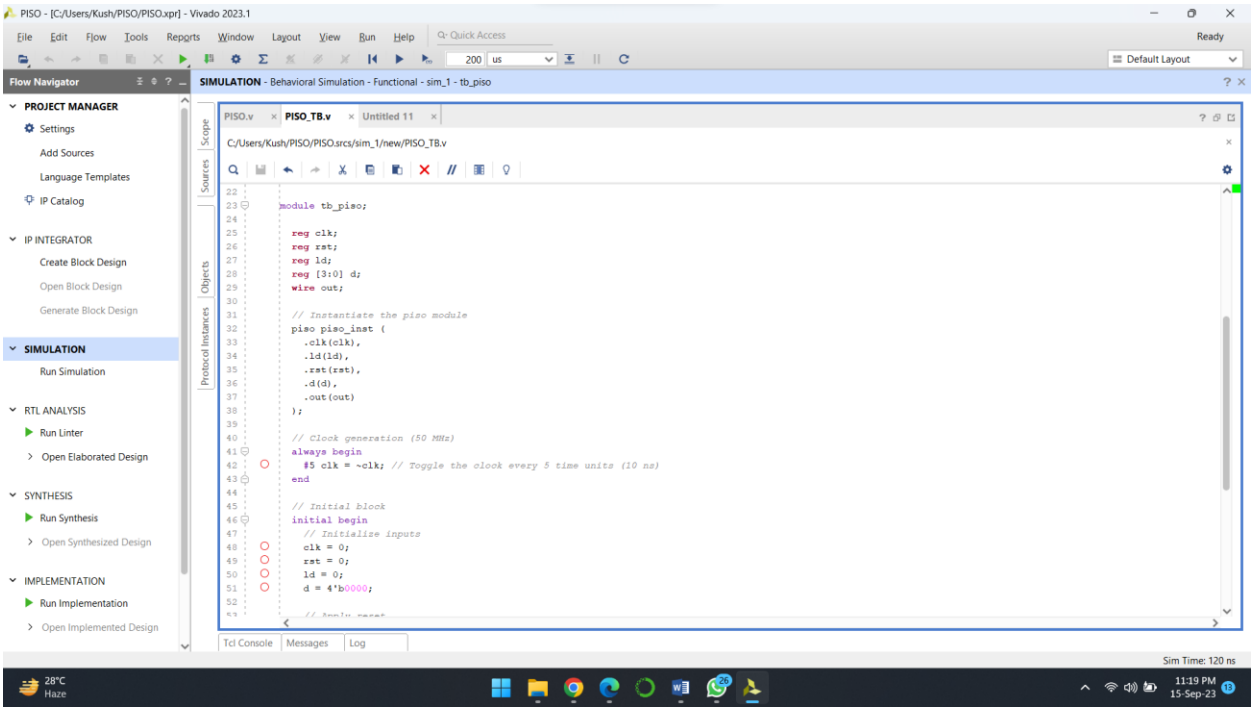
13. PARALLEL IN SERIAL OUT REGISTER:

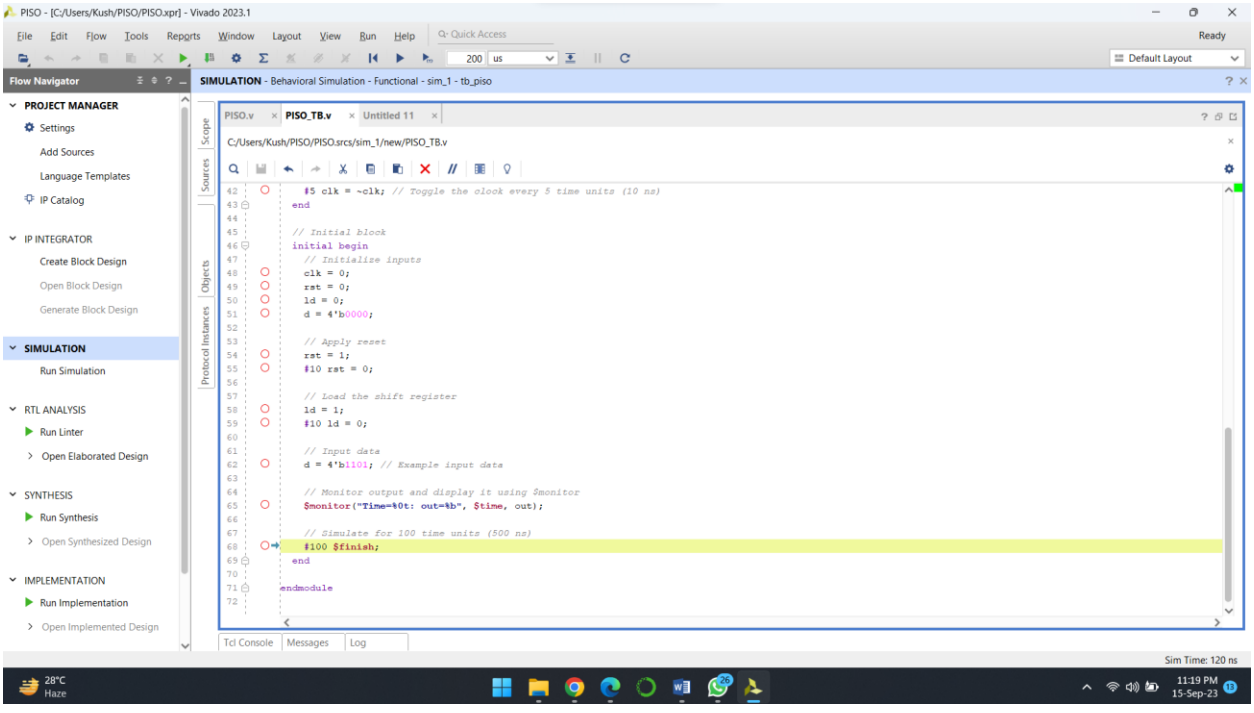
CODE:



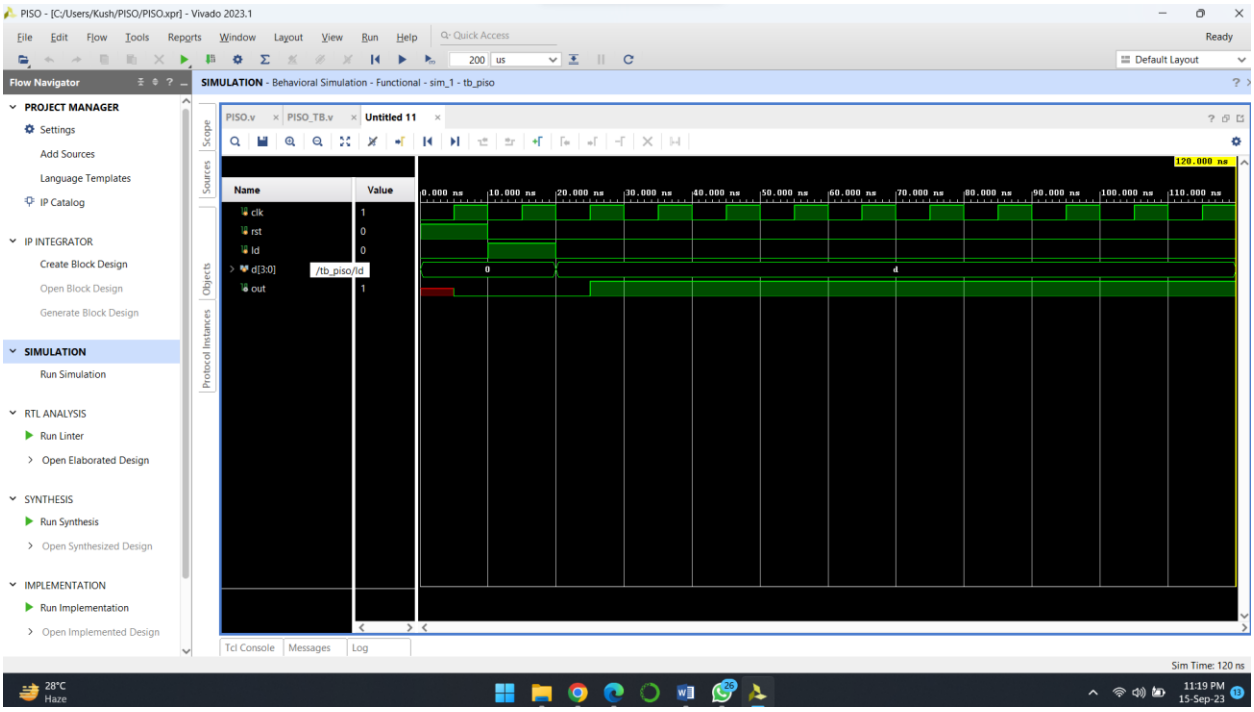



TESTBENCH:

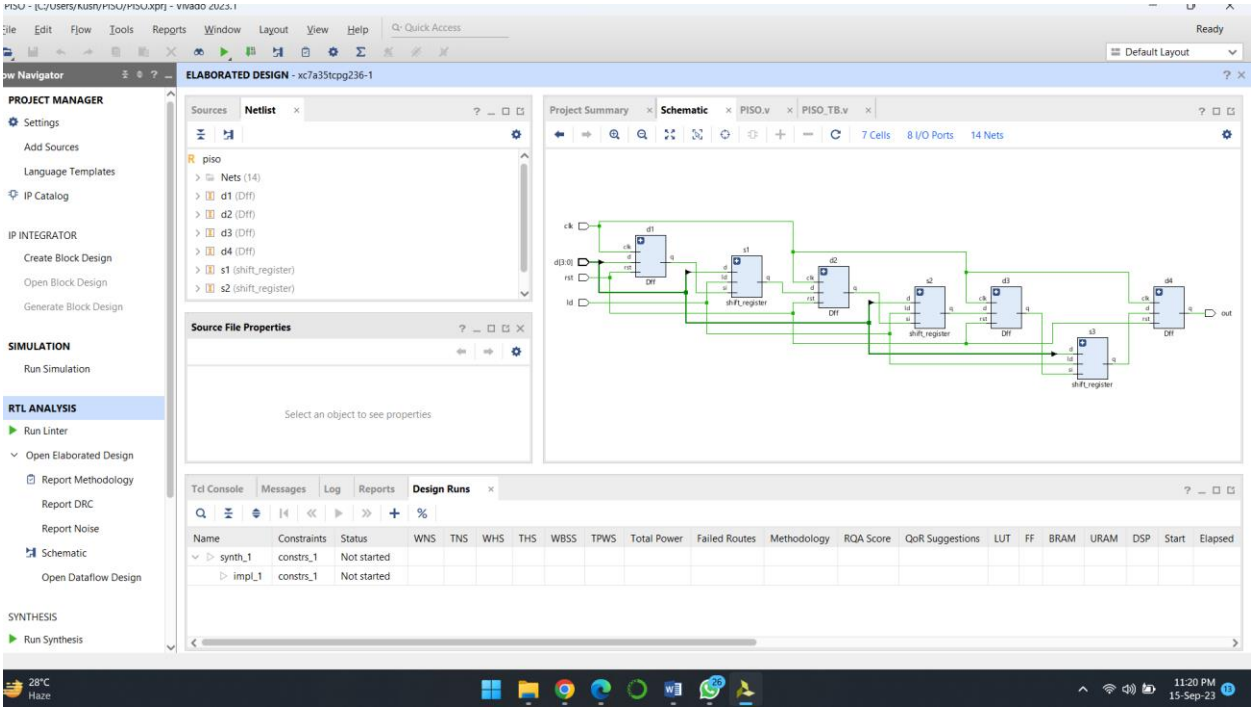

PISO - [C:/Users/Kush/PISO/PISO.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - tb_piso
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PISO.v
PISO_TB.v
Untitled 11
C:/Users/Kush/PISO/PISO.srcs/sim_1/new/PISO_TB.v
module tb_piso;
reg clk;
reg rst;
reg ld;
reg [3:0] d;
wire out;
// Instantiate the piso module
piso piso_inst (
.clk(clk),
.ld(ld),
.rst(rst),
.d(d),
.out(out)
);
// Clock generation (50 MHz)
always begin
#5 clk = ~clk; // Toggle the clock every 5 time units (10 ns)
end
// Initial block
initial begin
// Initialize inputs
clk = 0;
rst = 0;
ld = 0;
d = 4'b0000;
Tcl Console
Messages
Log
Sim Time: 120 ns
28°C
Haze
11:19 PM
15-Sep-23


PISO - [C:/Users/Kush/PISO/PISO.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - tb_piso
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PISO.v
PISO_TB.v
Untitled 11
C:/Users/Kush/PISO/PISO.srcs/sim_1/new/PISO_TB.v
#5 clk = ~clk; // Toggle the clock every 5 time units (10 ns)
end
// Initial block
initial begin
// Initialize inputs
clk = 0;
rst = 0;
ld = 0;
d = 4'b0000;
// Apply reset
rst = 1;
#10 rst = 0;
// Load the shift register
ld = 1;
#10 ld = 0;
// Input data
d = 4'b1101; // Example input data
// Monitor output and display it using $monitor
$monitor("Time=%0t: out=%b", $time, out);
// Simulate for 100 time units (500 ns)
#100 $finish;
end
endmodule
Tcl Console
Messages
Log
Sim Time: 120 ns
28°C
Haze
11:19 PM
15-Sep-23

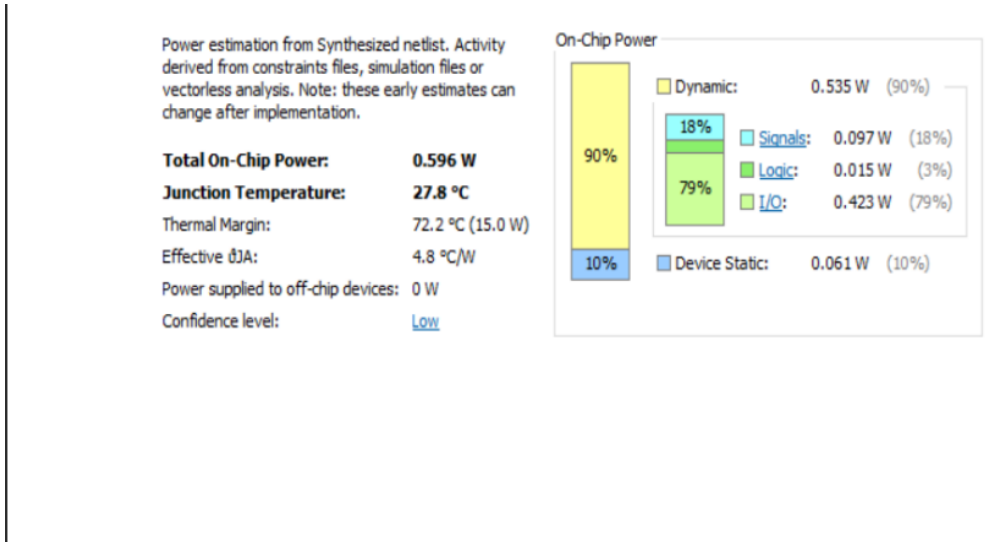
WAVEFORM:




RTL SCHEMATIC:

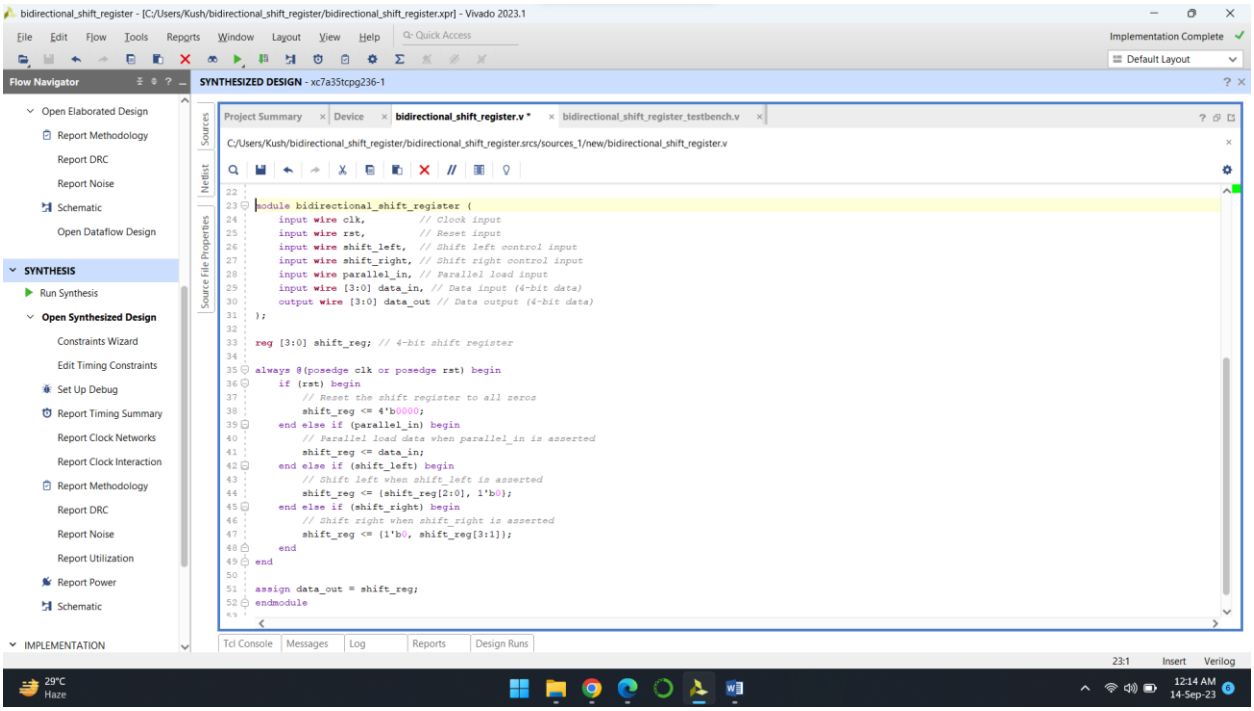


POWER REPORT:

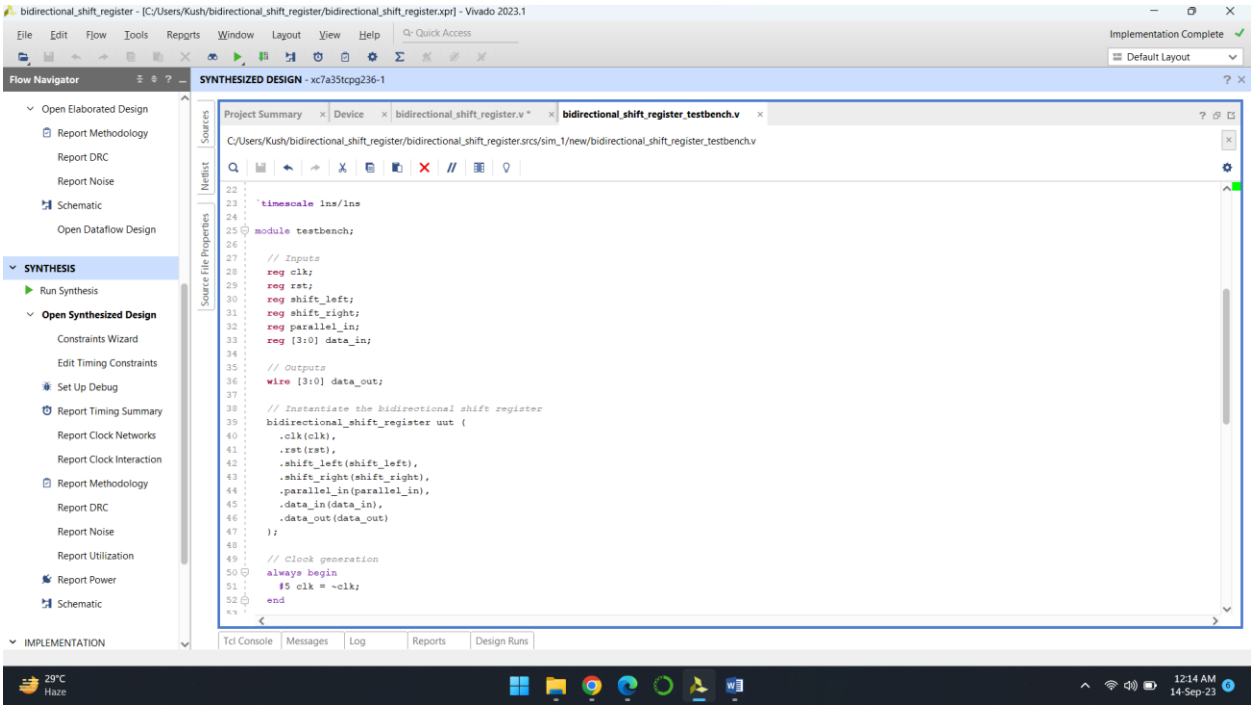


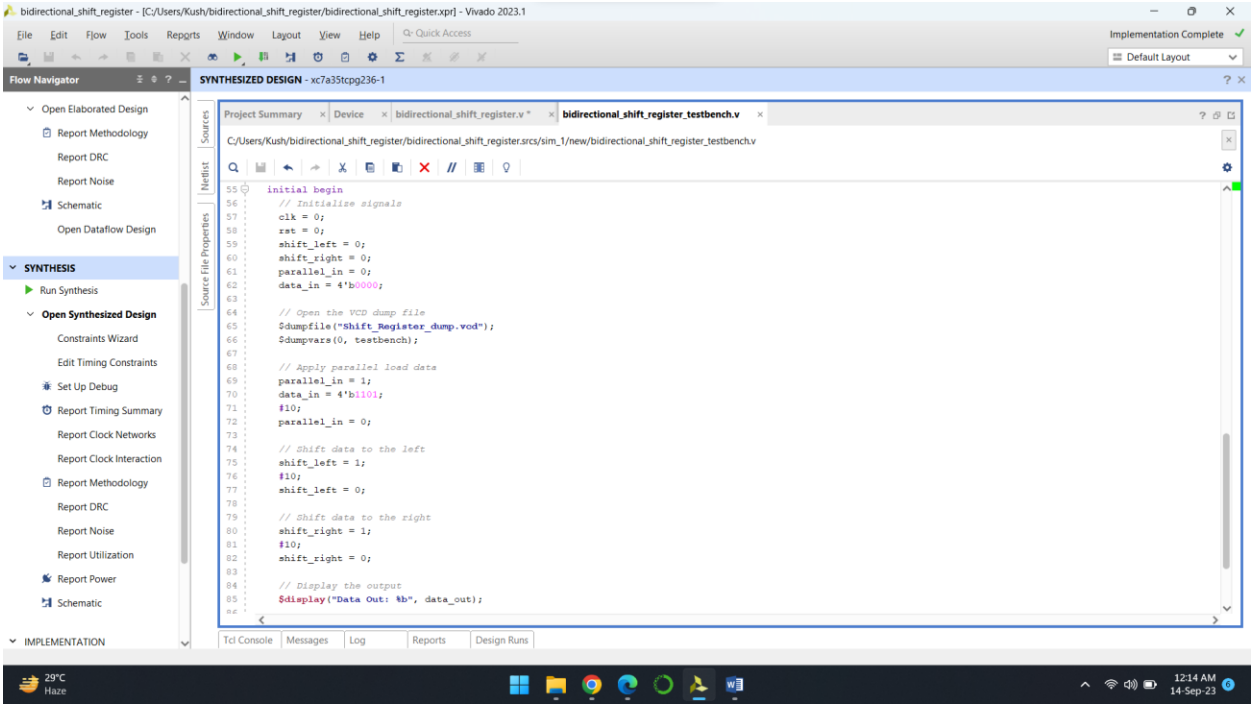

14. BIDIRECTION SHIFT REGISTER:

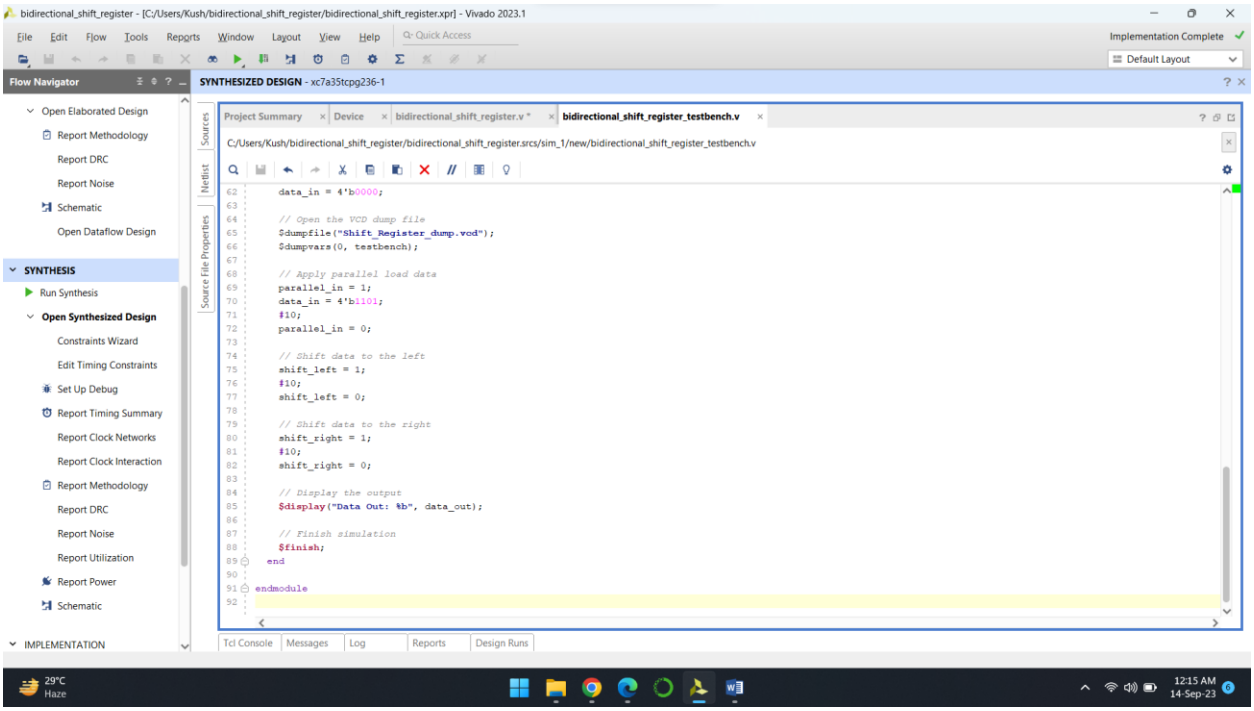
CODE:


bidirectional_shift_register - [C:/Users/Kush/bidirectional_shift_register/bidirectional_shift_register.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Implementation Complete
Default Layout
Flow Navigator
SYNTHESIZED DESIGN - xc7a35tcpg236-1
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
Open Dataflow Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Project Summary
Device
bidirectional_shift_register.v *
bidirectional_shift_register_testbench.v
C:/Users/Kush/bidirectional_shift_register/bidirectional_shift_register.srcs/sources_1/new/bidirectional_shift_register.v
Sources
Netlist
Source File Properties
module bidirectional_shift_register (
input wire clk, // Clock input
input wire rst, // Reset input
input wire shift_left, // Shift left control input
input wire shift_right, // Shift right control input
input wire parallel_in, // Parallel load input
input wire [3:0] data_in, // Data input (4-bit data)
output wire [3:0] data_out // Data output (4-bit data)
);
reg [3:0] shift_reg; // 4-bit shift register
always @(posedge clk or posedge rst) begin
if (rst) begin
// Reset the shift register to all zeros
shift_reg <= 4'b0000;
end else if (parallel_in) begin
// Parallel load data when parallel_in is asserted
shift_reg <= data_in;
end else if (shift_left) begin
// Shift left when shift_left is asserted
shift_reg <= {shift_reg[2:0], 1'b0};
end else if (shift_right) begin
// Shift right when shift_right is asserted
shift_reg <= {1'b0, shift_reg[3:1]};
end
end
assign data_out = shift_reg;
endmodule
Tcl Console
Messages
Log
Reports
Design Runs
23:1
Insert
Verilog
29°C
Haze
12:14 AM
14-Sep-23

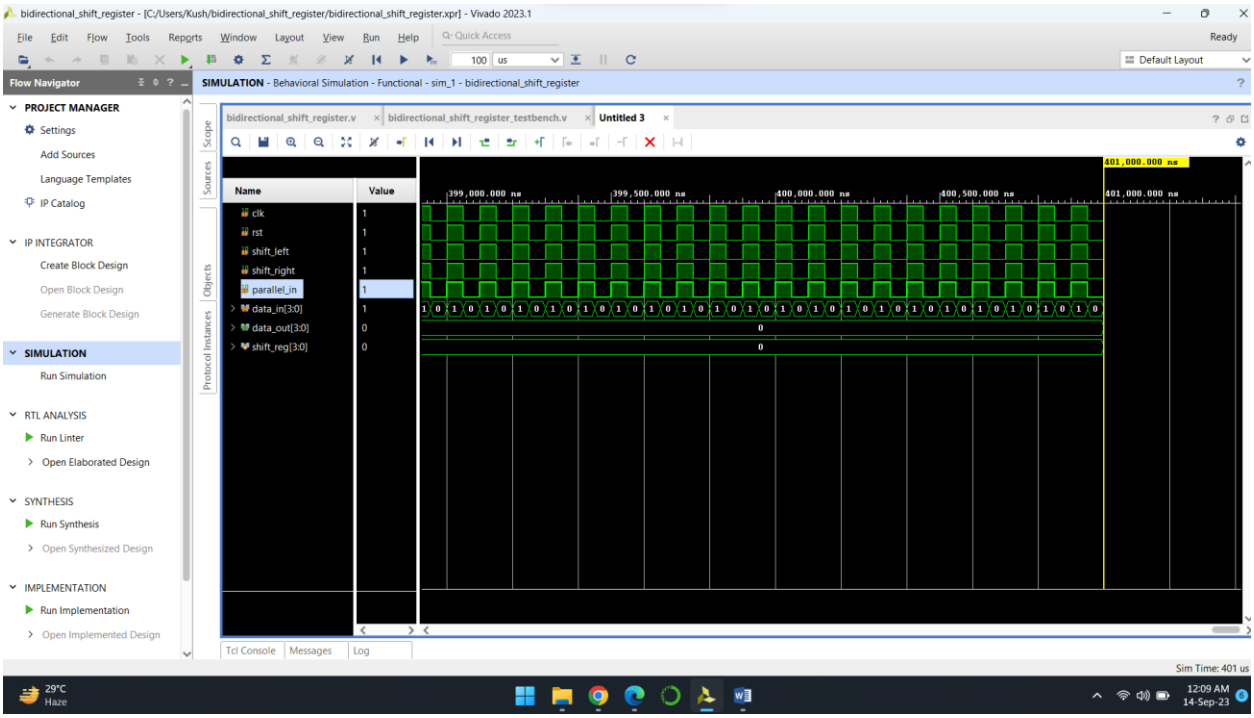
TESTBENCH:


bidirectional_shift_register - [C:/Users/Kush/bidirectional_shift_register/bidirectional_shift_register.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Implementation Complete
Default Layout
Flow Navigator
SYNTHESIZED DESIGN - xc7a35tcpg236-1
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
Open Dataflow Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Project Summary
Device
bidirectional_shift_register.v *
bidirectional_shift_register_testbench.v
C:/Users/Kush/bidirectional_shift_register/bidirectional_shift_register.srcs/sim_1/new/bidirectional_shift_register_testbench.v
Sources
Netlist
Source File Properties
`timescale 1ns/1ns
module testbench;
// Inputs
reg clk;
reg rst;
reg shift_left;
reg shift_right;
reg parallel_in;
reg [3:0] data_in;
// Outputs
wire [3:0] data_out;
// Instantiate the bidirectional shift register
bidirectional_shift_register uut (
.clk(clk),
.rst(rst),
.shift_left(shift_left),
.shift_right(shift_right),
.parallel_in(parallel_in),
.data_in(data_in),
.data_out(data_out)
);
// Clock generation
always begin
#5 clk = ~clk;
end
Tcl Console
Messages
Log
Reports
Design Runs
29°C
Haze
12:14 AM
14-Sep-23

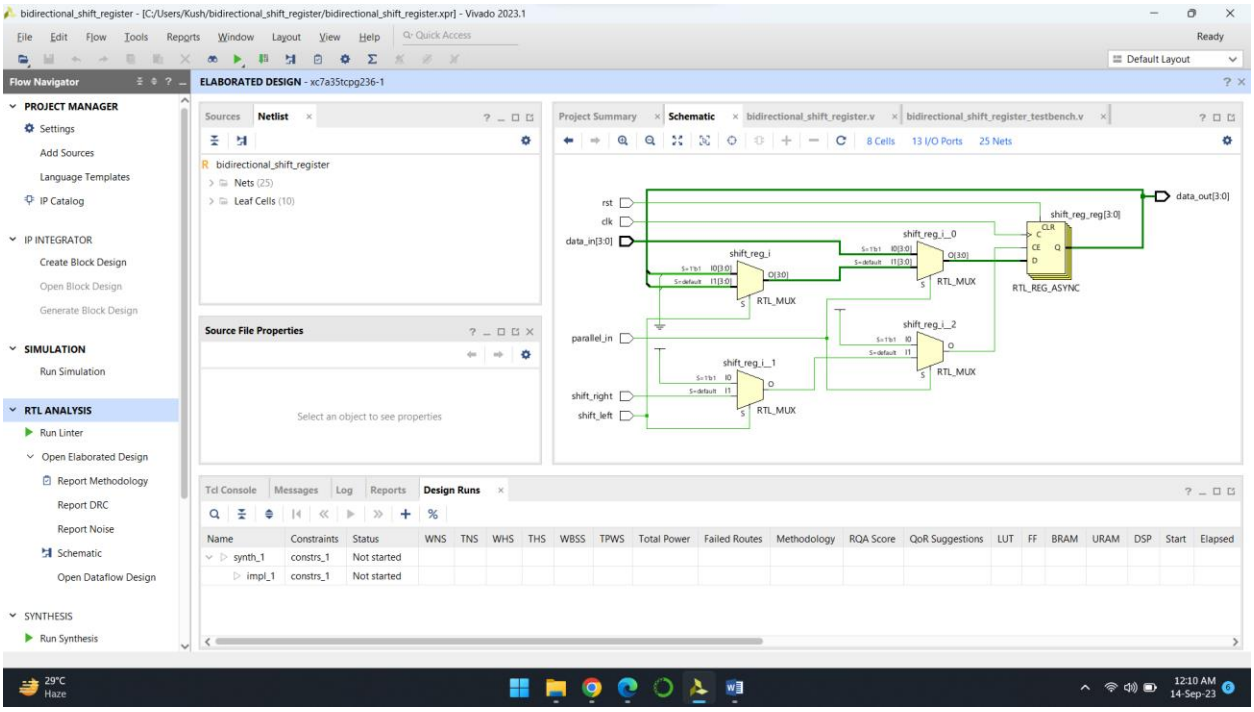

bidirectional_shift_register - [C:/Users/Kush/bidirectional_shift_register/bidirectional_shift_register.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Implementation Complete
Default Layout
Flow Navigator
SYNTHESIZED DESIGN - xc7a35tcpg236-1
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
Open Dataflow Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Project Summary
Device
bidirectional_shift_register.v *
bidirectional_shift_register_testbench.v
C:/Users/Kush/bidirectional_shift_register/bidirectional_shift_register.srcs/sim_1/new/bidirectional_shift_register_testbench.v
Sources
Netlist
Source File Properties
initial begin
// Initialize signals
clk = 0;
rst = 0;
shift_left = 0;
shift_right = 0;
parallel_in = 0;
data_in = 4'b0000;
// Open the VCD dump file
$dumpfile("Shift_Register_dump.vcd");
$dumpvars(0, testbench);
// Apply parallel load data
parallel_in = 1;
data_in = 4'b1101;
#10;
parallel_in = 0;
// Shift data to the left
shift_left = 1;
#10;
shift_left = 0;
// Shift data to the right
shift_right = 1;
#10;
shift_right = 0;
// Display the output
$display("Data Out: %b", data_out);
Tcl Console
Messages
Log
Reports
Design Runs
29°C
Haze
12:14 AM
14-Sep-23


bidirectional_shift_register - [C:/Users/Kush/bidirectional_shift_register/bidirectional_shift_register.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Implementation Complete
Default Layout
Flow Navigator
SYNTHESIZED DESIGN - xc7a35tcpg236-1
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
Open Dataflow Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Project Summary
Device
bidirectional_shift_register.v *
bidirectional_shift_register_testbench.v
C:/Users/Kush/bidirectional_shift_register/bidirectional_shift_register.srcs/sim_1/new/bidirectional_shift_register_testbench.v
Sources
Netlist
Source File Properties
data_in = 4'b0000;
// Open the VCD dump file
$dumpfile("Shift_Register_dump.vcd");
$dumpvars(0, testbench);
// Apply parallel load data
parallel_in = 1;
data_in = 4'b1101;
#10;
parallel_in = 0;
// Shift data to the left
shift_left = 1;
#10;
shift_left = 0;
// Shift data to the right
shift_right = 1;
#10;
shift_right = 0;
// Display the output
$display("Data Out: %b", data_out);
// Finish simulation
$finish;
end
endmodule
Tcl Console
Messages
Log
Reports
Design Runs
29°C
Haze
12:15 AM
14-Sep-23

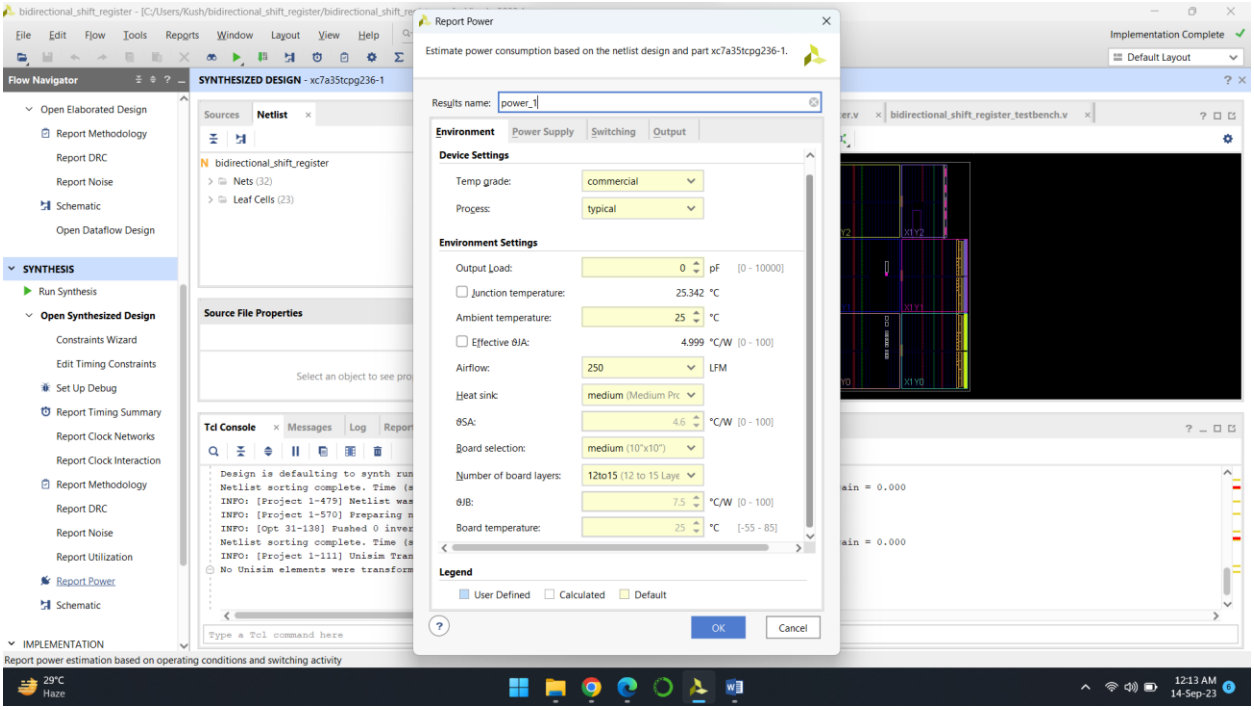
WAVEFORM:

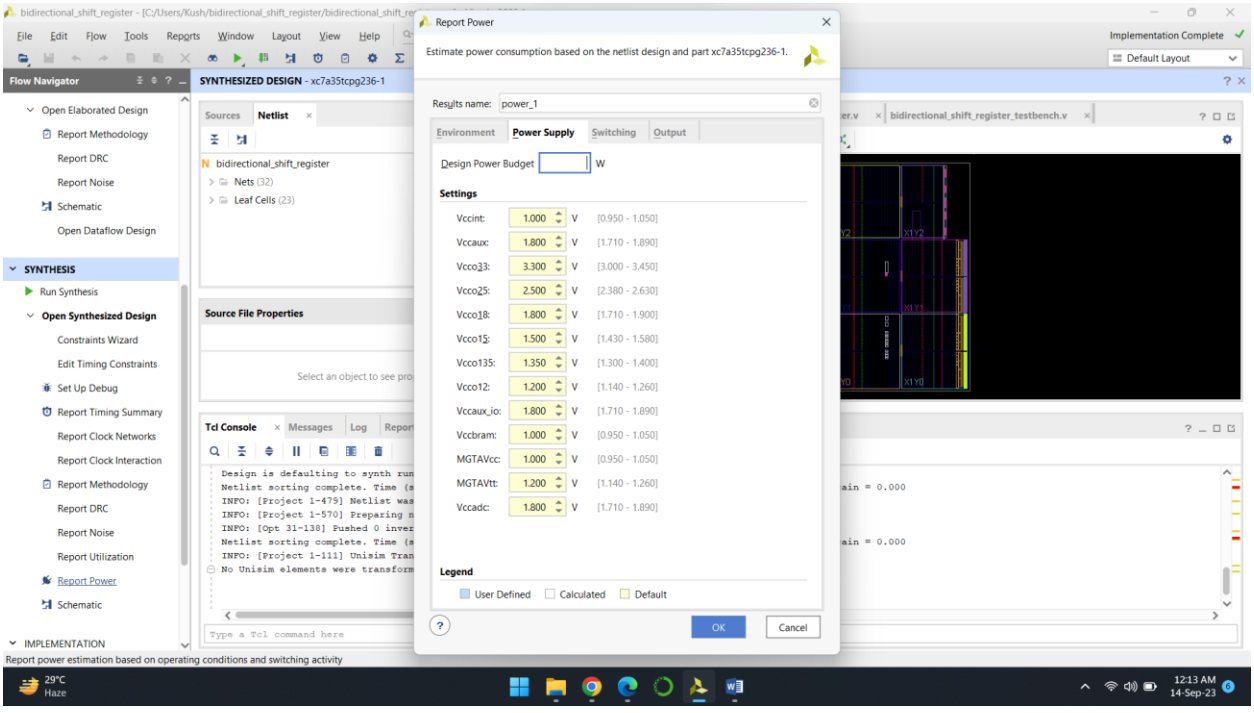



RTL SCHEMATIC:



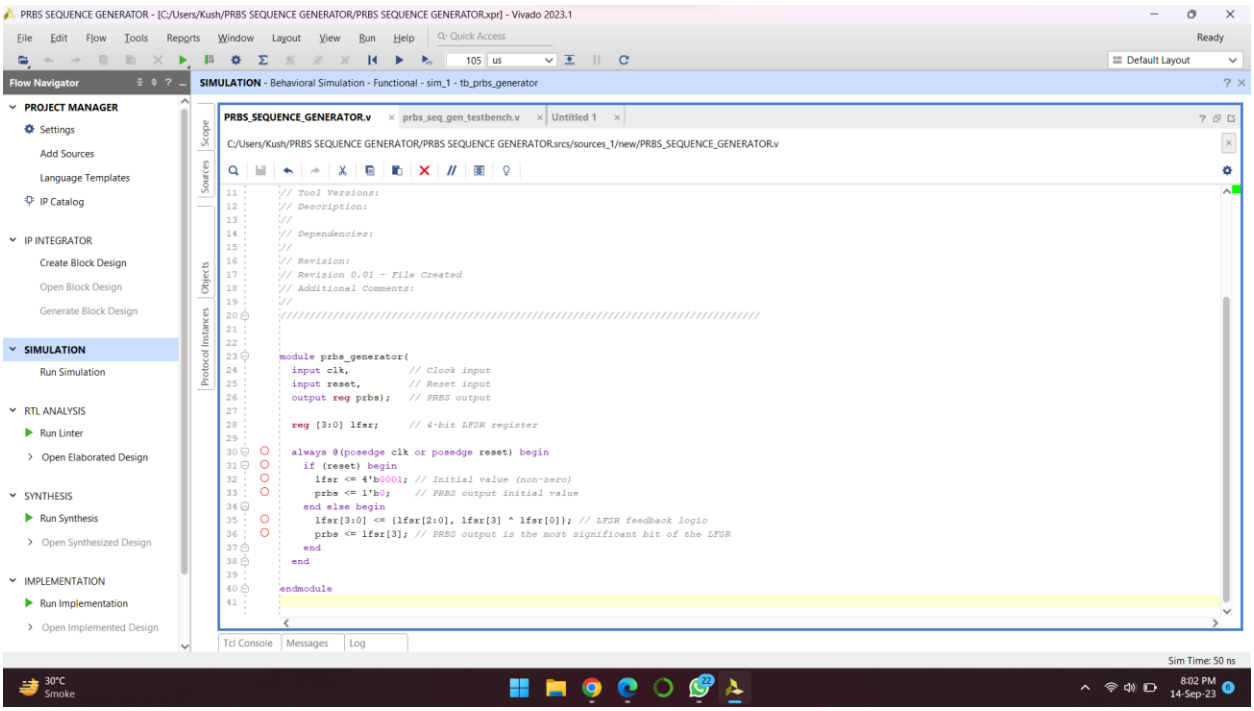
POWER REPORT:



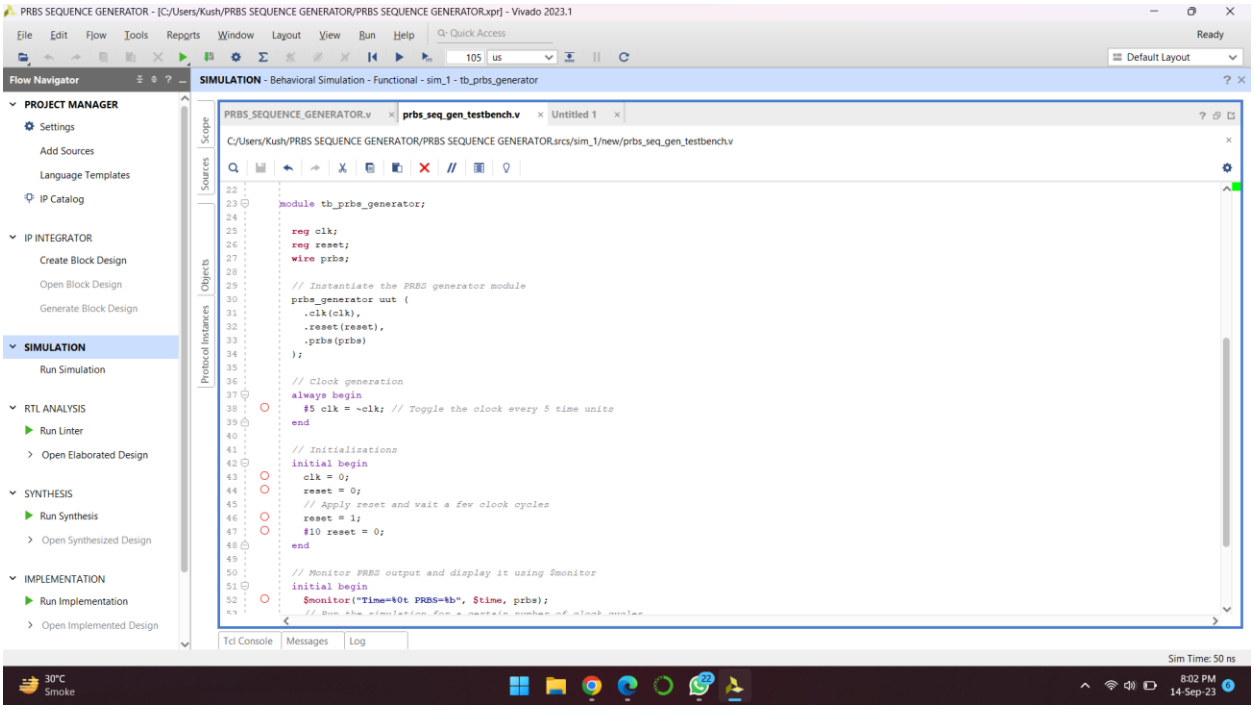



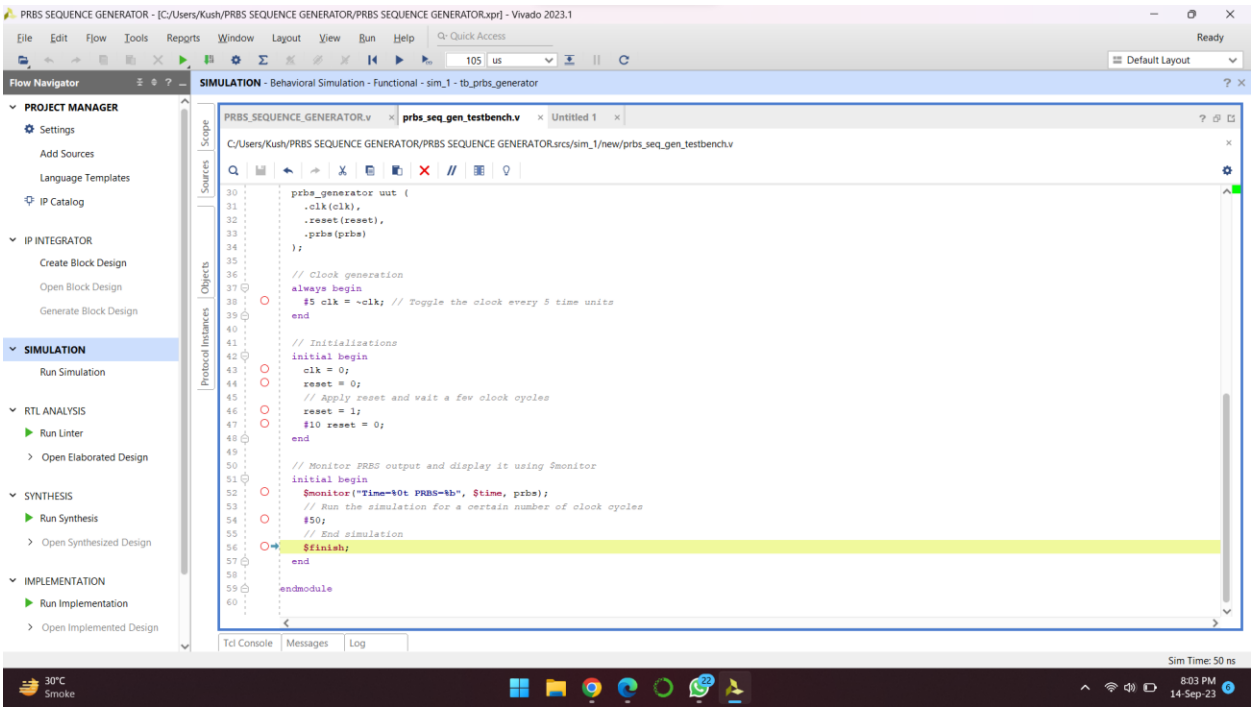

15. PRBS SEQUENCE GENERATOR:

CODE:

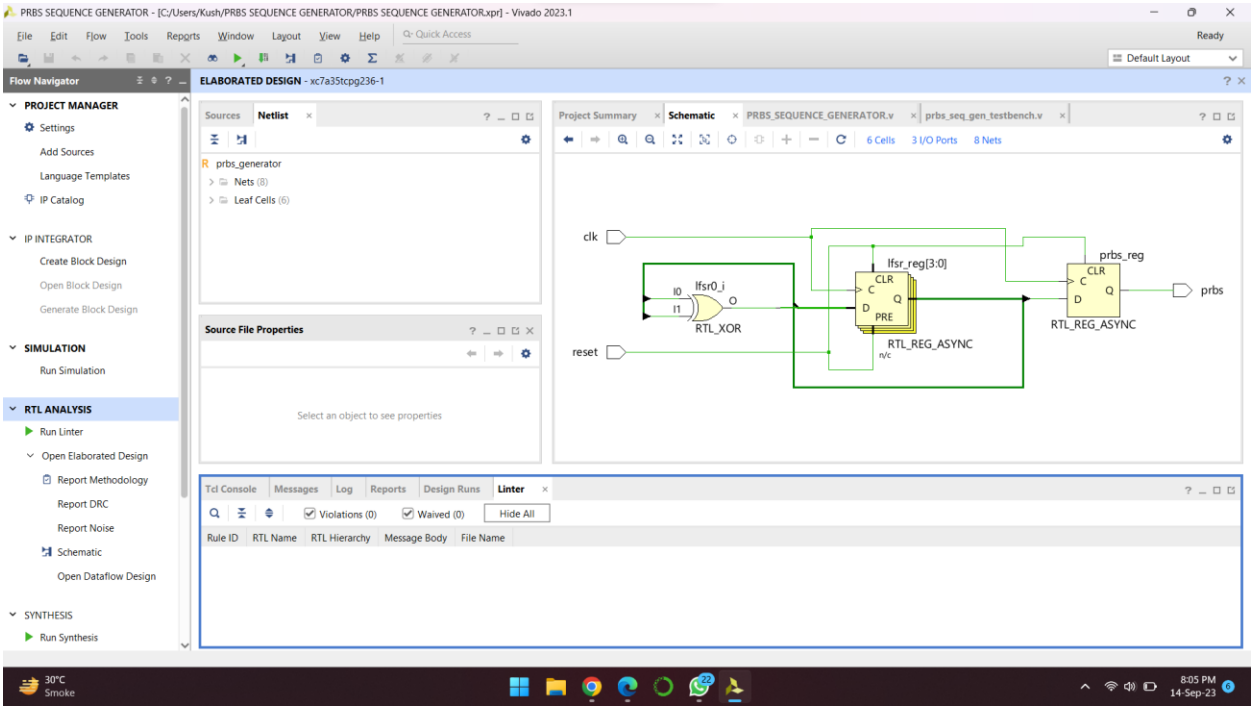

PRBS SEQUENCE GENERATOR - [C:/Users/Kush/PRBS SEQUENCE GENERATOR/PRBS SEQUENCE GENERATOR.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
SIMULATION - Behavioral Simulation - Functional - sim_1 - tb_prbs_generator
PRBS_SEQUENCE_GENERATOR.v
prbs_seq_gen_testbench.v
Untitled 1
C:/Users/Kush/PRBS SEQUENCE GENERATOR/PRBS SEQUENCE GENERATOR.srcs/sources_1/new/PRBS_SEQUENCE_GENERATOR.v
// Tool Versions:
// Description:
// Dependencies:
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
module prbs_generator(
input clk, // Clock input
input reset, // Reset input
output reg prbs); // PRBS output
reg [3:0] lfsr; // 4-bit LFSR register
always @(posedge clk or posedge reset) begin
if (reset) begin
lfsr <= 4'b0001; // Initial value (non-zero)
prbs <= 1'b0; // PRBS output initial value
end else begin
lfsr[3:0] <= (lfsr[2:0], lfsr[3] ^ lfsr[0]}; // LFSR feedback logic
prbs <= lfsr[3]; // PRBS output is the most significant bit of the LFSR
end
end
endmodule
Tcl Console
Messages
Log
Sim Time: 50 ns
30°C
Smoke
8:02 PM
14-Sep-23

TESTBENCH:

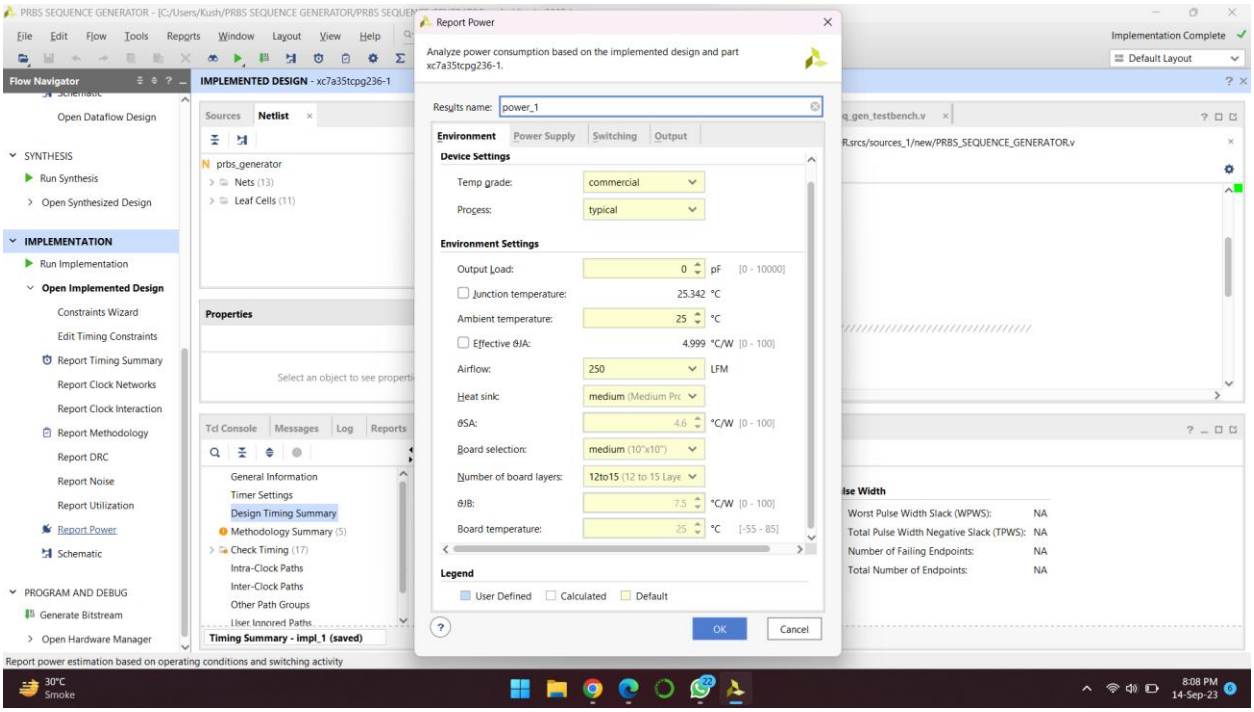

PRBS SEQUENCE GENERATOR - [C:/Users/Kush/PRBS SEQUENCE GENERATOR/PRBS SEQUENCE GENERATOR.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
SIMULATION - Behavioral Simulation - Functional - sim_1 - tb_prbs_generator
PRBS_SEQUENCE_GENERATOR.v
prbs_seq_gen_testbench.v
Untitled 1
C:/Users/Kush/PRBS SEQUENCE GENERATOR/PRBS SEQUENCE GENERATOR.srcs/sim_1/new/prbs_seq_gen_testbench.v
module tb_prbs_generator;
reg clk;
reg reset;
wire prbs;
// Instantiate the PRBS generator module
prbs_generator uut (
.clk(clk),
.reset(reset),
.prbs(prbs)
);
// Clock generation
always begin
#5 clk = ~clk; // Toggle the clock every 5 time units
end
// Initializations
initial begin
clk = 0;
reset = 0;
// Apply reset and wait a few clock cycles
reset = 1;
#10 reset = 0;
end
// Monitor PRBS output and display it using $monitor
initial begin
$monitor("Time=%0t PRBS=%b", $time, prbs);
Tcl Console
Messages
Log
Sim Time: 50 ns
30°C
Smoke
8:02 PM
14-Sep-23

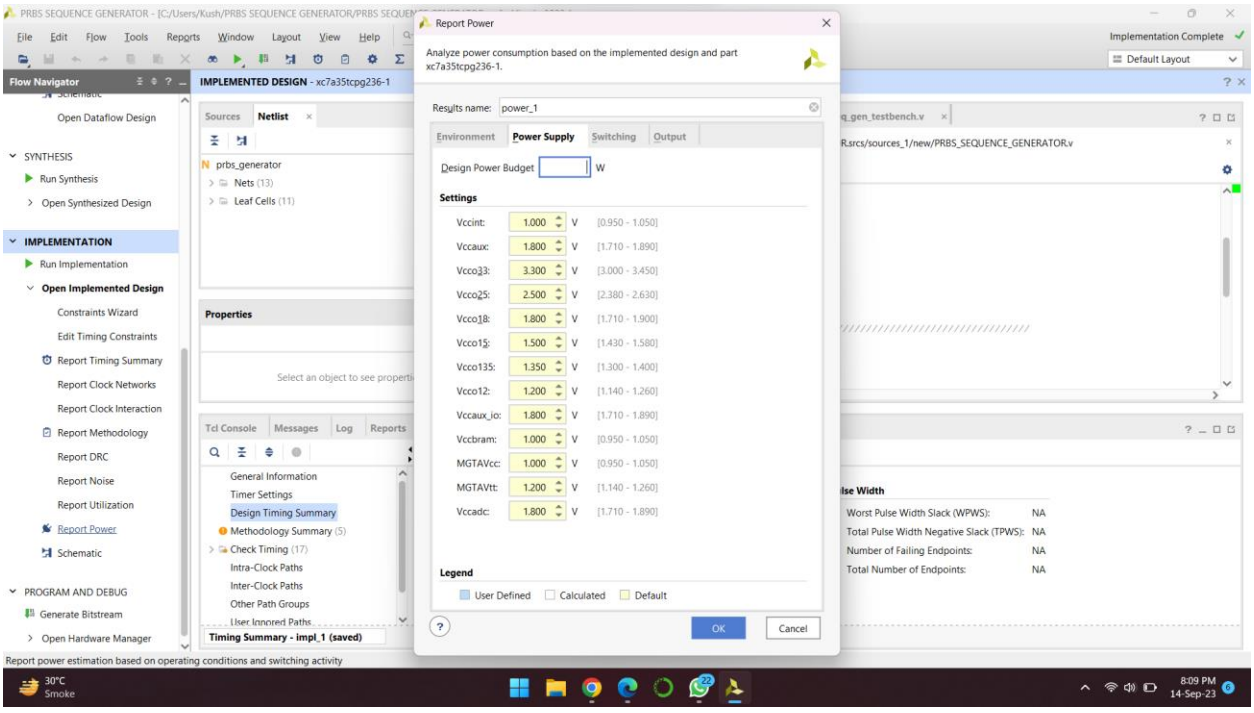



RTL SCHEMATIC:



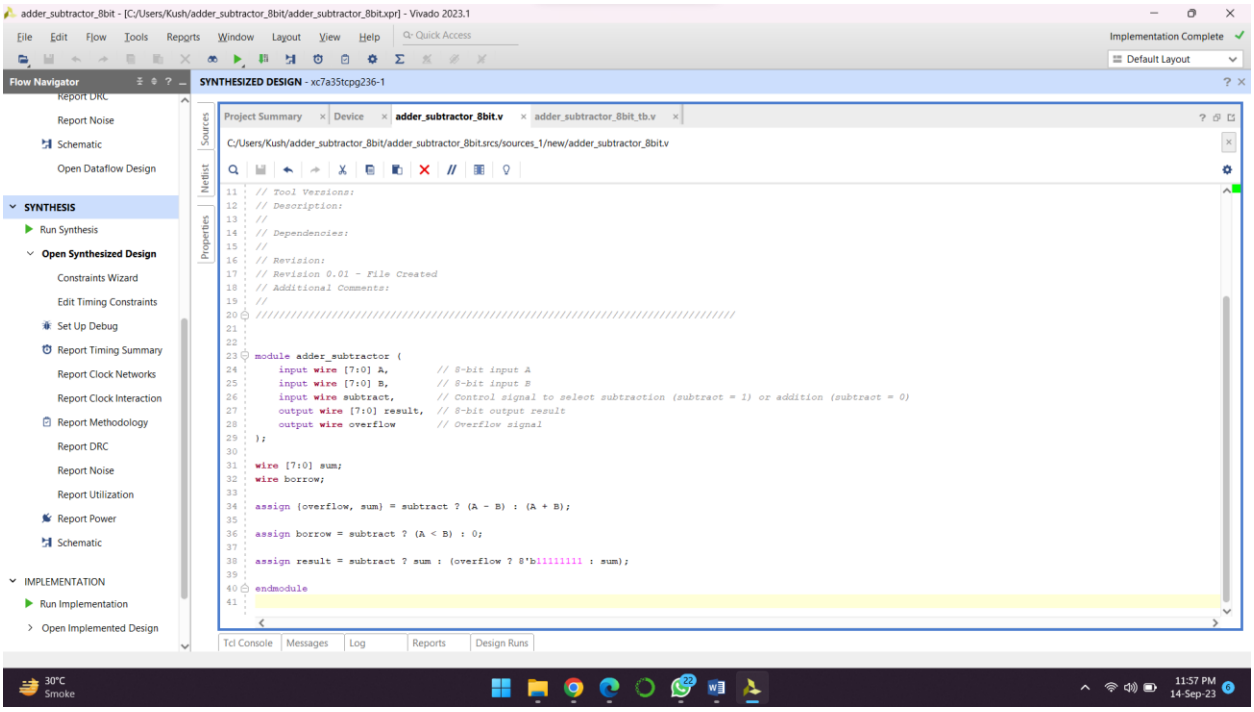
POWER REPORT:



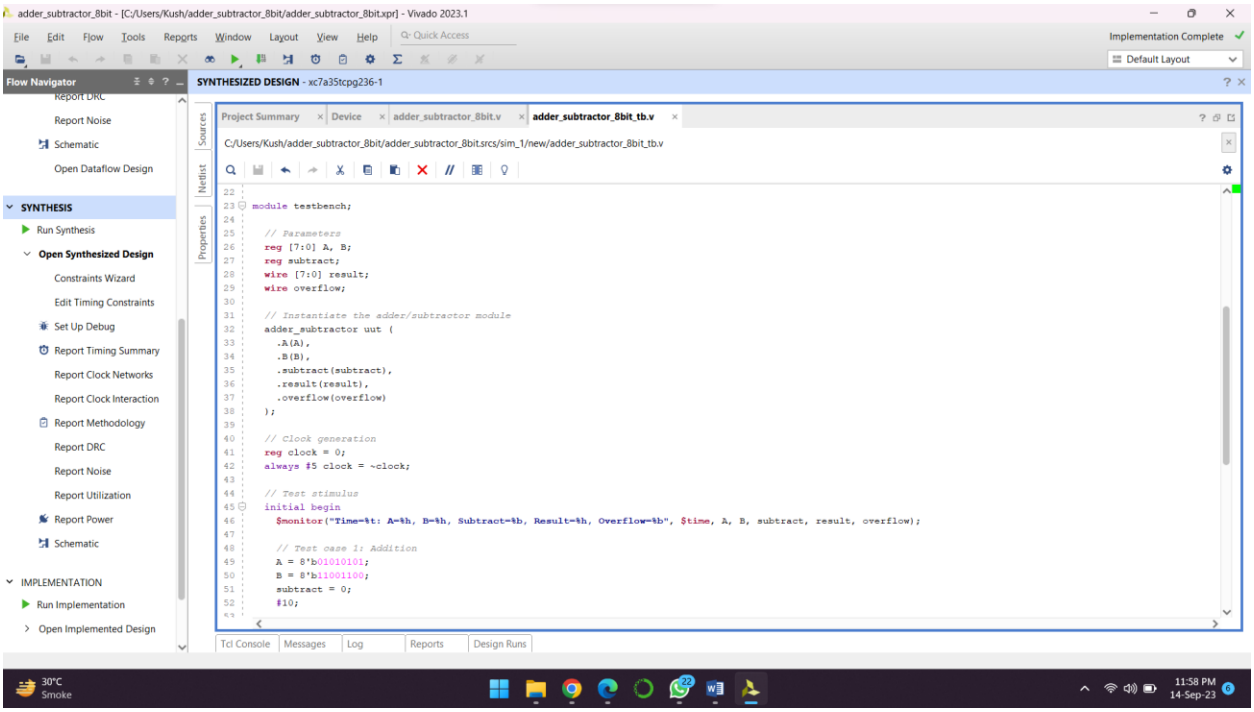

17. 8-BIT ADDER/SUBTRACTOR:

CODE:

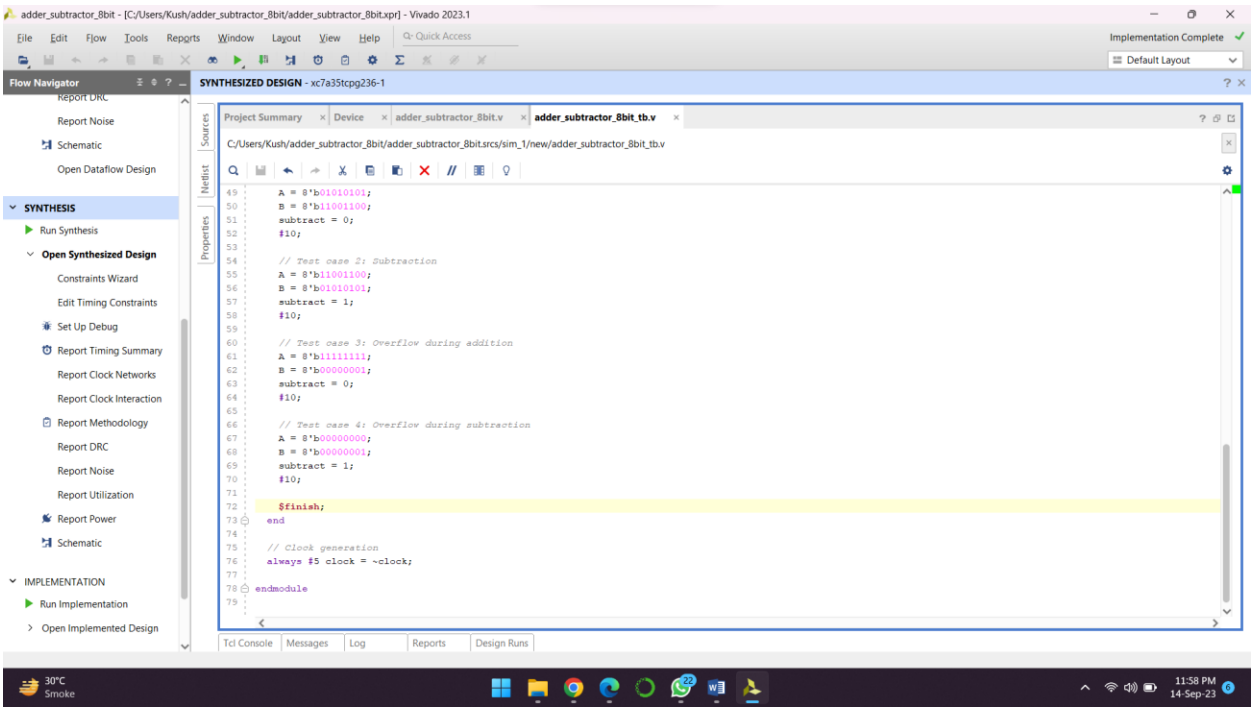



```
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module adder_subtractor (
    input wire [7:0] A,        // 8-bit input A
    input wire [7:0] B,        // 8-bit input B
    input wire subtract,       // Control signal to select subtraction (subtract = 1) or addition (subtract = 0)
    output wire [7:0] result,  // 8-bit output result
    output wire overflow       // Overflow signal
);

wire [7:0] sum;
wire borrow;

assign {overflow, sum} = subtract ? (A - B) : (A + B);

assign borrow = subtract ? (A < B) : 0;

assign result = subtract ? sum : (overflow ? 8'b11111111 : sum);

endmodule
```

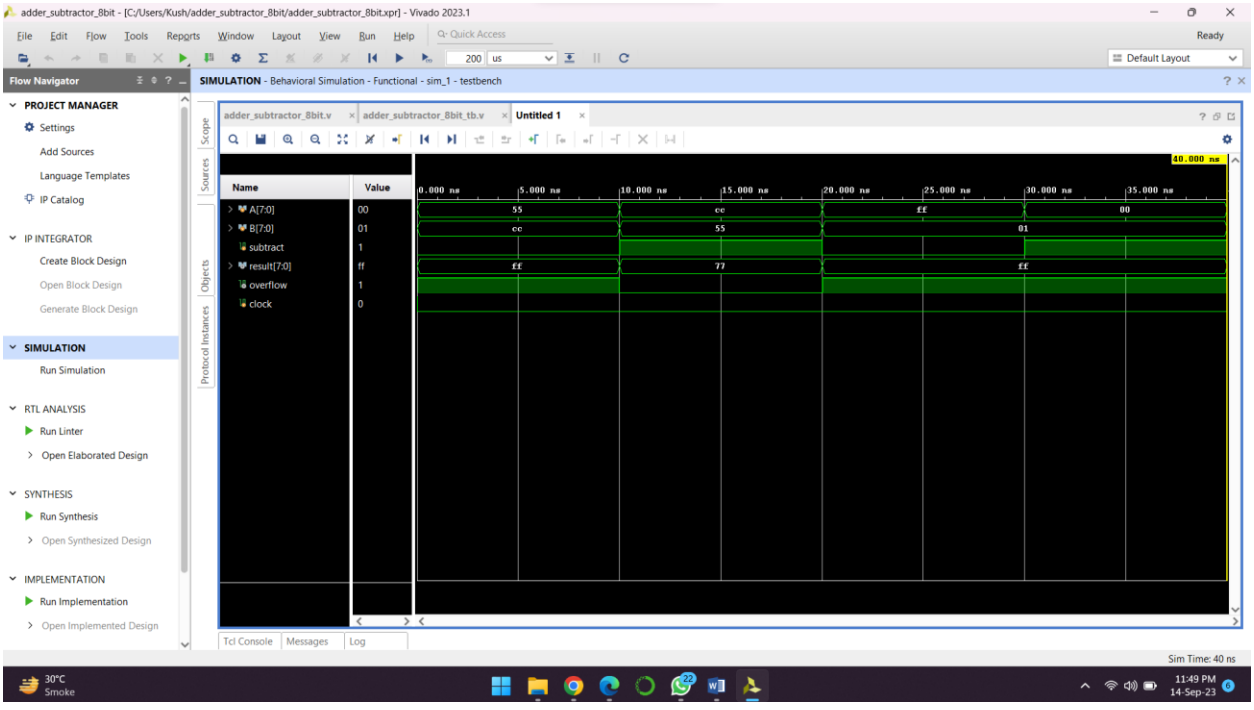
TESTBENCH:



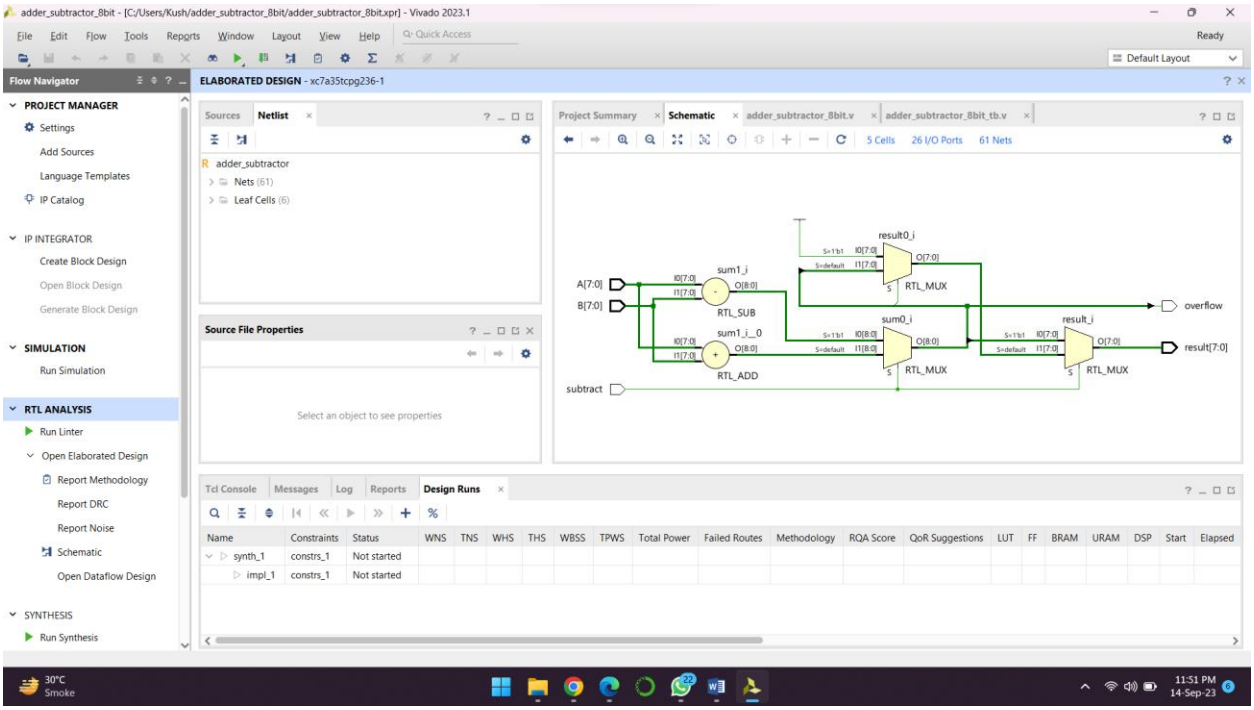

```
module testbench;

  // Parameters
  reg [7:0] A, B;
  reg subtract;
  wire [7:0] result;
  wire overflow;

  // Instantiate the adder/subtractor module
  adder_subtractor uut (
    .A(A),
    .B(B),
    .subtract(subtract),
    .result(result),
    .overflow(overflow)
  );

  // Clock generation
  reg clock = 0;
  always #5 clock = ~clock;

  // Test stimulus
  initial begin
    $monitor("Time=%t: A=%h, B=%h, Subtract=%b, Result=%h, Overflow=%b", $time, A, B, subtract, result, overflow);

    // Test case 1: Addition
    A = 8'b01010101;
    B = 8'b11001100;
    subtract = 0;
    #10;
```


adder_subtractor_8bit - [C:/Users/Kush/adder_subtractor_8bit/adder_subtractor_8bit.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Implementation Complete
Default Layout
Flow Navigator
SYNTHESIZED DESIGN - xc7a35tcpg236-1
Project Summary
Device
adder_subtractor_8bit.v
adder_subtractor_8bit_tb.v
C:/Users/Kush/adder_subtractor_8bit/adder_subtractor_8bit.srcs/sim_1/new/adder_subtractor_8bit_tb.v
A = 8'b01010101;
B = 8'b11001100;
subtract = 0;
#10;
// Test case 2: Subtraction
A = 8'b11001100;
B = 8'b01010101;
subtract = 1;
#10;
// Test case 3: Overflow during addition
A = 8'b11111111;
B = 8'b00000001;
subtract = 0;
#10;
// Test case 4: Overflow during subtraction
A = 8'b00000000;
B = 8'b00000001;
subtract = 1;
#10;
$finish;
end
// Clock generation
always #5 clock = ~clock;
endmodule
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Run Implementation
Open Implemented Design
Tcl Console
Messages
Log
Reports
Design Runs
30°C
Smoke
11:58 PM
14-Sep-23

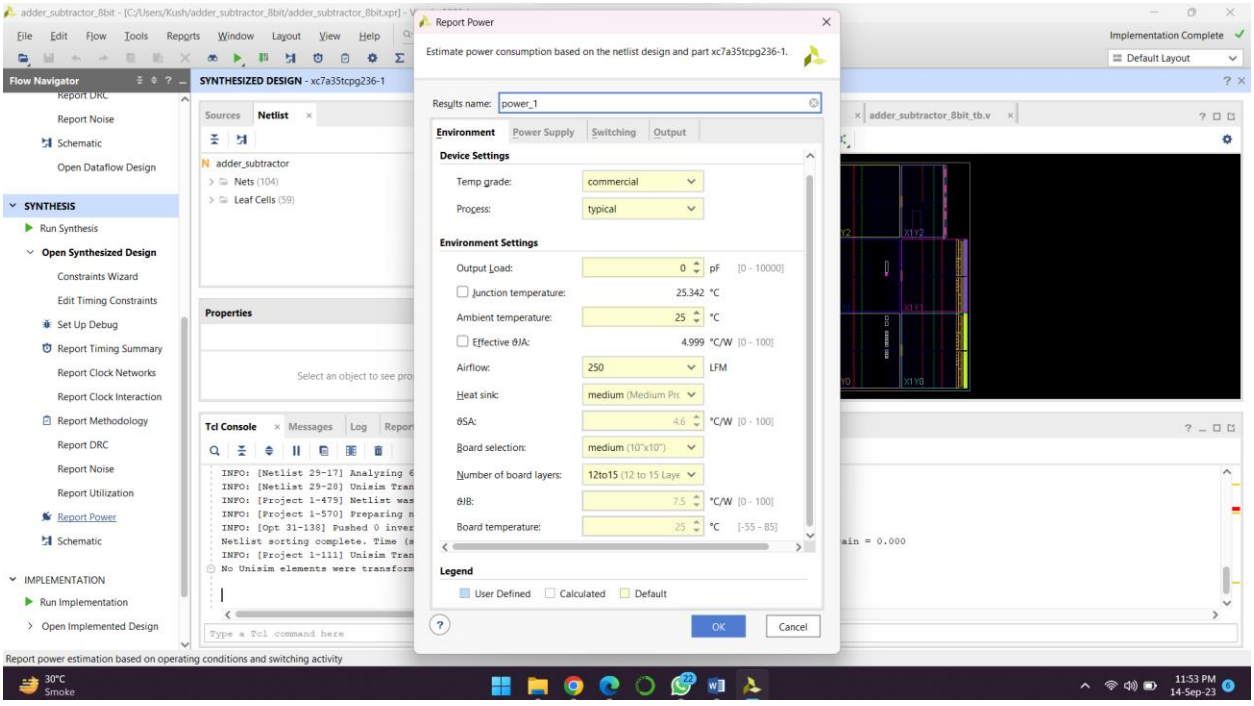
WAVEFORM:

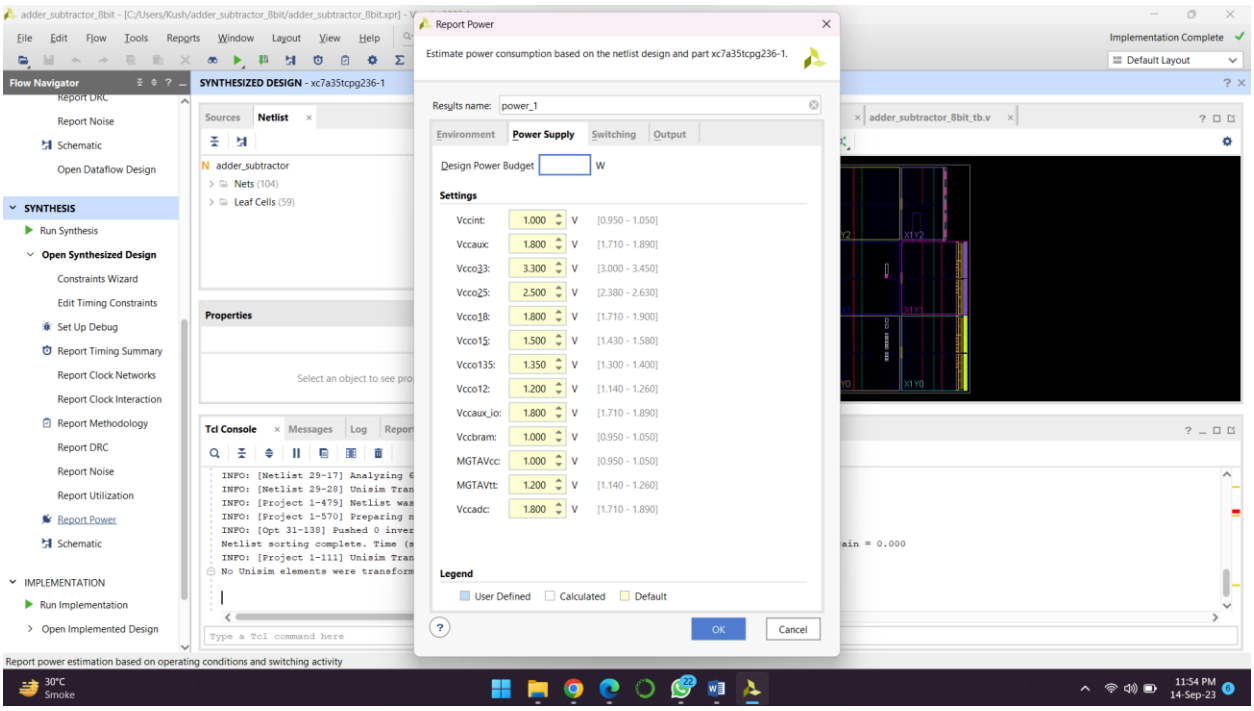

adder_subtractor_8bit - [C:/Users/Kush/adder_subtractor_8bit/adder_subtractor_8bit.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench
adder_subtractor_8bit.v
adder_subtractor_8bit_tb.v
Untitled 1
Name
Value
A[7:0]
B[7:0]
subtract
result[7:0]
overflow
clock
00
01
1
ff
1
0
0.000 ns
5.000 ns
10.000 ns
15.000 ns
20.000 ns
25.000 ns
30.000 ns
35.000 ns
40.000 ns
55
ee
ff
00
cc
55
01
ff
77
ff
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
Tcl Console
Messages
Log
Sim Time: 40 ns
30°C
Smoke
11:49 PM
14-Sep-23

RTL SCHMETIC:



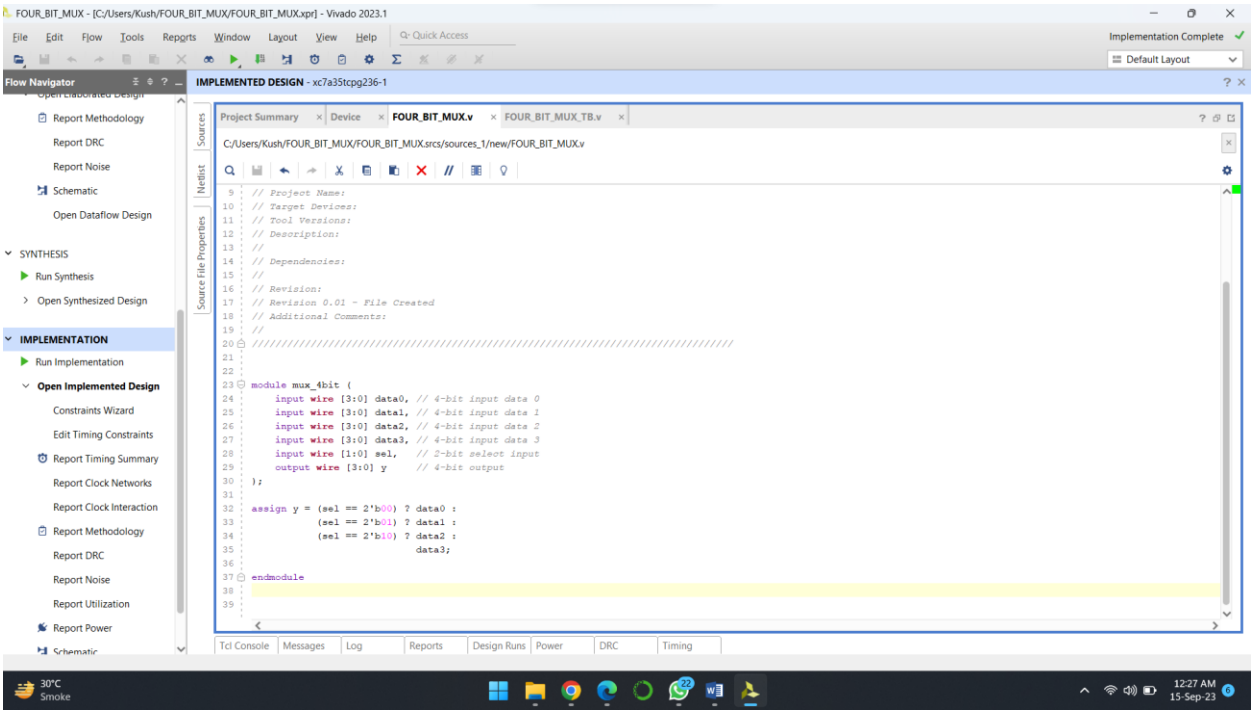

POWER REPORT:



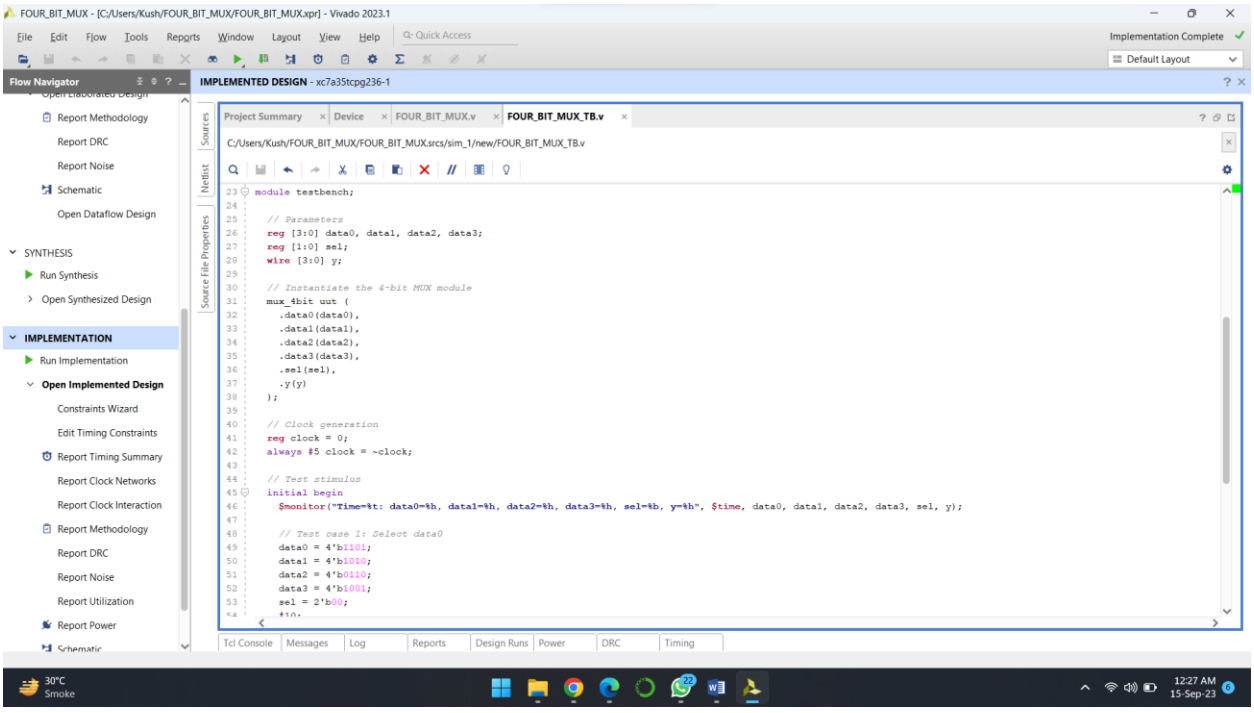


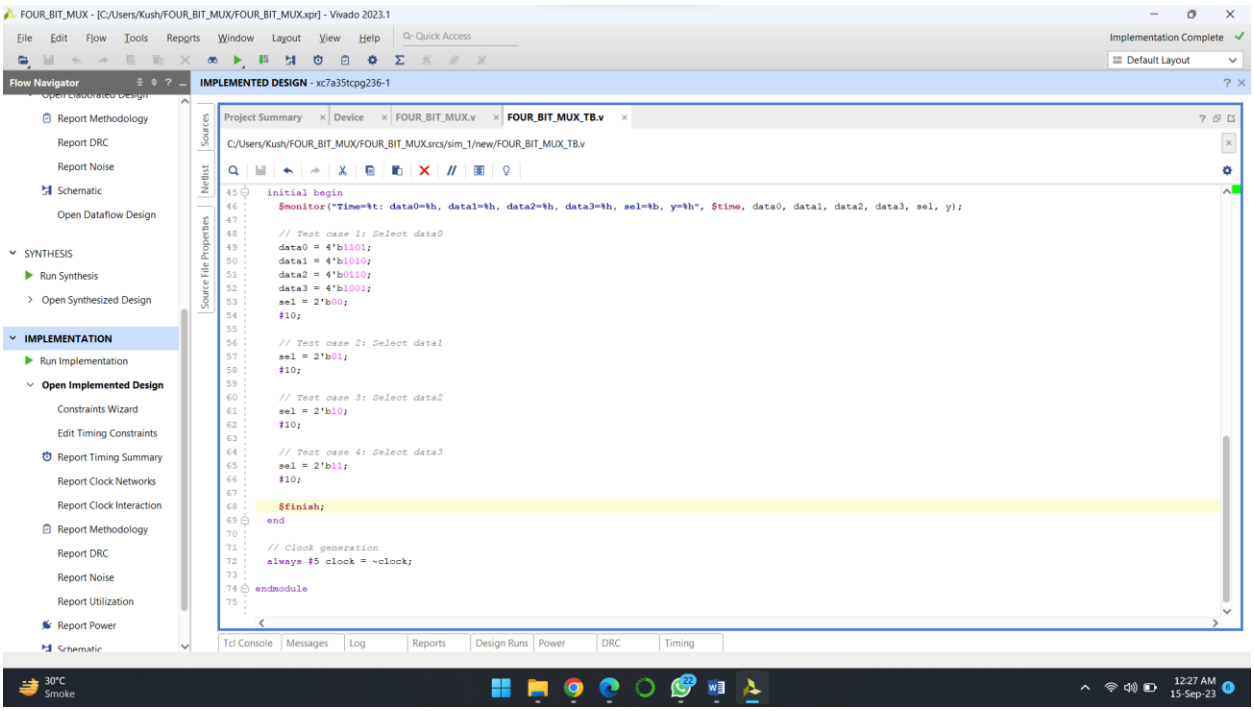

18. 4-BIT MULTIPLIER:

CODE:

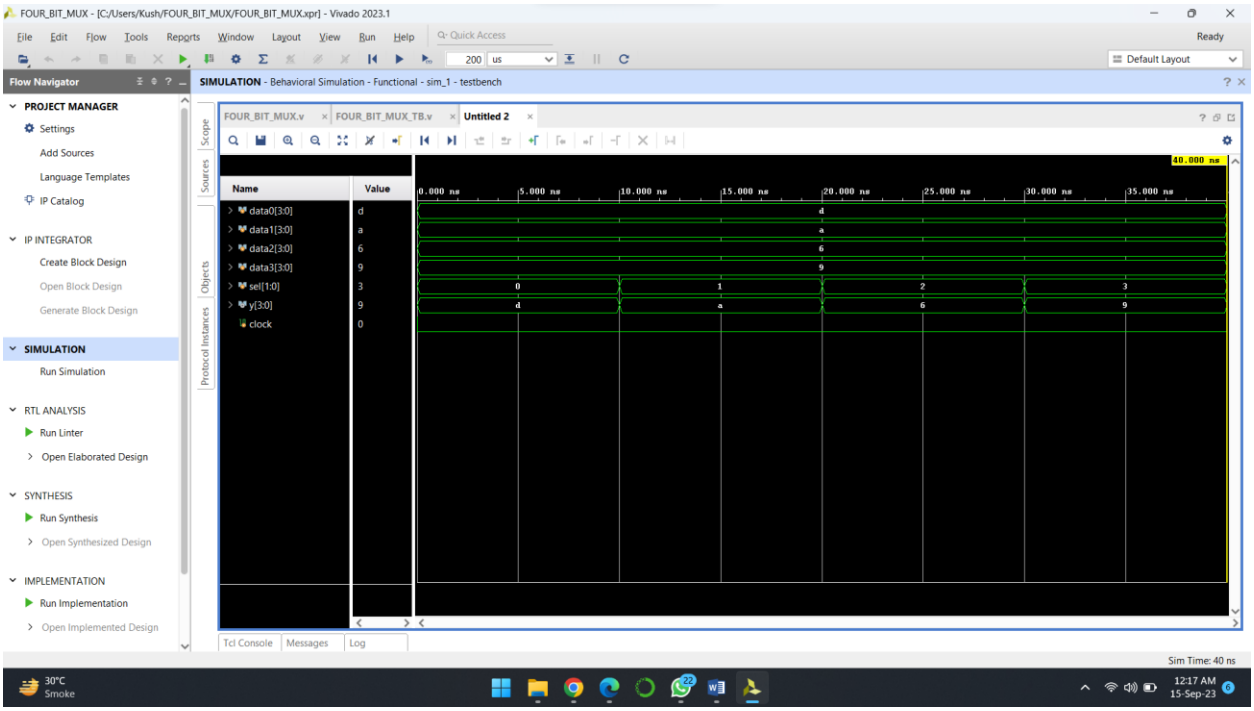

FOUR_BIT_MUX - [C:/Users/Kush/FOUR_BIT_MUX/FOUR_BIT_MUX.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Implementation Complete
Default Layout
Flow Navigator
IMPLEMENTED DESIGN - xc7a35tcpg236-1
Report Methodology
Report DRC
Report Noise
Schematic
Open Dataflow Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
Constraints Wizard
Edit Timing Constraints
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Project Summary
Device
FOUR_BIT_MUX.v
FOUR_BIT_MUX_TB.v
C:/Users/Kush/FOUR_BIT_MUX/FOUR_BIT_MUX.srcs/sources_1/new/FOUR_BIT_MUX.v
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
module mux_4bit (
input wire [3:0] data0, // 4-bit input data 0
input wire [3:0] data1, // 4-bit input data 1
input wire [3:0] data2, // 4-bit input data 2
input wire [3:0] data3, // 4-bit input data 3
input wire [1:0] sel, // 2-bit select input
output wire [3:0] y // 4-bit output
);
assign y = (sel == 2'b00) ? data0 :
(sel == 2'b01) ? data1 :
(sel == 2'b10) ? data2 :
data3;
endmodule
Tcl Console
Messages
Log
Reports
Design Runs
Power
DRC
Timing
30°C
Smoke
12:27 AM
15-Sep-23

TESTBENCH:

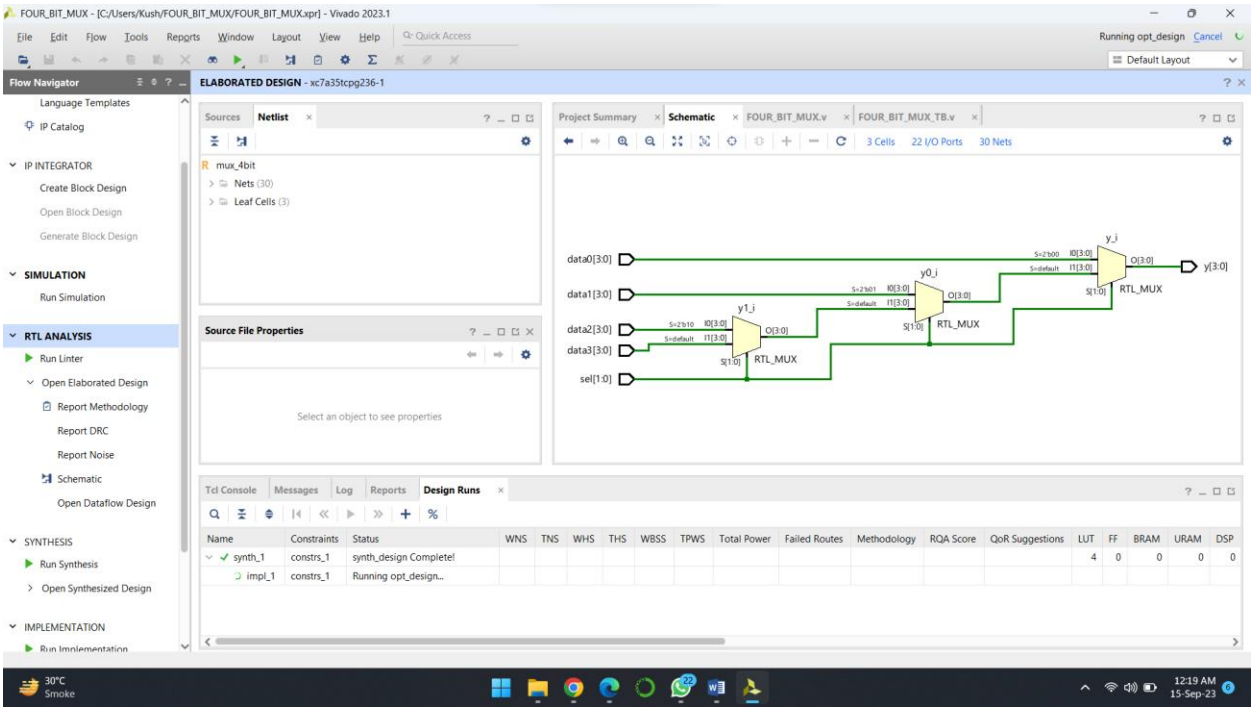

FOUR_BIT_MUX - [C:/Users/Kush/FOUR_BIT_MUX/FOUR_BIT_MUX.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Implementation Complete
Default Layout
Flow Navigator
IMPLEMENTED DESIGN - xc7a35tcpg236-1
Report Methodology
Report DRC
Report Noise
Schematic
Open Dataflow Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
Constraints Wizard
Edit Timing Constraints
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Project Summary
Device
FOUR_BIT_MUX.v
FOUR_BIT_MUX_TB.v
C:/Users/Kush/FOUR_BIT_MUX/FOUR_BIT_MUX.srcs/sim_1/new/FOUR_BIT_MUX_TB.v
module testbench;
// Parameters
reg [3:0] data0, data1, data2, data3;
reg [1:0] sel;
wire [3:0] y;
// Instantiate the 4-bit MUX module
mux_4bit uut (
.data0(data0),
.data1(data1),
.data2(data2),
.data3(data3),
.sel(sel),
.y(y)
);
// Clock generation
reg clock = 0;
always #5 clock = ~clock;
// Test stimulus
initial begin
$monitor("Time=%t: data0=%h, data1=%h, data2=%h, data3=%h, sel=%b, y=%h", $time, data0, data1, data2, data3, sel, y);
// Test case 1: Select data0
data0 = 4'b1101;
data1 = 4'b1010;
data2 = 4'b0110;
data3 = 4'b1001;
sel = 2'b00;
Tcl Console
Messages
Log
Reports
Design Runs
Power
DRC
Timing
30°C
Smoke
12:27 AM
15-Sep-23


FOUR_BIT_MUX - [C:/Users/Kush/FOUR_BIT_MUX/FOUR_BIT_MUX.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Implementation Complete
Default Layout
Flow Navigator
IMPLEMENTED DESIGN - xc7a35tcpg236-1
Report Methodology
Report DRC
Report Noise
Schematic
Open Dataflow Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
Constraints Wizard
Edit Timing Constraints
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Project Summary
Device
FOUR_BIT_MUX.v
FOUR_BIT_MUX_TB.v
C:/Users/Kush/FOUR_BIT_MUX/FOUR_BIT_MUX.srcs/sim_1/new/FOUR_BIT_MUX_TB.v
initial begin
$monitor("Time=%t: data0=%h, data1=%h, data2=%h, data3=%h, sel=%b, y=%h", $time, data0, data1, data2, data3, sel, y);
// Test case 1: Select data0
data0 = 4'b1101;
data1 = 4'b1010;
data2 = 4'b0110;
data3 = 4'b1001;
sel = 2'b00;
#10;
// Test case 2: Select data1
sel = 2'b01;
#10;
// Test case 3: Select data2
sel = 2'b10;
#10;
// Test case 4: Select data3
sel = 2'b11;
#10;
$finish;
end
// Clock generation
always #5 clock = ~clock;
endmodule
Tcl Console
Messages
Log
Reports
Design Runs
Power
DRC
Timing
30°C
Smoke
12:27 AM
15-Sep-23

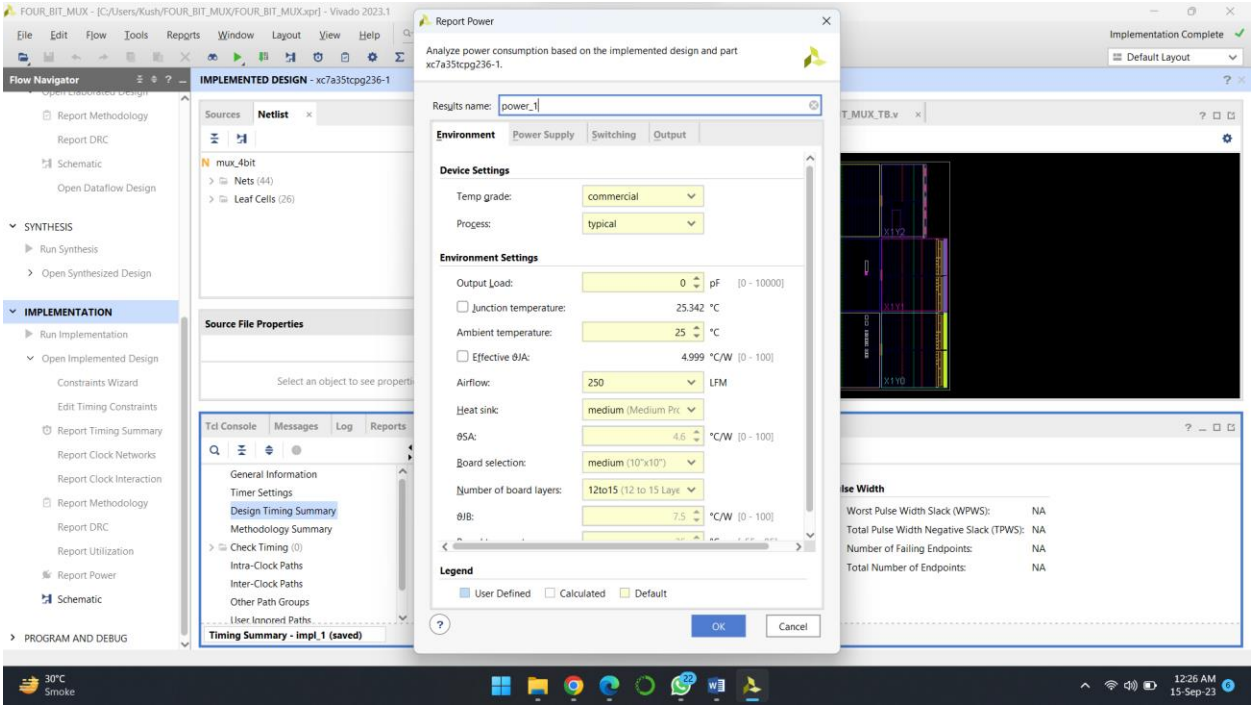
WAVEFORM:

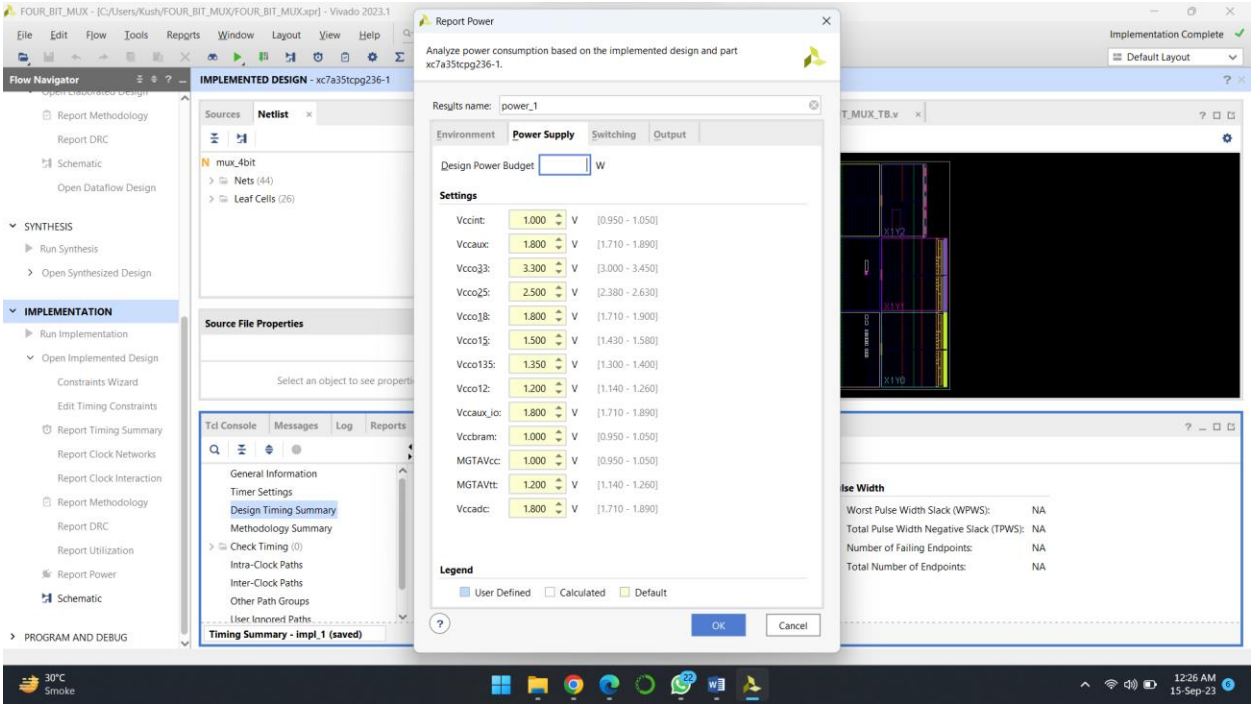

FOUR_BIT_MUX - [C:/Users/Kush/FOUR_BIT_MUX/FOUR_BIT_MUX.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
FOUR_BIT_MUX.v
FOUR_BIT_MUX_TB.v
Untitled 2
Name
Value
data0[3:0]
data1[3:0]
data2[3:0]
data3[3:0]
sel[1:0]
y[3:0]
clock
0.000 ns
5.000 ns
10.000 ns
15.000 ns
20.000 ns
25.000 ns
30.000 ns
35.000 ns
40.000 ns
Tcl Console
Messages
Log
Sim Time: 40 ns
30°C
Smoke
12:17 AM
15-Sep-23

RTL SCHEMATIC:



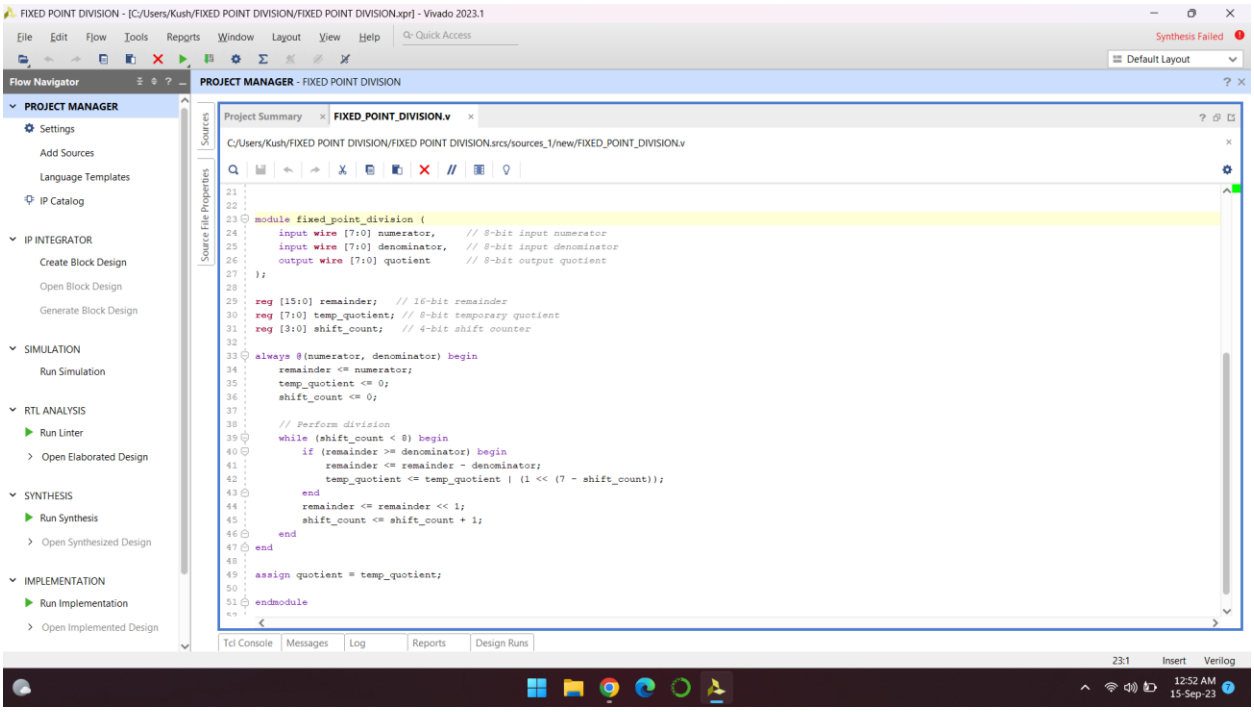

POWER REPORT:



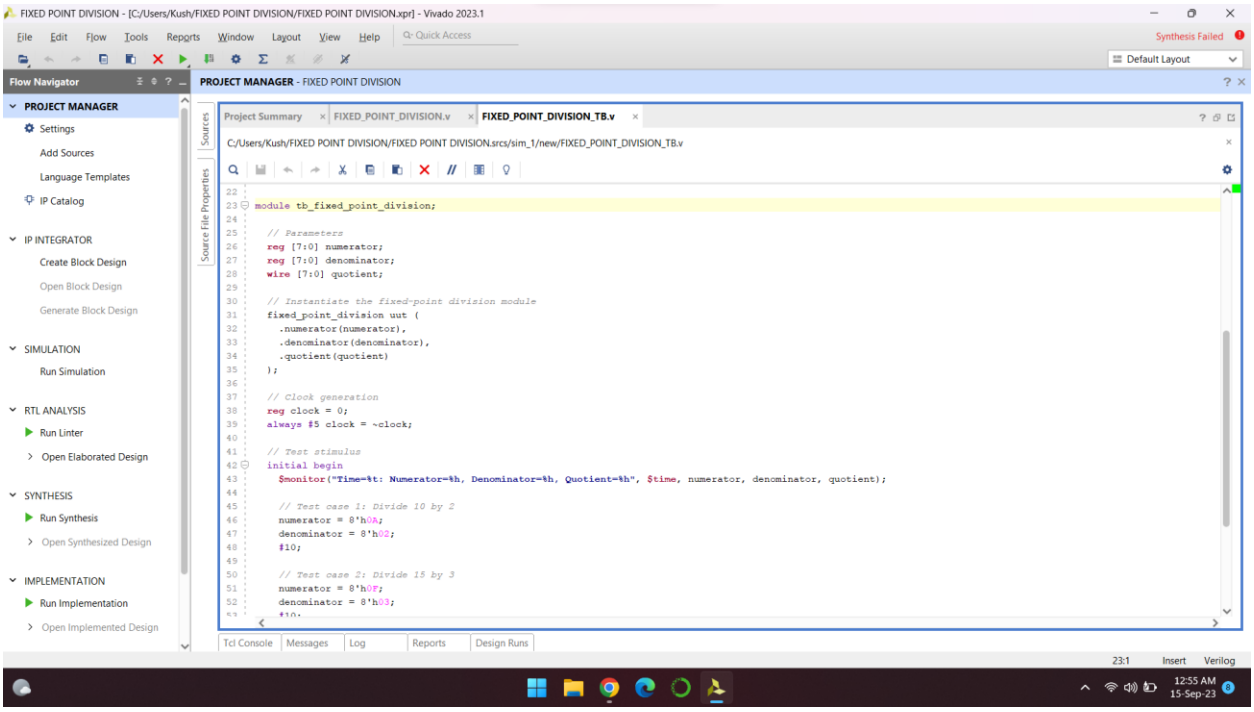



## 19. FIXED POINT DIVISION:

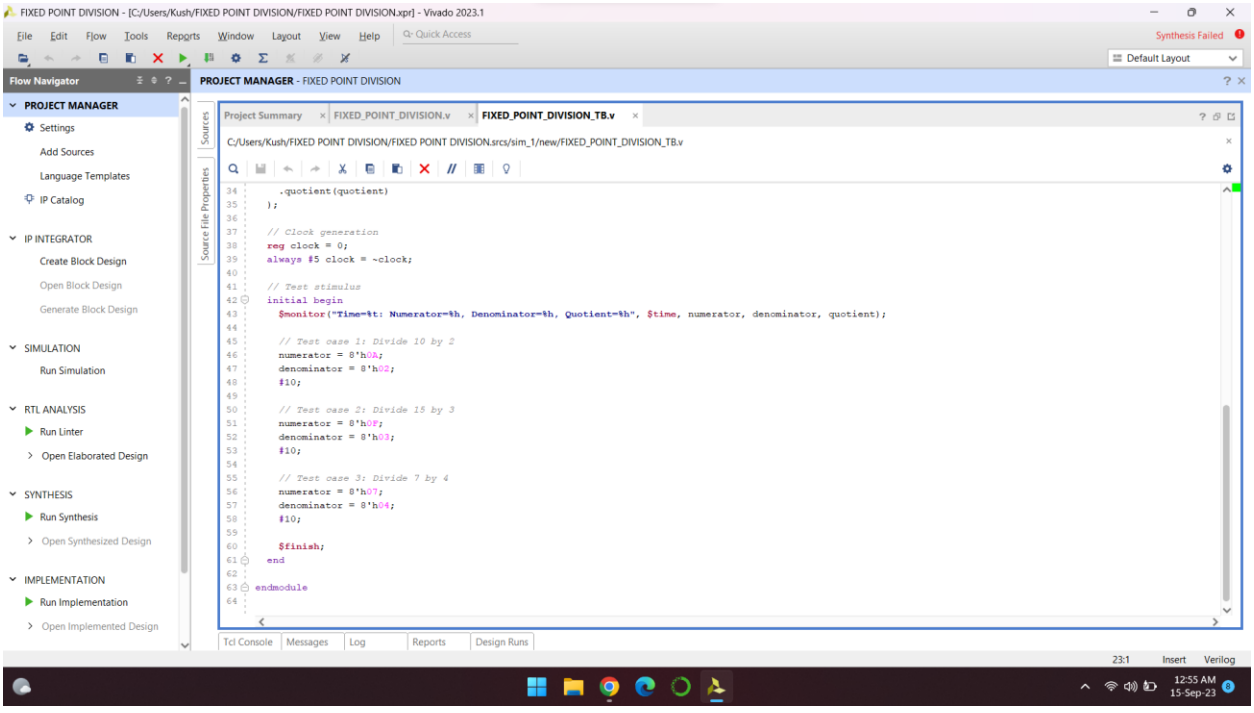
CODE:



TESTBENCH:

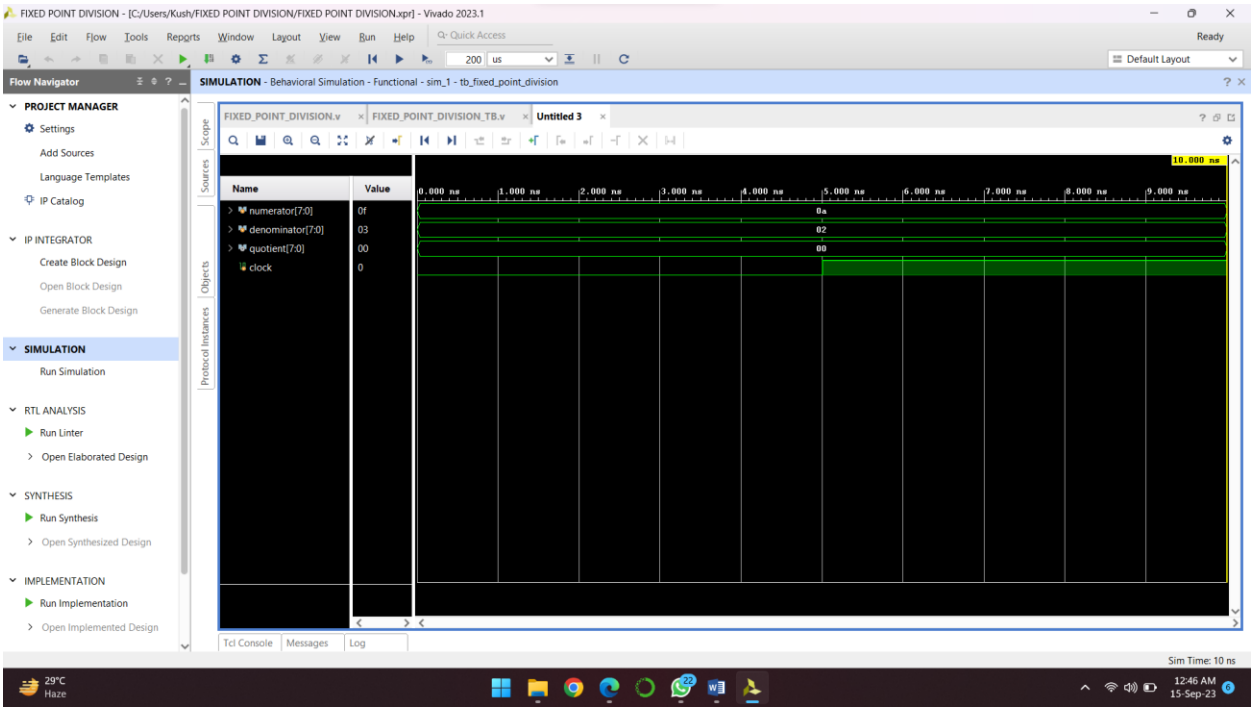



```
module tb_fixed_point_division;

  // Parameters
  reg [7:0] numerator;
  reg [7:0] denominator;
  wire [7:0] quotient;

  // Instantiate the fixed-point division module
  fixed_point_division uut (
    .numerator(numerator),
    .denominator(denominator),
    .quotient(quotient)
  );

  // Clock generation
  reg clock = 0;
  always #5 clock = ~clock;

  // Test stimulus
  initial begin
    $monitor("Time=%t: Numerator=%h, Denominator=%h, Quotient=%h", $time, numerator, denominator, quotient);

    // Test case 1: Divide 10 by 2
    numerator = 8'h0A;
    denominator = 8'h02;
    #10;

    // Test case 2: Divide 15 by 3
    numerator = 8'h0F;
    denominator = 8'h03;
```

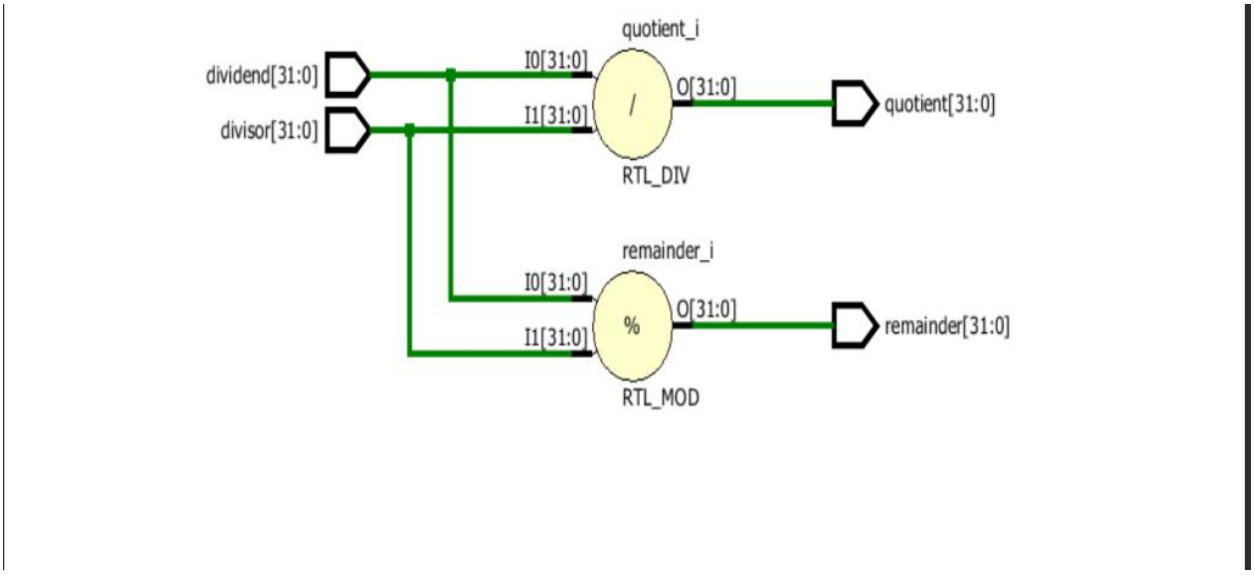



```
    .quotient(quotient)
  );

  // Clock generation
  reg clock = 0;
  always #5 clock = ~clock;

  // Test stimulus
  initial begin
    $monitor("Time=%t: Numerator=%h, Denominator=%h, Quotient=%h", $time, numerator, denominator, quotient);

    // Test case 1: Divide 10 by 2
    numerator = 8'h0A;
    denominator = 8'h02;
    #10;

    // Test case 2: Divide 15 by 3
    numerator = 8'h0F;
    denominator = 8'h03;
    #10;

    // Test case 3: Divide 7 by 4
    numerator = 8'h07;
    denominator = 8'h04;
    #10;

    $finish;
  end

endmodule
```

WAVEFORM:



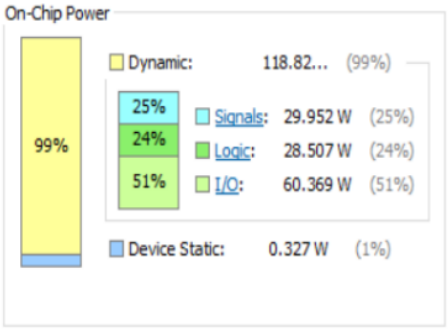

RTL SCHEMATIC:




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 ϑJA: | 4.8 °C/W |
| Power supplied to off-chip devices: | 0 W |
| Confidence level: | Low |

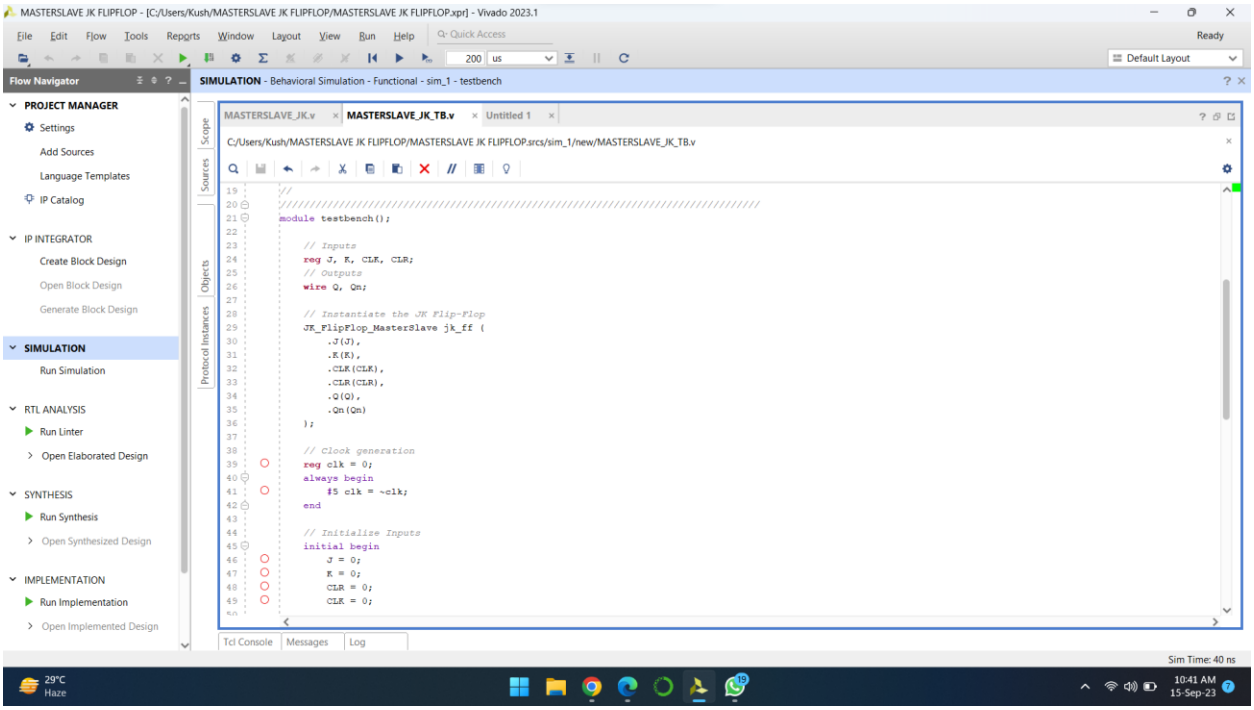


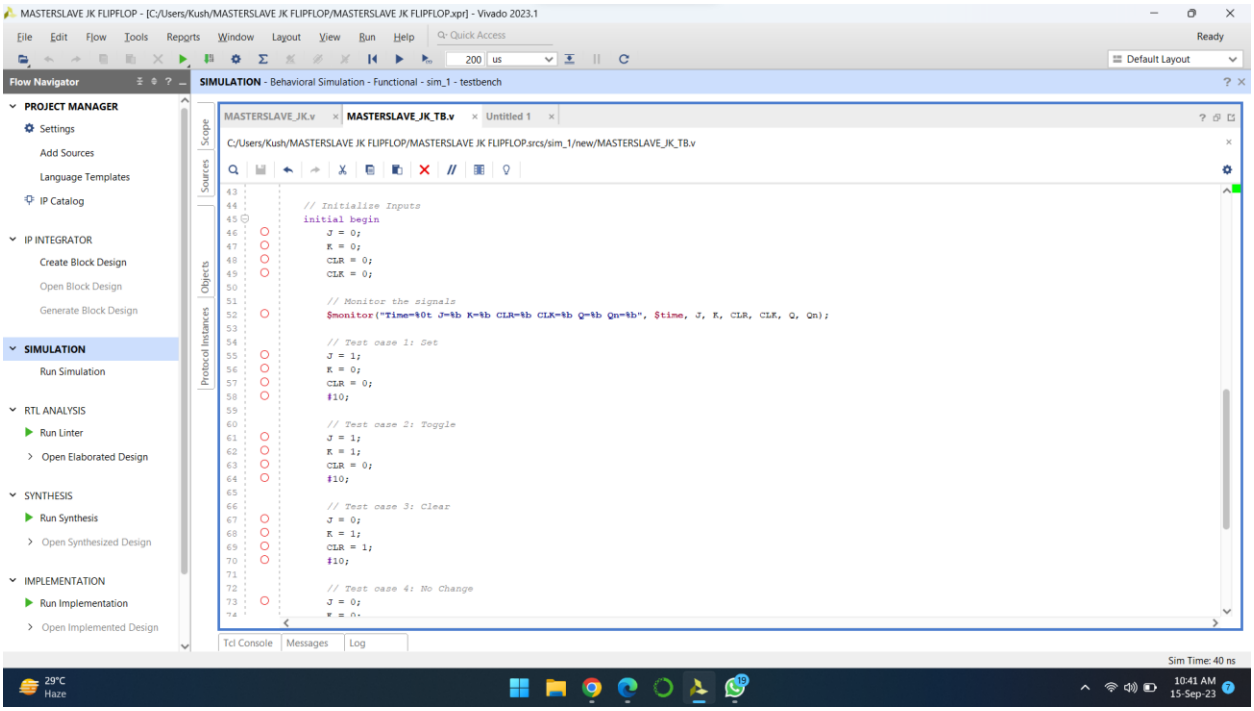

## 20. MASTER SLAVE JK FLIP FLOP:

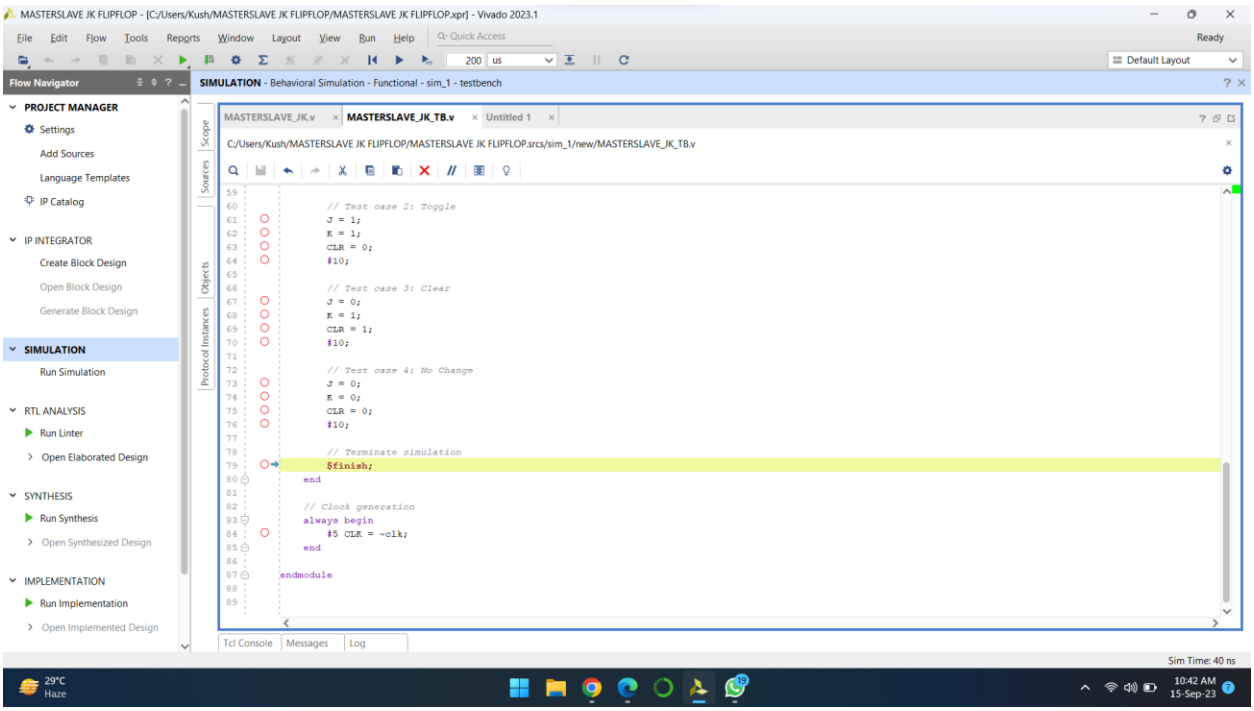
CODE:

```
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module JK_FlipFlop_MasterSlave (
    input wire J, // Input J
    input wire K, // Input K
    input wire CLK, // Clock input
    input wire CLR, // Clear input
    output wire Q, // Output Q
    output wire Qn // Output Q' (complement of Q)
);

    reg Q_next; // Next state of Q

    always @(posedge CLK or posedge CLR) begin
        if (CLR) begin
            Q_next <= 1'b0; // Clear the flip-flop
        end else begin
            case({J, K})
                2'b00: Q_next <= Q_next; // No change
                2'b01: Q_next <= 1'b0; // Clear (K=1)
                2'b10: Q_next <= 1'b1; // Set (J=1)
                2'b11: Q_next <= ~Q_next; // Toggle (J=K=1)
            endcase
        end
    end

    assign Q = Q_next;
    assign Qn = ~Q_next;

endmodule
```

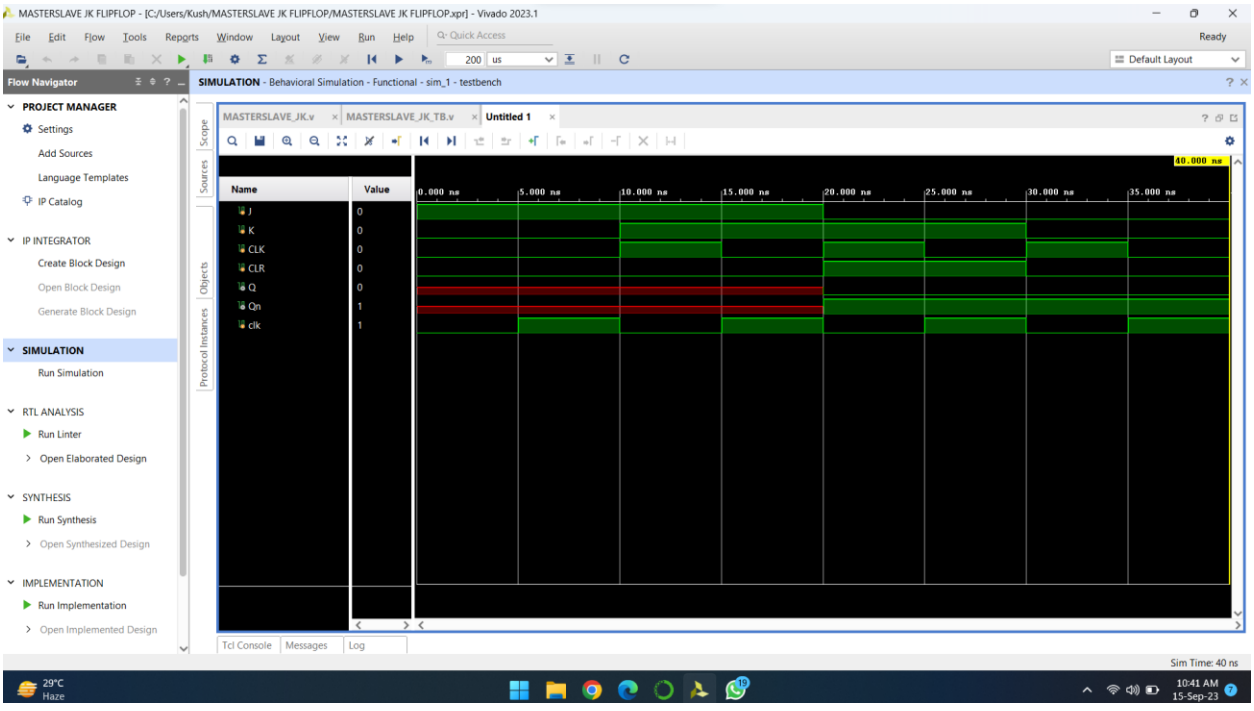
TESTBENCH:


MASTERSLAVE JK FLIPFLOP - [C:/Users/Kush/MASTERSLAVE JK FLIPFLOP/MASTERSLAVE JK FLIPFLOP.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
MASTERSLAVE_JK.v
MASTERSLAVE_JK_TB.v
Untitled 1
C:/Users/Kush/MASTERSLAVE JK FLIPFLOP/MASTERSLAVE JK FLIPFLOP.srcs/sim_1/new/MASTERSLAVE_JK_TB.v
Tcl Console
Messages
Log
Sim Time: 40 ns
29°C
Haze
10:41 AM
15-Sep-23

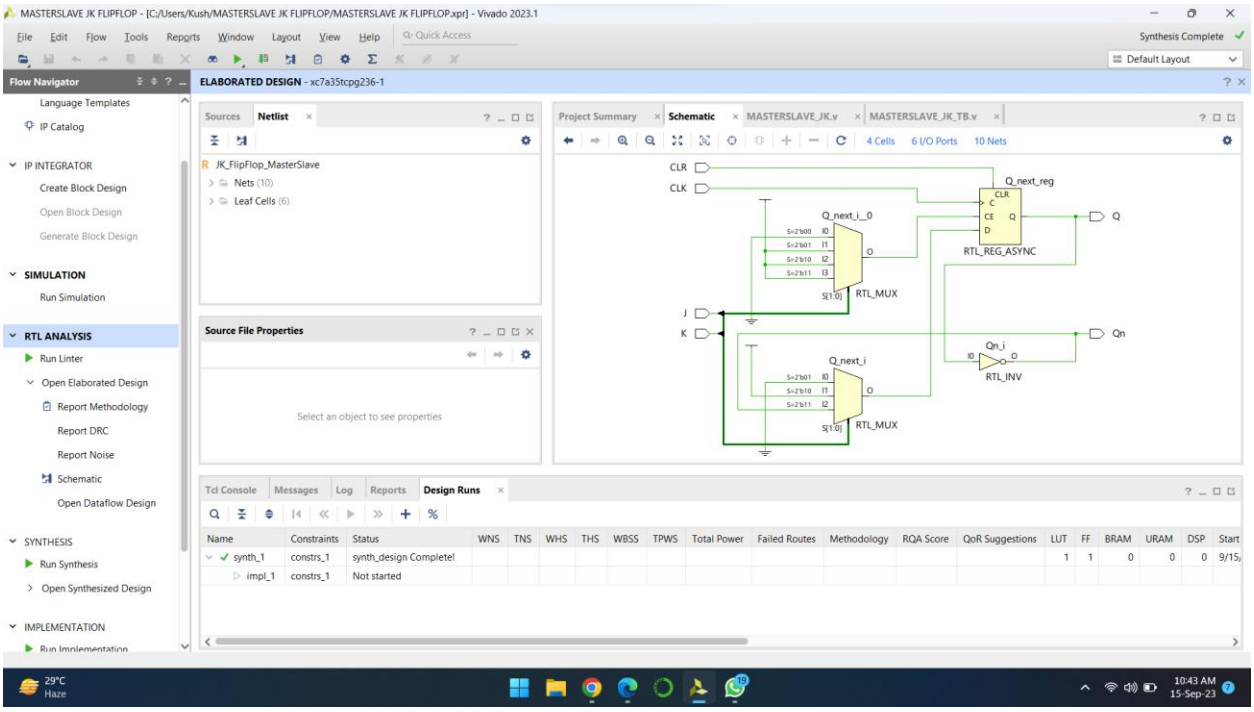

MASTERSLAVE JK FLIPFLOP - [C:/Users/Kush/MASTERSLAVE JK FLIPFLOP/MASTERSLAVE JK FLIPFLOP.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
MASTERSLAVE_JK.v
MASTERSLAVE_JK_TB.v
Untitled 1
C:/Users/Kush/MASTERSLAVE JK FLIPFLOP/MASTERSLAVE JK FLIPFLOP.srcs/sim_1/new/MASTERSLAVE_JK_TB.v
Tcl Console
Messages
Log
Sim Time: 40 ns
29°C
Haze
10:41 AM
15-Sep-23


MASTERSLAVE JK FLIPFLOP - [C:/Users/Kush/MASTERSLAVE JK FLIPFLOP/MASTERSLAVE JK FLIPFLOP.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench
MASTERSLAVE_JK.v
MASTERSLAVE_JK_TB.v
Untitled 1
C:/Users/Kush/MASTERSLAVE JK FLIPFLOP/MASTERSLAVE JK FLIPFLOP.srcs/sim_1/new/MASTERSLAVE_JK_TB.v
// Test case 2: Toggle
J = 1;
K = 1;
CLR = 0;
#10;
// Test case 3: Clear
J = 0;
K = 1;
CLR = 1;
#10;
// Test case 4: No Change
J = 0;
K = 0;
CLR = 0;
#10;
// Terminate simulation
$finish;
end
// Clock generation
always begin
#5 CLK = ~clk;
end
endmodule
Tcl Console
Messages
Log
Sim Time: 40 ns
29°C
Haze
10:42 AM
15-Sep-23

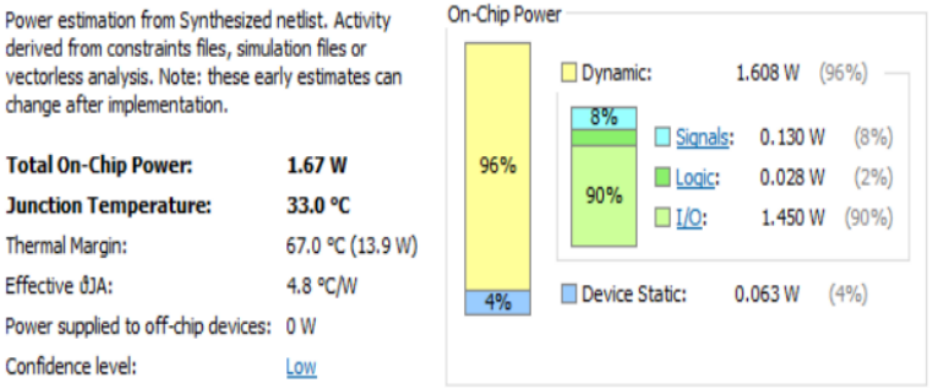
WAVEFORM:


MASTERSLAVE JK FLIPFLOP - [C:/Users/Kush/MASTERSLAVE JK FLIPFLOP/MASTERSLAVE JK FLIPFLOP.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench
MASTERSLAVE_JK.v
MASTERSLAVE_JK_TB.v
Untitled 1
Name
Value
J
K
CLK
CLR
Q
Qn
clk
0.000 ns
5.000 ns
10.000 ns
15.000 ns
20.000 ns
25.000 ns
30.000 ns
35.000 ns
40.000 ns
Tcl Console
Messages
Log
Sim Time: 40 ns
29°C
Haze
10:41 AM
15-Sep-23

RTL SCHEMATIC:




POWER REPORT:



21. POSITIVE EDGE DETECTOR:

CODE:

posedge_detct.v | posedge_tb.v | Untitled 1

C:/Users/DELL/positive edge detector/positive edge detector.srcs/sources_1/new/posedge_detct.v

```
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
```

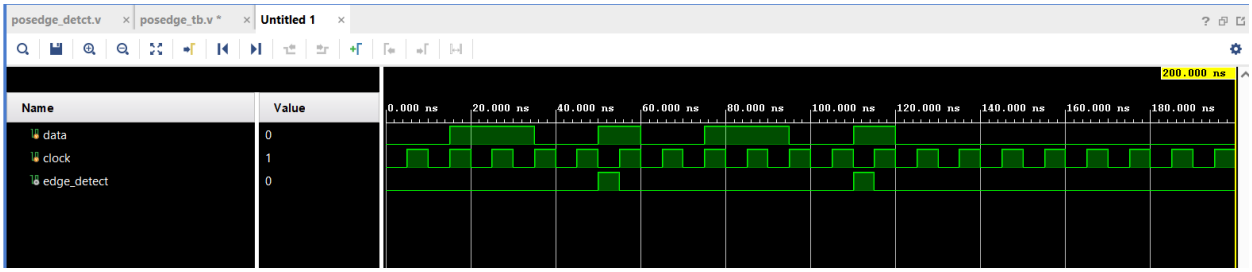
TESTBENCH:

posedge_detct.v | posedge_tb.v * | Untitled 1

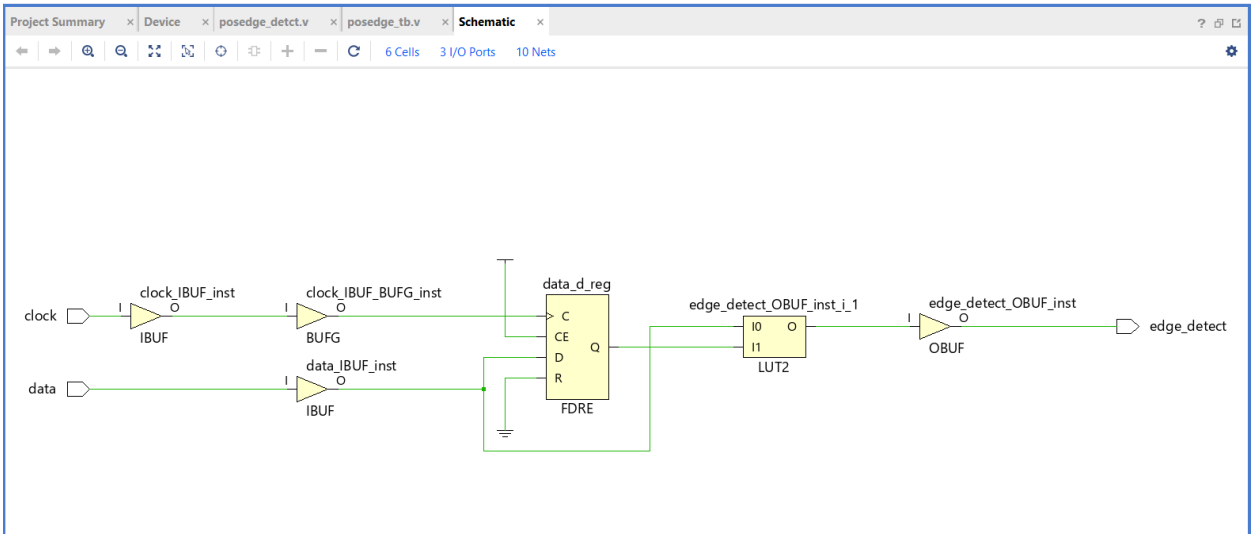
C:/Users/DELL/positive edge detector/positive edge detector.srcs/sim_1/new/posedge_tb.v

```
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
```

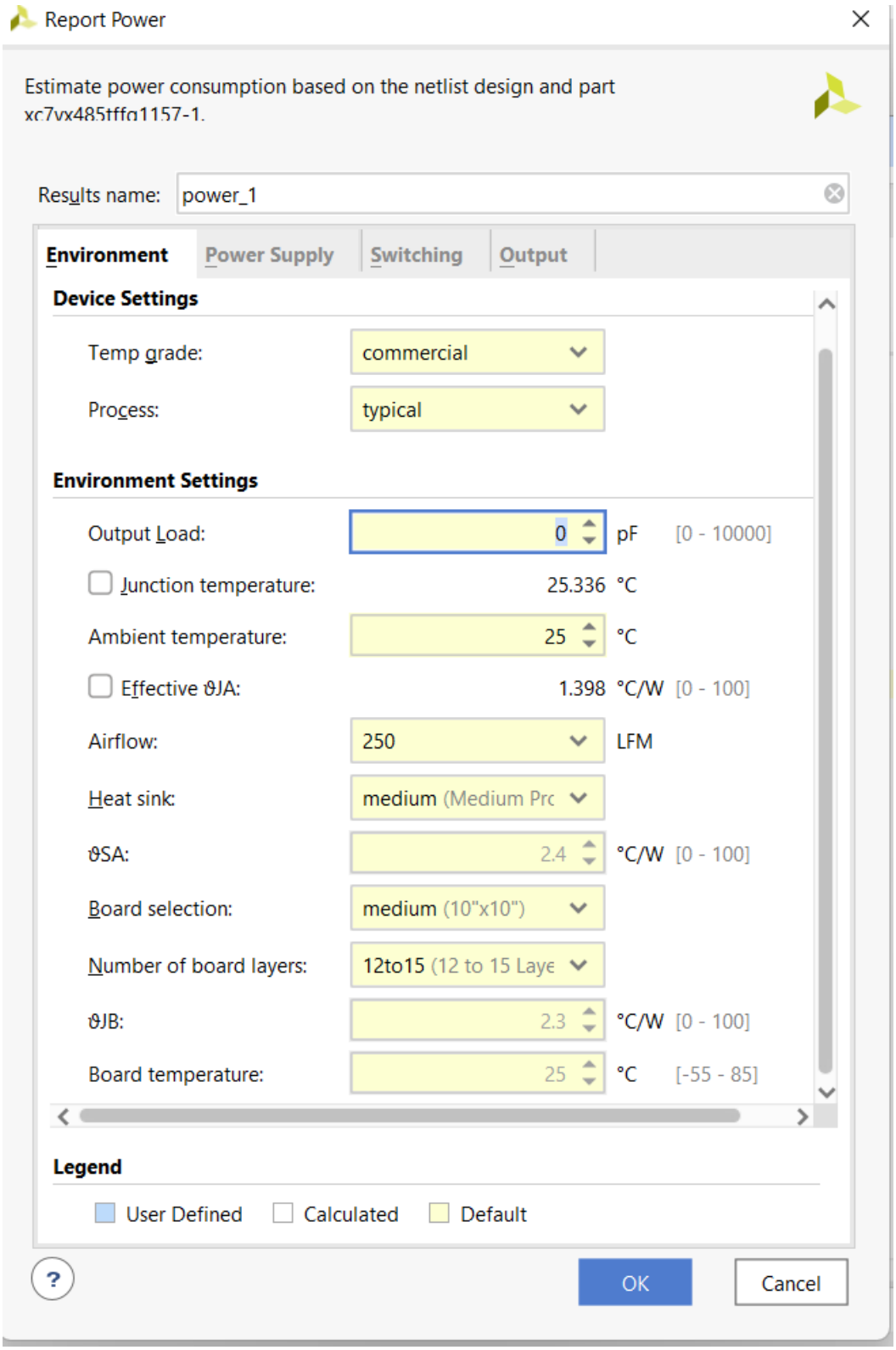
WAVEFORM:



RTL SCHEMATIC:




POWER REPORT:


Report Power
Estimate power consumption based on the netlist design and part xc7vx485tffg1157-1.
Results name: power_1
Environment
Power Supply
Switching
Output
Device Settings
Temp grade: commercial
Process: typical
Environment Settings
Output Load: 0 pF [0 - 10000]
Junction temperature: 25.336 °C
Ambient temperature: 25 °C
Effective ϑJA: 1.398 °C/W [0 - 100]
Airflow: 250 LFM
Heat sink: medium (Medium Prc
ϑSA: 2.4 °C/W [0 - 100]
Board selection: medium (10"x10")
Number of board layers: 12to15 (12 to 15 Laye
ϑJB: 2.3 °C/W [0 - 100]
Board temperature: 25 °C [-55 - 85]
Legend
User Defined
Calculated
Default
?
OK
Cancel

22. BCD ADDER:

CODE:

BCD_ADDER.v

C:/Users/Kush/BCD_ADDER/BCD_ADDER.srcs/sources_1/new/BCD_ADDER.v

```
module full_adder(
input a,
input b,
input c_in,
output sum,
output c_out
);
assign sum = a ^ b ^ c_in;
assign c_out = a&b | a&c_in | b&c_in;

endmodule
module _4_bit_adder(

input [3:0]a,
input [3:0]b,
input c_in,
output [3:0]sum,
output c_out

);
wire w1,w2,w3;

full_adder fa1(a[0],b[0],c_in,sum[0],w1);
full_adder fa2(a[1],b[1],w1,sum[1],w2);
full_adder fa3(a[2],b[2],w2,sum[2],w3);
full_adder fa4(a[3],b[3],w3,sum[3],c_out);

endmodule
module bcd_adder(
input [3:0]a,
input [3:0]b,
input c_in,
output c_out,
output [3:0]sum
);
wire w1,w2,p1,p2,c0;
wire [3:0]data,z;

and (p1,z[3],z[2]);
and (p2,z[3],z[1]);
or (c_out,p1,p2,w1);
```

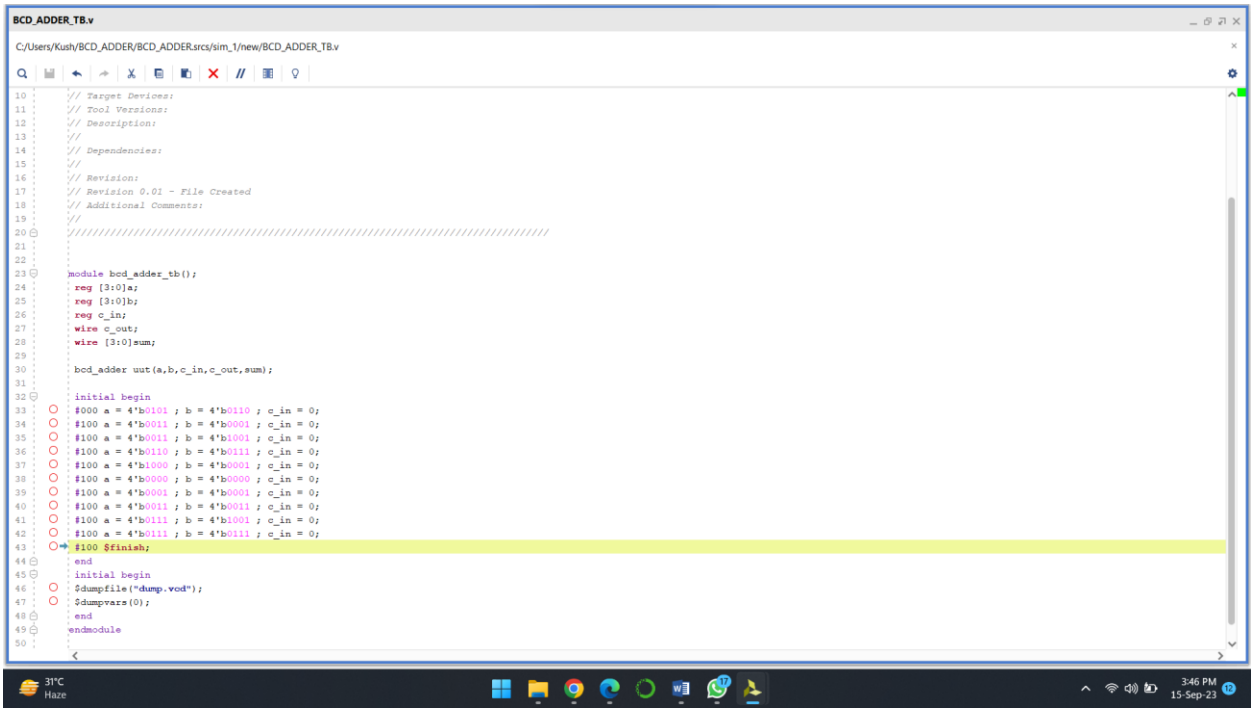


BCD_ADDER.v

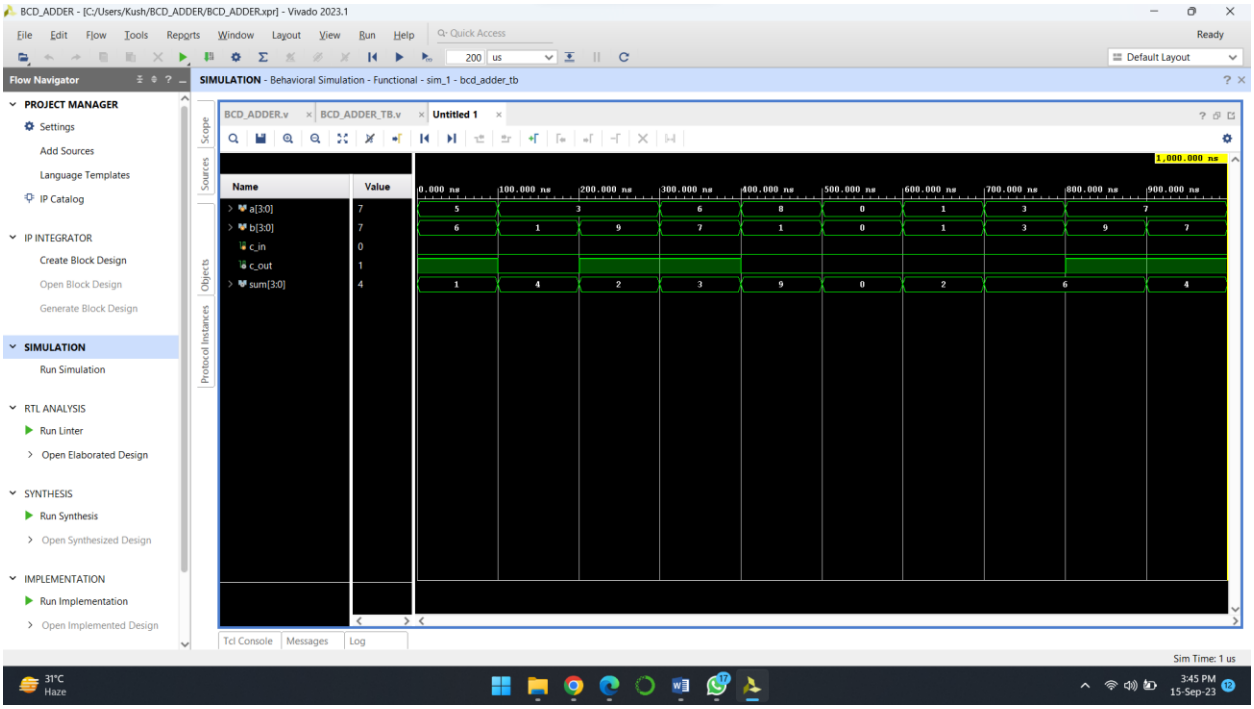
C:/Users/Kush/BCD_ADDER/BCD_ADDER.srcs/sources_1/new/BCD_ADDER.v

```
input [3:0]a,
input [3:0]b,
input c_in,
output [3:0]sum,
output c_out

);
wire w1,w2,w3;

full_adder fa1(a[0],b[0],c_in,sum[0],w1);
full_adder fa2(a[1],b[1],w1,sum[1],w2);
full_adder fa3(a[2],b[2],w2,sum[2],w3);
full_adder fa4(a[3],b[3],w3,sum[3],c_out);

endmodule
module bcd_adder(
input [3:0]a,
input [3:0]b,
input c_in,
output c_out,
output [3:0]sum
);
wire w1,w2,p1,p2,c0;
wire [3:0]data,z;

and (p1,z[3],z[2]);
and (p2,z[3],z[1]);
or (c_out,p1,p2,w1);
xor (c0,c_out,c_out);

//assign data ={ 1'b0,c_out,c_out,1'b0};
assign data[0]=c0;
assign data[1]=c_out;
assign data[2]=c_out;
assign data[3]=c0;

_4_bit_adder a1(a,b,c_in,z,w1);
_4_bit_adder a2(data,z,c_in,sum,w2);

endmodule
```

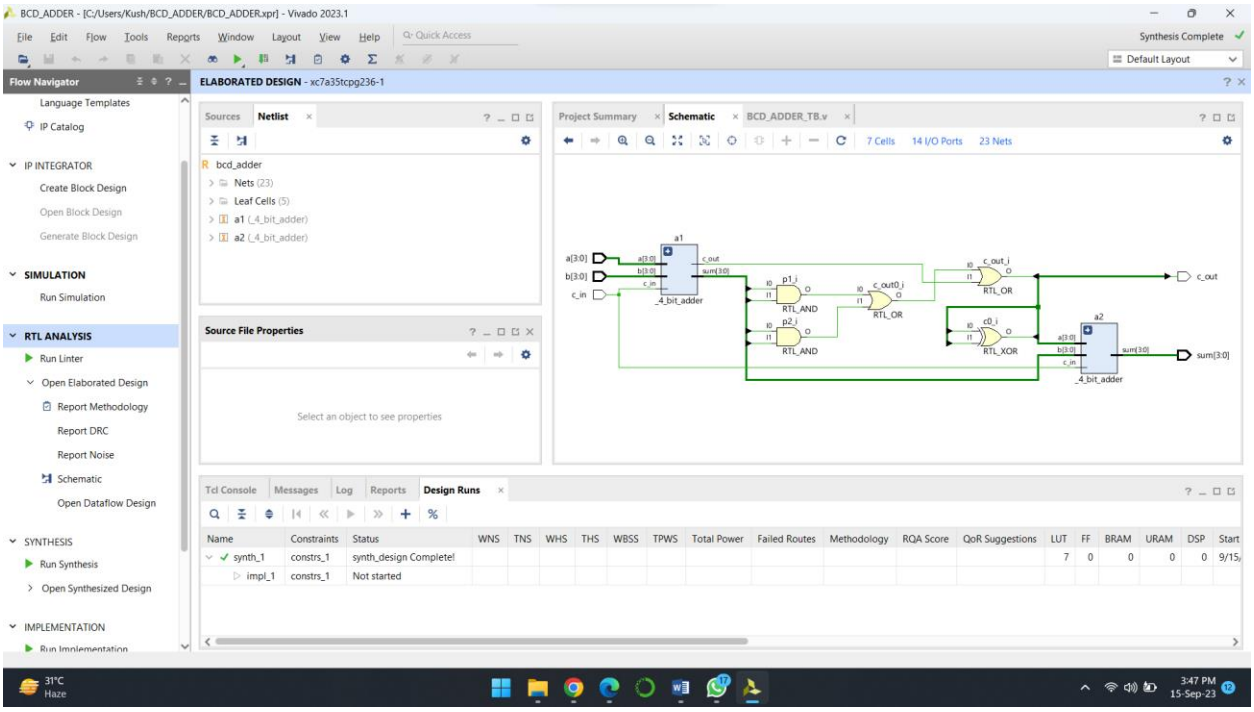
TESTBENCH:



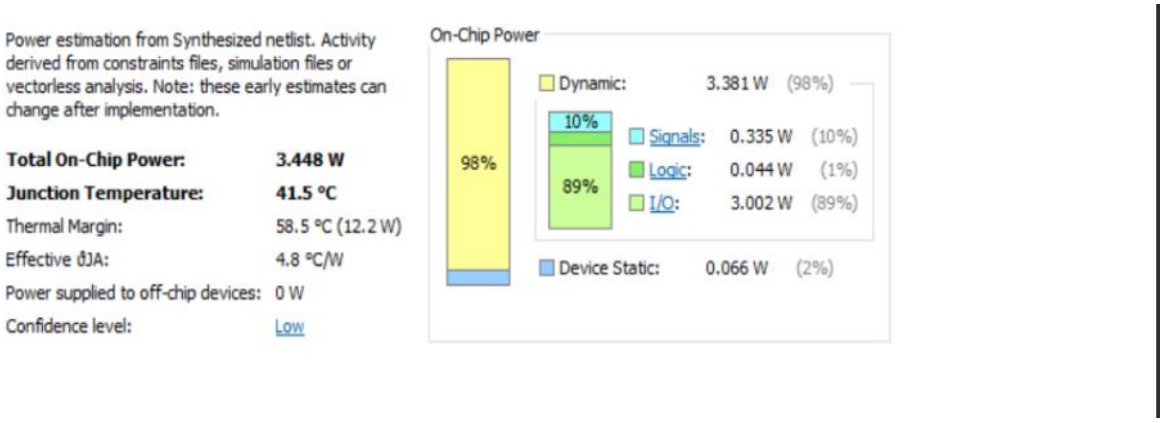

WAVEFORM:



RTL SCHEMATIC:




POWER REPORT:



23. 4-BIT CARRY SELECT ADDER:

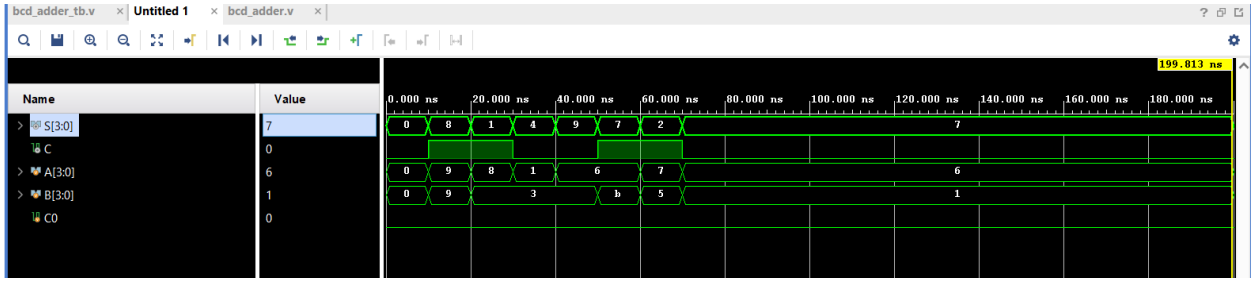
CODE:

```
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
```

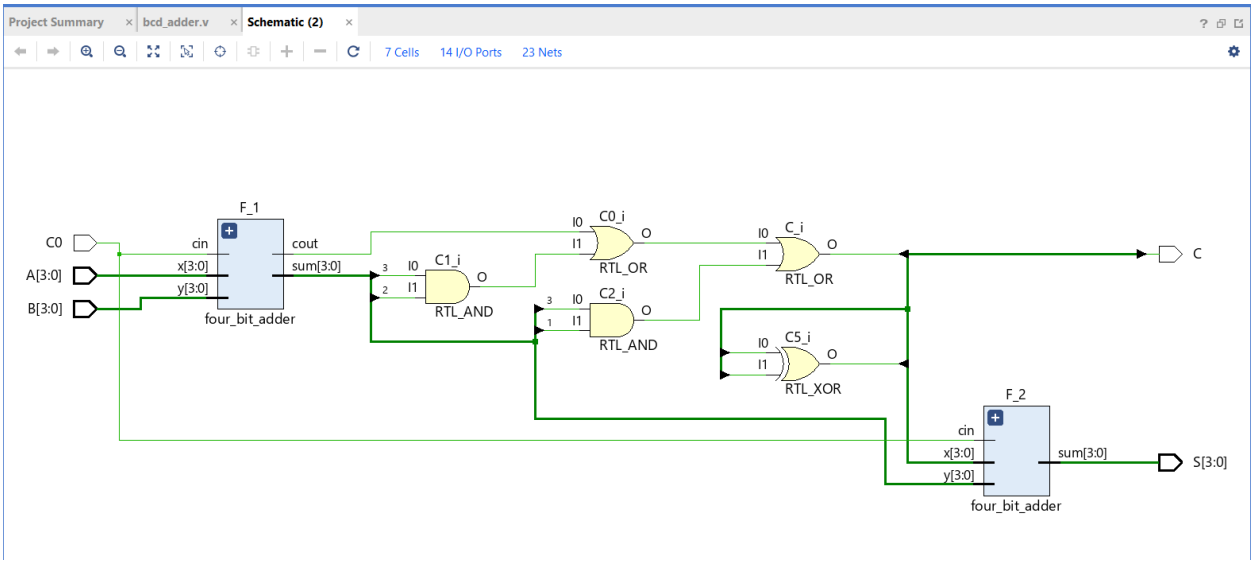
TESTBENCH:

```
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
```

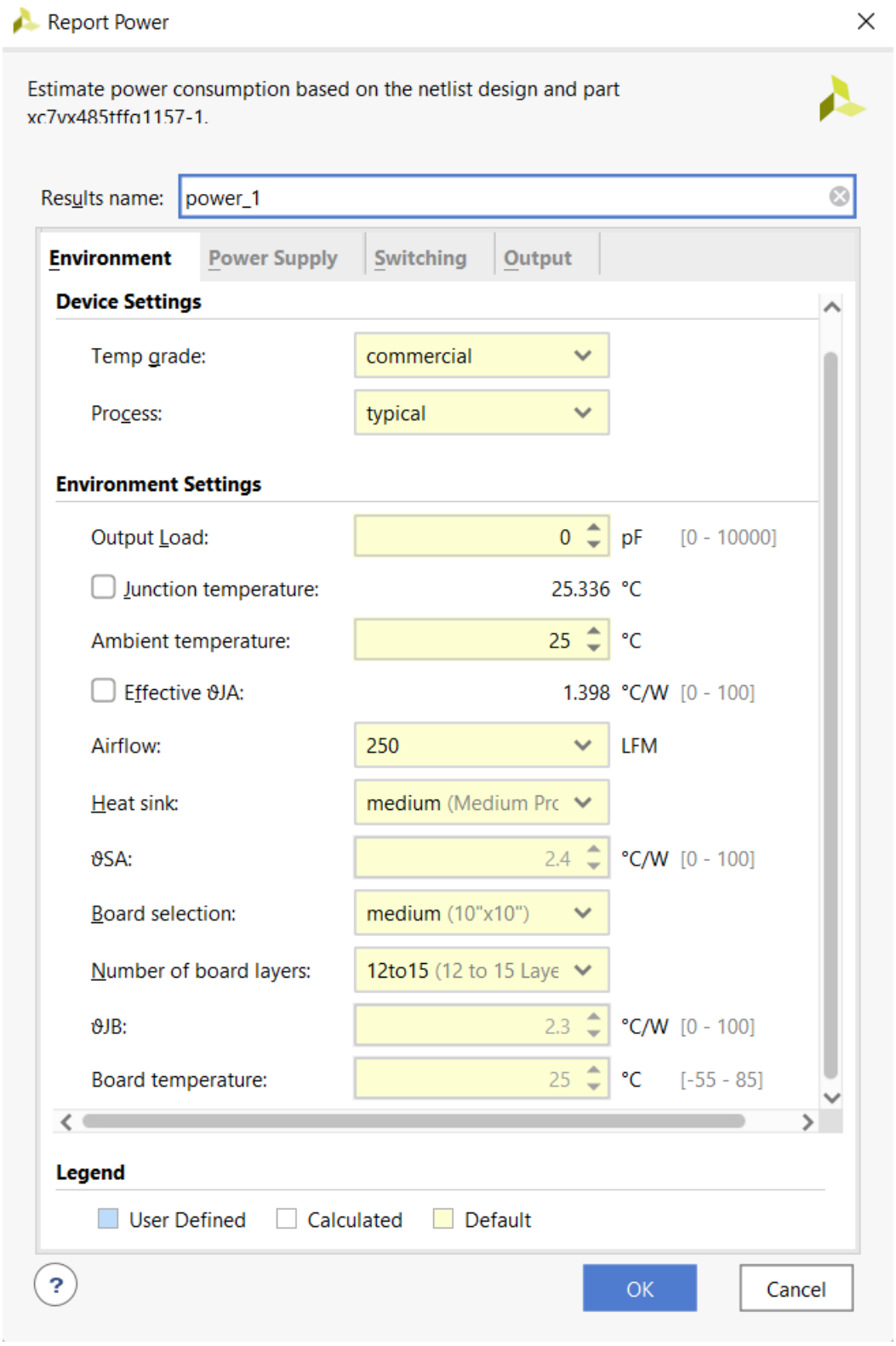
WAVEFORM:



RTL SCHEMATIC:



POWER REPORT:



24. MOORE FSM 1010 SEQUENCE DETECTOR:

CODE:

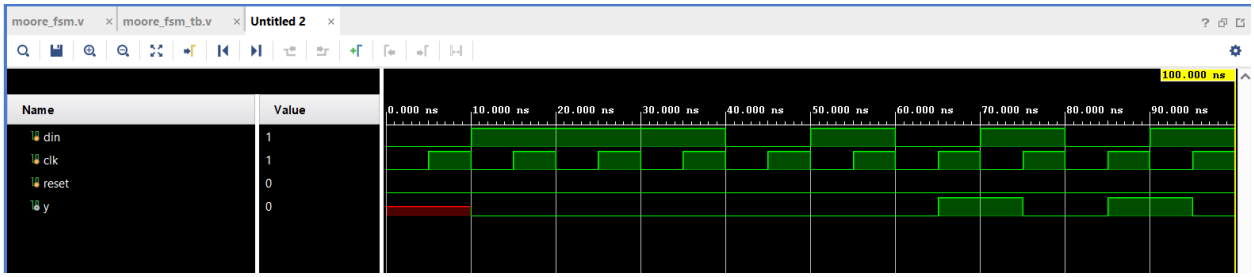
```
module moore_fsm(din, reset, clk, y);
input din;
input clk;
input reset;
output reg y;
reg [2:0] cst, nst;
localparam S0 = 3'b000,
           S1 = 3'b001,
           S2 = 3'b010,
           S3 = 3'b100,
             S4 = 3'b101;
always @(cst or din)
 begin
 case (cst)
   S0: if (din == 1'b1)
   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

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

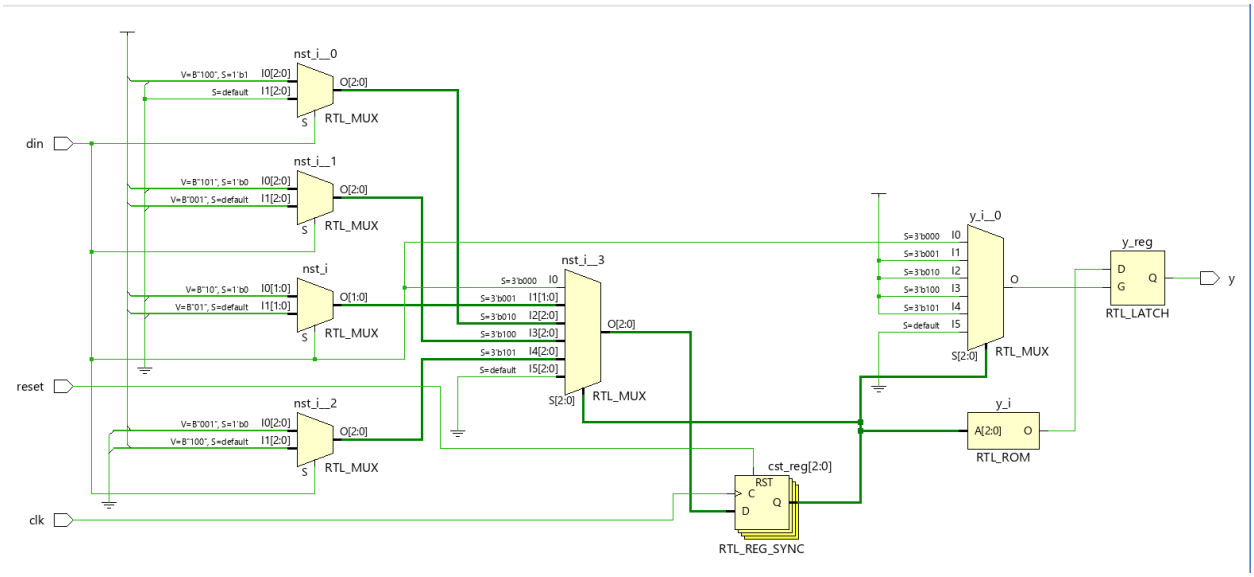
TESTBENCH:

```
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;

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

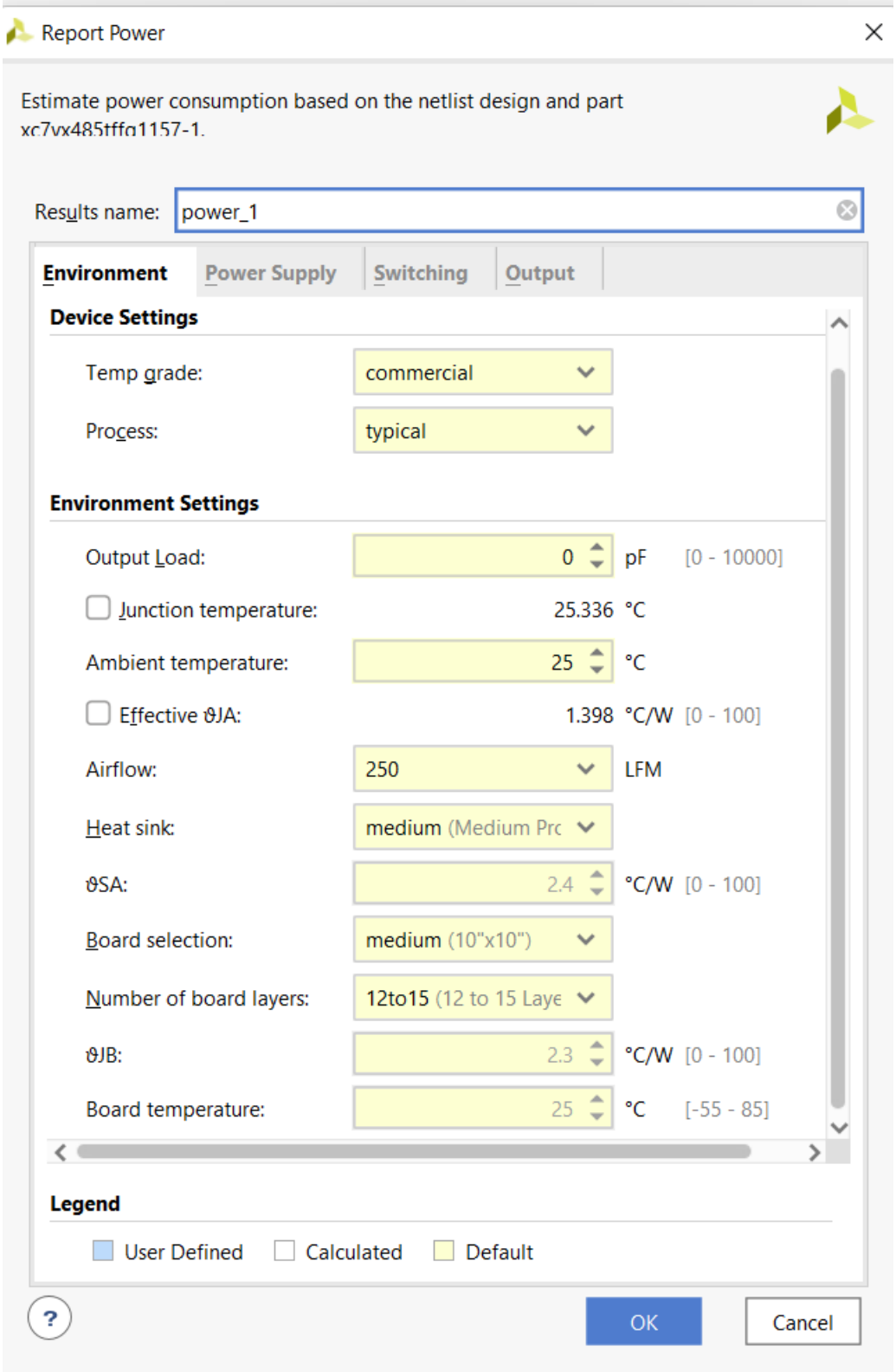
WAVEFORM:



RTL SCHEMATIC:



POWER REPORT:



25. N:1 MUX:

CODE:

n1_mux.v × n1_mux_tb.v × Untitled 3 ×

C:/Users/DELL/n1 mux/n1 mux.srcs/sources_1/new/n1_mux.v

```
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:

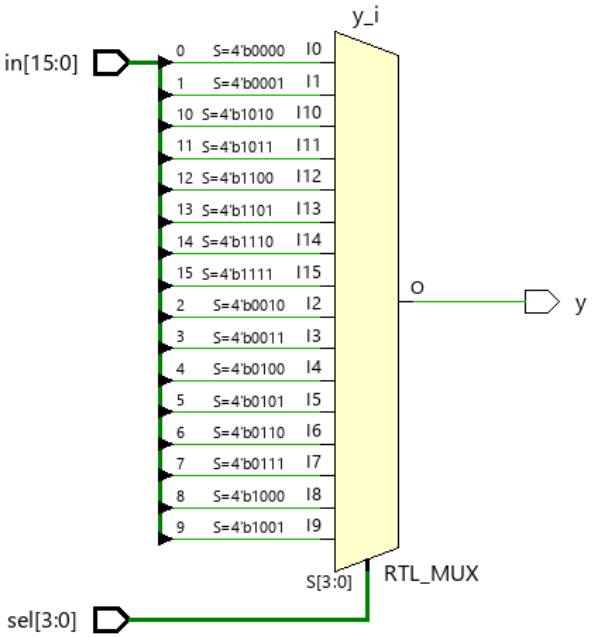
n1_mux.v | n1_mux_tb.v | Untitled 3

C:/Users/DELL/n1 mux/n1 mux.srcs/sim_1/new/n1_mux_tb.v

```
module n_1mux_tb;
    reg [15:0] in;
    reg [3:0] sel;
    wire y;
    n_1mux uut (
        .in(in),
        .sel(sel),
        .y(y)
    );
initial begin
in=16'b1000000000000000; sel=4'b0000;
#10 in=16'b0100000000000010; sel=4'b0001;
#10 in=16'b0010000000000100; sel=4'b0010;
#10 in=16'b0001000000000000; sel=4'b0011;
#10 in=16'b0000100000011000; sel=4'b0100;
#10 in=16'b0000010000000000; sel=4'b0101;
#10 in=16'b0000001000000000; sel=4'b0110;
#10 in=16'b0000000111100000; sel=4'b0111;
#10 in=16'b0000000010000000; sel=4'b1000;
#10 in=16'b0000000001000000; sel=4'b1001;
#10 in=16'b0000011000100000; sel=4'b1010;
#10 in=16'b0000000000010000; sel=4'b1011;
#10 in=16'b0000000000001000; sel=4'b1100;
#10 in=16'b0000000000000100; sel=4'b1101;
#10 in=16'b0000000000000010; sel=4'b1110;
#10 in=16'b1000000000000001; sel=4'b1111;
end

initial
begin
$monitor("in=%b | sel=%b | out=%b",in,sel,y);
```

WAVEFORM:


RTL SCHEMATIC:



POWER REPORT: