VDF Project Part-1

Group Number: 7

Team Members: Saksham Gupta (2019199)
Samaksh Gupta (2019200)
Shragvi Sidharth Jha (2019207)

Introduction

In this part of the project we will cover the various steps involved in the VLSI Design Flow before Physical Design.

Our design consists of

- **Frequency Divider by 6**
- **Four bit Comparator**
- **Bitwise XOR**
- **Addition block**

The flow is first creating the RTL based on our specifications. Then robustly testing the RTL. By generating corresponding test vectors we can test for code coverage. Later on we will synthesize the design, for different constraints and check the logical equivalence to ensure that synthesized netlist is equivalent to our RTL. Finally we will close off with performing static timing analysis and scan chain insertion to perform the design for testability step(DFT).

Part - 1

Design Specifications & Design Requirements
For this design we read A,B,C as inputs into a register. Based on the value C takes we perform operations on A&B and write the outputs to the out port again through a register. The clk & rst signals are also added as inputs. The rst is synchronous.

Specifications & Assumptions:

- Inputs: A[9:0], B[9:0], C[8:0]
- Output: out[10:0]
- Design is synchronous and operations are performed on the positive edge of the clock.
- Even the reset port is synchronous with the clock so on the corresponding next positive clock edge the input registers will be reset with the value 0.
- For our design based on the value C, different operations take place.
    - When C[6:0] + C[8:2] < 63 ⇒ Frequency Divider by 6
    - When 64 < C[6:0] + C[8:2] < 99 ⇒ 4-bit Comparator
    - When 99 < C < 128 ⇒ bitwise A XOR B
    - Else A+B[9:0]
- Based on these specifications we chose the size of A, B, C, and the output.

Part - 2

Simulation & Code Coverage Analysis

Verilog Codes & Test Benches

- rtl_topmodule.v
- add.v
- clk_divider.v
- xor.v
- comparator.v
- testbench1.v
- testbench2.v
- testbench3.v

An additional top.v is also used for some of the design steps, this contains all of the previously mentioned design file excluding the testbenches in a single file.

rtl_topmodule.v

```
module rtl_module (
    input clk,
    input rst,

    input wire [9:0] A, // Input A
    input wire [9:0] B, // Input B
    input wire [8:0] C, // Input C

    output reg [10:0] out // Output
);
    reg [9:0] A_reg, B_reg;
    reg [8:0] C_reg;
    reg [10:0] out_reg;

    // always block to sample input data
    always @(posedge clk) begin
        if (rst) begin // if reset is 1
            A_reg <= 10'd0;
            B_reg <= 10'd0;
            C_reg <= 9'd0;
        end else begin // else we take the input
            A_reg <= A;
            B_reg <= B;
            C_reg <= C;
        end
    end
```

```
    // Enable CLK divider
    wire out1;

    // Instantiate CLK divider
    clk_divider #(.div_value(3)) in1(
        .clk_in(clk),
        .clk_out(out1)
    );

    wire [2:0] out2;
    // Instantiate the Comparator
    comparator in2(
        .A(A_reg[3:0]),
        .B(B_reg[3:0]),
        .A_GT_B(out2[0]), // in the case of comparator we assign the
indivivudal outputs to the output wire
        .A_LT_B(out2[1]),
        .A_EQ_B(out2[2])
    );

    wire [9:0] out3;
    // Instantiate the bitwise XOR module
    bitwise_xor in3(
        .A(A_reg),
        .B(B_reg),
        .A_xor_B(out3)
    );

    wire [10:0] out4;
    // Instantiate the addition block
    add in4(
        .A(A_reg),
        .B(B_reg),
        .A_add_B(out4)
    );

    // select output based on the C value
    always @(*) begin
        if (C_reg[6:0] + C_reg[8:2] < 63) begin
            out_reg = {10'd0, out1};
        end else if (C_reg[6:0] + C_reg[8:2] > 7'd64 && C_reg[6:0] +
C_reg[8:2] < 7'd99) begin
```

```
            out_reg = {8'd0, out2};
        end else if (C_reg > 98 && C_reg < 128) begin
            out_reg = {1'd0, out3};
        end else begin // our design performs the addition operation if
all other cases are not met
            out_reg = out4;
        end
    end

    // output to FF sampling
    always @(posedge clk) begin
        if(rst) begin
            out <= 11'd0; // reset the output if reset == HIGH
        end else begin
            out <= out_reg; // else we pass the output to the port
        end
    end
endmodule
```

add.v

```
module add(
    input [9:0] A,
    input [9:0] B,

    output reg [10:0] A_add_B
);
    always @(*) begin // combinational block that calculates A+B
        A_add_B = A + B;
    end
endmodule
```

clk_divider.v

```
module clk_divider
#(parameter div_value = 3)(
    input clk_in,
    output reg clk_out = 0
);
    // reg [1:0] count_reg = 0, count_next = 0;
    reg [1:0] count_reg = 0;

    // at every clock edge we are essentially counting till the div
factor. Which in this case is 3
```

```
    always @(posedge clk_in) begin
        if (count_reg == div_value-1) begin
            count_reg <= 0;
        end
        else begin
            count_reg <= count_reg + 1;
        end
    end

    // if the count == div_value then we invert the output clk
    always @(posedge clk_in) begin
        if (count_reg == div_value-1) begin
            clk_out <= ~clk_out;
        end
        else begin
            clk_out <= clk_out;
        end
    end
endmodule
```

comparator.v

```
module comparator(
    input [3:0] A,
    input [3:0] B,
    output reg A_GT_B,
    output reg A_LT_B,
    output reg A_EQ_B
);
    always @(*) begin // in the combinational block

        A_EQ_B = 0; // by default the outputs are 0
        A_GT_B = 0;
        A_LT_B = 0;

        if(A > B) begin // for the particular case of A>B we write 1 to
the corresponding port
            A_GT_B = 1;
        end
        if (A < B) begin
            A_LT_B = 1;
        end
        if (A == B) begin
```

```
            A_EQ_B = 1;
        end
    end
endmodule
```

xor.v

```
module bitwise_xor(
    input [9:0] A,
    input [9:0] B,

    output reg [9:0] A_xor_B
);
    always @(*) begin // in the combinational block
        A_xor_B = A ^ B; // we are performing the A xor B operation
    end
endmodule
```

Test Benches, Simulation Results & Coverage Reports

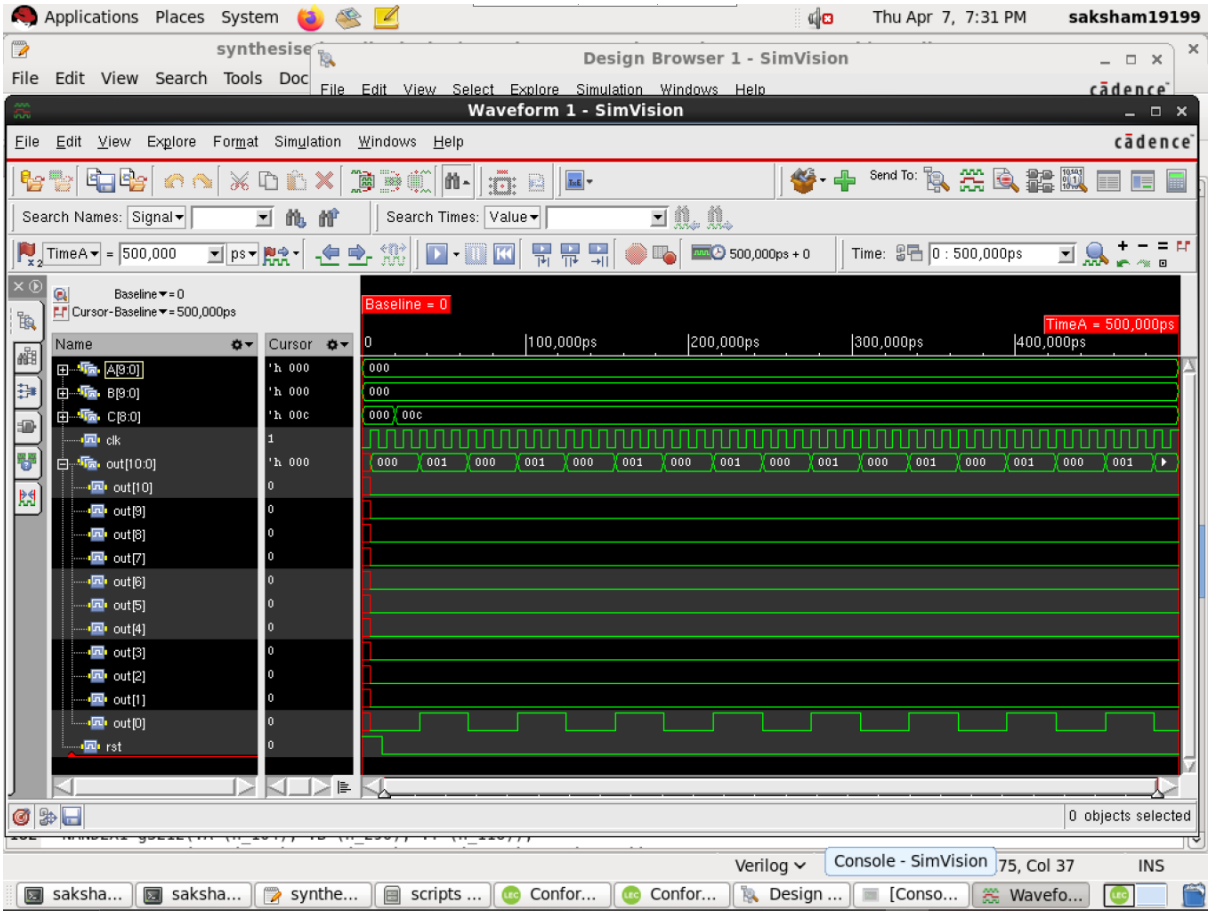
testbench1.v

This only tests clk_divider, the test vector is passed in such a way that only those blocks are checked.

```
`timescale 1ns / 1ps

module test_module();
    reg clk;
    reg rst;

    reg [9:0] A, B;
    reg [8:0] C;

    wire [10:0] out;

    // Initialize DUT
    rtl_module dut(
        .clk(clk),
        .rst(rst),
        .A(A),
        .B(B),
        .C(C),
        .out(out)
    );

    // Test vector block
    initial begin
        $display($time, "======= TestBench-1 =======");
        clk = 1'b0;

        A = 0; B = 0; C = 0;

        // the test vector is supplied is supplied in such a way that
only the clock divider operation occurs
        rst = 1;
        #12 rst = 0;
        #8 C = 12; // Case for clock divider
    end

    always #5 clk = ~clk;

    // Creating VCD file
```

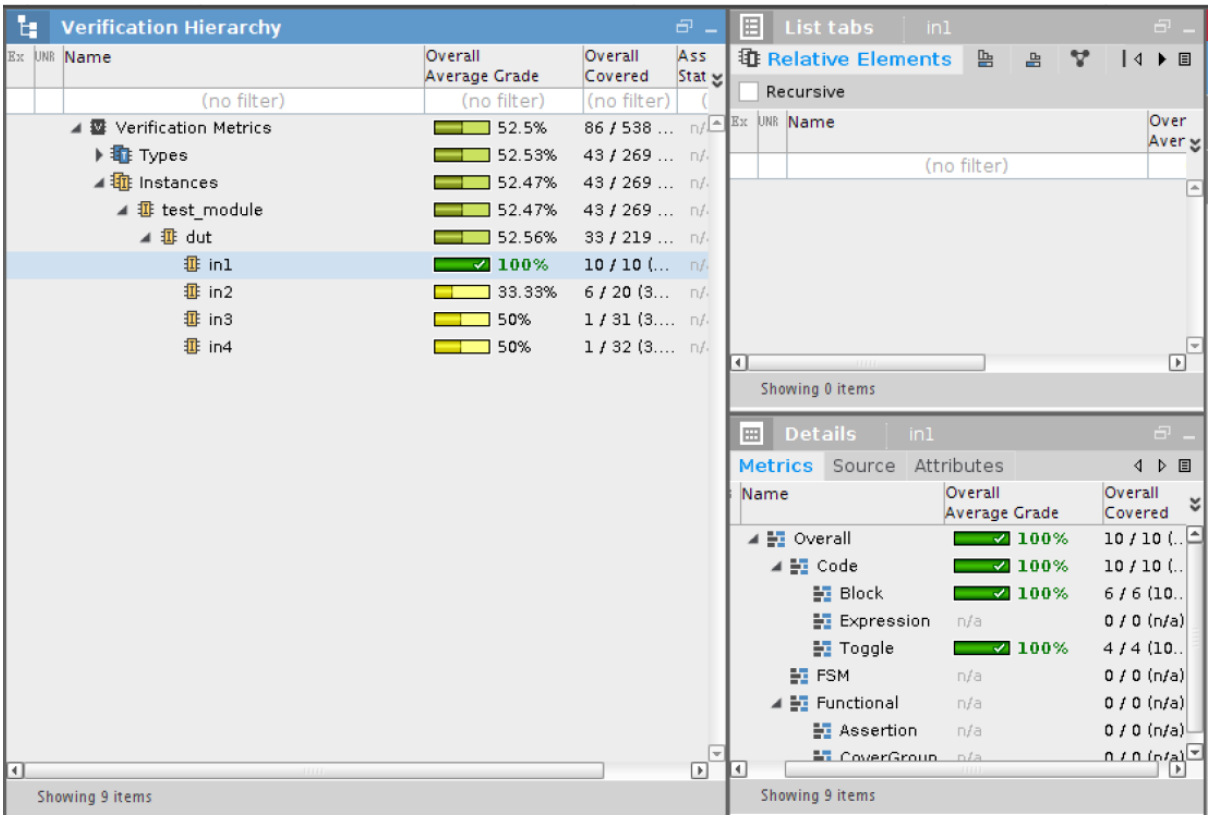
```
    initial begin
        $dumpfile("testbench1.vcd");
        $dumpvars;
    end

    // monitor inputs & output
    initial begin
        $monitor("%d,\t%d,\t%d,\t%d,\t%d,\t%d,\t%d", $time, clk, rst,
A, B, C, out);
    end

    initial begin
        #500 $finish;
    end
endmodule
```

Waveform



Here we can see that our output clock which is presented at output out[0]. For 6 input cycles we have one output cycle. The variable C has a value which meets the requirements of clk_divider.

Code Coverage



Report showing Coverage only for in1(clk_divider)

From this report we can see that only for in1 which is the clk_divider we get 100% code coverage which consists of 100% block coverage and 100% toggle coverage. As the testbench is written with testing only the clk_divider in mind this result is expected. While the entire dut instance has a code coverage of 52.56%, the reason for this is that other blocks such as the comparator and bitwise xor is not tested.

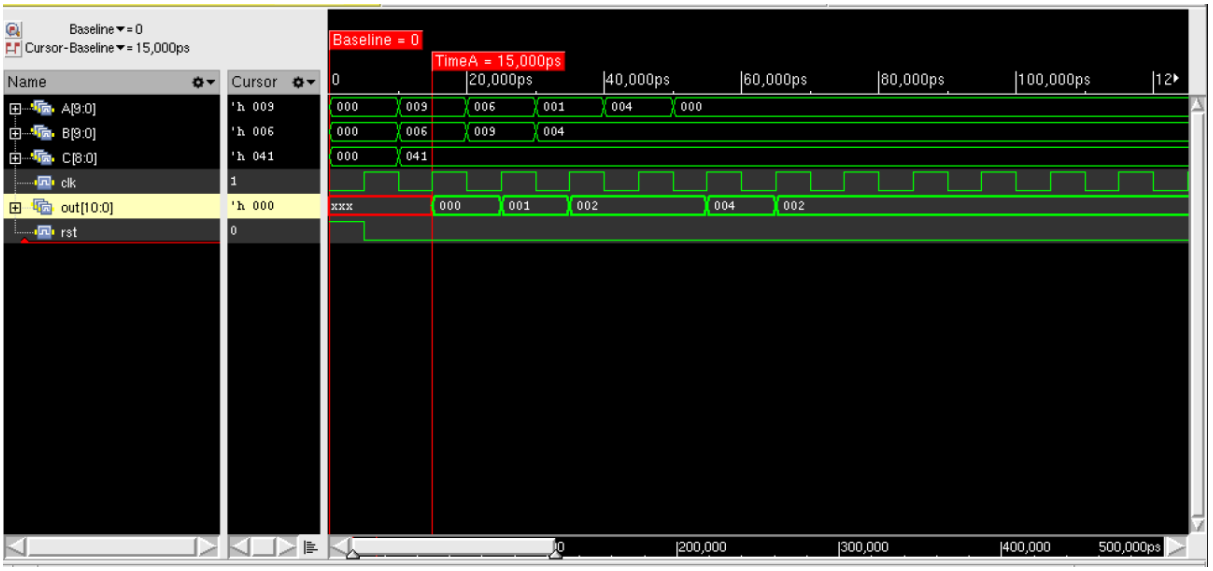
testbench2.v

This test bench only tests the comparator. The test vector is passed in such a way that only the comparator is tested.

```
module test_module();
    reg clk;
    reg rst;

    reg [9:0] A, B;
    reg [8:0] C;

    wire [10:0] out;

    // Initialize DUT
    rtl_module dut(
        .clk(clk),
        .rst(rst),
        .A(A),
        .B(B),
        .C(C),
        .out(out)
    );

    // Test vector block
    initial begin
        $display($time, "======= TestBench-2 =======");
        clk = 1'b0;

        A = 0; B = 0; C = 0;

        rst = 1;
        #5 rst = 0;
        #5 C = 65; // Case for Comparator
        A = 4'b1001; B = 4'b0110; // GT
        #10 A = 4'b0110; B = 4'b1001; // LT
        #10 A = 4'b0001; B = 4'b0100;
    #10 A = 4'b0100; // toggle EQ signal
    #10 A = 4'b0000;
    end

    always #5 clk = ~clk;

    // Creating VCD file
    initial begin
```

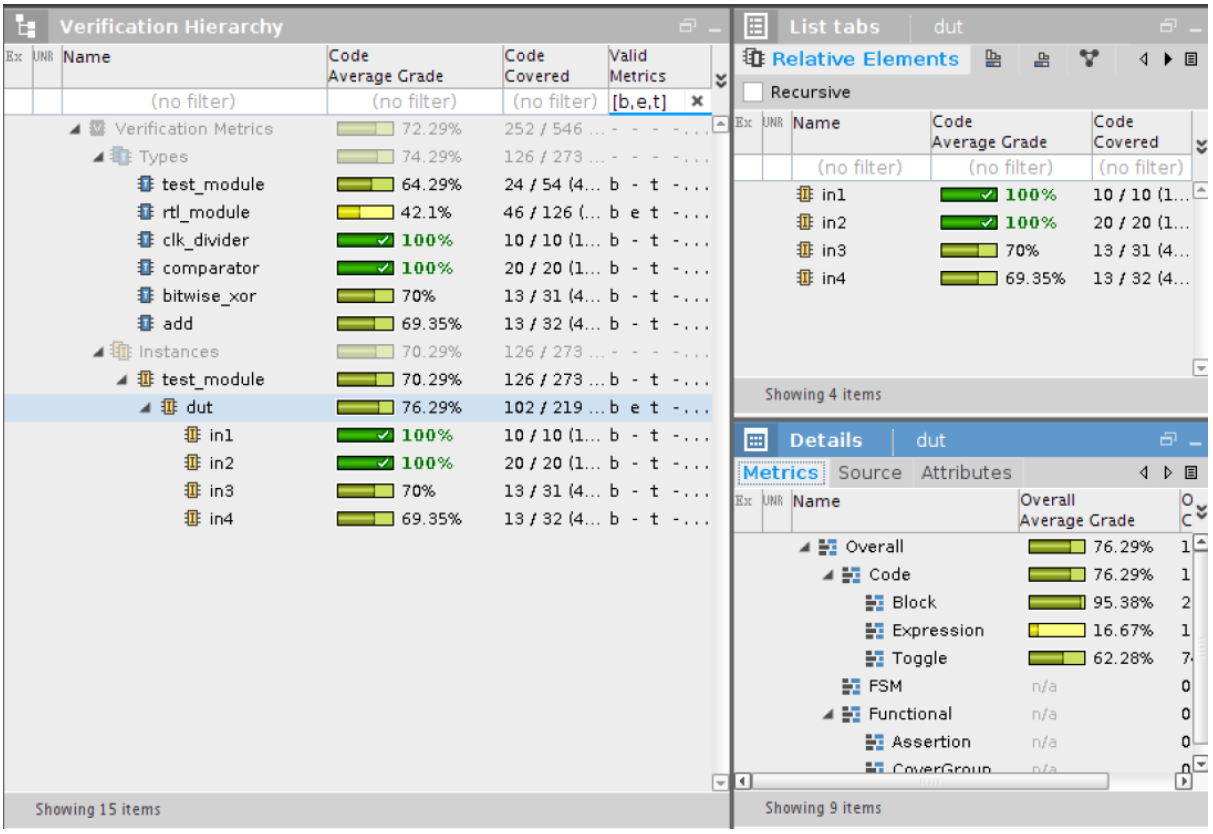
```
        $dumpfile("testbench2.vcd");
        $dumpvars;
    end

    // monitor inputs & output
    initial begin
        $monitor("%d,\t%d,\t%d,\t%d,\t%d,\t%d,\t%d", $time, clk, rst,
A, B, C, out);
    end

    initial begin
        #500 $finish;
    end
endmodule
```

waveform



From the waveform we can see that our outputs for the different test vectors we supply are correct. As we had combined the individual outputs by bitwise assignment for Greater than case we 1 as the output as the out[0] as set high. For less than out[1] was HIGH so the output is 2 while for the equal to case out[2] = HIGH so the output is 4. We also added some additional test vectors which helped in ensuring 100% toggle coverage.

Code Coverage



Here we can see that there is 100% code coverage for in1 which is the clk_divider and for in2 which is the comparator. Reason for 100% coverage for in1 is when we had reset the circuit and C == 0 this was the case that matches for the clk_divider hence the code coverage is 100%.

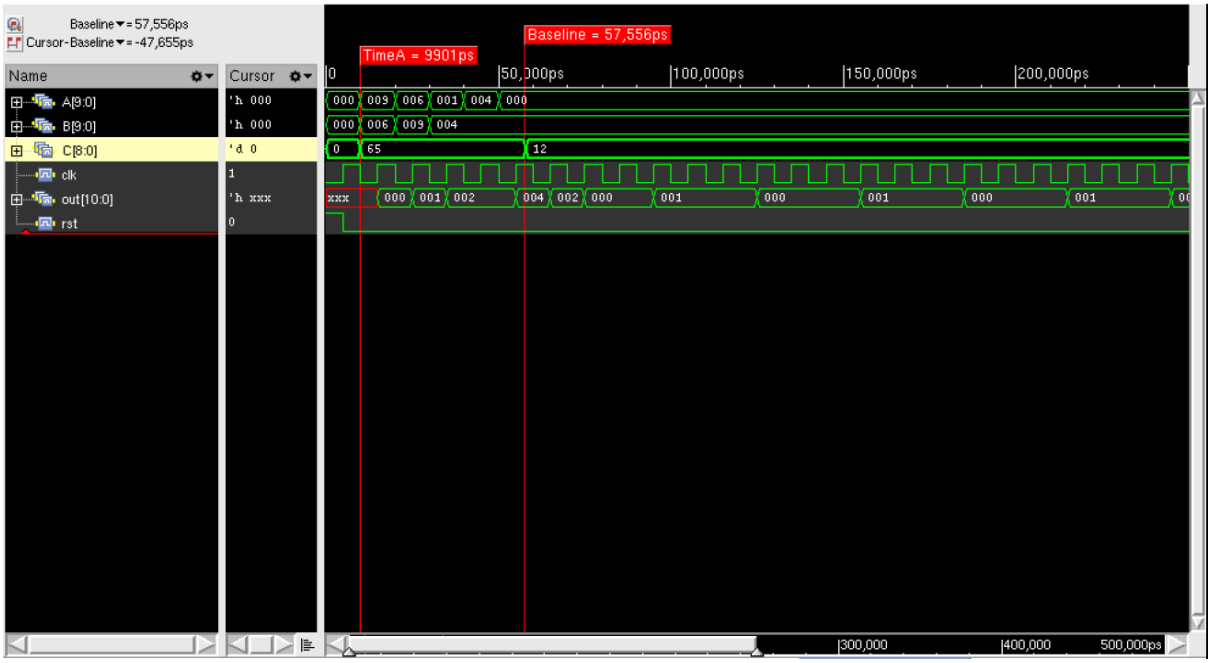
While for in2 we have also achieved 100% toggle coverage along with block coverage by providing the correct test vectors. But the overall code coverage for **dut** is 76.29% this is due to the fact that in3 & in4 is not tested.

testbench3.v

```
`timescale 1ns / 1ps

module test_module();
    reg clk;
    reg rst;

    reg [9:0] A, B;
    reg [8:0] C;

    wire [10:0] out;

    // Initialize DUT
    rtl_module dut(
        .clk(clk),
        .rst(rst),
        .A(A),
        .B(B),
        .C(C),
        .out(out)
    );

    // Test vector block
    initial begin
        $display($time, "======= TestBench-3 =======");
        clk = 1'b0;

        A = 0; B = 0; C = 0;

        // Comparator
        rst = 1;
        #5 rst = 0;
    #5 C = 65; // Case for Comparator
        A = 4'b1001; B = 4'b0110; // GT
        #10 A = 4'b0110; B = 4'b1001; // LT
        #10 A = 4'b0001; B = 4'b0100;
    #10 A = 4'b0100; // toggle EQ signal
    #10 A = 4'b0000;

        // Case for clock divider
        #8 C = 12;

    // XOR
```

```
        #250 C = 100;
        A = 32; B = 80;

    // ADD
        #10 C = 511;
        A = 60; B = 81;
    end

    always #5 clk = ~clk;

    // Creating VCD file
    initial begin
        $dumpfile("testbench3.vcd");
        $dumpvars;
    end

    // monitor inputs & output
    initial begin
        $monitor("%d,\t%d,\t%d,\t%d,\t%d,\t%d,\t%d", $time, clk, rst,
A, B, C, out);
    end

    initial begin
        #500 $finish;
    end
endmodule
```

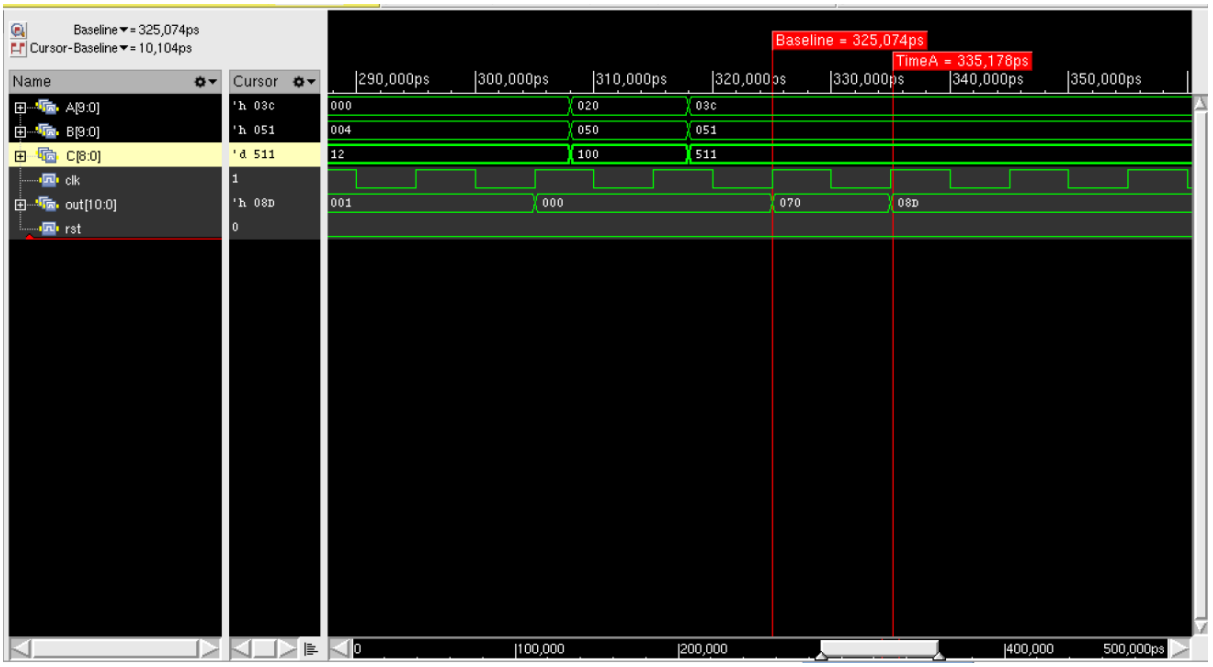
Waveform



Waveform only showing the clk_divider & comparator

For the second case we can see that our output clock which is presented at output out[0]. For 6 input cycles we have one output cycle. The variable C == 12, which meets the requirements of clk_divider.

For the first case we can see that our outputs for the different test vectors we supply are correct. As we had combined the individual outputs by bitwise assignment for Greater than case we 1 as the output as the out[0] as set high. For less than out[1] was HIGH so the output is 2 while for the equal to case out[2] = HIGH so the output is 4. We also added some additional test vectors which helped in ensuring 100% toggle coverage.



Waveform showing the result for bitwise XOR & Add

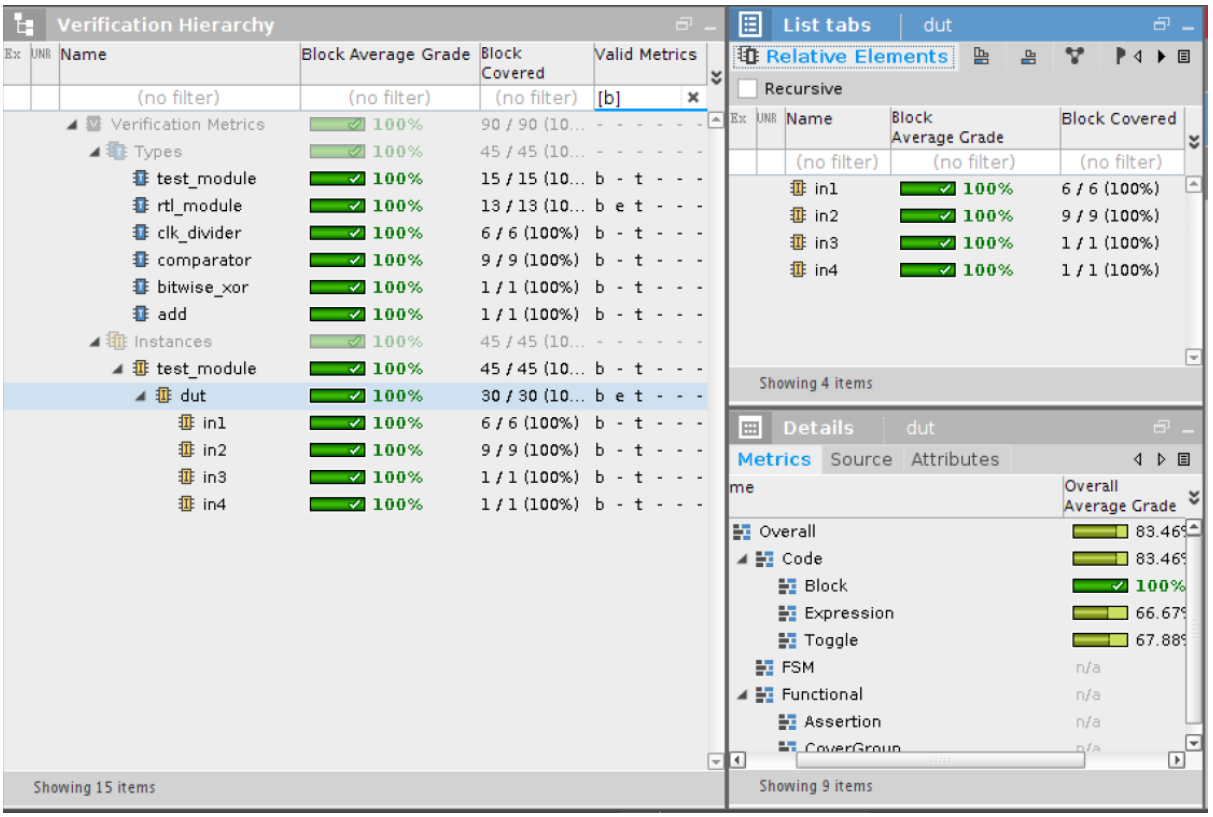
For the third case of C == 100, the output is A XOR B. We have also verified that the output of 70 for input of 20 & 50 is correct.

For the last case of C == 511, the output for the addition case is also correct.

0x3c + 0x51 = 0x8D.

Code Coverage

Block Coverage:

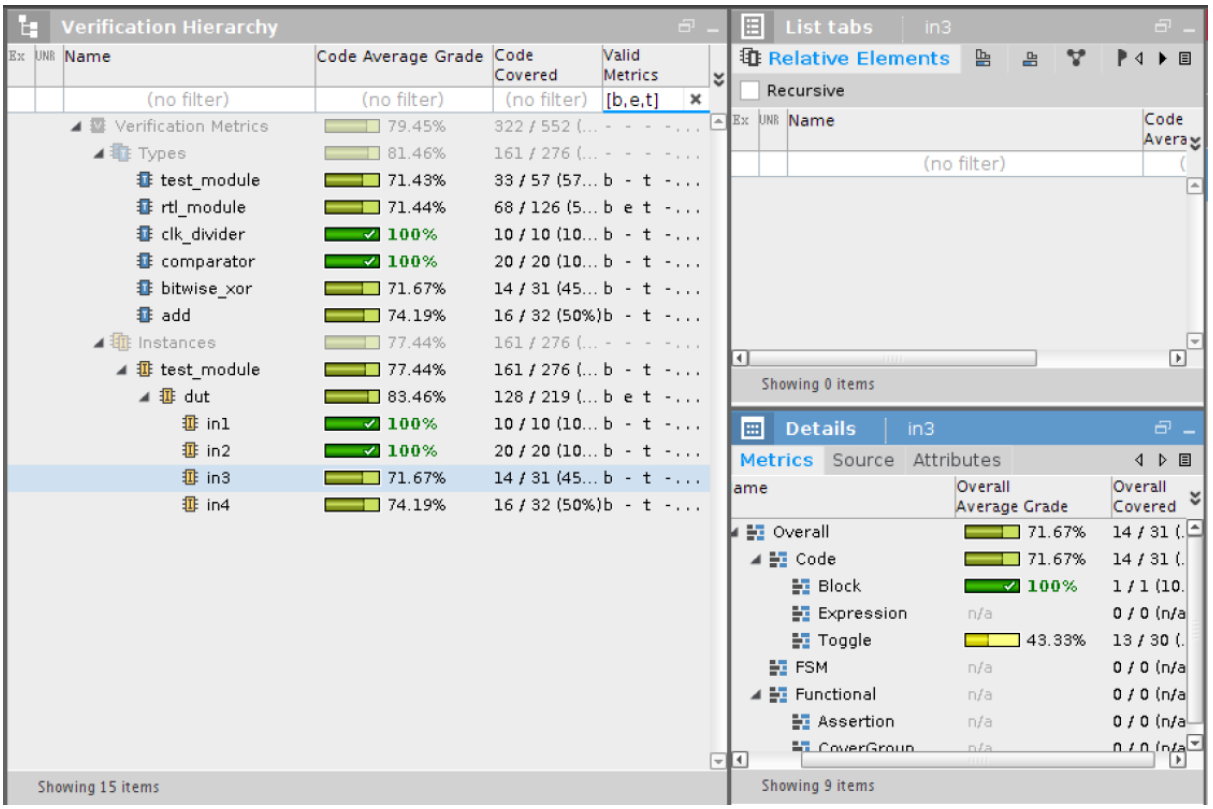


100% Block coverage is present for all the blocks.

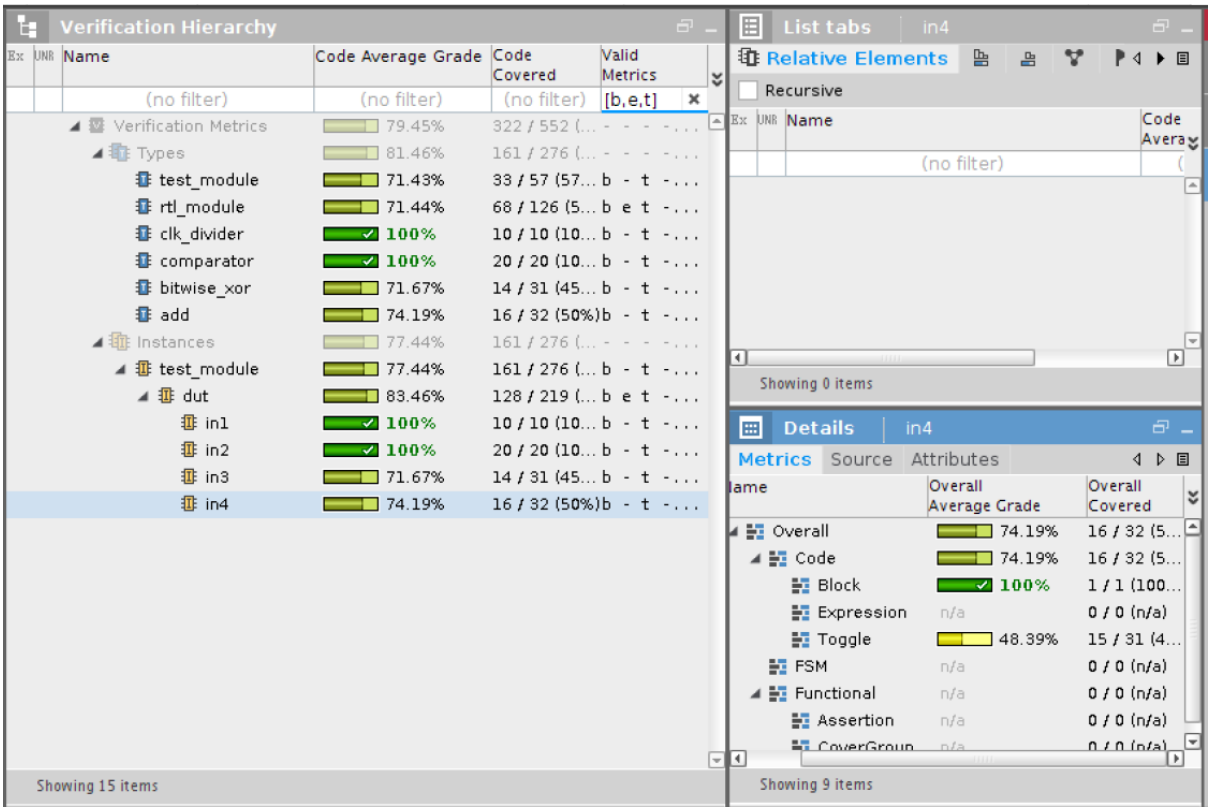
100% block coverage for the **dut** which is the instantiation of the rtl_module, our top module. More detailed analysis for the individual block such as in3 and in4 are given below.

Code Coverage (Block Coverage + Toggle Coverage):

For in3:



For in4:



Block coverage for the xor block we implemented is 100% but the toggle coverage is not. The way toggle coverage is checked by measuring where certain bits transitioned from 0 → 1 and 1 → 0. Meeting this requirement was straightforward for clk_divider and comparator. While for xor & add we would require more test vectors.
Due to the lower toggle coverage score we get a lower code coverage.
A similar case for the addition block. Our block coverage is 100% but the toggle is lower for the same reasons we mentioned above.

Final Results

TestBench No.	Block Coverage	Toggle Coverage	Code Coverage
Testbench1	90.26%	20.75%	52.56%
Testbench2	95.38%	62.28%	76.29%
Testbench3	100%	67.88%	83.46%

Final Analysis:
The simulation is done with the help of three different testbenches. Two testbenches which the dut partially. While the final test bench which tests the entire design. For the final test bench we can notice that the coverage for block coverage is also 100%, The toggle coverage is lower, that is due to the fact that for testing in3 and in4 more test vectors would be required to measure all the transitions for all the bits of the inputs and outputs. Although the final code coverage for testbench3 is not 100% the block coverage is 100%, which ensures that all the various lines we implemented are running correctly.

PART 3
Synthesis

Aim: Perform the logical synthesis of the RTL design using the constraint file. Write the three different constraints and generate the three different netlists. Three different constraints are written in such a way that the design can be synthesized for the minimum area, best timing with negative slack and intermediate constraints. Study the impact of the changing constraints on the quality of results.

Tool: Cadence Genus Synthesis Solution

Theory

After writing the Verilog code, the RTL needs to be converted into the netlist. The tool used for doing the logical synthesis is Genus of Cadence. **Synthesis** is the step where a constraint file is used to generate the netlist. It includes the realization of the gate-level netlist, which has all the functionalities specified in the RTL Verilog code. The type of synthesis done is the logical synthesis used to realize the various types of logical circuits. The netlist can also be generated to meet specific requirements such as the frequency, power, area etc.

About the Tool

The tool takes the information on the type of functionality that the synthesis needs to be performed on. The information on the type of functionality is provided through the RTL Verilog Code. The tool then specifies the constraints to generate the synthesized netlist to analyze it in terms of power, performance and area. This information is provided with the help of the SDC file. The constraints are used to instruct the tool to generate a netlist that should meet the different power, timing and area requirements. To see the effect of different timing constraints, three different constraint files are used to generate the netlist. The first requirement is to generate the netlist with a minimum area by keeping the timing constraints highly relaxed.
The following constraints are used in the design :

- Create_clock -name -period -waveform {} [get_ports clock] : This constraint provides the clock source , the period and the waveform. Waveform specifies the rising and falling edge of the clock.

- Set_clock_transition -rise and set_clock_transition -fall : This attribute is used to set the slew on the clock.

- Set_clock_uncertainty: This is used to apply the uncertainty in the clock. The uncertainty applied is used to reduce the clock period. This constraint specifies the clock jitter and adds some pessimism to the design constraints.

- Clock_latency: This command is used to specify the clock latency. There are two types of latencies available one is the source latency, and the other is the network latency.

- Set_input_delay -min and set_input_delay -max: The signal gets delayed by some specified amount by specifying this constraint. Max delay gives the maximum delay by which signal gets delayed or is the time taken by the input signal to be available at the input of the flip flop.

- Set_output_delay -max and set_output_delay -min: This constraint specifies the time taken by the output signal to reach the reach output port and ensures that the design meets the setup and hold violations.

TCL Commands

- Set_attr lib_search_path: this command is used to search the library in the given path

- Set_attr hdl_search_path: this command is used to search the RTL folder where the Verilog codes are saved

- Set_attr library slow.lib : this command is used to read the slow.lib file present in the library search path

- Read_hdl : this command is used to read all the HDL .v files present in the search path.

- Elaborate : this command is used to create a design from the Verilog modules. Using this command the undefined modules are said to be unresolved or are treated as the blackboxes, as if there is nothing inside that module.

- Read_sdc : this command is used to provide the constraint to the design by reading the .sdc file

- Synthesize -to_mapped -effort medium : this command is used to synthesize the top-level designs and synthesize all the RTL blocks specified in the .v files, optimizing the design and performing the technology mapping. -to mapped, command specifies the mapping of the design to the cells defined in the technology library. The tool performs logic optimization, which provides the smallest possible implementation of the synthesized netlist to meet the desired timing goals. Effort describes the effort which is used to optimize the design, which has been set as medium in our case.

- Write_hdl > syn_report/synthesised_netlist.v: write_hdl command is used to write the gate level synthesized netlist by instantiating all cells from the target technology library in the path provided after “>”.

- Write_sdc > syn_report/dc_file_for_physical_design.sdc : write_sdc command writes the design constraints in the sdc format in the specified path.

- Write_script > syn_report/synthesis_script_sdc.g : using the write_script command the tool writes the script file in the specified path. This is used to generate the script file that contains the design rule constraints of the design.

- Report timing > syn_report/synthesis_timing_report.rep : report timing command is used to generate the timing report of the current design. The report gives the timing report of the worst path by taking the start and the endpoint. Depending upon the worst-case path, the tool computes the slack.

- Report power > syn_report/synthesis_power_report.rep : this command is used to generate the report for the total power consumed in the given path.

- Report gates > syn_report/synthesis_cell_report.rep : this command provides the cell area and describes the area of the sequential and combinational blocks. It also includes the area of buffered logic if used in the synthesized design. In our case, an extra instance has been added by the tool to synthesize the clock divider module, which adds to the area.

- Report area > syn_report/synthesis_area_report.rep: this is used to give the area in the form of modules. This command gives the area of all the cells used in each

module, with a description of the area of every module. Also, it gives the combined area of each of the modules or blocks in the top-level design.

Constraints 3(a)

The synthesis tool does not accept the set_max_area constraint, as the tool has already been optimized to always synthesize with minimum area necessary to meet the design goals. Therefore, other attributes are varied to get the minimum area netlist. The timing constraints are kept relaxed for ensuring the minimum area. The time period is given as 18ns. The clock transition for rise and fall has been given as 0.5ns. The uncertainty is 0.5 ns and the latency is also kept the same . The effect of these attributes are also studied in the timing report generated in the synthesis process. The maximum and minimum input delay is 1.2ns and 0.5ns respectively. The maximum and minimum output delay is 1.5ns and 0.8ns respectively.

```
create_clock -name clk -period 8.0 [get_ports "clk"]
set_clock_transition -rise 0.5 [get_clocks "clk"]
set_clock_transition -fall 0.5 [get_clocks "clk"]
set_clock_uncertainty 0.5 [get_clocks "clk"]
set_clock_latency 0.5 [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "rst"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "rst"] -clock [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "A"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "A"] -clock [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "B"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "B"] -clock [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "C"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "C"] -clock [get_clocks "clk"]

set_input_transition -max 1.2 [get_ports rst]
set_input_transition -min 0.9 [get_ports rst]

set_input_transition -max 1.2 [get_ports A]
set_input_transition -min 0.9 [get_ports A]

set_input_transition -max 5.0 [get_ports B]
set_input_transition -min 0.9 [get_ports B]

set_input_transition -max 1.2 [get_ports C]
set_input_transition -min 0.9 [get_ports C]

set_output_delay -max 1.5 [get_ports "out"] -clock [get_clocks "clk"]
set_output_delay -min 0.8 [get_ports "out"] -clock [get_clocks "clk"]
```

Observation and Analysis for 3(a)

Area Report

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 19.13-s073_1
  Generated on:           Apr 07 2022  11:10:00 pm
  Module:                 rtl_module
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

 Instance          Module          Cell Count  Cell Area  Net Area   Total Area  Wireload
-------------------------------------------------------------------------------------------
rtl_module                             166   1390.425     0.000     1390.425    <none> (D)
  in1      clk_divider_div_value3        7     68.121     0.000       68.121    <none> (D)

 (D) = wireload is default in technology library
```

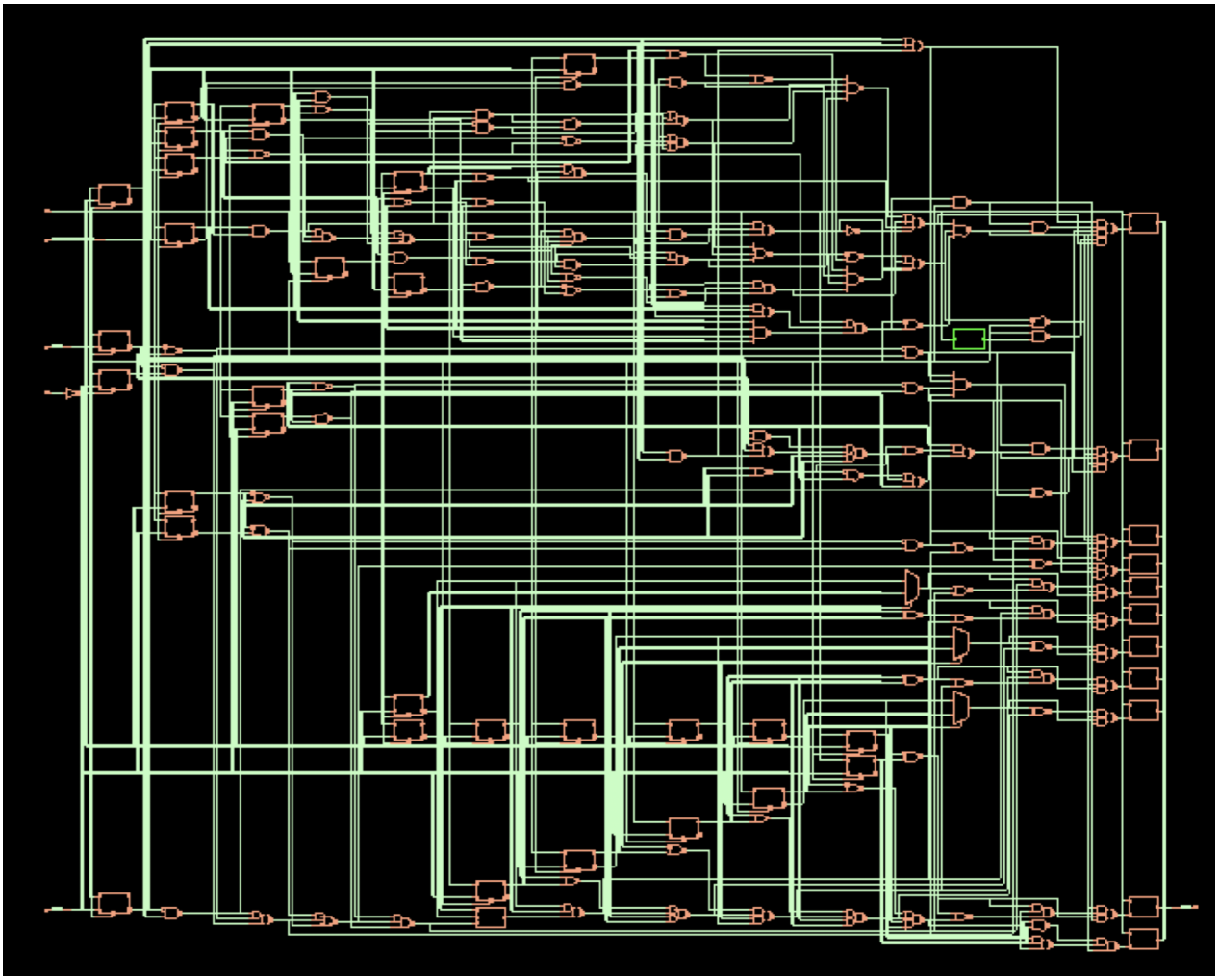
Timing Report

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 19.13-s073_1
  Generated on:           Apr 07 2022  11:09:59 pm
  Module:                 rtl_module
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

             Pin                      Type       Fanout Load Slew Delay Arrival
                                                        (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------------------
(clock clk)                         launch                                0 R
                                    latency                        +500     500 R
(constraints_project__line_13)      ext delay                      +1200    1700 F
B[9]                                in port           1  1.7 5000    +0    1700 F
B_reg_reg[9]/D                <<<   DFFTRXL                          +0    1700
B_reg_reg[9]/CK                     setup                     500 +1967    3667 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                         capture                               8000 R
                                    latency                        +500    8500 R
                                    uncertainty                    -500    8000 R
-------------------------------------------------------------------------------
Cost Group   : 'clk' (path_group 'clk')
Timing slack :    4333ps
Start-point  : B[9]
End-point    : B_reg_reg[9]/D
```

Power Report

```
Instance: /rtl_module
Power Unit: W
PDB Frames: /stim#0/frame#0
  ------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total    Row%
  ------------------------------------------------------------------------
       memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
     register     4.84265e-06  1.06737e-04  2.77957e-06  1.14359e-04  75.34%
        latch     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        logic     2.48904e-06  1.95919e-05  8.15388e-06  3.02348e-05  19.92%
         bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        clock     0.00000e+00  0.00000e+00  7.19887e-06  7.19887e-06   4.74%
          pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
           pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  ------------------------------------------------------------------------
    Subtotal     7.33169e-06  1.26329e-04  1.81323e-05  1.51793e-04 100.00%
  Percentage           4.83%       83.22%       11.95%      100.00% 100.00%
  ------------------------------------------------------------------------
```

Elaborated Schematic Image



Cell Report

```
====================================================================
  Generated by:           Genus(TM) Synthesis Solution 19.13-s073_1
  Generated on:           Apr 07 2022  11:36:56 pm
  Module:                 rtl_module
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
====================================================================


   Gate    Instances    Area   Library
-------------------------------------
AND2X1             2     9.083    slow
AND2XL             1     4.541    slow
AOI211X1           1     5.298    slow
AOI21X1           10    45.414    slow
AOI21XL            1     4.541    slow
AOI22X1            3    18.166    slow
CLKINVX1           1     2.271    slow
DFFQX1             2    31.790    slow
DFFQXL            11   174.844    slow
DFFTRXL           29   592.653    slow
INVX1              1     2.271    slow
INVXL              1     2.271    slow
MXI2XL             3    18.166    slow
NAND2BX1          10    45.414    slow
NAND2XL           16    48.442    slow
NAND3X1            1     4.541    slow
NAND4XL            4    21.193    slow
NOR2BX1            7    31.790    slow
NOR2BXL            3    13.624    slow
NOR2XL            18    54.497    slow
NOR3X1             1     4.541    slow
OA21X1             1     6.812    slow
OAI211X1           2    10.597    slow
OAI21X1           12    54.497    slow
OAI222XL           3    24.978    slow
OAI22X1           10    60.552    slow
OAI2BB1X1          1     5.298    slow
OR2X1              1     4.541    slow
OR2XL              2     9.083    slow
SDFFQX1            1    20.436    slow
XNOR2X1            3    24.978    slow
XNOR2XL            4    33.304    slow
-------------------------------------
total            166  1390.425


     Type      Instances   Area   Area %
----------------------------------------
sequential            43  819.723   59.0
inverter               3    6.812    0.5
logic                120  563.891   40.6
physical_cells         0    0.000    0.0
----------------------------------------
total                166 1390.425  100.0
```

Explanation:

- By varying the value of the time period over a wide range, it was observed that at 8 ns the area was saturated, i.e. no more reduction in the area was observed by increasing the value of the time period beyond this time.
- The number of instances observed are 166 and the minimum area is 1390 with a slack of 4333 ps. It can be observed that as the timing constraint starts to become relaxed, the tool uses the cells of a small area from the library. Whenever the cell of the smallest area is picked, the tool stops to optimize the area further. Hence, the number of cells also reduces in number.
- Since no wireload model was used for synthesis, the wireload is taken as “none” in the area report.
- The power report shows leakage, internal, switching and the total power in watts for all the different circuits or components which have been used in the design.
- From the timing report, we can understand the computation of slack. In the above screenshot of the timing report, the starting point, the end point and the slack obtained for the worst path are visible. It can be seen that the start point is B[9] and the end point is B_reg_reg[9]/D. The arrival time computation starts from the start point and all the delays in the path are added. The final arrival time (AT) is computed by the tool by adding the setup margin of the capture flip flop used in the design. From the above screenshot it can be seen that the setup delay is 1967 ps , hence the net arrival time computed by the tool is 1700 ps +1967 ps = 3667 ps. The required time is decided by the clock time period as the data is captured by the capture flop at the next clock edge, so here the clock time period is 8000 ps, but since the latency provided in the constraint is 500 ps and the uncertainty is also 500 ps. Therefore, the latency would delay the clock by 500 ps to reach the capture flop while the uncertainty would reduce the clock time period by 500 ps , therefore the net required time (RT) is computed as = (8000 ps + 500 ps) - 500 ps = 8000 ps. Since the slack is the difference between the Required time and the arrival time i.e. slack = RT - AT = 8000 ps - 3667 ps, hence the resultant slack is 4333 ps.

Constraints 3(b)

In this part, we had to synthesize the RTL for best timing while keeping the timing constraints tight. The time period was decreased in order to make the timing constraints more tight until a negative slack was observed. All the other attributes were kept constant in the constraints. As the time period reduces the frequency of the clock increases, hence the clock arrives the capture flop much earlier than the data, which can cause setup violations to occur. Therefore, the maximum timing constraint i.e. tperiod + tcapture > tlaunch + tc-q + tlogic +tsetup will not be met. If the time period is reduced then the timing constraint equation will no longer be true and if the clock arrives much earlier than data then there is a high probability of data loss and data corruption. Therefore, the impact of change in time period is chosen to obtain the negative slack in this case.

```
create_clock -name clk -period 2.37 [get_ports "clk"]
set_clock_transition -rise 0.5 [get_clocks "clk"]
set_clock_transition -fall 0.5 [get_clocks "clk"]
set_clock_uncertainty 0.5 [get_clocks "clk"]
set_clock_latency 0.5 [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "rst"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "rst"] -clock [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "A"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "A"] -clock [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "B"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "B"] -clock [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "C"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "C"] -clock [get_clocks "clk"]

set_input_transition -max 1.2 [get_ports rst]
set_input_transition -min 0.9 [get_ports rst]

set_input_transition -max 1.2 [get_ports A]
set_input_transition -min 0.9 [get_ports A]

set_input_transition -max 5.0 [get_ports B]
set_input_transition -min 0.9 [get_ports B]

set_input_transition -max 1.2 [get_ports C]
set_input_transition -min 0.9 [get_ports C]

set_output_delay -max 1.5 [get_ports "out"] -clock [get_clocks "clk"]
set_output_delay -min 0.8 [get_ports "out"] -clock [get_clocks "clk"]
```

Observation and Analysis for 3(b)

Area Report

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 19.13-s073_1
  Generated on:           Apr 08 2022  01:20:24 am
  Module:                 rtl_module
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

 Instance          Module          Cell Count  Cell Area  Net Area   Total Area  Wireload
-------------------------------------------------------------------------------------------
rtl_module                             236   1625.821     0.000     1625.821    <none> (D)
  in1      clk_divider_div_value3        7     68.121     0.000       68.121    <none> (D)

 (D) = wireload is default in technology library
```

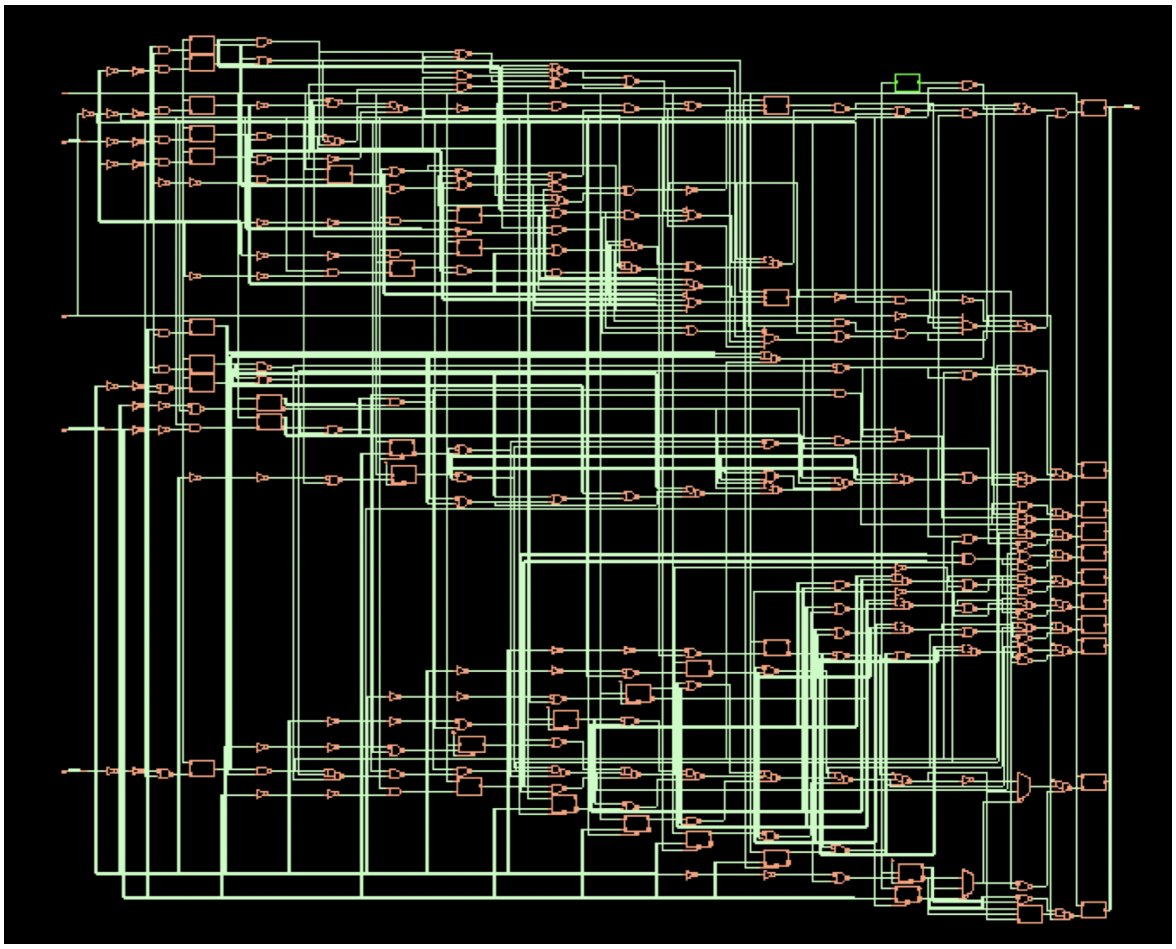
Timing Report

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 19.13-s073_1
  Generated on:           Apr 08 2022  01:20:24 am
  Module:                 rtl_module
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

               Pin                          Type       Fanout Load Slew Delay Arrival
                                                             (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------------------------
(clock clk)                             launch                                    0 R
                                        latency                          +500     500 R
out_reg[0]/CK                                                     500             500 R
out_reg[0]/Q                            DFFQXL              1  0.0   26  +372     872 F
out[0]                            <<<   interconnect                 26    +0     872 F
                                        out port                           +0     872 F
(constraints_project__line_31_62_1)     ext delay                        +1500    2372 F
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                             capture                                  2370 R
                                        latency                          +500    2870 R
                                        uncertainty                      -500    2370 R
-------------------------------------------------------------------------------------
Cost Group   : 'clk' (path_group 'clk')
Timing slack :      -2ps (TIMING VIOLATION)
Start-point  : out_reg[0]/CK
End-point    : out[0]
```

Power Report

```
Instance: /rtl_module
Power Unit: W
PDB Frames: /stim#0/frame#0
  ------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total    Row%
  ------------------------------------------------------------------------
       memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
     register     4.60106e-06  3.60336e-04  8.79681e-06  3.73734e-04  67.67%
        latch     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        logic     4.33847e-06  1.10901e-04  3.89368e-05  1.54176e-04  27.91%
         bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        clock     0.00000e+00  0.00000e+00  2.44025e-05  2.44025e-05   4.42%
          pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
           pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  ------------------------------------------------------------------------
    Subtotal     8.93953e-06  4.71237e-04  7.21361e-05  5.52313e-04 100.00%
  Percentage           1.62%       85.32%       13.06%      100.00% 100.00%
  ------------------------------------------------------------------------
```

Elaborated Schematic Image



Cell Report

```
==========================================================
  Generated by:           Genus(TM) Synthesis Solution 19.13-s073_1
  Generated on:           Apr 08 2022  01:20:24 am
  Module:                 rtl_module
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
==========================================================


   Gate    Instances    Area   Library
-----------------------------------------
AND2X1             4    18.166    slow
AND2XL            15    68.121    slow
AOI21X1            7    31.790    slow
AOI21X2            4    33.304    slow
AOI21XL            6    27.248    slow
AOI32X1            2    13.624    slow
CLKBUFX2           1     4.541    slow
CLKINVX1          24    54.497    slow
CLKINVX12          2    24.221    slow
CLKINVX2           1     3.784    slow
CLKINVX3           2     9.083    slow
DFFQX1            16   254.318    slow
DFFQX2             4    66.607    slow
DFFQXL            11   174.844    slow
DFFRHQX1           1    20.436    slow
DFFRX1             1    21.950    slow
DFFTRXL            4    81.745    slow
DFFX1              5    94.613    slow
INVX1              2     4.541    slow
INVXL              5    11.354    slow
MXI2X1             1     6.812    slow
MXI2XL             1     6.055    slow
NAND2BX1          10    45.414    slow
NAND2BX2           1     6.812    slow
NAND2X1            4    15.138    slow
NAND2X2            3    18.166    slow
NAND2X4            1     9.840    slow
NAND2XL           12    36.331    slow
NAND4BX1           1     7.569    slow
NAND4BXL           1     6.812    slow
NAND4X6            1    23.464    slow
NOR2BX1           10    45.414    slow
NOR2BX2            1     6.812    slow
NOR2BXL           14    63.580    slow
NOR2X1             1     3.784    slow
NOR2X2             1     6.055    slow
NOR2XL            16    48.442    slow
NOR4BX1            1     6.812    slow
OAI211X1           1     5.298    slow
OAI21X1           11    49.955    slow
OAI21X2            2    16.652    slow
OAI21XL            2     9.083    slow
OAI222X1           1     9.083    slow
OAI22X1            3    18.166    slow
OAI22XL            6    36.331    slow
OAI2BB1X1          2    10.597    slow
OR2X1              1     4.541    slow
OR2XL              3    13.624    slow
SDFFQX1            1    20.436    slow
XNOR2X1            3    24.978    slow
XNOR2XL            2    16.652    slow
XOR2XL             1     8.326    slow
-----------------------------------------
total            236  1625.821


     Type      Instances   Area   Area %
-----------------------------------------------
sequential            43  734.950   45.2
inverter              36  107.480    6.6
buffer                 1    4.541    0.3
logic                156  778.850   47.9
physical_cells         0    0.000    0.0
-----------------------------------------------
total                236 1625.821  100.0
```

Explanation:

- By varying the value of the time period over a wide range, it was observed that at 2.37 ns we got the slack as negative and for any value above that, slack was either 0 or positive.
- The number of instances observed are 236 and the area is 1625.821 with a slack of -2 ps. It can be observed that as the timing constraint starts to become tight, the tool uses the cells of a larger area from the library to meet the tight time constraints.
- Since no wireload model was used for synthesis, the wireload is taken as “none” in the area report.
- The power report shows leakage, internal, switching and the total power in watts for all the different circuits or components which have been used in the design.
- From the timing report, we can understand the computation of slack. In the above screenshot of the timing report, the starting point, the end point and the slack obtained for the worst path are visible. It can be seen that the start point is out_reg[0]/CK and the end point is out[0]. The arrival time computation starts from the start point and all the delays in the path are added. The final arrival time (AT) is computed by the tool by adding the setup margin of the capture flip flop used in the design. From the above screenshot it can be seen that the external delay is 1500 ps , hence the net arrival time computed by the tool is 872 ps + 1500 ps = 2372 ps. The required time is decided by the clock time period as the data is captured by the capture flop at the next clock edge, so here the clock time period is 2370 ps, but since the latency provided in the constraint is 500 ps and the uncertainty is also 500 ps. Therefore, the latency would delay the clock by 500 ps to reach the capture flop while the uncertainty would reduce the clock time period by 500 ps , therefore the net required time (RT) is computed as = (2370 ps + 500 ps) - 500 ps = 2370 ps. Since the slack is the difference between the Required time and the arrival time i.e. slack = RT - AT = 2370 ps - 2372 ps, hence the resultant slack is -2 ps.

Constraints 3(c)

In this part, the constraint file had to be written in such a way that the timing constraints lie between that of part (a) and part (b). Since in both the parts, we only changed the time period so for this part as well, we only change the time period and choose the value of time period to be between 2.37 ns and 8 ns.

```
create_clock -name clk -period 4.0 [get_ports "clk"]
set_clock_transition -rise 0.5 [get_clocks "clk"]
set_clock_transition -fall 0.5 [get_clocks "clk"]
set_clock_uncertainty 0.5 [get_clocks "clk"]
set_clock_latency 0.5 [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "rst"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "rst"] -clock [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "A"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "A"] -clock [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "B"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "B"] -clock [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "C"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "C"] -clock [get_clocks "clk"]

set_input_transition -max 1.2 [get_ports rst]
set_input_transition -min 0.9 [get_ports rst]

set_input_transition -max 1.2 [get_ports A]
set_input_transition -min 0.9 [get_ports A]

set_input_transition -max 5.0 [get_ports B]
set_input_transition -min 0.9 [get_ports B]

set_input_transition -max 1.2 [get_ports C]
set_input_transition -min 0.9 [get_ports C]

set_output_delay -max 1.5 [get_ports "out"] -clock [get_clocks "clk"]
set_output_delay -min 0.8 [get_ports "out"] -clock [get_clocks "clk"]
```

Observation and Analysis for 3(c)

Area Report

```
 1 ============================================================
 2   Generated by:           Genus(TM) Synthesis Solution 19.13-s073_1
 3   Generated on:           Apr 07 2022  11:17:09 pm
 4   Module:                 rtl_module
 5   Technology library:     slow
 6   Operating conditions:   slow (balanced_tree)
 7   Wireload mode:          enclosed
 8   Area mode:              timing library
 9 ============================================================
10
11  Instance          Module          Cell Count  Cell Area  Net Area   Total Area  Wireload
12 -----------------------------------------------------------------------------------------
13 rtl_module                            171   1395.724 |   0.000     1395.724    <none> (D)
14   in1      clk_divider_div_value3        7     68.121     0.000       68.121    <none> (D)
15
16  (D) = wireload is default in technology library
```

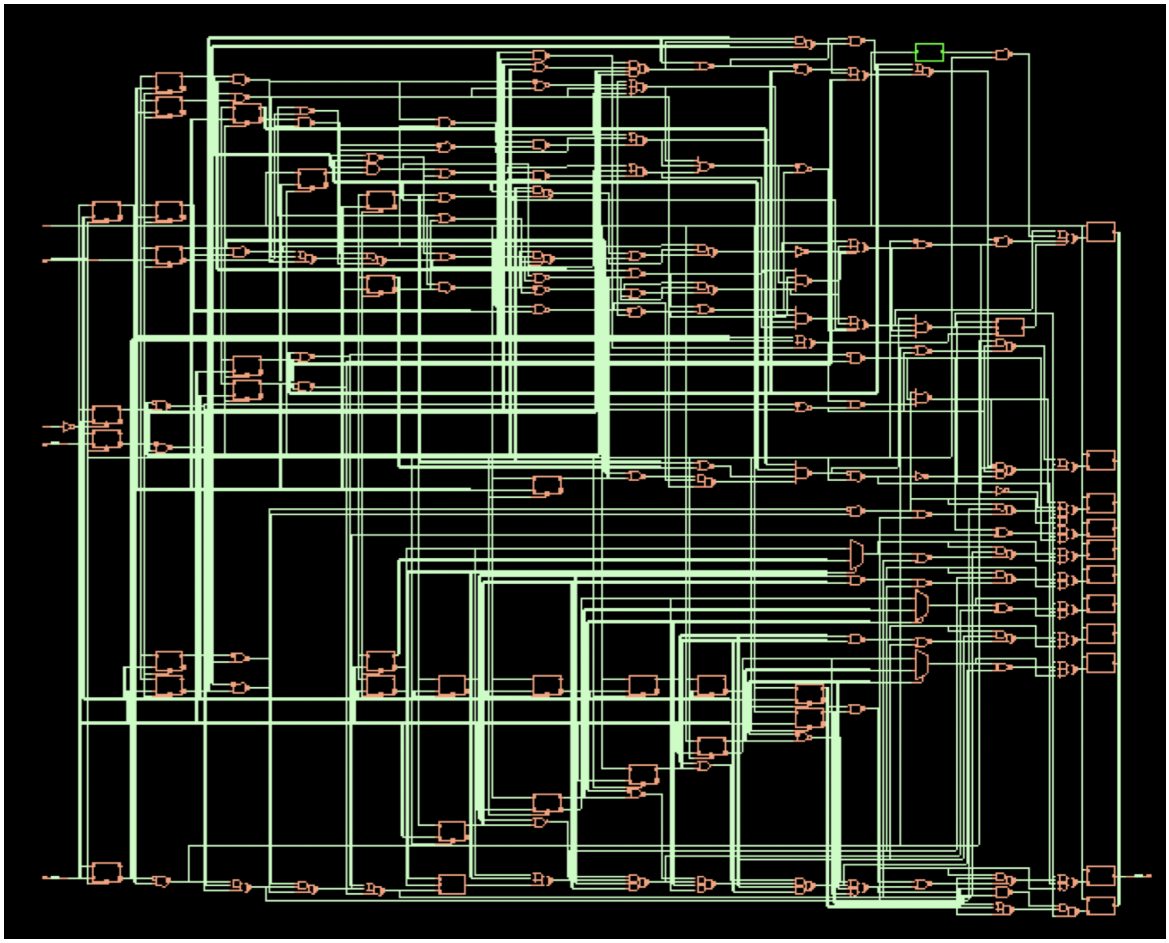
Timing Report

```
 5   Technology library:     slow
 6   Operating conditions:   slow (balanced_tree)
 7   Wireload mode:          enclosed
 8   Area mode:              timing library
 9 ============================================================
10
11             Pin                       Type       Fanout Load Slew Delay Arrival
12                                                      (fF) (ps)  (ps)   (ps)
13 -------------------------------------------------------------------------------
14 (clock clk)                          launch                                  0 R
15                                      latency                  +500     500 R
16 (constraints_project__line_7)        ext delay                +1200    1700 F
17 rst                                  in port           1  2.7 1200    +0    1700 F
18 g528/A                                                              +0    1700
19 g528/Y                               CLKINVX1         33 54.3  618  +736    2436 R
20 g2617/D                                                             +0    2436
21 g2617/Y                              NAND4XL          11 18.2  932  +757    3192 F
22 g2604/A1N                                                           +0    3192
23 g2604/Y                              OAI2BB1X1         1  2.9  107  +266    3459 F
24 g2600/C0                                                            +0    3459
25 g2600/Y                              OAI211X1          1  1.6  119   +61    3520 R
26 out_reg[0]/D                  <<<    DFFQXL                          +0    3520
27 out_reg[0]/CK                        setup                    500  +160    3680 R
28 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
29 (clock clk)                          capture                                 4000 R
30                                      latency                  +500    4500 R
31                                      uncertainty              -500    4000 R
32 -------------------------------------------------------------------------------
33 Cost Group   : 'clk' (path_group 'clk')
34 Timing slack :     320ps
35 Start-point  : rst
36 End-point    : out_reg[0]/D
```

Power Report

```
Instance: /rtl_module
Power Unit: W
PDB Frames: /stim#0/frame#0
  ------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total    Row%
  ------------------------------------------------------------------------
       memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
     register     4.84265e-06  2.13120e-04  5.47979e-06  2.23442e-04  76.13%
        latch     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        logic     2.44785e-06  3.78117e-05  1.54102e-05  5.56697e-05  18.97%
         bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        clock     0.00000e+00  0.00000e+00  1.43977e-05  1.43977e-05   4.91%
          pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
           pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  ------------------------------------------------------------------------
    Subtotal     7.29050e-06  2.50932e-04  3.52877e-05  2.93510e-04 100.01%
  Percentage           2.48%       85.49%       12.02%      100.00% 100.00%
  ------------------------------------------------------------------------
```

Elaborated Schematic Image



Cell Report

```
====================================================================
  Generated by:           Genus(TM) Synthesis Solution 19.13-s073_1
  Generated on:           Apr 07 2022  11:41:11 pm
  Module:                 rtl_module
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
====================================================================


   Gate    Instances    Area   Library
--------------------------------------
AND2X1             1     4.541    slow
AOI211X1           1     5.298    slow
AOI21X1            7    31.790    slow
AOI21XL            6    27.248    slow
AOI22X1            3    18.166    slow
AOI32X1            1     6.812    slow
CLKINVX1           1     2.271    slow
DFFQX1             2    31.790    slow
DFFQXL            11   174.844    slow
DFFTRXL           29   592.653    slow
INVX1              2     4.541    slow
INVXL              2     4.541    slow
MXI2XL             3    18.166    slow
NAND2BX1          11    49.955    slow
NAND2XL           11    33.304    slow
NAND3X1            1     4.541    slow
NAND4XL            4    21.193    slow
NOR2BX1            9    40.873    slow
NOR2BXL            2     9.083    slow
NOR2XL            25    75.690    slow
NOR3X1             1     4.541    slow
OAI211X1           3    15.895    slow
OAI21X1           11    49.955    slow
OAI21XL            1     4.541    slow
OAI222XL           1     8.326    slow
OAI22X1            3    18.166    slow
OAI22XL            7    42.386    slow
OAI2BB1X1          2    10.597    slow
OR2XL              3    13.624    slow
SDFFQX1            1    20.436    slow
XNOR2X1            3    24.978    slow
XNOR2XL            3    24.978    slow
--------------------------------------
total            171  1395.724


     Type      Instances   Area   Area %
-----------------------------------------
sequential            43  819.723   58.7
inverter               5   11.354    0.8
logic                123  564.647   40.5
physical_cells         0    0.000    0.0
-----------------------------------------
total                171 1395.724  100.0
```

Explanation:

- By varying the value of the time period over a wide range, it was observed that at 2.37 ns we got the slack as negative and for a time period equal to 8 ns we got the minimum area. So for this part we have chosen the time period as 4 ns which lies between these 2 values.
- The number of instances observed are 171 and the area is 1395.724 with a slack of 320 ps.
- Since no wireload model was used for synthesis, the wireload is taken as “none” in the area report.
- The power report shows leakage, internal, switching and the total power in watts for all the different circuits or components which have been used in the design.
- From the timing report, we can understand the computation of slack. In the above screenshot of the timing report, the starting point, the end point and the slack obtained for the worst path are visible. It can be seen that the start point is rst and the end point is out reg[0]/D. The arrival time computation starts from the start point and all the delays in the path are added. The final arrival time (AT) is computed by the tool by adding the setup margin of the capture flip flop used in the design. From the above screenshot it can be seen that the setup delay is 160 ps , hence the net arrival time computed by the tool is 3520 ps + 160 ps = 3680 ps. The required time is decided by the clock time period as the data is captured by the capture flop at the next clock edge, so here the clock time period is 4000 ps, but since the latency provided in the constraint is 500 ps and the uncertainty is also 500 ps. Therefore, the latency would delay the clock by 500 ps to reach the capture flop while the uncertainty would reduce the clock time period by 500 ps , therefore the net required time (RT) is computed as = (4000 ps + 500 ps) - 500 ps = 4000 ps. Since the slack is the difference between the Required time and the arrival time i.e. slack = RT - AT = 4000 ps - 3680 ps, hence the resultant slack is 320 ps.

Conclusion - Impact on quality of results

It can be seen that there is a trade-off between the performance and area. If the performance is compromised like in 3(a) minimum area case, where the frequency of operation is much slower, the area gets improved i.e the area is minimized. But, as the timing constraint becomes much tighter where the slack becomes negative, i.e. the setup violations start to occur. The frequency of operation is very high because the cells of a large area are used by the tool compared to case 3(a). Also, the number of cells is increased, and the tool uses the large cells from the library to allow the cells to switch faster. For the constraint when the period is given as 4 ns, the area is 1395, which is lesser than the area obtained at T = 2.37 ns but greater than the area obtained at T = 8 ns, therefore this is the optimal time at which the circuit will operate correctly without much compromising with power, performance and area. At these timing constraints, the timing slack is 320 ps, the total power consumption is 293.510 uW, and the area is 1395. VLSI designing is all about making the trade-offs and coming up with a design that does not compromise one factor to a greater extent. Therefore, this part of the project helps to understand the impact of changing timing constraints on the quality of results, i.e., power, performance, and area.

PART 4
Formal Equivalence Checking

Aim: To do the Formal Equivalence Checking between the RTL file containing verilog code and the netlists generated in 3(a), 3(b) and 3(c). Then, make the changes in the netlist to create a bad netlist to make it non-equivalent with the RTL. Study the failure for this netlist and analyze the results. Report the different error messages that are reported by the formal equivalence tool.

Tool: Cadence Conformal

Theory and About the Tool

Formal Equivalence Checking is part of the formal verification in which two designs are verified for the same functional behavior. Conformal is a very efficient tool to carry out the equivalence as it takes much less time to establish the equivalence between the two designs. Conformal, by default, uses name-based mapping to carry out the equivalence between the two designs. It also maps the registers of one design to the registers available in the other design. Also, the tool works in two modes: first is the setup mode, in which it reads the golden design, revised design, the library and the constraints that are used. The other mode is the LEC mode, in which it performs the mapping of the input/output ports and does the register matching and then carries out the comparison between the two. If any mismatches are found, then the tool issues the report as non-equivalent, and a diagnosis can be made to see the various un-mapped ports in the two designs. Finally, after the complete verification is carried out, the equivalence checking report as “Pass” is generated.
In this part, the formal equivalence checking is performed three times to verify the three generated netlists with different constraints against the RTL Verilog Code.

TCL Commands

Initially, to run the conformal equivalence tool, a do file is written which has all the commands for running the equivalence between the two designs. The do file contains the commands which are explained below:

- Read_library : this command is used to read the library.

- Read_design [specify all .v files] -golden : this command reads all the .v files and set them as the golden design
- Read_design [read the synthesized netlist.v file] -revised: this command reads the synthesized netlist and sets it as the revised netlist.
- Set_system_mode lec: this command is used to set the mode to the lec mode
- Add_compared_points -all: the tool compares the designs hierarchically, then compares the individual pairs of the inputs/outputs and the registers in a bottom-up fashion.
- Compare: This command converts the design into the canonical form.
- Report_messages -compare -verb: this command is used to report all the messages and warnings while the tool makes the comparison
- Report_compare_data -noneq: this command is used to generate the non-equivalent points in the design
- Report_verification : This command generated the verification report
- Write_verification_information: This command generates the final verification information, which shows the result as "pass" or “noneq”, which means the two designs are non-equivalent.

Results

The below screenshots show the successful equivalence checking report generated from the tool checking for all the 3 netlists. We have verified that all the netlists are equivalent to the RTL by getting “pass” results and all the mapped points equivalent to each other.

RTL vs Netlist-3(a)

```
// Mapping key points ...
================================================================================
Mapped points: SYSTEM class
--------------------------------------------------------------------------------
Mapped points     PI     PO     DFF       Total
--------------------------------------------------------------------------------
Golden            31     11     43        85
--------------------------------------------------------------------------------
Revised           31     11     43        85
================================================================================
0
// Command: add_compared_points -all
// 54 compared points added to compare list
0
// Command: compare
================================================================================
Compared points      PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent           11     43        54
================================================================================
0
// Command: report_messages -compare -verb
0
// Command: report_compare_data -noneq
0 Non-equivalent point(s) reported
0 compared point(s) reported
================================================================================
Compared points      PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent           11     43        54
================================================================================
0

// Command: report_verification
================================================================================
                            Verification Report
--------------------------------------------------------------------------------
Category                                                                  Count
--------------------------------------------------------------------------------
1. Non-standard modeling options used:                                      0
--------------------------------------------------------------------------------
2. Incomplete verification:                                                 0
--------------------------------------------------------------------------------
3. User modification to design:                                             0
--------------------------------------------------------------------------------
4. Conformal Constraint Designer clock domain crossing checks recommended:  0
--------------------------------------------------------------------------------
5. Design ambiguity:                                                        0
--------------------------------------------------------------------------------
6. Compare Results:                                                      PASS
================================================================================
pass
```

RTL vs Netlist-(3b)

```
// Mapping key points ...
================================================================================
Mapped points: SYSTEM class
--------------------------------------------------------------------------------
Mapped points     PI     PO     DFF       Total
--------------------------------------------------------------------------------
Golden            31     11     43        85
--------------------------------------------------------------------------------
Revised           31     11     43        85
================================================================================
0
// Command: add_compared_points -all
// 54 compared points added to compare list
0
// Command: compare
================================================================================
Compared points      PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent           11     43        54
================================================================================
0
// Command: report_messages -compare -verb
0
// Command: report_compare_data -noneq
0 Non-equivalent point(s) reported
0 compared point(s) reported
================================================================================
Compared points      PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent           11     43        54
================================================================================
0

// Command: report_verification
================================================================================
                           Verification Report
--------------------------------------------------------------------------------
Category                                                                  Count
--------------------------------------------------------------------------------
1. Non-standard modeling options used:                                      0
--------------------------------------------------------------------------------
2. Incomplete verification:                                                 0
--------------------------------------------------------------------------------
3. User modification to design:                                             0
--------------------------------------------------------------------------------
4. Conformal Constraint Designer clock domain crossing checks recommended:  0
--------------------------------------------------------------------------------
5. Design ambiguity:                                                        0
--------------------------------------------------------------------------------
6. Compare Results:                                                      PASS
================================================================================
pass
```

RTL vs Netlist-(3c)

```
// Mapping key points ...
================================================================================
Mapped points: SYSTEM class
--------------------------------------------------------------------------------
Mapped points     PI     PO     DFF       Total
--------------------------------------------------------------------------------
Golden            31     11     43        85
--------------------------------------------------------------------------------
Revised           31     11     43        85
================================================================================
0
// Command: add_compared_points -all
// 54 compared points added to compare list
0
// Command: compare
================================================================================
Compared points      PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent           11     43        54
================================================================================
0
// Command: report_messages -compare -verb
0
// Command: report_compare_data -noneq
0 Non-equivalent point(s) reported
0 compared point(s) reported
================================================================================
Compared points      PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent           11     43        54
================================================================================
0
// Command: report_verification
================================================================================
                           Verification Report
--------------------------------------------------------------------------------
Category                                                                  Count
--------------------------------------------------------------------------------
1. Non-standard modeling options used:                                      0
--------------------------------------------------------------------------------
2. Incomplete verification:                                                 0
--------------------------------------------------------------------------------
3. User modification to design:                                             0
--------------------------------------------------------------------------------
4. Conformal Constraint Designer clock domain crossing checks recommended:  0
--------------------------------------------------------------------------------
5. Design ambiguity:                                                        0
--------------------------------------------------------------------------------
6. Compare Results:                                                      PASS
================================================================================
pass
```

Bad Netlist

Now we have to manually change the generated netlists in order to make it non-equivalent with the RTL. We have made the bad netlist in several different ways in order to get different errors and interpret or analyze the results.

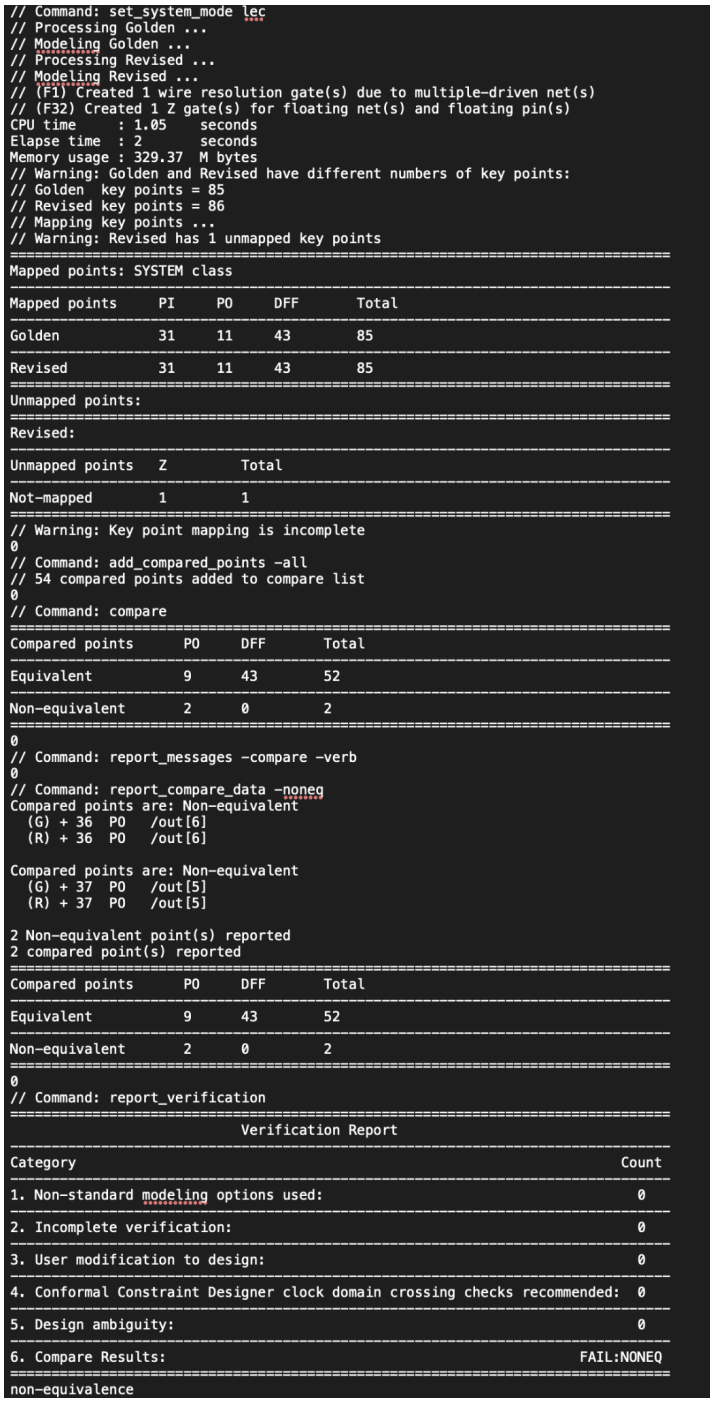
Bad Netlist 1

Firstly, the netlist that was generated in the 3(a) for minimum area part is used for converting into the bad netlist: Below shown is the snippet from the synthesized netlist 3(a). Below is attached a screenshot with correct mapping of out[6] in the top module, and in the next screenshot shows the changed value of the out[6] to out[5] in the same module. This creates the unmapped register i.e. out[6]. (line no. 124)

```
119   DFFQXL \out_reg[1] (.CK (clk), .D (n_140), .Q (out[1]));
120   DFFQXL \out_reg[2] (.CK (clk), .D (n_135), .Q (out[2]));
121   DFFQXL \out_reg[3] (.CK (clk), .D (n_137), .Q (out[3]));
122   DFFQXL \out_reg[4] (.CK (clk), .D (n_138), .Q (out[4]));
123   DFFQXL \out_reg[5] (.CK (clk), .D (n_136), .Q (out[5]));
124   DFFQXL \out_reg[6] (.CK (clk), .D (n_139), .Q (out[6]));
125   DFFQXL \out_reg[7] (.CK (clk), .D (n_142), .Q (out[7]));
126   DFFQXL \out_reg[8] (.CK (clk), .D (n_148), .Q (out[8]));
```

```
119   DFFQXL \out_reg[1] (.CK (clk), .D (n_140), .Q (out[1]));
120   DFFQXL \out_reg[2] (.CK (clk), .D (n_135), .Q (out[2]));
121   DFFQXL \out_reg[3] (.CK (clk), .D (n_137), .Q (out[3]));
122   DFFQXL \out_reg[4] (.CK (clk), .D (n_138), .Q (out[4]));
123   DFFQXL \out_reg[5] (.CK (clk), .D (n_136), .Q (out[5]));
124   DFFQXL \out_reg[6] (.CK (clk), .D (n_139), .Q (out[5]));
125   DFFQXL \out_reg[7] (.CK (clk), .D (n_142), .Q (out[7]));
126   DFFQXL \out_reg[8] (.CK (clk), .D (n_148), .Q (out[8]));
```

After running the tool with a bad netlist, we get the following results.



```
// Command: set_system_mode lec
// Processing Golden ...
// Modeling Golden ...
// Processing Revised ...
// Modeling Revised ...
// (F1) Created 1 wire resolution gate(s) due to multiple-driven net(s)
// (F32) Created 1 Z gate(s) for floating net(s) and floating pin(s)
CPU time     : 1.05    seconds
Elapse time  : 2       seconds
Memory usage : 329.37  M bytes
// Warning: Golden and Revised have different numbers of key points:
// Golden  key points = 85
// Revised key points = 86
// Mapping key points ...
// Warning: Revised has 1 unmapped key points
================================================================================
Mapped points: SYSTEM class
--------------------------------------------------------------------------------
Mapped points     PI     PO     DFF       Total
--------------------------------------------------------------------------------
Golden            31     11     43        85
--------------------------------------------------------------------------------
Revised           31     11     43        85
================================================================================
Unmapped points:
================================================================================
Revised:
--------------------------------------------------------------------------------
Unmapped points   Z         Total
--------------------------------------------------------------------------------
Not-mapped        1         1
================================================================================
// Warning: Key point mapping is incomplete
0
// Command: add_compared_points -all
// 54 compared points added to compare list
0
// Command: compare
================================================================================
Compared points      PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent           9      43        52
--------------------------------------------------------------------------------
Non-equivalent       2      0         2
================================================================================
0
// Command: report_messages -compare -verb
0
// Command: report_compare_data -noneq
Compared points are: Non-equivalent
  (G) + 36  PO   /out[6]
  (R) + 36  PO   /out[6]

Compared points are: Non-equivalent
  (G) + 37  PO   /out[5]
  (R) + 37  PO   /out[5]

2 Non-equivalent point(s) reported
2 compared point(s) reported
================================================================================
Compared points      PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent           9      43        52
--------------------------------------------------------------------------------
Non-equivalent       2      0         2
================================================================================
0
// Command: report_verification
================================================================================
                           Verification Report
--------------------------------------------------------------------------------
Category                                                                  Count
--------------------------------------------------------------------------------
1. Non-standard modeling options used:                                      0
--------------------------------------------------------------------------------
2. Incomplete verification:                                                 0
--------------------------------------------------------------------------------
3. User modification to design:                                             0
--------------------------------------------------------------------------------
4. Conformal Constraint Designer clock domain crossing checks recommended:  0
--------------------------------------------------------------------------------
5. Design ambiguity:                                                        0
--------------------------------------------------------------------------------
6. Compare Results:                                                FAIL:NONEQ
================================================================================
non-equivalence
```

Result: In the above screenshot, we can see that the key point mapping is incomplete. After comparing, two compared points show non-equivalent and therefore the final compare result comes out to be “NONEQ” which means non-equivalence. There is a port missing in the primary output under the unmapped points category. Here, in this

case the out[6] takes the value of the input port B which then is used for generating the output. But, here this point is unreachable in the netlist. Therefore the out[6] pin is not mapped and the non-equivalent report contains the broken connection dues to the absence of mapping of one register.

Bad Netlist 2

The second bad netlist is generated from the netlist generated from the 3(b) part. This netlist is changed such that there are some undriven nets available at the output in the module. The screenshot attached below shows the correct netlist and the next screenshot shows the bad netlist in which in line 196 the output Y is mapped to 0 and the nets of A and B for the Nand gate have been interchanged. Hence, this generates the non-equivalent report as the result.

```
193   OAI21X1 g2307(.A0 (n_105), .A1 (n_116), .B0 (n_123), .Y (n_127));
194   NOR2XL g2308(.A (n_101), .B (n_121), .Y (n_125));
195   NAND2X1 g2309(.A (n_81), .B (n_121), .Y (n_124));
196   NAND2XL g2310(.A (n_116), .B (n_105), .Y (0));
197   OAI211X1 g2311(.A0 (B_reg[2]), .A1 (n_46), .B0 (n_49), .C0 (n_114),
198        .Y (n_122));
```

```
193   OAI21X1 g2307(.A0 (n_105), .A1 (n_116), .B0 (n_123), .Y (n_127));
194   NOR2XL g2308(.A (n_101), .B (n_121), .Y (n_125));
195   NAND2X1 g2309(.A (n_81), .B (n_121), .Y (n_124));
196   NAND2XL g2310(.A (n_105), .B (n_116), .Y (n_123));
197   OAI211X1 g2311(.A0 (B_reg[2]), .A1 (n_46), .B0 (n_49), .C0 (n_114),
198        .Y (n_122));
```

After running the tool with a bad netlist, we get the following results.

```
0
// Command: read_design syn_report_3b/synthesised_netlist_bad.v -verilog -revised
// Parsing file syn_report_3b/synthesised_netlist_bad.v ...
// Revised root module is set to 'rtl_module'
// Warning: (RTL2.5) Net is referenced without an assignment. Design verification will be based on set_undriven_signal setting (occurrence:1)
// Warning: (RTL9.2) Design has irregularly used inout/output expression (occurrence:1)
// Warning: (RTL14) Signal has input but it has no output (occurrence:9)
// Warning: (HRC3.6) Port connection width mis-matches. Undriven signals are floating (occurrence:1)
// Warning: There are 1 undriven nets in Revised
// Note: Read VERILOG design successfully
0
// Command: set_system_mode lec
// Processing Golden ...
// Modeling Golden ...
// Processing Revised ...
// Modeling Revised ...
// (F32) Created 1 Z gate(s) for floating net(s) and floating pin(s)
CPU time      : 1.03    seconds
Elapse time   : 2       seconds
Memory usage : 329.38  M bytes
// Warning: Golden and Revised have different numbers of key points:
// Golden  key points = 85
// Revised key points = 86
// Mapping key points ...
// Warning: Revised has 1 unmapped key points
================================================================================
Mapped points: SYSTEM class
--------------------------------------------------------------------------------
Mapped points     PI     PO     DFF       Total
--------------------------------------------------------------------------------
Golden            31     11     43        85
--------------------------------------------------------------------------------
Revised           31     11     43        85
================================================================================
Unmapped points:
================================================================================
Revised:
--------------------------------------------------------------------------------
Unmapped points   Z         Total
--------------------------------------------------------------------------------
Not-mapped        1         1
================================================================================
// Warning: Key point mapping is incomplete
0
// Command: add_compared_points -all
// 54 compared points added to compare list
0
// Command: compare
================================================================================
Compared points      PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent           11     32        43
--------------------------------------------------------------------------------
Non-equivalent       0      11        11
================================================================================
0
// Command: report_messages -compare -verb
0
// Command: report_compare_data -noneq
Compared points are: Non-equivalent
  (G) + 72  DFF  /out_reg[10]
  (R) + 84  DFF  /out_reg[10]/U$1/U$1
Due to these Non-equivalent points:
[DATA]
  (G) + 639 MUX  /U$9/U$11
  (R) + 592 INV  /g2240/U$3

Compared points are: Non-equivalent
  (G) + 73  DFF  /out_reg[9]
  (R) + 83  DFF  /out_reg[9]/U$1/U$1
Due to these Non-equivalent points:
[DATA]
  (G) + 638 MUX  /U$9/U$10
  (R) + 584 INV  /g2238/U$4

Compared points are: Non-equivalent
  (G) + 74  DFF  /out_reg[8]
  (R) + 82  DFF  /out_reg[8]/U$1/U$1
Due to these Non-equivalent points:
[DATA]
  (G) + 637 MUX  /U$9/U$9
  (R) + 603 INV  /g2242/U$4

Compared points are: Non-equivalent
  (G) + 75  DFF  /out_reg[7]
  (R) + 81  DFF  /out_reg[7]/U$1/U$1
Due to these Non-equivalent points:
```

```
  (R) + 617 INV  /g2248/U$4

Compared points are: Non-equivalent
  (G) + 76  DFF  /out_reg[6]
  (R) + 80  DFF  /out_reg[6]/U$1/U$1
Due to these Non-equivalent points:
[DATA]
  (G) + 635 MUX  /U$9/U$7
  (R) + 632 INV  /g2255/U$4

Compared points are: Non-equivalent
  (G) + 77  DFF  /out_reg[5]
  (R) + 79  DFF  /out_reg[5]/U$1/U$1
Due to these Non-equivalent points:
[DATA]
  (G) + 634 MUX  /U$9/U$6
  (R) + 724 INV  /g2268/U$4

Compared points are: Non-equivalent
  (G) + 78  DFF  /out_reg[4]
  (R) + 78  DFF  /out_reg[4]/U$1/U$1
Due to these Non-equivalent points:
[DATA]
  (G) + 633 MUX  /U$9/U$5
  (R) + 735 INV  /g2270/U$4

Compared points are: Non-equivalent
  (G) + 79  DFF  /out_reg[3]
  (R) + 77  DFF  /out_reg[3]/U$1/U$1
Due to these Non-equivalent points:
[DATA]
  (G) + 632 MUX  /U$9/U$4
  (R) + 710 INV  /g2267/U$4

Compared points are: Non-equivalent
  (G) + 80  DFF  /out_reg[2]
  (R) + 76  DFF  /out_reg[2]/U$1/U$1
Due to these Non-equivalent points:
[DATA]
  (G) + 631 MUX  /U$9/U$3
  (R) + 701 INV  /g2263/U$5

Compared points are: Non-equivalent
  (G) + 81  DFF  /out_reg[1]
  (R) + 75  DFF  /out_reg[1]/U$1/U$1
Due to these Non-equivalent points:
[DATA]
  (G) + 630 MUX  /U$9/U$2
  (R) + 690 INV  /g2262/U$3

Compared points are: Non-equivalent
  (G) + 82  DFF  /out_reg[0]
  (R) + 74  DFF  /out_reg[0]/U$1/U$1
Due to these Non-equivalent points:
[DATA]
  (G) + 629 MUX  /U$9/U$1
  (R) + 671 INV  /g2260/U$3

11 Non-equivalent point(s) reported
11 compared point(s) reported
================================================================================
Compared points      PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent           11     32        43
--------------------------------------------------------------------------------
Non-equivalent       0      11        11
================================================================================
0
// Command: report_verification
================================================================================
                            Verification Report
--------------------------------------------------------------------------------
Category                                                                  Count
--------------------------------------------------------------------------------
1. Non-standard modeling options used:                                      0
--------------------------------------------------------------------------------
2. Incomplete verification:                                                 0
--------------------------------------------------------------------------------
3. User modification to design:                                             0
--------------------------------------------------------------------------------
4. Conformal Constraint Designer clock domain crossing checks recommended:  0
--------------------------------------------------------------------------------
5. Design ambiguity:                                                        0
--------------------------------------------------------------------------------
6. Compare Results:                                              FAIL:NONEQ
================================================================================
```

Result: In the above screenshot, the details for the warning generated for the undriven net is visible. It shows by making the bad netlist there is an undriven net found in the golden and the revised netlist. It can be seen that the warning is displayed for the floating assignment which is assigned with a don't care value. Hence, it can be inferred from the above warnings that the output Y of the module in the netlist which is mapped to 0 creates a floating net or floating pin. Also, the wire interchange between A and B port of the gate causes the unmapping of the net from the module. Therefore, we get a floating pin or net and one unmapped key point in the above attached screenshot. The final result of the equivalence check is non-equivalence since due to the unmapped point, there are 11 non-equivalence reported by the tool.

Bad Netlist 3

The second bad netlist is generated from the netlist generated from the 3(c) part. This netlist is changed such that there are no unmapped points reported after the equivalence check and still we do not get the pass result. The screenshot attached below shows the correct netlist and the next screenshot shows the bad netlist in which the nets for port AN for the two NOR gates have been interchanged in line 20 and 21.

```
14  SDFFQX1 clk_out_reg(.CK (clk_in), .D (n_0), .SI (clk_out), .SE (n_2),
15       .Q (clk_out));
16  DFFQX1 \count_reg_reg[1] (.CK (clk_in), .D (n_7), .Q (count_reg[1]));
17  DFFQX1 \count_reg_reg[0] (.CK (clk_in), .D (n_8), .Q (count_reg[0]));
18  NAND2BX1 g63(.AN (count_reg[0]), .B (count_reg[1]), .Y (n_2));
19  INVXL g65(.A (clk_out), .Y (n_0));
20  NOR2BXL g2(.AN (count_reg[0]), .B (count_reg[1]), .Y (n_7));
21  NOR2BXL g67(.AN (n 2), .B (count reg[0]), .Y (n 8));
```

```
14  SDFFQX1 clk_out_reg(.CK (clk_in), .D (n_0), .SI (clk_out), .SE (n_2),
15       .Q (clk_out));
16  DFFQX1 \count_reg_reg[1] (.CK (clk_in), .D (n_7), .Q (count_reg[1]));
17  DFFQX1 \count_reg_reg[0] (.CK (clk_in), .D (n_8), .Q (count_reg[0]));
18  NAND2BX1 g63(.AN (count_reg[0]), .B (count_reg[1]), .Y (n_2));
19  INVXL g65(.A (clk_out), .Y (n_0));
20  NOR2BXL g2(.AN (n_2), .B (count_reg[1]), .Y (n_7));
21  NOR2BXL g67(.AN (count reg[0]), .B (count reg[0]), .Y (n 8));
```

After running the tool with a bad netlist, we get the following results.

```
// Command: read_design ../rtl/testbench1/top.v -verilog -golden
// Parsing file ../rtl/testbench1/top.v ...
// Golden root module is set to 'rtl_module'
// Warning: (RTL1.5b) Potential loss of RHS msb or carry-out bit (occurrence:1)
// Warning: (RTL7.10a) Comparison with different data sizes (occurrence:2)
// Warning: (DIR6.1) Ignored compiler directive is detected (occurrence:1)
// Warning: (IGN1.1) Initial construct is not supported. The entire initial construct will be ignored (occurrence:2)
// Note: Read VERILOG design successfully
0
// Command: read_design syn_report_3c/synthesised_netlist_bad.v -verilog -revised
// Parsing file syn_report_3c/synthesised_netlist_bad.v ...
// Revised root module is set to 'rtl_module'
// Warning: (RTL14) Signal has input but it has no output (occurrence:22)
// Note: Read VERILOG design successfully
0
// Command: set_system_mode lec
// Processing Golden ...
// Modeling Golden ...
// Processing Revised ...
// Modeling Revised ...
CPU time     : 1.01    seconds
Elapse time  : 2       seconds
Memory usage : 329.38  M bytes
// Mapping key points ...
================================================================================
Mapped points: SYSTEM class
--------------------------------------------------------------------------------
Mapped points     PI     PO     DFF       Total
--------------------------------------------------------------------------------
Golden            31     11     43        85
--------------------------------------------------------------------------------
Revised           31     11     43        85
================================================================================
0
// Command: add_compared_points -all
// 54 compared points added to compare list
0
// Command: compare
================================================================================
Compared points      PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent           11     41        52
--------------------------------------------------------------------------------
Non-equivalent       0      2         2
================================================================================
0
// Command: report_messages -compare -verb
0
// Command: report_compare_data -noneq
Compared points are: Non-equivalent
  (G) + 83  DFF  /in1/count_reg_reg[1]
  (R) + 44  DFF  /in1/count_reg_reg[1]/U$1/U$1
Due to these Non-equivalent points:
[DATA]
  (G) + 649 MUX  /in1/U$1/U$2
  (R) + 278 INV  /in1/g2/U$3

Compared points are: Non-equivalent
  (G) + 84  DFF  /in1/count_reg_reg[0]
  (R) + 45  DFF  /in1/count_reg_reg[0]/U$1/U$1
Due to these Non-equivalent points:
[DATA]
  (G) + 645 MUX  /in1/U$1/U$1
  (R) + 95  INV  /in1/g67/U$3

2 Non-equivalent point(s) reported
2 compared point(s) reported
================================================================================
Compared points      PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent           11     41        52
--------------------------------------------------------------------------------
Non-equivalent       0      2         2
================================================================================
0
```

```
// Command: report_verification
================================================================================
                           Verification Report
--------------------------------------------------------------------------------
Category                                                                  Count
--------------------------------------------------------------------------------
1. Non-standard modeling options used:                                      0
--------------------------------------------------------------------------------
2. Incomplete verification:                                                 0
--------------------------------------------------------------------------------
3. User modification to design:                                             0
--------------------------------------------------------------------------------
4. Conformal Constraint Designer clock domain crossing checks recommended:  0
--------------------------------------------------------------------------------
5. Design ambiguity:                                                        0
--------------------------------------------------------------------------------
6. Compare Results:                                                 FAIL:NONEQ
================================================================================
non-equivalence
```

Result: In the above screenshot, we can see that there are no unmapped points reported by the tool but after the compare command, two non-equivalent points have been reported due to which we get the final result as non-equivalence. From this we understand that it is not necessary to get the pass result in equivalence when there are no unmapped points, correct instantiation to their respective ports is also important.

Conclusion: It is observed that the equivalence checking is a very suitable environment to carry out the verification. This allows to verify whether two designs are functionally equivalent. This is done by establishing one to one equivalence between the Verilog and the generated RTL netlist.

PART 5
Static Timing Analysis

Aim: To perform Static Timing analysis (STA) on the netlist generated in parts (3a), (3b), and (3c). We will be using the constraints file from (3c) to perform STA on the three netlists. Further, we will be explaining the observations and justifying the slack we are getting for the three netlists.

Tool: Cadence Tempus

Theory

Static Timing Analysis is the design step in which we check for timing violations across different paths (especially the worst path) in our circuit design. STA gives the bounds of delay essential to meet timing constraints and for our circuit to function as we intend. The delays might change in the later phases of design. However, we must ensure that whatever we have designed so far is well between the safe window.

Setup Analysis: The data on the input pin of the capture flip flop must arrive 'Setup Time' before the next clock edge arrives on the flip flop.

Hold Analysis: A new data must not arrive on the input pin of the capture flip flop for 'Hold Time' after the clock edge has arrived on the flip flop.

The slack in both cases must be positive to ensure that no timing violation has occurred in the design. In general, Setup violations can be removed by making the design slower (i.e. increasing the clock period). However, hold violations are much harder or even impossible to remove. You must redesign the system to correct hold violations.

We will be using the constraints file with T_{clk} = 4.0 for all three netlists to perform Static Timing Analysis.

- **Constraint- 3a -Relaxed: T_{clk}= 8.0 ns**
- **Constraint- 3b -Tight: T_{clk}= 2.37 ns**
- **Constraint- 3c -Middle: T_{clk}= 4.0 ns**

About the Tool

The tool gives certain estimates about the timing information of our circuit. It is not precise, but still, it gives us an idea about how our timing is in the current situation. We must keep in mind that they will worsen as we move on in further design phases. So, if we already have our delay close to either the upper or lower bound, we might violate the timing constraints as we progress. So, it is better to make our delays more flexible by changing certain things in the design at the present stage itself.

- The tool would first divide the design into several paths called ‘Timing Paths’. These timing paths can be provided to the tool by the user via the script file. A timing path contains information about the start point and the endpoint of the data, as well as the clock.

- Calculates each path's arrival time and required time and computes the slack for each path.

- Checks for any violations taking place in the design.

Constraints (3C)

```
create_clock -name clk -period 4.0 [get_ports "clk"]
set_clock_transition -rise 0.5 [get_clocks "clk"]
set_clock_transition -fall 0.5 [get_clocks "clk"]
set_clock_uncertainty 0.5 [get_clocks "clk"]
set_clock_latency 0.5 [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "rst"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "rst"] -clock [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "A"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "A"] -clock [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "B"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "B"] -clock [get_clocks "clk"]

set_input_delay -max 1.2 [get_ports "C"] -clock [get_clocks "clk"]
set_input_delay -min 0.5 [get_ports "C"] -clock [get_clocks "clk"]

set_input_transition -max 1.2 [get_ports rst]
set_input_transition -min 0.9 [get_ports rst]

set_input_transition -max 1.2 [get_ports A]
set_input_transition -min 0.9 [get_ports A]

set_input_transition -max 5.0 [get_ports B]
set_input_transition -min 0.9 [get_ports B]

set_input_transition -max 1.2 [get_ports C]
set_input_transition -min 0.9 [get_ports C]

set_output_delay -max 1.5 [get_ports "out"] -clock [get_clocks "clk"]
set_output_delay -min 0.8 [get_ports "out"] -clock [get_clocks "clk"]
```

We use the above constraints file to perform STA on all three netlists.

TCL Commands

- check_timing: Performs completeness checks on the timing constraints specified for a design. It generates all the warnings related to timing issues in the design.

- report_timing: Shows the timing paths, i.e. the critical timing path available in the design.

- report_analysis_coverage: Generates a report about the coverage of the timing checks.

- report_analysis_summary: Analyses the paths from reg to reg.

- report_annotated_parasitics: Reports all the parasitics present in the current design

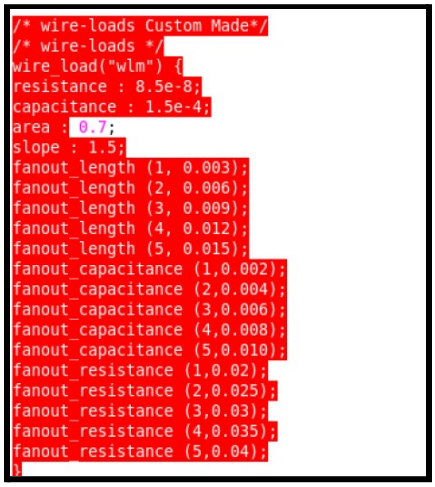
- report_clocks: Reports the clock, for example, the clock period and pulse width and reports if there are generated or propagated clock.

- report_case_analysis: Reports the case analysis on the ports and pins. Also, reports the pins that are set as the case analysis.

- report_constraints -all_violators: Provides the setup and holds violations. Shows the setup path (max delay path) and the hold path(min delay path).

- PBA and GBA reports have been generated as well. (Last 5 lines)

WIRE DELAY MODEL



```
/* wire-loads Custom Made*/
/* wire-loads */
wire_load("wlm") {
resistance : 8.5e-8;
capacitance : 1.5e-4;
area : 0.7;
slope : 1.5;
fanout_length (1, 0.003);
fanout_length (2, 0.006);
fanout_length (3, 0.009);
fanout_length (4, 0.012);
fanout_length (5, 0.015);
fanout_capacitance (1,0.002);
fanout_capacitance (2,0.004);
fanout_capacitance (3,0.006);
fanout_capacitance (4,0.008);
fanout_capacitance (5,0.010);
fanout_resistance (1,0.02);
fanout_resistance (2,0.025);
fanout_resistance (3,0.03);
fanout_resistance (4,0.035);
fanout_resistance (5,0.04);
}
```

NETLIST-A

TCL File

```
file mkdir sta_after_synthesis_3a/reports^M
set report_dir sta_after_synthesis_3a/reports^M

read_lib ../lib/90/lib/slow_wlm.lib^M
read_verilog syn_report_3a/synthesised_netlist.v^M
set_top_module rtl_module^M
read_sdc ../constraints/constraints_project_3c.sdc^M

#Wireload Model
set_wire_load_model -name "wlm"
set_wire_load_mode segmented

check_timing > $report_dir/check_timing.rpt^M
report_timing > $report_dir/timing_report.rpt^M
report_analysis_coverage > $report_dir/analysis_coverage.rpt^M
report_analysis_summary > $report_dir/analysis_summary.rpt^M

#report_annotated_parasitics > $report_dir/annotated.rpt^M

report_clocks > $report_dir/clocks.rpt^M
report_case_analysis > $report_dir/case_analysis.rpt^M
report_constraints -all_violators > $report_dir/allviolations.rpt

#wireload report
report_wire_load > $report_dir/wireload.rpt

report_timing -retime path_slew_propagation -max_path 30 -nworst 30 -path_type full_clock > $report_dir/pba.rpt
report_timing -late -max_paths 30 -nworst 30 -path_type full_clock -net > $report_dir/gba_setup.rpt
report_timing -early -max_paths 30 -nworst 30 -path_type full_clock -net > $report_dir/gba_hold.rpt
report_timing -max_paths 30 -nworst 30 -retime path_slew_propagation > $report_dir/pba_setup.rpt
report_timing -early -max_path 30 -nworst 30 -retime path_slew_propagation > $report_dir/pba_hold.rpt^M
#exit
~
```

Analysis Coverage report

```
#################################################################
#  Generated by:      Cadence Tempus 20.10-p003_1
#  OS:                Linux x86_64(Host ID edaserver4)
#  Generated on:      Thu Apr  7 23:26:16 2022
#  Design:            rtl_module
#  Command:           report_analysis_coverage > $report_dir/analysis_coverage.rpt
#################################################################
    ----------------------------------------------------------------------------------------------
                                TIMING CHECK COVERAGE SUMMARY
    ----------------------------------------------------------------------------------------------
     Check Type                    No. of    Met             Violated         Untested
                                   Checks
    ----------------------------------------------------------------------------------------------
    ExternalDelay (Early)          11        11 (100%)       0 (0%)           0 (0%)
    ExternalDelay (Late)           11        11 (100%)       0 (0%)           0 (0%)
    Hold                           74        33 (44%)        41 (55%)         0 (0%)
    PulseWidth                     86        86 (100%)       0 (0%)           0 (0%)
    Setup                          74        74 (100%)       0 (0%)           0 (0%)
    ----------------------------------------------------------------------------------------------
```

All Violations Report

No Setup violations

```
#####################################################################
#  Generated by:      Cadence Tempus 20.10-p003_1
#  OS:                Linux x86_64(Host ID edaserver4)
#  Generated on:      Thu Apr  7 23:26:16 2022
#  Design:            rtl_module
#  Command:           report_constraints -all_violators > sta_after_synthesis_3a/reports/allviolations.rpt
#####################################################################
# format : frame 0 : split 1

max_delay/setup
---------------------------
No paths found
```

Hold Violations Exist

```
min_delay/hold
---------------------------
      ------------------------------------------------------------------------------------------------------
       End Point                                  Slack      Cause
      ------------------------------------------------------------------------------------------------------
      in1/count_reg_reg[1]/D f                   -0.269     VIOLATED
      in1/count_reg_reg[0]/D f                   -0.264     VIOLATED
      in1/clk_out_reg/D f                        -0.183     VIOLATED
      in1/clk_out_reg/SE f                       -0.176     VIOLATED
      in1/clk_out_reg/SI f                       -0.160     VIOLATED
      out_reg[0]/D f                             -0.082     VIOLATED
      C_reg_reg[8]/D f                           -0.065     VIOLATED
      C_reg_reg[7]/D f                           -0.065     VIOLATED
      C_reg_reg[6]/D f                           -0.065     VIOLATED
      C_reg_reg[5]/D f                           -0.065     VIOLATED
      C_reg_reg[4]/D f                           -0.065     VIOLATED
      C_reg_reg[3]/D f                           -0.065     VIOLATED
      C_reg_reg[2]/D f                           -0.065     VIOLATED
      C_reg_reg[1]/D f                           -0.065     VIOLATED
      C_reg_reg[0]/D f                           -0.065     VIOLATED
      B_reg_reg[9]/D f                           -0.065     VIOLATED
      B_reg_reg[8]/D f                           -0.065     VIOLATED
      B_reg_reg[7]/D f                           -0.065     VIOLATED
      B_reg_reg[6]/D f                           -0.065     VIOLATED
      B_reg_reg[5]/D f                           -0.065     VIOLATED
      B_reg_reg[4]/D f                           -0.065     VIOLATED
      B_reg_reg[3]/D f                           -0.065     VIOLATED
      B_reg_reg[2]/D f                           -0.065     VIOLATED
      B_reg_reg[1]/D f                           -0.065     VIOLATED
      B_reg_reg[0]/D f                           -0.065     VIOLATED
      A_reg_reg[9]/D f                           -0.065     VIOLATED
      A_reg_reg[8]/D f                           -0.065     VIOLATED
```

```
      A_reg_reg[7]/D f                           -0.065     VIOLATED
      A_reg_reg[6]/D f                           -0.065     VIOLATED
      A_reg_reg[5]/D f                           -0.065     VIOLATED
      A_reg_reg[4]/D f                           -0.065     VIOLATED
      A_reg_reg[3]/D f                           -0.065     VIOLATED
      A_reg_reg[2]/D f                           -0.065     VIOLATED
      A_reg_reg[1]/D f                           -0.065     VIOLATED
      A_reg_reg[0]/D f                           -0.065     VIOLATED
      out_reg[10]/D f                            -0.058     VIOLATED
      out_reg[8]/D f                             -0.047     VIOLATED
      out_reg[6]/D f                             -0.047     VIOLATED
      out_reg[9]/D f                             -0.041     VIOLATED
      out_reg[5]/D f                             -0.033     VIOLATED
      out_reg[7]/D f                             -0.032     VIOLATED
      ------------------------------------------------------------------------------------------------------
```

No Other Violations

```
Check type : clock_period
---------------------------
No paths found

Check type : skew
---------------------------
No paths found

Check type : pulse_width
---------------------------
 No violating Checks with given description found

Check type : max_transition
---------------------------
 No Violations found

Check type : min_transition
---------------------------
 No Violations found

Check type : max_capacitance
---------------------------
 No Violations found

Check type : min_capacitance
---------------------------
 No Violations found

Check type : max_fanout
---------------------------
 No Violations found

Check type : min_fanout
```

Timing Report

```
#  Generated by:      Cadence Tempus 20.10-p003_1
#  OS:                Linux x86_64(Host ID edaserver4)
#  Generated on:      Thu Apr  7 23:26:16 2022
#  Design:            rtl_module
#  Command:           report_timing > $report_dir/timing_report.rpt
#####################################################################
Path 1: MET Setup Check with Pin B_reg_reg[0]/CK
Endpoint:   B_reg_reg[0]/D (v) checked with  leading edge of 'clk'
Beginpoint: B[0]           (v) triggered by  leading edge of 'clk'
Path Groups: {clk}
Other End Arrival Time          0.500
- Setup                         1.967
+ Phase Shift                   4.000
- Uncertainty                   0.500
= Required Time                 2.033
- Arrival Time                  1.700
= Slack Time                    0.333
     Clock Rise Edge                          0.000
     + Input Delay                            1.200
     + Network Insertion Delay                0.500
     = Beginpoint Arrival Time                1.700
      ------------------------------------------------------------
      Instance      Arc     Cell     Delay  Arrival  Required
                                             Time     Time
      ------------------------------------------------------------
      -             B[0] v  -        -      1.700    2.033
      B_reg_reg[0]  D v     DFFTRXL  0.000  1.700    2.033
      ------------------------------------------------------------
```

ANALYSIS

Initially, the netlist was generated while keeping a very relaxed constraint of Tclk=8.0ns. However, the STA has been performed on it with Tclk=4.0ns. Which relative to 8.0ns is a tighter constraint for this netlist.

The STA showed us that we have no Setup Violations. All the data successfully reaches the flip flop's input before the setup time of the next clock edge. The hold violations will be eliminated when we do **clock tree synthesis**.

Required Time: The data must reach the next flip flop ‘setup time’ before the next clock edge arrives. Since T_{clk}=4ns and we have an uncertainty of 0.5. In the worst case for setup analysis, the clock can only come after 3.5ns (4-0.5). However, we have a delay of 0.5ns too. This delay and uncertainty cancel, and we are left with 4ns as the time to the next clock edge. The setup time of a flip flop is 1.967ns. Hence, we want our data to be available within 2.033ns (4-1.967) from when the current clock edge has arrived.

Arrival Time: This is the time our data took to reach the input pin of the flip flop. It is estimated as 1.7ns by the tool. We wanted our data to come before 3.033ns, and it came at 1.7ns only. This implies our constraint is met, and we don't have a violation.

Slack: Required Time - Arrival Time= 2.033-1.7= 0.333ns

Critical Path: The following is the critical path according to the tool. That is the path with the worst slack. From B[0] to B_reg_reg[0], both are triggered at the positive edge of the clock.

```
#############################################################
Path 1: MET Setup Check with Pin B_reg_reg[0]/CK
Endpoint:   B_reg_reg[0]/D (v) checked with  leading edge of 'clk'
Beginpoint: B[0]           (v) triggered by  leading edge of 'clk'
Path Groups: {clk}
```

CONCLUSION

The netlist generated in this part was done using the most relaxed constraints file so that we have the minimum area of the design. However, we ran the STA using a relatively tighter constraint. As a result, it is safe to assume that the slack we are getting, even though is positive, has a lower value in comparison to what it could've been if we would've used the relaxed T_{clk} while performing STA as well.

T_{clk} in the constraints file used for STA is not too small for the sizing in our design to cause setup failures. The constraint was supposed to be in between, such that we don't lose too much area and at the same time have a faster system. Had we got setup violations, we would've had to make our system slower by increasing the clock period.

Since the slack is positive, we do not have any setup violations in the netlist generated using T_{clk}=8ns, and STA was performed using T_{clk}=4ns.

NETLIST - B

TCL File

```
file mkdir sta_after_synthesis_3b/reports^M
set report_dir sta_after_synthesis_3b/reports^M

read_lib ../lib/90/lib/slow_wlm.lib^M
read_verilog syn_report_3b/synthesised_netlist.v^M
set_top_module rtl_module^M
read_sdc ../constraints/constraints_project_3c.sdc^M

#Wireload Model
set_wire_load_model -name "wlm"
set_wire_load_mode segmented

check_timing > $report_dir/check_timing.rpt^M
report_timing > $report_dir/timing_report.rpt^M
report_analysis_coverage > $report_dir/analysis_coverage.rpt^M
report_analysis_summary > $report_dir/analysis_summary.rpt^M

#report_annotated_parasitics > $report_dir/annotated.rpt^M

report_clocks > $report_dir/clocks.rpt^M
report_case_analysis > $report_dir/case_analysis.rpt^M
report_constraints -all_violators > $report_dir/allviolations.rpt

#wireload report
report_wire_load > $report_dir/wireload.rpt

report_timing -retime path_slew_propagation -max_path 30 -nworst 30 -path_type full_clock > $report_dir/pba.rpt
report_timing -late -max_paths 30 -nworst 30 -path_type full_clock -net > $report_dir/gba_setup.rpt
report_timing -early -max_paths 30 -nworst 30 -path_type full_clock -net > $report_dir/gba_hold.rpt

report_timing -max_paths 30 -nworst 30 -retime path_slew_propagation > $report_dir/pba_setup.rpt
report_timing -early -max_path 30 -nworst 30 -retime path_slew_propagation > $report_dir/pba_hold.rpt^M
#exit
~
~
~
```

Analysis Coverage Report

```
#################################################################
 Generated by:      Cadence Tempus 20.10-p003_1
 OS:                Linux x86_64(Host ID edaserver4)
 Generated on:      Fri Apr  8 01:20:43 2022
 Design:            rtl_module
 Command:           report_analysis_coverage > $report_dir/analysis_coverage.rpt
#################################################################
  ------------------------------------------------------------------------------------------------
                                TIMING CHECK COVERAGE SUMMARY
  ------------------------------------------------------------------------------------------------
   Check Type                     No. of    Met             Violated         Untested
                                  Checks
  ------------------------------------------------------------------------------------------------
  ExternalDelay (Early)          11        11 (100%)       0 (0%)           0 (0%)
  ExternalDelay (Late)           11        11 (100%)       0 (0%)           0 (0%)
  Hold                           49        31 (63%)        18 (36%)         0 (0%)
  PulseWidth                     88        86 (97%)        0 (0%)           2 (2%)
  Recovery                       2         0 (0%)          0 (0%)           2 (100%)
  Removal                        2         0 (0%)          0 (0%)           2 (100%)
  Setup                          49        49 (100%)       0 (0%)           0 (0%)
  ------------------------------------------------------------------------------------------------
```

All Violations Report

No Setup Violations

```
#  Generated by:      Cadence Tempus 20.10-p003_1
#  OS:                Linux x86_64(Host ID edaserver4)
#  Generated on:      Fri Apr  8 01:20:43 2022
#  Design:            rtl_module
#  Command:           report_constraints -all_violators > sta_after_synthesis_3b/reports/allviolations.rpt
#############################################################
# format : frame 0 : split 1

max_delay/setup
---------------------------
No paths found
```

Hold Violations Exist

```
min_delay/hold
---------------------------
      ------------------------------------------------------------------------------------------------
       End Point                                  Slack      Cause
      ------------------------------------------------------------------------------------------------
      in1/count_reg_reg[1]/D f                  -0.269     VIOLATED
      in1/count_reg_reg[0]/D f                  -0.264     VIOLATED
      in1/clk_out_reg/D f                       -0.177     VIOLATED
      in1/clk_out_reg/SE f                      -0.176     VIOLATED
      in1/clk_out_reg/SI f                      -0.144     VIOLATED
      out_reg[0]/D f                            -0.125     VIOLATED
      out_reg[10]/D f                           -0.100     VIOLATED
      out_reg[4]/D f                            -0.093     VIOLATED
      out_reg[5]/D f                            -0.073     VIOLATED
      A_reg_reg[9]/D f                          -0.065     VIOLATED
      A_reg_reg[8]/D f                          -0.065     VIOLATED
      A_reg_reg[7]/D f                          -0.065     VIOLATED
      A_reg_reg[6]/D f                          -0.065     VIOLATED
      out_reg[1]/D f                            -0.062     VIOLATED
      out_reg[8]/D f                            -0.052     VIOLATED
      out_reg[9]/D f                            -0.049     VIOLATED
      out_reg[6]/D f                            -0.036     VIOLATED
      out_reg[7]/D f                            -0.027     VIOLATED
      ------------------------------------------------------------------------------------------------
```

No Other Violations

```
Check type : clock_period
---------------------------
No paths found

Check type : skew
---------------------------
No paths found

Check type : pulse_width
---------------------------
 No violating Checks with given description found

Check type : max_transition
---------------------------
 No Violations found

Check type : min_transition
---------------------------
 No Violations found

Check type : max_capacitance
---------------------------
 No Violations found

Check type : min_capacitance
---------------------------
 No Violations found

Check type : max_fanout
---------------------------
 No Violations found

Check type : min_fanout
```

Timing Report

```
#####################################################################
#  Generated by:      Cadence Tempus 20.10-p003_1
#  OS:                Linux x86_64(Host ID edaserver4)
#  Generated on:      Fri Apr  8 01:20:43 2022
#  Design:            rtl_module
#  Command:           report_timing > $report_dir/timing_report.rpt
#####################################################################
Path 1: MET Setup Check with Pin out_reg[0]/CK
Endpoint:   out_reg[0]/D (^) checked with  leading edge of 'clk'
Beginpoint: rst          (v) triggered by  leading edge of 'clk'
Path Groups: {clk}
Other End Arrival Time          0.500
- Setup                         0.145
+ Phase Shift                   4.000
- Uncertainty                   0.500
= Required Time                 3.855
- Arrival Time                  2.242
= Slack Time                    1.612
     Clock Rise Edge                         0.000
     + Input Delay                           1.200
     + Network Insertion Delay               0.500
     = Beginpoint Arrival Time               1.700
      -------------------------------------------------------------
      Instance    Arc          Cell       Delay  Arrival  Required
                                                 Time     Time
      -------------------------------------------------------------
      -           rst v        -          -      1.700    3.312
      g542_0      A v -> Y ^   CLKINVX12  0.103  1.803    3.415
      g2490       A ^ -> Y v   CLKINVX3   0.104  1.908    3.520
      g2489       A v -> Y ^   CLKINVX3   0.124  2.031    3.644
      g2269       A2 ^ -> Y v  AOI32X1    0.146  2.177    3.790
      g2260       B0 v -> Y ^  OAI21X1    0.065  2.242    3.855
      out_reg[0]  D ^          DFFQXL     0.000  2.242    3.855
      -------------------------------------------------------------
```

ANALYSIS

Initially, the netlist was generated while keeping a very tight constraint of Tclk=2.37ns. However, the STA has been performed on it with Tclk=4.0ns. Which relative to 2.37ns is a relaxed constraint for this netlist.

The STA showed us that we have no Setup Violations. All the data successfully reaches the flip flop's input before the setup time of the next clock edge. The hold violations will be eliminated when we do **clock tree synthesis**.

Required Time: The data must reach the next flip flop ‘setup time’ before the next clock edge arrives. Since T_{clk}=4ns and we have an uncertainty of 0.5. In the worst case for setup analysis, the clock can only come after 3.5ns (4-0.5). However, we have a delay of 0.5ns too. This delay and uncertainty cancel, and we are left with 4ns as the time to the next clock edge. The setup time of a flip flop is 0.145ns. Hence, we want our data to be available within 3.855ns (4-0.145) from when the current clock edge has arrived.

Arrival Time: This is the time our data took to reach the input pin of the flip flop. It is estimated as 2.242ns by the tool. We wanted our data to come before 3.855, which came after 2.242ns only. This implies our constraint is met, and we don't have a violation.

Slack: Required Time - Arrival Time= 3.855-2.242= 1.612ns

Critical Path: The following is the critical path according to the tool. That is the path with the worst slack. From rst to out_reg[6], both are triggered at the positive edge of the clock.

```
#####################################################################
Path 1: MET Setup Check with Pin out_reg[0]/CK
Endpoint:   out_reg[0]/D (^) checked with  leading edge of 'clk'
Beginpoint: rst          (v) triggered by  leading edge of 'clk'
Path Groups: {clk}
```

CONCLUSION

The netlist generated in this part was done using the tightest constraints file to have the maximum operating frequency. However, we ran the STA using a relatively relaxed constraint. As a result, it is safe to assume that the slack we are getting, even though is positive, has a higher value in comparison to what it could've been if we had used the tight T_{clk} while performing STA as well.

T_{clk} in the constraints file is greater than the constraints used to generate the netlist. The constraint used in STA was supposed to be in between, such that we don't lose too much area and at the same time have a faster system. There was no chance of having a setup violation since the netlist was generated to perform on very tight constraints, and we gave relaxed constraints during STA.

Since the setup slack is positive, we do not have any setup violations in the netlist generated using T_{clk}=2.37ns, and STA was performed using T_{clk}=4ns.

NETLIST - C

TCL FILE

```
file mkdir sta_after_synthesis_3c/reports^M
set report_dir sta_after_synthesis_3c/reports^M

read_lib ../lib/90/lib/slow_wlm.lib^M
read_verilog syn_report_3c/synthesised_netlist.v^M
set_top_module rtl_module^M
read_sdc ../constraints/constraints_project_3c.sdc^M

#Wireload Model
set_wire_load_model -name "wlm"
set_wire_load_mode segmented

check_timing > $report_dir/check_timing.rpt^M
report_timing > $report_dir/timing_report.rpt^M
report_analysis_coverage > $report_dir/analysis_coverage.rpt^M
report_analysis_summary > $report_dir/analysis_summary.rpt^M

#report_annotated_parasitics > $report_dir/annotated.rpt^M

report_clocks > $report_dir/clocks.rpt^M
report_case_analysis > $report_dir/case_analysis.rpt^M
report_constraints -all_violators > $report_dir/allviolations.rpt

#wireload report
report_wire_load > $report_dir/wireload.rpt

report_timing -retime path_slew_propagation -max_path 30 -nworst 30 -path_type full_clock > $report_dir/pba.rpt
report_timing -late -max_paths 30 -nworst 30 -path_type full_clock -net > $report_dir/gba_setup.rpt
report_timing -early -max_paths 30 -nworst 30 -path_type full_clock -net > $report_dir/gba_hold.rpt

report_timing -max_paths 30 -nworst 30 -retime path_slew_propagation > $report_dir/pba_setup.rpt
report_timing -early -max_path 30 -nworst 30 -retime path_slew_propagation > $report_dir/pba_hold.rpt^M
#exit
~
```

Analysis Coverage Report

```
#################################################################
#  Generated by:      Cadence Tempus 20.10-p003_1
#  OS:                Linux x86_64(Host ID edaserver4)
#  Generated on:      Thu Apr  7 23:20:02 2022
#  Design:            rtl_module
#  Command:           report_analysis_coverage > $report_dir/analysis_coverage.rpt
#################################################################
    ----------------------------------------------------------------------------------------------
                                TIMING CHECK COVERAGE SUMMARY
    ----------------------------------------------------------------------------------------------
     Check Type                    No. of    Met             Violated         Untested
                                   Checks
    ----------------------------------------------------------------------------------------------
    ExternalDelay (Early)          11        11 (100%)       0 (0%)           0 (0%)
    ExternalDelay (Late)           11        11 (100%)       0 (0%)           0 (0%)
    Hold                           74        32 (43%)        42 (56%)         0 (0%)
    PulseWidth                     86        86 (100%)       0 (0%)           0 (0%)
    Setup                          74        74 (100%)       0 (0%)           0 (0%)
    ----------------------------------------------------------------------------------------------
```

All Violations Report

No Setup Violations

```
#############################################################
#  Generated by:      Cadence Tempus 20.10-p003_1
#  OS:                Linux x86_64(Host ID edaserver4)
#  Generated on:      Thu Apr  7 23:20:02 2022
#  Design:            rtl_module
#  Command:           report_constraints -all_violators > sta_after_synthesis_3c/reports/allviolations.rpt
#############################################################
# format : frame 0 : split 1

max_delay/setup
---------------------------
No paths found
```

Hold Violations Exist

```
min_delay/hold
---------------------------
     ------------------------------------------------------------------------------------------------
      End Point                                  Slack      Cause
     ------------------------------------------------------------------------------------------------
     in1/count_reg_reg[1]/D f                  -0.269     VIOLATED
     in1/count_reg_reg[0]/D f                  -0.264     VIOLATED
     in1/clk_out_reg/D f                       -0.184     VIOLATED
     in1/clk_out_reg/SE f                      -0.176     VIOLATED
     in1/clk_out_reg/SI f                      -0.161     VIOLATED
     out_reg[0]/D f                            -0.083     VIOLATED
     C_reg_reg[8]/D f                          -0.065     VIOLATED
     C_reg_reg[7]/D f                          -0.065     VIOLATED
     C_reg_reg[6]/D f                          -0.065     VIOLATED
     C_reg_reg[5]/D f                          -0.065     VIOLATED
     C_reg_reg[4]/D f                          -0.065     VIOLATED
     C_reg_reg[3]/D f                          -0.065     VIOLATED
     C_reg_reg[2]/D f                          -0.065     VIOLATED
     C_reg_reg[1]/D f                          -0.065     VIOLATED
     C_reg_reg[0]/D f                          -0.065     VIOLATED
     B_reg_reg[9]/D f                          -0.065     VIOLATED
     B_reg_reg[8]/D f                          -0.065     VIOLATED
     B_reg_reg[7]/D f                          -0.065     VIOLATED
     B_reg_reg[6]/D f                          -0.065     VIOLATED
     B_reg_reg[5]/D f                          -0.065     VIOLATED
     B_reg_reg[4]/D f                          -0.065     VIOLATED
     B_reg_reg[3]/D f                          -0.065     VIOLATED
     B_reg_reg[2]/D f                          -0.065     VIOLATED
     B_reg_reg[1]/D f                          -0.065     VIOLATED
     B_reg_reg[0]/D f                          -0.065     VIOLATED
     A_reg_reg[9]/D f                          -0.065     VIOLATED
     A_reg_reg[8]/D f                          -0.065     VIOLATED
```

```
    A_reg_reg[7]/D f                          -0.065     VIOLATED
    A_reg_reg[6]/D f                          -0.065     VIOLATED
    A_reg_reg[5]/D f                          -0.065     VIOLATED
    A_reg_reg[4]/D f                          -0.065     VIOLATED
    A_reg_reg[3]/D f                          -0.065     VIOLATED
    A_reg_reg[2]/D f                          -0.065     VIOLATED
    A_reg_reg[1]/D f                          -0.065     VIOLATED
    A_reg_reg[0]/D f                          -0.065     VIOLATED
    out_reg[8]/D f                            -0.058     VIOLATED
    out_reg[6]/D f                            -0.058     VIOLATED
    out_reg[9]/D f                            -0.054     VIOLATED
    out_reg[10]/D f                           -0.043     VIOLATED
    out_reg[7]/D f                            -0.036     VIOLATED
    out_reg[5]/D f                            -0.036     VIOLATED
    out_reg[4]/D f                            -0.022     VIOLATED
    ------------------------------------------------------------------------------------------------
```

No Other Violations

```
Check type : clock_period
---------------------------
No paths found

Check type : skew
---------------------------
No paths found

Check type : pulse_width
---------------------------
 No violating Checks with given description found

Check type : max_transition
---------------------------
 No Violations found

Check type : min_transition
---------------------------
 No Violations found

Check type : max_capacitance
---------------------------
 No Violations found

Check type : min_capacitance
---------------------------
 No Violations found

Check type : max_fanout
---------------------------
 No Violations found

Check type : min_fanout
```

Timing Report

```
#  Generated by:      Cadence Tempus 20.10-p003_1
#  OS:                Linux x86_64(Host ID edaserver4)
#  Generated on:      Thu Apr  7 23:20:02 2022
#  Design:            rtl_module
#  Command:           report_timing > $report_dir/timing_report.rpt
#############################################################
Path 1: MET Setup Check with Pin out_reg[2]/CK
Endpoint:   out_reg[2]/D (^) checked with  leading edge of 'clk'
Beginpoint: rst          (v) triggered by  leading edge of 'clk'
Path Groups: {clk}
Other End Arrival Time          0.500
- Setup                         0.168
+ Phase Shift                   4.000
- Uncertainty                   0.500
= Required Time                 3.832
- Arrival Time                  3.650
= Slack Time                    0.182
     Clock Rise Edge                      0.000
     + Input Delay                        1.200
     + Network Insertion Delay            0.500
     = Beginpoint Arrival Time            1.700
      ------------------------------------------------------------
      Instance    Arc          Cell      Delay  Arrival  Required
                                                Time     Time
      ------------------------------------------------------------
      -           rst v        -         -      1.700    1.882
      g528        A v -> Y ^   CLKINVX1  0.800  2.500    2.681
      g2617       D ^ -> Y v   NAND4XL   0.813  3.313    3.495
      g2610       B1 v -> Y ^  OAI222XL  0.337  3.650    3.832
      out_reg[2]  D ^          DFFQXL    0.000  3.650    3.832
```

ANALYSIS

The netlist was generated using Tclk=4.0ns, and the STA was performed using the same constraints as well.

The STA showed us that we have no Setup Violations. All the data successfully reaches the flip flop's input before the setup time of the next clock edge. The hold violations will be eliminated when we do **clock tree synthesis**.

Required Time: The data must reach the next flip flop ‘setup time’ before the next clock edge arrives. Since T_{clk}=4ns and we have an uncertainty of 0.5. In the worst case for setup analysis, the clock can only come after 3.5ns (4-0.5). However, we have a delay of 0.5ns too. This delay and uncertainty cancel, and we are left with 4ns as the time to the next clock edge. The setup time of a flip flop is 0.168ns. Hence, we want our data to be available within 3.832ns (4-0.168) from when the current clock edge has arrived.

Arrival Time: This is the time our data took to reach the input pin of the flip flop. It is estimated as 3.65ns by the tool. We wanted our data to come before 3.832ns, but it came after 3.65ns only. This implies our constraint is met, and we don't have a violation.

Slack: Required Time - Arrival Time= 3.832-3.65= 0.182ns

Critical Path: The following is the critical path according to the tool. That is the path with the worst slack. From rst to out_reg[2], both are triggered at the positive edge of the clock.

```
Path 1: MET Setup Check with Pin out_reg[2]/CK
Endpoint:   out_reg[2]/D (^) checked with  leading edge of 'clk'
Beginpoint: rst          (v) triggered by  leading edge of 'clk'
Path Groups: {clk}
```

CONCLUSION

The netlist generated in this part was done using the middle constraints file to have the trade-off between area and speed be balanced, and we ran the STA using the same constraints as well.

We got a positive value for the setup slack, which means that we have no setup violations in our design which uses constraints to have a balance of speed and area.

Since the setup slack is positive, we do not have any setup violations in the netlist generated using T_{clk}=4ns, and STA was performed using T_{clk}=4ns.

PART 6
Test Insertion

Aim: To insert a scan chain and synthesise a new netlist with this scan chain in place to enable test mode.

Tool: Cadence Genus

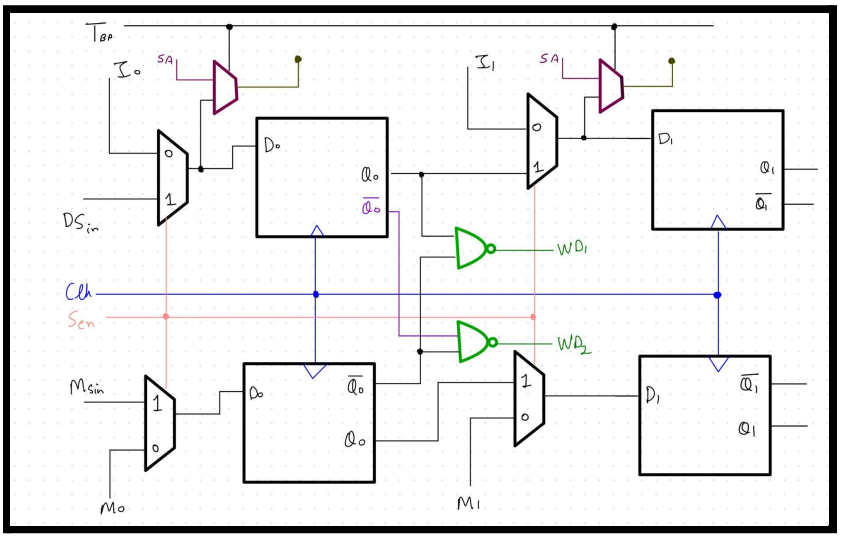
Theory

Scan chain is used for testing designs. It enables us to break a complicated combinational logic into simpler and smaller blocks, and hence it gets easier to test inputs on these smaller logic blocks.

As the design starts to get complex, the ability to provide unique test patterns and observe the outputs for verification would get complex. This problem of controllability and observability is further elevated when we switch to sequential logic.
For ‘n’ states in a sequential circuit, we have 2^n possible states, a massive space complexity order.

In this design, extra inputs are added to the design, such as the test mode TM, Scan enable (SE), Scan In (SI) and Scan out (SO). All the memory elements are replaced by the scan cells. Scan cells are connected to form the shift registers.

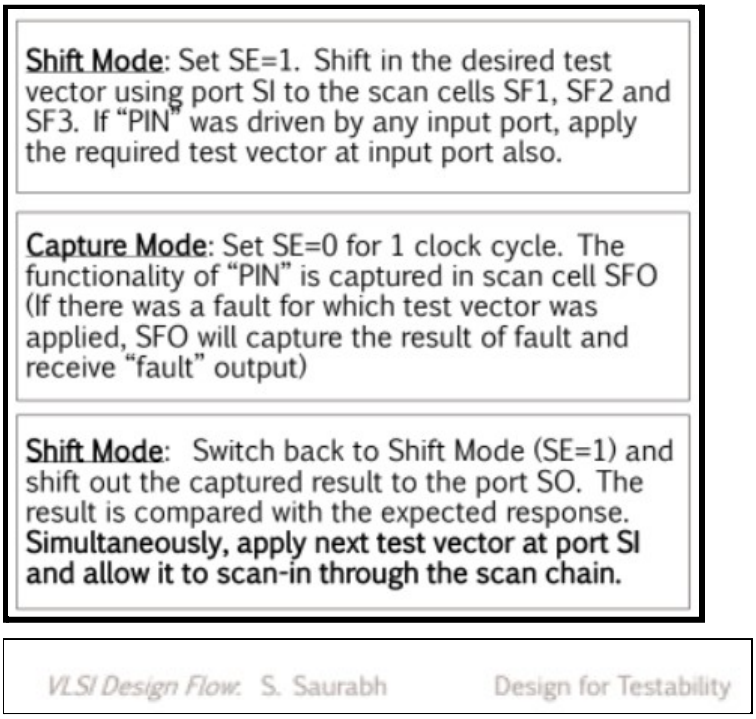
The following is an example of a Scan chain buffer on the input of a memory. The two outputs {wd1, wd2} are the signals that go to the write driver to control the voltage of the bit-lines.



When the S_en signal is raised to ‘1’, we enter the scan mode, and our data is passed on serially at every clock cycle. After all the flip flops are loaded, the relevant Wr_en signal is passed to the write driver. This signal must come at such a time when all the columns of the row have been loaded with the appropriate bit-line level. As we can see, if the S_en signal is 1, we pass the data of the previous flip flop into the next flip, and thus we make our system a shift register. At every clock cycle, a computation is performed between the ‘Mask Bit’ and the ‘Scan in’ bit, which is the input to the write driver.

If S_en is 0, then the data is loaded parallelly in all the flip flops simultaneously. This requires a bus width of 32 bits, and thus the required pitch (hence the area) would increase. But in this mode, we can compute all 32 inputs to the write driver in 1 clock cycle alone (considering the delay of combinational logic is negligible).

Modes of Operation



About the Tool

This is the tool that is used for the synthesis of the netlist. We provide an RTL file, constraints file, library and a TCL script to this software. This software then generates the appropriate netlist. This tool was also used in step 3 of the flow; however, for this part, we have made slight changes in the ‘tcl script’ to get the scan chain for testability.

TCL Commands

- **set_attribute dft_scan_style muxed_scan:** Scan style selection.
- **define_dft shift_enable -active high -create_port scan_en**: Active high scan enable is created via this command.
- **define_dft test_mode -active high -create_port test_mode:** Active high signal used to change the mode to the test mode.
- **define_dft test_clock clk:** Used to apply the clock to the design.
- **report dft_setup:** Used to report the set-up in the scan chain.
- **check_dft_rules >dft_report/dft_rules_report:** Used to verify the scan chain insertion. The DFT rules are checked, and a report is generated. Checks clock and asynchronous set/reset violations.
- **fix_dft_violations -test_control test_mode -async_set -async_reset -clock:** Used to fix any violation in the clock or the asynchronous set/reset.
- **synthesise -to_mapped:** RTL is converted to netlist by certain optimisations.
- **set_attr dft_min_number_of_scan_chains 2 top:** Used to set the minimum number of the scan chains.
- **connect_scan_chains -auto_create_chains -preview:** Used to connect the scan chains as preview.
- **connect_scan_chains -auto_create_chains:** Automatically connect the scan chain.

TCL Script

```
set_attr lib_search_path ../lib/90/lib
set_attr hdl_search_path ../rtl/testbench1
set_attr library slow_wlm.lib

read_hdl top.v

elaborate
read_sdc ../constraints/constraints_project_3c.sdc
report timing -lint
set_attribute dft_scan_style muxed_scan
define_dft shift_enable -active high -create_port scan_en
define_dft test_mode -active high -create_port test_mode
define_dft test_clock clk
report dft_setup
check_dft_rules >dft_report/dft_rules_report
fix_dft_violations -test_control test_mode -async_set -async_reset -clock
synthesize -to_mapped
set_attr dft_min_number_of_scan_chains 2 rtl_module
set_attr dft_mix_clock_edges_in_scan_chains true rtl_module
#replace_scan
connect_scan_chains -auto_create_chains -preview
connect_scan_chains -auto_create_chains
report qor
write_atpg -cadence > rtl_module.atpg

write_atpg -stil > rtl_module_still.atpg
write_scandef> dft_report/rtl_module.def
write_sdf -timescale ns -nonegchecks -recrem split -edges check_edge > dft_report/syn_report/delays.sdf
write_hdl > dft_report/synthesised_netlist.v
write_sdc > dft_report/sdc_file_for_physical_design.sdc
write_script > dft_report/synthesis_script_sdc.g
report timing > dft_report/synthesis_timing_report.rep
report power > dft_report/synthesis_power_report.rep
report gates > dft_report/synthesis_cell_report.rep
report area > dft_report/synthesis_area_report.rep
```

Area Report

```
  Generated by:           Genus(TM) Synthesis Solution 19.13-s073_1
  Generated on:           Apr 07 2022  11:27:35 pm
  Module:                 rtl_module
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
================================================================

 Instance          Module          Cell Count  Cell Area  Net Area   Total Area  Wireload
-------------------------------------------------------------------------------------------
rtl_module                             201   1617.495     0.000     1617.495    <none> (D)
  in1      clk_divider_div_value3        8     82.502     0.000       82.502    <none> (D)

 (D) = wireload is default in technology library
```

Timing Report

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 19.13-s073_1
  Generated on:           Apr 07 2022  11:27:35 pm
  Module:                 rtl_module
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================
            Pin                        Type       Fanout Load Slew Delay Arrival
    |                                                 (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------------------
(clock clk)                          launch                                  0 R
                                     latency                          +500     500 R
(constraints_project__line_7)        ext delay                        +1200    1700 F
rst                                  in port           2  4.4 1200     +0    1700 F
g538/A                                                                  +0    1700
g538/Y                               CLKINVX1         32 57.2  632   +751    2451 R
g2617/D                                                                 +0    2451
g2617/Y                              NAND4XL          11 18.2  933   +760    3211 F
g2610/B1                                                                +0    3211
g2610/Y                              OAI222XL          1  1.8  386   +328    3538 R
out_reg[2]/D                  <<<    SDFFQX1                            +0    3538
out_reg[2]/CK                        setup                        500  +260    3798 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                          capture                                 4000 R
                                     latency                          +500    4500 R
                                     uncertainty                      -500    4000 R
-------------------------------------------------------------------------------
Cost Group   : 'clk' (path_group 'clk')
Timing slack :     202ps
Start-point  : rst
End-point    : out_reg[2]/D
```

Power Report

```
Instance: /rtl_module
Power Unit: W
PDB Frames: /stim#0/frame#0
  ------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total    Row%
  ------------------------------------------------------------------------
       memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
     register     6.03202e-06  2.29126e-04  6.52775e-06  2.41686e-04  74.02%
        latch     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        logic     3.13918e-06  4.92062e-05  1.77386e-05  7.00840e-05  21.46%
         bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        clock     0.00000e+00  0.00000e+00  1.47420e-05  1.47420e-05   4.51%
          pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
           pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  ------------------------------------------------------------------------
    Subtotal     9.17120e-06  2.78332e-04  3.90084e-05  3.26512e-04  99.99%
  Percentage           2.81%       85.24%       11.95%      100.00% 100.00%
  ------------------------------------------------------------------------
```

Cell Report

```
==============================================================
  Generated by:           Genus(TM) Synthesis Solution 19.13-s073_1
  Generated on:           Apr 07 2022  11:27:35 pm
  Module:                 rtl_module
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
==============================================================
```

Gate	Instances	Area	Library
AND2X1	1	4.541	slow
AND2XL	24	108.994	slow
AOI211X1	1	5.298	slow
AOI21X1	7	31.790	slow
AOI21XL	6	27.248	slow
AOI22X1	3	18.166	slow
AOI32X1	1	6.812	slow
CLKINVX1	4	9.083	slow
INVX1	6	13.624	slow
INVXL	1	2.271	slow
MXI2XL	3	18.166	slow
NAND2BX1	10	45.414	slow
NAND2XL	11	33.304	slow
NAND3X1	1	4.541	slow
NAND4XL	4	21.193	slow
NOR2BX1	8	36.331	slow
NOR2XL	28	84.773	slow
NOR3X1	1	4.541	slow
OAI211X1	3	15.895	slow
OAI21X1	11	49.955	slow
OAI21XL	1	4.541	slow
OAI222XL	1	8.326	slow
OAI22X1	3	18.166	slow
OAI22XL	7	42.386	slow
OAI2BB1X1	2	10.597	slow
OR2XL	3	13.624	slow
SDFFQX1	36	735.707	slow
SDFFQXL	1	20.436	slow
SDFFTRX1	6	163.490	slow
XNOR2X1	3	24.978	slow
XNOR2XL	3	24.978	slow

```
----------------------------------------
total             201  1617.495
```

Type	Instances	Area	Area %
sequential	43	919.633	56.9
inverter	11	24.978	1.5
logic	147	672.884	41.6
physical_cells	0	0.000	0.0
total	201	1617.495	100.0

ANALYSIS

Netlist

```
module clk_divider_div_value3(clk_in, clk_out, DFT_sdi, DFT_sen,
     DFT_sdo);
  input clk_in, DFT_sdi, DFT_sen;
  output clk_out, DFT_sdo;
  wire clk_in, DFT_sdi, DFT_sen;
  wire clk_out, DFT_sdo;
  wire [1:0] count_reg;
  wire n_0, n_1, n_5, n_9, n_11;
  SDFFQX1 clk_out_reg(.CK (clk_in), .D (n_9), .SI (DFT_sdi), .SE
       (DFT_sen), .Q (clk_out));
  SDFFQX1 \count_reg_reg[1] (.CK (clk_in), .D (n_11), .SI
       (count_reg[0]), .SE (DFT_sen), .Q (DFT_sdo));
  SDFFQX1 \count_reg_reg[0] (.CK (clk_in), .D (n_5), .SI (clk_out), .SE
       (DFT_sen), .Q (count_reg[0]));
  NOR2XL g62(.A (count_reg[0]), .B (n_0), .Y (n_5));
  NOR2XL g67(.A (count_reg[0]), .B (n_1), .Y (n_0));
  CLKINVX1 g69(.A (DFT_sdo), .Y (n_1));
  XOR2XL g2(.A (clk_out), .B (n_0), .Y (n_9));
  AND2XL g72(.A (count_reg[0]), .B (n_1), .Y (n_11));
endmodule
```

The following generated netlist has extra input ports as the scan chain works in different modes. So, we need a MuX to select the mode of operation and this causes extra input ports to be considered in the top module generated via DFT.

Comparison Table

	WITHOUT DFT	DFT
AREA →	1395.724	1617.495
CELLS →	171	201
TIMING SLACK →	320 ps	202 ps
POWER →	0.293510 mW	0.326512 mW

When we introduced DFT: -

- *Area Increased*
- *Number of Cells Increased*
- *Timing slack Reduced*
- *Power consumption increased*

CONCLUSION (QoR)

Area and Number of Cells

Since, we have increased the total number of inputs on our system and we are using scan flip flops, this has caused an increase in the area as well as the number of cells after we have synthesized the netlist for DFT. We are now using MuXed cells (as explained in the theory part) and this is resulting in an increased area.

Power Consumption

Since, we are increasing the total number of cells in our system, this means we increase the leakage that can occur through these devices and through the routing of these devices. The switching power has increased as well, now we have MuXed cells, we have more possibilities of flipping. We have more input signals as well. This would cause an increase in the activity factor and as a result the switching power has increased too.

Timing Slack

Since, we have increased the load on the devices the slew and delays in the system have increased post dft. This results in arrival time increasing and as a result the slack of the system reduced. It is still positive and hence we do not have any setup violations, but the slack has reduced in comparison to synthesis without dft due to the less load on the devices.