



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

Scan insertion with DRC and coverage analysis

The domain of the Project:

DFT(Design for testability)

Team Mentors (and their designation):

Team Members:

1. Mr. P. Vijay Kumar B.Tech, 4th year pursuing ---- Team member
2. Mr. K. Tharun Krishna B.Tech, 4th year pursuing ---- Team member
3. Mr. P. Pavan kumar B.Tech, 4th year pursuing ---- Team member
4. Mr. K. Chandu B.Tech 4th year pursuing --- Team member

Period of the project

May 2025 to July 2025



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

Declaration

The project titled "**Scan insertion with DRC and coverage analysis**" has been mentored by **Ryan Ebenezer**, organised by SURE Trust, from May 2025 to July 2025, for the benefit of the educated unemployed rural youth for gaining hands-on experience in working on industry relevant projects that would take them closer to the prospective employer. I declare that to the best of my knowledge the members of the team mentioned below, have worked on it successfully and enhanced their practical knowledge in the domain.

Team Members:

1. Mr. P. Vijay Kumar
2. Mr. K. Tharun Krishna
3. Mr. P. Pavan kumar
4. Mr. K. Chandu

Mr.Ryan Ebenezer

DFT engineer—Struent semiconductors pvt ltd

Prof. Radhakumari

Executive Director & Founder
SURE Trust



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

Table of contents

1. Detailed report
2. Executive summary
3. Introduction
4. Project Objectives
5. Methodology & Results
6. Social / Industry relevance of the project
7. Learning & Reflection
8. Future Scope & Conclusion



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

Detailed Report:

DATE	MINUTES OF MEETING	ATTENDEES
26/05/25	INTRODUCTION TO PROJECT WORK	P. Vijay Kumar K. Tharun Krishna P. Pavan kumar K. Chandu
28/05/25	DRC INTRODUCTION	P. Vijay Kumar K. Tharun Krishna P. Pavan kumar K. Chandu
03/06/25	INTRODUCTION TO CLOCK TERMINOLOGY	P. Vijay Kumar K. Tharun Krishna P. Pavan kumar K. Chandu
07/06/25	CLOCK RULES EXPLANATION	P. Vijay Kumar K. Tharun Krishna P. Pavan kumar K. Chandu
16/06/25	PRACTICAL DESIGN IMPLEMENTATION	P. Vijay Kumar K. Tharun Krishna P. Pavan kumar K. Chandu
28/06/25	MODIFICATION OF DESIGN AND ADDING SOME BLACK BOX	P. Vijay Kumar K. Tharun Krishna P. Pavan kumar



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

		K. Chandu
03/07/25	CORRECTION OF EERORS AND ADDING PLL AND SOME OTHER LOGIC	P. Vijay Kumar K. Tharun Krishna P. Pavan kumar K. Chandu
06/07/25	SOLVING THE DRC'S FORMED IN DESIGN	P. Vijay Kumar K. Tharun Krishna P. Pavan kumar K. Chandu
09/07/25	FIXING OF E5 VIOLATION AND SCAN INSERTION	P. Vijay Kumar K. Tharun Krishna P. Pavan kumar K. Chandu



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)
DFT PROJECT REPORT

Minutes of the Meeting

Date : 26/05/2025(Monday)

Time : 8:30 pm

Venue : Google Meet

Agenda

- Introduction to project work (steps to follow till end of project)
- Discussed how to create a project report(including minutes of meeting, theoretical knowledge, hands on work if done).

Task assigned:

- 1)To create a word document
- 2)To recall all the concepts discussed earlier

Theoretical knowledge: The chip manufacturing process is prone to defects and the defects are commonly referred as faults. Design for testability (DFT) refers to those design techniques that make the task of testing the fault sites. A fault is testable if there exists a well-specified procedure to expose it in the actual silicon.

We have different types of faults like stuck at fault (defects which occurs when a circuit node is shorted to VDD (stuck-at-1 fault) or GND (stuck-at-0 fault) permanently). The fault can be at the input or output of a gate. Thus a simple 2-input AND gate has six possible stuck-at faults), transition faults(each port is tested for logic 0-to-1 transition delay or logic 1-to-0 transition delay). One important concept in dft is controlability and observability. We check if the inputs to fault site are controlable or not and we check if the output from fault site is observable or not. We use scan insertion to check the fault sites

1. Converting Regular Flop to Scan Flop.
2. Stitching the Scan Flops to form Scan Chains

During testing flops enter to scan mode by making scan enable to 1. We do testing by shift operation and capture operation. ATPG tool will develope some test patterns to check the fault site. We have different techniques like Mbist, Lbist, Scan chain etc to test the chip.



Minutes of the Meeting

Date : 28/05/2025(Wednesday)

Time : 9:00 pm

Venue : Google Meet

Agenda

- DRC introduction(like DRC analysis,fixing)
- Brief introduction to types of check rules(A,C,D,E etc)

Task assigned:

- 1)Design Rule overview
- 2)Scan insertion checking
- 3)How to troubleshoot Rules violations
- 4)Clock Rules,Clock terminology

Theoretical knowledge: Before scan insertion, the tool performs initial scannability checks when switching from setup mode to analysis mode.

- For gated clocks (primary clock inputs controlled by additional logic), a test procedure file is required to specify the conditions under which the clock can propagate through these gates.
- In the case of uncontrollable clock circuitry, the tool offers test logic insertion to make these nodes controllable during testing.
- If existing scan logic is present or a test procedure file specifies scan eligibility, the tool conducts deeper validation
- Once scan logic is inserted and the test procedure file is created, the user should return to **setup mode** to register this new data and then switch again to **analysis mode** to perform **full rule checks** before generating scan patterns.

Clock Signals

The tool considers any signal to be a clock if it can change the state of a sequential element. The transition of the clock from the off state to the on state is considered the **leading edge** of the clock, while the transition from the on state to the off state is considered the **trailing edge** of the clock.

Clock Cone - Region of logic influenced by a clock through combinational logic and TLAs



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

The clock cone is basically the fanout of the clock signal through strictly combinational logic and TLAs.

Effect Cone - Region affected by the output of a gate or flip-flop (influence zone)

The tools consider a gate pin or output pin to be in a clock's effect cone if there is a sequential element between the clock net and the gate pin or output pin.

clock source (synchronous or asynchronous) - A clock source is where a clock signal originates in the design.

derived clock source - is a defined instance pin where a reference pin has to be specified because a tracing path. A **derived clock source** is an internal instance pin where the tool **cannot trace back** to the original clock source

derived clock branch - A derived clock branch occurs when different branches of a clock tree (even from the same source) are unbalanced or asynchronous.

Minutes of the Meeting

Date : 03/06/2025(Tuesday)

Time : 9:30 pm

Venue : Google Meet

Agenda

- Introduction to Clock terminology
- How a tool will classify clock signals itself.

Task assigned:

1)To update word document

2)Overview of clock rules

- C1 (clock Controllability Violation)
- C2 (Missing Clock Definition Violation)
- C3 (Source-Sink Same Clock Violation (Transparency Check))
- C6 (Clock Capturing Its Own Effect (Race Condition Detection))

Theoretical Knowledge:

C1 :

Category: Clock



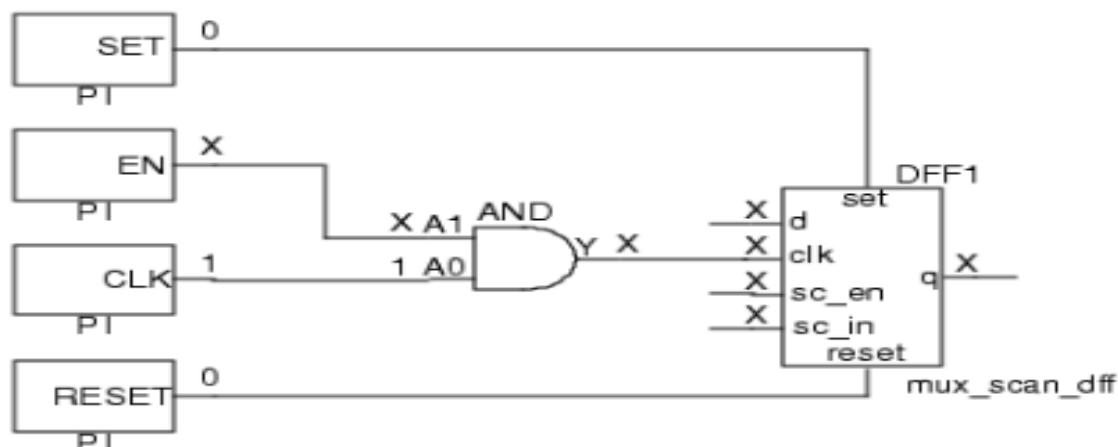
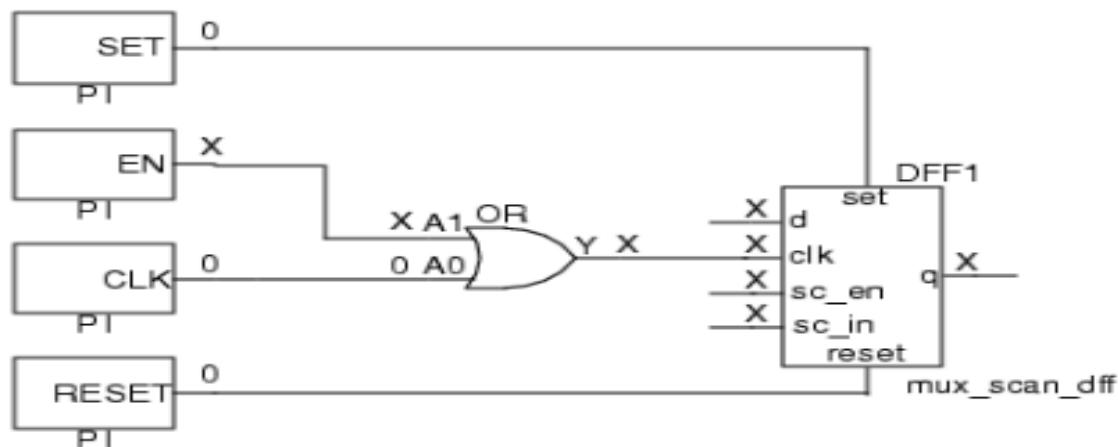
Default Handling: Error

- A scan or non-scan cell must not capture data when all specified clocks are set to their off states.
- One common root cause of these C1 violations is a **set/reset pin not defined as a clock**. You can define a set/reset pin as a clock to fix the problem and avoid ATPG performance impact.

Effect on Testability:

Failure to satisfy this rule can result in unstable scan and non-scan cell values going into or coming out of load_unload, or between cycles. This can result in lower test coverage.

Example:



The clock input value to the scan cell is X because the EN signal value is X. If EN is left at X, the signal arriving at the clk input of the scan cell cannot be held off.

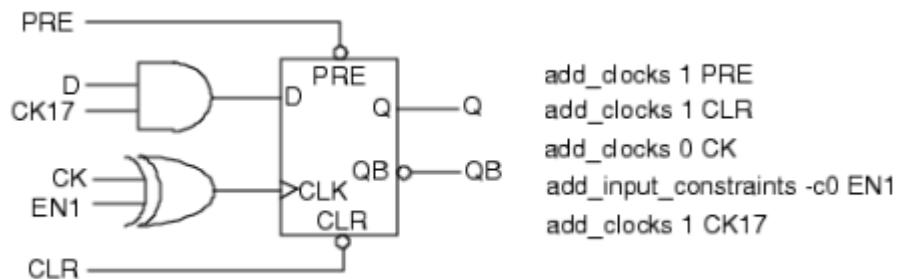


Default Handling: Warning

Each clock must have a structural path to the clock port of at least one memory element. The application performs this check by a backward tracing from the clock port, SET/RESET port, and read/write enables of all memory elements toward primary inputs.

This means that if a clock signal is declared it **must physically drive** a memory element like:

- Flip-flops
- Latches
- RAMs/ROMs (via enable, clk, etc.)
- **The rule violation occurs if a clock primary input cannot be reached.**
- Defining a pin to be a clock, when it does not behave as a clock, is the most usual cause of this error condition.
- Failure to satisfy this rule indicates a defined clock cannot capture data, thus reducing test coverage.



you get a C2 rules violation because while the CK17 signal appears to be a clock (due to its name), it cannot be reached from the clock port or SET/RESET ports of the flip-flop.

C3

Default Handling: Note

During scan or ATPG, we rely on predictable timing — data is launched from a source flip-flop, and then captured by a sink flip-flop on the next clock edge.

If both FFs are clocked at the **same time**, and there's **no logic or register delay** between them, the sink might accidentally capture the **new data** (rather than the old/stable data), breaking the test timing model.

This introduces ambiguity:



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

- Was the data captured from the previous or current cycle?
- It could cause test pattern invalidation or fault masking.

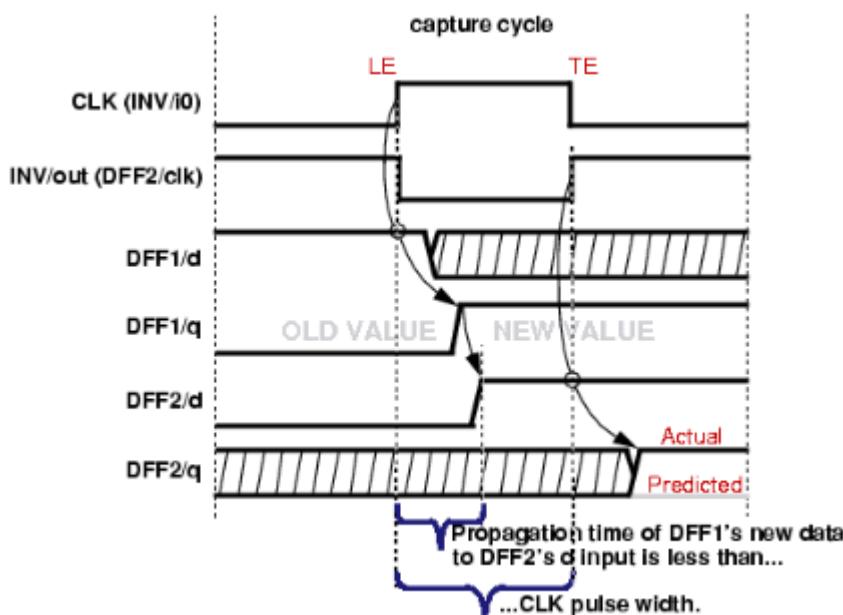
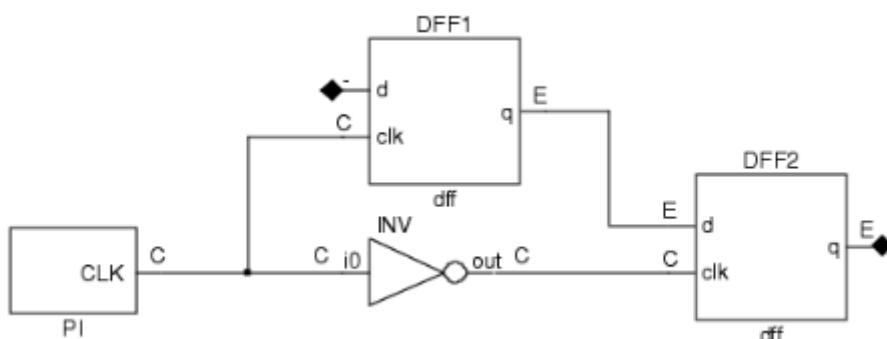
This is informational, but may need fixing if it affects ATPG coverage or pattern validity.

Fix Strategy	Description
Insert intermediate flip-flop or logic	Add pipeline stage or logic to delay data
Change clock phase	Clock source and sink on different edges (e.g., source on rising edge, sink on falling edge)
Accept with justification	If timing is met and patterns are safe, you can waive the note in tool constraints

A violation occurs when:

- The **clock input of a memory element is in the clock cone, and**
- Its data input is in the effect cone of the same clock.**

This implies a **data/clock dependency loop**, which may create race conditions.





Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

1. Flip-Flop Case:

- DFF1 is **leading-edge triggered** (rising edge).
- DFF2 is **trailing-edge triggered** (falling edge).
- DFF2.D (data input) is fed by logic depending on DFF1.Q, and both are driven by the **same clock**.

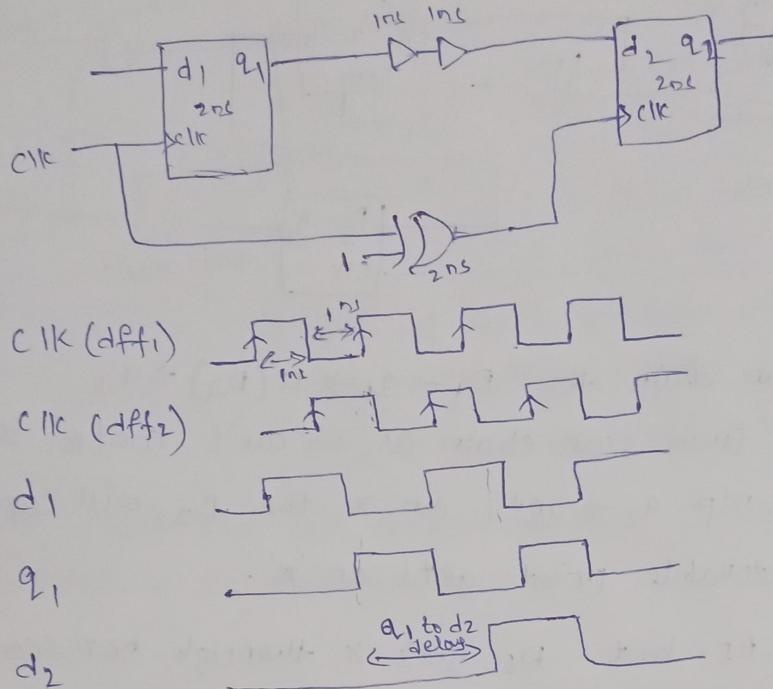
Problem: DFF2 could **capture unstable or new data** immediately after it was clocked into DFF1.

Backward-traces the clock net to determine which registers/memories are in its clock cone.

1. **Forward-traces outputs of memory elements** triggered by that clock to determine the **effect cone**.
2. Flags a violation if:
 - A data input of a flip-flop/latch/RAM is in the **effect cone**, and
 - Its clock or write enable is in the **clock cone**.



C₃ violation:-



q_1 to d_2 delay is greater than clock + XOR delay for second flip flop then capture will perform in another clock cycle which retain old data and no violation occurs.

If q_1 to d_2 delay is less than clock + XOR delay for second flip flop then capture will done in same clock cycle which will have a race condition (like some setup violations). then C₃ rule will get violated.

C6

Category: Clock

Default Handling: Warning

A clock must not affect data that it is capturing. If it does, a race condition may result that produces inaccurate simulation results.

A violation is reported if:



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

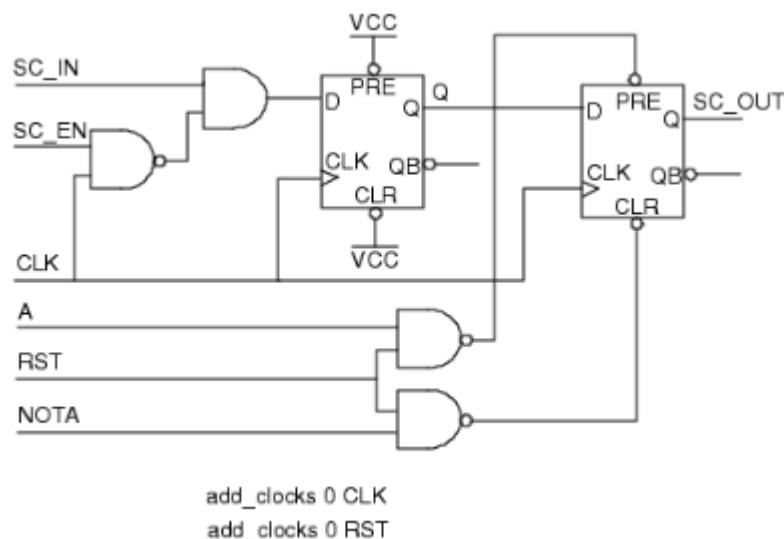
- A scannable flip-flop or latch has its clock and data inputs both located in the clock cone.
- This means the clock may influence both the control (CLK) and the data (D) — potentially in the same cycle

It can result in:

- Race conditions during scan testing.
- Unpredictable logic behavior in simulation.
- Incorrect ATPG patterns or false fault coverage.

The tool **traces forward** from a clock input to find its **clock cone**.

- If the **data input of a flip-flop/latch** is also affected by the same cone (i.e., influenced by gates toggled by the clock), a violation is flagged.

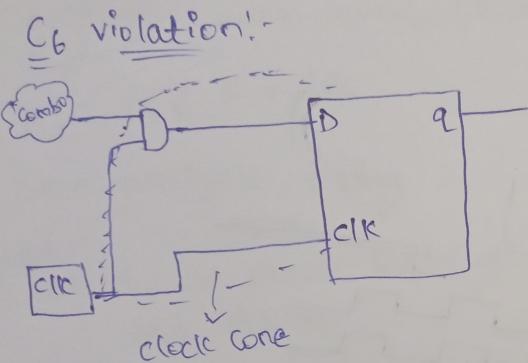


In this design, the CLK signal goes to both the CLK and D inputs of the first flip-flop. Thus, data can be captured in this flip-flop that may be affected by the capturing clock.

Solution Type	Description
Insert Register	Add an intermediate flip-flop to pipeline the data path.



Solution Type	Description
Add Buffer	Insert a non-clocked buffer or logic stage to break direct cone dependency.
Clock Domain Split	If data path can be split into different clock groups.



Here the d pin and clk pin both are influenced by clk. So if clk changes from 0 to 1 then data will get affected at the same time ff will capture the data get data which is earlier also the happens during capture and we predictable o/p.

→ The clock cone will consist both D and clk fine, so this creates a CG violation.

Minutes of the Meeting

Date : 07/06/2025(Tuesday)

Time : 6:00 pm

Venue : Google Meet

Agenda

- Explanation about C1 and C2 violations with their respective examples.
 - How tool can classify C1 and C2 violations and what are steps to prevent the C1 and C2 violations.



Task assigned:

1)To study about D5,E5,T3 DRC's.

2)To create individual circuits to C3,C6,D5,E5 DRC's and complete analysis how the violation occurred and explanation about it.

3)To update the circuits in the report.

Theoretical Knowledge:

D5:

Category: Data

Default Handling: Warning

The check is performed after scan elements have been identified. Any memory element **not connected to a scan chain** (i.e., not converted to a scan flip-flop) will be flagged. The tool will classify each non-scan memory element as:

Classification	Description
INIT-0	Initially at logic 0 in first capture cycle
INIT-1	Initially at logic 1 in first capture cycle
INIT-X	Unknown state at the beginning and can change during capture
TIE-0	Always held at logic 0 during capture
TIE-1	Always held at logic 1 during capture
TIE-X	Always at unknown state during capture

The default handling for this rule violation is warning. Failure to satisfy this rule results in some loss of test coverage.

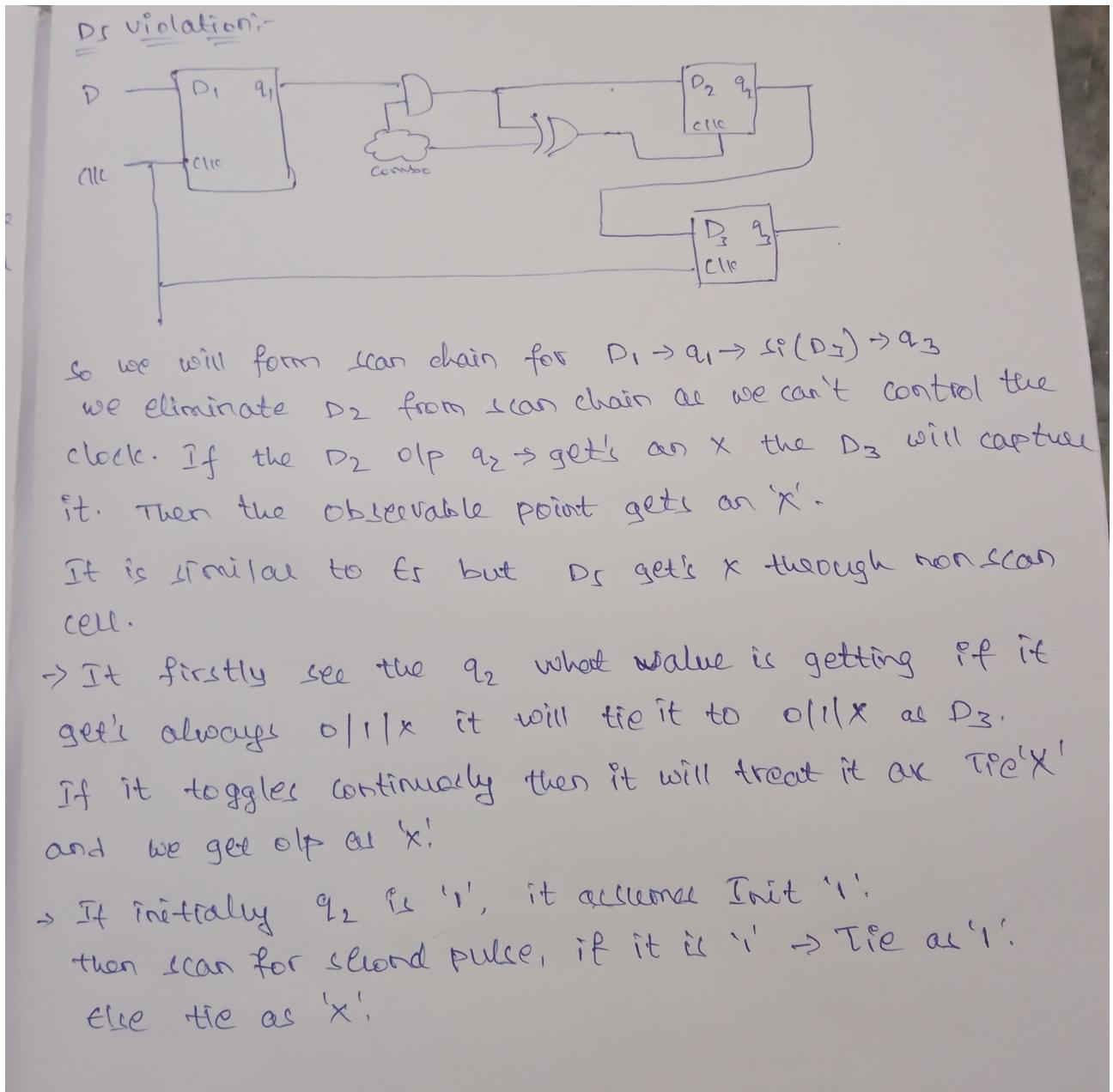
To resolve this D5 violation:

- Scan Insert Missing Elements:** Make sure all FFs and latches are **converted into scan cells** during scan insertion.
- Tie Certain FFs if Justified:**



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

- If some FFs are **redundant or tied off (TIE-0/TIE-1)** and do not affect test coverage, you may **justify their exclusion** with documentation.



E5:

Category: Extra

Default Handling: NOTE

An E5 violation occurs when an X-state (unknown logic value) is able to propagate to a Primary Output (PO) or any observable scan point.

When the application places constrained states on constrained pins and binary states on PIs and scan cells, X states must not propagate to an observable point. Failure to satisfy this rule results in the risk of X states propagating to an observable point.



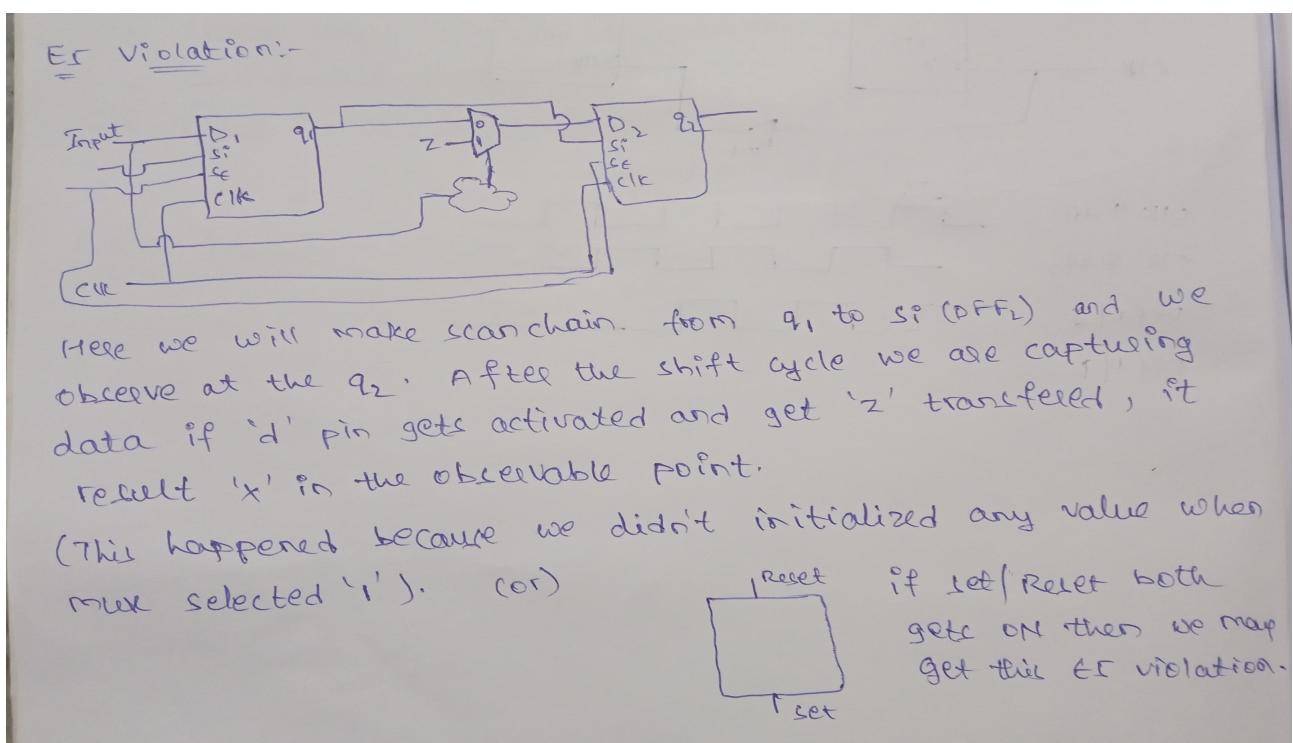
- BIST (Built-In Self-Test) → Xs reduce fault detection
- ROM/RAM uninitialized states → can be sources of X
- Tri-state contention or open circuits → can create X
- Improper latch or memory control signals → result in X

Design Fixes / Constraints:

- Use NO_Z constraints on **PIs** to prevent Z propagation.
- Ensure all **RAM/ROM blocks are initialized**.
- Properly configure **tri-state controls** and **latch enables**.
- Ensure **TIE-X logic** is not observable if not needed.

Scan Chain Control:

Apply valid control signals to **all scan-related memory or driver logic** to avoid undefined behavior.



T3:

Category: TRACE

Default Handling: Warning



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

The T3 trace violation occurs when the DFT tool fails to trace a complete sensitizable path from a scan chain output (scan_out) back to a scan chain input (scan_in) during scan shift path analysis. This means that during scan shifting, **some logic or gate is blocking the propagation of data**, breaking the shift path required for proper scan operation.

When the tool encounters the error, it reports a message similar to the following:

```
// Error: Scan chain chain1 blocked at gate /data0/reg_q_0_(360) after  
tracing 0 cells. (T3-1)
```

```
// Error: Rules checking unsuccessful, cannot exit SETUP mode.
```

Correct this error condition by accessing the simulated values of all time periods of the shift procedure.

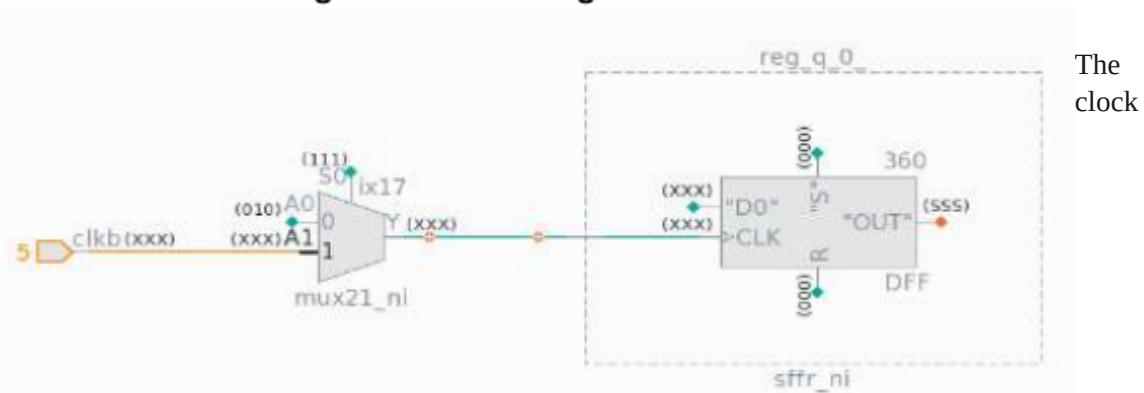
Scan chain S blocked at gate N (G) after tracing C cells. (T3-1)

This means:

- **S**: The scan chain being traced
- **N**: Name of the gate where tracing failed
- **G**: Gate type
- **C**: Number of scan cells successfully traced before blockage

For example :

Figure 7-51. Tracing T3 Violation



“CLK” should be 010, not XXX.

Trace backwards by clicking the pin for “CLK”

The “CLK” pin is driven by “clk”, which is not pulsed.

The solution is to modify the test procedure to pulse “clk” during shift.



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

Correct this error condition by accessing the simulated values of all time periods of the shift procedure.

Minutes of the Meeting

Date : 16/06/2025(Monday)

Time : 9:15 pm

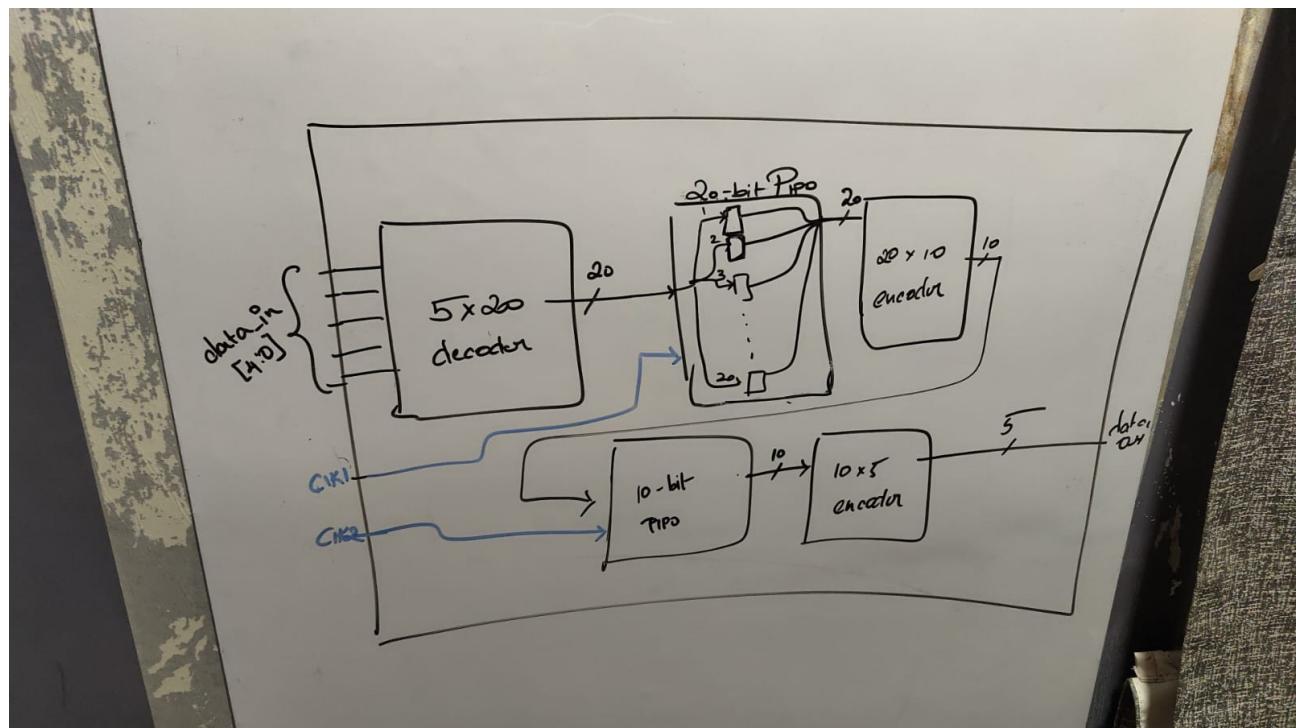
Venue : Google Meet

Agenda

- Individual report review.
- Individual QnA based on the DRCs (C,D,E,T)
- Commencing design and implementation phase.

Theoretical knowledge:

We want to write a code for the following design





Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

The design is as follows:

We have a 5 bit input data given to a 5x20 decoder and these 20 bit data will be given to 20 bit PIPO(parallel in parallel out).

After PIPO it is given to 20x10 encoder.

Then 10 bit data given to 10 bit PIPO.

Atlast 10 bit data is given to 10x5 encoder.

Then the 5 bit data is our final output.

Hands On Work:

Implemented the design, the verilog code is as follows:

GOLDEN DESIGN:

```
'timescale 1ns / 1ps

///////////////////////////////
module golden_design(refclk,clk2,data_in,reset,data_out);
    input refclk,clk2;
    input [4:0]data_in;
    input reset;
    output [4:0]data_out;

    wire [19:0] deco1,pipo1;
    wire [9:0] enco1,blackbox_out,pipo2;
    wire [9:0] x;

    decoder5x20 D1(data_in,deco1);
    pll PLL(refclk,clk1);
    pipo20 P1(clk1,reset,deco1,pipo1);
```



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)
encoder20x10 E1(pipo1,enco1);

 pipo10 P2(clk2,reset_out,x,pipo2);

 encoder10x5 E2(pipo2,data_out);

 blackbox BB(

 .reset_out(reset_out),

 .data_out(blackbox_out)

);

 xor (x[0],blackbox_out[0],enco1[0]);

 xor (x[1],blackbox_out[1],enco1[1]);

 xor (x[2],blackbox_out[2],enco1[2]);

 xor (x[3],blackbox_out[3],enco1[3]);

 xor (x[4],blackbox_out[4],enco1[4]);

 xor (x[5],blackbox_out[5],enco1[5]);

 xor (x[6],blackbox_out[6],enco1[6]);

 xor (x[7],blackbox_out[7],enco1[7]);

 xor (x[8],blackbox_out[8],enco1[8]);

 xor (x[9],blackbox_out[9],enco1[9]);

endmodule

module pll(

 input refclk,

 output clk

);

 //assign clk=refclk;

endmodule

module blackbox(

 output reset_out,



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)
output [9:0]data_out

```
);  
  
//assign reset_out = 1'b0;  
  
//assign data_out = 10'd0;  
  
endmodule
```

```
module decoder5x20 (
```

```
    input [4:0] in,  
    output [19:0] out  
);
```

```
wire A, B, C, D, E;
```

```
wire nA, nB, nC, nD, nE;
```

```
assign A = in[4];
```

```
assign B = in[3];
```

```
assign C = in[2];
```

```
assign D = in[1];
```

```
assign E = in[0];
```

```
not g0(nA, A);
```

```
not g1(nB, B);
```

```
not g2(nC, C);
```

```
not g3(nD, D);
```

```
not g4(nE, E);
```

```
// out[0] = ~A & ~B & ~C & ~D & ~E
```

```
and g5(out[0], nA, nB, nC, nD, nE);
```



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

// out[1] = ~A & ~B & ~C & ~D & E

and g6(out[1], nA, nB, nC, nD, E);

// out[2] = ~A & ~B & ~C & D & ~E

and g7(out[2], nA, nB, nC, D, nE);

// out[3] = ~A & ~B & ~C & D & E

and g8(out[3], nA, nB, nC, D, E);

// out[4] = ~A & ~B & C & ~D & ~E

and g9(out[4], nA, nB, C, nD, nE);

// out[5] = ~A & ~B & C & ~D & E

and g10(out[5], nA, nB, C, nD, E);

// out[6] = ~A & ~B & C & D & ~E

and g11(out[6], nA, nB, C, D, nE);

// out[7] = ~A & ~B & C & D & E

and g12(out[7], nA, nB, C, D, E);

// out[8] = ~A & B & ~C & ~D & ~E

and g13(out[8], nA, B, nC, nD, nE);

// out[9] = ~A & B & ~C & ~D & E

and g14(out[9], nA, B, nC, nD, E);



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

// out[10] = ~A & B & ~C & D & ~E

and g15(out[10], nA, B, nC, D, nE);

// out[11] = ~A & B & ~C & D & E

and g16(out[11], nA, B, nC, D, E);

// out[12] = ~A & B & C & ~D & ~E

and g17(out[12], nA, B, C, nD, nE);

// out[13] = ~A & B & C & ~D & E

and g18(out[13], nA, B, C, nD, E);

// out[14] = ~A & B & C & D & ~E

and g19(out[14], nA, B, C, D, nE);

// out[15] = ~A & B & C & D & E

and g20(out[15], nA, B, C, D, E);

// out[16] = A & ~B & ~C & ~D & ~E

and g21(out[16], A, nB, nC, nD, nE);

// out[17] = A & ~B & ~C & ~D & E

and g22(out[17], A, nB, nC, nD, E);

// out[18] = A & ~B & ~C & D & ~E

and g23(out[18], A, nB, nC, D, nE);

// out[19] = A & ~B & ~C & D & E



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)
and g24(out[19], A, nB, nC, D, E);

endmodule

// D latch gate-level

module d_latch (

 input wire d,

 input wire en,

 output wire q

);

 wire dbar, s, r, qbar;

 not u1(dbar, d);

 and u2(s, d, en);

 and u3(r, dbar, en);

 nor u4(q, r, qbar);

 nor u5(qbar, s, q);

endmodule

// Master-Slave D flip-flop

module dff_gate (

 input wire clk,

 input wire d,

 input wire reset,

 output wire q

);

 wire nclk,nreset;



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

```
wire qm,d_mux;  
  
not u1(nclk, clk);  
  
not u2(nreset, reset);  
  
and(d_mux,d,nreset);  
  
d_latch master (  
    .d(d_mux),  
    .en(nclk),  
    .q(qm)  
);  
  
d_latch slave (  
    .d(qm),  
    .en(clk),  
    .q(q)  
);  
endmodule  
  
// 20-bit PIPO register  
module pipo20 (  
    input wire clk,  
    input wire reset,  
    input wire [19:0] d_in,  
    output wire [19:0] q_out  
);  
    genvar i;  
    generate  
        for (i = 0; i < 20; i = i + 1) begin : dff_array  
            dff_gate dff_inst (  
                .
```



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)
.clk(clk),

.reset(reset),

.d(d_in[i]),

.q(q_out[i])

);

end

endgenerate

endmodule

module encoder20x10 (in,out);

input [19:0] in;

output [9:0] out;

assign out[9:5] = 5'b0;

or(out[0], in[1] , in[3] , in[5] , in[7] , in[9] , in[11] , in[13] , in[15] , in[17] , in[19]);

or(out[1] , in[2] , in[3] , in[6] , in[7] , in[10] , in[11] , in[14] , in[15] , in[18] , in[19]);

or(out[2] , in[4] , in[5] , in[6] , in[7] , in[12] , in[13] , in[14] , in[15] , in[19]);

or(out[3] , in[8] , in[9] , in[10] , in[11] , in[12] , in[13] , in[14] , in[15]);

or(out[4] , in[16] , in[17] , in[18] , in[19]);

endmodule

module pip010 (

input wire clk,

input wire reset,

input wire [9:0] d_in,

output wire [9:0] q_out



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

);

```
genvar i;  
generate  
  for (i = 0; i < 10; i = i + 1) begin : dff_array  
    dff_gate dff_inst (  
      .clk(clk),  
      .reset(reset),  
      .d(d_in[i]),  
      .q(q_out[i])  
    );  
  end  
endgenerate  
endmodule
```

```
module encoder10x5(in, out);
```

```
  input [9:0] in;  
  output [4:0] out;
```

```
  wire P0, P1, P2, P3, P4, P5, P6, P7, P8, P9;
```

```
// Intermediate wires for inverted inputs to simplify logic
```

```
  wire not_in_0, not_in_1, not_in_2, not_in_3, not_in_4, not_in_5, not_in_6, not_in_7,  
  not_in_8, not_in_9;
```

```
  not (not_in_0, in[0]);  
  not (not_in_1, in[1]);  
  not (not_in_2, in[2]);
```



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)
not (not_in_3, in[3]);

not (not_in_4, in[4]);

not (not_in_5, in[5]);

not (not_in_6, in[6]);

not (not_in_7, in[7]);

not (not_in_8, in[8]);

not (not_in_9, in[9]);

// Priority terms (P_N)

assign P9 = in[9];

and (P8, in[8], not_in_9);

and (P7, in[7], not_in_8, not_in_9);

and (P6, in[6], not_in_7, not_in_8, not_in_9);

and (P5, in[5], not_in_6, not_in_7, not_in_8, not_in_9);

and (P4, in[4], not_in_5, not_in_6, not_in_7, not_in_8, not_in_9);

and (P3, in[3], not_in_4, not_in_5, not_in_6, not_in_7, not_in_8, not_in_9);

and (P2, in[2], not_in_3, not_in_4, not_in_5, not_in_6, not_in_7, not_in_8, not_in_9);

and (P1, in[1], not_in_2, not_in_3, not_in_4, not_in_5, not_in_6, not_in_7, not_in_8, not_in_9);

and (P0, in[0], not_in_1, not_in_2, not_in_3, not_in_4, not_in_5, not_in_6, not_in_7, not_in_8, not_in_9);

// Output bit 0 (LSB)

or (out[0], P1, P3, P5, P7, P9);

// Output bit 1

or (out[1], P2, P3, P6, P7);



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

// Output bit 2

or (out[2], P4, P5, P6, P7);

// So out[3] is 1 for P8 or P9

or (out[3], P8, P9);

assign out[4] = 1'b0;

endmodule

TESTBENCH

`timescale 1ns / 1ps

//////////

module golden_design_tb;

reg refclk,clk2,reset;

reg [4:0] data_in;

wire [4:0] data_out;

golden_design

dut.refclk(refclk),.reset(reset),.clk2(clk2),.data_in(data_in),.data_out(data_out));

initial begin

refclk=1;

forever #5 refclk = ~refclk;

end



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

initial begin

clk2=1;

forever #5 clk2 = ~clk2;

end

initial begin

reset=1'b1;

#10 reset=1'b0;

data_in = 5'b00001;

#10; data_in= 5'b00100;

#10; data_in = 5'b00011;

#10; data_in = 5'b00111;

#10; data_in = 5'b11111;

#40;

\$finish;

end

initial

\$monitor("data_in = %b and data_out = %b ", data_in,data_out);

endmodule

Minutes of the Meeting

Date : 28/06/2025(Monday)

Time : 9:15 pm

Venue : Google Meet

Agenda



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

- To verify the design.

Theoretical Knowledge:

Adding some black box to the previous design and creating inbuilt E5 violation in the design.

Minutes of the Meeting

Date : 03/07/2025(Thursday)

Time : 9:30 pm

Venue : Google Meet

Agenda

Creating the pll blackbox and some other logic.

Theoretical knowledge:

Clocks: (2)---Refclk from top level to PLL , Clk2 from top level to 10 pipo registers.

Internally Generated clocks:(1)---Clk1 from PLL to 20 pipo registers.

Resets:(1) Reset from top level

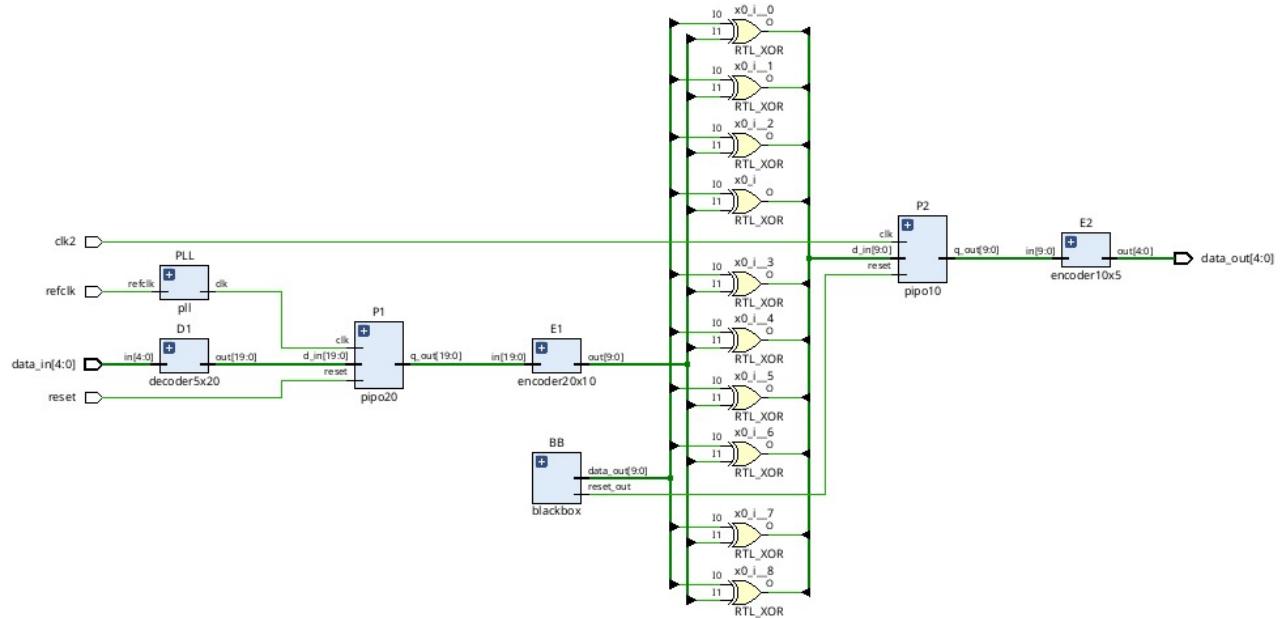
Internally Generated resets: (1)--- reset from Black box to 10 pipo registers/

Black boxes:(2) PLL and empty Black box

Practical Implementation:



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)



Minutes of the Meeting

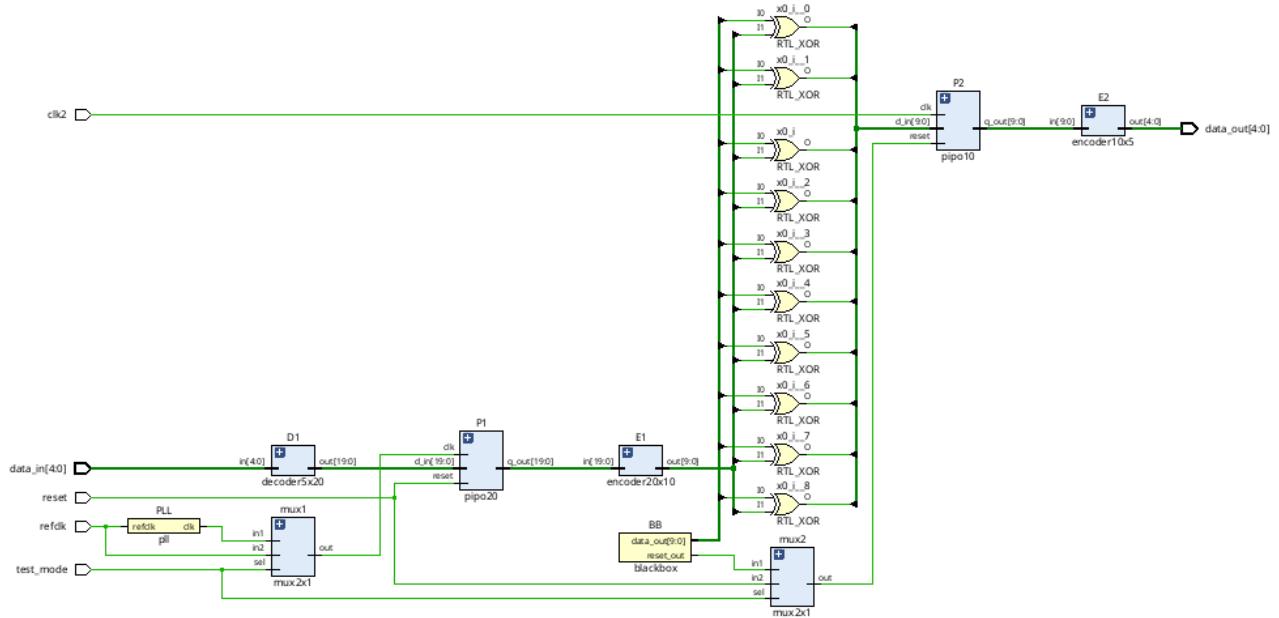
Date : 06/07/2025(Sunday)

Time : 9:30 pm

Venue : Google Meet

Agenda

Solving the DRC's occurred in the design like c6,c9 by adding some mux.



Minutes of the Meeting

Date : 09/07/2025(wednesday)

Time : 9:30 pm

Venue : Google Meet

Agenda :

To solve E5 violation caused by Black box by knowing the concept of configuring test points.

Adding scan insertion to the design.

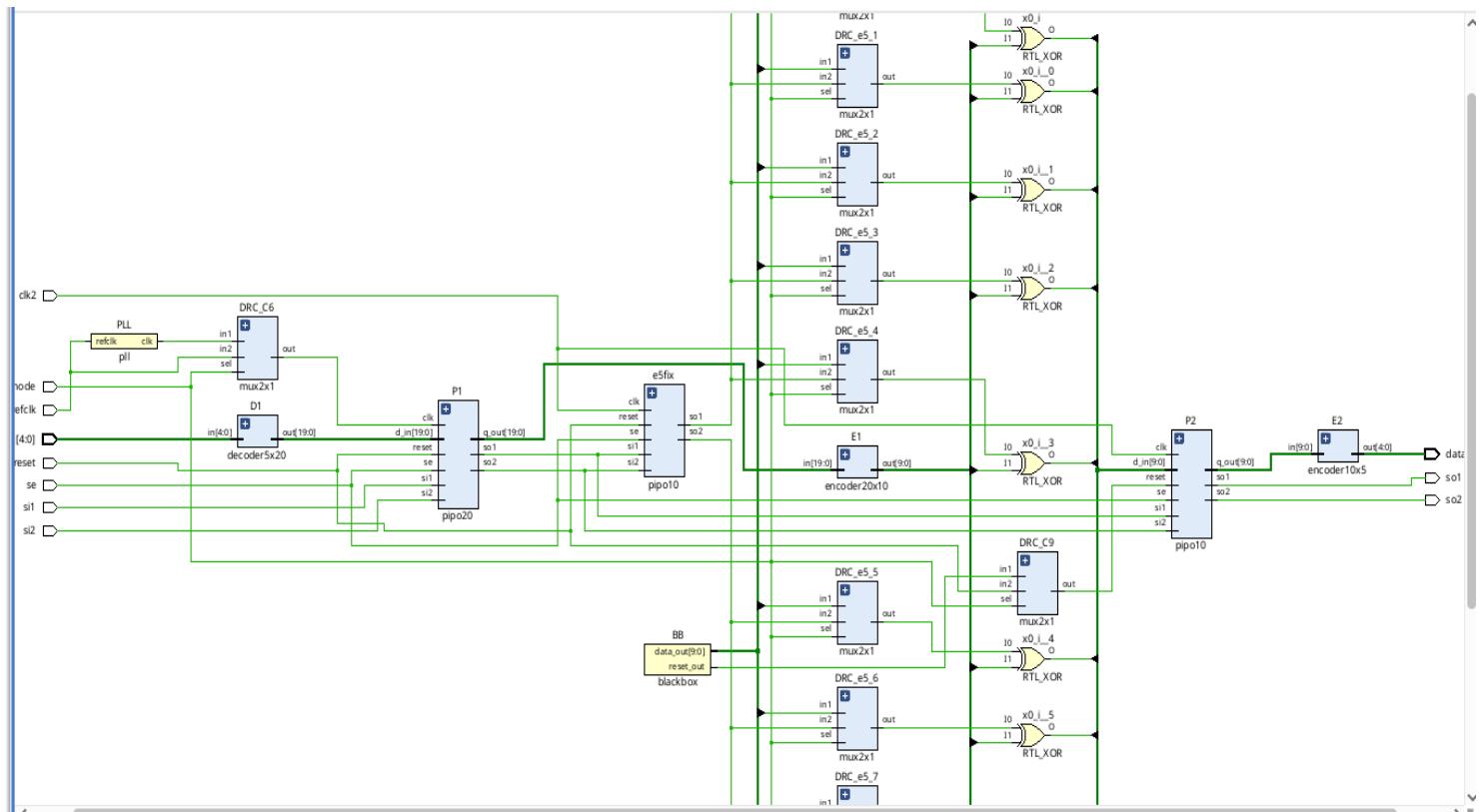
Theoretical knowledge:

To avoid E5 violation we use mux to get two inputs by having selection as testmode. If test mode is 0 we have input coming from blackbox. If test mode is 1 we should get input from a test vector but if we use test vector from top it degrades the coverage and also if we tie it to either 0 / 1 we may loose the coverage as we cant test all stuck at faults.

To overcome this we use **Dynamic test points** as they get test vector from a pipo which gets input from the previous scan flops.



Hands on :





Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

Executive Summary

This project focuses on the **design, modification, and verification of a digital circuit** while ensuring compliance with industry-standard **Design Rule Checks (DRC)**. The work covers multiple aspects of physical and logical design such as clock terminology, clock rules implementation, black box insertion, scan chain insertion, error correction, and resolving violations. The project highlights the **practical challenges faced during VLSI design**, especially with clocking, PLL integration, and scan insertion. Finally, violations such as **E5 errors** were analyzed and corrected, ensuring that the final design met sign-off requirement



Introduction

Background and Context of the Project

With the continuous scaling of semiconductor technology, digital designs have become more complex, requiring strict adherence to **Design Rule Checks (DRC)** and robust clocking strategies. Clock management, design modifications, black box integration, and scan insertion are critical steps in achieving a **functional, manufacturable, and testable design**. This project was carried out to simulate an **industry-oriented design environment**, where each stage of the flow — from implementation to DRC fixing — mirrors real VLSI practices.

Problem Statement or Goals of the Project

The main challenges addressed in this project include:

- Understanding and applying **clock terminology and rules** to ensure proper timing.
- Implementing a **practical design** and integrating additional logic such as PLL and black boxes.
- Identifying and **resolving DRC and E5 violations** encountered during design flow.
- Performing **scan chain insertion** to enhance design-for-testability (DFT).
- Delivering a **final clean design** that is error-free and industry compliant.

Scope and Limitations of the Project

Scope:

- Covers **clock rule implementation**, design modification, and PLL insertion.
- Focuses on **error identification and correction** in physical design.
- Includes **DRC fixing and scan insertion** for improving manufacturability and testability.

Innovation Component in the Project

- **Integration of black boxes** to simulate **third-party IP insertion**, reflecting real SoC integration challenges.



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

- **PLL addition** for clock generation and synchronization, making the design closer to real-world silicon requirements.
- **Systematic fixing of E5 violations** and DRCs, showcasing a structured debugging approach.
- Inclusion of **scan insertion**, demonstrating awareness of DFT concepts and bridging functional design with testability.



Project Objectives

Objectives and Goals of the Project

The objectives of this project were:

1. To understand and apply **clock terminology and rules** for reliable circuit timing.
2. To implement a **practical digital design** and modify it by adding black boxes and PLL logic.
3. To identify and **correct design errors** that arise during synthesis and implementation.
4. To analyze and resolve **Design Rule Check (DRC) violations** systematically.
5. To specifically **fix E5 violations** encountered in the design flow.
6. To perform **scan chain insertion** to enhance testability and design validation.
7. To gain hands-on experience with **industry-relevant VLSI design methodologies**.

Expected Outcomes and Deliverables

- A **functional and verified design** free from DRC and E5 violations.
- Successful **integration of PLL and black box modules** into the design.
- A design enhanced with **scan chains** to support testability.
- A **documented methodology** highlighting the steps of error detection, correction, and verification.
- A **comprehensive project report** demonstrating both the technical execution and the industry relevance of the work.



Methodology and Results

Methods / Technology Used

- **Digital Design Flow:** The project followed a structured VLSI design methodology including design entry, synthesis, modification, verification, and DRC fixing.
- **Clock Domain Handling:** Implemented clock rules, managed skew, latency, and integrated PLL for clock synchronization.
- **Error Correction:** Identified and corrected **E5 violations** and DRC errors during place-and-route.
- **Design-for-Testability (DFT):** Inserted scan chains and verified scan shifting integrity using test patterns.
- **Iterative Debugging Approach:** Errors and violations were resolved in multiple iterations until a **clean design** was achieved.

Tools / Software Used

- **EDA Tools:**
 - Xilinx Vivado
- **Simulation Tools:** *Xilinx Vivado* for functional verification.
- **Version Control:** Daily report for tracking modifications and sharing final code/report.

Project Architecture

The **project architecture** followed a hierarchical digital design flow:

1. **Input RTL Design** → Base module developed.
2. **Clock Rule Integration** → Defined skew, latency, and duty cycle parameters.
3. **Design Modification** → Added black boxes and PLL module.
4. **Synthesis & Implementation** → Logical to physical conversion.
5. **Error Detection** → DRC and E5 violations identified.
6. **Violation Fixing** → Placement/routing fixes applied.
7. **Scan Chain Insertion** → DFT enhancement for testability.



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust - IERY)

8. Verification & Final Clean Design → Functional + DRC verified.

Project GitHub Link:



Learning and Reflection

Learning & Reflection

New Learnings (Technology & Management)

- **Technical Learnings**

- Gained a strong understanding of **clock terminology and clock rules**, including skew, latency, duty cycle, and jitter.
- Learned how to **integrate PLLs** for clock synchronization and how to manage black box modules in design flow.
- Acquired hands-on knowledge of **Design Rule Checks (DRC)**, identifying violations, and systematically fixing them.
- Practiced **error correction of E5 violations** and understood their impact on physical design.
- Implemented **scan chain insertion** and verified functionality through test patterns, reinforcing concepts of **Design for Testability (DFT)**.
- Improved scripting skills in **TCL/Python** to automate design tasks and manage reports.

- **Project/Management Learnings**

- Developed the ability to **plan design iterations** effectively, handling errors step by step.
- Gained experience in **collaborative documentation and version control** using GitHub.
- Learned the importance of **time management** in completing multiple design cycles within deadlines.
- Enhanced **problem-solving mindset**, as each violation required careful debugging and innovative fixes.



Conclusion and Future Scope

Conclusion

Recap of Objectives and Achievements

The main objectives of this project were to:

- Understand and apply **clock terminology and rules** for reliable design implementation.
- Modify the design by adding **PLL logic and black box modules**.
- Identify and resolve **DRC violations** and specifically address **E5 errors**.
- Enhance the design with **scan chain insertion** for improved testability.

Achievements:

- Successfully implemented clock rules and ensured proper synchronization using PLL.
- Modified the design to integrate black boxes without functional conflicts.
- Identified and corrected **all DRC violations**, ensuring a clean design ready for sign-off.
- Resolved **E5 violations**, demonstrating practical debugging skills.
- Performed **scan insertion** and verified scan integrity through pattern shifting.
- Documented the entire flow systematically, reflecting **real-world VLSI practices**.