Expanding role of Static Signoff in Verification Coverage

Vikas Sachdeva

Head of Business, APAC

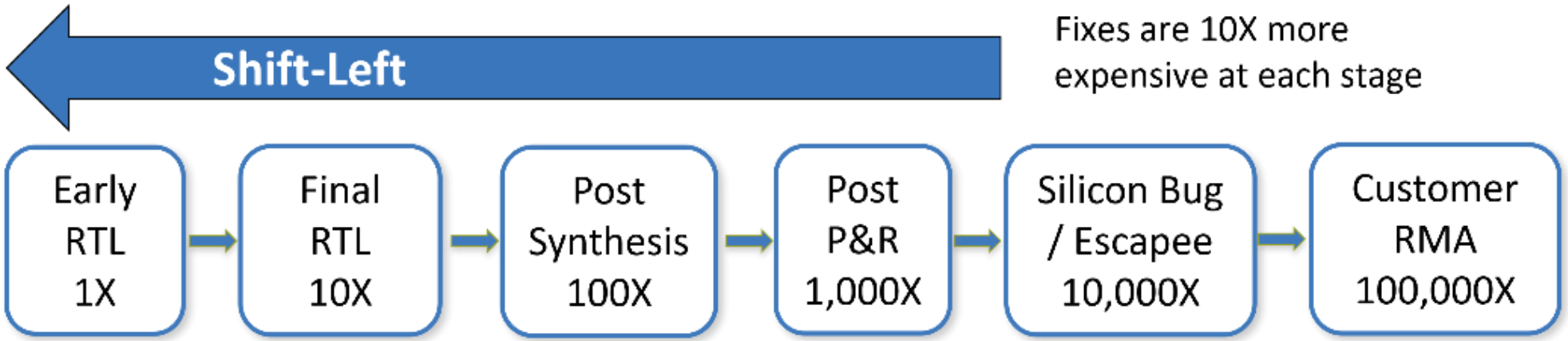
Vikas Sachdeva

Vikas Sachdeva serves as the Head of Business Development for the APAC region at Real Intent, where he leads business development and product strategy for the company's key static signoff products. A graduate of the Indian Institute of Technology Delhi, Vikas is an entrepreneur and technologist with a deep expertise in EDA and semiconductors. He is passionate about technology, product innovation, and nurturing the next generation of talent in VLSI. Additionally, he is a best-selling author on Amazon with his book, "Becoming Irreplaceable."

Static Signoff Applications

- Static Timing Analysis
- Others?

Shift Left


Shift-Left
Fixes are 10X more expensive at each stage
Early RTL 1X
Final RTL 10X
Post Synthesis 100X
Post P&R 1,000X
Silicon Bug / Escapee 10,000X
Customer RMA 100,000X

INTRODUCTION

is the earliest possible efficient verification of each design step

is earliest possible efficient verification of each design step

efficiently enables shift left


Back to Basics

Static Verification vs Dynamic Verification

- Dynamically computes design behavior to find failures
- Coverage limited to test cases

Static Verification vs Dynamic Verification

Dynamic verification

- Dynamically computes design behavior to find failures
- Coverage limited to test cases

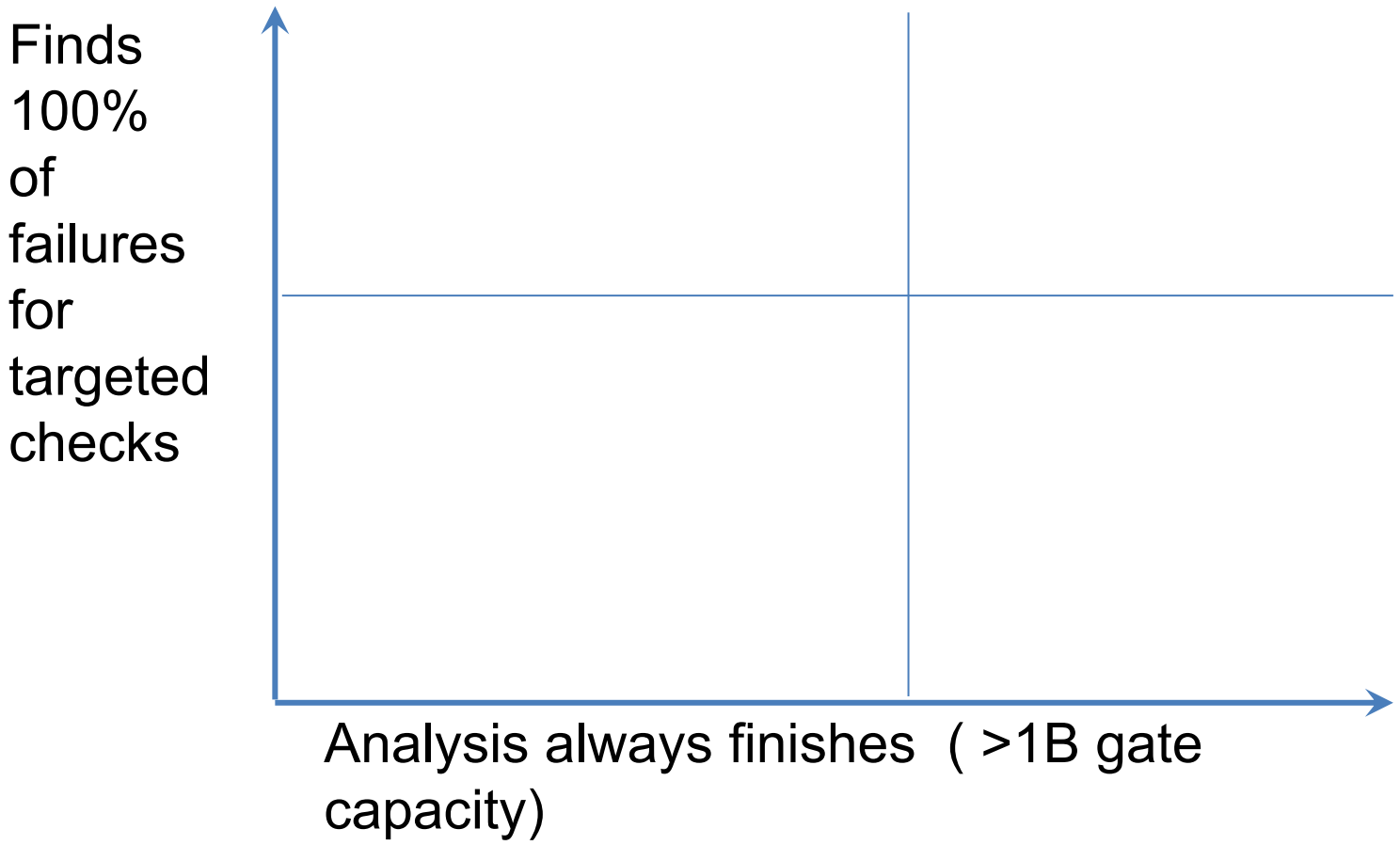
Simulation
Emulation

Static verification

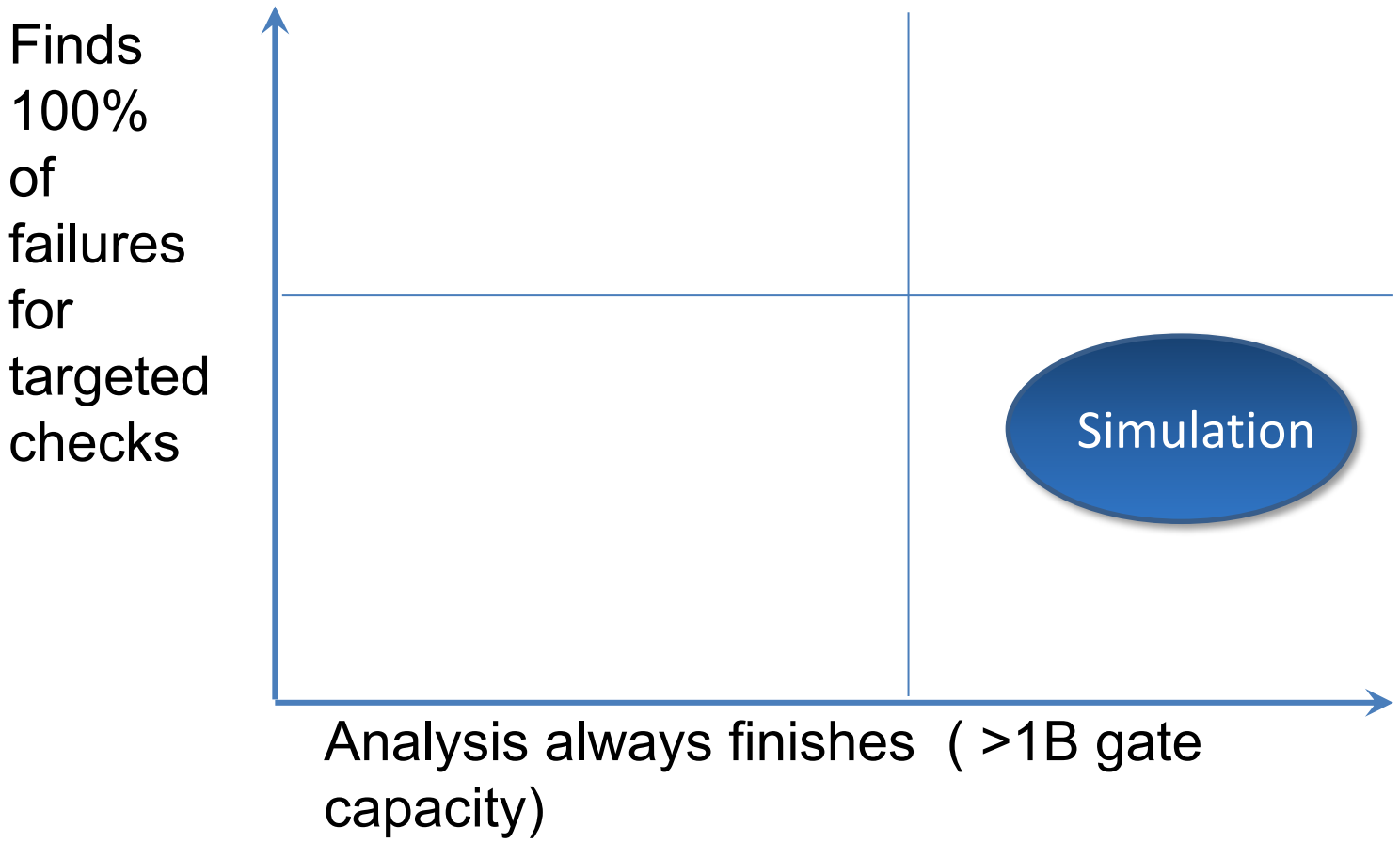
- Utilizes search & analysis to find ALL targeted failures
- Test cases not required

CDC
Lint…
Formal
STA
DRC

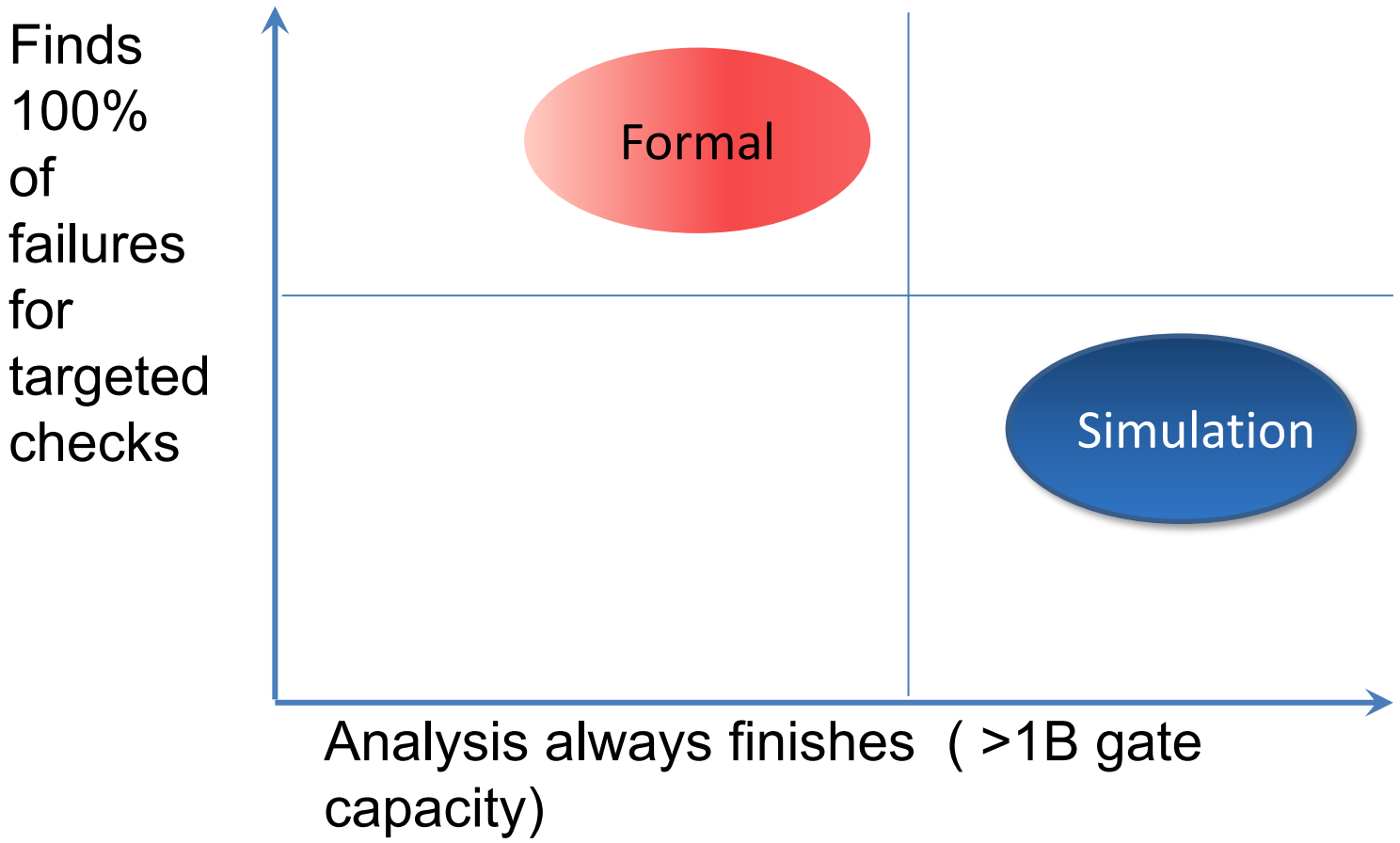
Static Sign-off vs Formal & Simulation



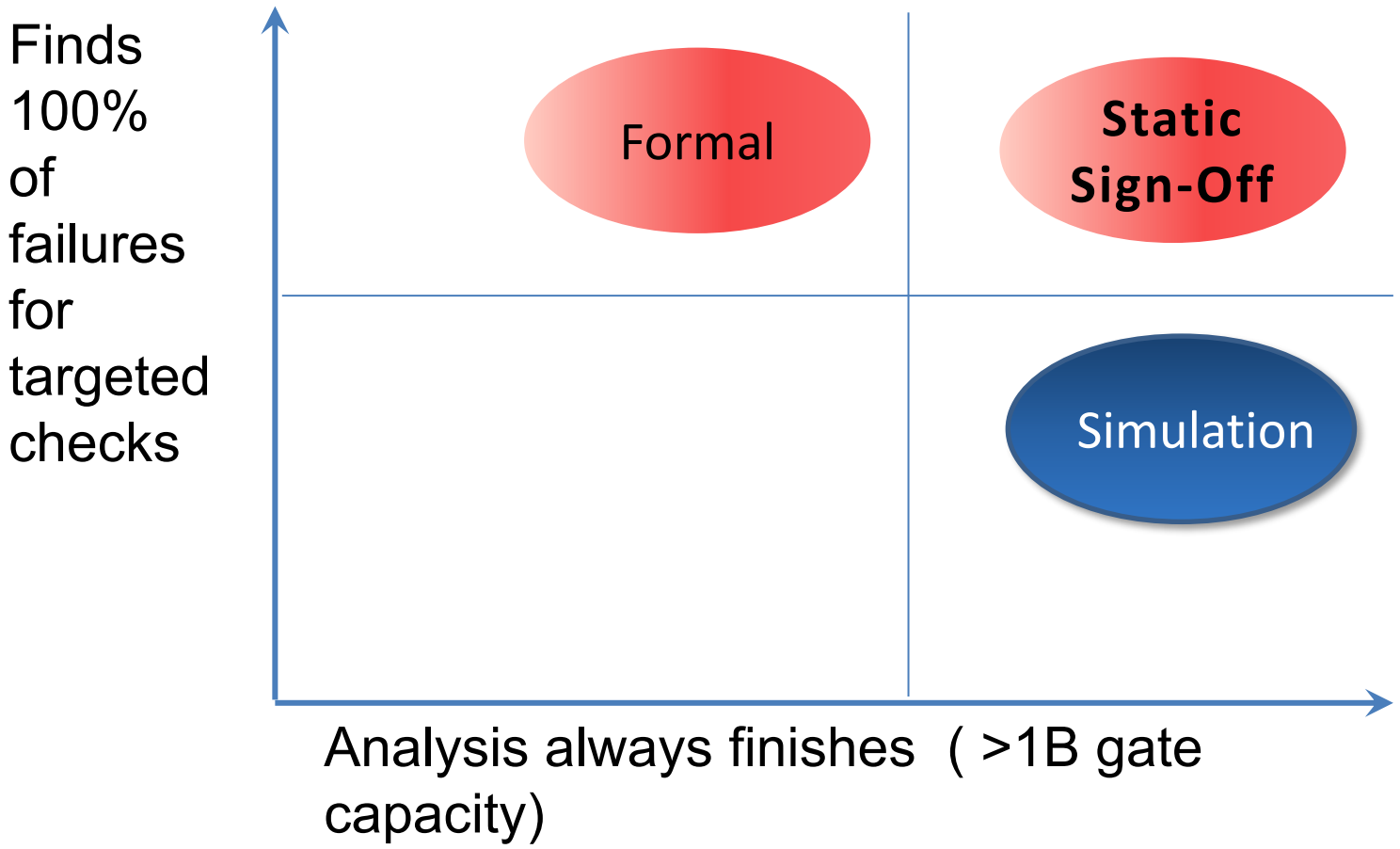
Static Sign-off vs Formal & Simulation



Static Sign-off vs Formal & Simulation



Static Sign-off vs Formal & Simulation

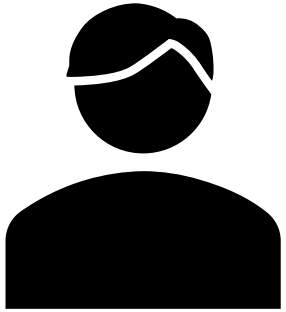


Static Sign-off but …

- Static sign-off has advantages
- But
  - What drives it expansion?
  - How do you remove the inertia/friction?
  - How do you achieve your goals of shift left?

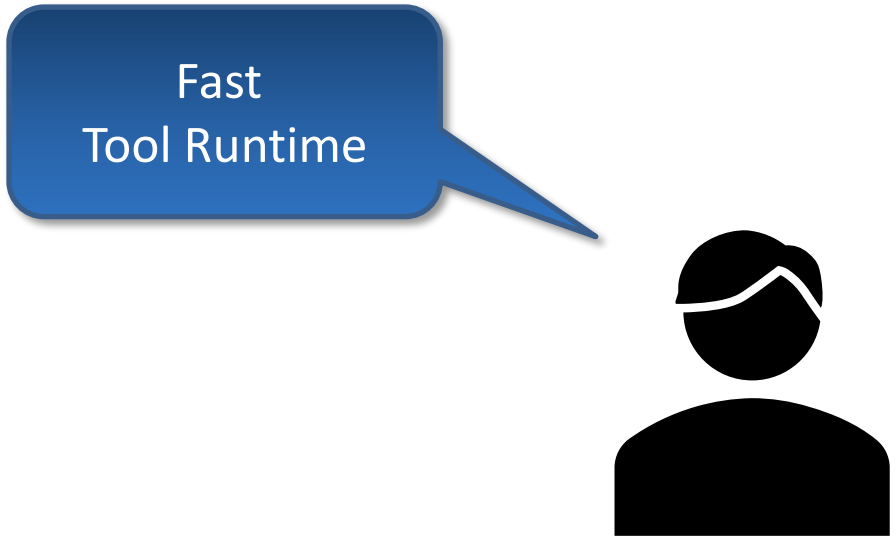
Superior User Experience Drives Shift Left

Four enabling elements of user experience



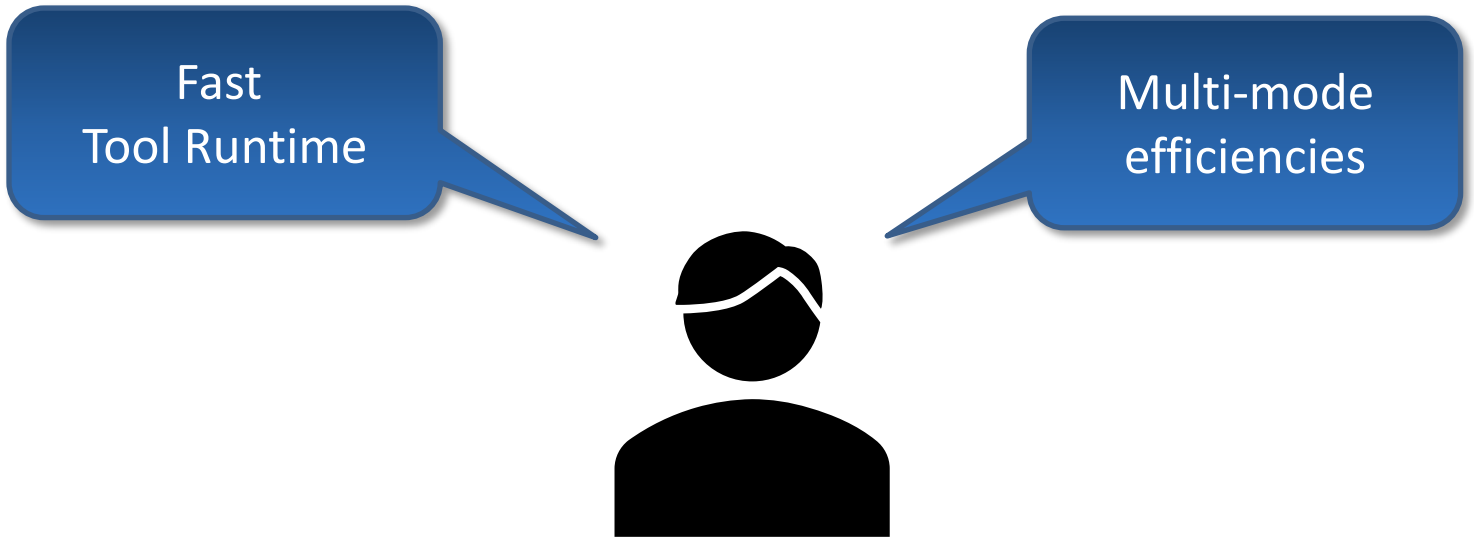
Superior User Experience Drives Shift Left

Four enabling elements of user experience



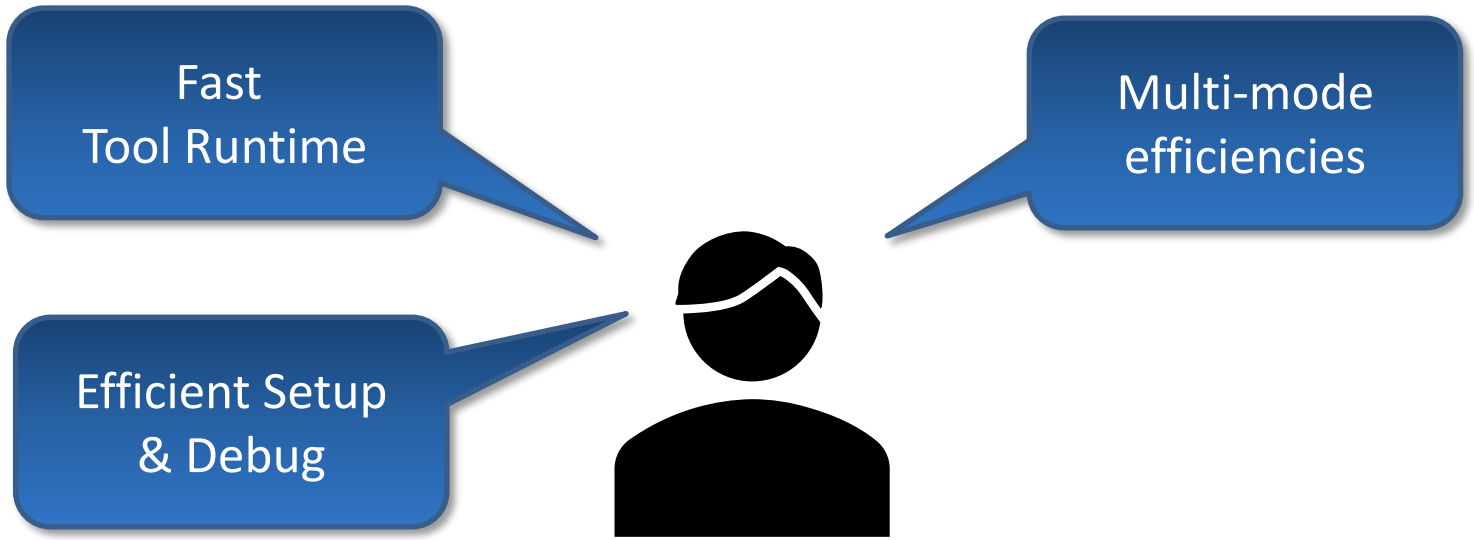
Superior User Experience Drives Shift Left

Four enabling elements of user experience



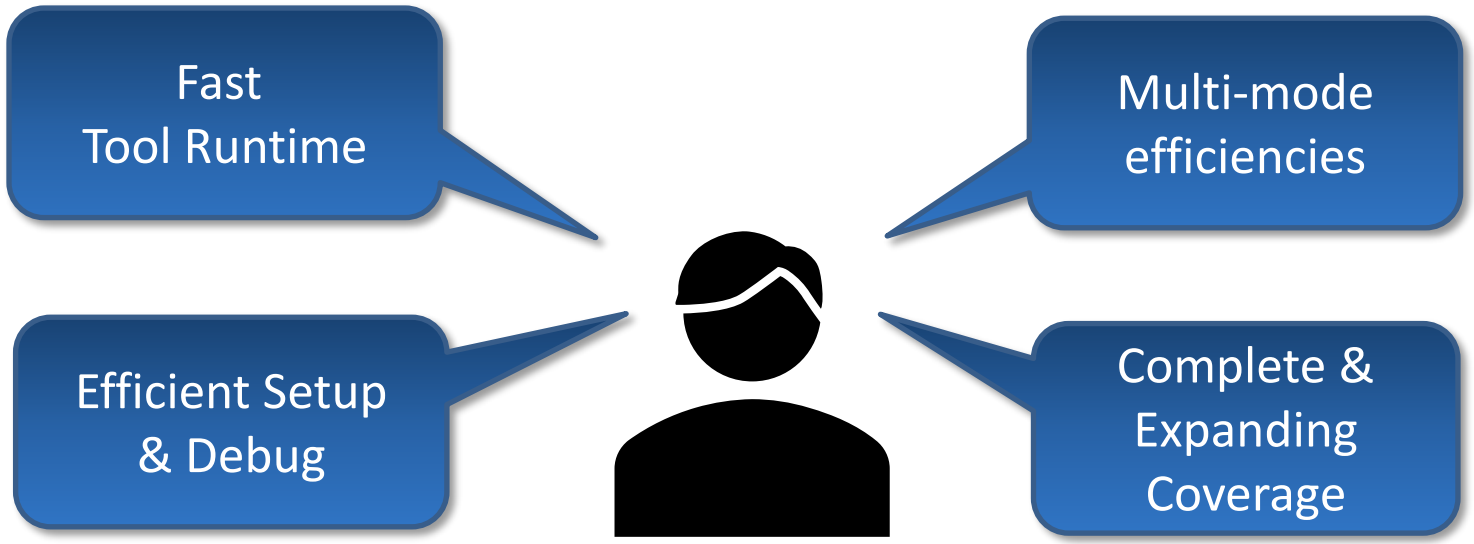
Superior User Experience Drives Shift Left

Four enabling elements of user experience



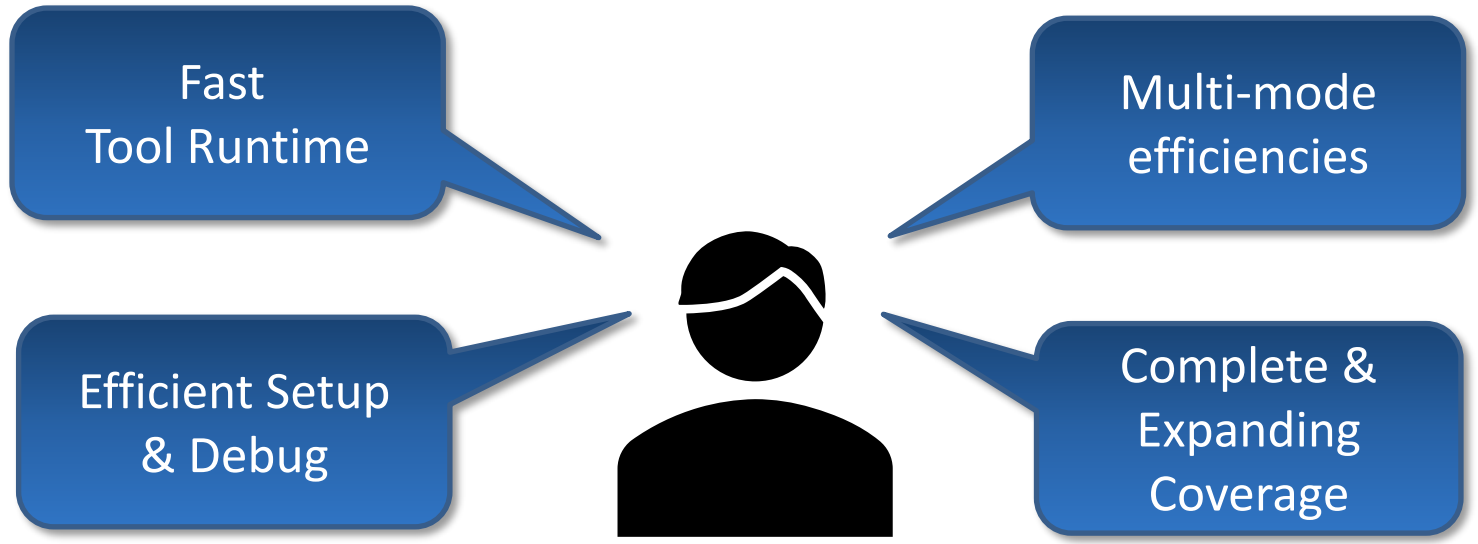
Superior User Experience Drives Shift Left

Four enabling elements of user experience



Superior User Experience Drives Shift Left

Four enabling elements of user experience

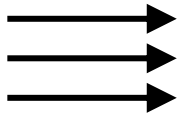


Advanced tools & methodologies
maximize engineering efficiency during design

Enabling Faster Runtime

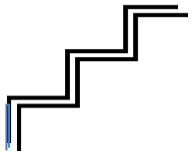
Customized static sign-off engines

Parallel Processing



Hierarchy & abstraction

Incremental analysis

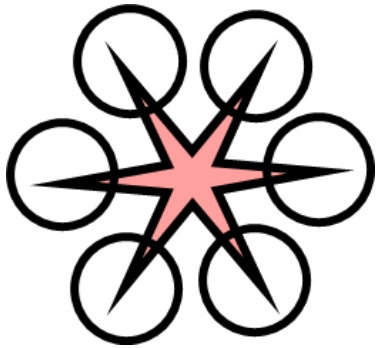


Multi-Mode Efficiencies

Multimode tools = highest engineering efficiency

1 set up. Complete coverage in 1 run. 1 consolidated report.

- Multi-Mode CDC
- Multi-Scenario RDC
- Multi-Test Mode DFT
- Multi-Policy Lint



Enabling Efficient Setup & Debug

Unified, automated setup across applications → Faster ramp up

Enabling Efficient Setup & Debug

Unified, automated setup across applications → Faster ramp up

Configurations & high granularity rules → Low noise reporting

Enabling Efficient Setup & Debug

Unified, automated setup across applications → Faster ramp up

Configurations & high granularity rules → Low noise reporting

Application-customize Debug: CDC, Lint, RDC…

- Customizable reporting -- hierarchy/organization
- Root cause grouping -- faster refinement
- Targeted attributes & CLIs -- searching & filtering
- Pattern matching-based error targeting

Enabling Efficient Setup & Debug

Unified, automated setup across applications → Faster ramp up

Configurations & high granularity rules → Low noise reporting

Application-customize Debug: CDC, Lint, RDC…

- Customizable reporting -- hierarchy/organization
- Root cause grouping -- faster refinement
- Targeted attributes & CLIs -- searching & filtering
- Pattern matching-based error targeting

Complete, Expanding Coverage of Failure Modes

Clock domain crossing

- Multi-Mode – for all clocking configurations
- Hierarchical analysis with flat accuracy
- Glitch checking to prevent netlist failure
- Design-aware Dynamic CDC verification

Complete, Expanding Coverage of Failure Modes

Clock domain crossing

- Multi-Mode – for all clocking configurations
- Hierarchical analysis with flat accuracy
- Glitch checking to prevent netlist failure
- Design-aware Dynamic CDC verification

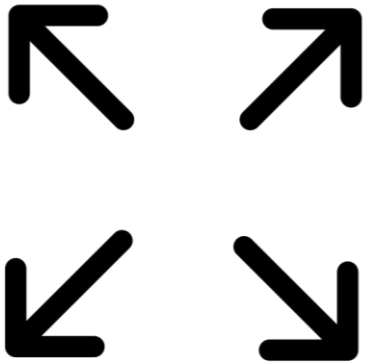
Higher coverage with fine-grained, non-overlapping rules

- Lint - 600 checks
- DFT - 100 checks

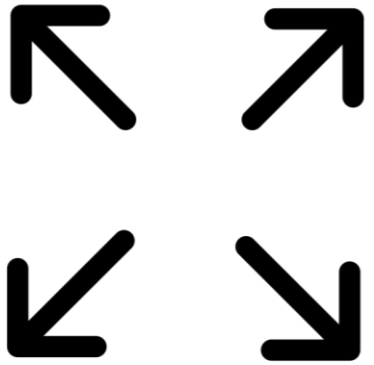
Expanding Coverage -- New Applications

New failure modes

- RDC
- Glitch



Expanding Coverage -- New Applications



New failure modes

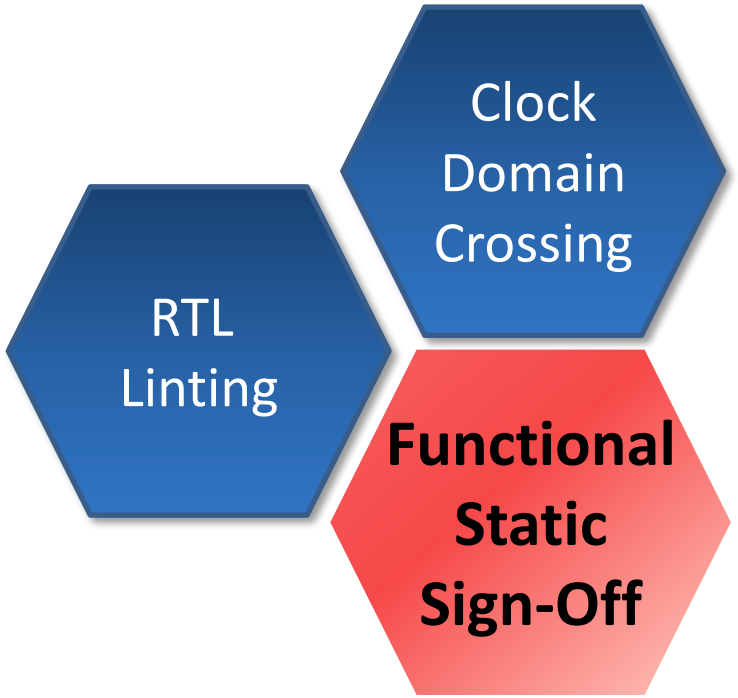
- RDC
- Glitch

New “Shift Left” applications -- streamline RTL & netlist checking

- Connectivity
- DFT
- Design Initialization

Functional Static Sign-Off Expanding Applications

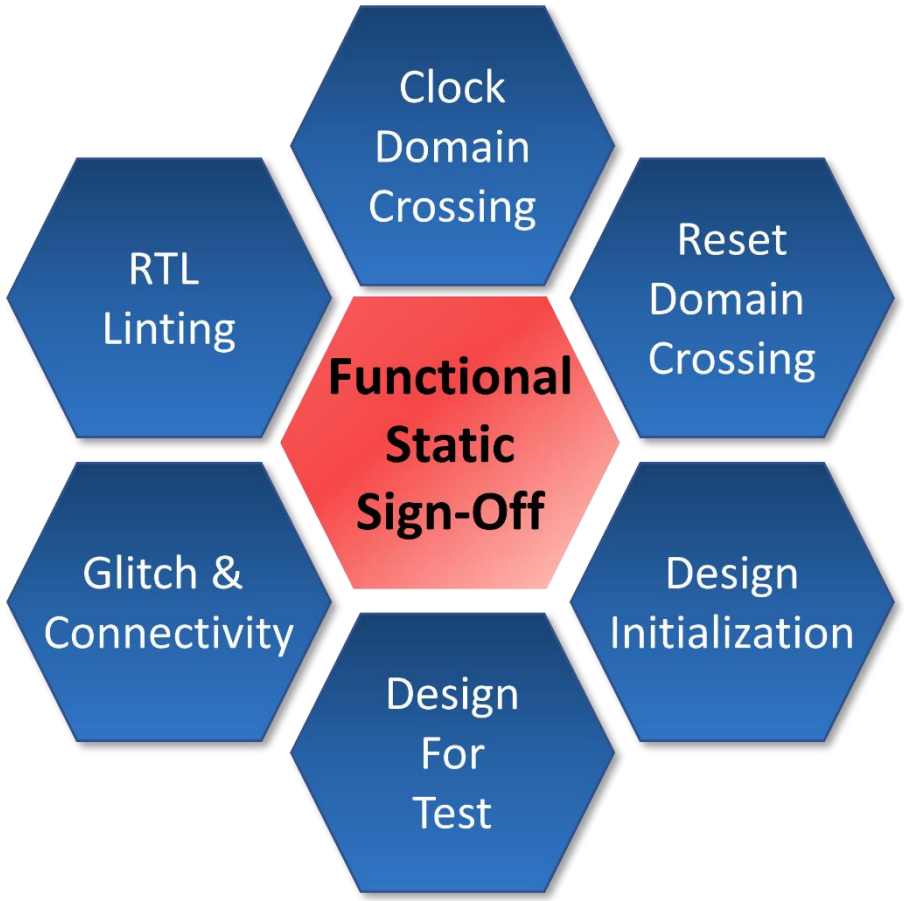
Functional static sign-off began with RTL Linting & CDC



Functional Static Sign-Off Expanding Applications

Functional static sign-off began with RTL Linting & CDC

The target applications continuously expand



Superior User Experience Driving Shift Left

Four enabling elements of user experience

- Fast Tool Runtimes
- Multimode
- Efficient set up & debug
- Expanding coverage of failure modes

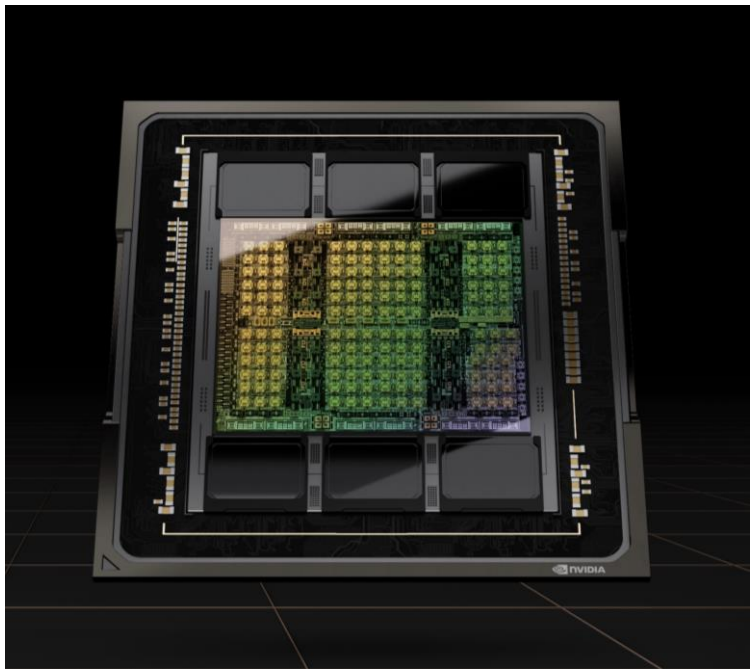
Engineering ROI expanding
functional static sign-off domains & usage

Let’s examine this claim!

ASYNCHRONOUS LOGIC SIGN-OFF BEYOND CDC

Synchronous vs Asynchronous Logic Verification

- Scale of Async logic usage
- Large variation across top level blocks
- 10’s to 100’s of clock domain
- 100’s to >100K synchronizers
- 1% to 70% flops with async resets



Source: www.realintent.com

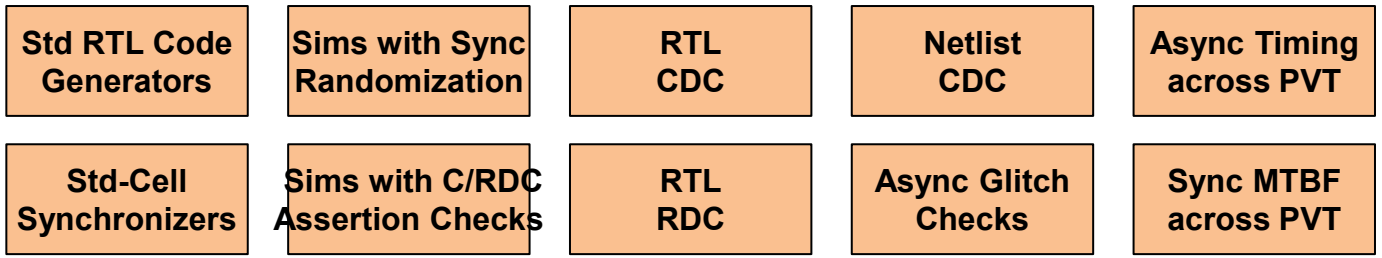
Synchronous vs Asynchronous Logic Verification

- Synchronous paths are >99% of typical designs
- Relatively SAFE from Metastability/Randomness
- Verified efficiently with high-confidence using the Core ASIC sign-off flows

Source: www.realintent.com

Synchronous vs Asynchronous Logic Verification

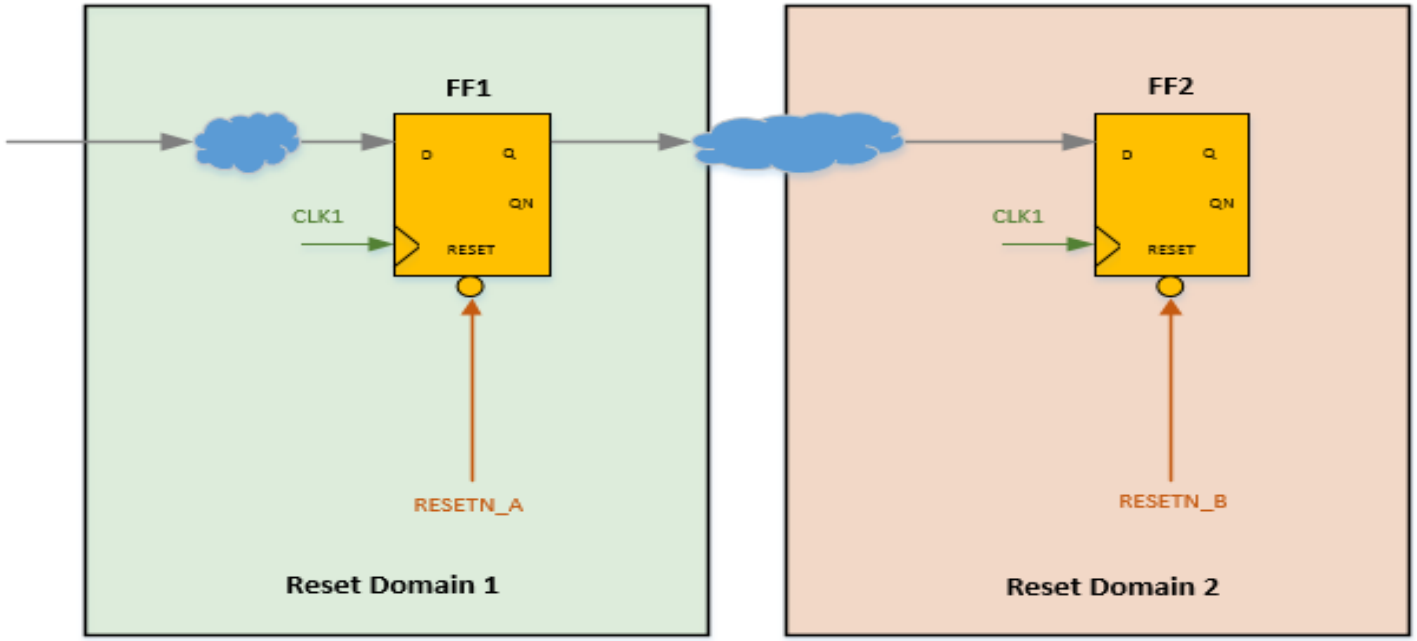
- Asynchronous paths are <1% of typical designs
- Significant RISK of Metastability/Randomness bugs
- Verification requires many specialized flows beyond just structural CDC



Source: www.realintent.com

What is RDC Analysis?

Even one Asynchronous reset in the design can cause RDC problems

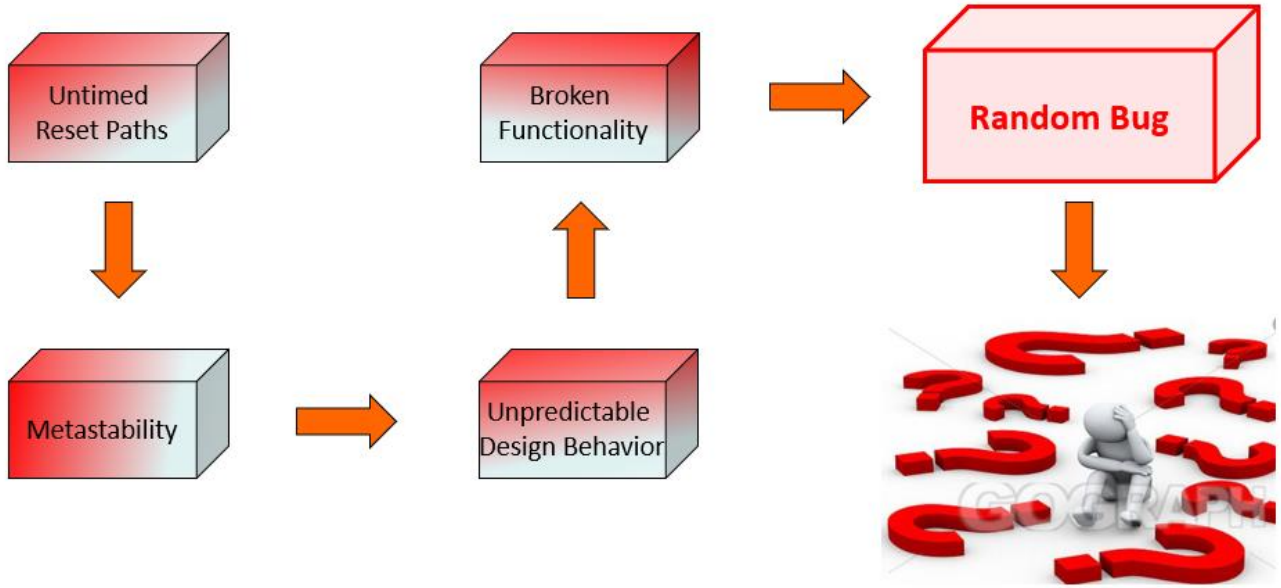


To reset only the faulty logic, several localized reset domains needed e. g. Automotive

Reset Domain : Part of the design that can be reset independently of other such parts of the design (other reset domains).

Why is Reset Domain Verification Needed?

- RDC issues are less likely, but they do occur
- RDC issues are extremely difficult to debug in silicon
- Multiple reset types and their interactions multiply risk



Why RDC?

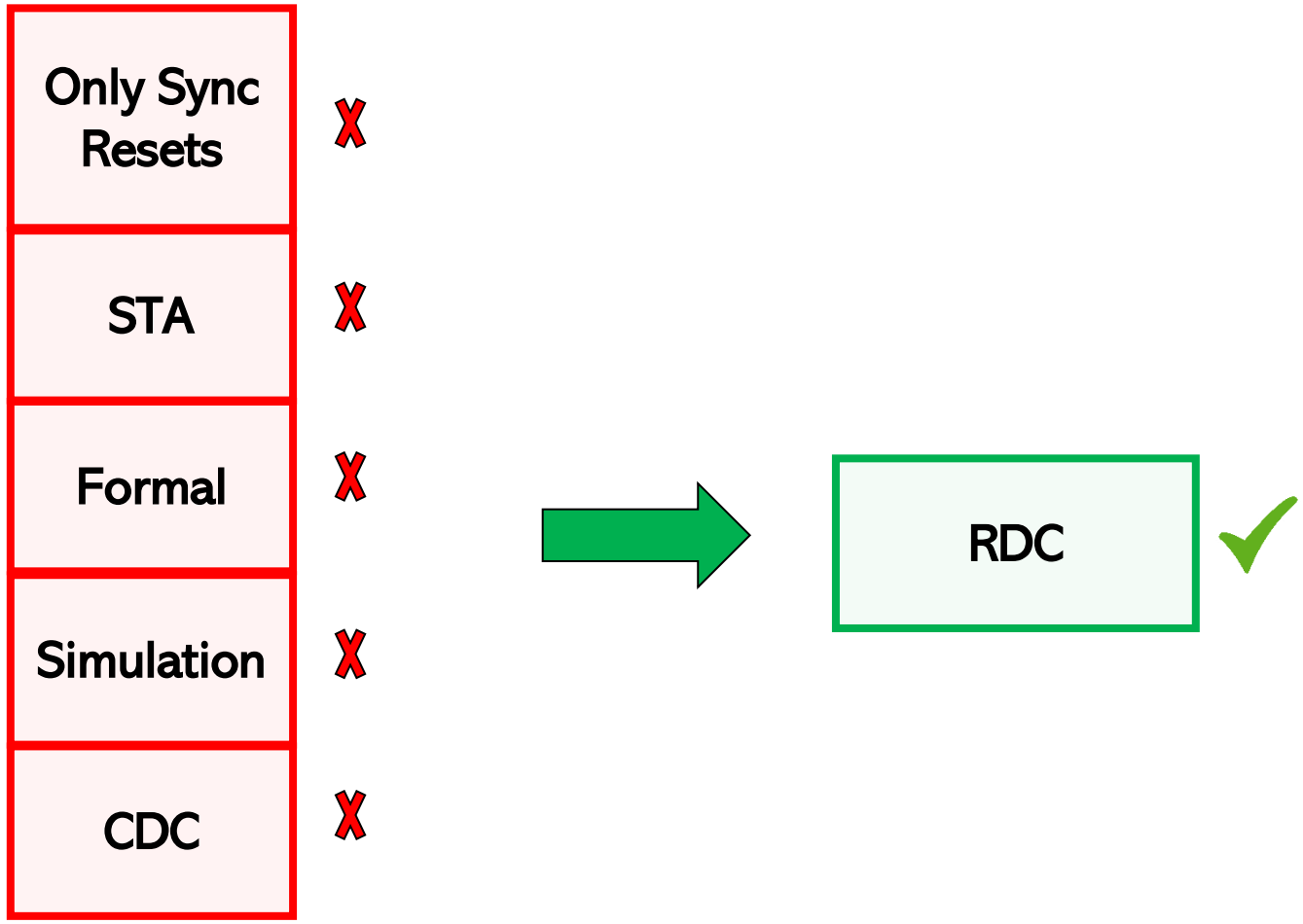
- Why RDC Signoff is important now?
  - Number of software resets increasing, possible some parts of design under reset while some parts in functional operations
  - Different power domains need different resets
- Isn’t reset controller logic already designed so these problems don’t occur
  - Yes but none of the flows STA, functional verification catch for these specific issues to ensure sign-off. Using RDC tool is only reliable way to safely verify reset logic is designed to ensure without metastability issues

Why RDC?

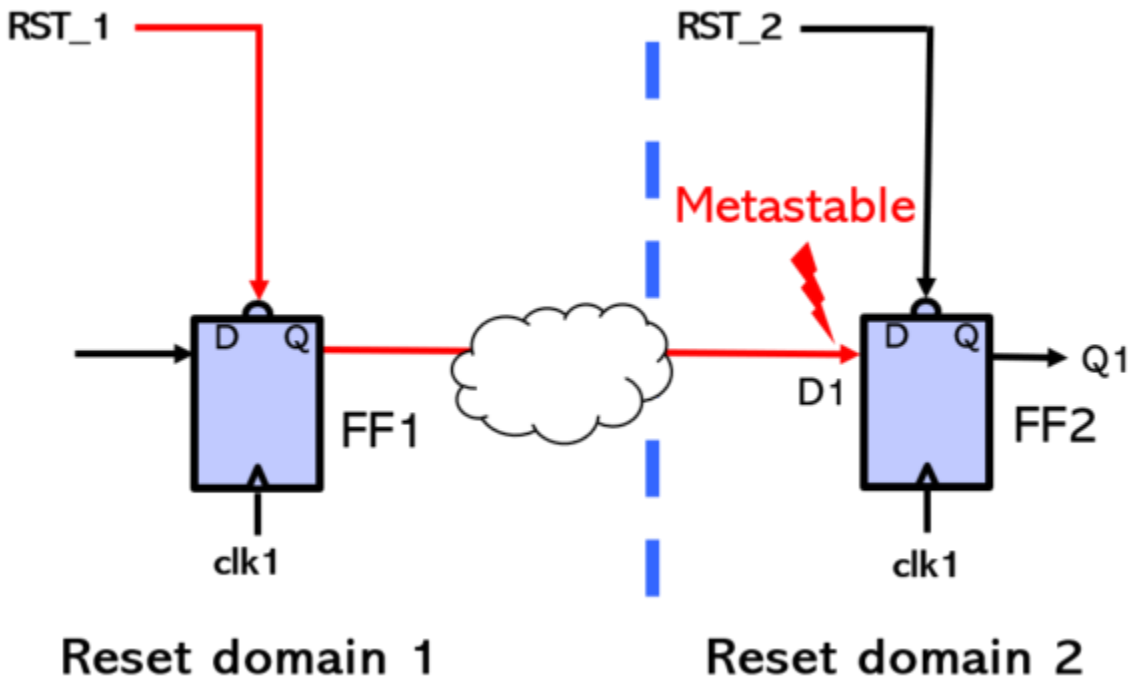
- Does CDC tool cover RDC
  - No, CDC tool looks at asynchronous paths. RDC failures can occur between synchronous clock domains also and are not same as CDC failures.
- Why we have not seen failures yet?
  - Unlike CDC frequency of reset operation is much less than clocks
  - The reset effect has to propagate through functional logic and is dependent upon state
  - Depends on actual delays, so may not always show as error but intermittent failures in some chips (lower yield)
- RDC issues HAVE led to chip failures in multiple design houses!

No substitutes for RDC Analysis

Specific solution needed to pinpoint unsafe paths

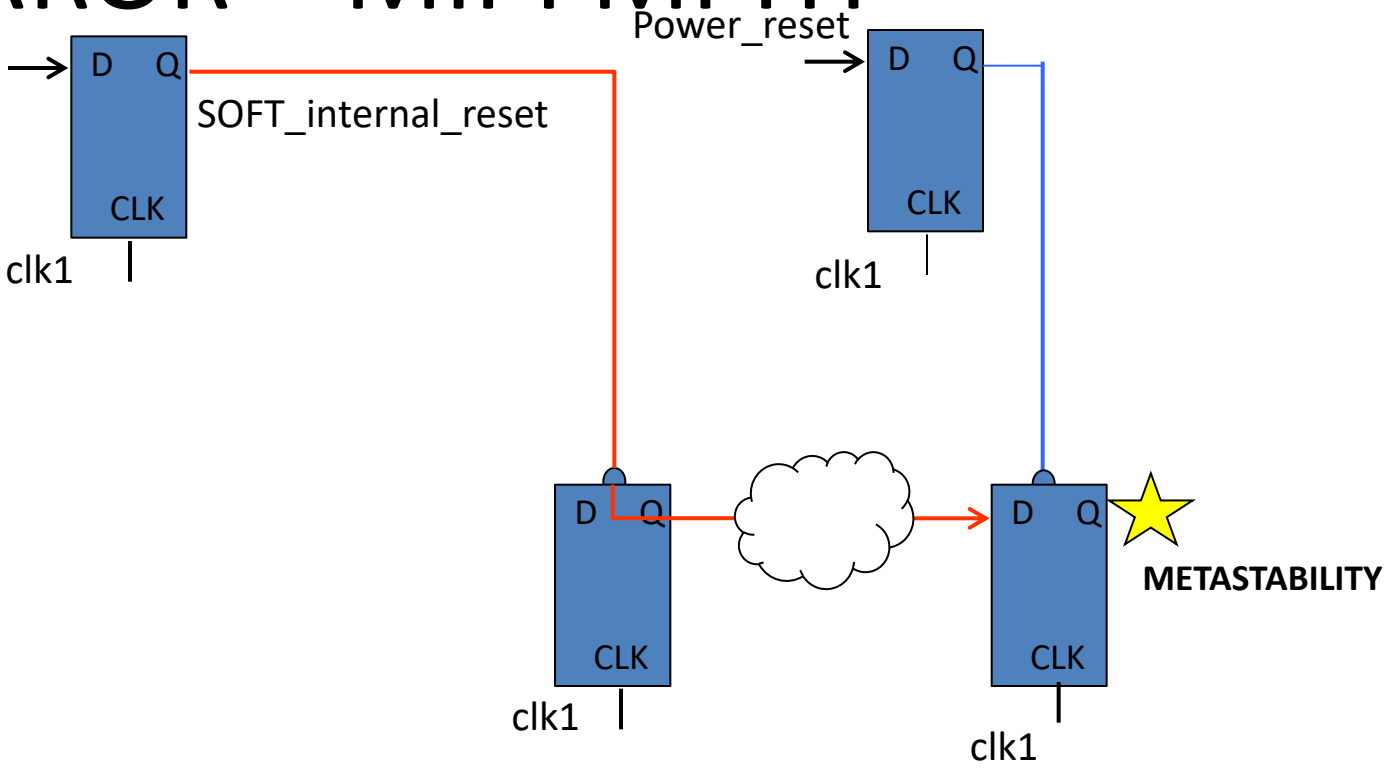


RDC ERROR – Mobile SoC



Problem	RST_1 assertion creates an ***untimed*** path. Fault **might** be detectable during gate-level sims but no guarantee

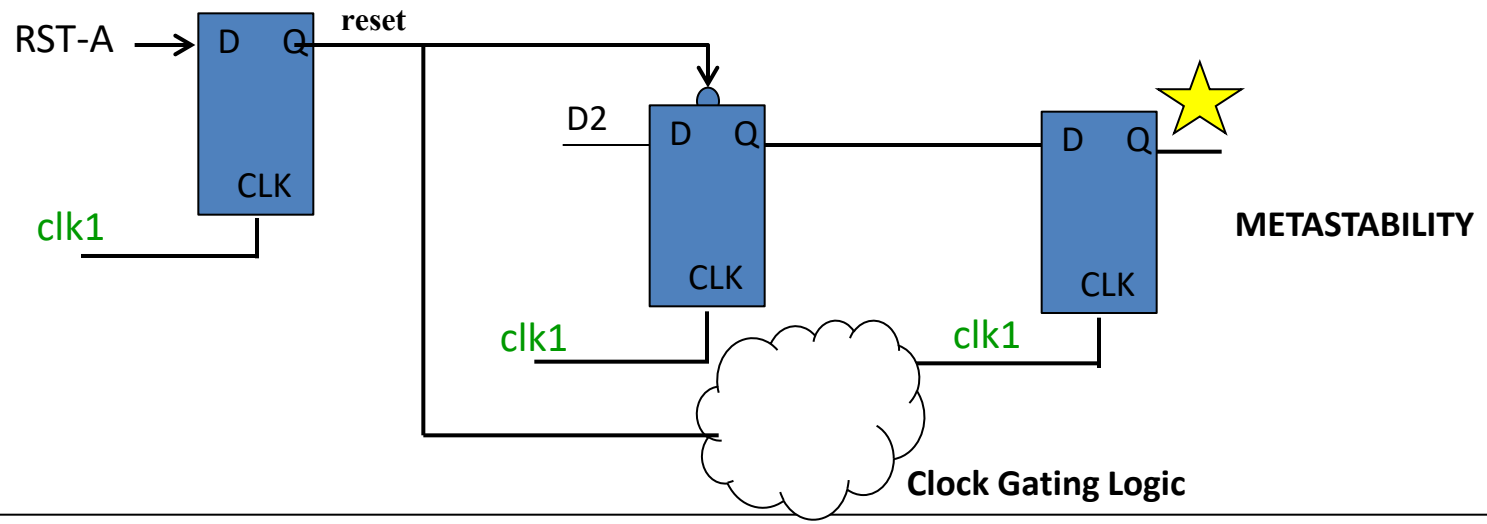
RDC ERROR – MIPI MPHY



Problem	SOFT_internal_reset assertion while Power_reset is de-asserted assertion creates an ***untimed*** path. Fault **might** be detectable during gate-level sims but no guarantee

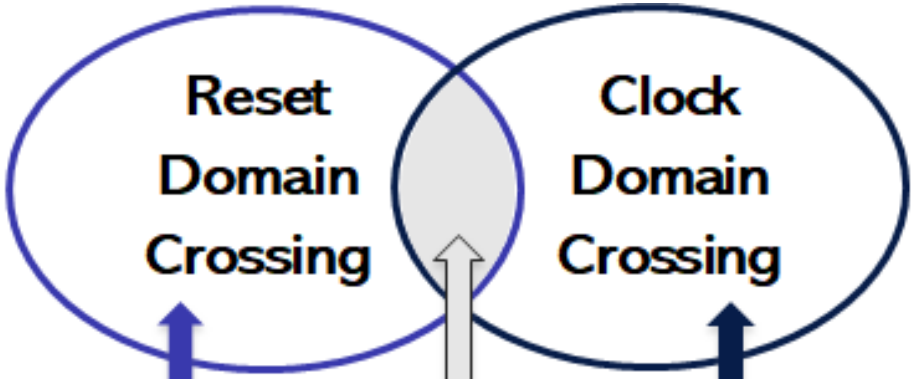
RDC ERROR – Modem IP

- Design assumption is to have clock gated at MFs prior metastability can pass to MF.
- Simulation tool did not fail, as well as seen that clock gated on the time.
- Simulation “ticks” based and depends on internal events , so there is a race between signals and in simulation gate closed “tick”



Problem reset assertion creates an ***untimed*** path.
Fault **might** be detectable during gate-level sims but no guarantee

CDC vs RDC



a. RDC errors - can occur in same clock domain

b. Analysis Scope - Global

c. RDC-specific analysis required to identify all RDC issues with low noise

d. Mean Time Between Failures - High

Sign-off uses static analysis

Can cause metastability, glitches & functional correlation errors

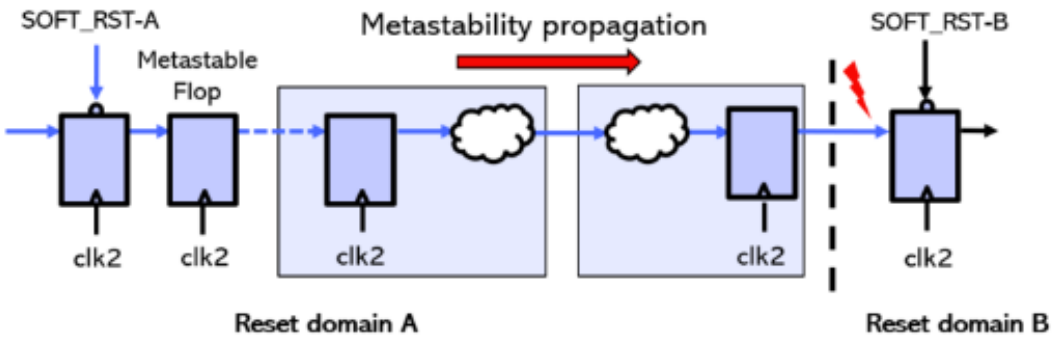
a. CDC errors - occur across clock domains

b. Analysis Scope - Local at CDC interfaces

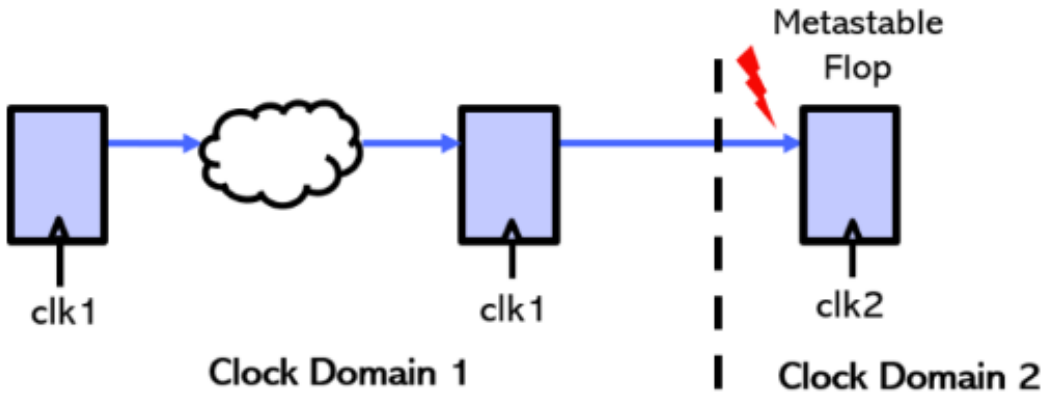
c. CDC-specific analysis required to identify all CDC issues with low noise

d. Mean Time Between Failures - Low

RDC Needs Global Analysis



Global analysis required to find all RDC issues

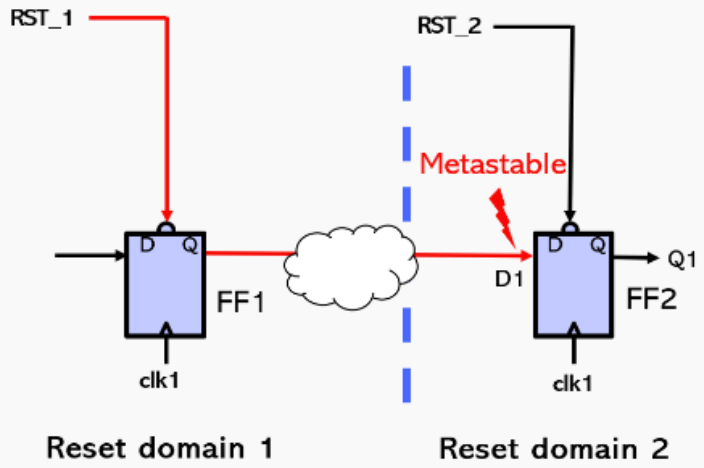


Local analysis (CDC interfaces) required to find all CDC issues

RDC Needs **Automatic Advanced** Analysis

Structural analysis only

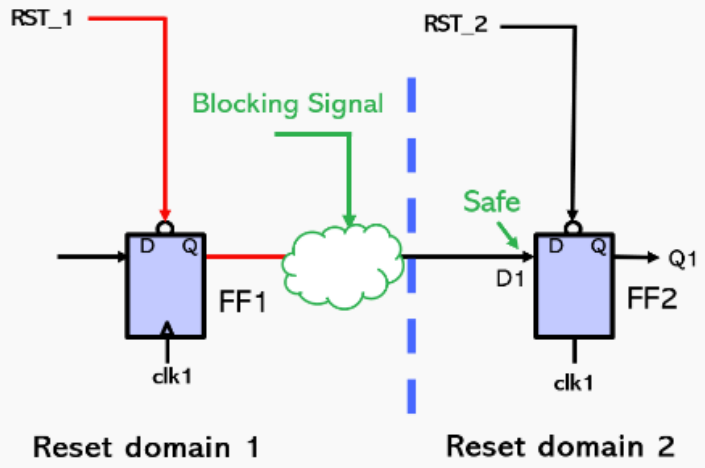
False positive error



Analysis shows the assertion of RST_1 creates an untimed path from FF1 to FF2, which can cause metastability when RST_1 asserts and RST_2 to FF2 is de-asserted.

Advanced RDC functional analysis

Identifies protection



Running RDC-specific functional analysis of the combinational logic shows there is a blocking signal. The reset domain crossing path is proven safe, and no false error is reported.

RDC Sign-off With Meridian RDC

Shift Left

Static Sign-Off

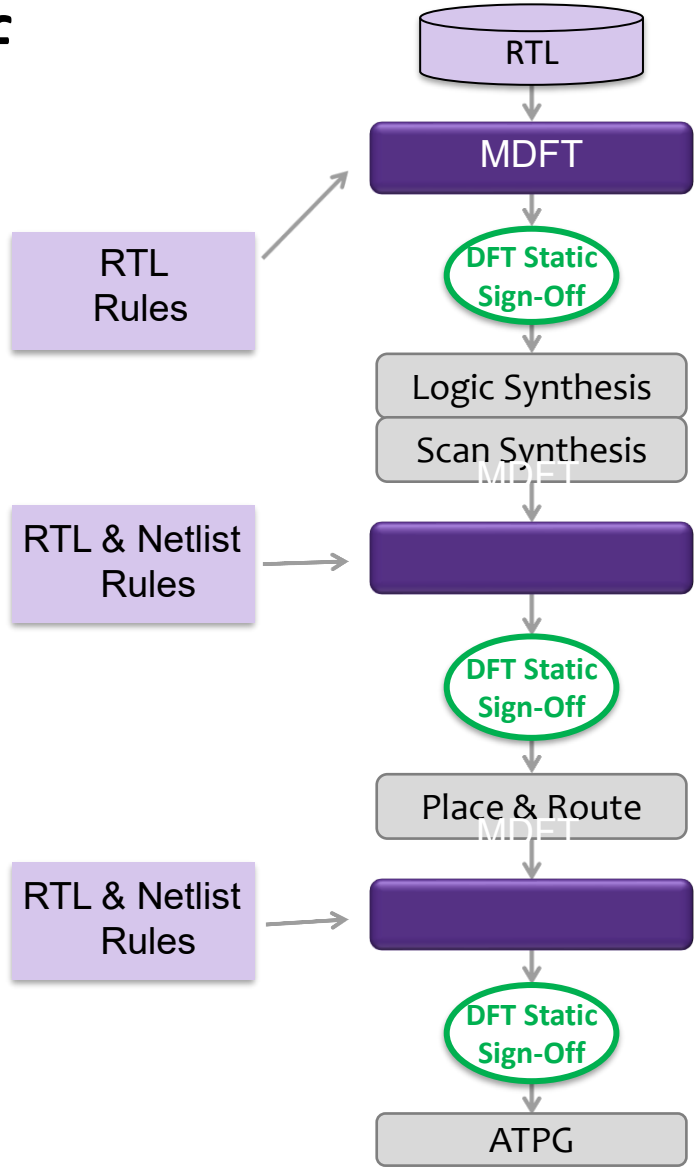
DFT COMPLIANCE, CHECKING AND ENABLING SHIFT LEFT USING STATIC SIGN-OFF

DFT Challenges and Trends

- Shift Left
  - ATPG typically occurs after P&R – but fixes are 10X more expensive at each stage
  - ATPG attempted before P&R, but overkill

Shifting Left with DFT Static Sign-off

Design RTL	Prepare for scan synthesis, to ensure RTL is scan friendly
Gate-level Netlist with Scan Chains	To verify correctness of scan implementation
Netlist with Scan Chains Reordered	To ensure scan reordering does not create issues



Approach to DFT Static Sign-off

- Multimode
  - Multiple sets of rules per run, reducing setup time and speeding up runtime
    - Multiple ATPG partitions, multiple sets of constraints per partition
- High capacity and performance
  - Multi-million gate design in minutes
  - Low peak memory footprint
- Specialized, fine-grained rules
  - High coverage at all design stages
  - Faster debug and root cause analysis
- Low noise
  - To minimize false positives and error duplication
- Fits easily with existing flows and DFT/ATPG tools

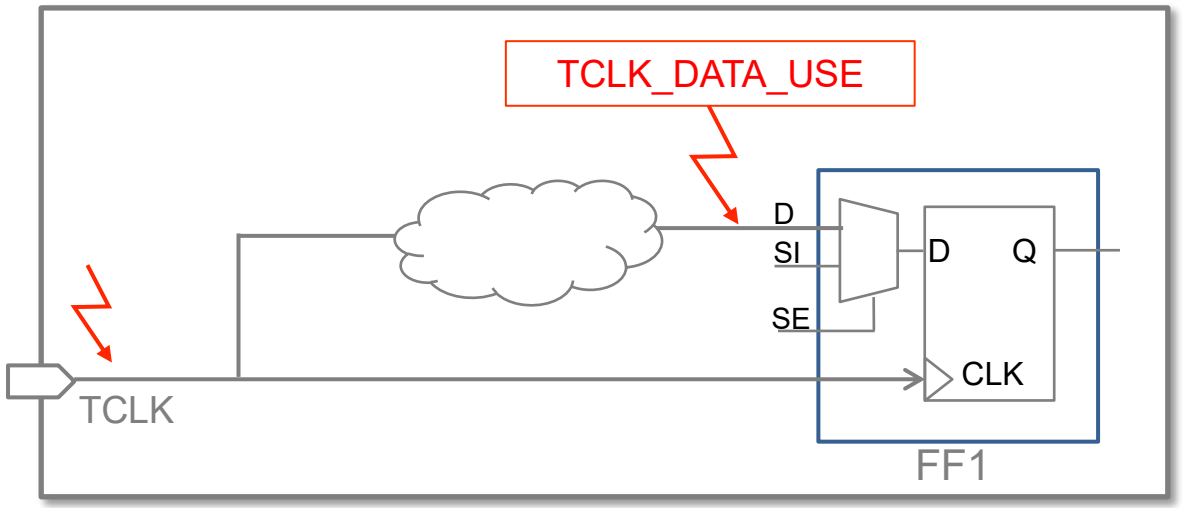
Design issues that affect Fault Coverage

- Uncontrollable test clocks
- Uncontrollable/incorrectly constrained async set/reset pins of flip-flops
- Loss of connectivity/controllability between signals, due to –
  - Design bugs, such as undriven/unloaded nets, combinational feedback loops, and tristate busses with potential for bus contention
  - Specification errors in test mode constraints, such as incorrect or insufficient test mode constants

TCLK_DATA_USE (Category: CLOCK)

Test Clock Drives Data Input of FF in Scan Hierarchy

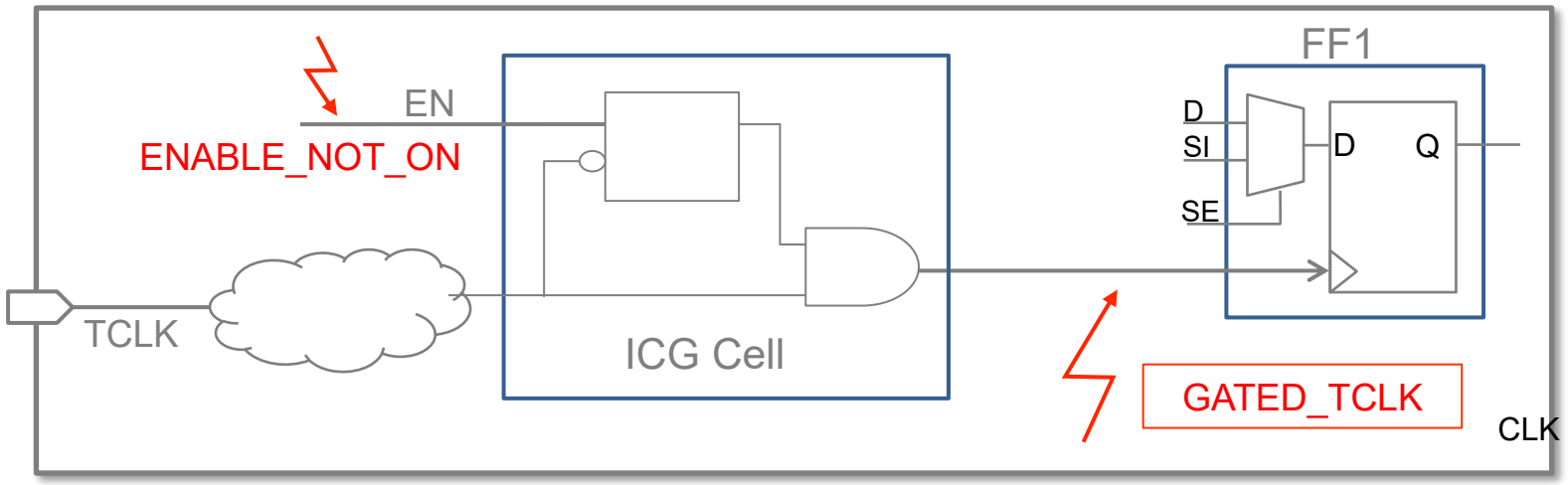
- Cause:
  - Test clock drives data or async set/reset inputs of FFs (e.g., FF1) in a scan hierarchy (RTL/Gate)
- Impact:
  - Metastability during capture, since data transitions happen at or close to the clock edge, hence affected flip-flop data can get corrupted, causing loss of both controllability and observability, hence lower fault coverage



GATED_TCLK (Category: CLOCK)

Test Clock Not Enabled During Scan Shift

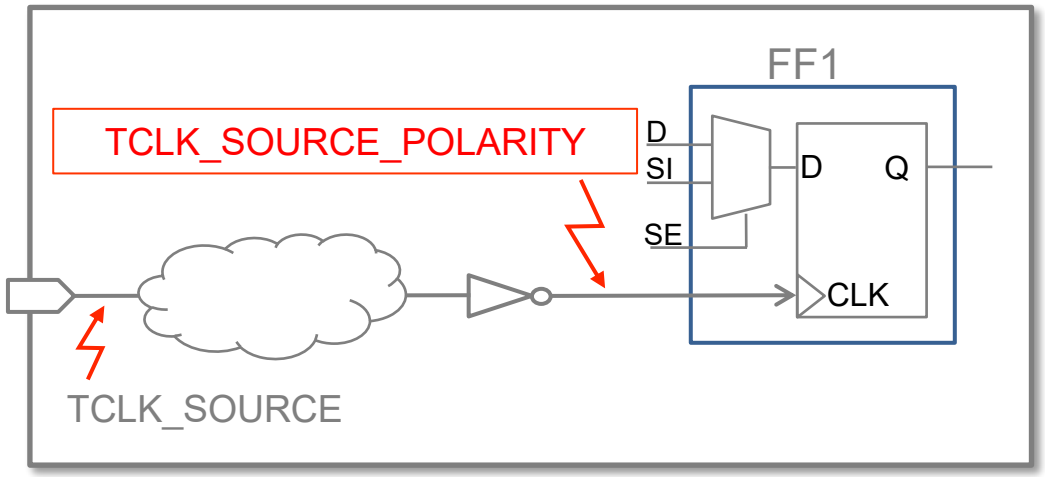
- Cause:
  - Gated test clock is not enabled during scan shift (RTL/Gate)
- Impact:
  - Failure to load scan chain during shift, causing loss of controllability (observability) of the affected flip-flop during scan load (unload), lowering fault coverage



TCLK_SOURCE_POLARITY (Category: CLOCK)

Test Clock & Test Source Clock Have Different Polarities

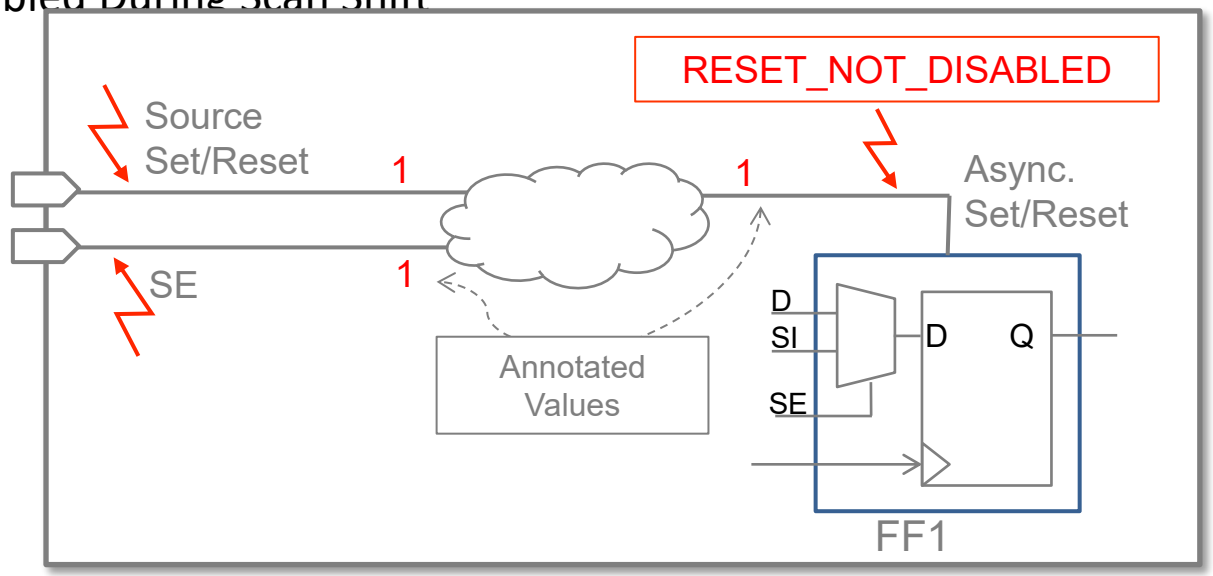
- Cause:
  - Test clock (e.g., CLK in FF1) does not have the same polarity as the test source clock (RTL/Gate)
- Impact:
  - Non-controllable test clock during scan test, causing loss of fault coverage



RESET_NOT_DISABLED (Category: ASYNC_RESET)

Async. Set/Reset Not Disabled During Scan Shift

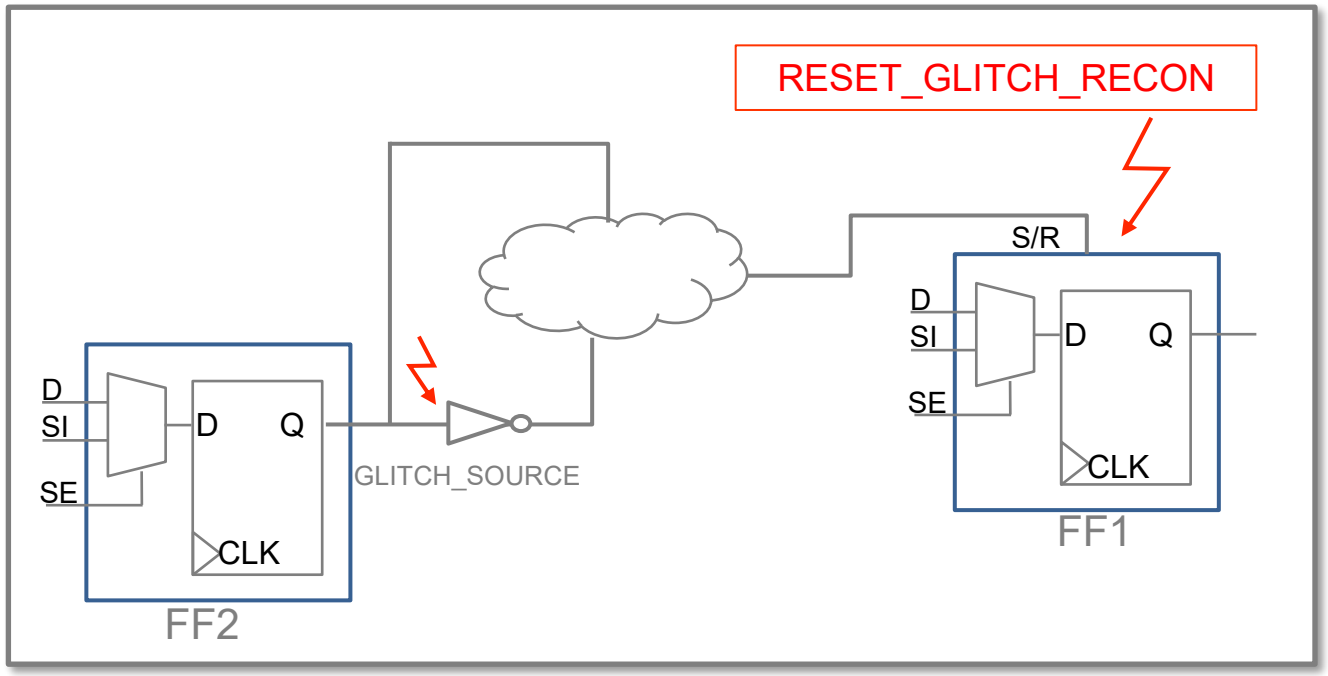
- Cause:
  - In shift mode, there is at least one path to propagate asserted value (or X) from source Set/Reset to FF Set/Reset pin, where FF belongs to the scan hierarchy (RTL/Gate)
- Impact:
  - Scan load/unload Shift data erased by non-disabled asynchronous set/reset signal, causing loss of fault coverage



RESET_GLITCH_RECON (Category: ASYNC_RESET)

Reconvergence of Set/Reset with Opposite Polarity

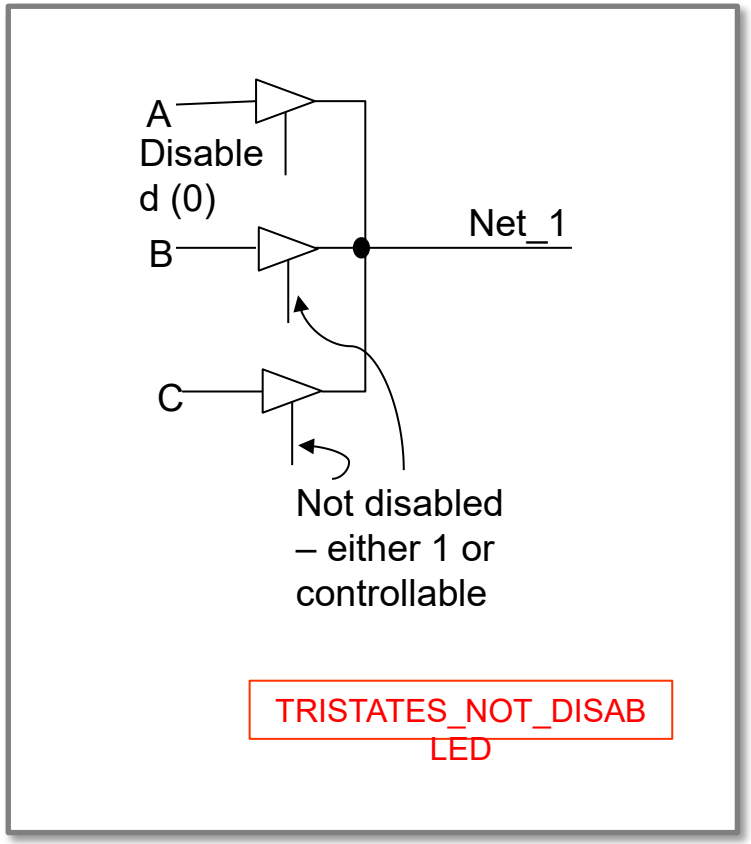
- Cause:
  - Signal re-converges with itself with opposite polarity and drives the set/reset pin of FF (e.g., FF1) in a scan hierarchy (RTL/Gate)
    - Self-loop between the output of FF2 to its D-input can be either ignored or checked; if ignored, then the flip-flop is considered as a glitch source, otherwise it is not.
- Impact:
  - Glitchy set/reset, FF not able to shift and/or capture test data, causing loss of fault coverage



TRISTATES_NOT_DISABLED (Category: CONNECTIVITY)

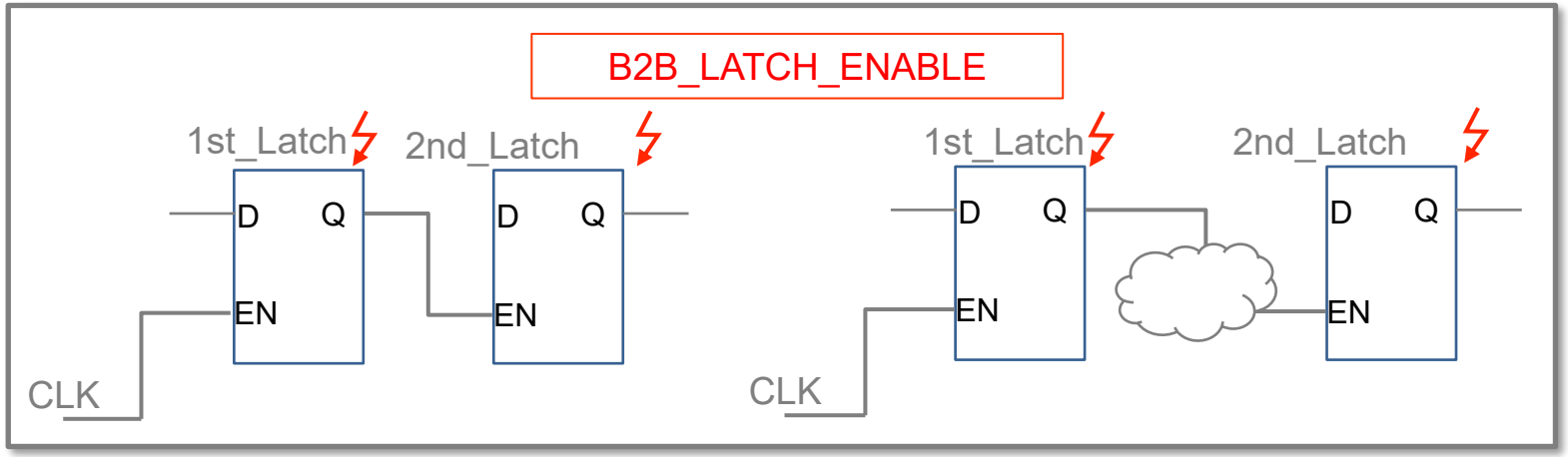
A net has multiple non-disabled tristate drivers

- Cause:
  - A design has >1 non-disabled tristate drivers. (RTL/Gate)
- Impact:
  - Loss of fault coverage due to unreliable shift and/or capture



B2B_LATCH_ENABLE (Category: SCAN_CHAIN)

- Cause:
  - Two latches are connected back-to-back with Q of one latch driving EN of the next latch. (Gate)
- Impact:
  - Test data loss while data is being shifted through scan chain
  - Timing problems



DFT Sign-off With Meridian DFT

Shift Left

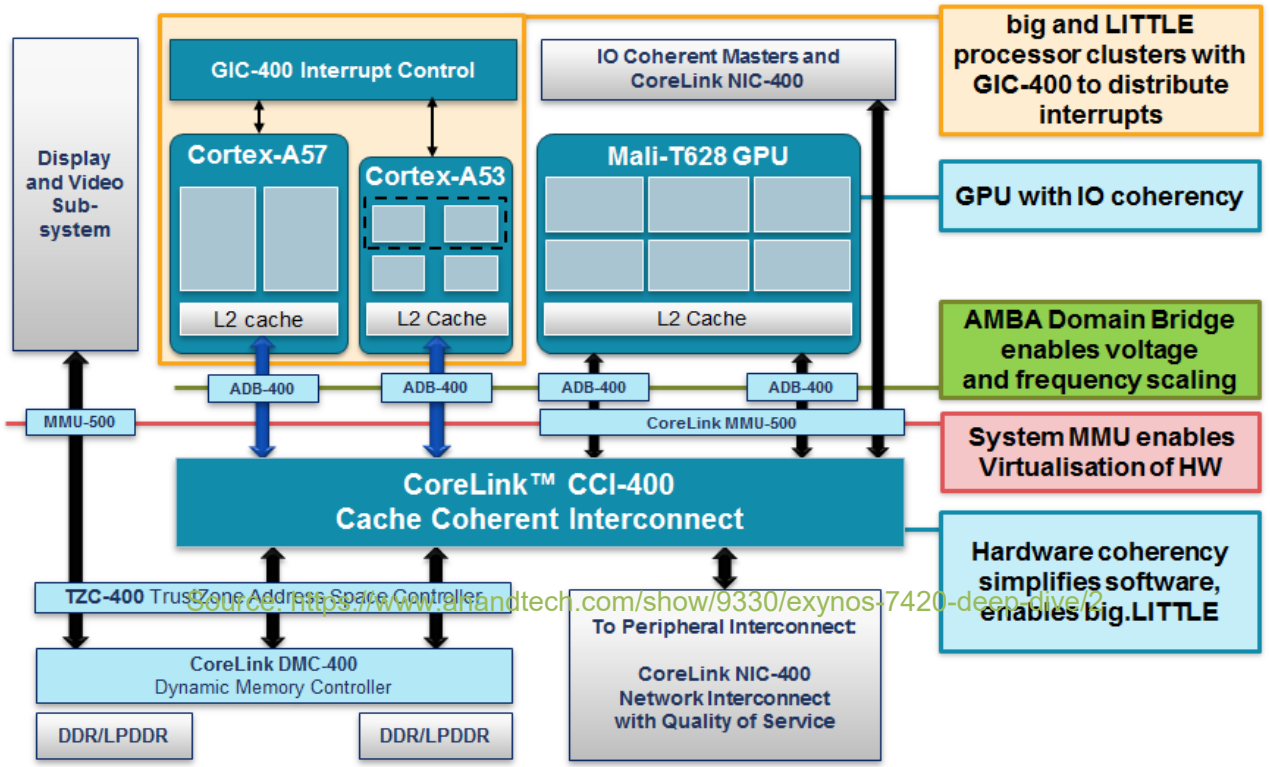
Static Sign-Off

ARCHITECTURAL COMPLIANCE USING STATIC SIGN-OFF

Structured Design Methodologies Manage Complexity

Architecture implemented using functional components within a structure

Connectivity rules specify architectural component interconnections



Structured Design Methodologies Manage Complexity

Architecture implemented using functional components within a structure

Connectivity rules specify architectural component interconnections

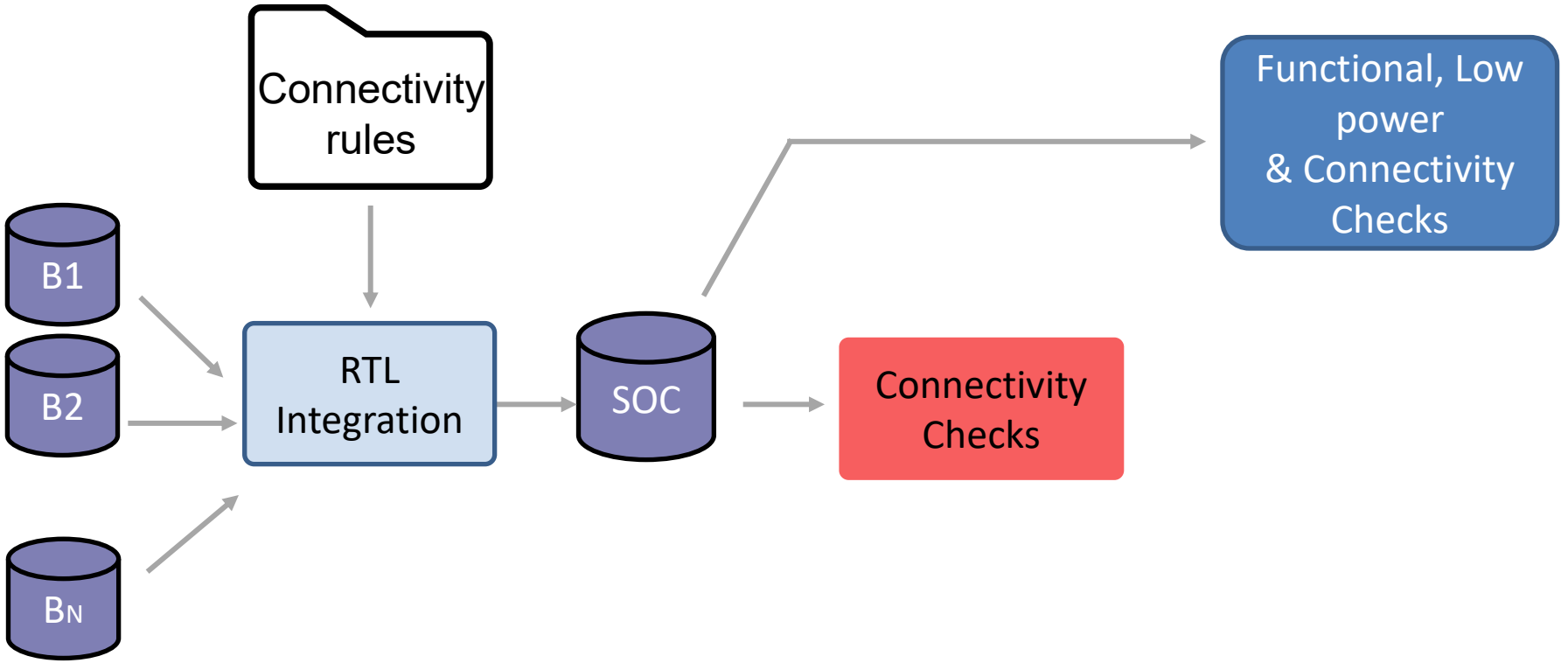
- Bus protocols, power, debug logic, PD constraints, memory, DFT connectivity…
- Facilitates automatic design integration

Shift Left: Earliest possible efficient verification of a design step

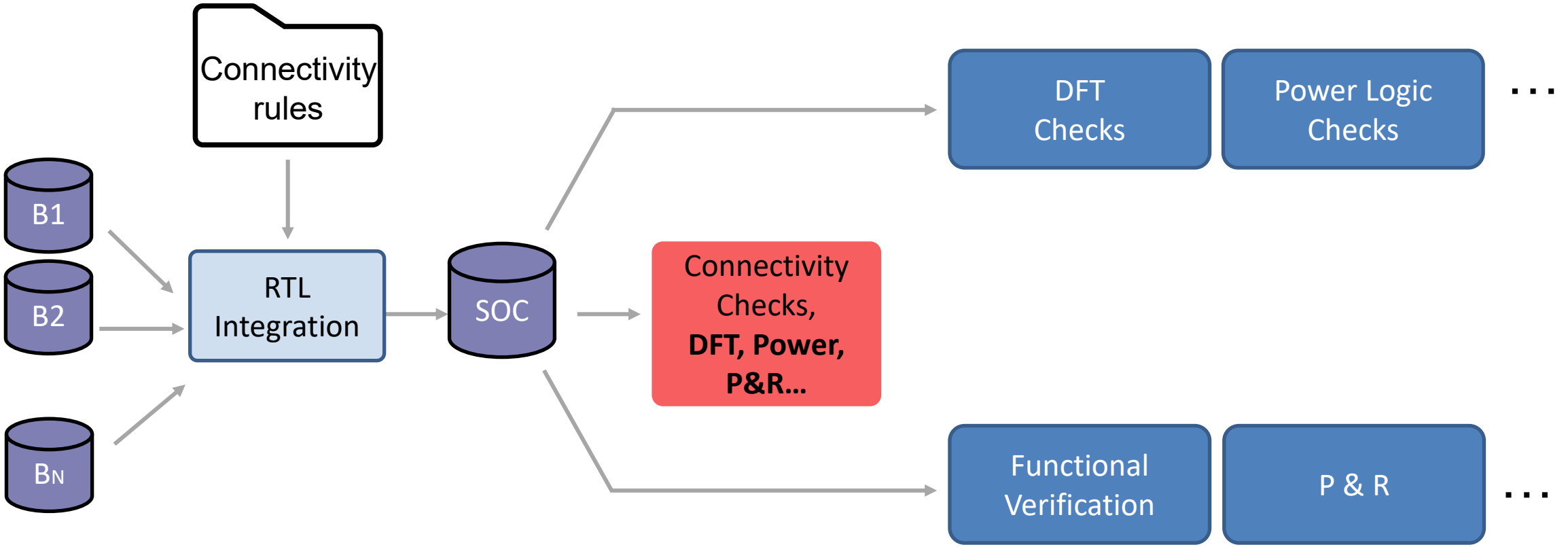
Connectivity Checking: Efficient shift left verification of architecture compliance

CONNECTIVITY CHECKING USING STATIC-SIGNOFF

Connectivity Checking Shift Left In Action

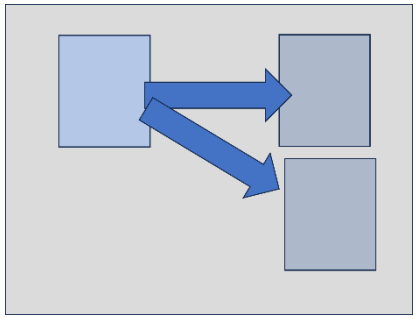


Connectivity Checking Shift Left In Action

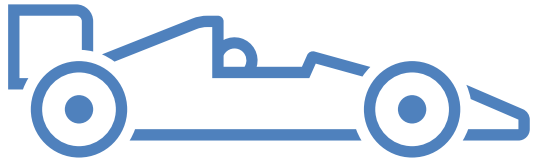


Static Connectivity Checking Requirements

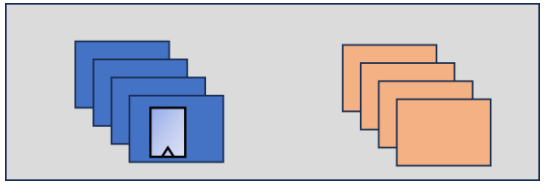
Diverse and Granular Checks



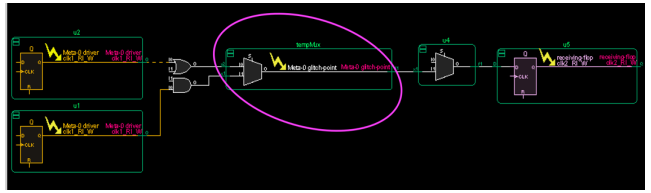
Full Chip Capacity



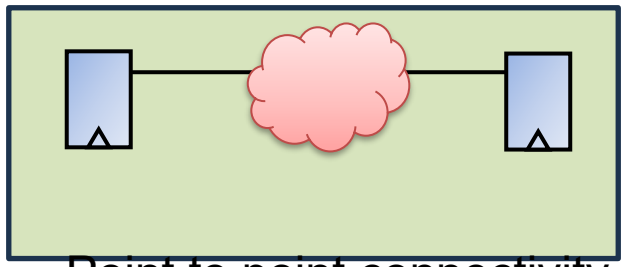
Ease of Specification



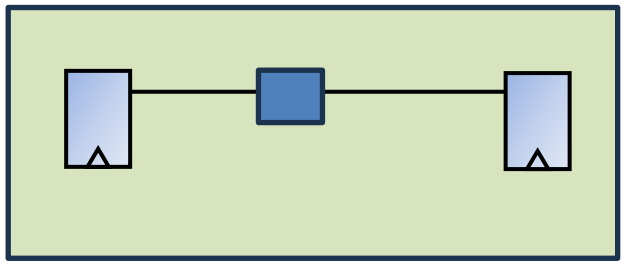
Ease of Debug



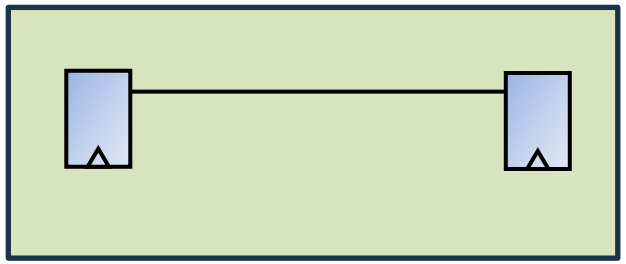
Diverse and Granular Checks



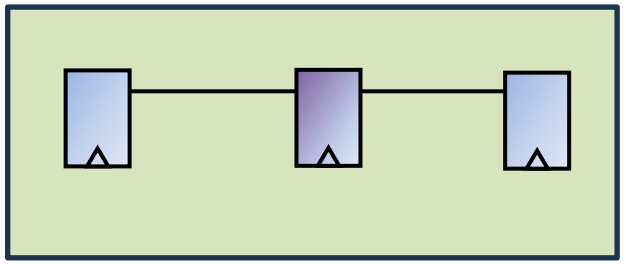
Point to point connectivity



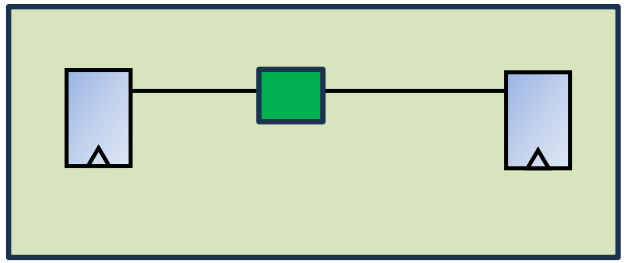
Point to point connectivity **through** objects



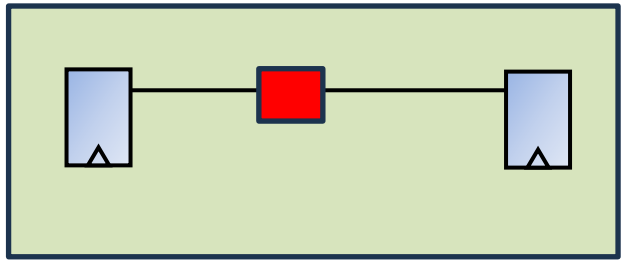
Point to point **direct** connectivity



Point to point
Connectivity through transparent
Sequential or modules/instances



Point to point connectivity **allowed** objects



Point to point connectivity **disallow** objects

Ease of specification - Example

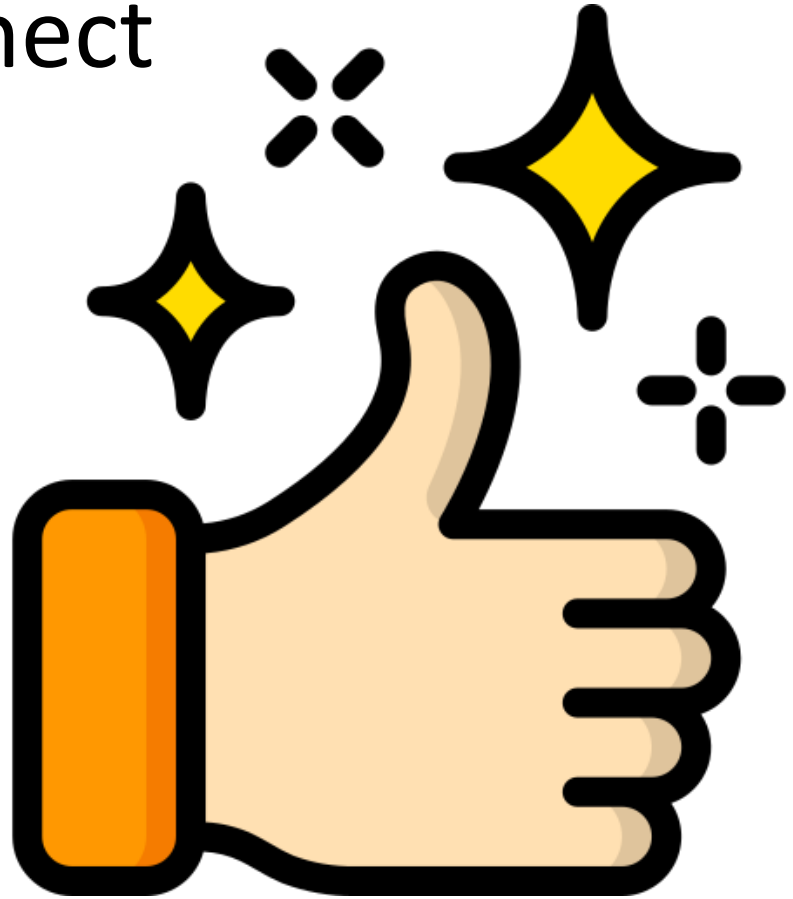
- Just 3 commands can define a check at block or full chip level to avoid improper connectivity from non-retention reset to retention FFs
- Rule defining check can be easily enabled or disabled using **enable_rules / disable_rules** commands

```
create_group  –name
Non_Retention_Reset_Sub_Group_1
  –scope {I1  I2.I3 } –signals  { RST_* }
  –module Non_Ret_Mod –exclude_instances {u_inst1.u_inst2.*}

create_group  –name  Retention_FF_Sub_Group_1
  –scope { I4* }   –FF { retention }

set_not_connect
  –from_group Non_Retention_Reset_Sub_Group_1
  –to_group Retention_FF_Sub_Group_1
  –dont_trace_group  { CLAMP_CELL_instance_group }
  –rule NONRET_RST_TO_RET_FF
```

Architectural and Connectivity Sign-off With Meridian Safeconnect

Shift Left

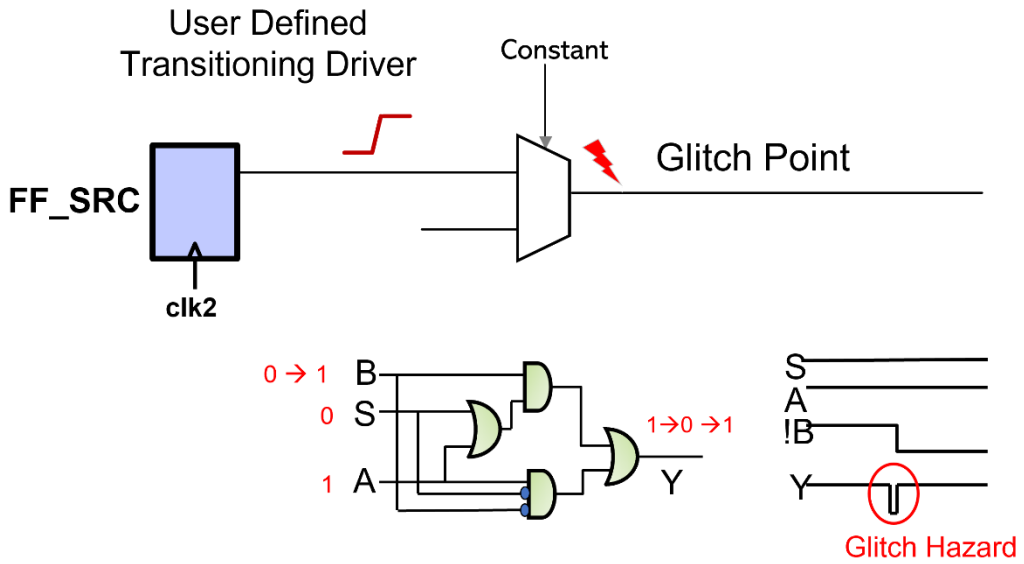
Static Sign-Off



GLITCH CHECKING METHODOLOGY USING STATIC-SIGNOFF

Glitches Fail Chips

- Glitch = transition shorter than signal's clock period
- Async Interactions within designs become vulnerable to glitches at netlist
- Not caught by STA & functional simulation



Structured Glitch Verification Methodology

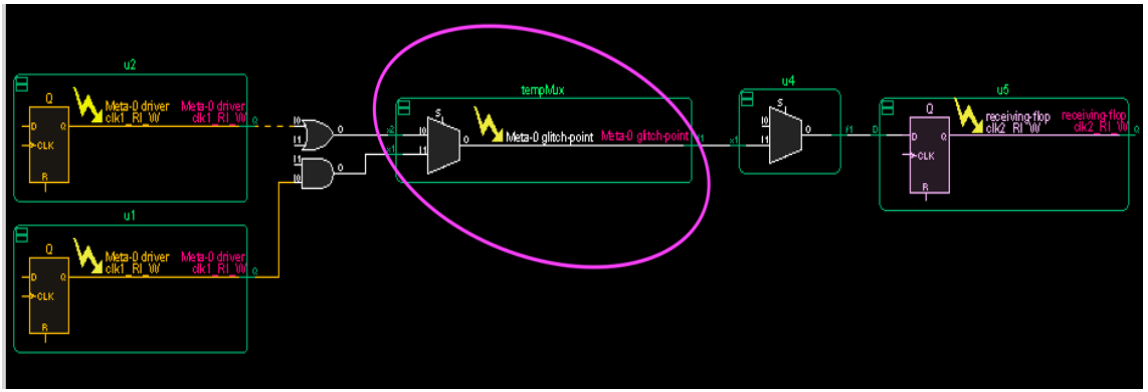
- CDC glitch failures encapsulated within interfaces
  - Using pre-verified components for CDC interfaces is standard practice
  - Structural glitch avoidance principles can be included within components

Structured Glitch Verification Methodology

- CDC glitch failures encapsulated within interfaces
  - Using pre-verified components for CDC interfaces is standard practice
  - Structural glitch avoidance principles can be included within components
- For timing exceptions & power management logic, path structuring may be necessary with synthesis tool restrictions for path during optimization.
  - Insert “do not touch” component to partition paths
  - Verify connectivity in netlist
  - Verify glitch in netlist

Glitch sign-off – IP level, Chip level

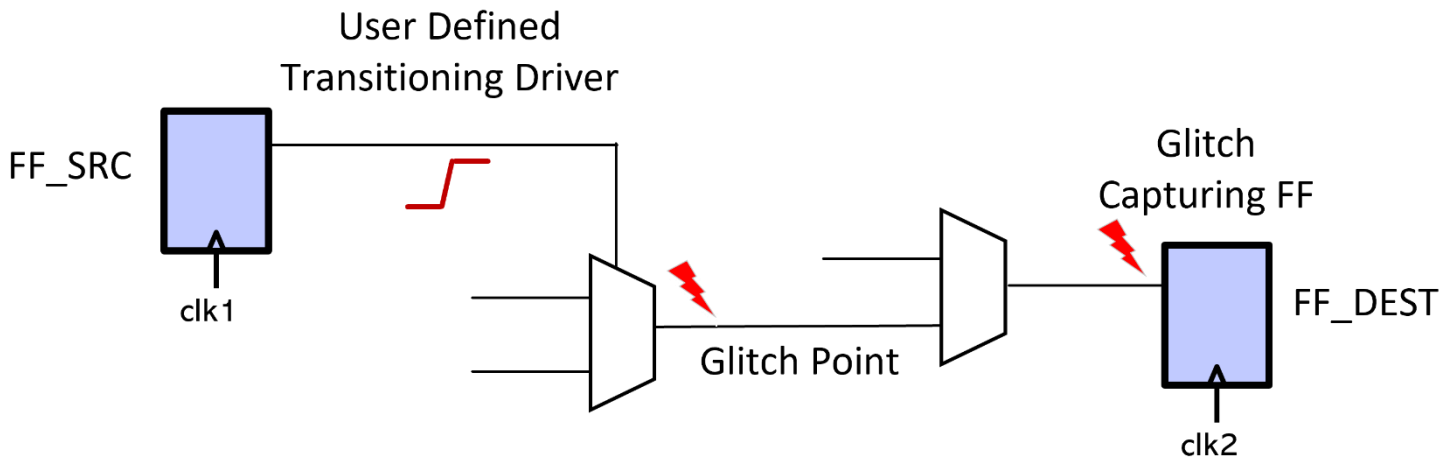
- In Async paths, Glitch can be fatal
  - False Paths, Multi-Cycle Paths, Clamps, Global Signals, Power control signals
  - Logic reordering, restructuring, retiming, optimization
- Numerous companies had *late-stage* netlist-glitch failures
  - IP vendor provided glitchy-IP (@outputs) to customer
  - Automotive chip had glitch-potential, designers were unaware
  - Memory-controller chip went through multiple ECOs because of glitch failures



Glitch Detected on path to Analog IP

Example Glitch Report

- Source to destination path should not be glitchy
- Check can be easily specified



- Glitch source flop/signal, glitch point and glitch capturing flops reported

Glitch Sign-off With Meridian Safeconnect

Shift Left

Static Sign-Off

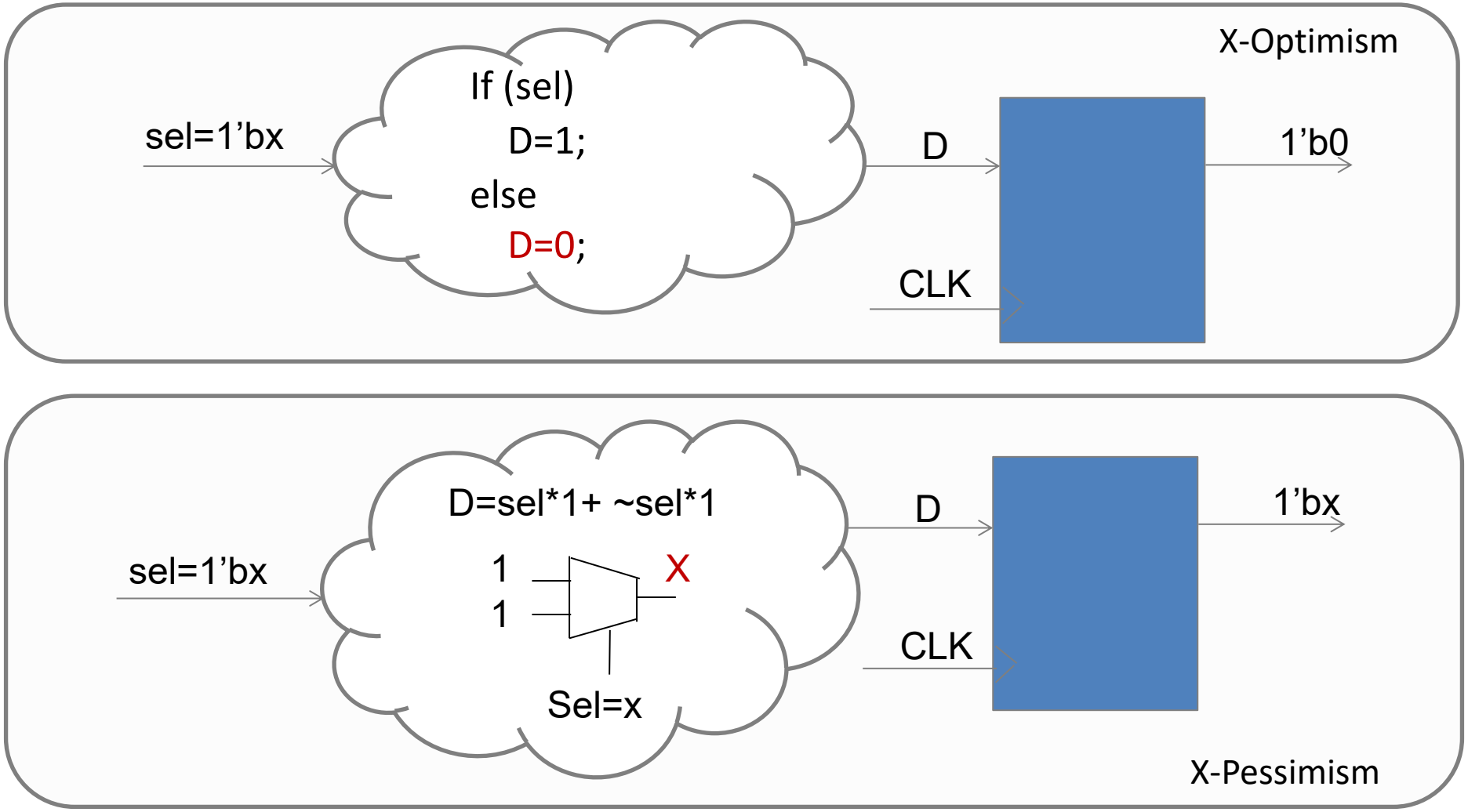
ADVANCED METHODOLOGY TO IDENTIFY X-INITIALIZATION SOURCE ERRORS AND FIX THEM TO PREVENT THE ERROR FROM PROPAGATING

What Are X Sources?

X Source : a flop or input port which is in unknown value at the end of a given reset scenario

Potential X Sources	
Uninitialized 4-state variables (Uninit)	Out-of-range bit-selects and array indices (OutOfRange)
Low power logic shutdown or power-up (NonRetention)	Logic gates with unknown output values (Explicit)
Unconnected module input ports (Undriven)	Setup or hold timing violations (Netlist only)
Multi-driver conflicts (Bus Contention)	User-assigned X values in hardware models (Explicit)
Operations with an unknown result (RAMs, FIFOs)	Testbench X injection (User)

X Propagation: Two Problematic Scenarios



The Impact of X Propagation in the Design

Uncontrolled Design Behavior

- Uninitialized and Reverted to X Flops
- Incorrect reset type and value
- Hardware security exposure

Sub-optimal Design Quality

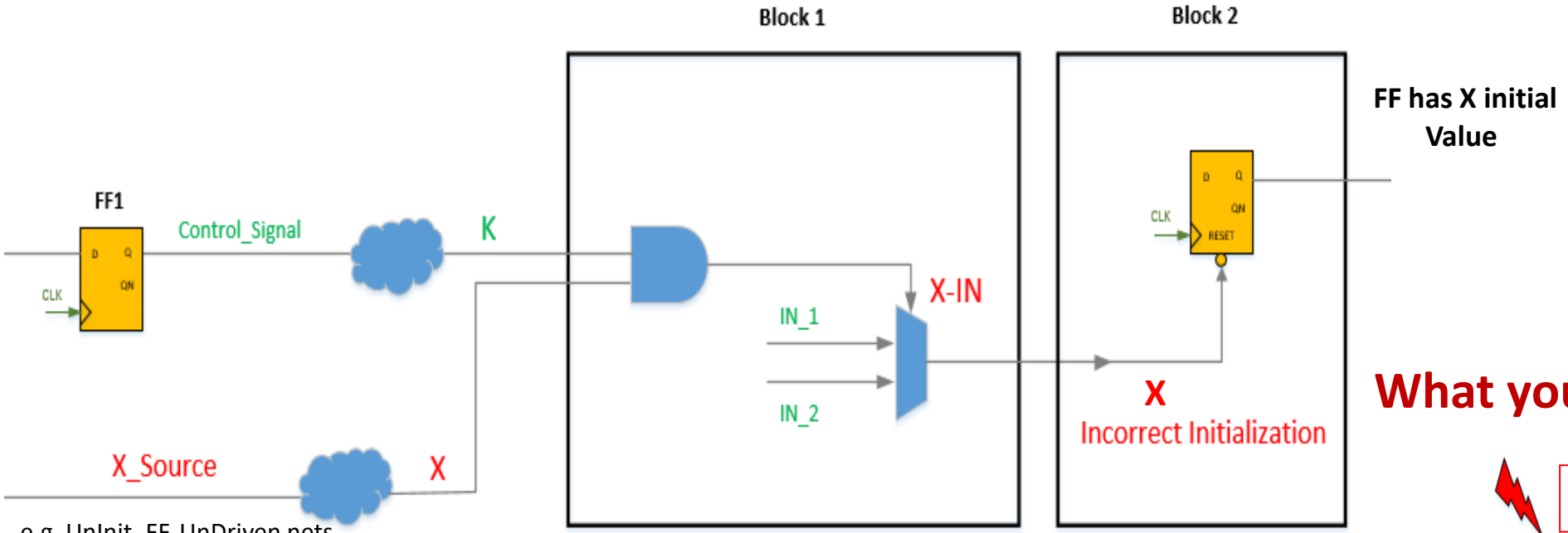
- Long initialization latency
- Inefficient reset routing
- Excessive initialization power

Breakdown of Validation Flow

- Incorrect simulation
- Incomplete simulation
- Inefficient gate-level debug

Ignoring the impact of design X values causes silicon failures.

Incorrect Initialization Causes Design Failure



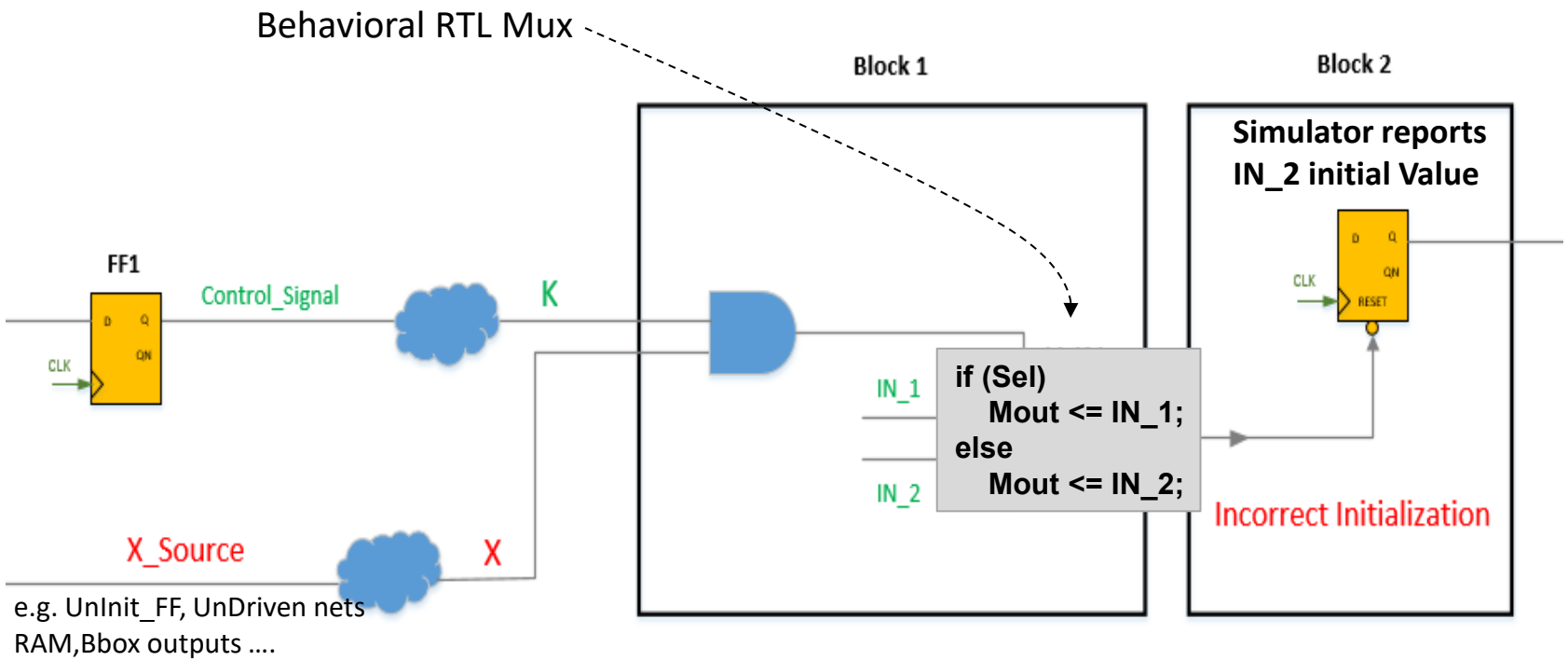
e.g. UnInit_FF, UnDriven nets
RAM,Bbox outputs ….

What you'd want to know:

- X-Source
- X-Reset

- Functional non-determinism in a design is a failure
- Incorrect design initialization causes erroneous design behavior, sub-par design quality, security vulnerability

Simulation Can Mask X Issues Due to X-Optimism

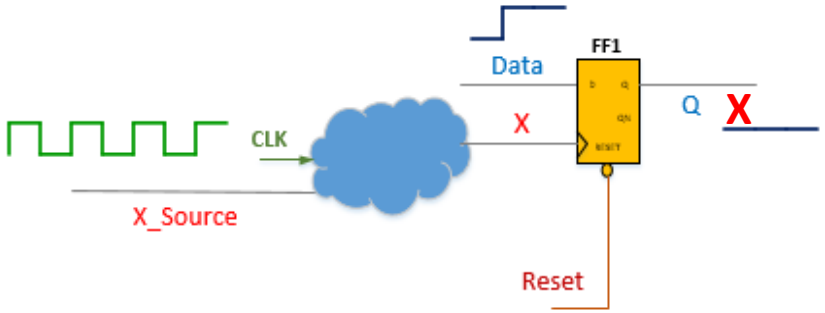


What you'd want to know:

- X-Source
- X-Reset
- X-Optimism

- Can not rely on simulation for correct initialization because simulator can mask X initialization, and propagates wrong value (X-Optimism)
- Must fix X-Sources before RTL simulation

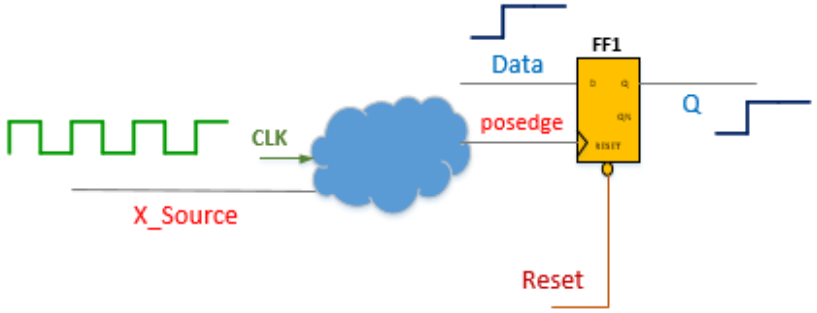
Simulation Can Miss Dangerous X on Clock



Initialization Problem

With Reset de-asserted, Clock 0->x should be treate
as **an undetermined edge**

=> FF1.Q should become **X**



Missed by Simulation

However, in Simulation, when Reset is de-asserted,
Clock 0->x is treated as a **posedge** (X-Optimism)

=> FF1.Q changes to 1

Static Sign-off Can Complete / Enhance Simulation-Only Flow

Simulation	Static Sign-off
Test bench coverage dependent	Comprehensive
Simulation needs mature RTL	Early
Simulation needs vectors	Low effort
Band-Aid for symptom, X remains in GLS	Addresses root cause
No optimization	Optimization potential

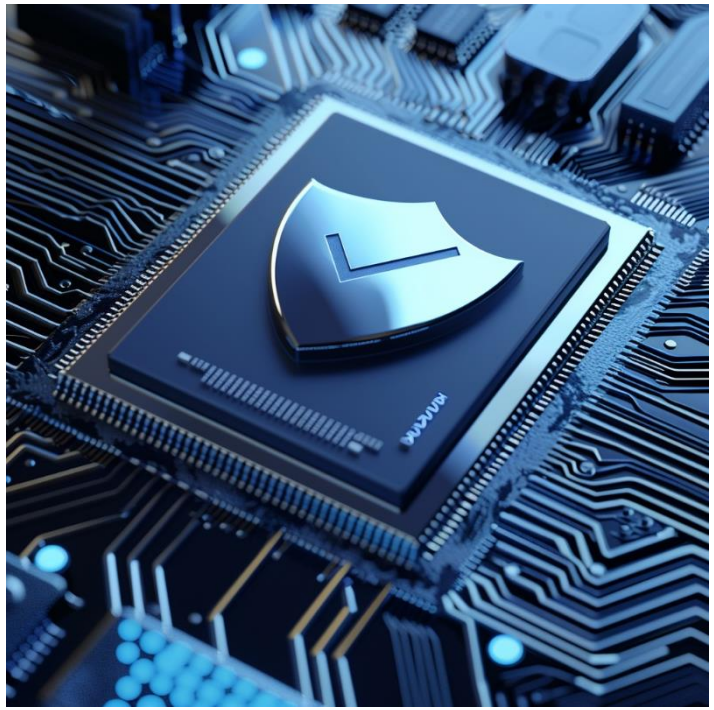
accellera SYSTEMS INITIATIVE™

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Reset and X Sign-off With RXV

Shift Left

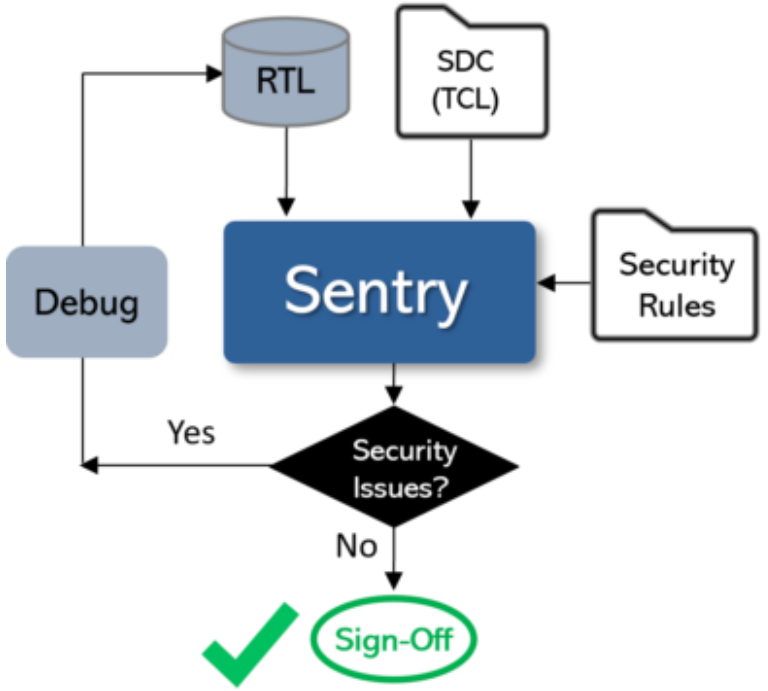
Static Sign-Off



Hardware Security Sign-Off

Sentry – Hardware Security Sign-Off

- Sentry is the industry's **fastest**, **highest capacity** hardware security **static sign-off** tool.
- Ensures the security and integrity of data on hardware devices.
- Ensures all paths adhere to stringent security protocols
- Protects designs against potential security vulnerabilities that could allow malicious code to be executed in the hardware.



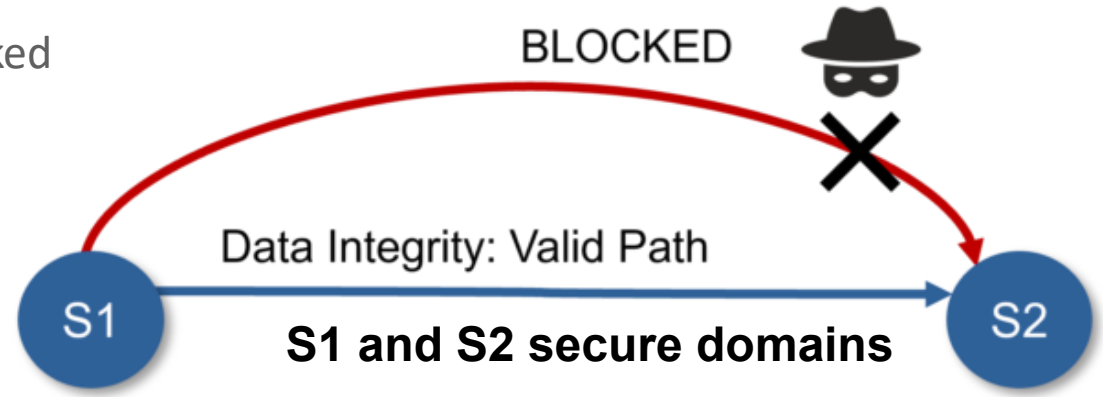
Sentry – Meticulous Path Verification

- In a single run, the tool performs meticulous path verifications simultaneously across multiple security specifications:
  - **Data integrity** – verifies that secure data transfers between protected domains without any corruption or unauthorized access
  - **Leakage prevention** – ensures sensitive data cannot reach unauthorized domains where it could be compromised
  - **Interference safeguarding** – stops unauthorized data from reaching and interfering with secure domains and their assets

Sentry enables early hardware security sign-off at scale. It can run a million gate design in only a few minutes. Running a **hundred million gate** design only takes a couple of hours.

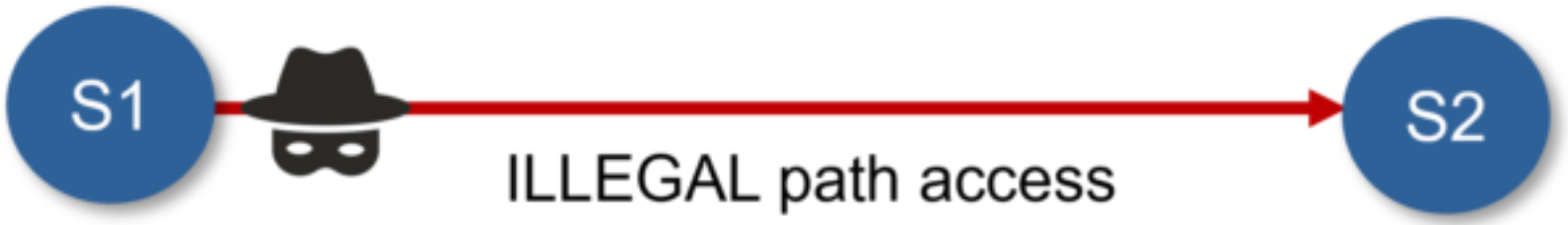
Data Integrity

- Sentry verifies that secure data transfers between protected domains without any corruption or unauthorized access
- Sentry will test for potential blocks to the paths and whether the path is vulnerable to unauthorized data transfers
- Representative access control check:
  - Ensure that the registers' read & write permissions are correctly set by checking that only specific processes can access certain registers, and under defined conditions.
  - Verify that only the CPU can write to configuration registers and that peripheral devices have restricted read-only access
  - Check whether access to any register is blocked



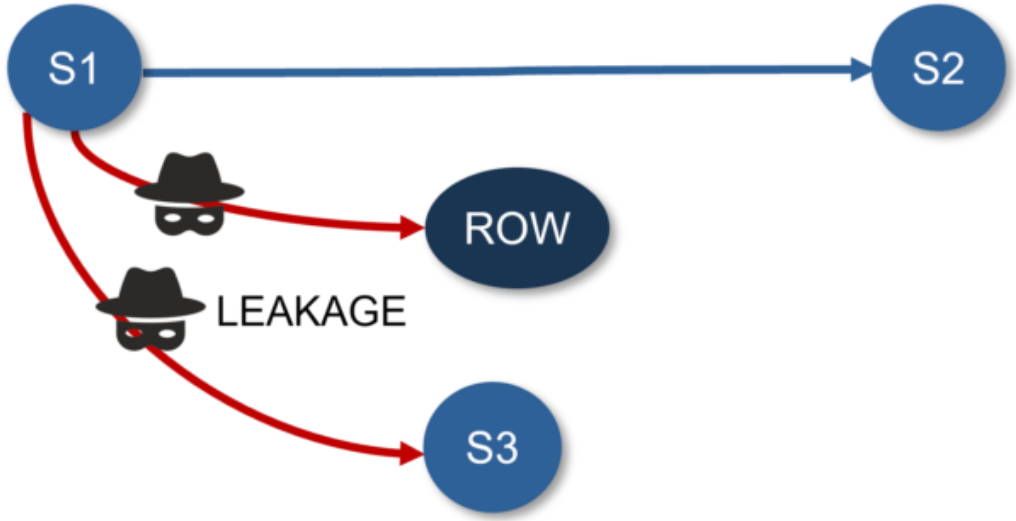
Illegal Path Access

- There is no illegal or unauthorized access or ensure something “bad” does not happen, such as writing to a read-only asset.
- Representative critical component isolation check:
  - Verify the integrity of bus separation or firewall mechanisms used to prevent third party IPs from accessing secured registers.



Leakage Prevention

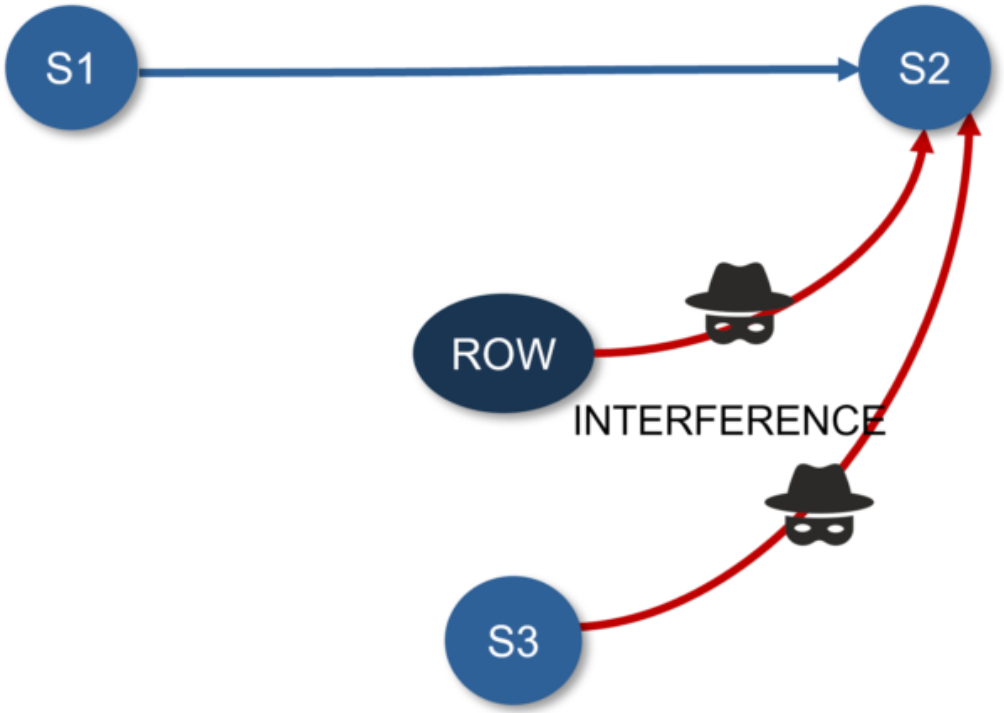
- Sentry can be used to ensure sensitive data cannot reach unauthorized domains where it could be compromised.
- Ensure that all data paths handling sensitive information are secure and isolated from non-secure data paths.
- Implement dedicated buses or secure channels for sensitive data, and use multiplexers and demultiplexers with access control checks to prevent unauthorized data redirection.



S1, S2, and S3 are all secure
ROW refers to the rest of the world

Interference Safeguarding

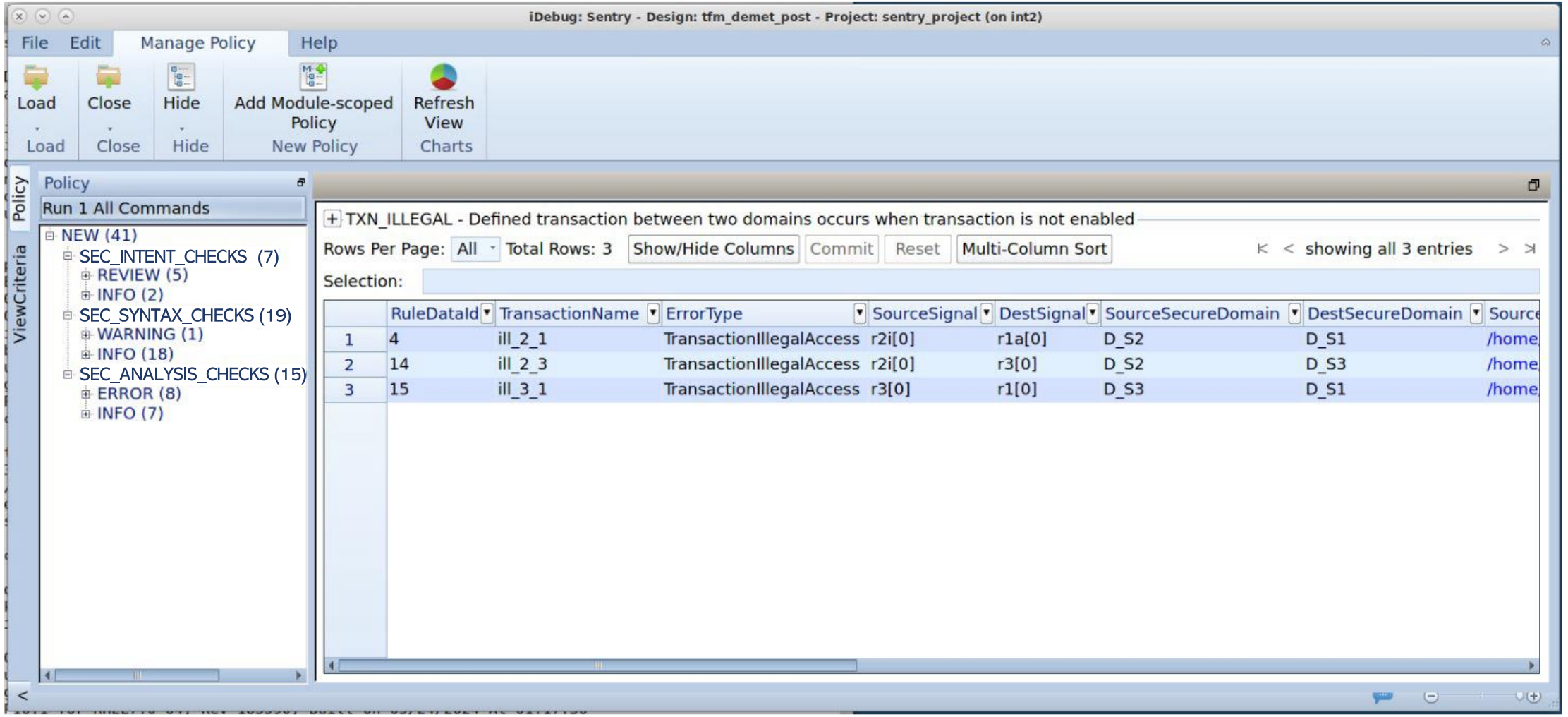
- Hardware designers can sign-off that there are safeguards to stop unauthorized data from reaching and interfering with secure domains.
- Access control logic check:
  - Verify that access control logic correctly grants access to authorized assets and blocks unauthorized ones.
- Data path monitoring and alerting check
  - Verify that suspicious or unauthorized data transfer activities that cross domain boundaries are detected.
  - Verify that unauthorized data transfer activities alerts are triggered and protective actions initiated, such as disconnecting the data path or initiating a secure reset.



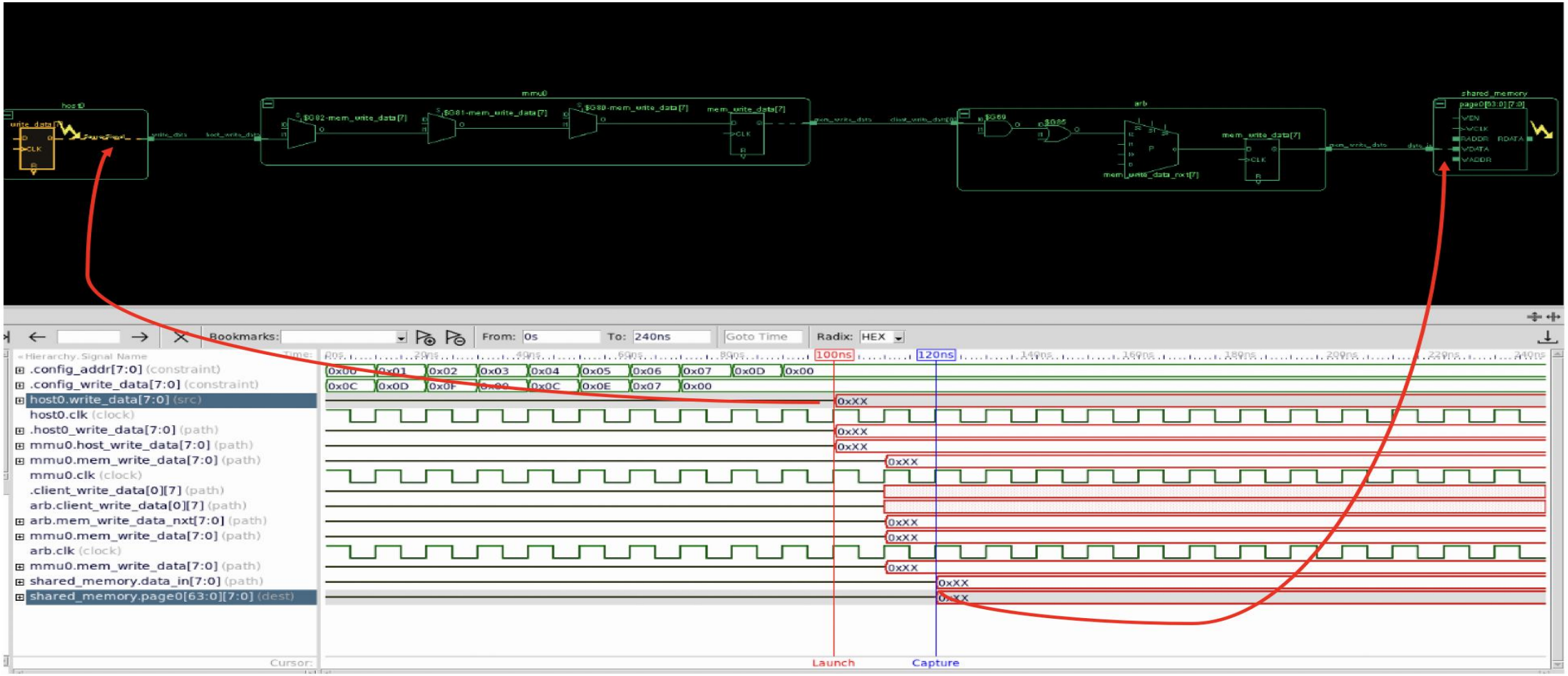
Transaction Checks

CHECK	Description
BLOCKED VALID	Flags a fail if there are no data transfer paths possible for a defined transaction, when the enable is true. This is checked for every defined transaction between a pair of domains.
LEAK NO_LEAK	Flags a fail if there is a data transfer path from the source domain to an external domain, when the enable is true. For every defined transaction between a pair of domains, this is checked for each external domain.
INTERFERENCE NO_INTERFERENCE	Flags a fail if there is a data transfer path from an external domain to the destination domain, when the enable is true. For every defined transaction between a pair of domains, this is checked for each external domain.
PATH NO_PATH	Flags a fail when there is a data transfer path possible between a pair of domains, when their transaction enable is false. This is checked for every defined transaction between a pair of domains.

iDebug: Report View



Advanced Debug for faster RCA: Valid



Advanced Debug for faster RCA: Blocked / Illegal

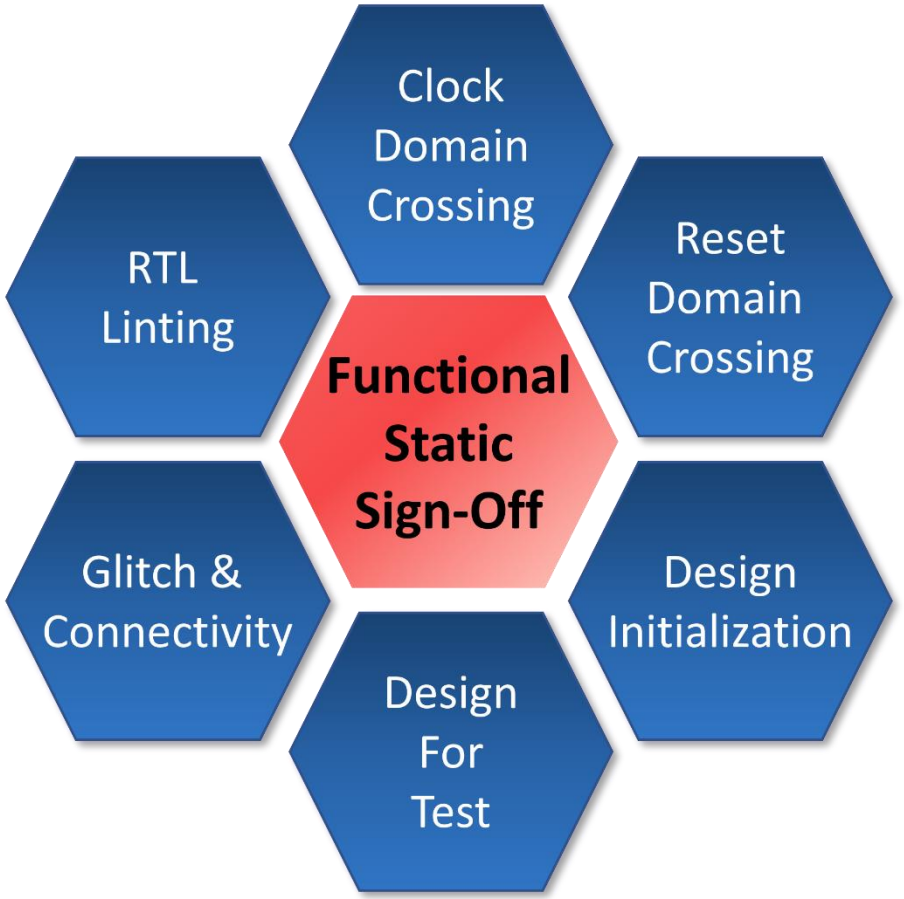
SENTRY – Hardware Security Sign-Off Product

- Industry's fastest, highest-capacity hardware security static sign-off tool
  - 1 million gates in minutes
  - 100 million gates in a couple hours.
- Data integrity – verifies that secure data transfers between protected domains without any corruption or unauthorized access
- Leakage prevention – ensures sensitive data cannot reach unauthorized domains where it could be compromised
- Interference safeguarding – stops unauthorized data from reaching and interfering with secure domains and their assets
- Auto Checks
- Advanced Debug Capability for Faster Root Cause Analysis

Functional Static Sign-Off Expanding Applications

Functional static sign-off began with RTL Linting & CDC

The target applications continuously expand



Real Intent static sign-off products lead the market in breadth, precision, performance, and capacity.

Questions?

Source: istockphoto