SEMISRAEL 2025 EXPO

Liberating Verification from Boolean Shackles

Oren Katzir
Vice President Applications Engineering
Real Intent Inc

REAL INTENT

Hardware Has Changed – Verification Flows Haven’t Kept Pace

- Today’s designs include chiplets, AI accelerators, GPUs, multi-die interconnects, heterogeneous IP, and power/thermal/security concerns — all beyond what “classic” UVM/testbench flows were designed for.
- Yet most flows still look like they did a decade ago: regression dashboards, random seeds, code coverage reports, waveform debug.

Verification is Boolean Heavy

- Simulation
  - At its heart, simulation evaluates **Boolean signal activity over time**.
  - Coverage (toggle, branch, functional) is still Boolean: “hit/not hit.”
- Formal
  - Formal engines convert the design into a **Boolean satisfiability (SAT/SMT)** problem.
  - Properties (SVA/PSL) are essentially **Boolean expressions over time** (e.g., signal A must imply signal B within N cycles).

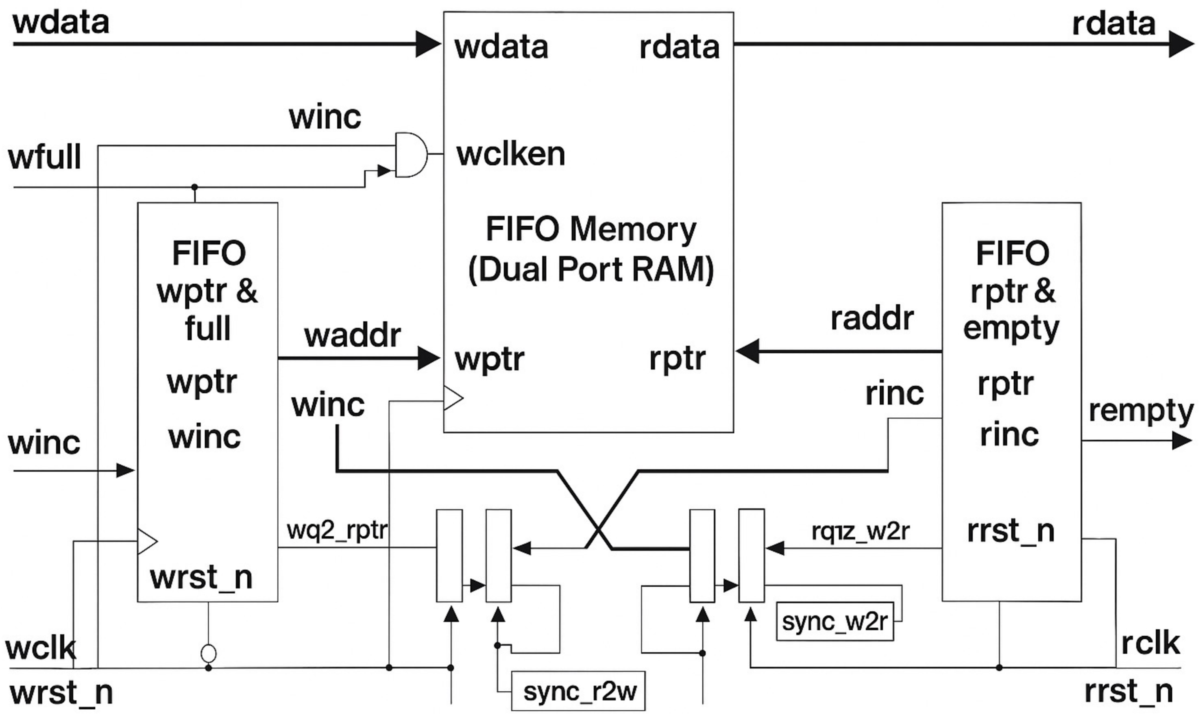
Early RTL design needs a different approach

Static Sign-off: Minimally Boolean

Liberates from
Boolean shackles

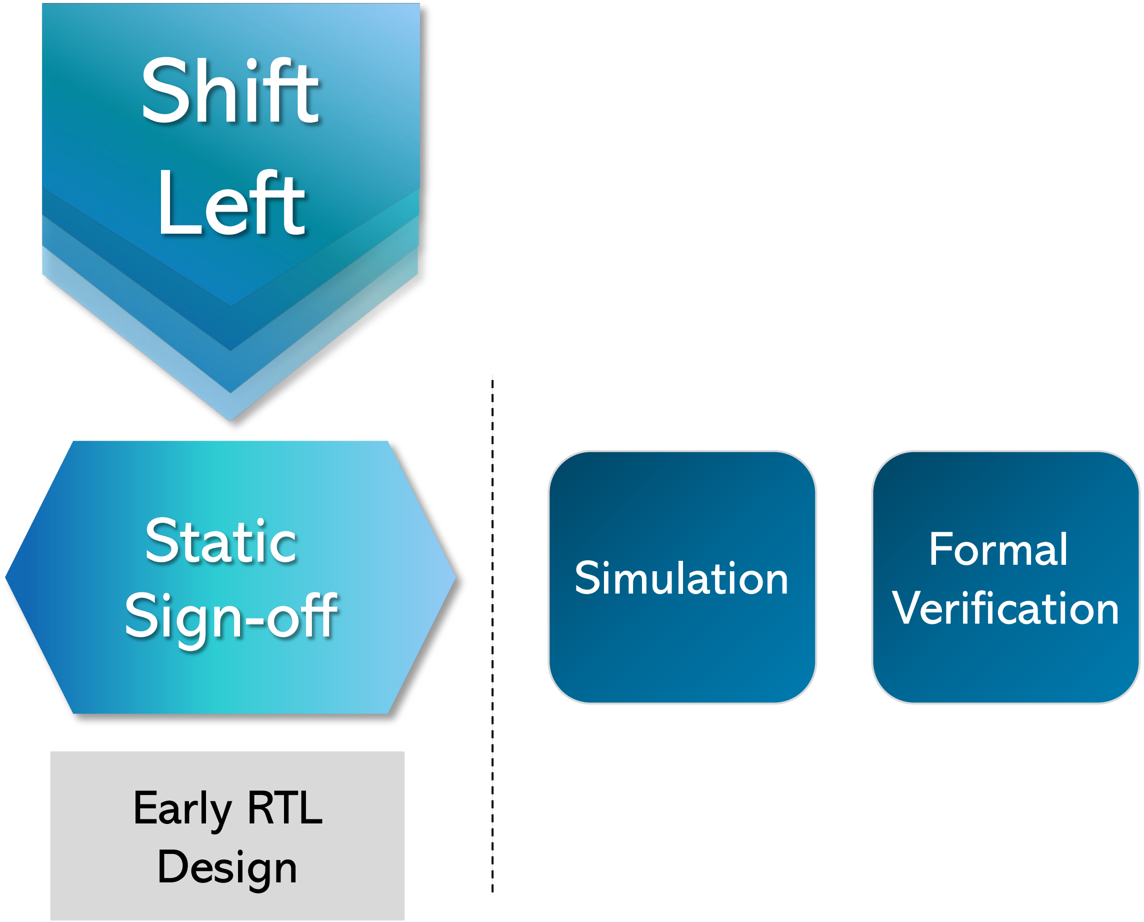
Asynchronous FIFO – Static Analysis

- Static analysis automatically abstract structures
  - RAM
  - Synchronizers
  - Pointer comparison
- Verifies correctness at abstract level
- Minimally Boolean!



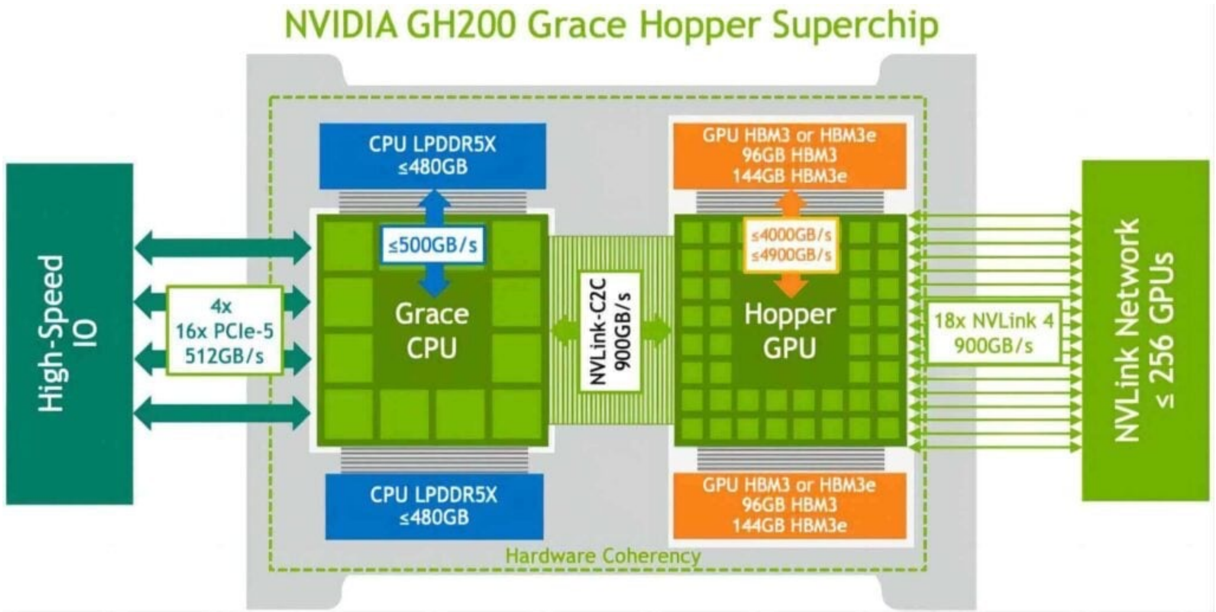
REAL INTENT

Static Sign-off: Starts Early



- Benefits
  - Identify defects early: minimize costly rework, save time & resources
  - Improve design quality: address design flaws, optimize architecture, higher quality designs
  - Shorten design cycle
  - Risk reduction: catch failures, security vulnerabilities for robust design

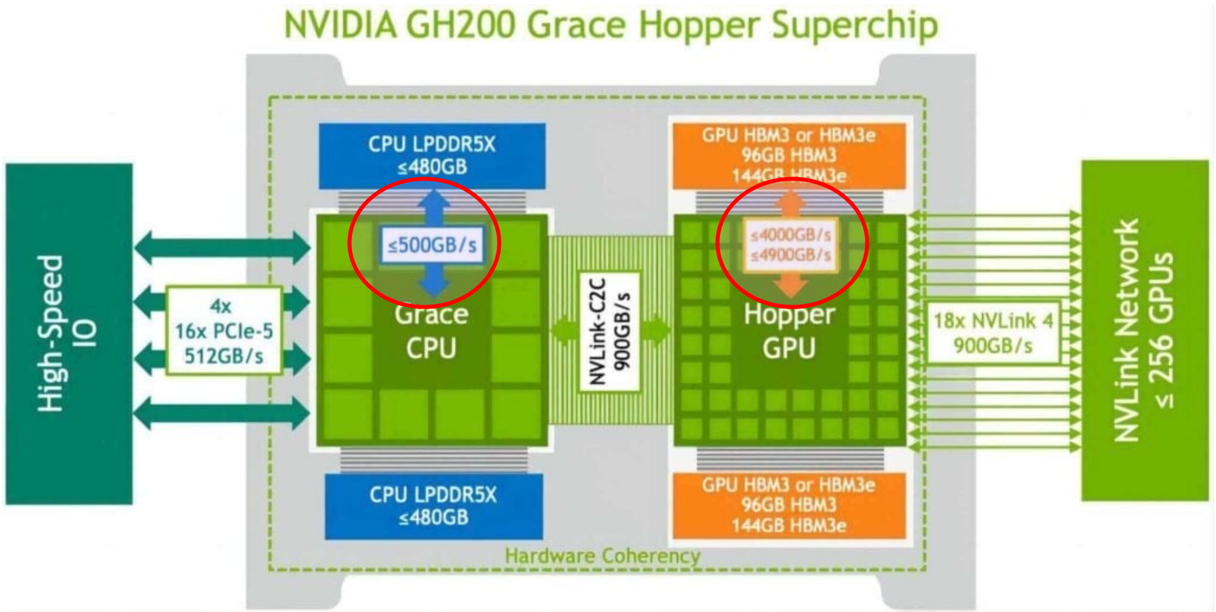
Verification Challenges AI Chips



- Massive Asynchronous paths and power domain verification – CPU-GPU-HBM
- CPU + GPU + High-Bandwidth Memories (HBM3/3e + LPDDR5X)
  - Protocol correctness
  - Cache coherency
  - Memory consistency
- NVLink-C2C
  - Interconnect verification
  - Topology correctness routing, congestion …
- PCIe Gen5, HBM3(e) interfaces
  - I/O Verification
- Security and Reliability
  - Ensuring isolation between CPU/GPU address spaces
  - Preventing privilege escalation through NV Link or memory mapping
- System Level Verification

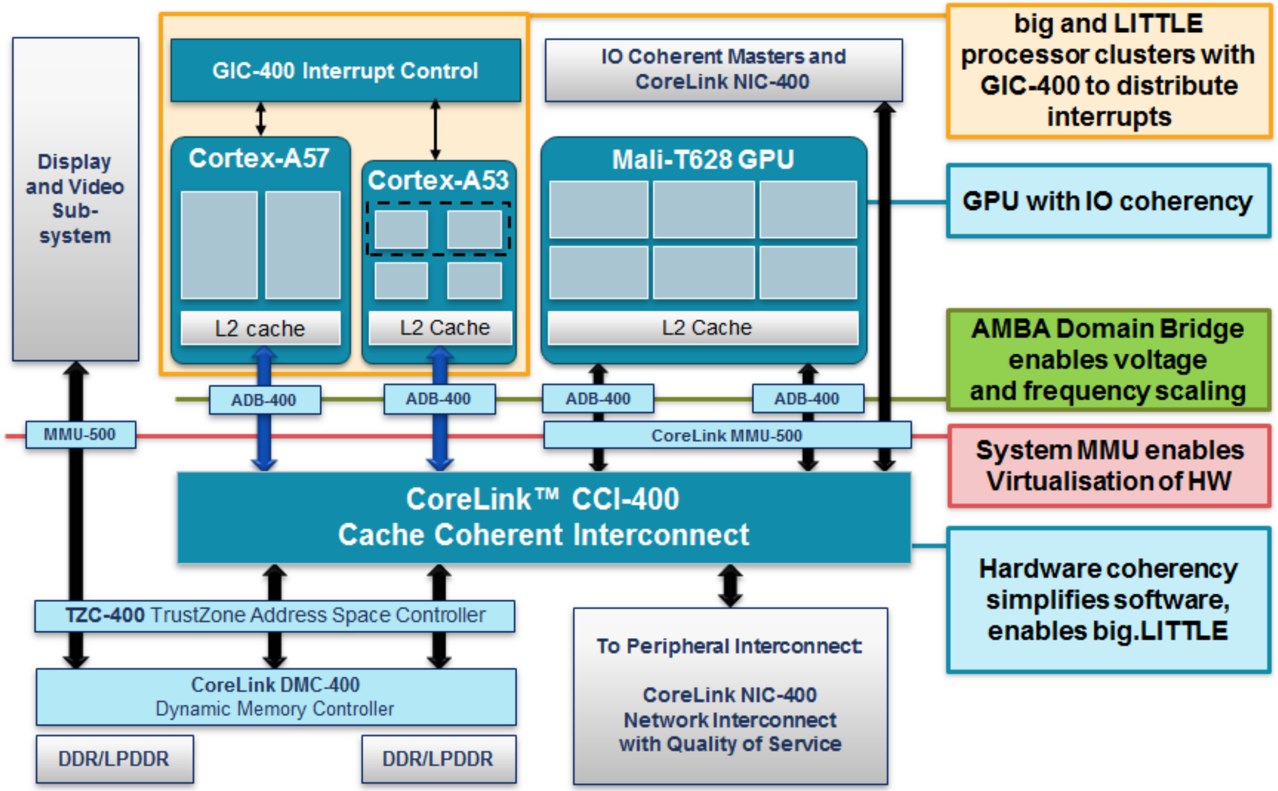
REAL INTENT

Asynchronous Path Verification Challenges



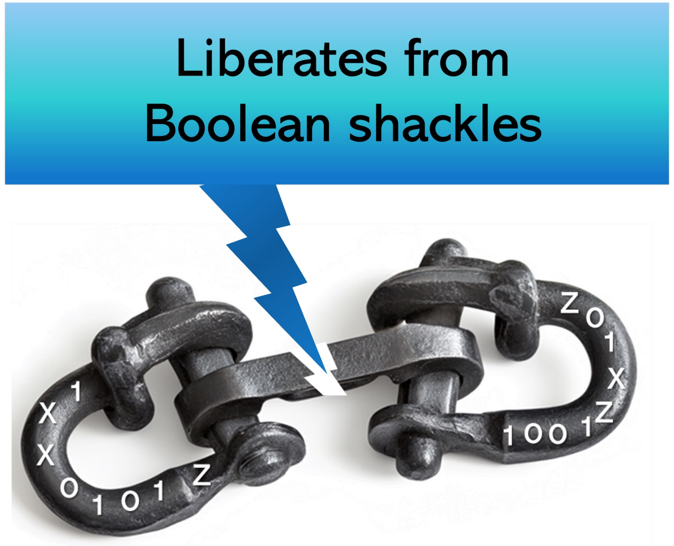
- Multi clock domains verification due to power requirements
- Handshake functional correctness
- Safety from Glitch Hazards
- Functional correctness under metastability
- No reset metastability

Verification Challenges Non-AI Chips



- Cortex-A57/A53 clusters
  - CPU Verification
- CoreLink CCI-400 cache coherent interconnect
  - Interconnect Verification
- GIC interrupt controller
  - Correct distribution between big and LITTLE clusters.
- Design Initialization
- Security (TrustZone, TZC-400)
  - World separation: Normal world vs Secure world must be strictly isolated.
  - Illegal access prevention: Verifying that GPU/DMA cannot access secure-only memory.
  - Corner case: Speculative loads/stores bypassing TZC checks.

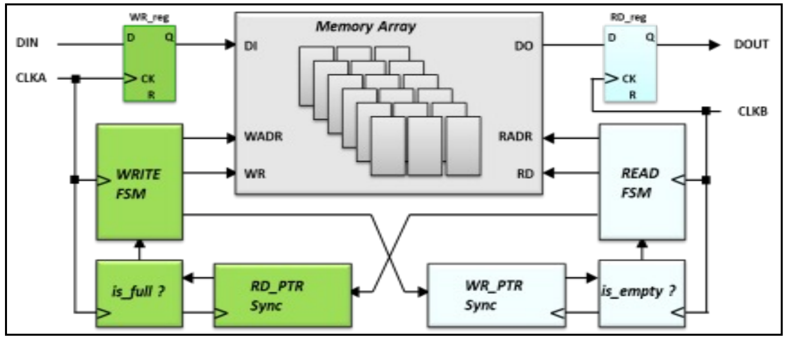
REAL INTENT



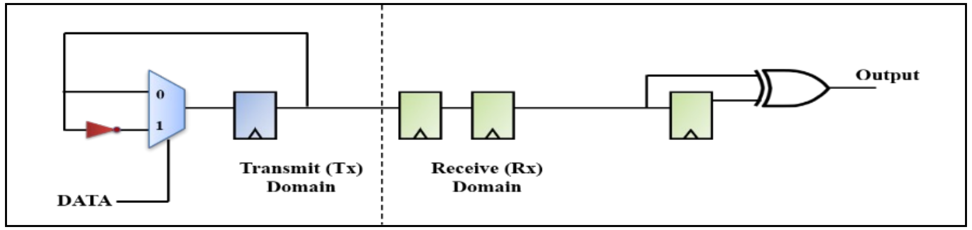
Minimally Boolean

Where can we use Static technology?

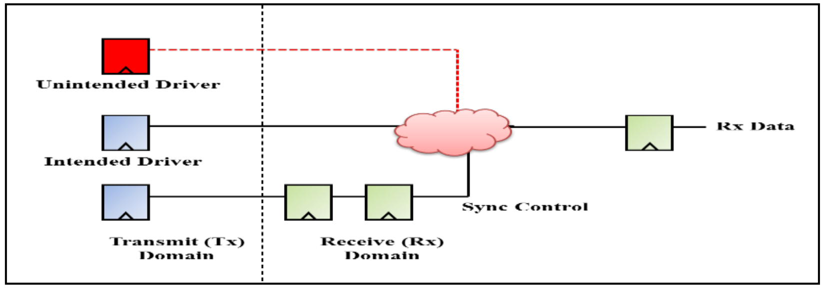
Handshake Functional Correctness



FIFO Handshake

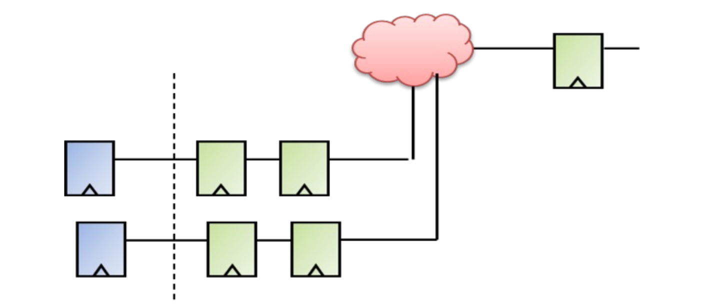


Auto-Identified Pulse Synchronizer

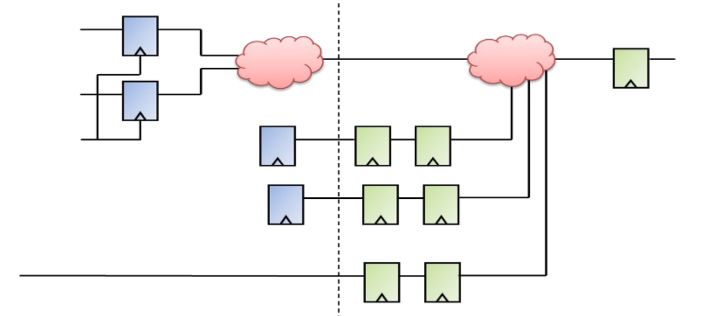


Unintended Driver – ERROR

Unsafe Reconvergence

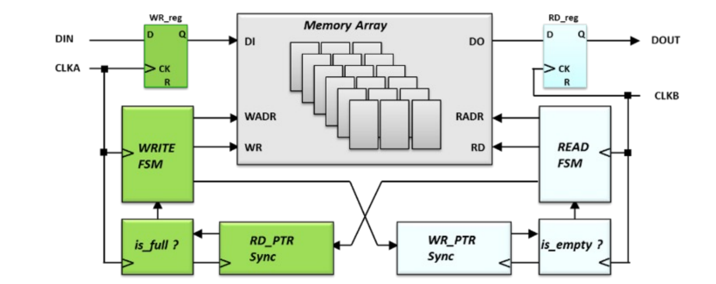


Recon without interface

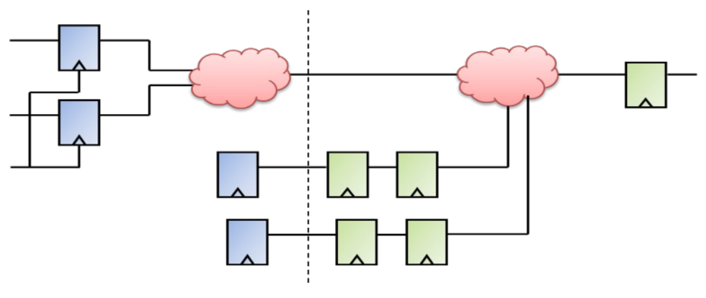


Recon of global control signals

Safe Reconvergence



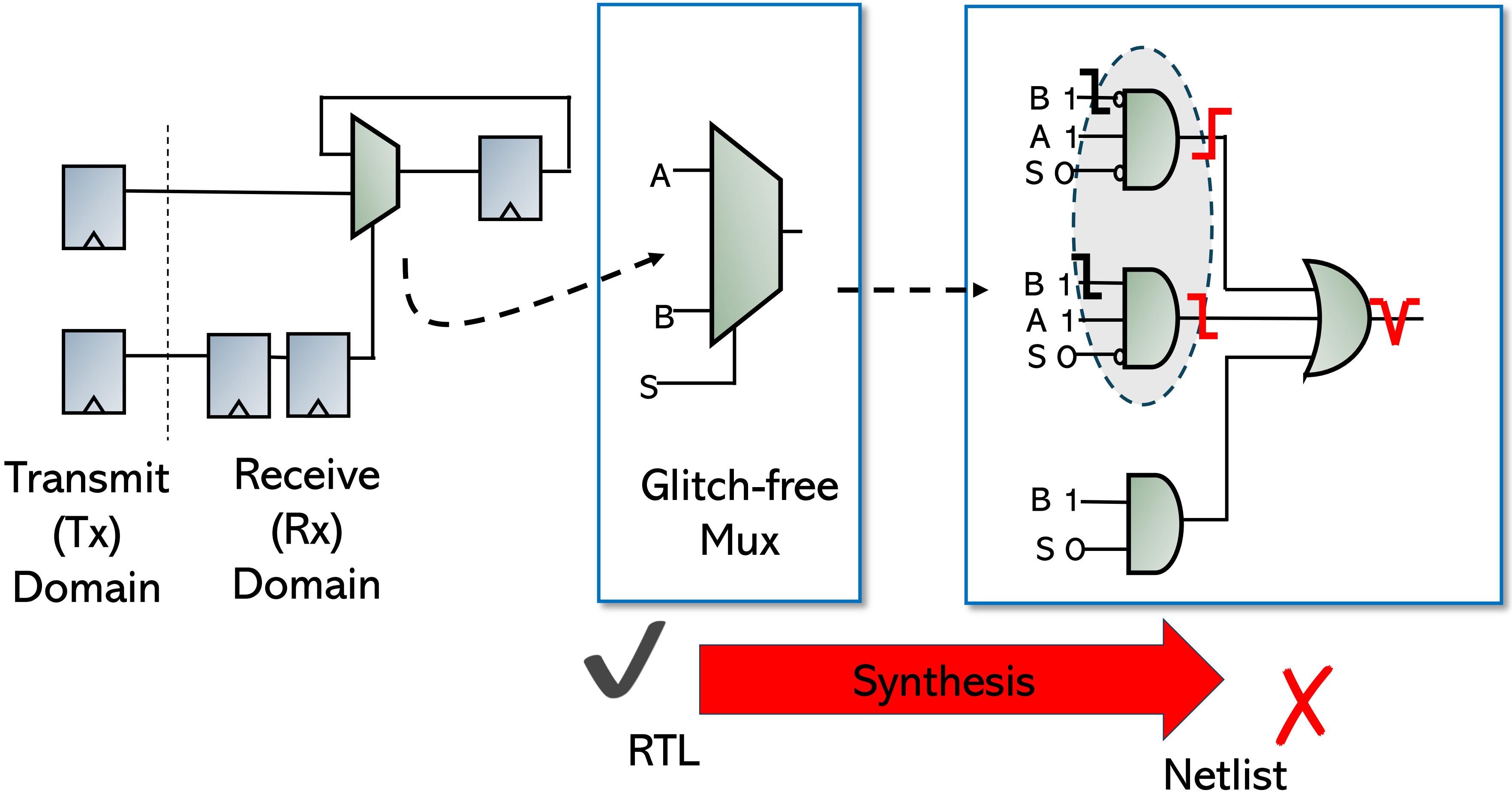
Recon within interface (FIFO)



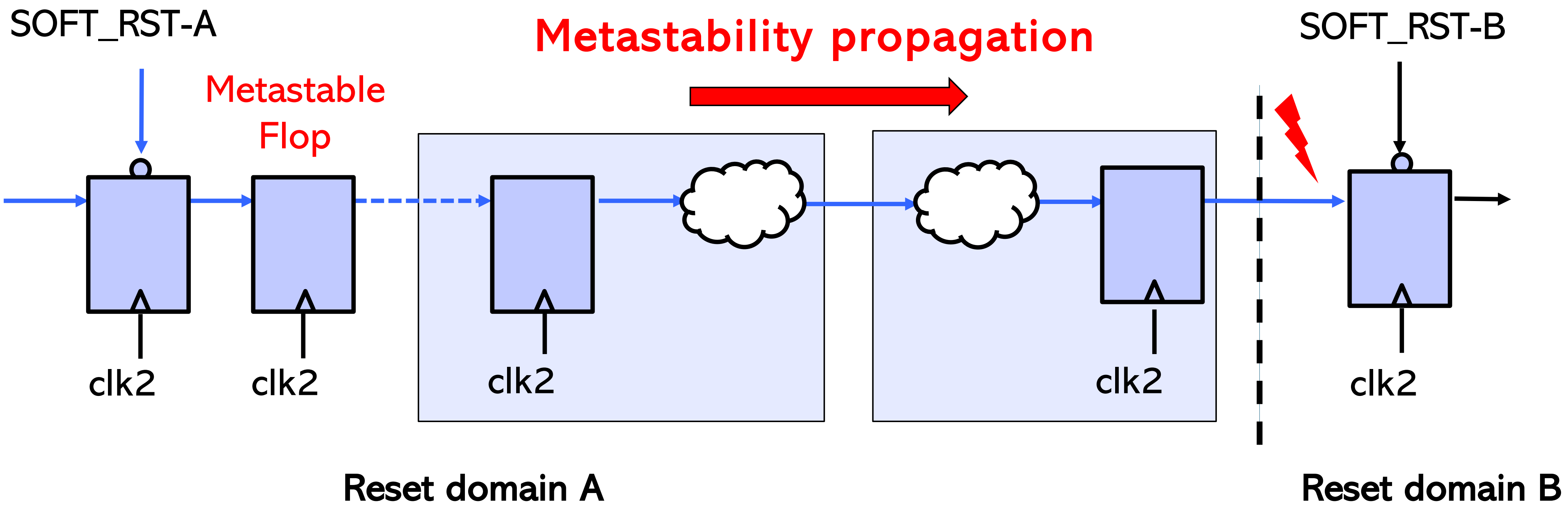
Recon within interface (Load Control)

REAL INTENT

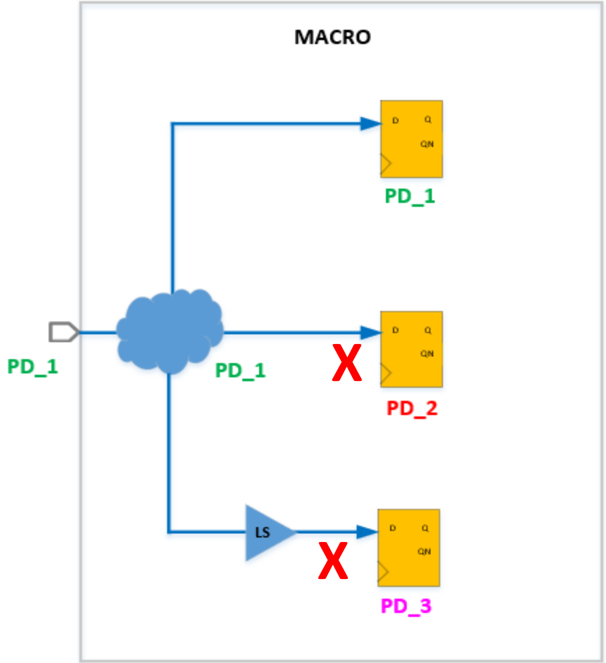
Safety from Glitch Hazards



No Reset Metastability

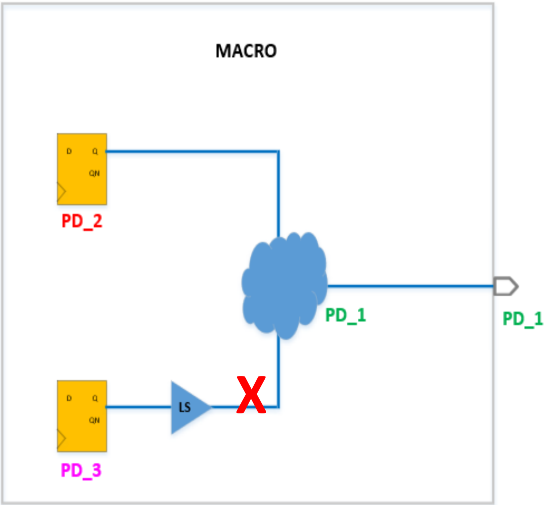


Power Isolation Verification

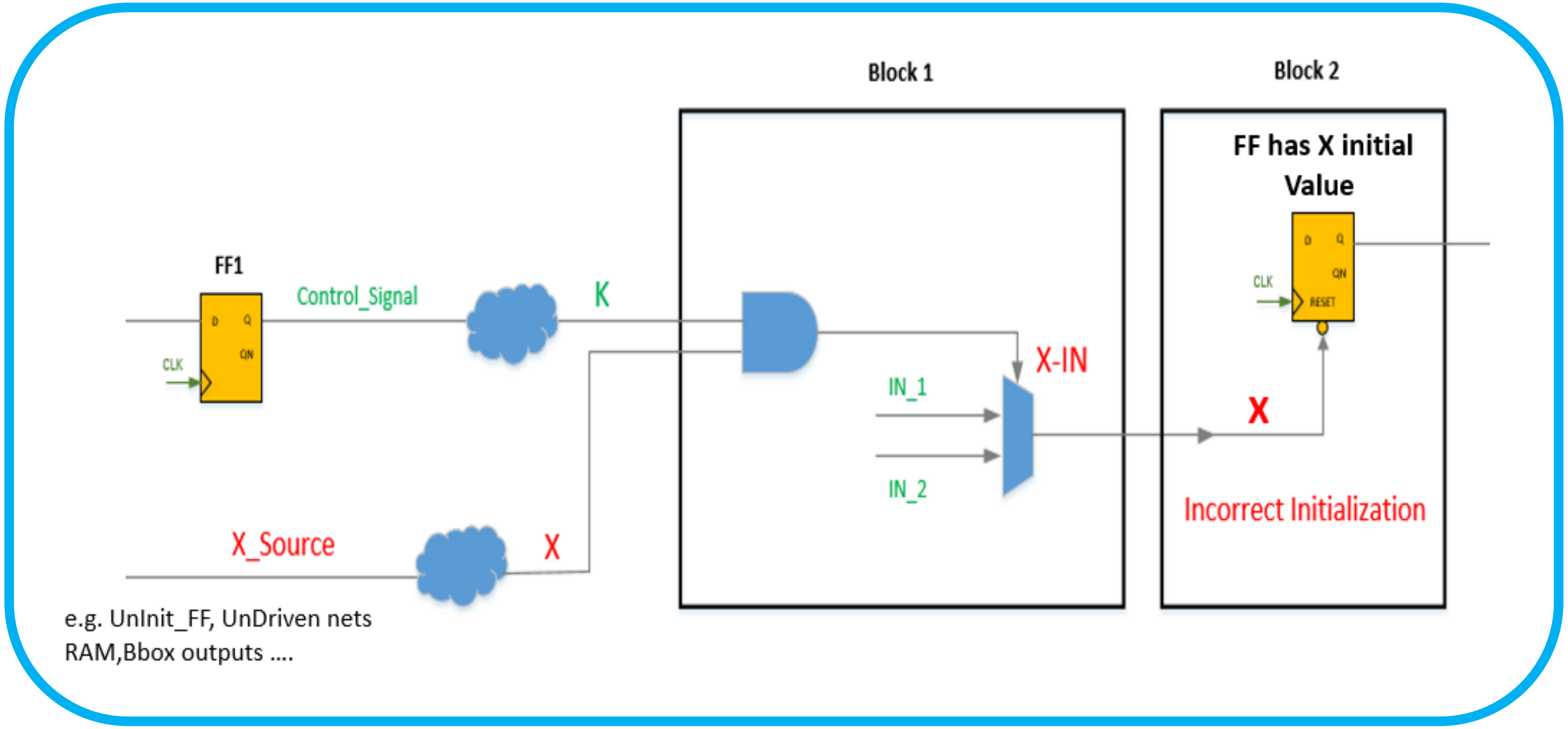


Check that any input of the macro can only drive FFs in the same power domain

Check that any output of the macro can only be driven by FFs in same power domain

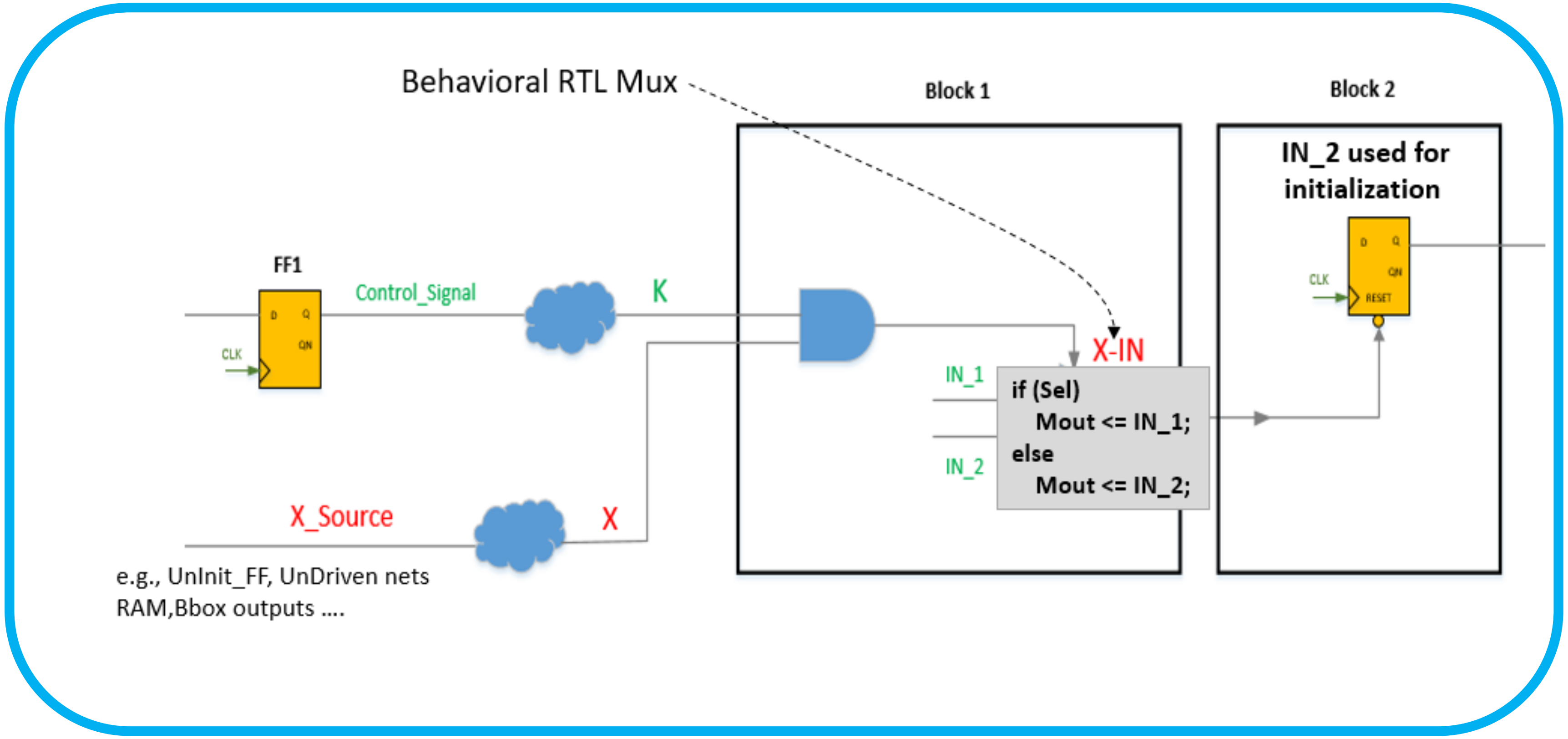


Incorrect Design Initialization (1)



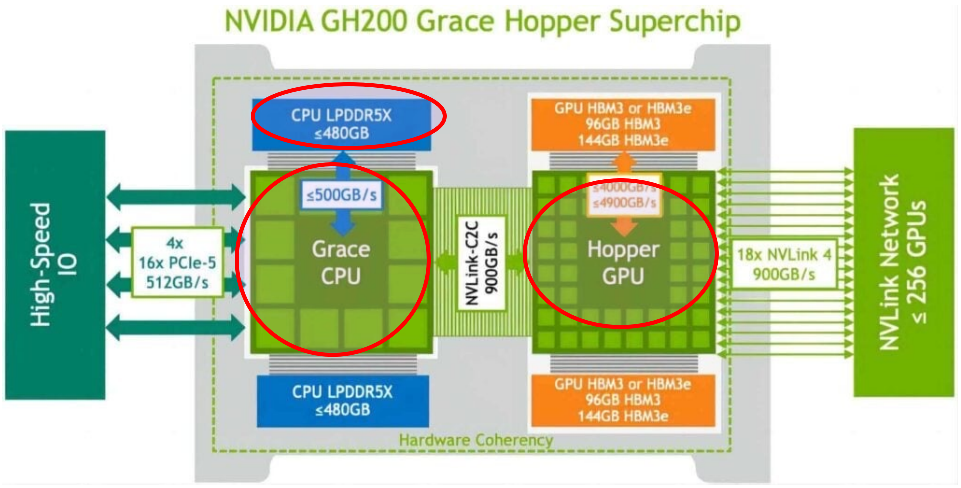
REAL INTENT

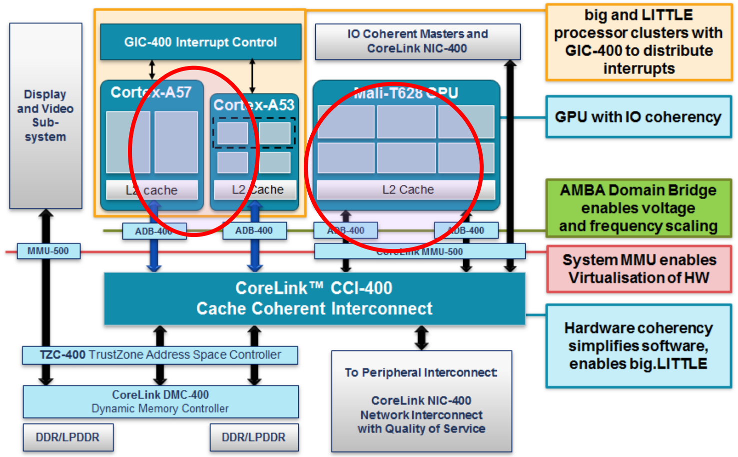
X-Optimistic Code Breaks Downstream Logic



REAL INTENT

CPU/GPU Verification Challenges



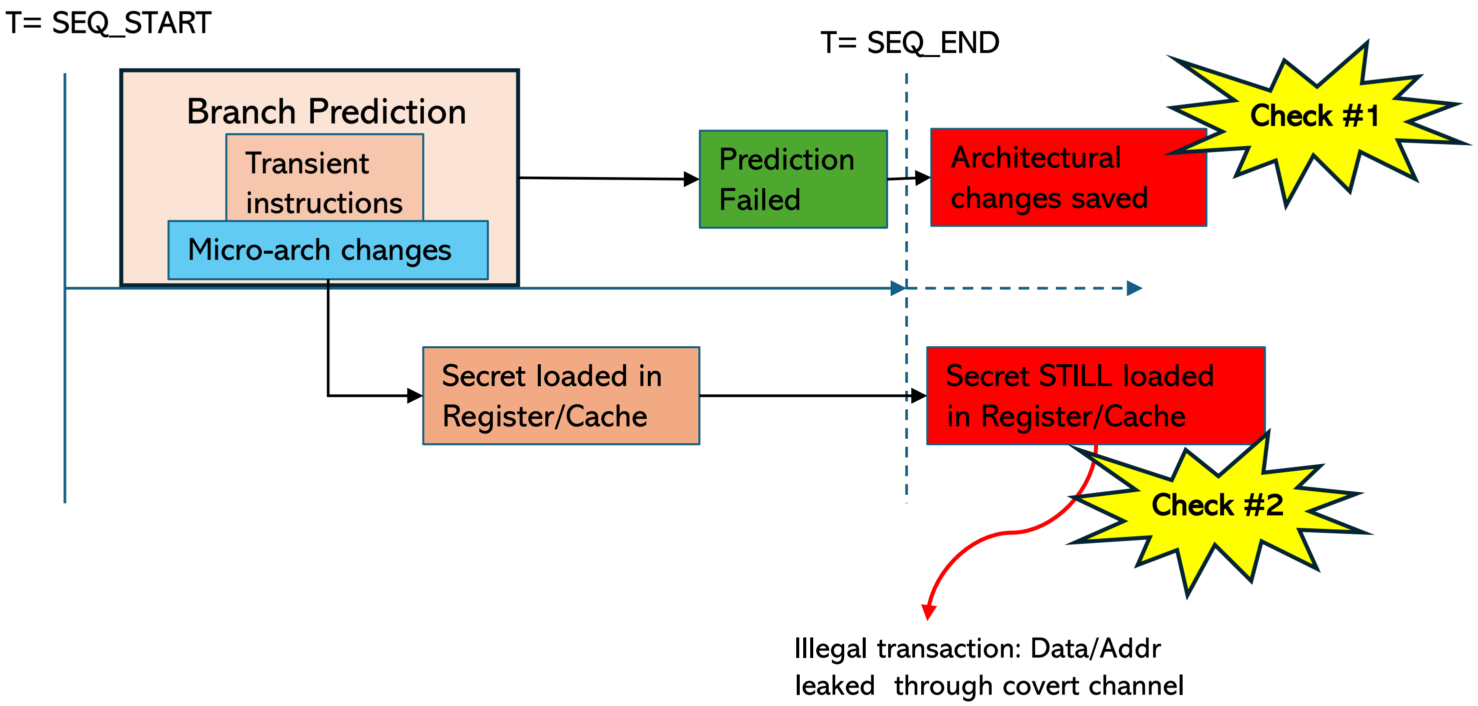


- Side-channel vulnerabilities (e.g., Spectre, Meltdown) arise from speculation, caches, and timing.
- Correct handling of exceptions, interrupts, and privilege levels must be validated across all instruction flows
- Root of Trust Verification

Speculative Execution Vulnerabilities

Leakage Scenario

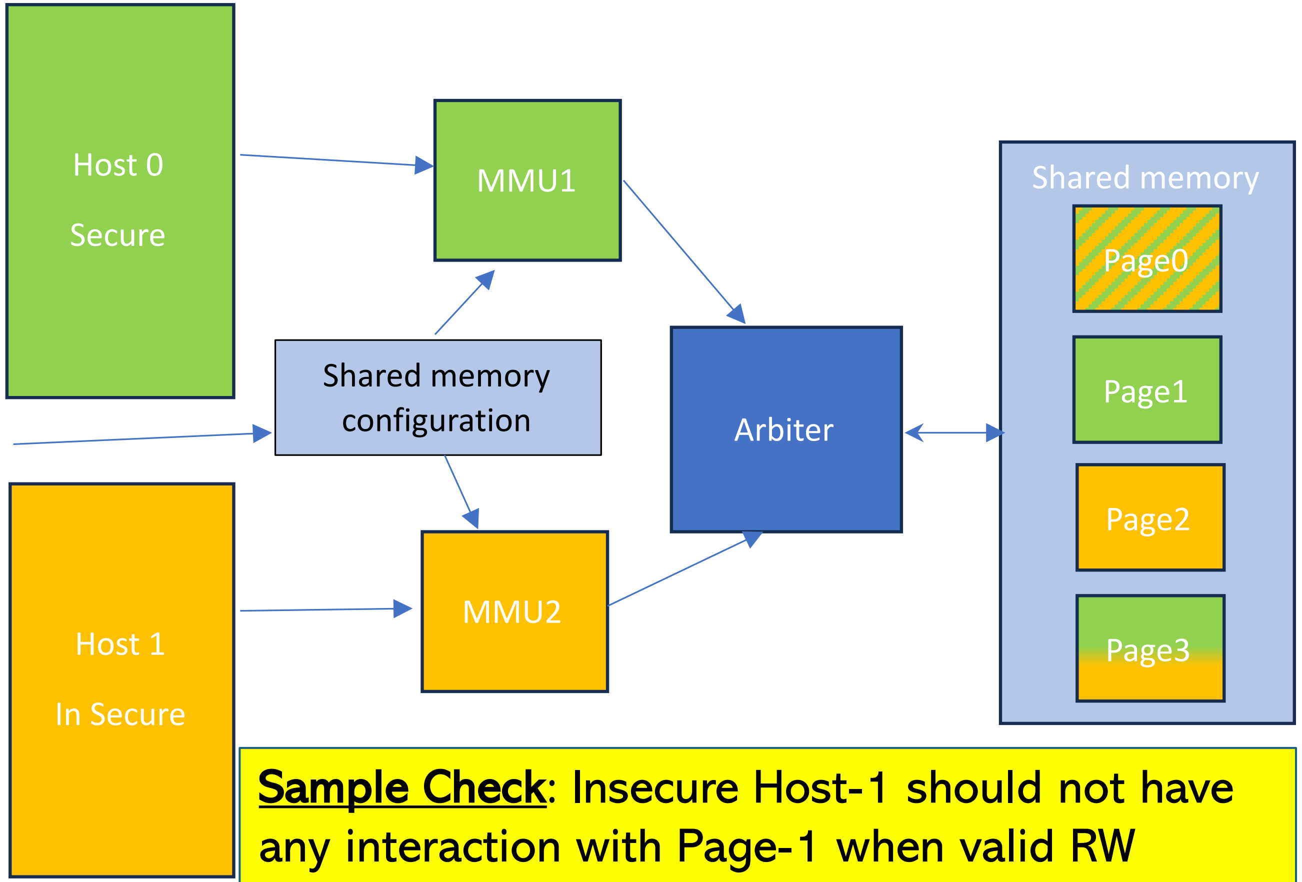
- Expose architecturally restricted data
- Allow predictor training in one domain to influence behaviour in another domain
- Caused by stale/incorrect data forwarding



REAL INTENT

Privilege Access in Multi CPU System

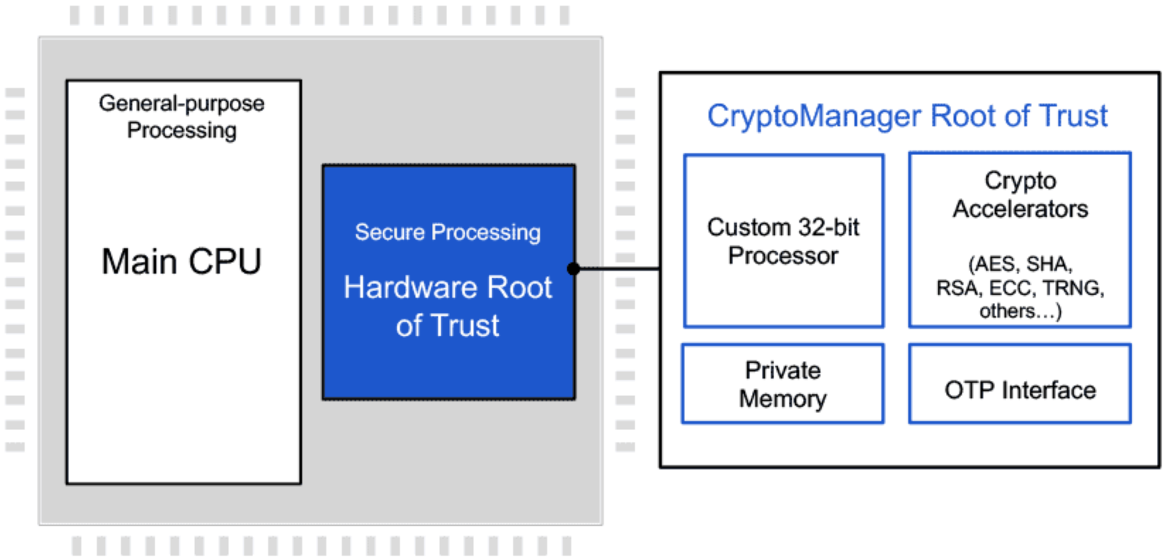
- Host0 and Host1 share the memory
  - Host0: Secure
  - Host1: Insecure.
- Shared Memory: 4 pages
- MMU's configures page is read or write to/from a host.
- Example configuration in the MMU's:
  - Page 0 is shared RW and accessible by both
  - Page 1 is exclusive RW to Host0
  - Page 2 is exclusive RW to Host1
  - Page 3 is mailbox RW by Host0, read only by Host1
- Configuration is written into the shared memory configuration block via a sequence of writes from an external entity.



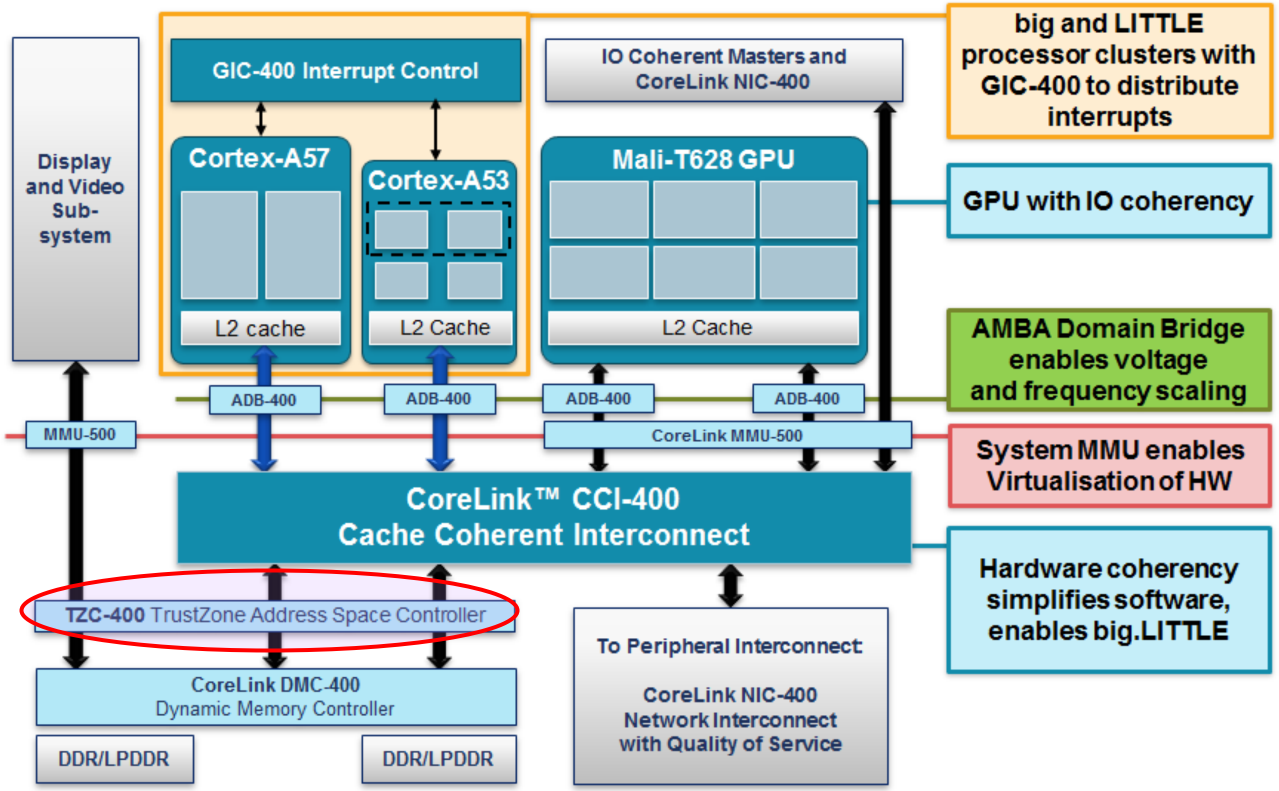
Sample Check: Insecure Host-1 should not have any interaction with Page-1 when valid RW transaction is happening between secured Host-0 and Page-1.

CPU- Hardware Root Of Trust

- Secure Key Storage
  - Key material must be stored securely to prevent compromise, as the exposure of cryptographic keys can lead to systemic security failures.
- Side Channel Detection
  - Hardware is often susceptible to side channel attacks where attackers glean sensitive information

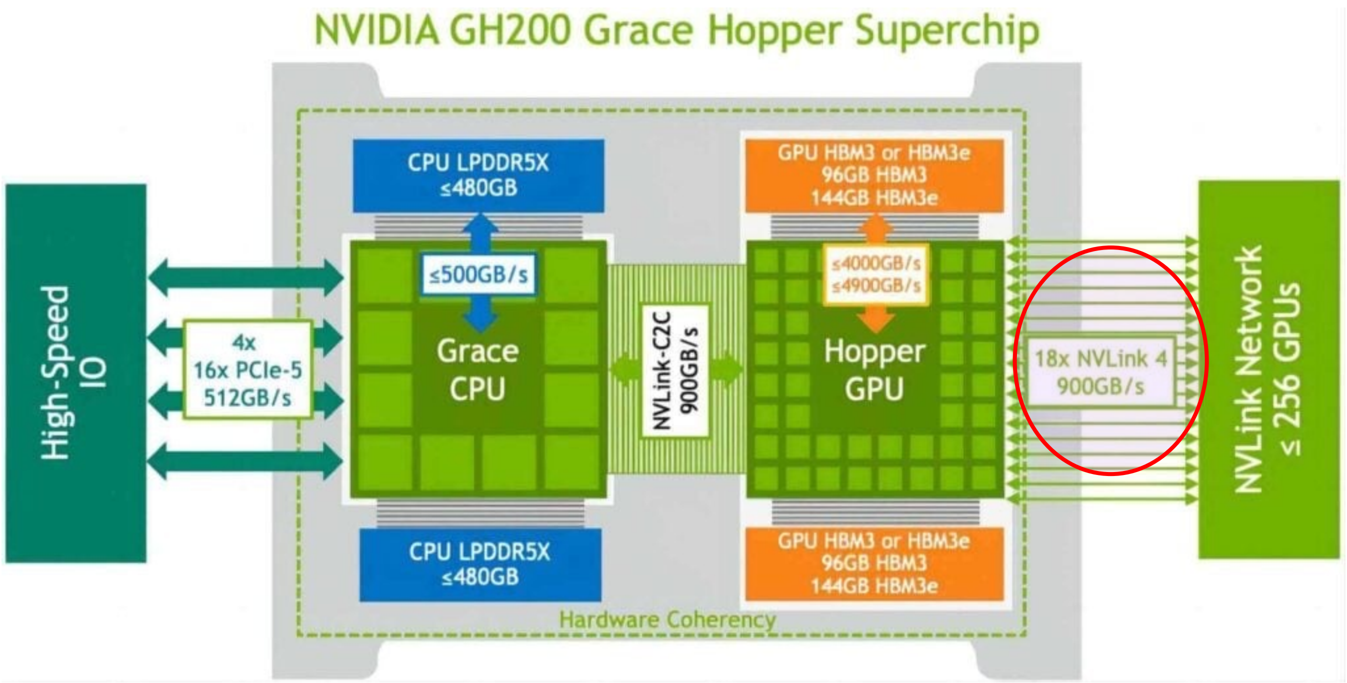


Security IPs – Verification Challenges



- Security (TrustZone, TZC-400)
  - World separation: Normal world vs Secure world must be strictly isolated.
  - Illegal access prevention: Verifying that GPU/DMA cannot access secure-only memory.
  - Corner case: Speculative loads/stores bypassing TZC checks.

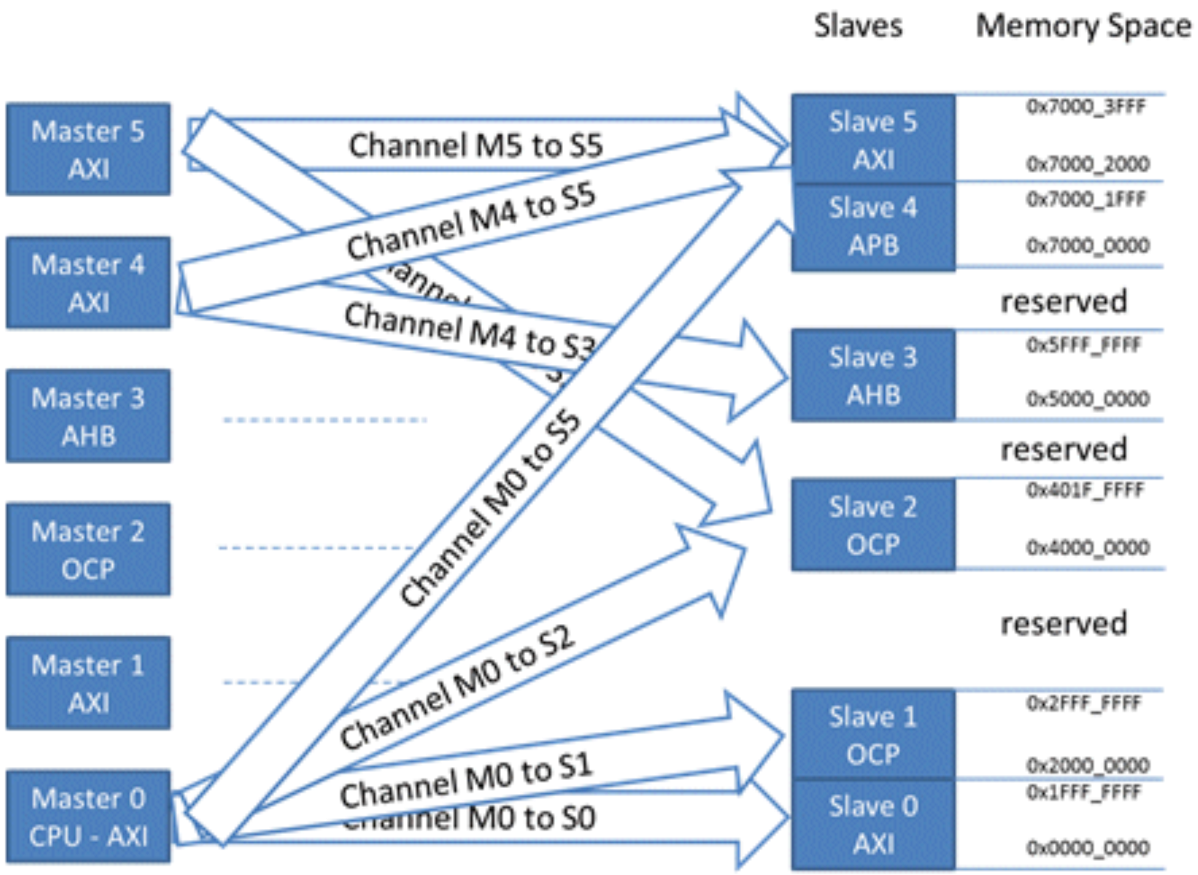
Interconnect Verification



- Data Integrity Verification
- Adress Mapping Verification
- Routing Configuration Verification
- Quality of Service
- Security Access Verification

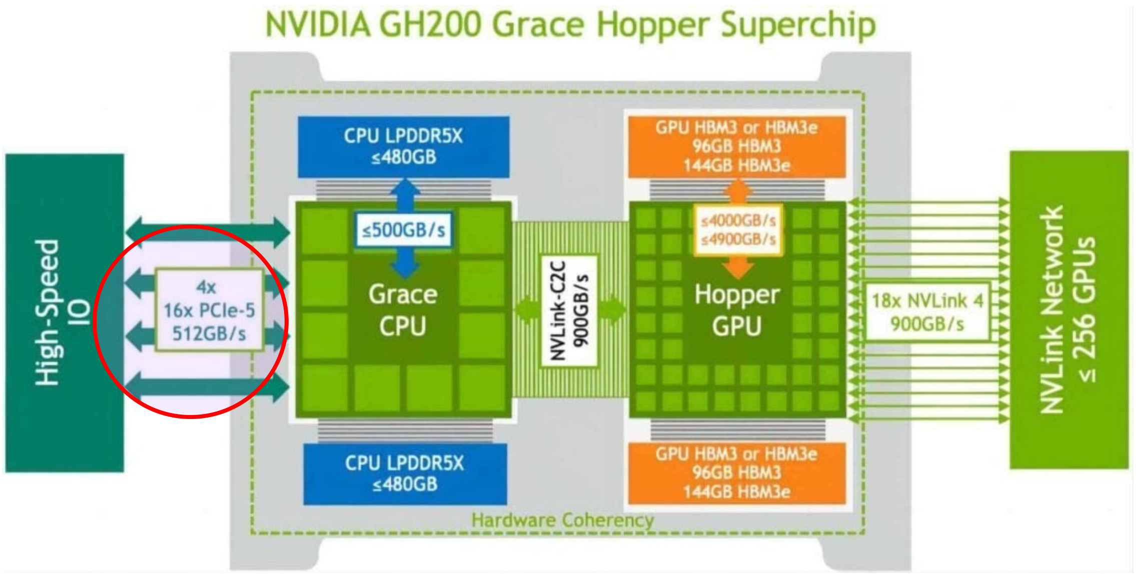
Interconnect Verification

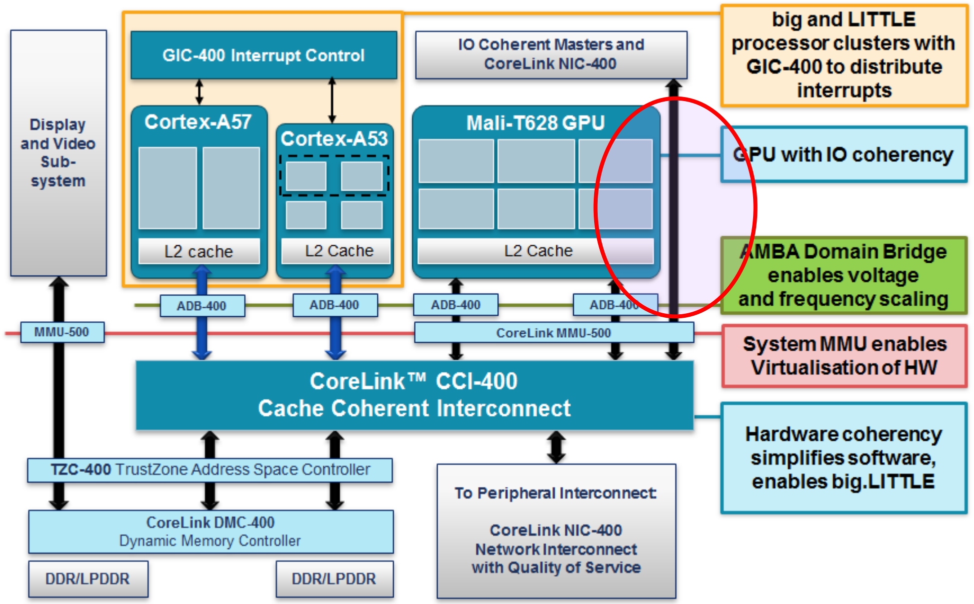
- Reachability Verification
  - A particular master connected to slave
  - Protocol ports connected correctly
  - Connections through complex sequential logic (bridges etc.)
  - Negative verification – masters not connected to slaves as per specification



REAL INTENT

IO Verification

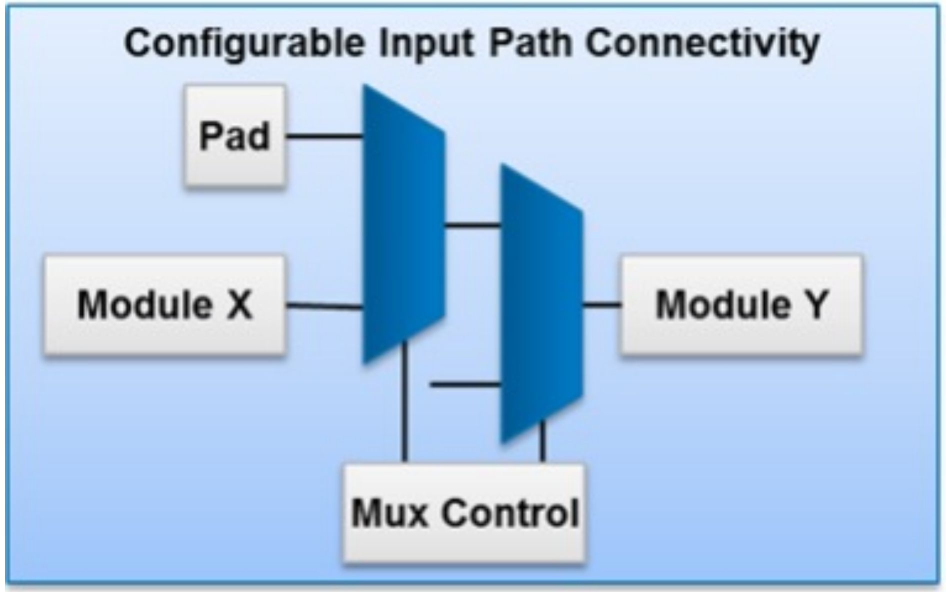


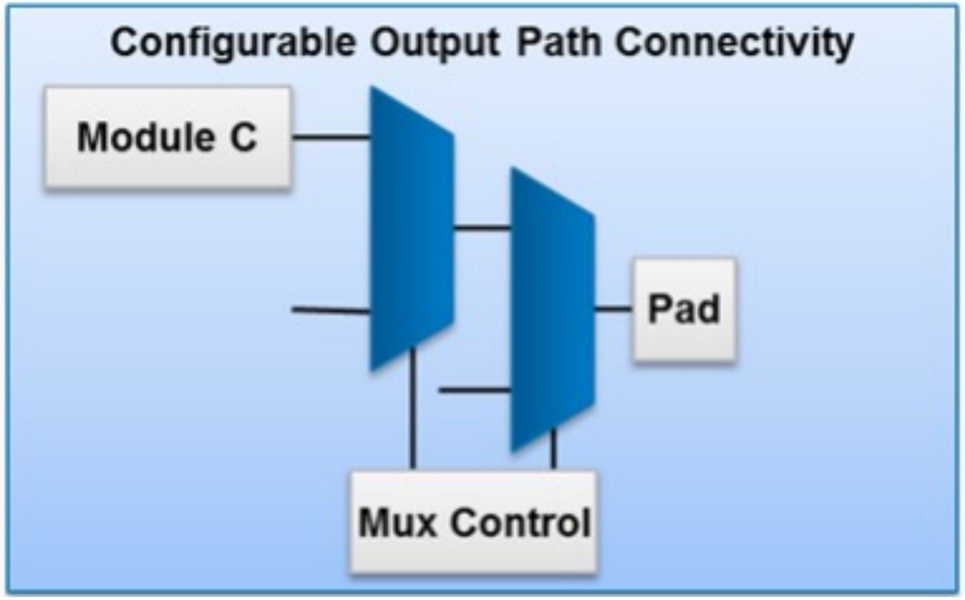


REAL INTENT

IO Verification

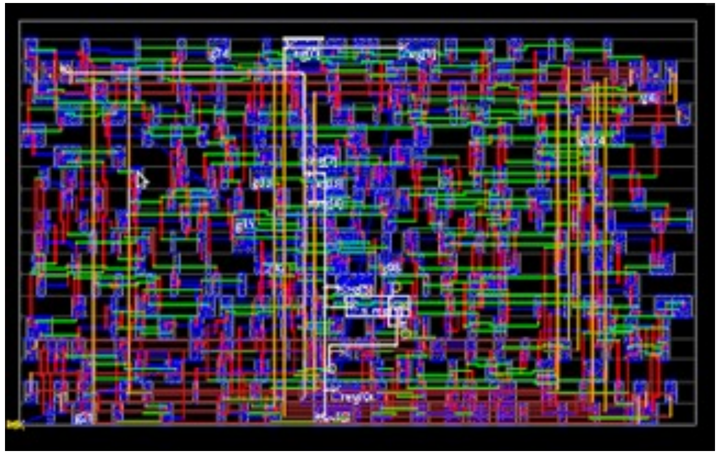
- Source to Destination connection
  - Based upon conditions

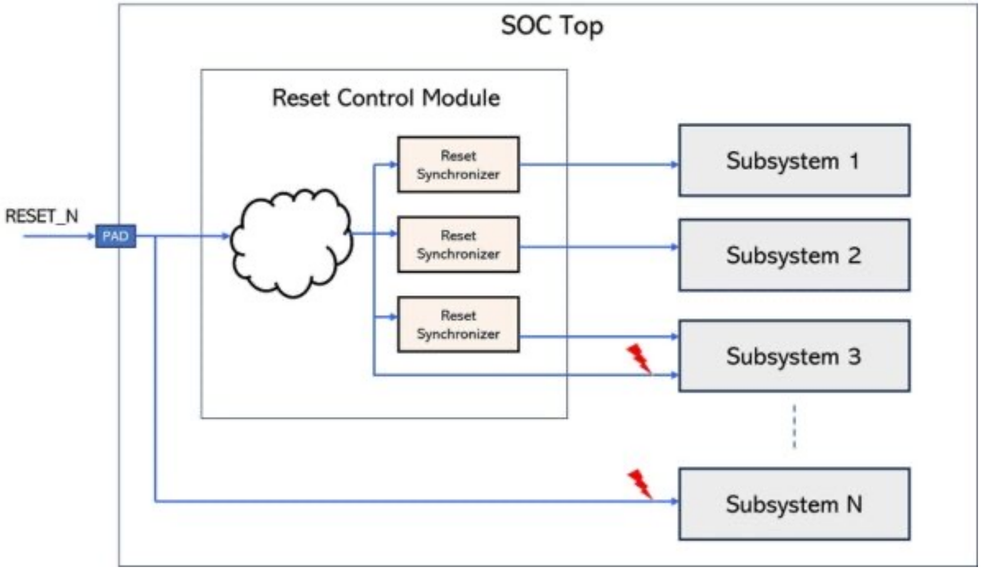


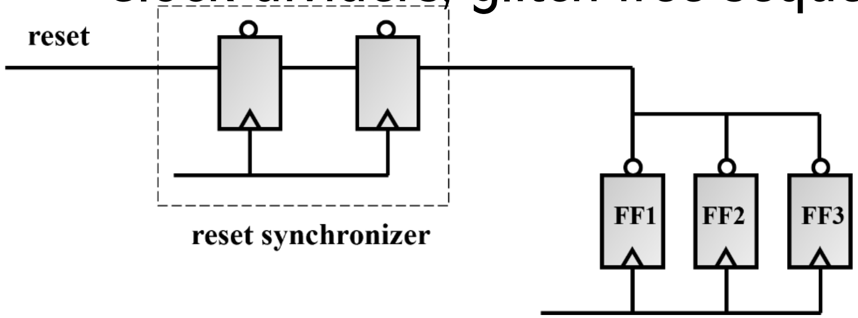


IO Global Signals

- Reset, Clock and Global signals
  - Connected to all the flops in the design
  - Millions of paths
  - Polarity is important
  - Can propagate through specialized cells
    - Reset synchronizers
    - Clock dividers, glitch free sequential muxe



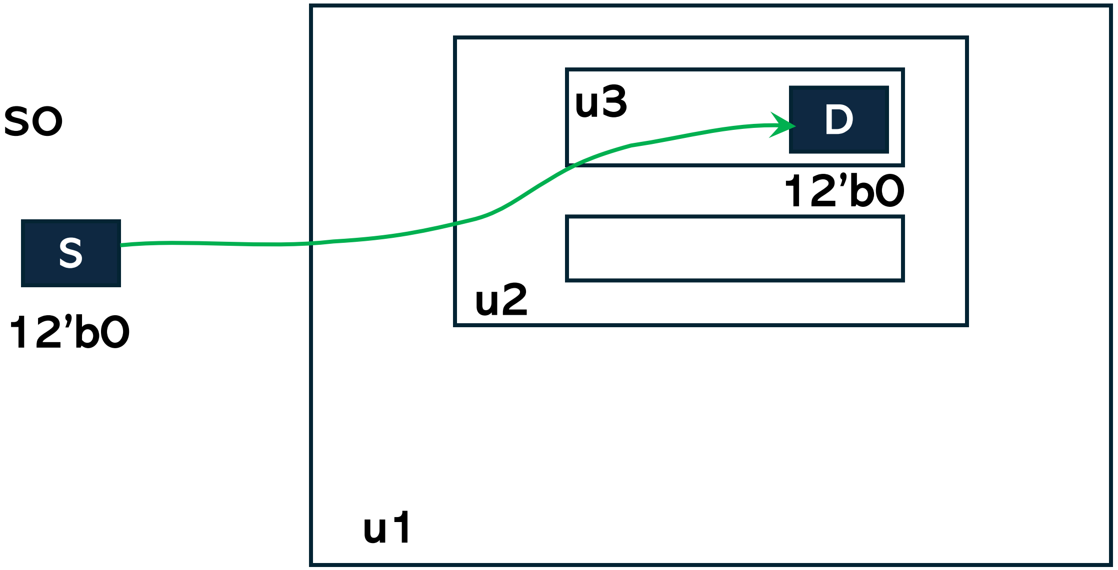




REAL INTENT

IO Registers Verification

- System Registers, Configuration registers
  - Specific connectivity register based
  - Can propagate through sequential
  - Not just connection, value propagation also important



RTL Linting → Lint error free

Clock Domain Crossing → CDC error free

Reset Domain Crossing → RDC error free

Design for Testability → DFT error free

Connectivity → Connectivity error free

Hardware Security → HW Security protected

Initialization Checker → Initialization, X-Prop

Minimally Bolean

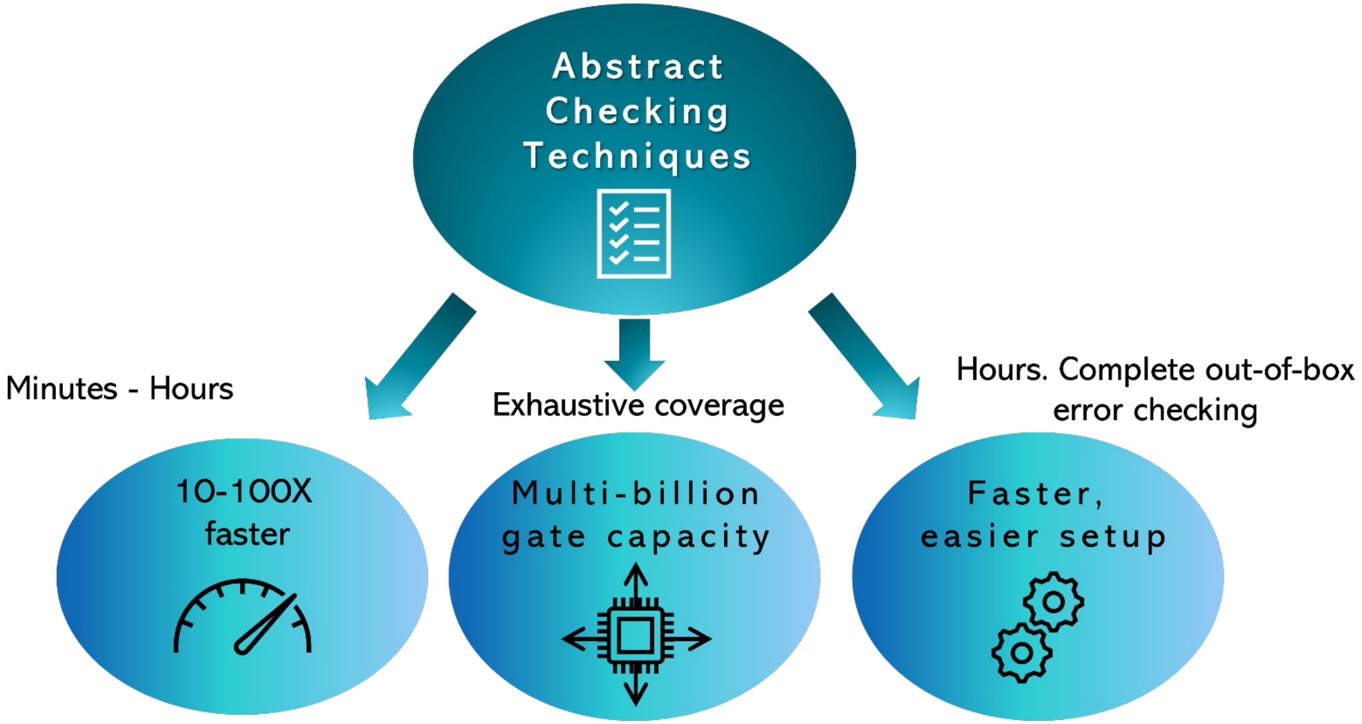
Simulation

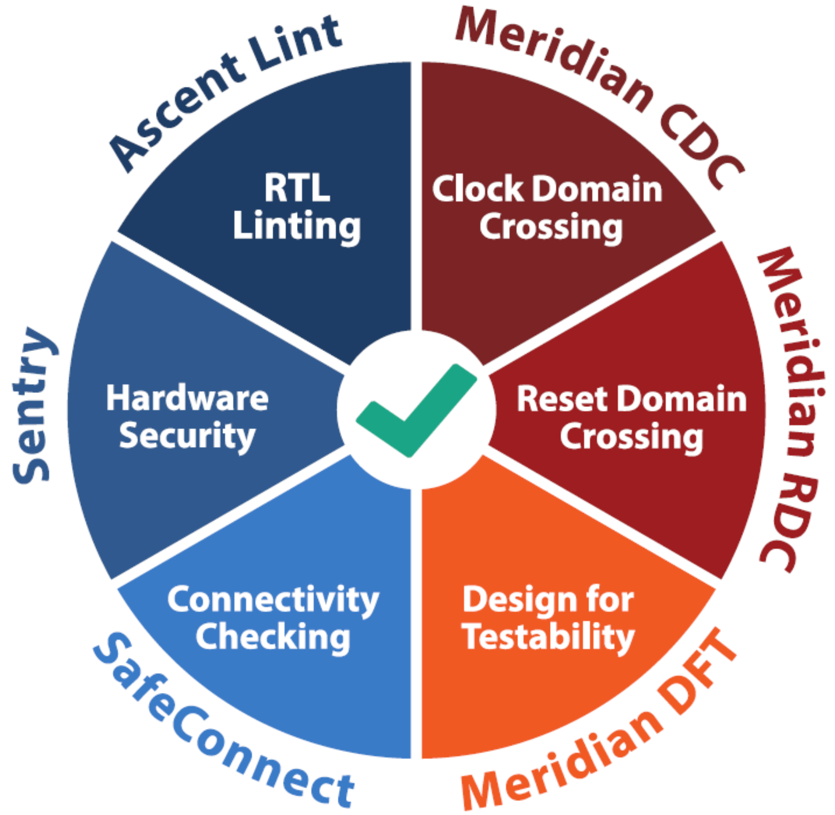
Formal Verification

Simulation and Formal have their role to play

→ Use Static Signoff effectively and judiciously, to make verification faster, more predictable, and tapeout chips without risk

Signoff with Static Minimally Boolean





Liberates from
Boolean shackles

For more information and content please contact us at ri-sales-il@realintent.com

REAL INTENT