Liberating Verification from Boolean Shackles

Vikas Sachdeva
Head of Business, APAC and Europe
Varun Sharma (Hardware Security Expert) & Saurav Choudhary (Connectivity Expert)
Real Intent

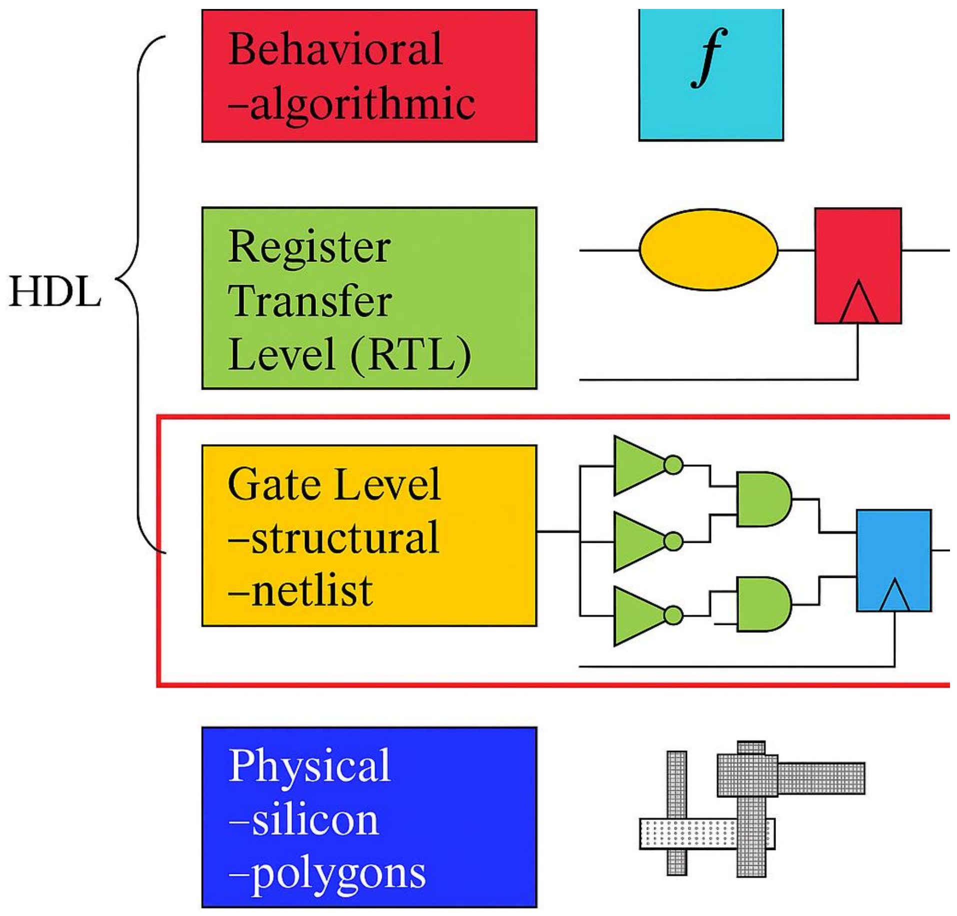
Vikas Sachdeva is the Head of Business, **APAC & Europe** at **Real Intent**, where he drives product strategy for the company's flagship static signoff solutions. He leads global product management and key customer accounts, playing a pivotal role in ensuring customer success and expanding Real Intent's footprint across geographies.

An alumnus of the **Indian Institute of Technology, Delhi**, Vikas is passionate about technology, product innovation, and building ecosystems that scale. Beyond his corporate leadership, he is the author of the Amazon bestseller *Becoming Irreplaceable*, where he shares practical insights on personal growth and professional impact.

Liberating Verification from Boolean Shackles

Vikas Sachdeva
Head of Business, APAC and Europe
Varun Sharma (Hardware Security Expert) & Saurav Choudhary (Connectivity Expert)
Real Intent

Abstraction in Hardware Design



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).

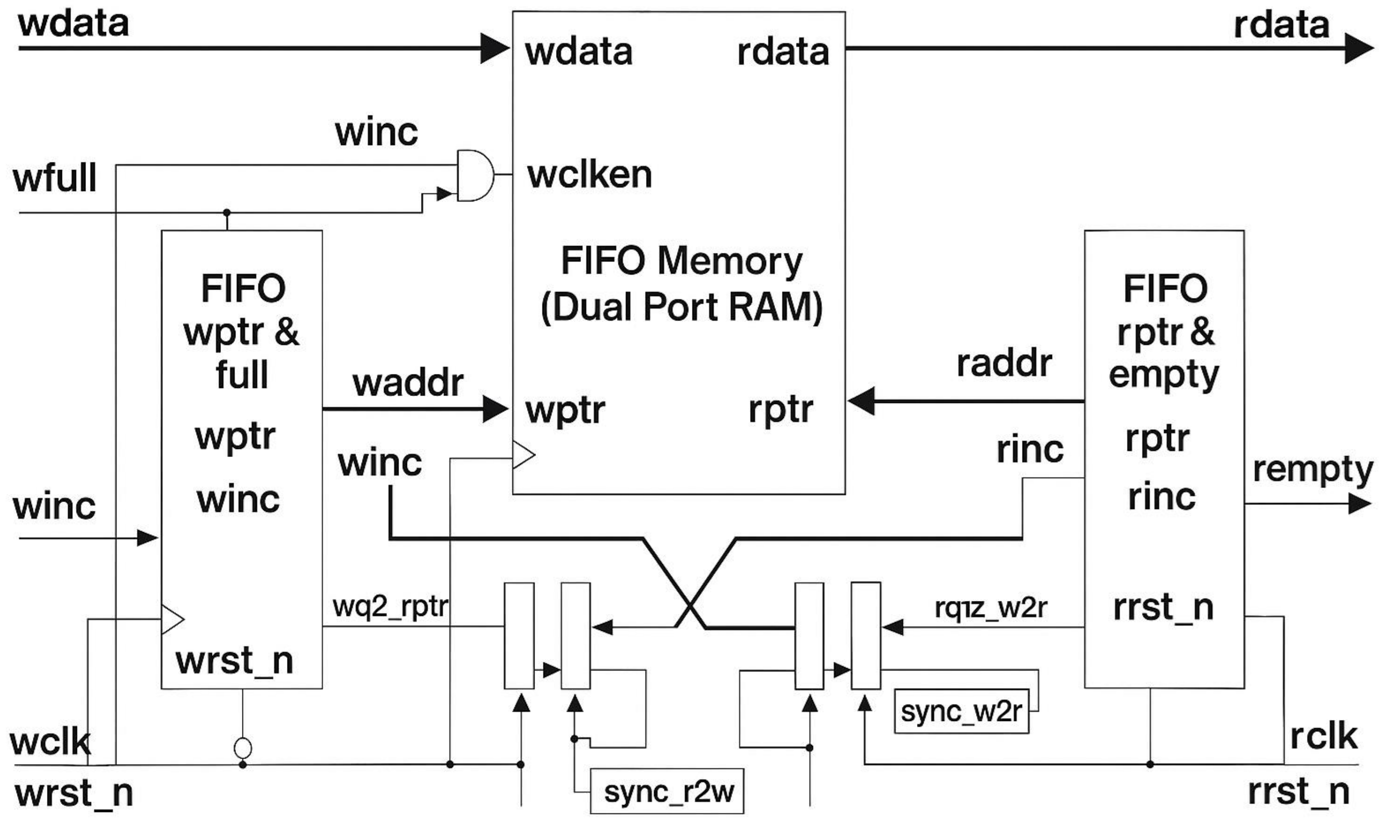
Early RTL design needs a different approach

Static Sign-off: Minimally Boolean

Liberates from Boolean shackles

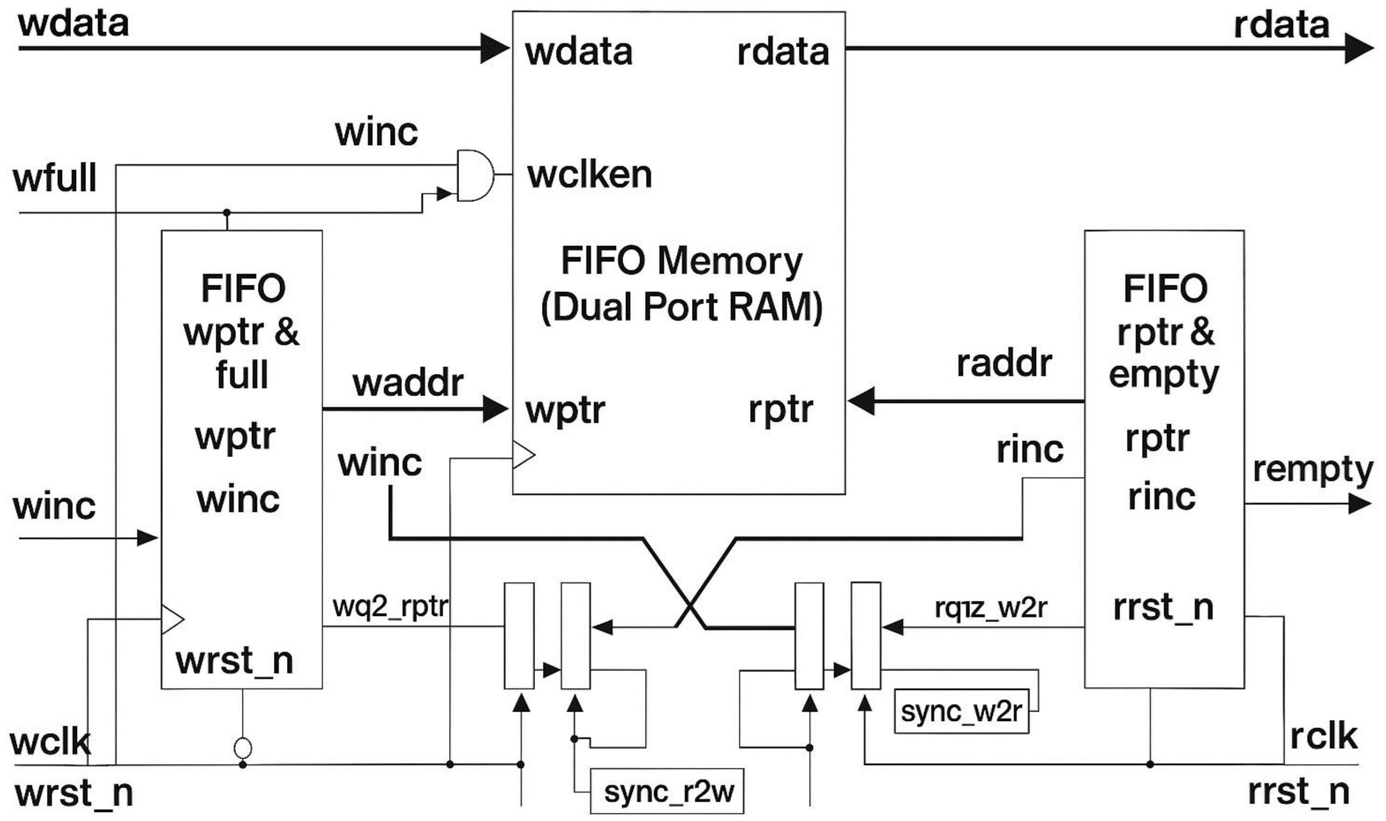
Asynchronous FIFO - Simulation

- Testbench drives input signals
- Outputs and internal signals are observed
- Coverage checks are Boolean



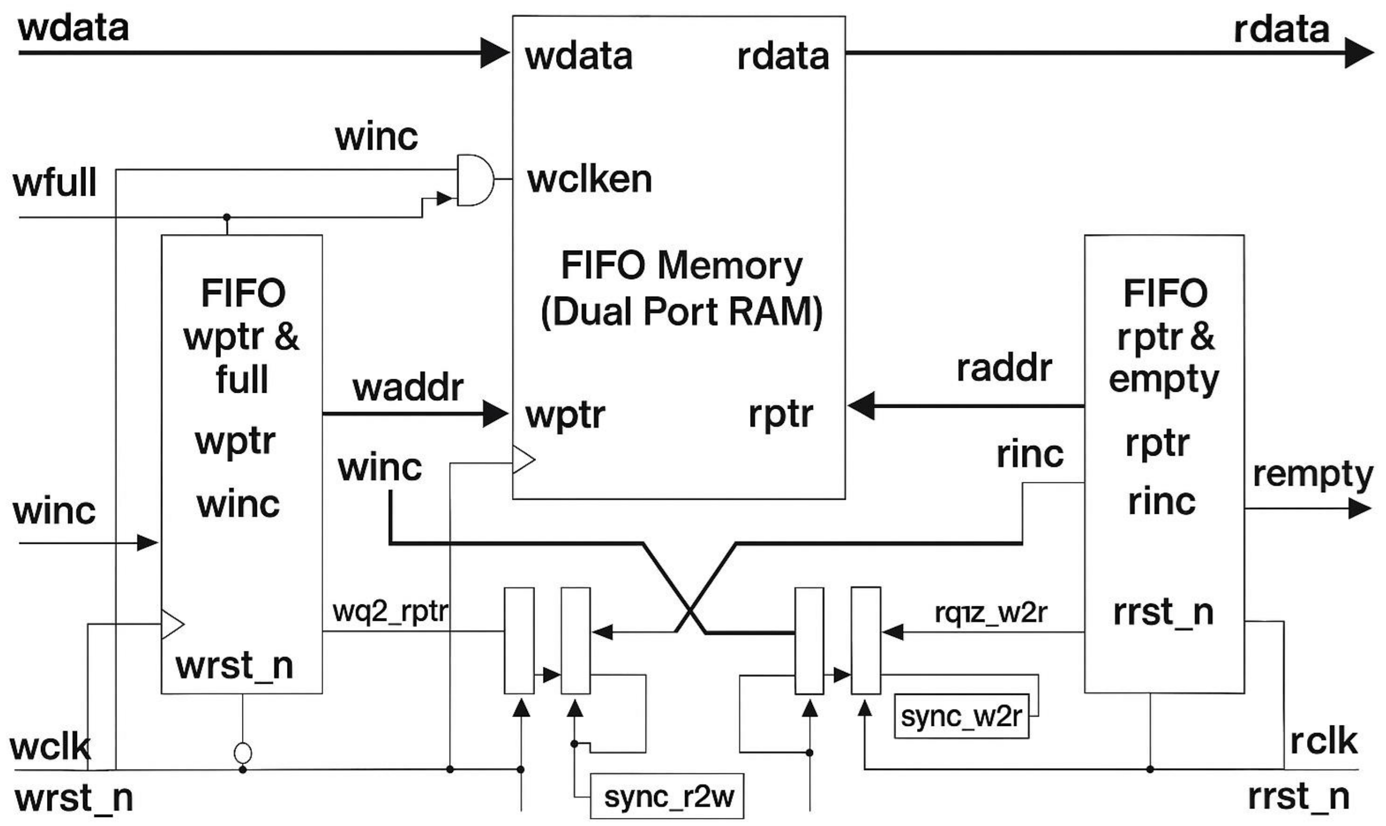
Asynchronous FIFO - Formal

- Assertions are created
- The tool turns these into Boolean constraints + SAT queries.
- Analysis is still mostly Boolean
- State space explosion happens on large size designs



Asynchronous FIFO – Static Analysis

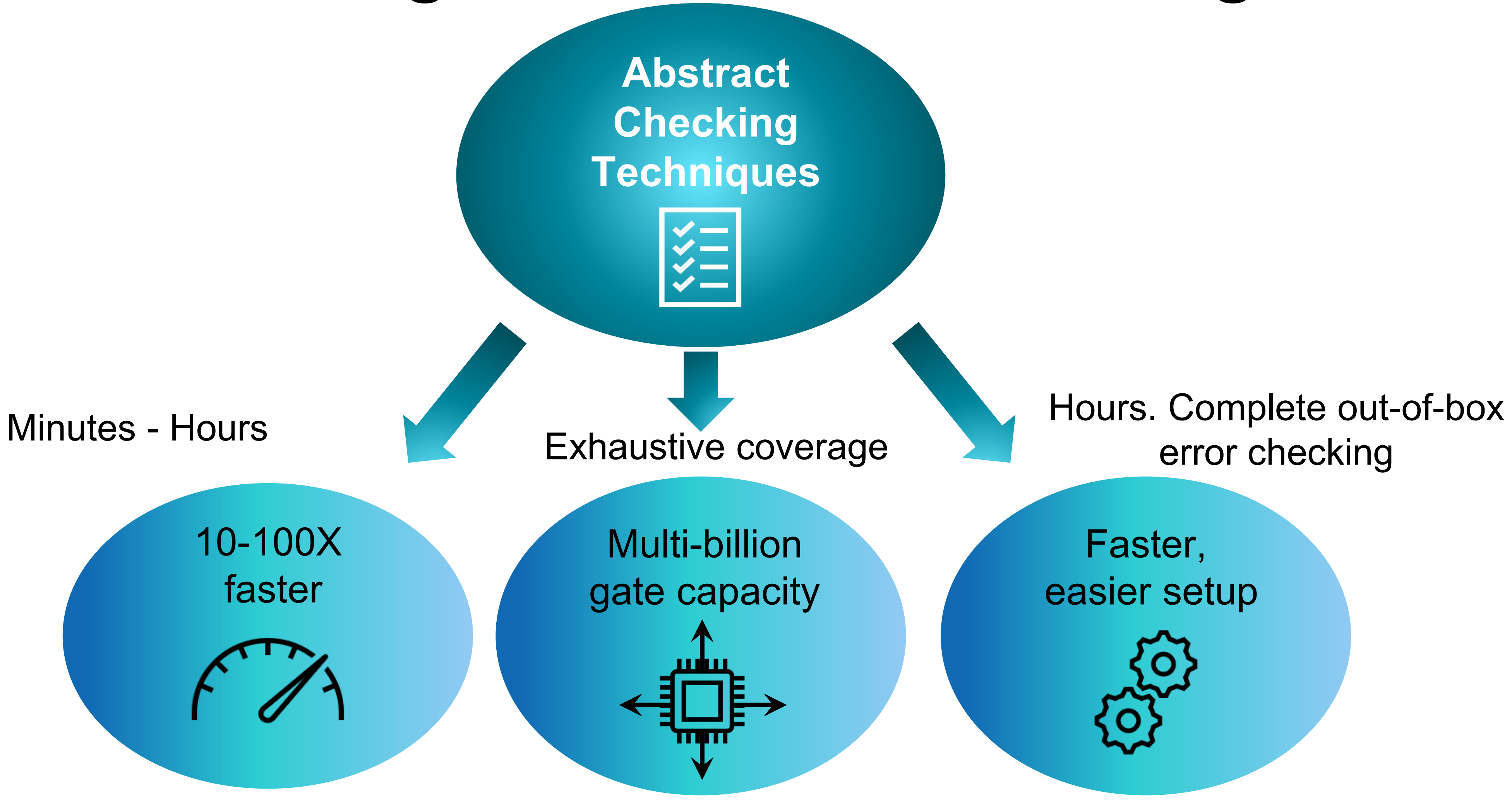
- Static analysis automatically identifies abstract structures
  - RAM
  - Synchronizers
  - Pointer comparison
- Verifies correctness at abstract level
- **Minimally Boolean!**



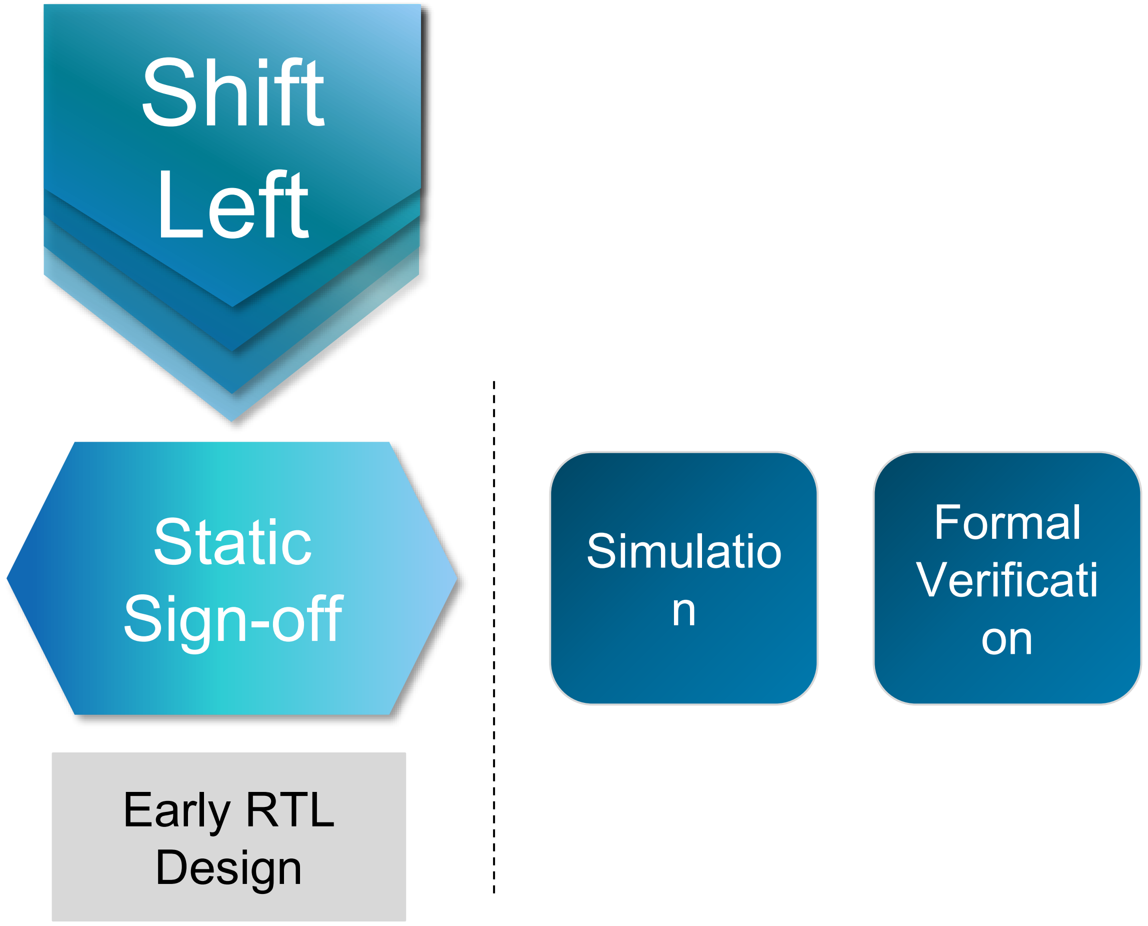
Static vs Simulation vs Formal Comparison

Simulation	Formal	Static
Testbench Needed	Assertions Needed	**Automatic based on abstract structures**
Will miss issues (not comprehensive)	Will miss issues (inconclusive)	**Always catch**
Long time to setup (write appropriate large number of testbenches)	Long time to setup (write appropriate assertions and then assumptions and wiggling to find real failures)	**Fast setup**
Long runtime	Long runtime	**Fast runtime**
No expert user needed	Expert User needed	**No expert user needed**
Large number of resources needed (compute and human)	Large number of resources needed (compute and expert human)	**Low compute and human resources**
Run on any size	Cannot run at top	**Run on any size**

Static sign-Off: Abstract checking


Abstract Checking Techniques
Minutes - Hours
Exhaustive coverage
Hours. Complete out-of-box error checking
10-100X faster
Multi-billion gate capacity
Faster, easier setup

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
  - Faster time-to-market: shorten design cycle
  - Cost reduction: fixing issues late / missed failures; both very costly
  - Risk reduction: catch failures, security vulnerabilities for robust design

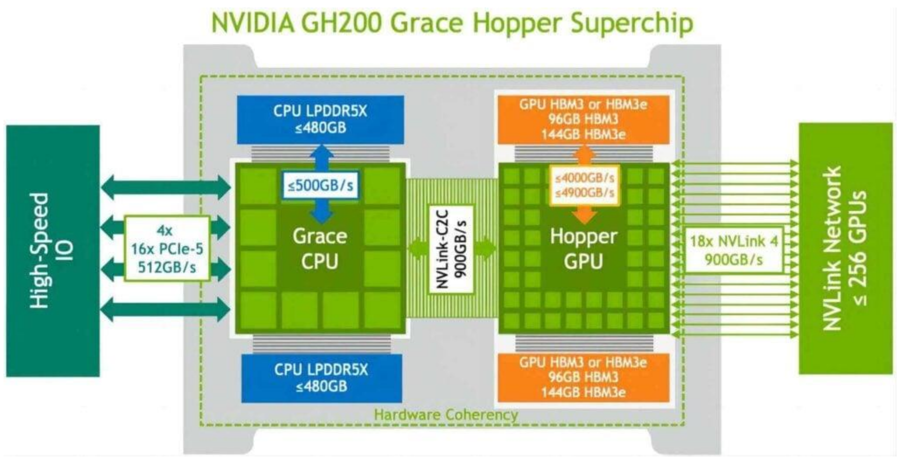
Explosive Chip Market Growth with AI

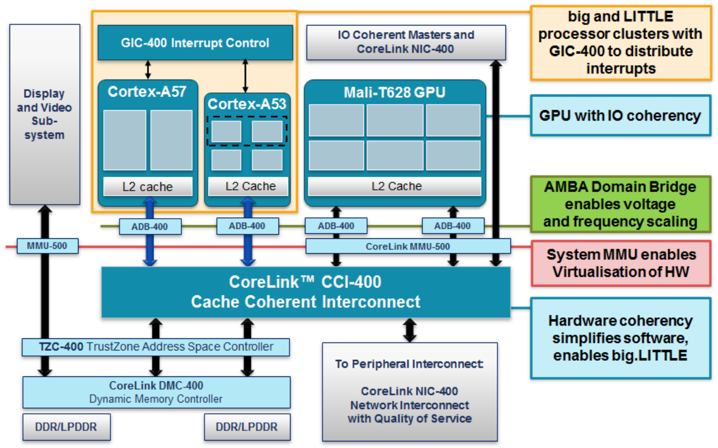
AI Datacenters

Edge computing and AI Accelerators

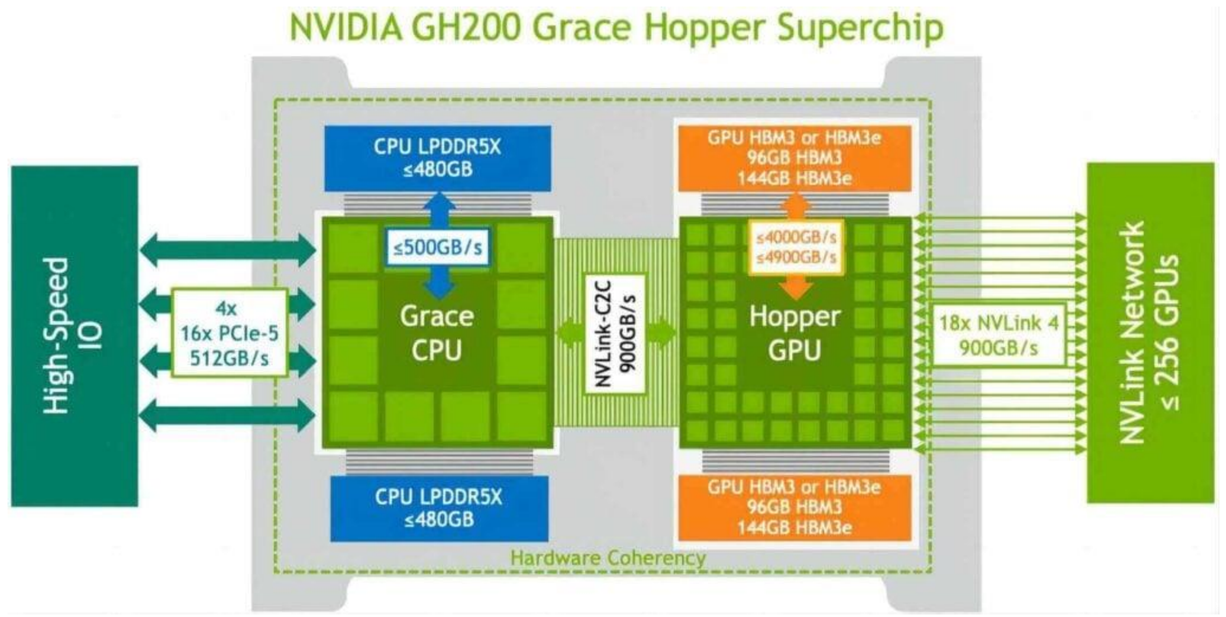
On-Device AI

Verification Challenges of these Chips



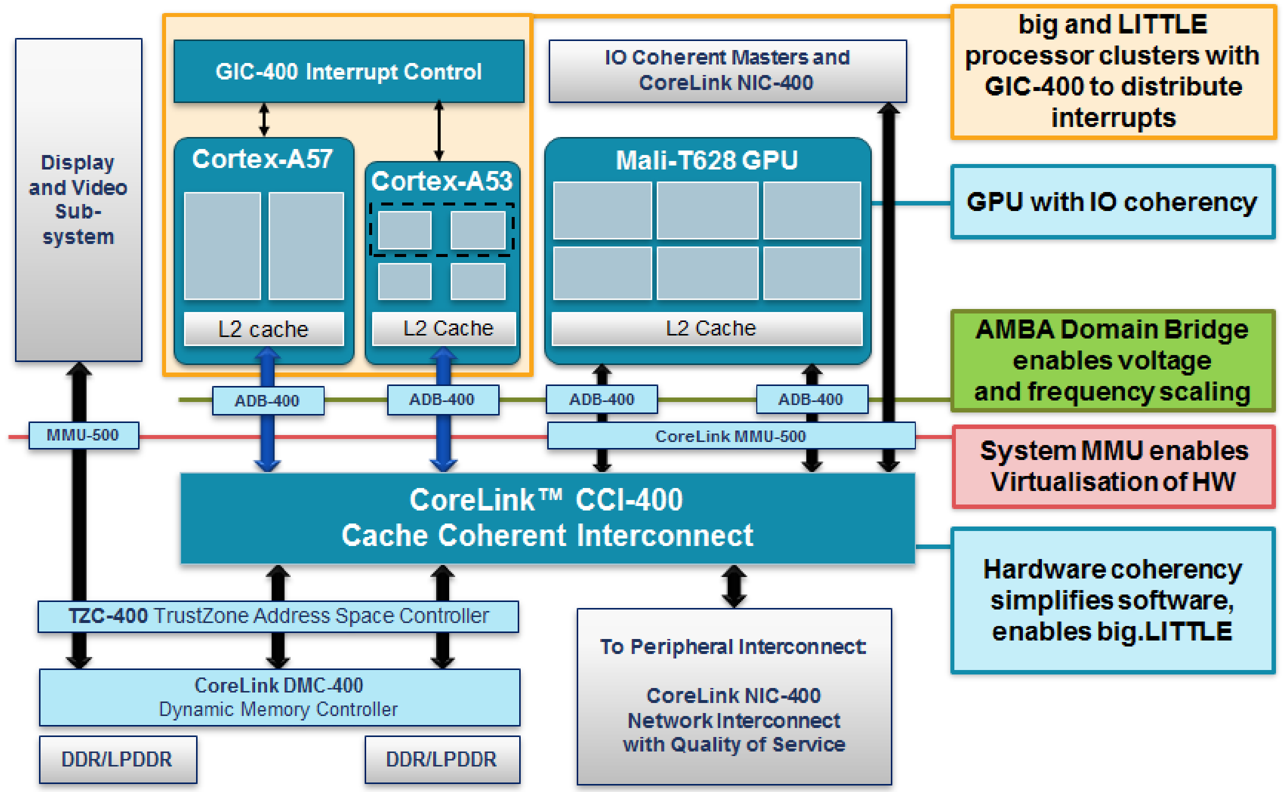

Display and Video Sub-system
GIC-400 Interrupt Control
Cortex-A57
L2 cache
Cortex-A53
L2 Cache
IO Coherent Masters and CoreLink NIC-400
Mali-T628 GPU
L2 Cache
big and LITTLE processor clusters with GIC-400 to distribute interrupts
GPU with IO coherency
ADB-400
ADB-400
ADB-400
ADB-400
AMBA Domain Bridge enables voltage and frequency scaling
MMU-500
CoreLink MMU-500
System MMU enables Virtualisation of HW
CoreLink™ CCI-400 Cache Coherent Interconnect
Hardware coherency simplifies software, enables big.LITTLE
TZC-400 TrustZone Address Space Controller
CoreLink DMC-400 Dynamic Memory Controller
DDR/LPDDR
DDR/LPDDR
To Peripheral Interconnect
CoreLink NIC-400 Network Interconnect with Quality of Service

Verification Challenges AI Chips



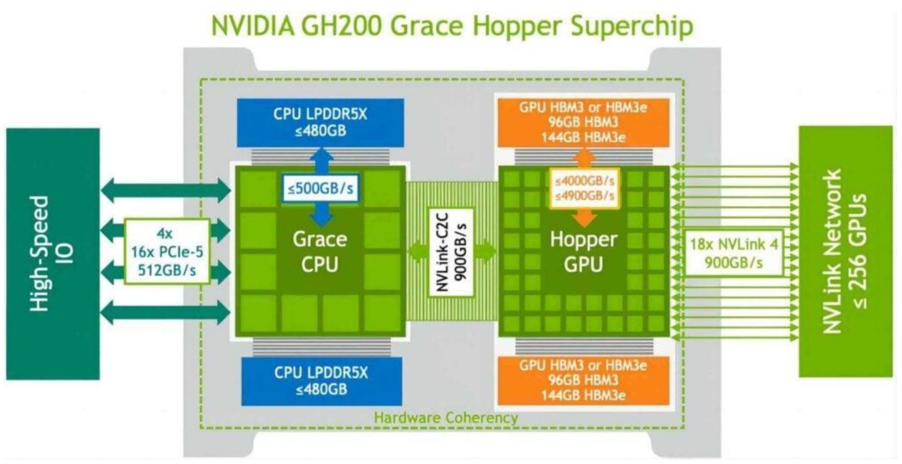
- Massive Asynchronous paths and power domain verification – CPU-GPU-HBM
- CPU + GPU + High-Bandwidth Memories (HBM3/3e + LPDDR5X)
  - Protocol correctness
  - Cache coherency
  - Memory consistency
- NVLink-C2C
  - Interconnect verification
  - Topology correctness routing, congestion …
- PCIe Gen5, 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

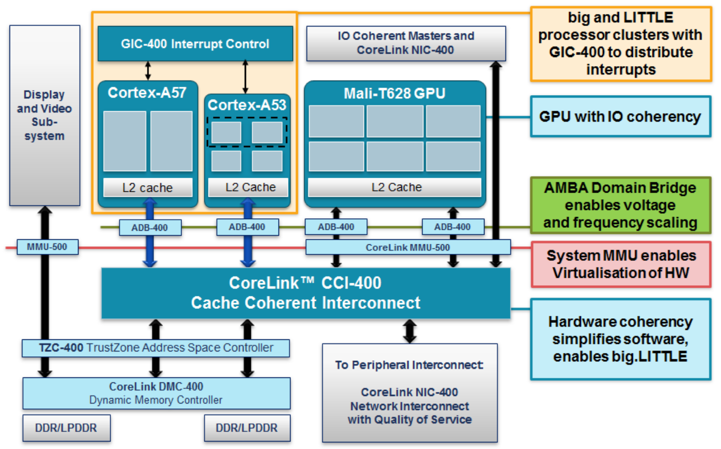
Verification Challenges Non-AI Chips


Display and Video Sub-system
GIC-400 Interrupt Control
IO Coherent Masters and CoreLink NIC-400
big and LITTLE processor clusters with GIC-400 to distribute interrupts
Cortex-A57
Cortex-A53
Mali-T628 GPU
GPU with IO coherency
L2 cache
L2 Cache
L2 Cache
AMBA Domain Bridge enables voltage and frequency scaling
ADB-400
ADB-400
ADB-400
ADB-400
MMU-500
CoreLink MMU-500
System MMU enables Virtualisation of HW
CoreLink™ CCI-400 Cache Coherent Interconnect
Hardware coherency simplifies software, enables big.LITTLE
TZC-400 TrustZone Address Space Controller
To Peripheral Interconnect
CoreLink NIC-400 Network Interconnect with Quality of Service
CoreLink DMC-400 Dynamic Memory Controller
DDR/LPDDR
DDR/LPDDR

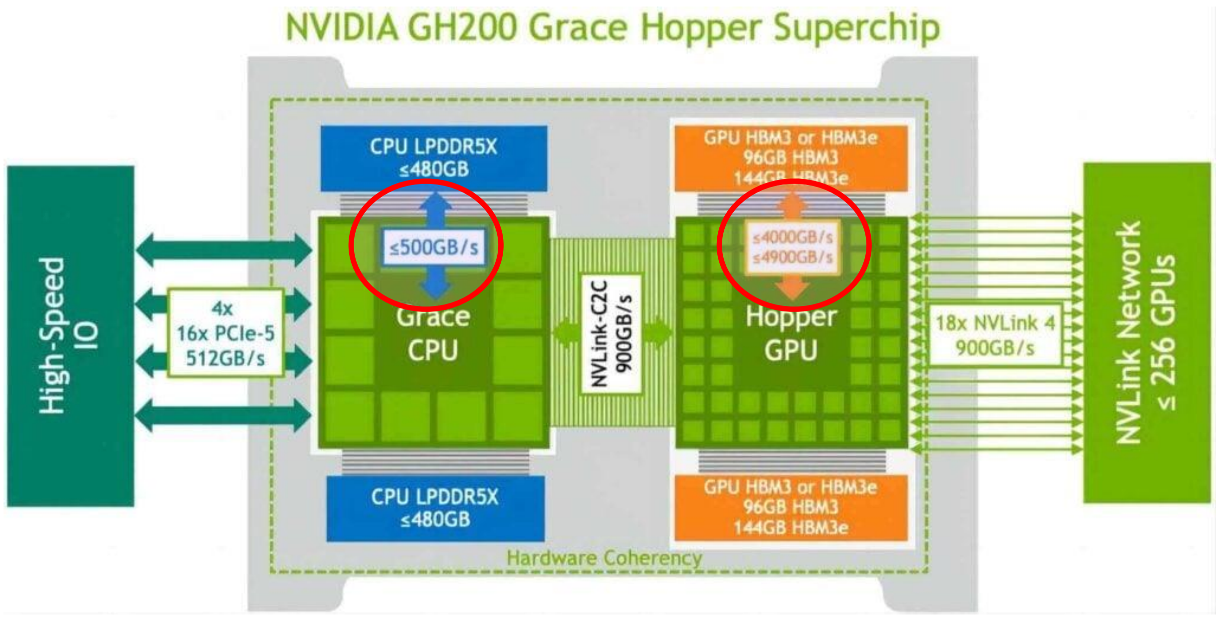
- Cortex-A57/A53 clusters
  - CPU Verification
- CoreLink CCI-400 cache coherent interconnect
  - Interconnect Verification
- GIC interrupt controller
  - Correct distribution between big and LITTLE clusters.
- Security (TrustZone, TZC-400)
  - World separation: Normal world vs Secure world must be strictly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

Asynchronous Path Verification Challenges



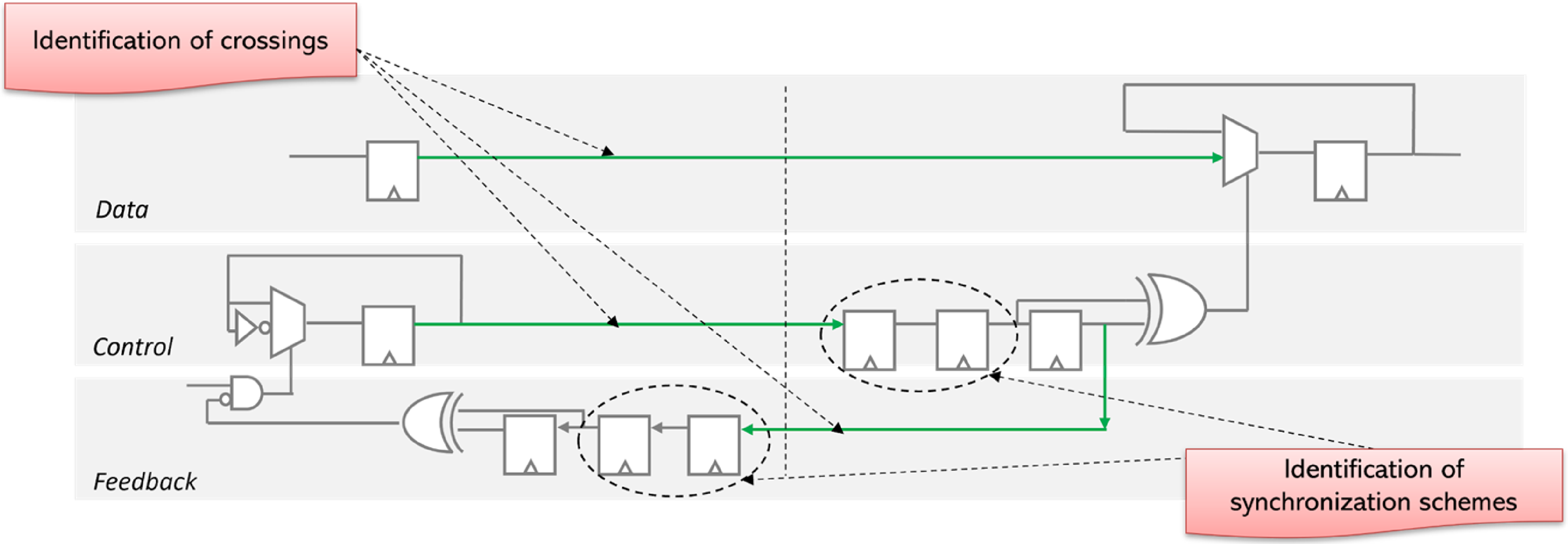

Display and Video Sub-system
GIC-400 Interrupt Control
Cortex-A57
L2 cache
Cortex-A53
L2 Cache
IO Coherent Masters and CoreLink NIC-400
Mali-T628 GPU
L2 Cache
big and LITTLE processor clusters with GIC-400 to distribute interrupts
GPU with IO coherency
ADB-400
ADB-400
ADB-400
ADB-400
AMBA Domain Bridge enables voltage and frequency scaling
MMU-500
CoreLink MMU-500
System MMU enables Virtualisation of HW
CoreLink™ CCI-400 Cache Coherent Interconnect
Hardware coherency simplifies software, enables big.LITTLE
TZC-400 TrustZone Address Space Controller
CoreLink DMC-400 Dynamic Memory Controller
DDR/LPDDR
DDR/LPDDR
To Peripheral Interconnect:
CoreLink NIC-400 Network Interconnect with Quality of Service

Asynchronous Path Verification Challenges



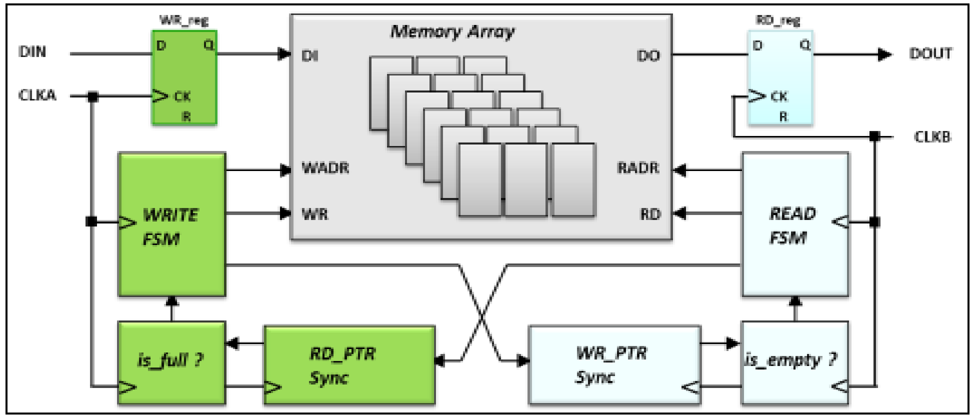
- Multi clock domains verification due to power requirements
- Handshake functional correctness
- Safety from Glitch Hazards
- Functional correctness under metastability
- No reset metastability

Handshake Functional Correctness

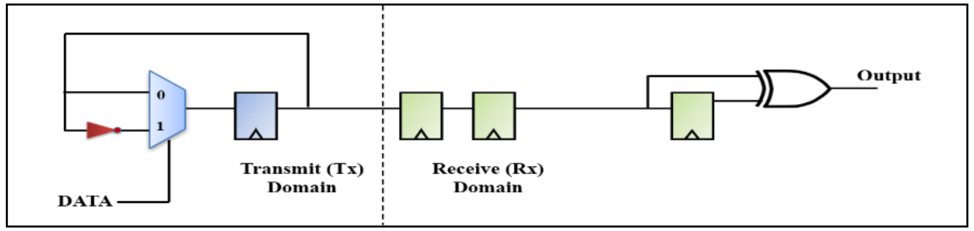

Identification of crossings
Data
Control
Feedback
Identification of synchronization schemes

- Interpret underlying functional intent in multimode context
- Accurately identify CDC issues when crossings can be synchronous in some modes or asynchronous in some modes
- Automatically and in one run

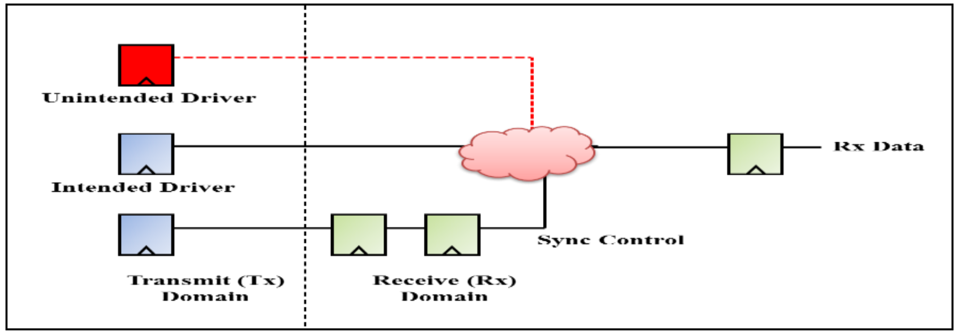
Handshake Functional Correctness



FIFO Handshake

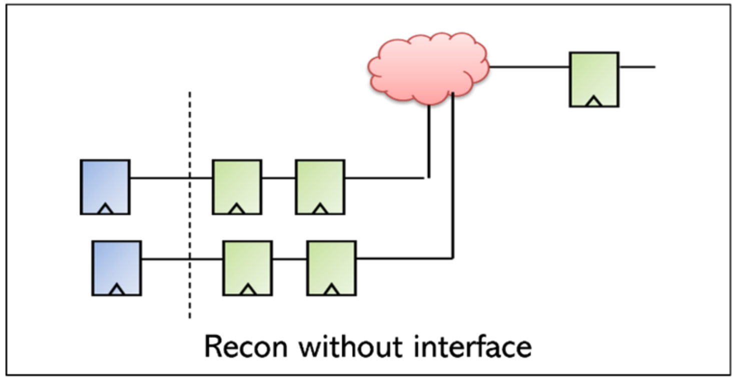


Auto-Identified Pulse Synchronizer

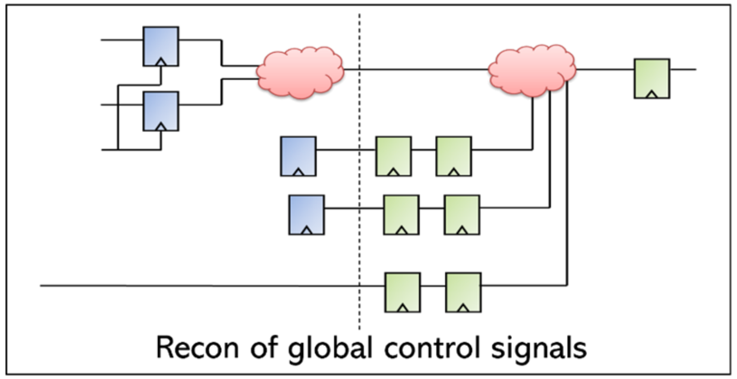


Unintended Driver – ERROR

Unsafe Reconvergence

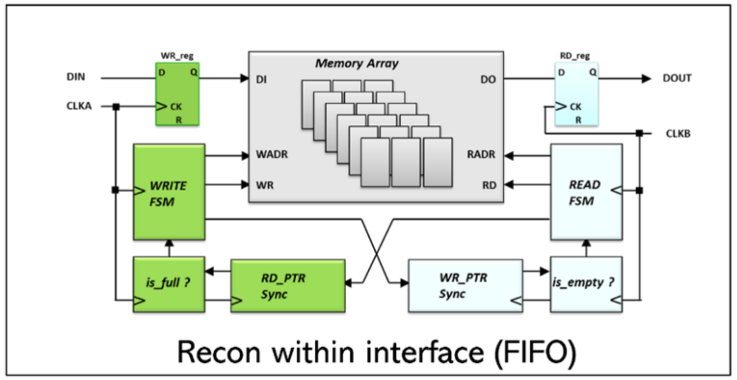


Recon without interface

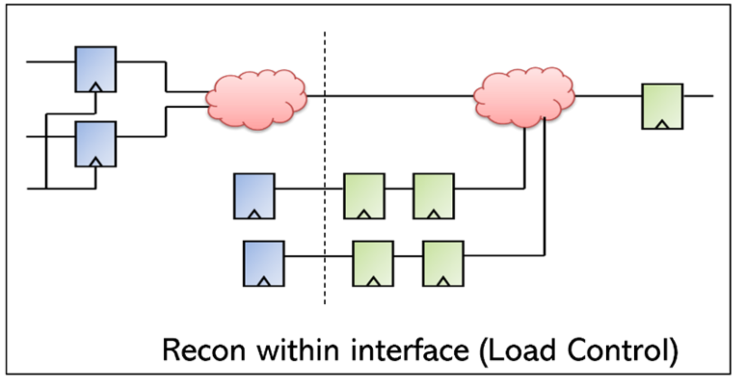


Recon of global control signals

Safe Reconvergence



Recon within interface (FIFO)

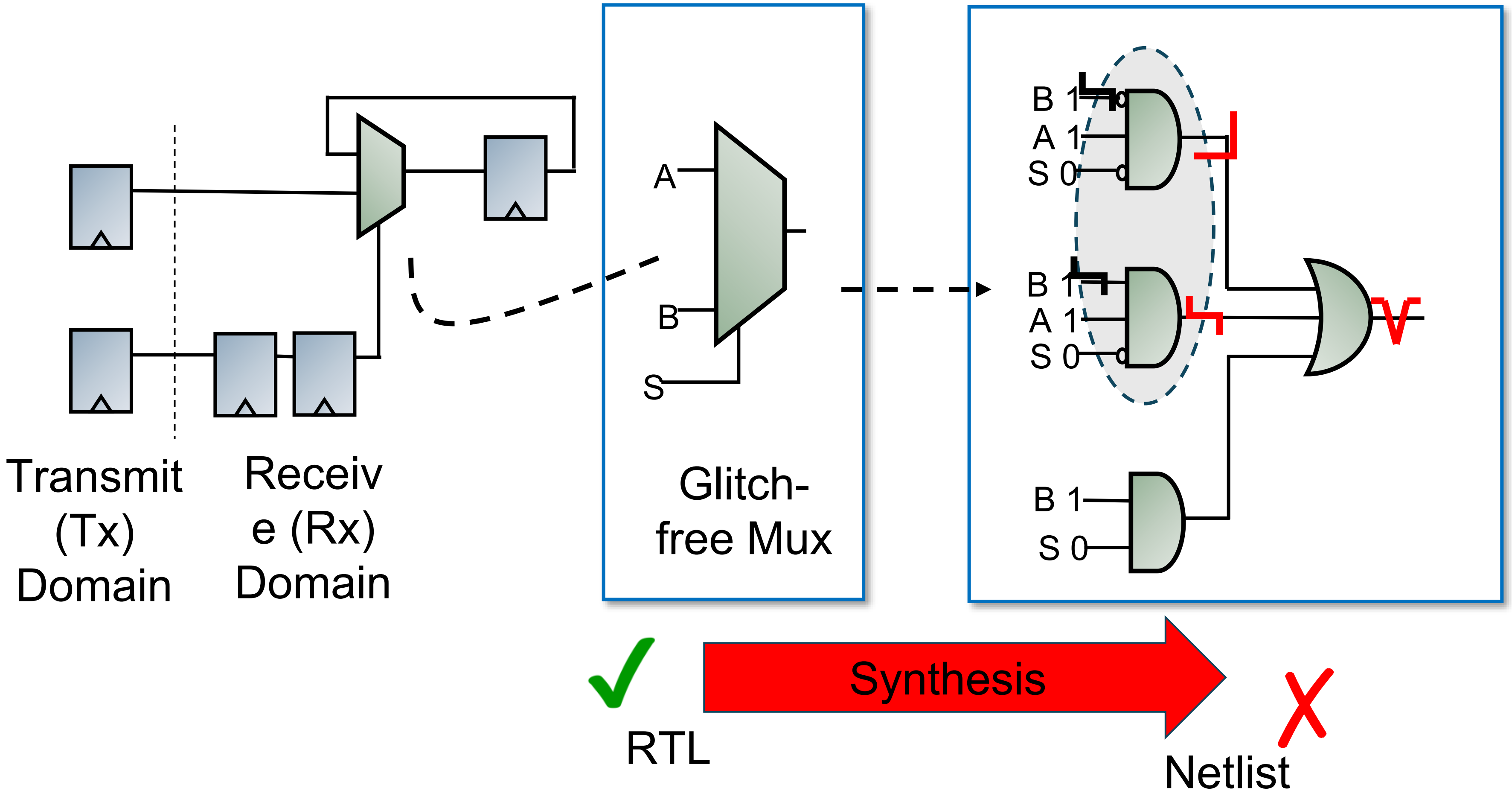


Recon within interface (Load Control)

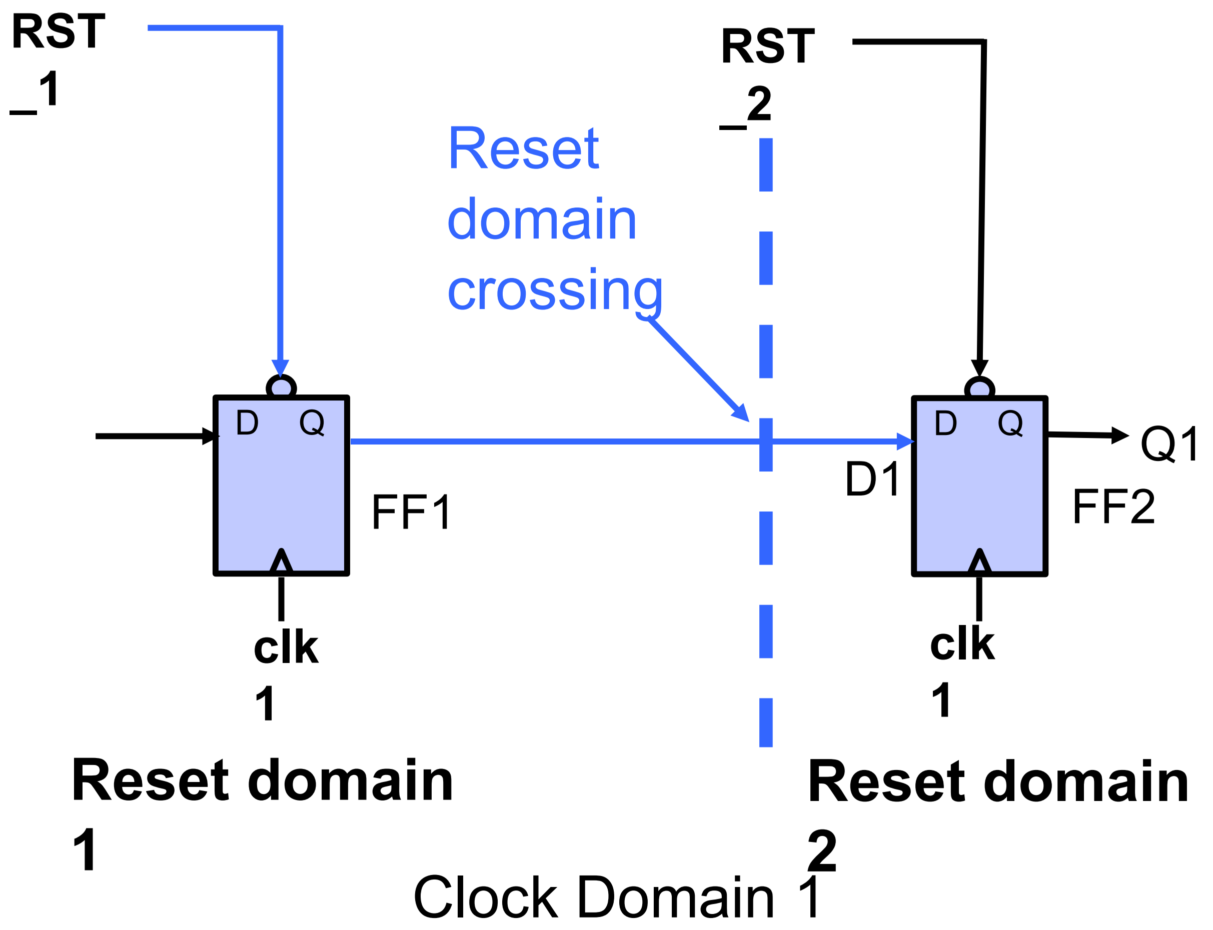
accellera SYSTEMS INITIATIVE

2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

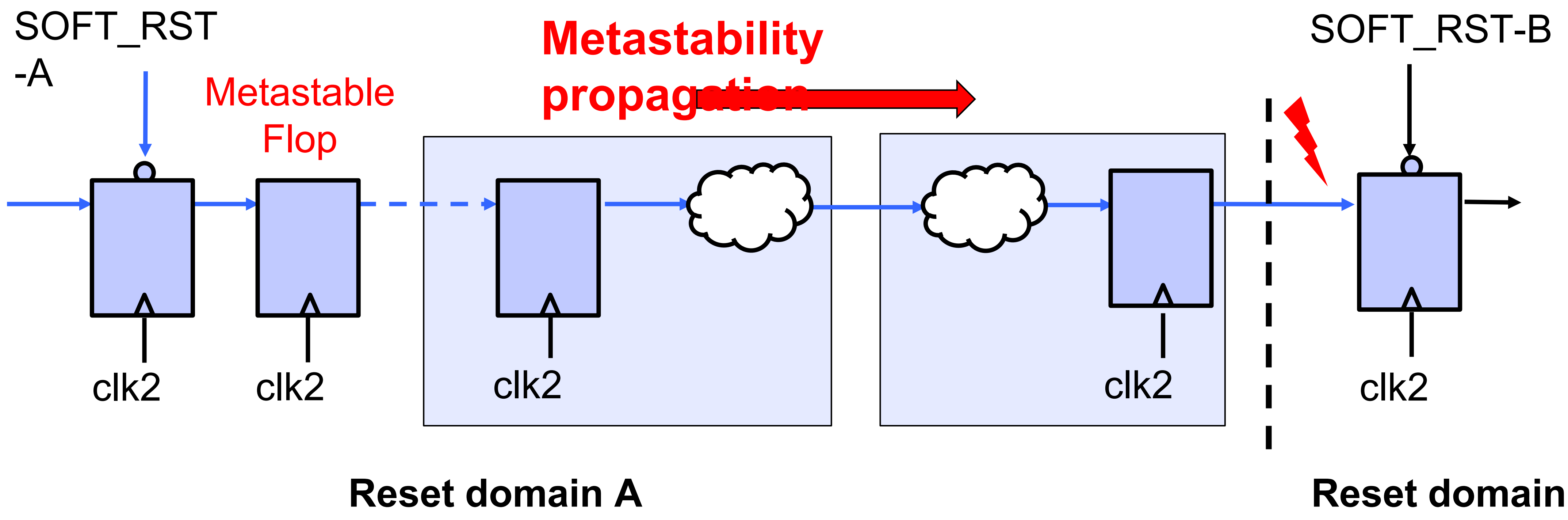
Safety from Glitch Hazards



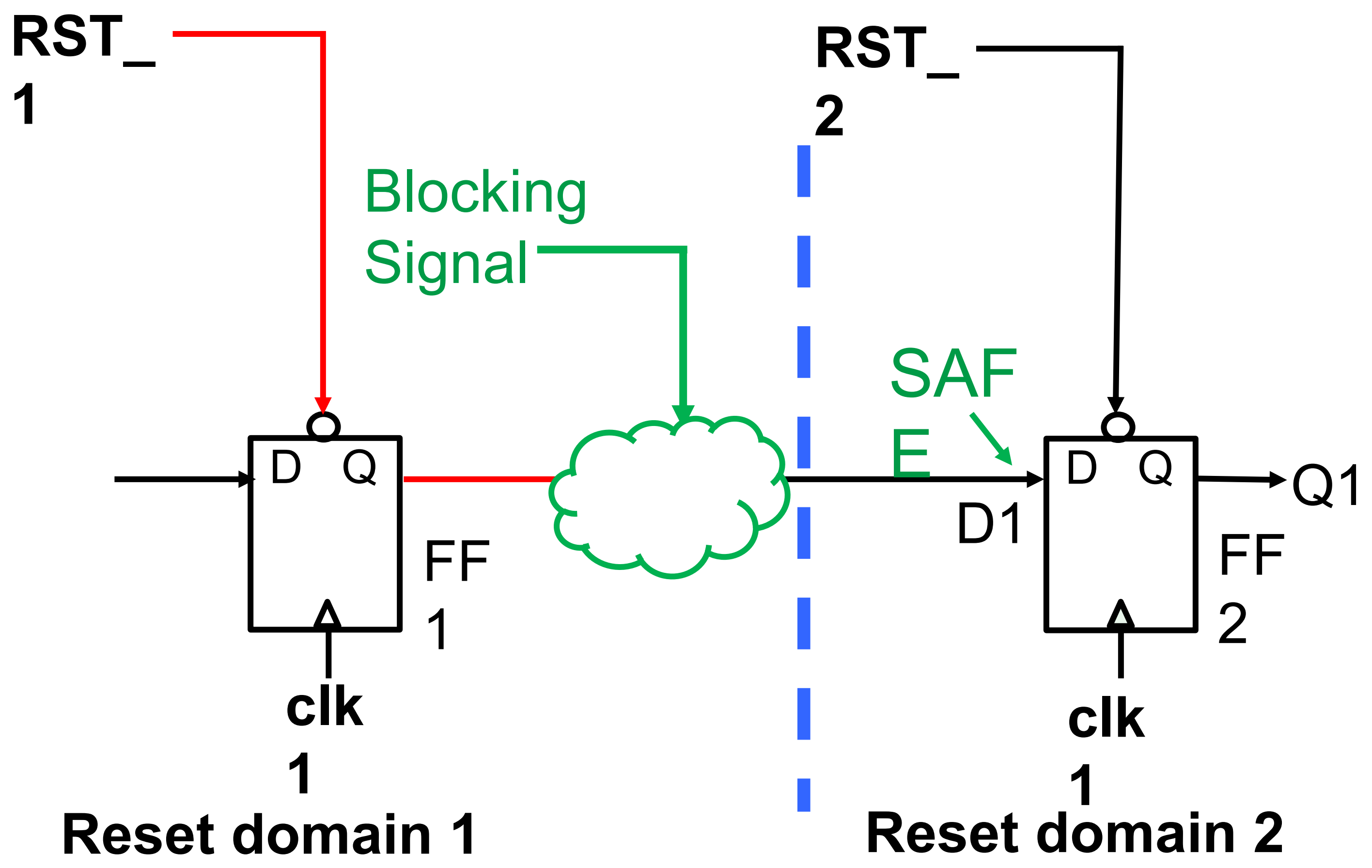
No Reset Metastability



No Reset Metastability

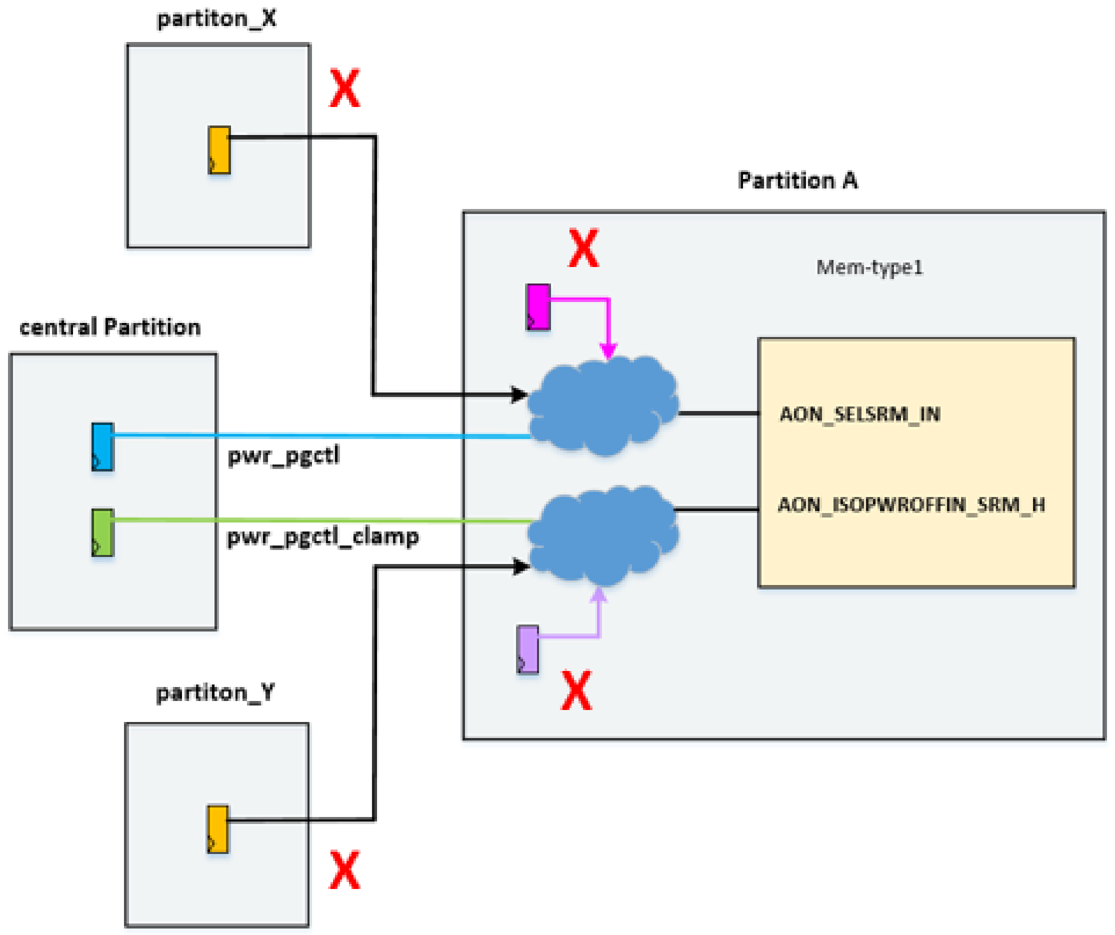


No Reset Metastability

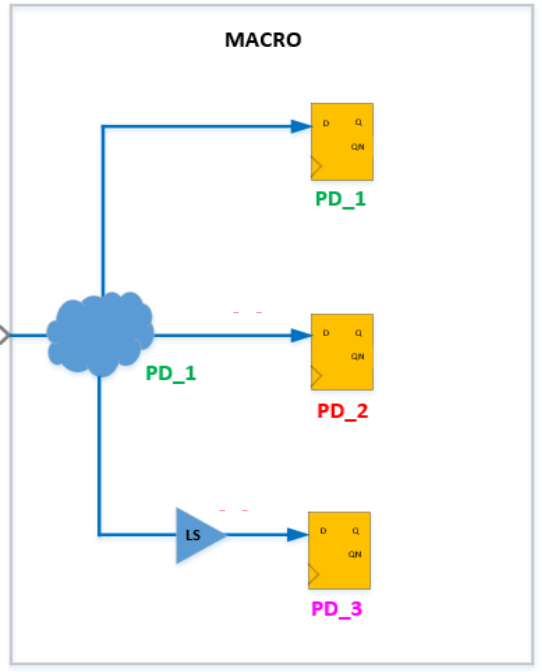


Power Logic Verification

Check memory inputs in a partition are driven without any extra connections by power and clamp pgctl pins respectively from one central partition



Power Isolation Verification

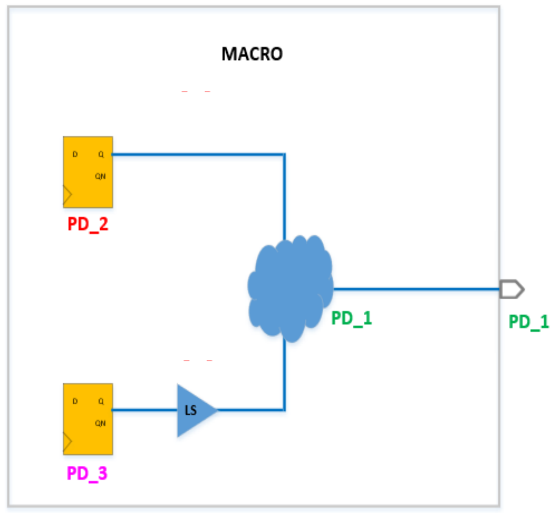


Check that any input of the macro can only drive FFs in the same power domain

- create_group –name GRP1 –module {MACRO} –all_inputs
- trace_all_paths –rule POWER_ISOLATION_DIFFERENT_PD
  –from_group GRP1
  –source_pd

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

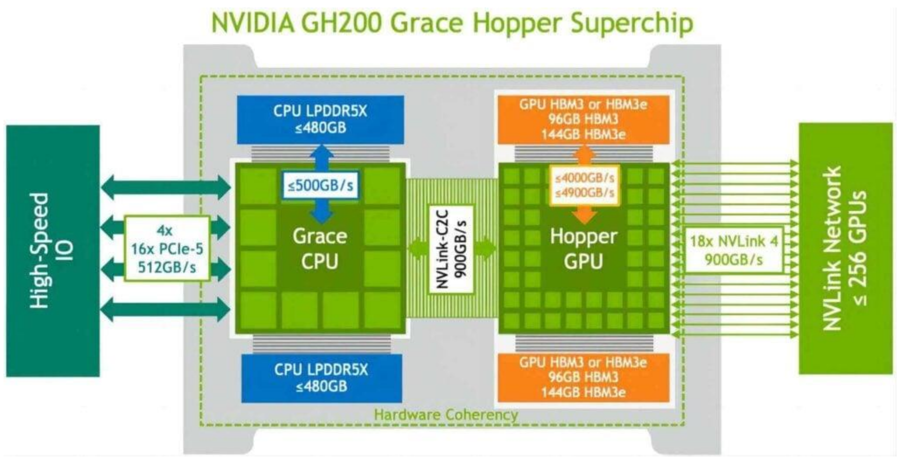
- create_group –name GRP2 –module {MACRO} –all_outputs
- trace_all_paths –rule POWER_ISOLATION_DIFFERENT_PD
  –to_group GRP2
  –destination_pd

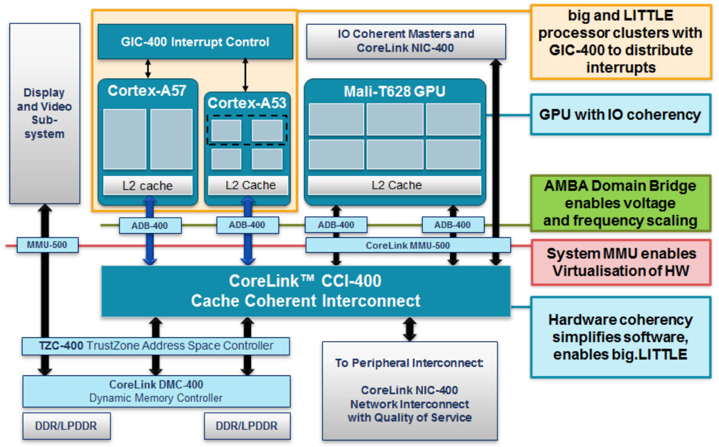


Automatic & Low-noise Static Tools are Imperative to Success

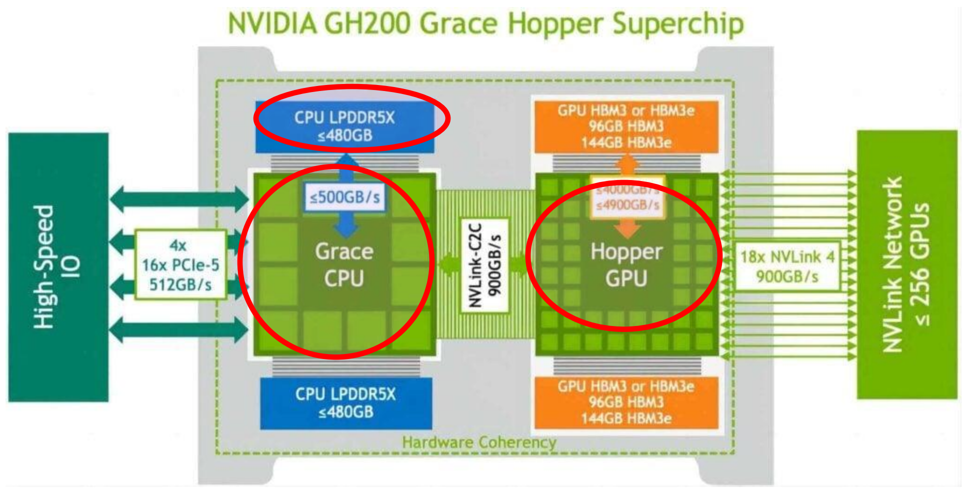
- Automatic setup enable out of the box setup and reduce cost
- Automatic checks speed up bug detection
- Automatic checks mitigate risks early through **scalable**, **systematic verification** — from individual IP blocks to **full-chip** designs.
- Low noise is imperative as large noise discourages detailed analysis can causes issues to slip through

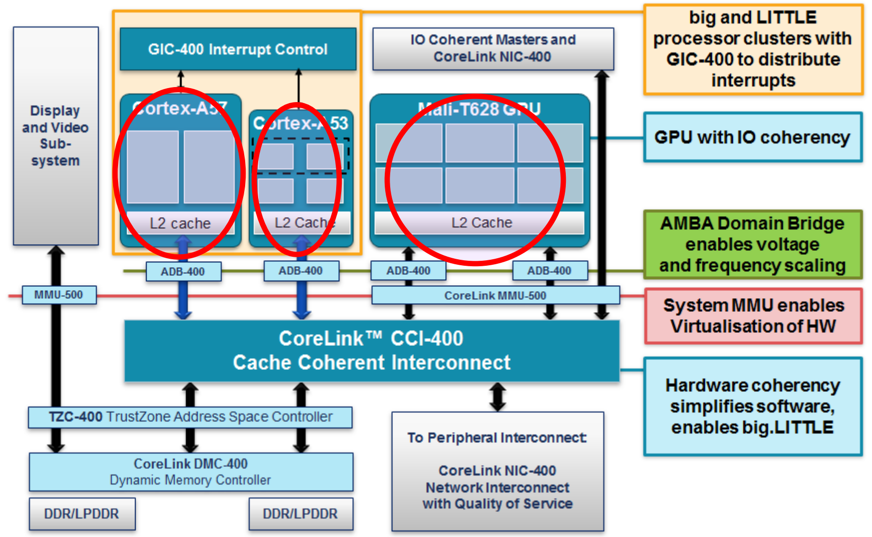
CPU/GPU Verification Challenges




Display and Video Sub-system
GIC-400 Interrupt Control
Cortex-A57
L2 cache
Cortex-A53
L2 Cache
IO Coherent Masters and CoreLink NIC-400
Mali-T628 GPU
L2 Cache
big and LITTLE processor clusters with GIC-400 to distribute interrupts
GPU with IO coherency
ADB-400
ADB-400
ADB-400
ADB-400
AMBA Domain Bridge enables voltage and frequency scaling
MMU-500
CoreLink MMU-500
System MMU enables Virtualisation of HW
CoreLink™ CCI-400 Cache Coherent Interconnect
Hardware coherency simplifies software, enables big.LITTLE
TZC-400 TrustZone Address Space Controller
CoreLink DMC-400 Dynamic Memory Controller
DDR/LPDDR
DDR/LPDDR
To Peripheral Interconnect:
CoreLink NIC-400 Network Interconnect with Quality of Service

CPU/GPU Verification Challenges




GIC-400 Interrupt Control
IO Coherent Masters and CoreLink NIC-400
big and LITTLE processor clusters with GIC-400 to distribute interrupts
Display and Video Sub-system
Cortex-A57
Cortex-A53
Mali-T628 GPU
GPU with IO coherency
L2 cache
L2 Cache
L2 Cache
AMBA Domain Bridge enables voltage and frequency scaling
ADB-400
ADB-400
ADB-400
ADB-400
MMU-500
CoreLink MMU-500
System MMU enables Virtualisation of HW
CoreLink™ CCI-400 Cache Coherent Interconnect
Hardware coherency simplifies software, enables big.LITTLE
TZC-400 TrustZone Address Space Controller
To Peripheral Interconnect
CoreLink DMC-400 Dynamic Memory Controller
CoreLink NIC-400 Network Interconnect with Quality of Service
DDR/LPDDR
DDR/LPDDR

- 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

Security Vulnerabilities Risk for All CPU Types

Meltdown and Spectre: 'worst ever' CPU bugs affect virtually all computers

Everything from smartphones and PCs to cloud computing affected by major security flaw found in Intel and other processors - and fix could slow devices

Security experts find vulnerability in ARM's memory tagging extensions

by Bob Yirka, Phys.org

ENDPOINT SECURITY

GhostWrite Vulnerability Facilitates Attacks on Devices With RISC-V CPU

[illegible], a RISC-V CPU vulnerability that can be exploited to gain full access to targeted devices.

New SLAP & FLOP Attacks Expose Apple M-Series Chips to Speculative Execution Exploits

Intel's latest security update has this 'big warning' for AMD and Nvidia customers

According to the report, AMD had "4.4x more firmware vulnerabilities in their hardware root-of-trust" and "1.8x more firmware vulnerabilities in their confidential computing technologies" compared to Intel. The company also criticised Nvidia in the GPU category, stating that it only had high-severity vulnerabilities (18) in 2024.

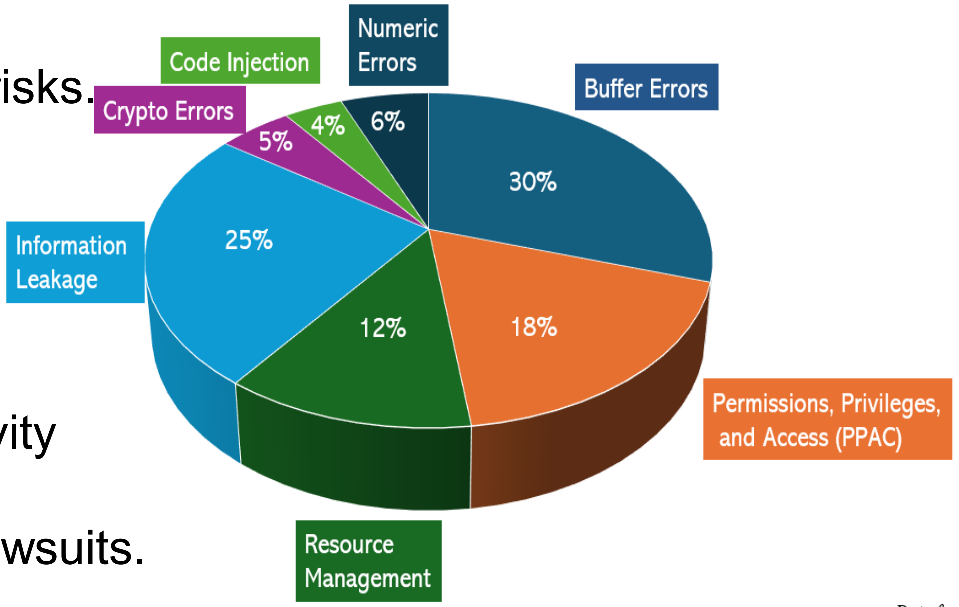
Security Becoming Core Part Of Chip Design — Finally

One of the big changes for hardware and system security is a recognition that it's no longer someone else's problem. What used to be an afterthought is now a competitive edge that needs to be baked into a design at the architectural level.

The High Cost of Security Vulnerabilities in Chips

Missing a security flaw can cost tens of millions of dollars—a critical business risk

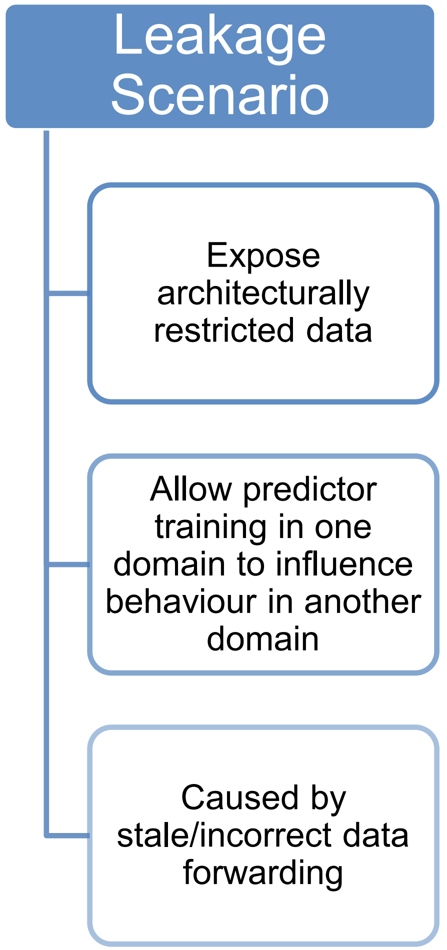
- Direct Financial Costs 💰
  - Cyberattack & Ransomware – Data breaches & extortion risks.
  - Remediation & Incident Response – Investigations & recovery efforts.
  - Revenue Loss – Business disruptions & customer fallout.
- Indirect Financial Costs ⚠️
  - Downtime & Operational Disruptions – Impacting productivity & supply chain.
  - Regulatory & Legal Penalties – Compliance violations & lawsuits.
- Strategic Consequences
  - Loss of Competitive Advantage – A compromised product damages innovation.
  - Erosion of Trust – Customer & partner confidence declines.



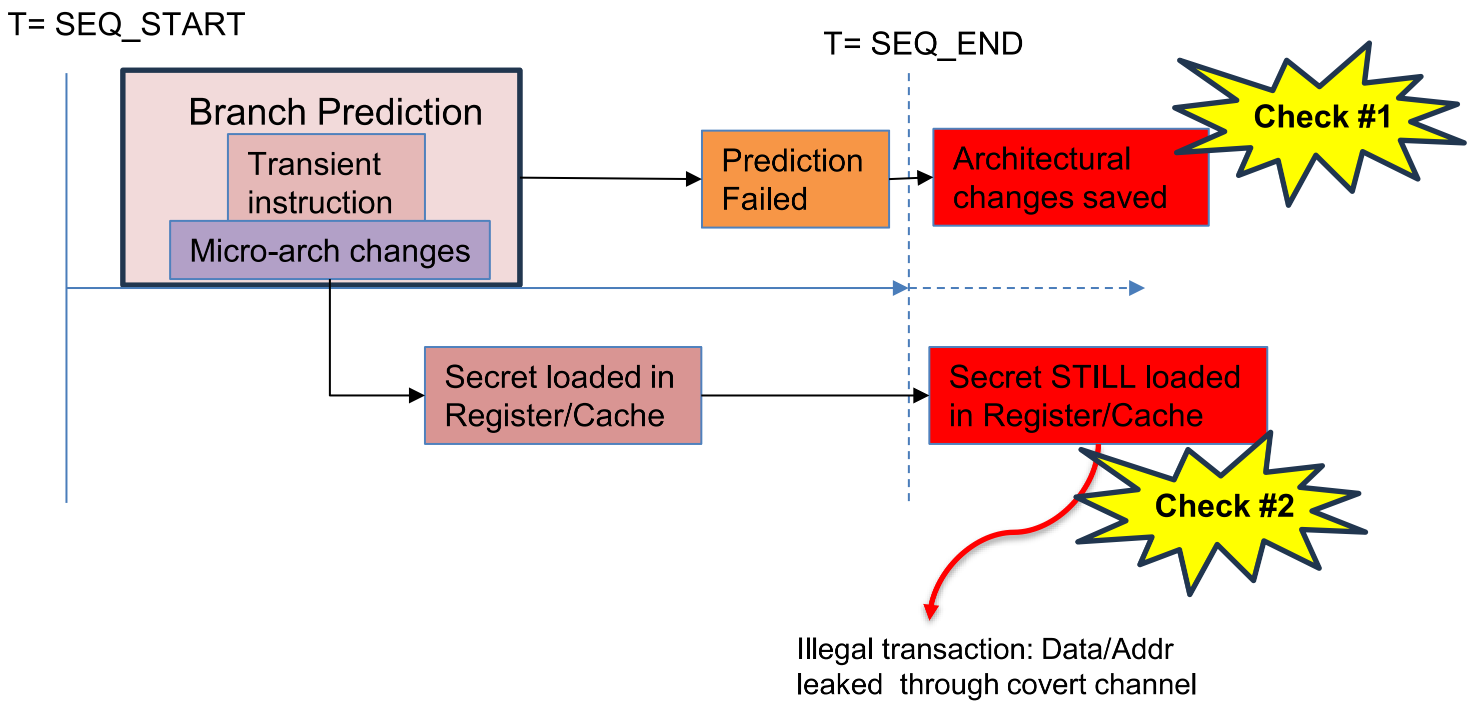
Source: Data from MITRE/NIST

Hardware security signoff is crucial

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

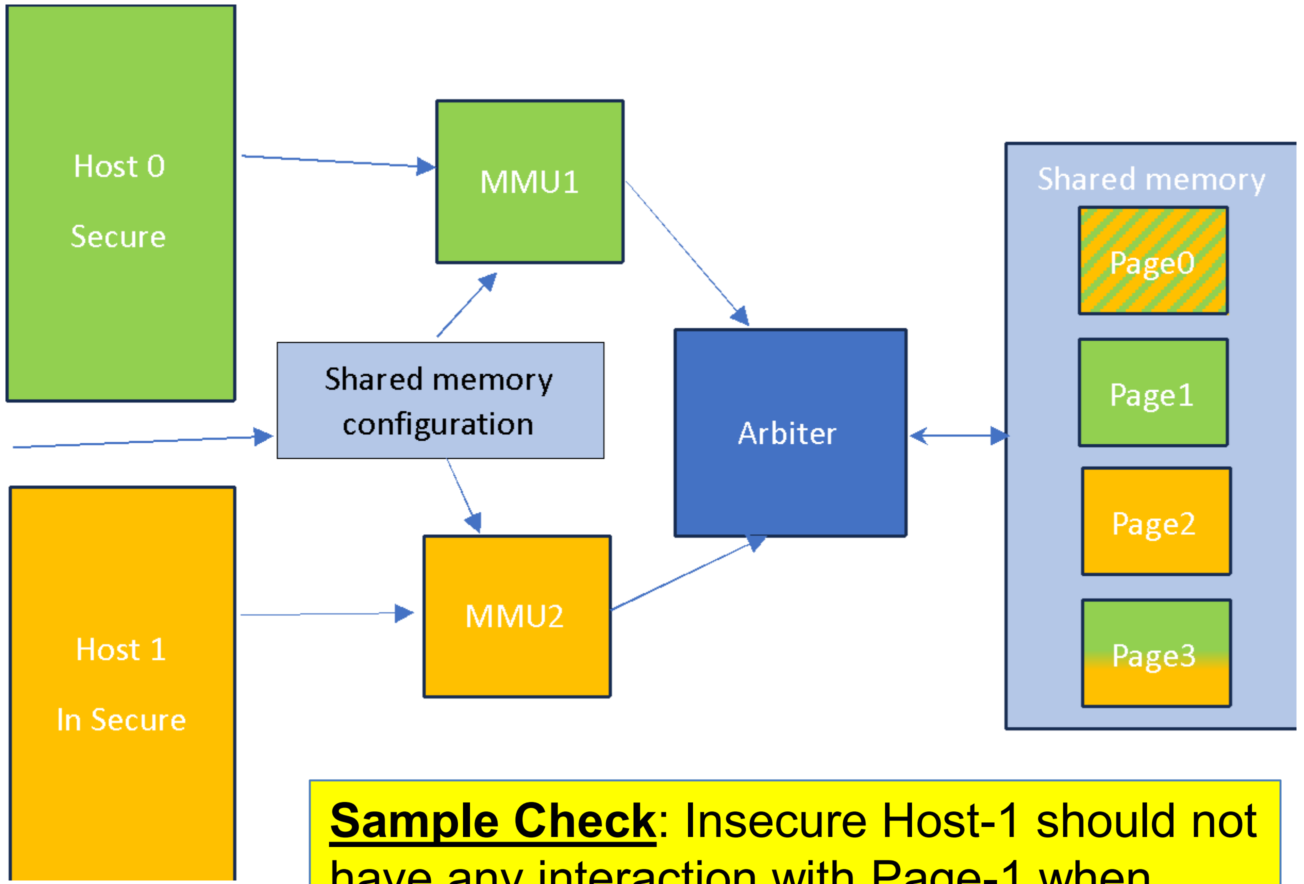


T= SEQ_START
T= SEQ_END
Branch Prediction
Transient instruction
Micro-arch changes
Prediction Failed
Architectural changes saved
Check #1
Secret loaded in Register/Cache
Secret STILL loaded in Register/Cache
Check #2
Illegal transaction: Data/Addr leaked through covert channel

Modern microprocessors are core to the applications that power cars, data centers, and mobile phones.
As their performance has grown, so have their vulnerabilities.

Privilege Access in Multi CPU System

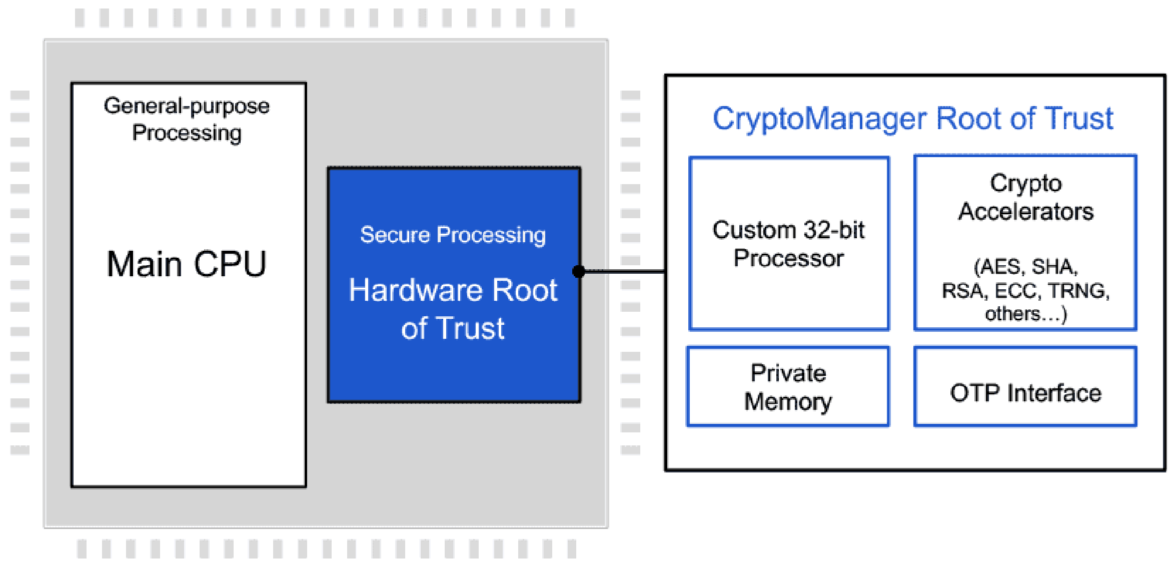
- Host0 and Host1 share the memory
  - Host0: Secure
  - Host1: Insecure.
- Shared Memory: 4 pages
- MMU's configures page is read or write to/from a host.
- Example configuration in the MMU's:
  - Page 0 is shared RW and accessible by both
  - Page 1 is exclusive RW to Host0
  - Page 2 is exclusive 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.

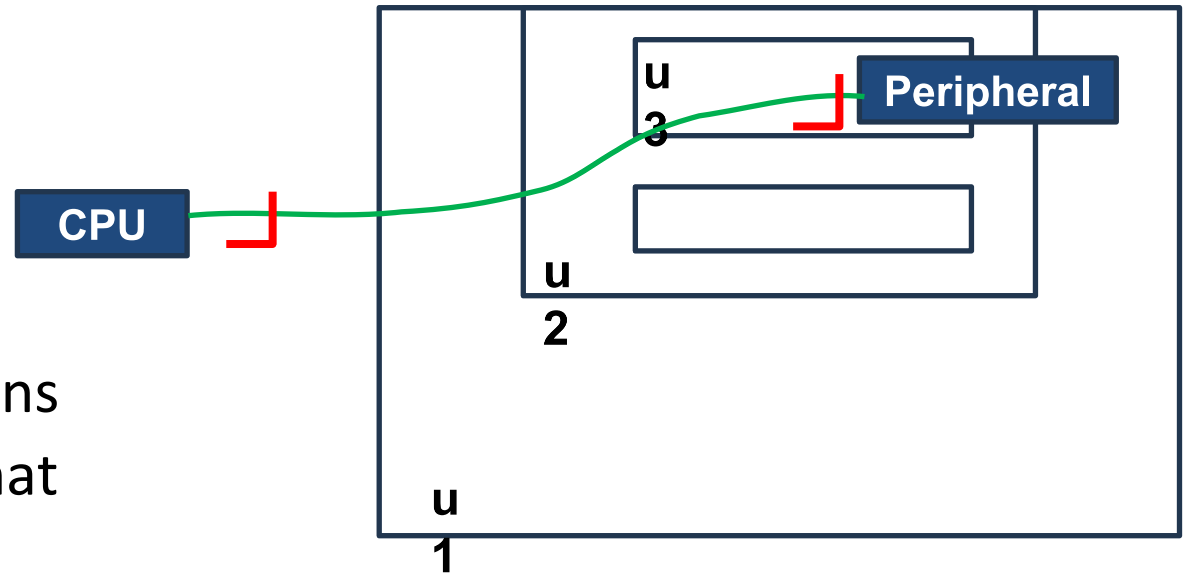
CPU- Hardware Root Of Trust

- 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

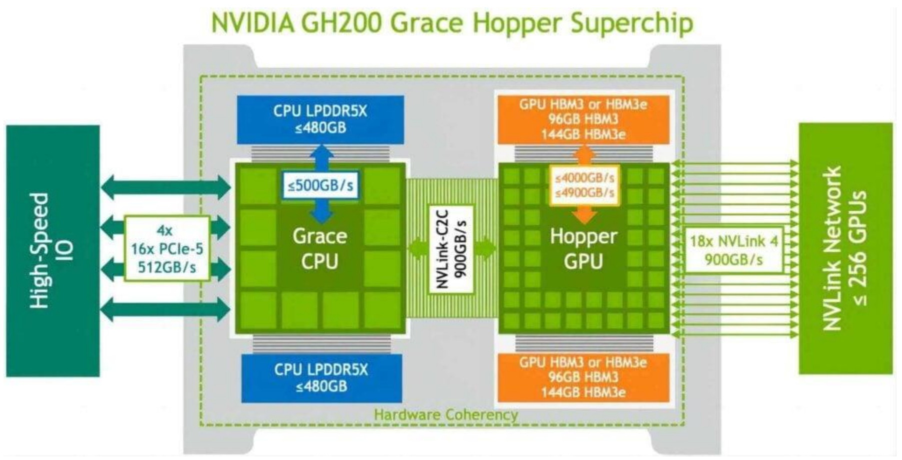


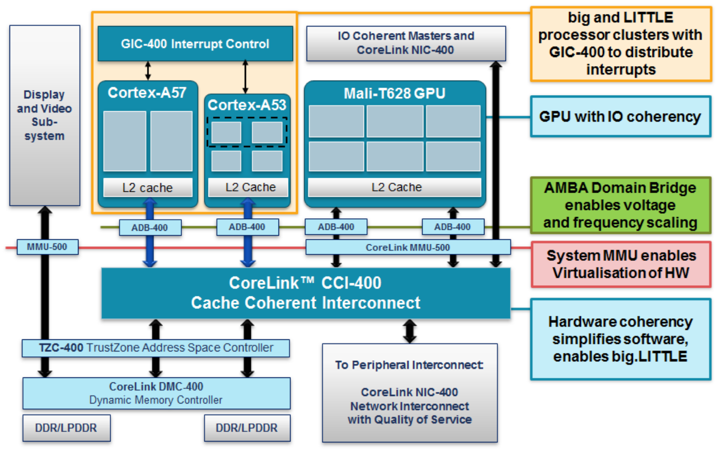
Interrupt Verification

- Specific connectivity register based
- Can propagate through special cells like synchronizers
- Polarity of connection important
- No additional drivers or receivers
- Mask the interrupt under certain conditions
- Read and clear interrupt status register that generated the interrupt signal
  - WI/WO acknowledgment depending on interrupt type
  - Unmask the interrupt under certain conditions

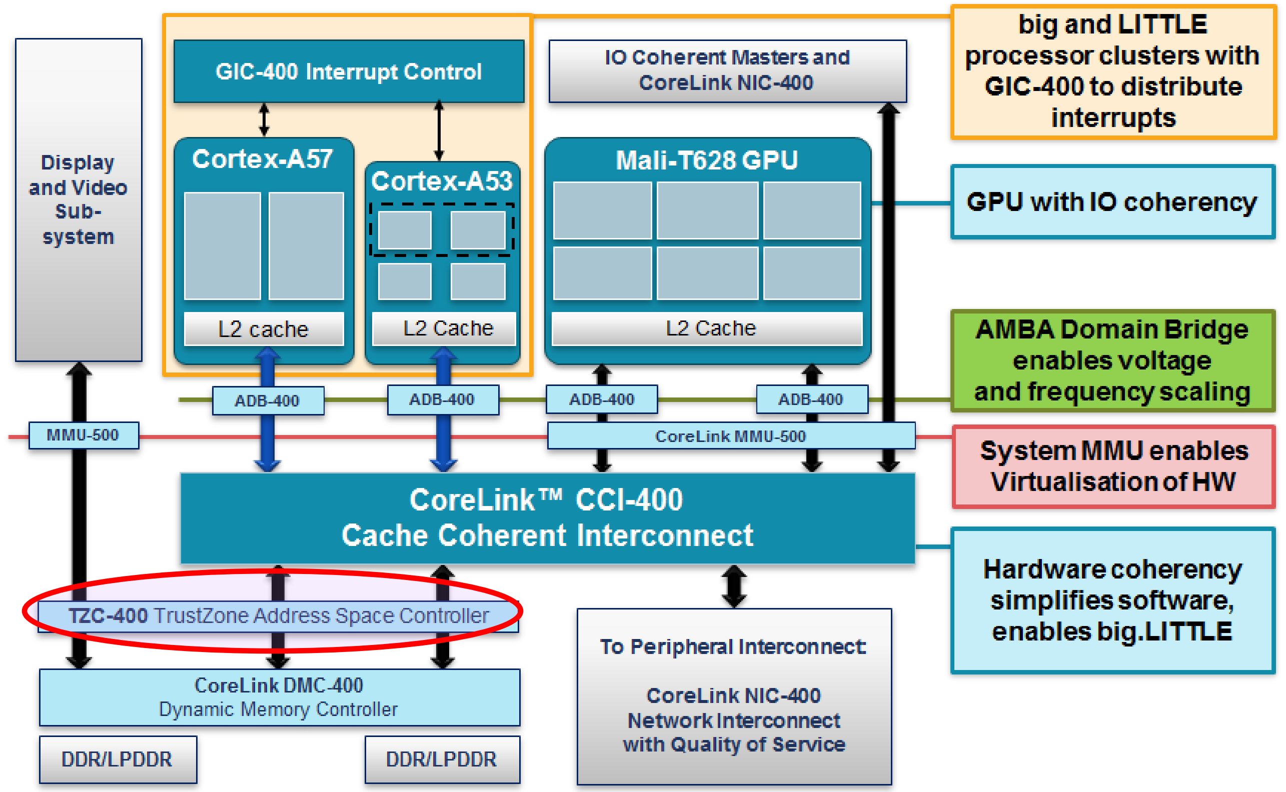


Security IPs Verification Challenges




Display and Video Sub-system
GIC-400 Interrupt Control
Cortex-A57
L2 cache
Cortex-A53
L2 Cache
IO Coherent Masters and CoreLink NIC-400
Mali-T628 GPU
L2 Cache
big and LITTLE processor clusters with GIC-400 to distribute interrupts
GPU with IO coherency
ADB-400
ADB-400
ADB-400
ADB-400
AMBA Domain Bridge enables voltage and frequency scaling
MMU-500
CoreLink MMU-500
System MMU enables Virtualisation of HW
CoreLink™ CCI-400 Cache Coherent Interconnect
Hardware coherency simplifies software, enables big.LITTLE
TZC-400 TrustZone Address Space Controller
CoreLink DMC-400 Dynamic Memory Controller
DDR/LPDDR
DDR/LPDDR
To Peripheral Interconnect:
CoreLink NIC-400 Network Interconnect with Quality of Service

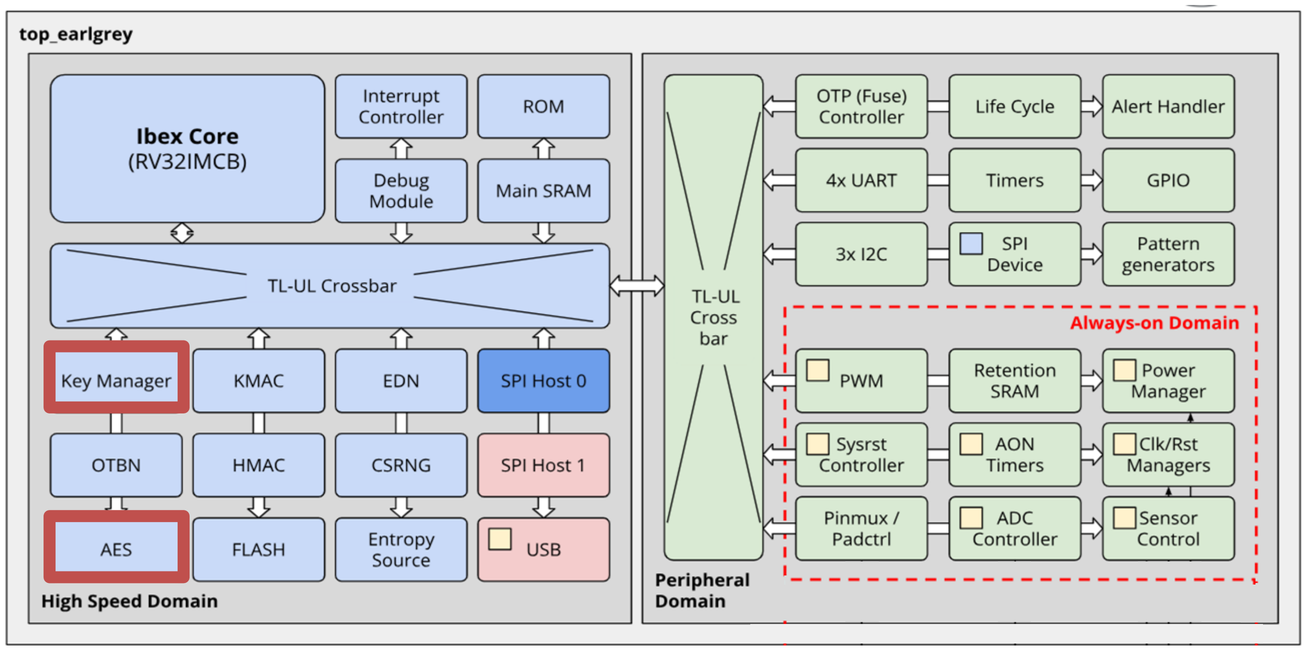
Security IPs – Verification Challenges



Display and Video Sub-system
GIC-400 Interrupt Control
IO Coherent Masters and CoreLink NIC-400
big and LITTLE processor clusters with GIC-400 to distribute interrupts
Cortex-A57
Cortex-A53
Mali-T628 GPU
GPU with IO coherency
L2 cache
L2 Cache
L2 Cache
AMBA Domain Bridge enables voltage and frequency scaling
ADB-400
ADB-400
ADB-400
ADB-400
MMU-500
CoreLink MMU-500
System MMU enables Virtualisation of HW
CoreLink™ CCI-400 Cache Coherent Interconnect
Hardware coherency simplifies software, enables big.LITTLE
TZC-400 TrustZone Address Space Controller
To Peripheral Interconnect
CoreLink NIC-400 Network Interconnect with Quality of Service
CoreLink DMC-400 Dynamic Memory Controller
DDR/LPDDR
DDR/LPDDR

- 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.

Crypto Cores, AES, Key Manager Security Vulnerabilities



- Encryption/Decryption block
- Partition assets: Confidential, Exempt, and Exposed signals
  - Confidential signals hold secure data
  - Exposed signals are world readable
  - Exempt signals hold post-encryption data

Abstract Data

Illegal data transfer:
Confidential □ Exposed

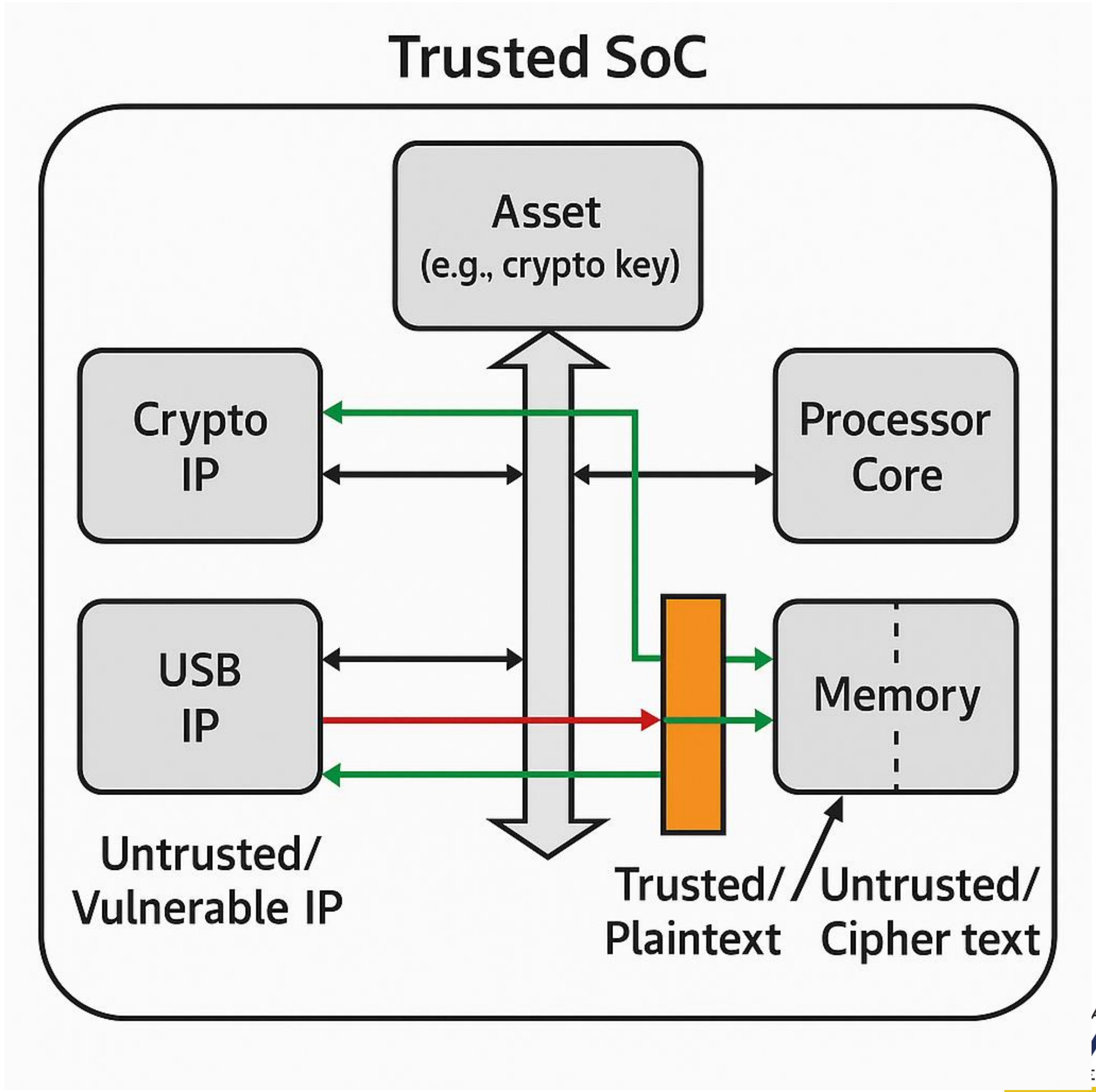
Valid data transfer:
Exempt □ Exposed

Illegal data transfer:
Confidential □ Exempt

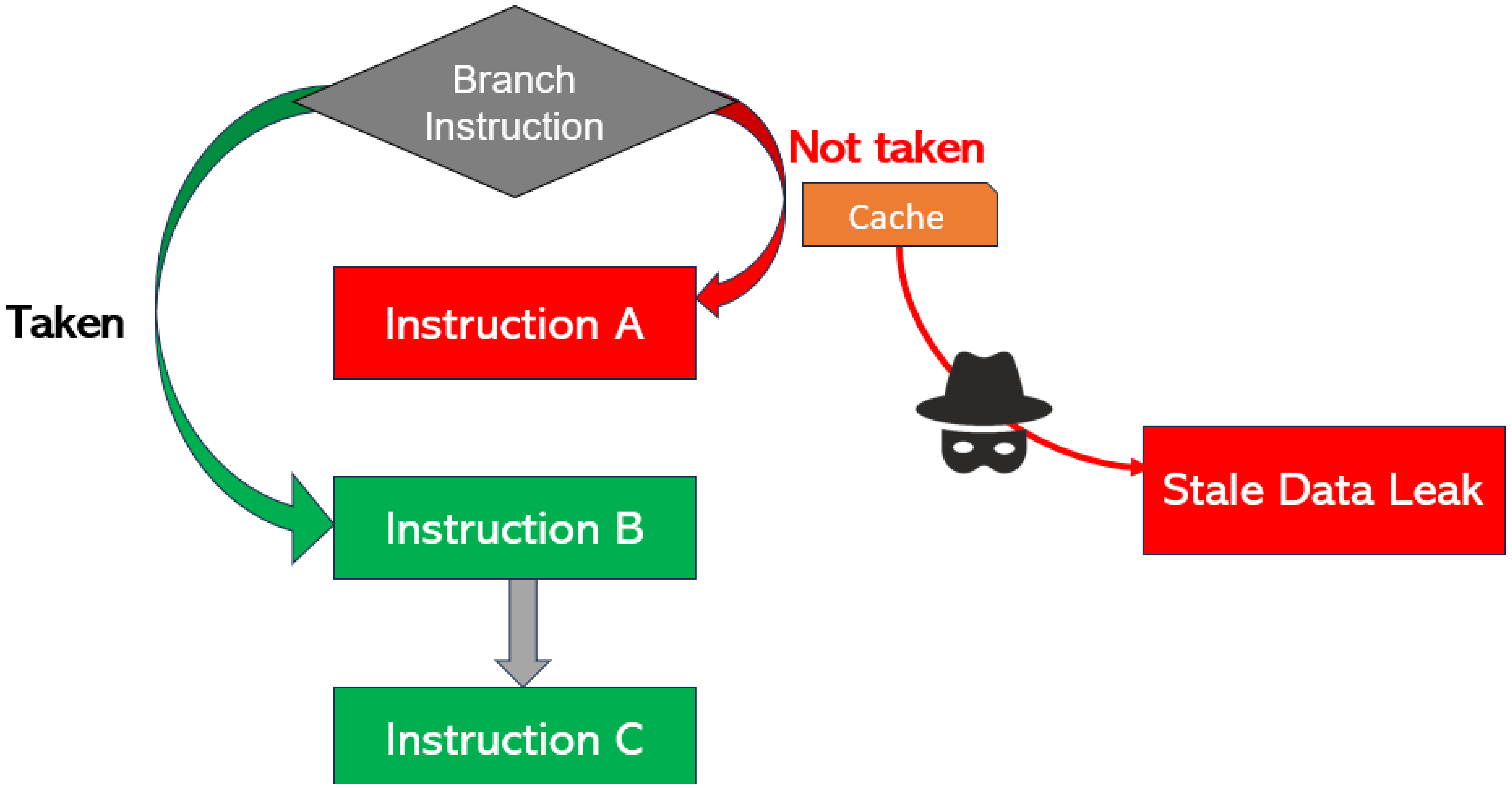
Valid data transfer:
Confidential □ Encrypted □ Exempt

Illegal Access Prevention

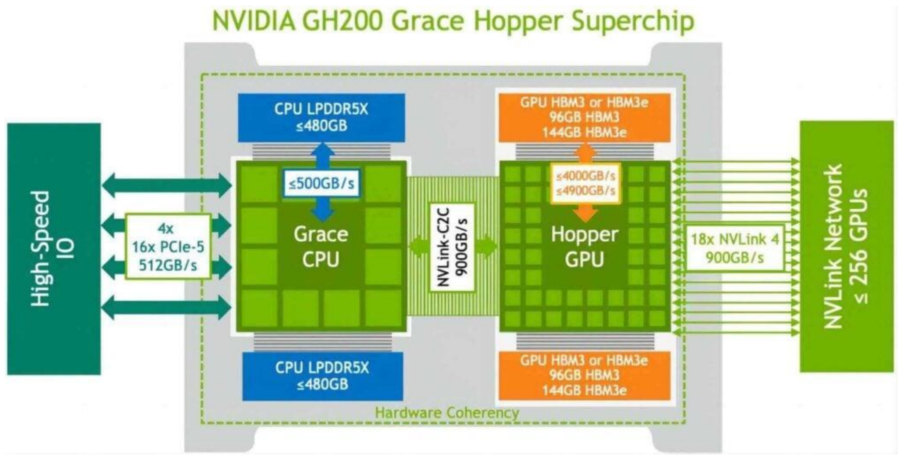
- Only Allowed Host can access
- Access permissions can be static or configurable

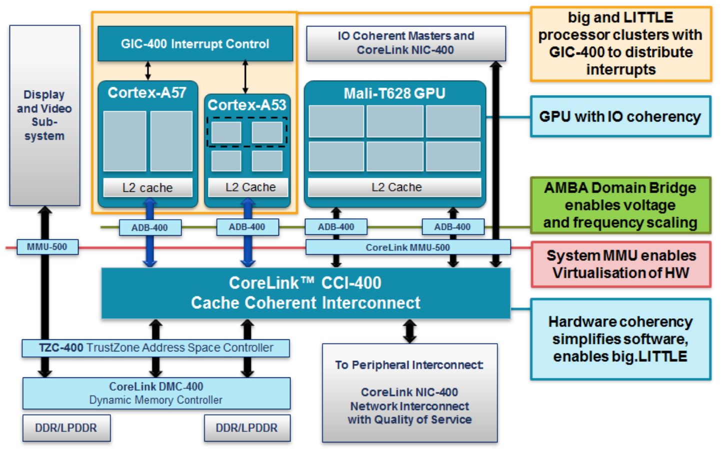


Speculative Load/Store Verification

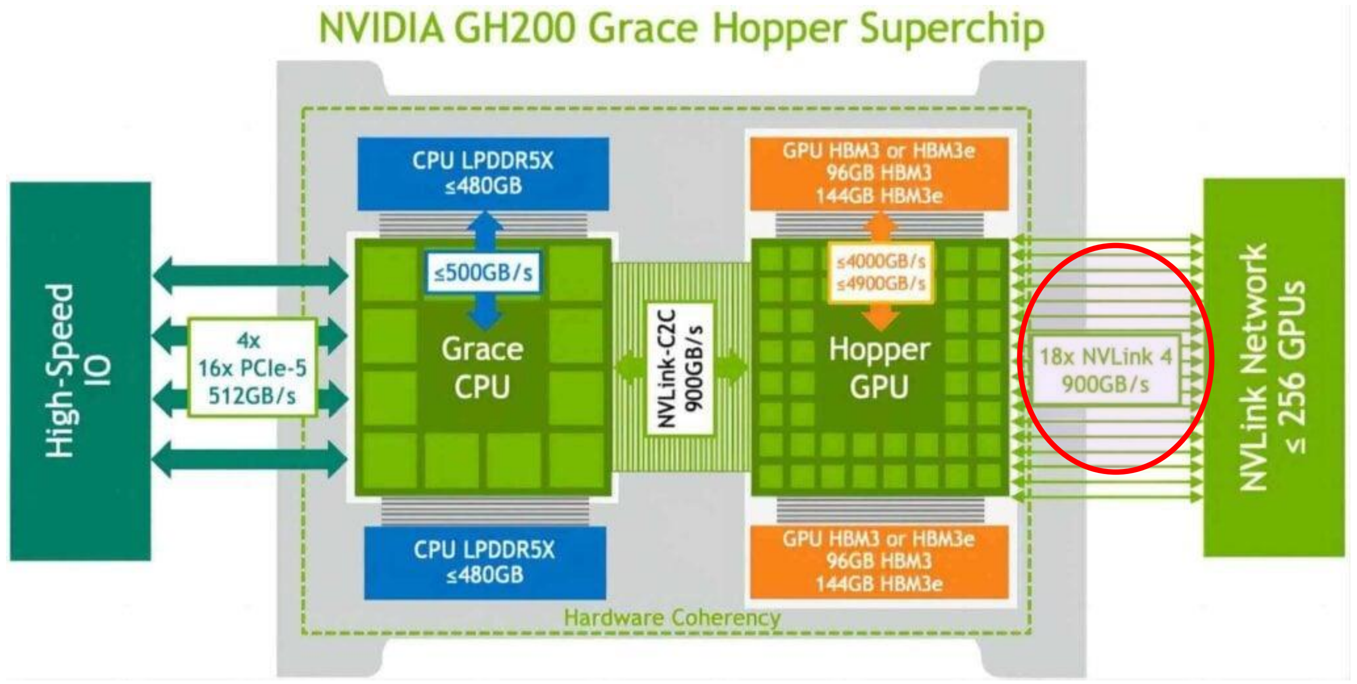


Interconnect Verification Challenges




Display and Video Sub-system
GIC-400 Interrupt Control
Cortex-A57
L2 cache
Cortex-A53
L2 Cache
IO Coherent Masters and CoreLink NIC-400
Mali-T628 GPU
L2 Cache
ADB-400
ADB-400
ADB-400
ADB-400
MMU-500
CoreLink MMU-500
CoreLink™ CCI-400 Cache Coherent Interconnect
TZC-400 TrustZone Address Space Controller
CoreLink DMC-400 Dynamic Memory Controller
DDR/LPDDR
DDR/LPDDR
To Peripheral Interconnect: CoreLink NIC-400 Network Interconnect with Quality of Service
big and LITTLE processor clusters with GIC-400 to distribute interrupts
GPU with IO coherency
AMBA Domain Bridge enables voltage and frequency scaling
System MMU enables Virtualisation of HW
Hardware coherency simplifies software, enables big.LITTLE

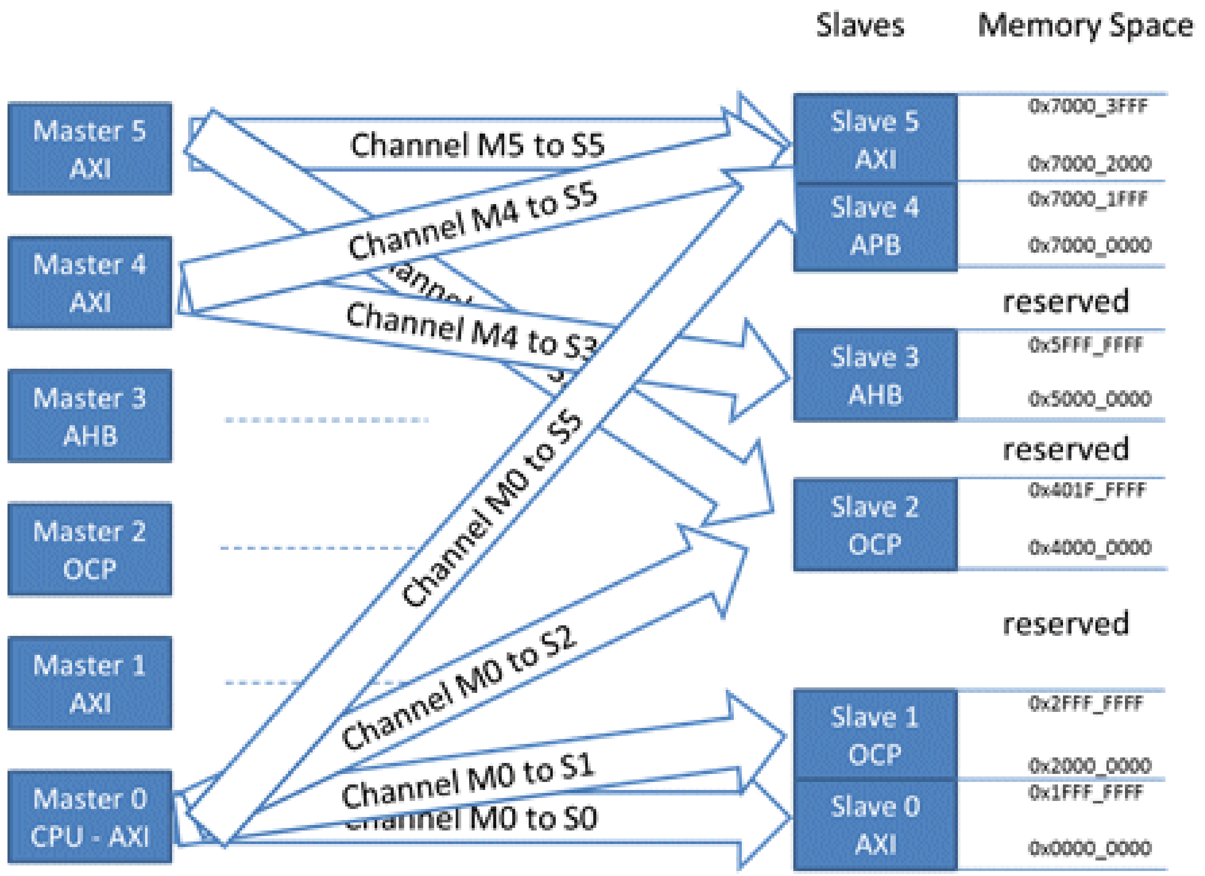
Interconnect Verification



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

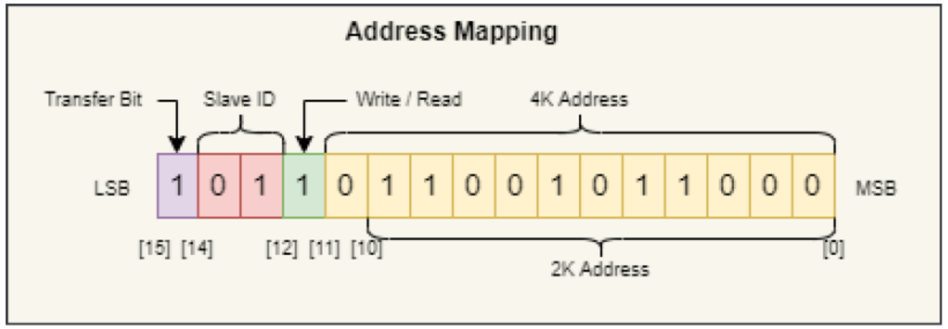
Interconnect Verification

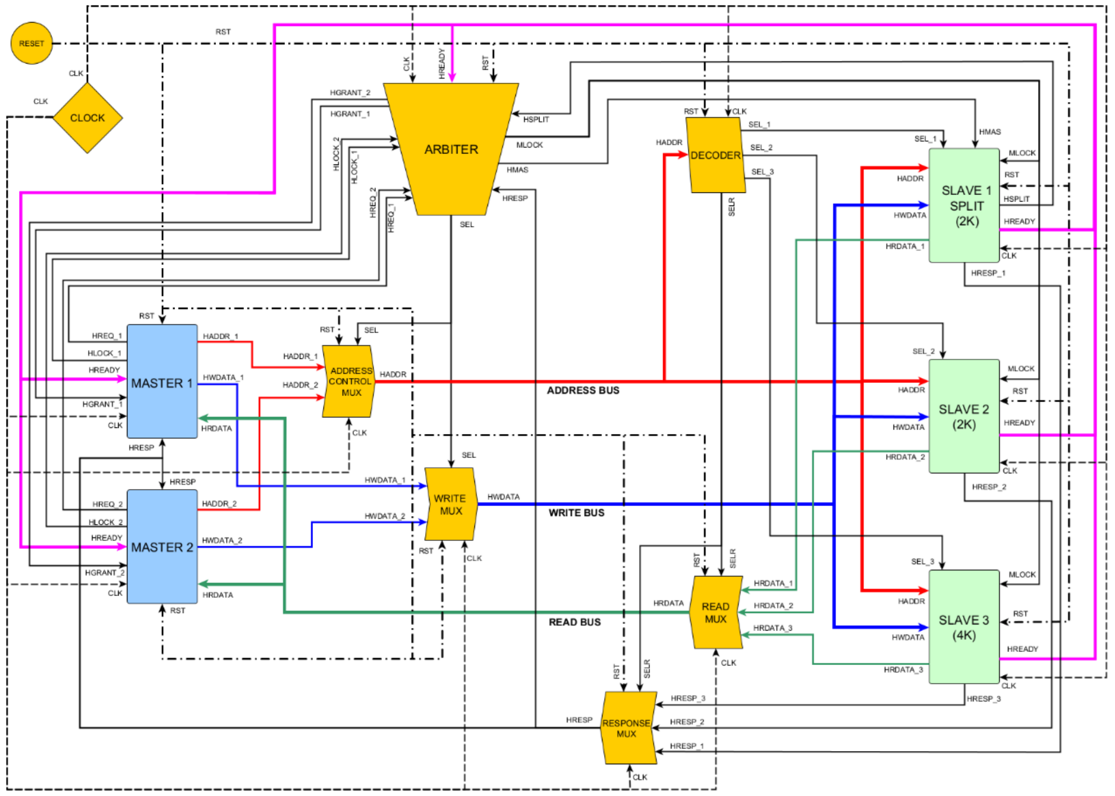
- Reachability Verification
  - A particular master connected to slave
  - Protocol ports connected correctly
  - Connections through complex sequential logic (bridges etc.)
  - Negative verification – masters not connected to slaves as per specification



Address Mapping Verification

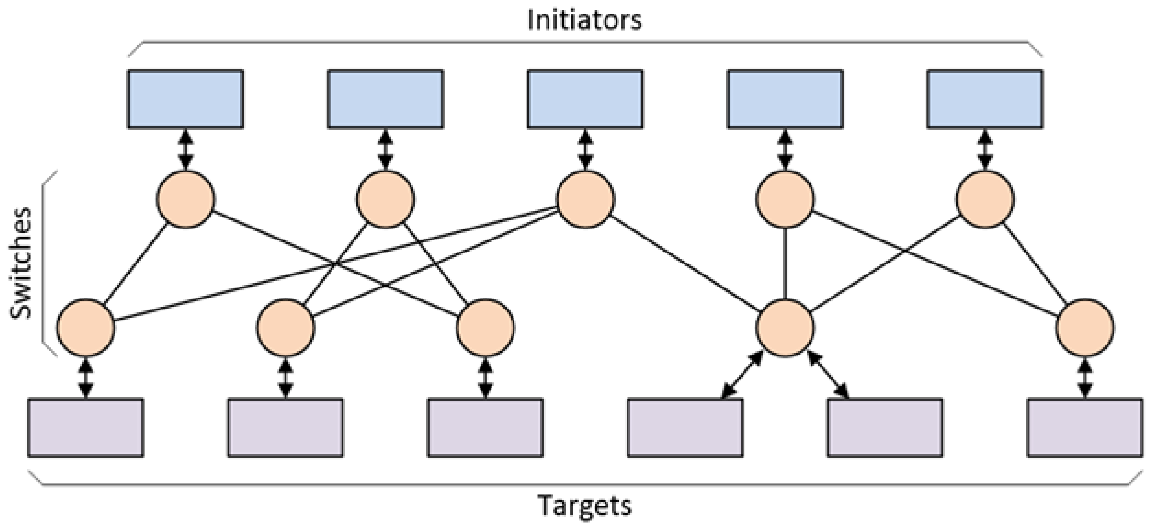
- Simple Data transfer
  - Master1 requests for bus access
  - Check
    - One cycle later Master1 granted bus access
  - Master1 deploys address and control data on address bus
  - Check
    - Sel pin toggles high for slave, HWDATA gets master DATA one cycle later
    - One cycle later HWRITE goes high and HRESP = 00 (meaning Ok transfer)

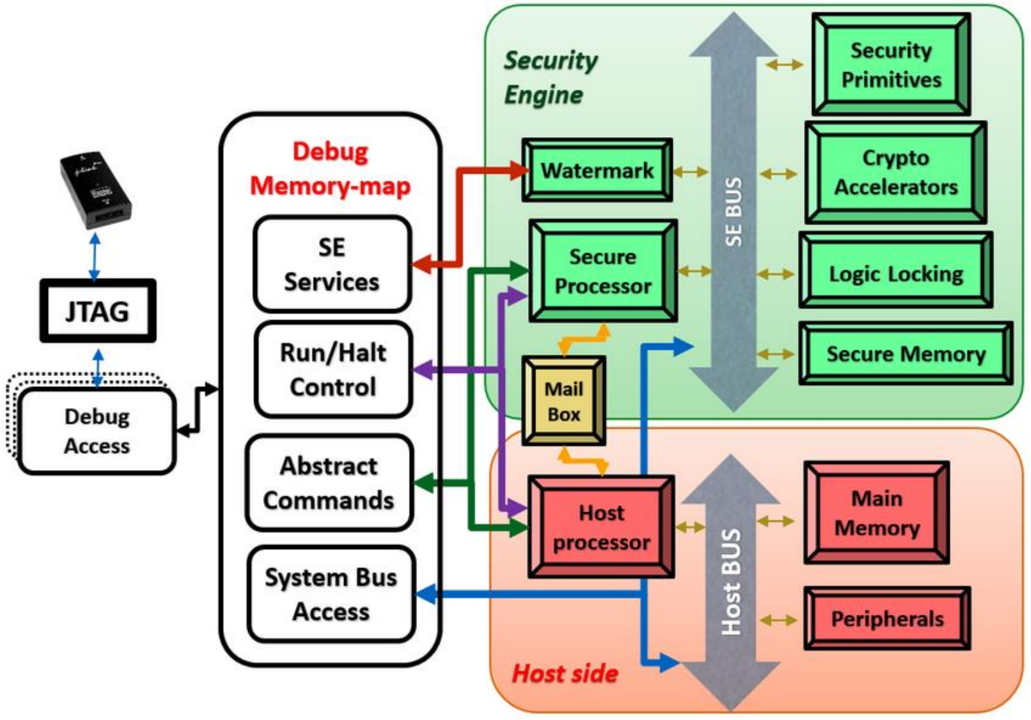




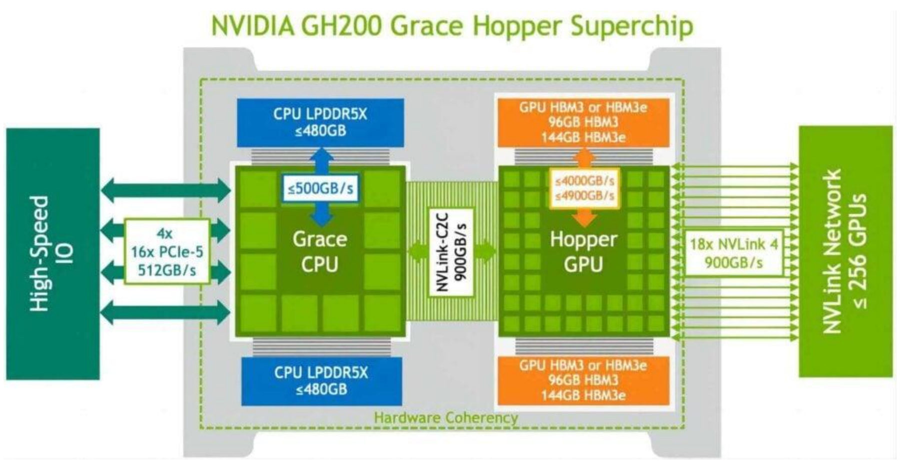
Secure Access Verification

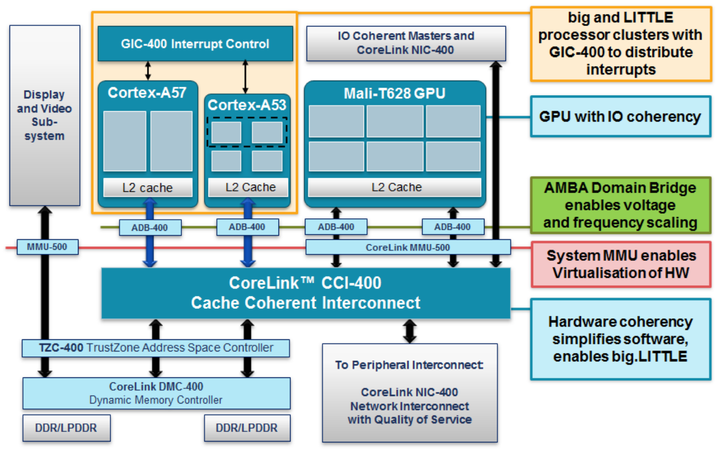
- Secure Interconnect initiator, target access
- Debug interface protection



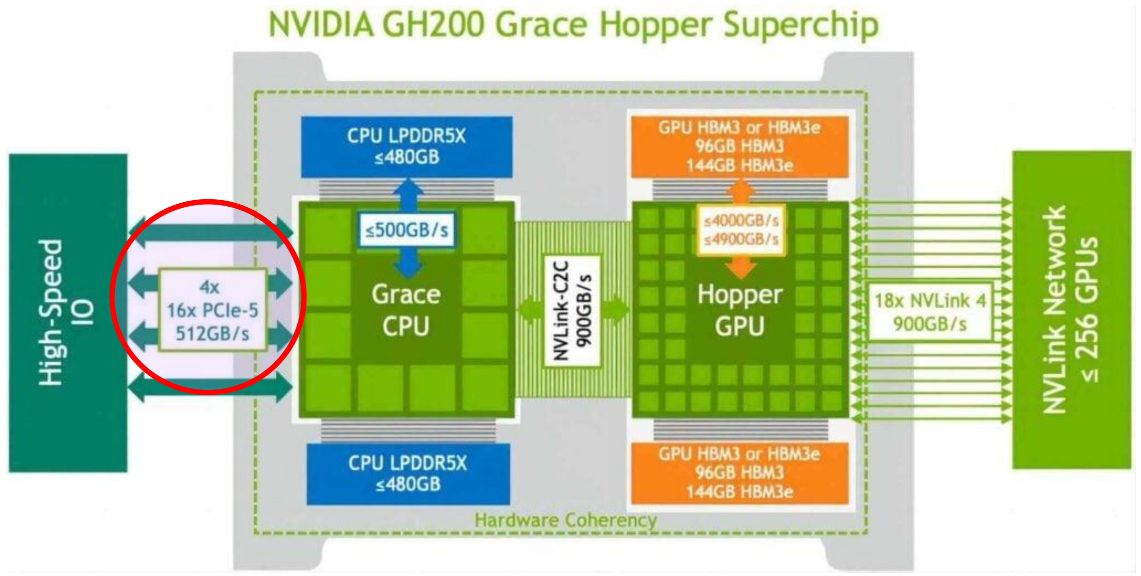


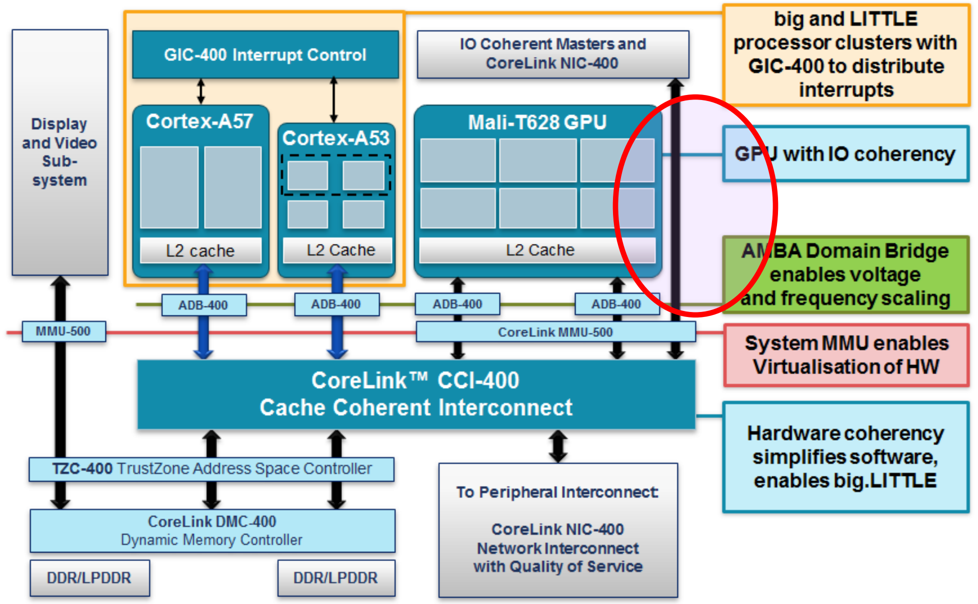
IO Verification Challenges




Display and Video Sub-system
GIC-400 Interrupt Control
Cortex-A57
L2 cache
Cortex-A53
L2 Cache
IO Coherent Masters and CoreLink NIC-400
Mali-T628 GPU
L2 Cache
big and LITTLE processor clusters with GIC-400 to distribute interrupts
GPU with IO coherency
ADB-400
ADB-400
ADB-400
ADB-400
AMBA Domain Bridge enables voltage and frequency scaling
MMU-500
CoreLink MMU-500
System MMU enables Virtualisation of HW
CoreLink™ CCI-400 Cache Coherent Interconnect
Hardware coherency simplifies software, enables big.LITTLE
TZC-400 TrustZone Address Space Controller
CoreLink DMC-400 Dynamic Memory Controller
DDR/LPDDR
DDR/LPDDR
To Peripheral Interconnect: CoreLink NIC-400 Network Interconnect with Quality of Service

IO Verification

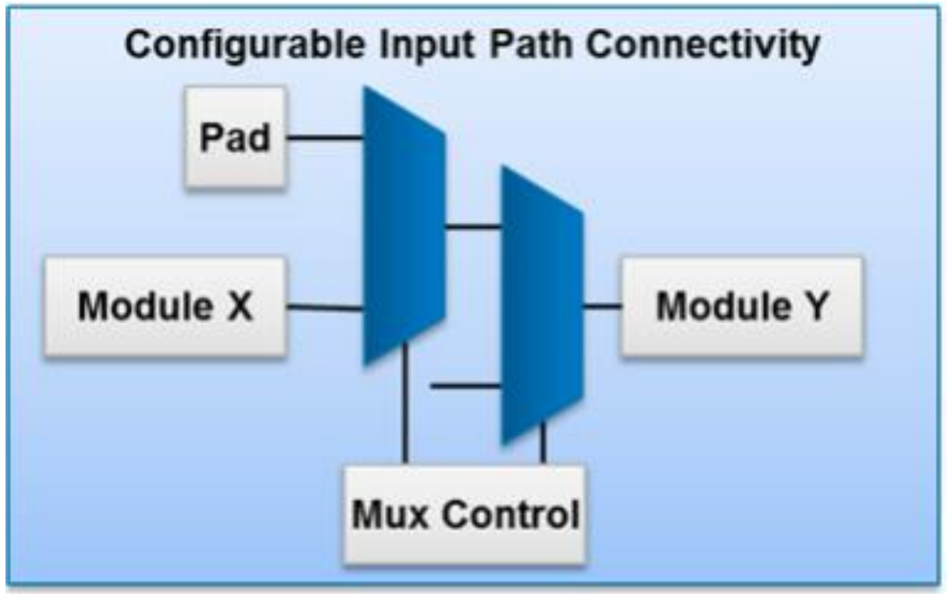


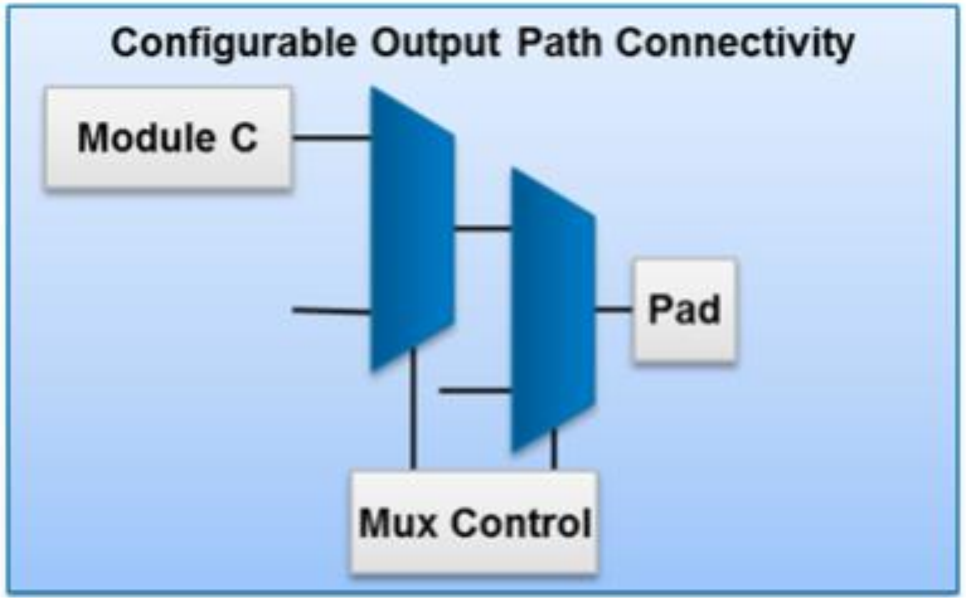


GIC-400 Interrupt Control
IO Coherent Masters and CoreLink NIC-400
big and LITTLE processor clusters with GIC-400 to distribute interrupts
Display and Video Sub-system
Cortex-A57
Cortex-A53
Mali-T628 GPU
GPU with IO coherency
L2 cache
L2 Cache
L2 Cache
AMBA Domain Bridge enables voltage and frequency scaling
ADB-400
ADB-400
ADB-400
ADB-400
MMU-500
CoreLink MMU-500
System MMU enables Virtualisation of HW
CoreLink™ CCI-400 Cache Coherent Interconnect
Hardware coherency simplifies software, enables big.LITTLE
TZC-400 TrustZone Address Space Controller
CoreLink DMC-400 Dynamic Memory Controller
DDR/LPDDR
DDR/LPDDR
To Peripheral Interconnect
CoreLink NIC-400 Network Interconnect with Quality of Service

IO Verification

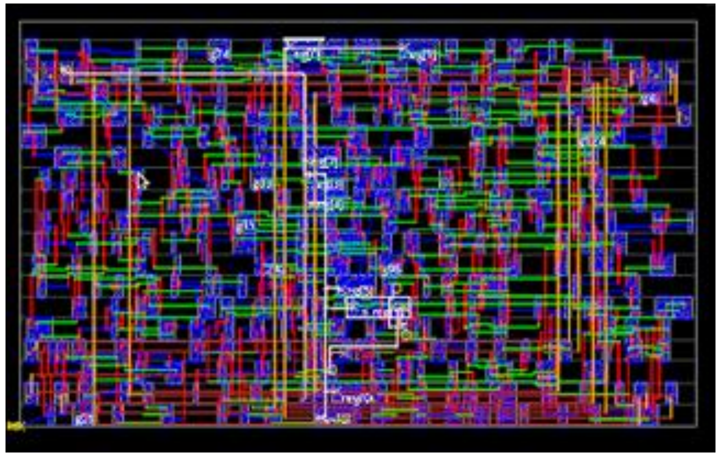
- Source to Destination connection
  - Based upon conditions

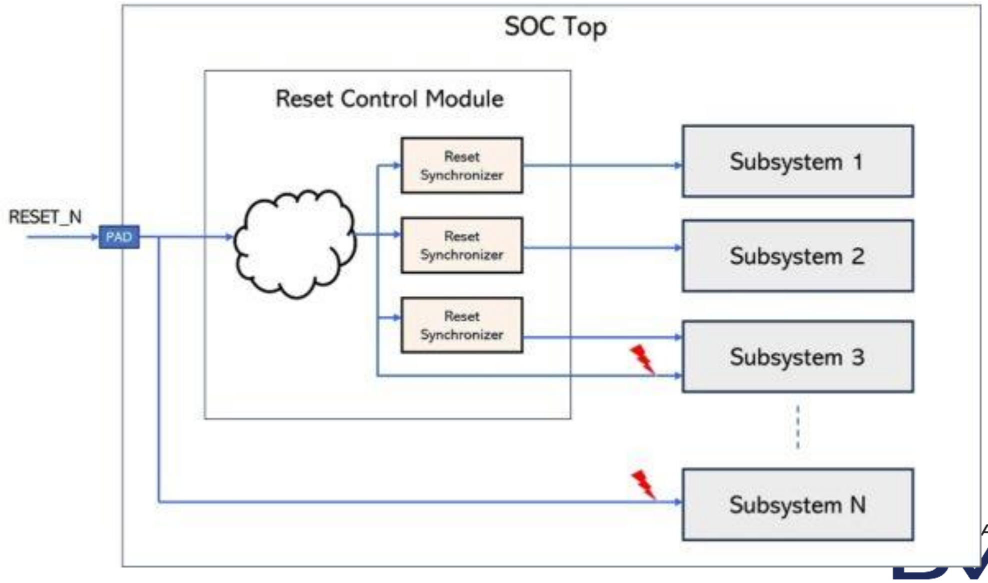


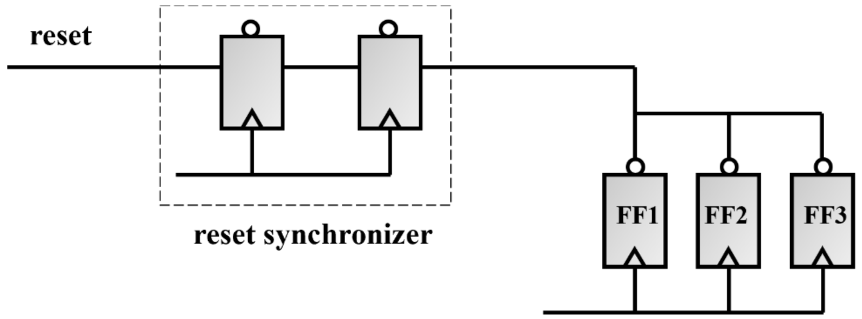


IO Global Signals

- 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 muxes

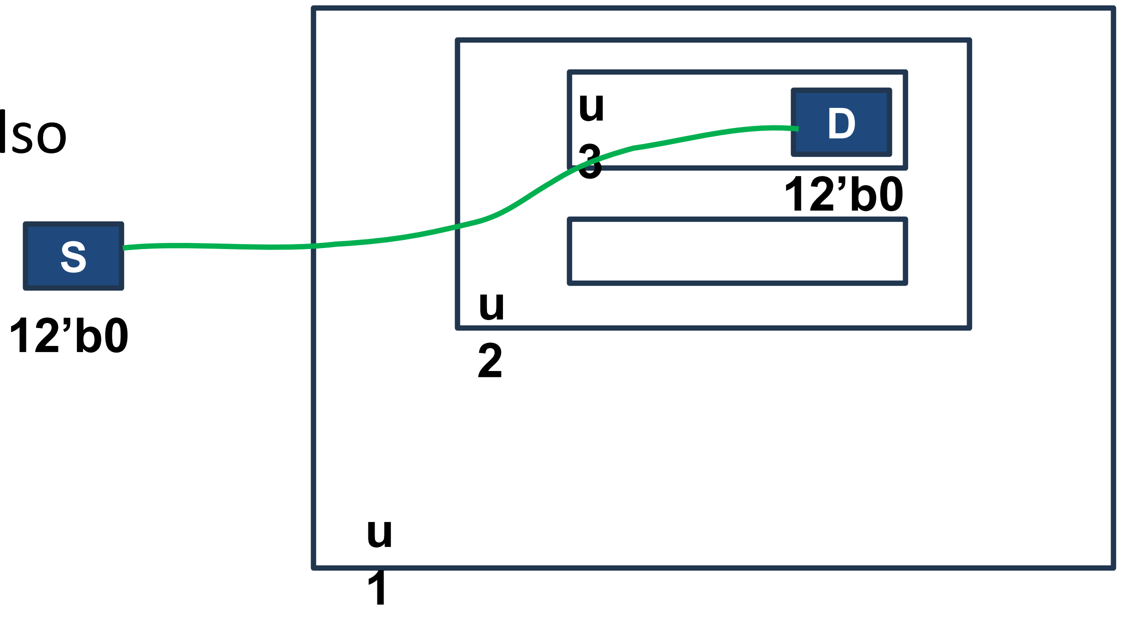






IO Registers Verification

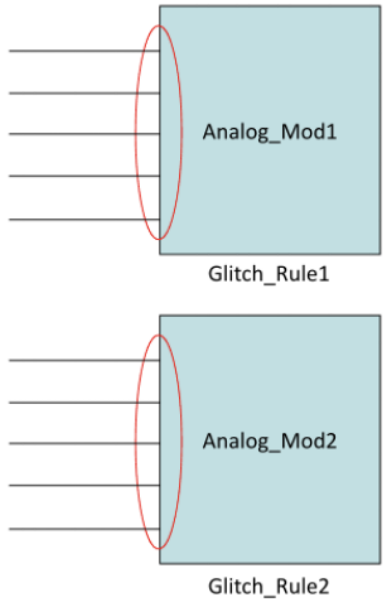
- System Registers, Configuration registers
  - Specific connectivity register based
  - Can propagate through sequential
  - Not just connection, value propagation also important

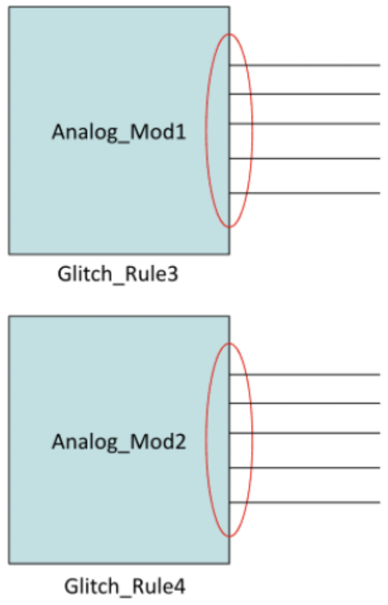


IO A/D Interface Glitch Verification

create_group –name AM1_IN –module Analog_Mod1 –all_inputs
set_no_glitch -to_group {AM1_IN} –rule Glitch_Rule1

create_group –name AM2_IN –module Analog_Mod2 –all_inputs
set_no_glitch -to_group {AM2_IN} –rule Glitch_Rule2





create_group –name AM1_OUT –module Analog_Mod1 –all outputs
set_no_glitch -from_both_group {AM1_OUT} –rule Glitch_Rule3

create_group –name AM2_OUT –module Analog_Mod2 –all_outputs
set_no_glitch -from_both_group {AM2_OUT} –rule Glitch_Rule4

Addressing These Challenges with Static Signoff

Liberated from Boolean shackles

Static Signoff Paradigm is Expanding

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

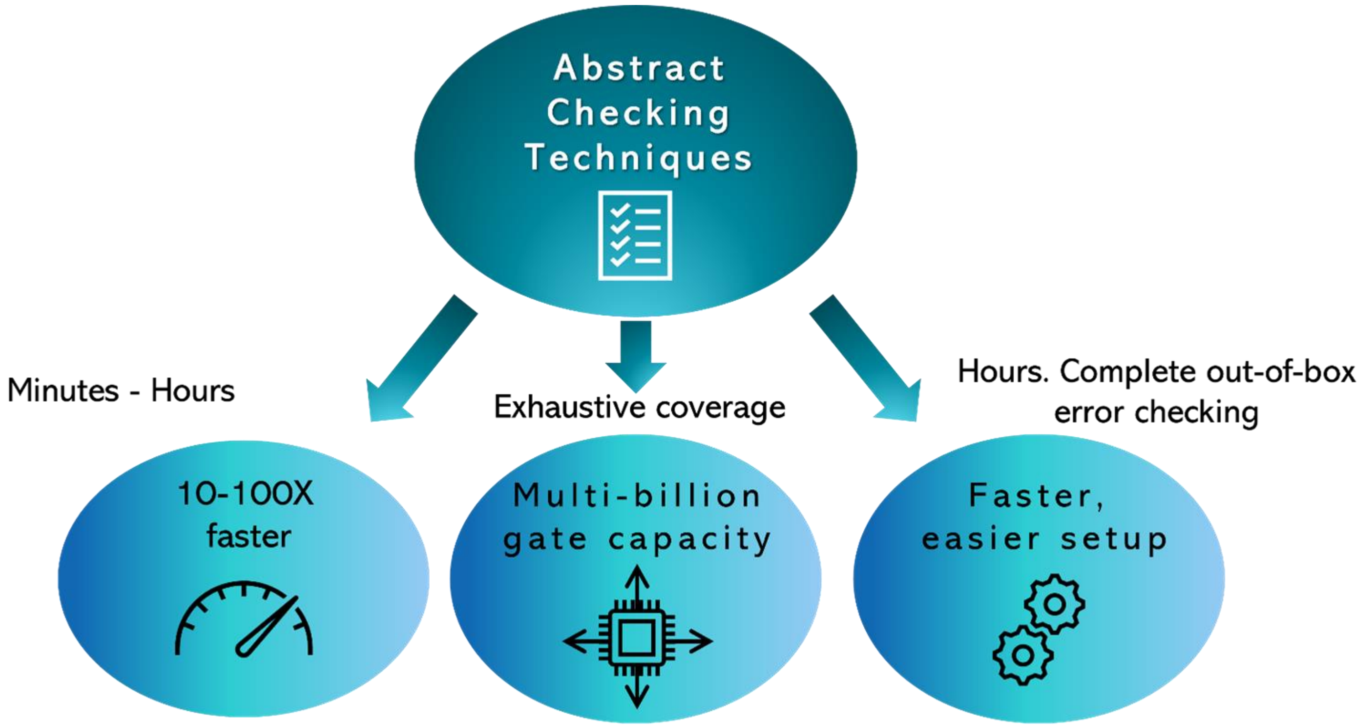
Simulatio
n

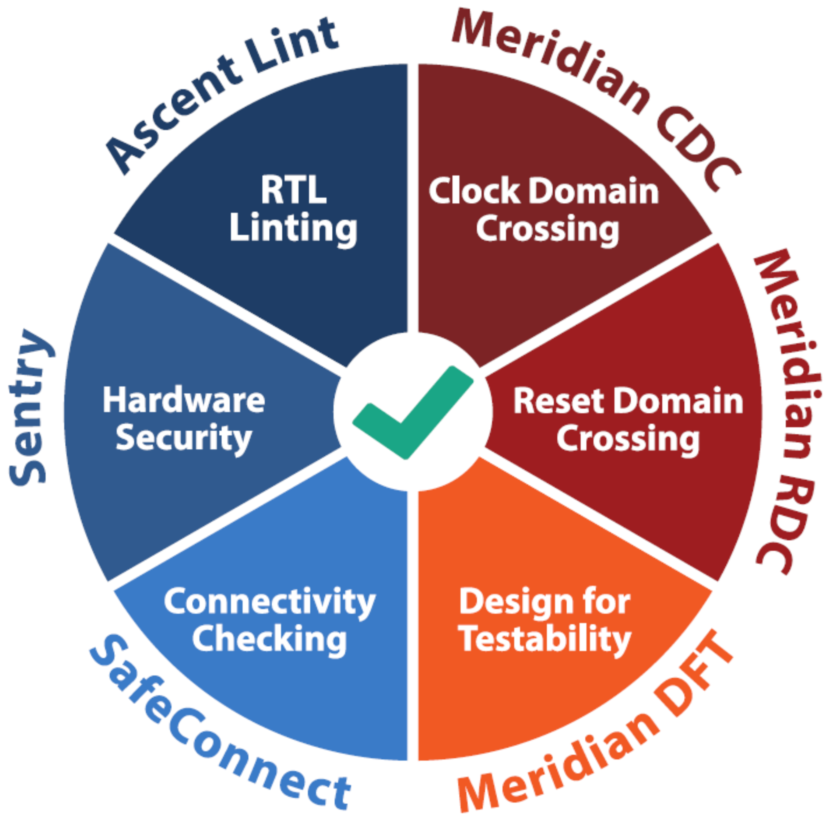
Formal
Verificati
on

Simulation and Formal have their role to play

Use Static Signoff effectively and judiciously to make verification faster, more predictable, and tape-out chips without risk

Signoff with Static


Abstract Checking Techniques
Minutes - Hours
Exhaustive coverage
Hours. Complete out-of-box error checking
10-100X faster
Multi-billion gate capacity
Faster, easier setup



Where Our Technology Is Adding Value

SPACEX

WD Western Digital®

Google

ICsense THE IC DESIGN COMPANY

paloalto NETWORKS®

SAMSUNG

maxim integrated™

JARIET Technologies

nVIDIA®

dialog SEMICONDUCTOR

NEC

MARVELL™

ORACLE

MicroVision

TARANA

gyrfalcon technology

FORTINET®

Microsoft

knowles

PETAiO

HAILO Empowering Intelligence

CERAGON

Viasat™

NAVSEA NAVAL SEA SYSTEMS COMMAND

U.S.ARMY

Materials Mag!c Hitachi Metals

SatixFy

D E Shaw Research

Esperanto TECHNOLOGIES

HARRIS®

LOCKHEED MARTIN

AVID®

Valens

accellera SYSTEMS INITIATIVE™

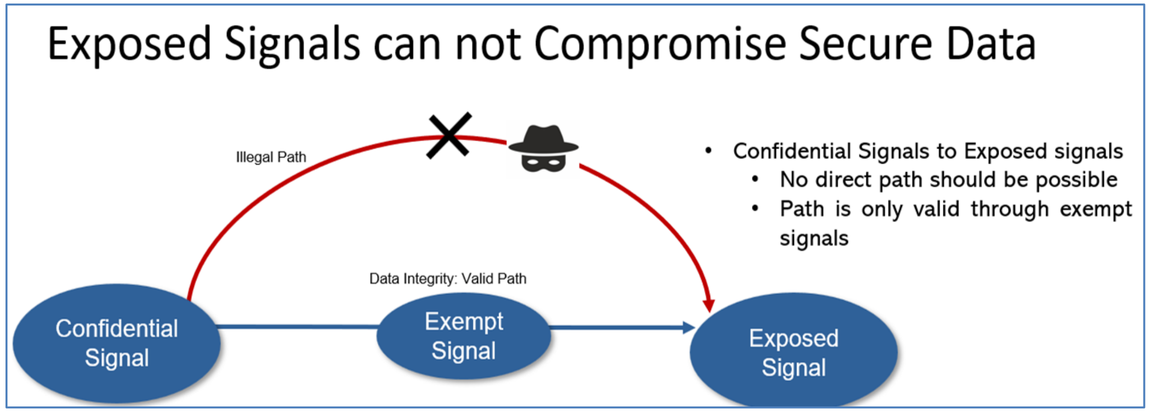
And the list keeps on growing….

2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

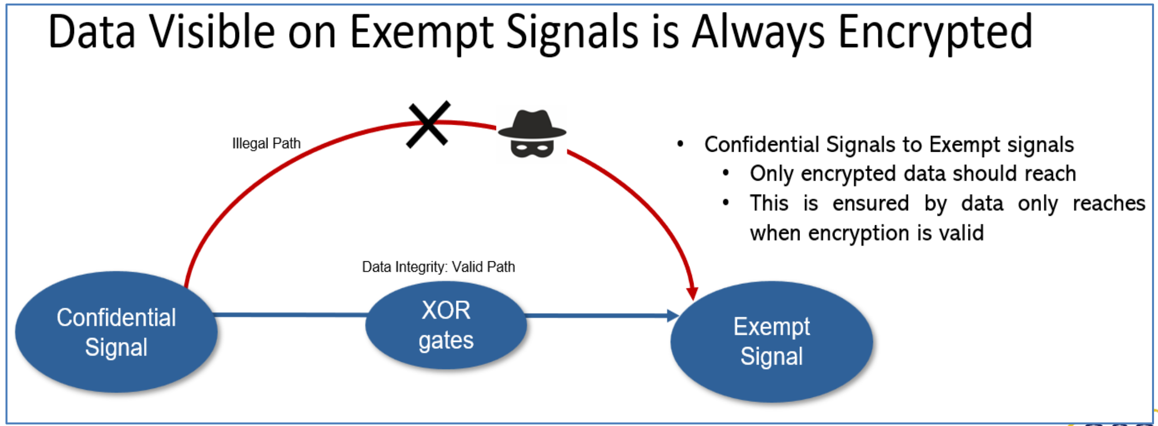
QnA

Confidential Signal *Exposed* in Crypto IPs

- **We can prove functionality…**
  - Core like AES passes all FIPS/NIST test vectors
  - All modes verified: ECB, CBC, CTR, GCM
  - DV environment confirms liveness functionality
- **…but what if Confidential signal available at Exposed signal?**
  - Micro-architectural behavior not in spec
  - Simulation Limitation: Negative scenario can't be made unless known vectors for it
  - Formal Limitation: Not able to conclusively prove Safety properties
- **How do you verify the illegal path doesn't expose confidential signals?**


Exposed Signals can not Compromise Secure Data
Illegal Path
Data Integrity: Valid Path
Confidential Signal
Exempt Signal
Exposed Signal

- Confidential Signals to Exposed signals
  - No direct path should be possible
  - Path is only valid through exempt signals


Data Visible on Exempt Signals is Always Encrypted
Illegal Path
Data Integrity: Valid Path
Confidential Signal
XOR gates
Exempt Signal

- Confidential Signals to Exempt signals
  - Only encrypted data should reach
  - This is ensured by data only reaches when encryption is valid

2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

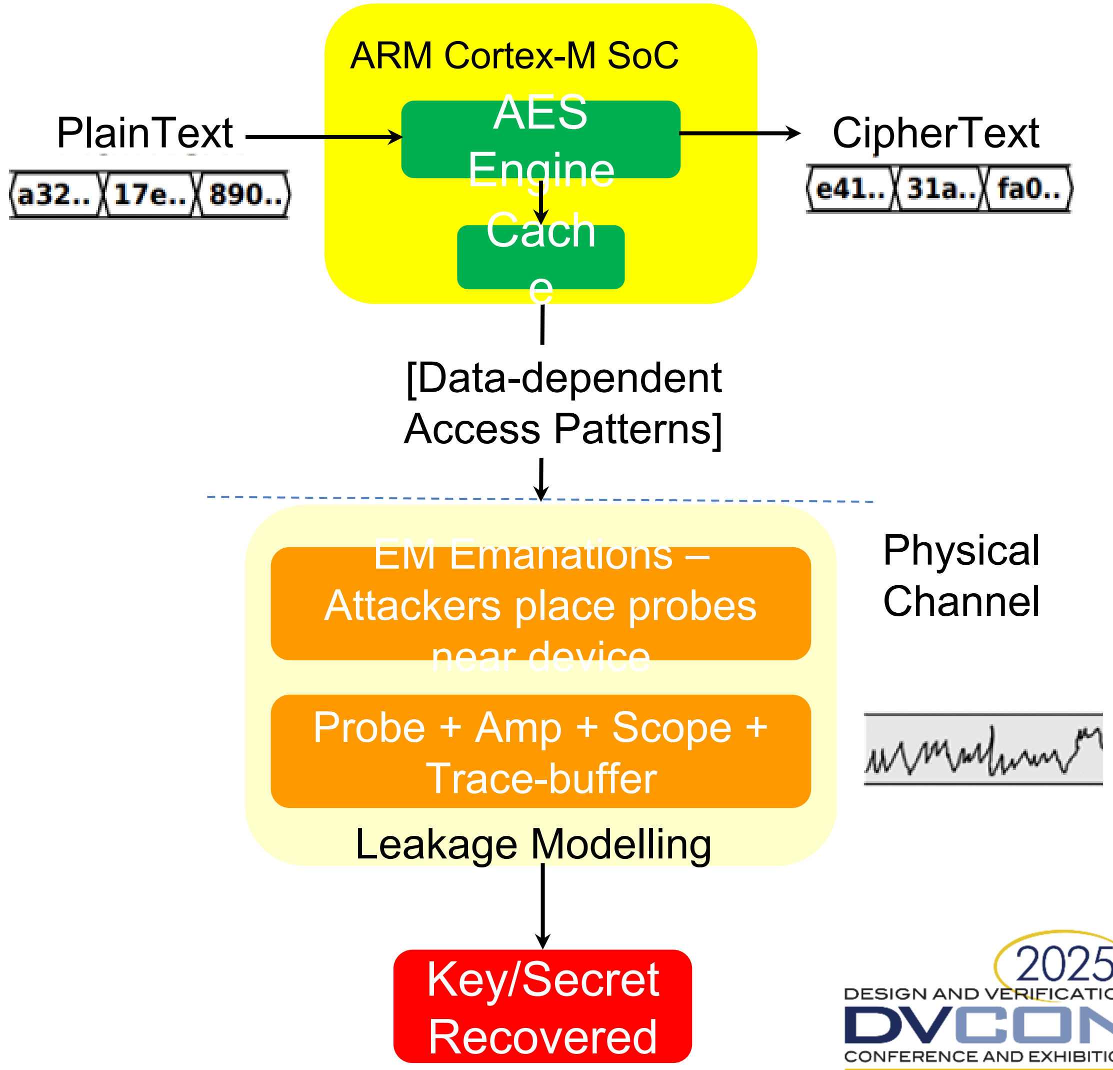
accellera SYSTEMS INITIATIVE™

Cache Physical Attacks — *Overlooked* Threat

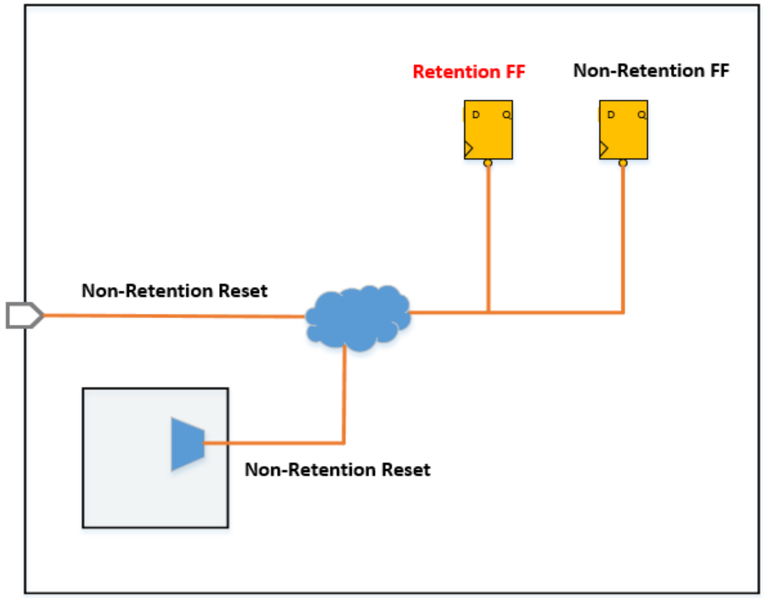
- **Impact**
  - Exploits physical leakage, bypasses logical protections
  - Demonstrates vulnerability in cache-based crypto implementations
- **Verification Challenge for You**
  - **How would you verify the absence of such cache leakage in your design?**

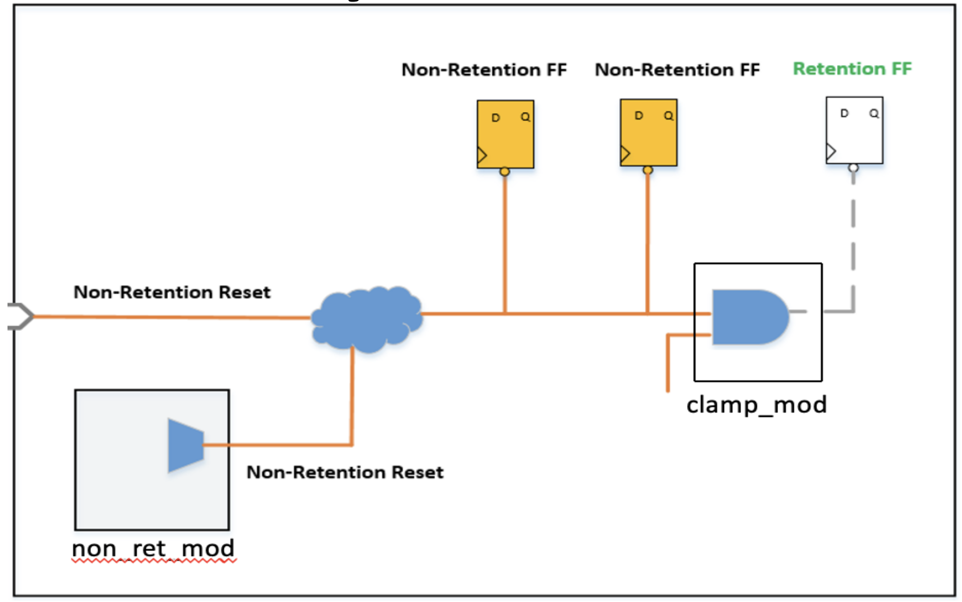
  → Simulation? Lab EM probes? Formal? Are they complete ??

  → What defines “acceptable” leakage in your flow?



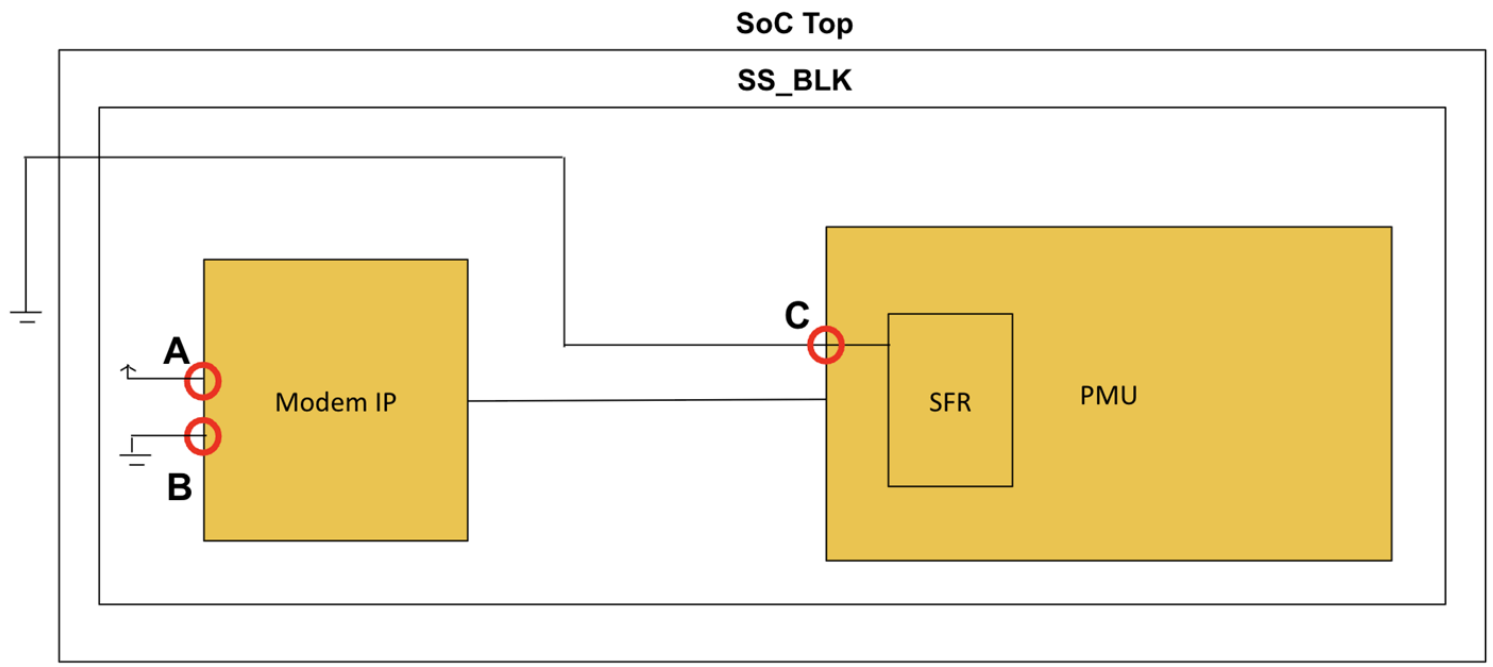
Power logic connectivity verification





- Retention FFs must be reset only by retention resets, and non-retention FFs only by non-retention resets.
- Driving a retention FF with a non-retention reset (without clamp) overwrites its saved state during power-down.
- This ensures true retention behaviour and avoids unintended resets.
- **How do you generate and validate the list of retention FFs in your design flow?**

Toggle coverage



- **Toggle coverage** checks whether a signal actually transitions during simulation or remains tied to a constant.
- **Limitation of simulation:** It is tedious to rely on, and issues like stuck or misconnected ports can still be missed.
- **How do you perform toggle coverage in your flow?**