2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION

UNITED STATES

SAN JOSE, CA, USA

FEBRUARY 24-27, 2025

Hierarchical Formal Verification and Progress Checking of Network-On-Chip Design

Pritam Roy, Ping Yeung, Joon Hong, Abhishek Desai, Aishwarya Raj,

Chirag Agarwal, Dhruvin Patel

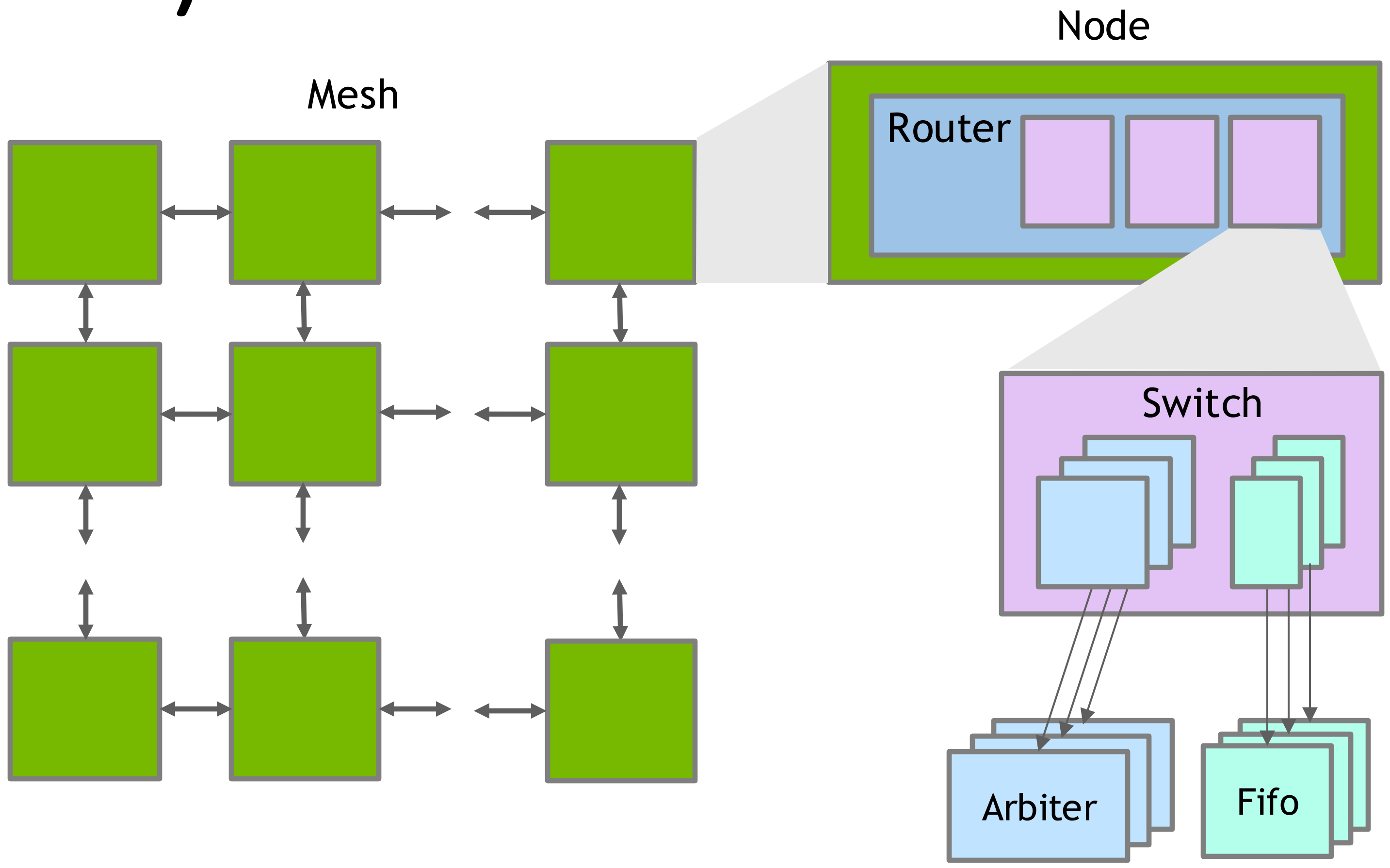
NVIDIA Corp.

Agenda

- Network-on-chip (NoC)
- Formal Strategy
- Node Formal test plan
- Switch Formal test plan
- Abstraction strategies
- Bug Examples
- Results
- Benefits of this approach

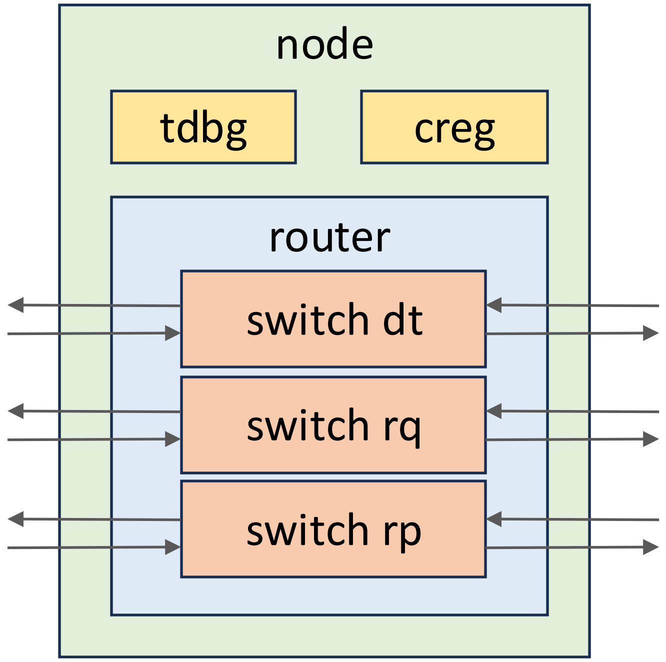
Network-on-chip (NoC)

- Mesh
  - Node (16x to many)
    - Router
      - Switch (3x to 9x)
        - Arbiter (many)
        - Fifo (many)



NoC: Node

- 4 Types: ASN,BSN,CSN,MSN
  - Tdbg: test, clock, reset, debug
  - Creg: control registers



NoC: Formal Strategy

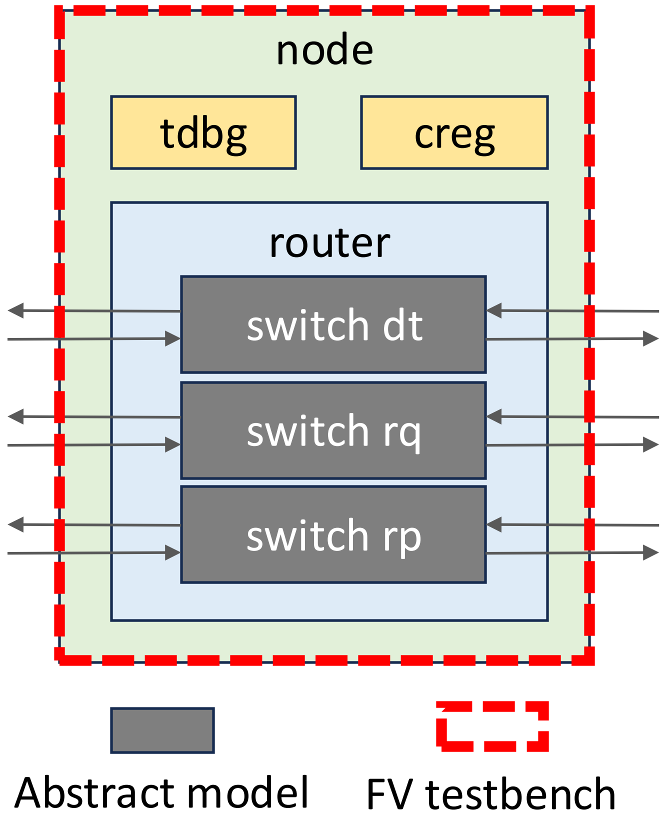
- Starting from the high-level module, create a formal testbench for the entire design
  - Develop relevant properties such as guaranteed data delivery and forward progress checking
  - Verify as many formal properties as possible at this level
  - Move to the next step if undetermined properties remain

- Identifying the complex blocks by analyzing the bounded properties
  - Blocks causing undetermined results (e.g., routers in the NOC).
  - Modules with high complexity (e.g., fifos, arbiters, and data paths in routers).

- Turing the identified complex blocks into abstract models
  - Replace the RTL implementations with simpler models or assumptions.
  - Use proven assertions from the lower level to create assumptions for the boundary boxes.

NoC: Node Formal Testbench

- Formal testbench
- Abstract model
  - Switch dt, rq, rp

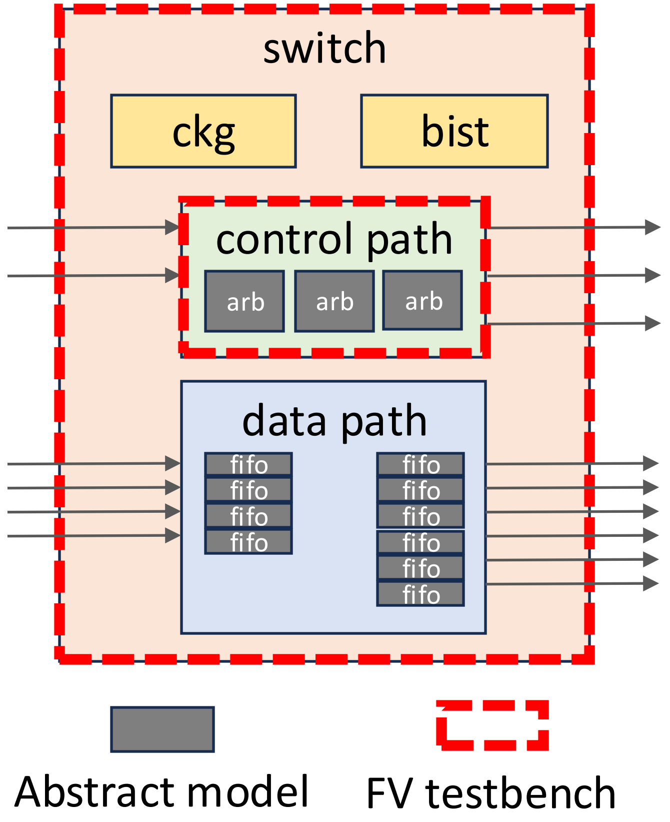


NoC: Node Formal test plan

Task	Planning	Implementation	Closure
Block	Divide and conquer: Use abstract models to re-place switch units	Capture Interfaces: Cache, memory interfaces Bridge, IO, auxiliary interfaces Switch interfaces	Total 1300+ constraints simulation integrated
Function	Prioritize: Node-to-node interfaces Steering logic Credit flow path Data flow path	End-to-End Checking: Credit checking: Src to dst credit & valid Data integrity checking: packet transfer, parity, latency Forward progress checking	Total 1700+ properties 80+ simulation-resistant issues and bugs
Complexity	Decompose: Design scaling Testbench functionality Helper/cover assertions	Abstraction Techniques: Reset abstractions Symbolic sets for data transfers	Formal Coverage: Functional coverage Assertion COI coverage Required proof depth

NoC: Switch Formal Testbench

- Formal testbench
- Switch
  - control path
    - abstract model: arbiters
  - data path
    - abstract model: fifos



NoC: abstraction strategies

- FIFOs: Symbolic depth between 1 and MAX_DEPTH, stable depth during formal runs, full when size reaches depth, and cannot pop elements if size is 0.
- Credit Reduction: Ensure max_credit is within [1, MAX_CREDITS] for formal verification, finding resource issues like starvation/deadlock around the last available credit soon.
- Memories: Symbolic abstraction for reading/writing a single address location with a minimum 1-cycle delay.
- Memories with Bypass: Data sent to RAM or directly read into output with 0 or more cycle delay.
- RFC (8R/8W) Memories with Bypass: Multiple simultaneous reads/writes, exclusive and exhaustive write address ranges, and bypass saves 1 cycle if read address matches write addresses.

NoC: Switch Formal test plan

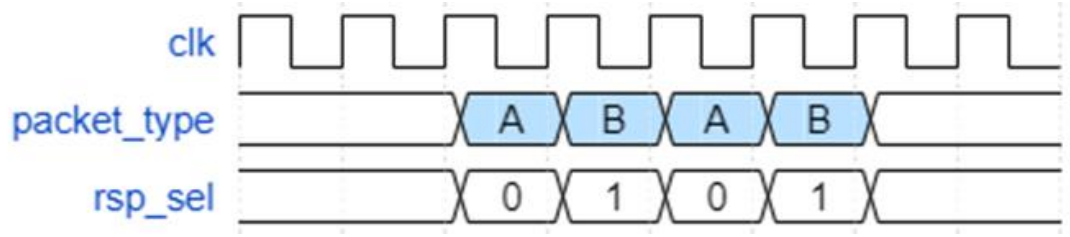
Task	Planning	Implementation	Closure
Block	Divide and conquer: Interfaces, data paths, and control paths	Capture Interfaces: Client inputs/outputs Target inputs/outputs	Validate Constraints: Simulation integrated
Function	Prioritize: Design integrity, forward progress, prioritization, deadlock, and starvation	End-to-End Checkers: Target arbitration Data integrity (Wolper) Forward progress checks	Issues found when stressed under different traffic distribution and arbitration schemes
Complexity	Design scaling: Reduce sizes of storage elements, ports, bursts, and transfer credits.	Abstraction Techniques: Use symmetric elements and symbolic variables on client and target pairs	Formal Coverage: Line: 100% Condition: 100%

Ipshita Tripathi, Ankit Saxdna, et al., "Process & Proof for Formal Signoff - Live Case Study," DVCon 2016

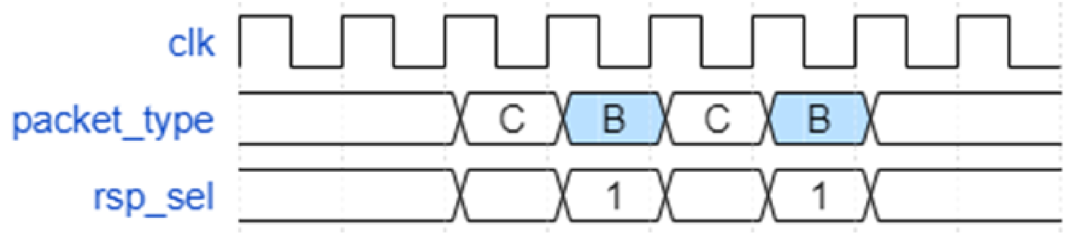
NoC: Bug Examples

- Steering logic issues
  - There are many corner cases when packets flow through different kinds of nodes.
- Connectivity issues
  - incorrect wire connections
  - stuck-at-signal connections
  - data-loss connections
  - timeout connections,
- Crediting issues
  - Credit tracking, and end-to-end checkers
  - to ensure that credits were not lost and returned to the originating sources correctly

Waveform of fair round robin arbitration



Waveform of unfair round robin arbitration



NoC: Results

Nodes	Interfaces	Registers	Logics	Constraints	RTL Asserts	FV Checkers
ASN w/ Switch Full	412	19K	320K	952	1743	502
ASN w/ Switch AM	412	11K	282K	1372	527	732
ASN w/ Switch BBX	412	10K	241K	952	527	502
BSN w/ Switch Full	48	203K	751K	1150	3170	890
BSN w/ Switch AM	48	75K	661K	1360	790	890
BSN w/ Switch BBX	48	27K	419K	1150	790	890
CSN w/ Switch Full	120	69K	1890K	12108	9245	17464
CSN w/ Switch AM	120	42K	1080K	10878	4327	17464
CSN w/ Switch BBX	120	41K	1078K	9898	4327	17244
MSN w/ Switch Full	32	13K	271K	837	1565	590
MSN w/ Switch AM	32	10K	224K	1233	490	746
MSN w/ Switch BBX	32	10K	209K	837	490	590

Benefits of this approach

- Hierarchical Refinement
  - Simplifies complex designs by focusing formal verification on critical blocks.
  - Reduces undetermined properties step by step.
- Assume-Guarantee Efficiency
  - Proven properties from higher levels reduce complexity for lower-level verification.
  - Ensures logical consistency across abstraction levels.
- Focus on Problematic Blocks
  - By iteratively targeting bounded properties, effort is concentrated on the most complex or problem-atic areas.
- Verification of Data Integrity Across Levels
  - Ensure data integrity and functional equivalence at each boundary:
  - Cross-check inputs and outputs between boundary-boxed blocks
  - Validate assumptions by verifying end-to-end integrity.

2025
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 24-27, 2025

Q&A

accellera
SYSTEMS INITIATIVE