2026 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SANTA CLARA, CA, USA
MARCH 2 - 5, 2026

Scaling Formal Verification of Network-On-Chip Using Path Decomposition

Bilal Ahmed, Umar Yaqoob, Bilal Zafar, Misbahud din
10xEngineers

10x ENGINEERS

Agenda

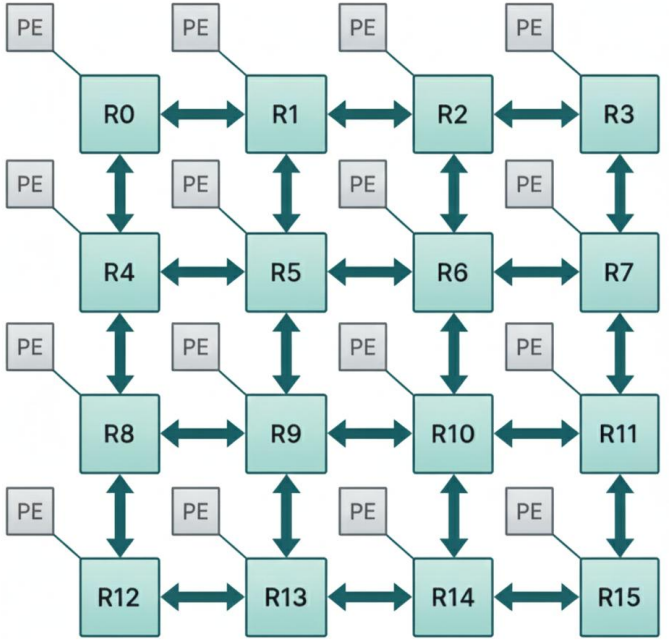
- Motivation
- Network on Chips
- NoC Properties
- Formal Verification Strategy
- Abstraction Strategy
- Path Decomposition
- Bug Examples
- Results
- Benefits of the Approach
- Future Work

Motivation

- NoC is one of the most critical communication backbone of modern large-scale SoCs
- Simulation struggles to prove livelock and deadlock freedom
- Naive formal approaches fail to converge beyond small NoC instances
- Formal verification scalability becomes a challenge with network size and topology
- A scalable formal sign-off methodology is still missing

Network On Chip(NoC)

- Networks-on-Chip provide scalable, high-bandwidth communication for modern multi-core and heterogeneous SoCs.
- Their distributed routing and buffering make exhaustive verification challenging.
- Router micro-architecture includes:
  - Routing decoder
  - Arbitration logic
  - Multiple input/output FIFOs
  - Distributed channels and flow control across the network



NoC Properties

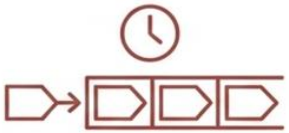
Simulation-based techniques struggle with the concurrency of NoCs and cannot guarantee the absence of rare bugs like deadlock or starvation. Formal verification offers exhaustive proofs, but must cover these critical global properties.

Deadlock Freedom

The network must never reach a circular dependency where packets cannot advance.

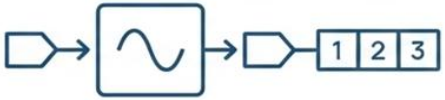
Livelock Freedom

Packets must not circulate indefinitely without delivery.



Starvation Freedom

Every packet must eventually make forward progress.



Data Integrity & Ordering

Packets arrive at their correct destination, uncorrupted, and in sequence.

Liveness of Injection

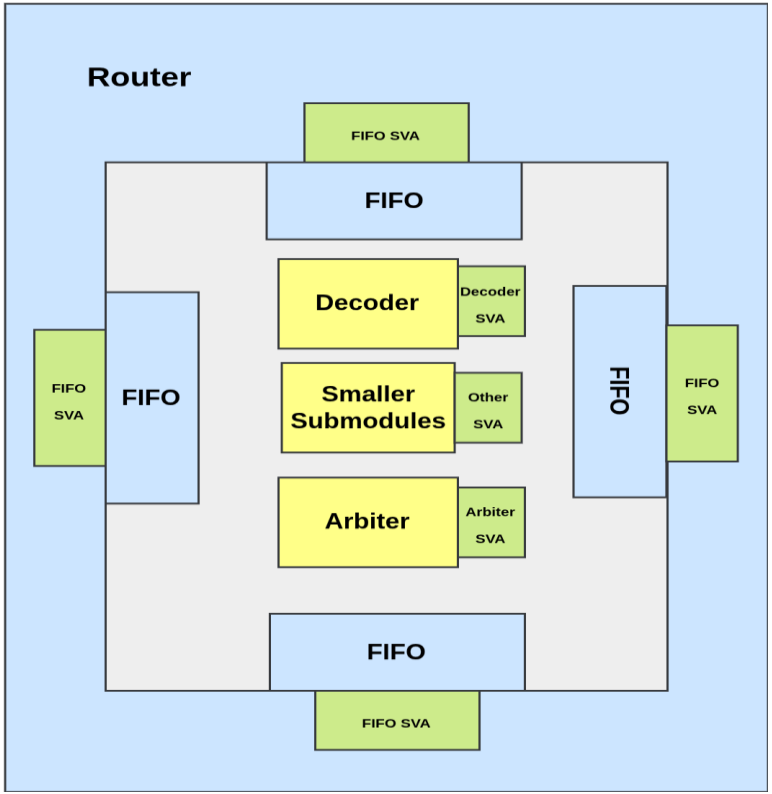
The system must make progress under continuous injection.

Formal Verification Strategy

- Compositional reasoning
  - Verify individual blocks, decoder, router, arbiter, fifos etc.
- End to End Verification
  - Packet Tracker Methodology
  - Parameter Reduction
  - Case Splitting
  - Packet Coloring
  - Path Decomposition

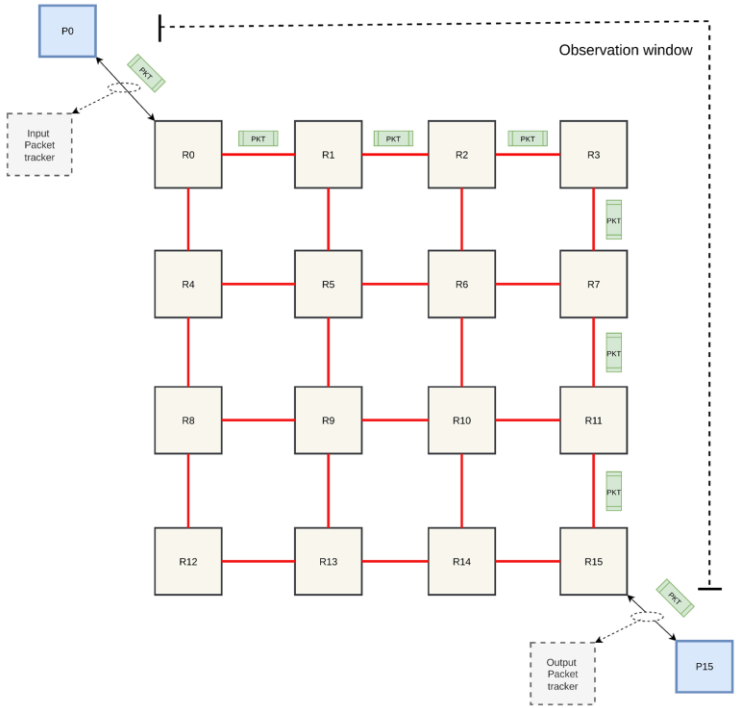
Compositional Reasoning

- Decompose the NoC into independently verifiable components(routers, arbiters, FIFOs, channels)
- Prove local correctness properties for each component
- Assume local properties as guarantees to help prove network-level properties
- Avoid flat, monolithic formal analysis, enabling scalability beyond small NoC instances



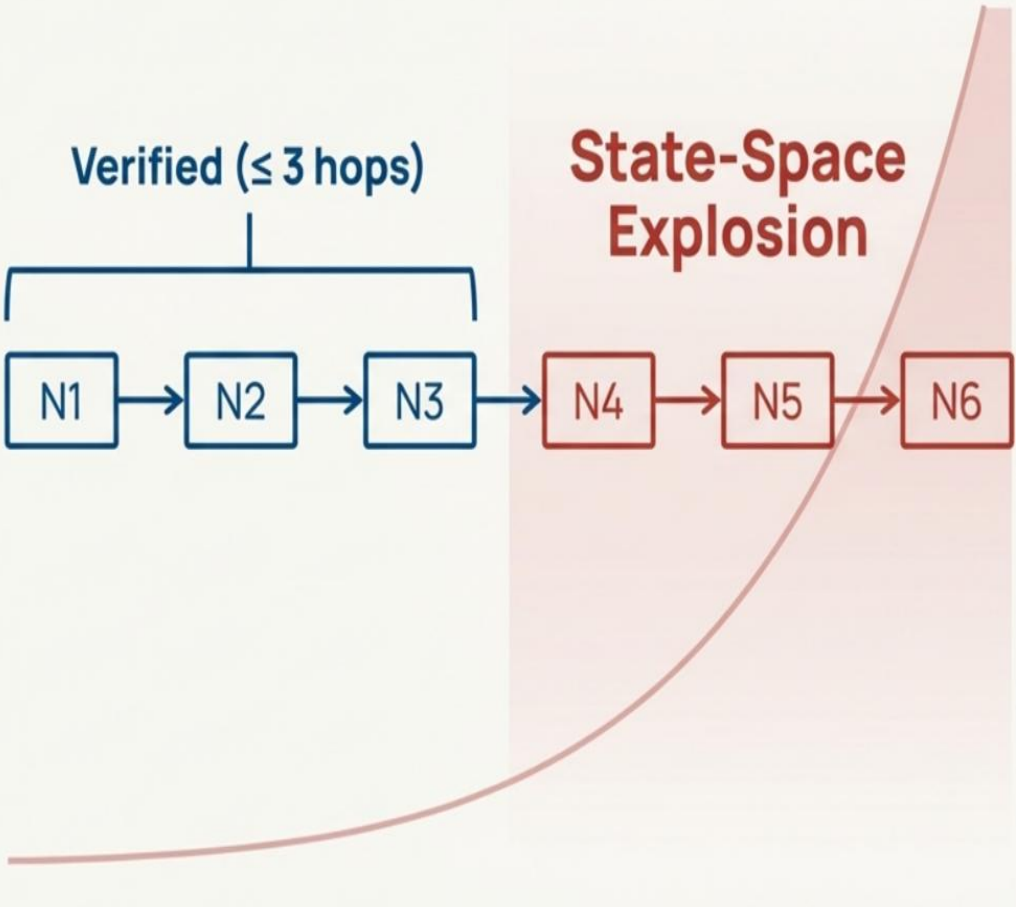
Packet Tracker Methodology

- End-to-end packet tracking methodology
- Symbolic packet tracking, ensuring packet integrity and ordering from source to destination
- Packet coloring to keep track of source/destination
- Case splitting to track each path separately
- Even after abstraction, longer paths still struggle to converge



Scalability Wall

- Despite the advances, end to end proofs hit scalability wall
- Proves for shorter paths 2-3 hops
- As path length grows, state space expands super exponentially
- Extrapolated time for 6 hops was calculated to be ~2 years!

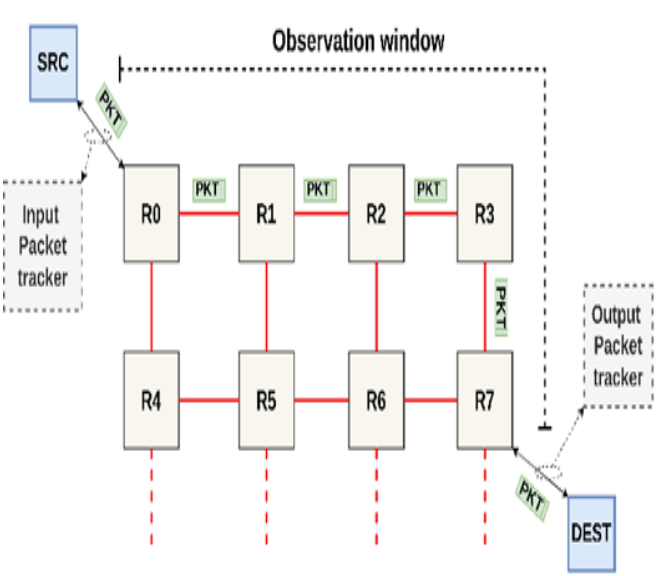


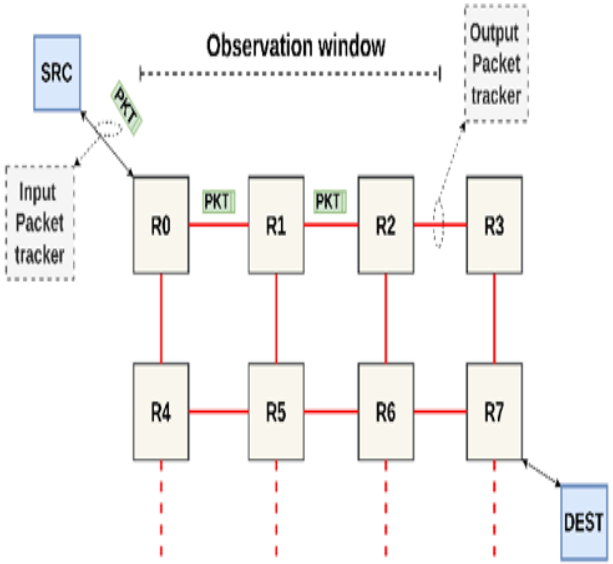
Our Solution: Path Decomposition

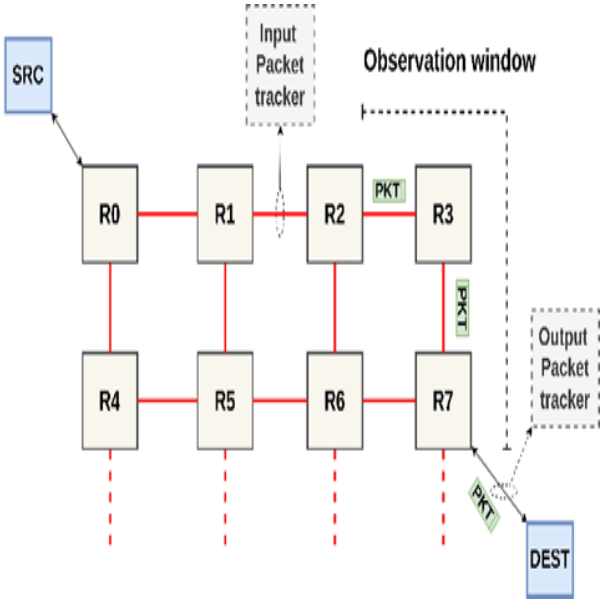
- Predictive routing model in Python
- Longer Paths are broken down into subpaths (divide and conquer)
- Overlapping subpaths to ensure correctness
- Intractable problem to a tractable problem
- Turning a super exponential problem to a near linear problem
- If the average verification time for a two-hop subpath is T_2 , then the total verification time for a path of k hops can be approximated as:

$$T_k \approx \frac{k}{2} \times T_2$$

Path Decomposition(Cont...)







Results

- Complete end-to-end verification of all 240 source–destination pairs in a 4×4 mesh NoC using path decomposition
- Liveness properties (deadlock and starvation freedom) converged within 72 hours
- All experiments run on a standard workstation (Intel Core i5, 16 GB RAM)
- Achieved formal coverage closure across all active modules after excluding dead code, ensuring exhaustive property checking

Results(Cont...)

NoC Type	Configuration	Verification Status	Coverage
BSG NoC	2x2 Simple Mesh	Complete	100%
BSG NoC	4x4 Simple Mesh	Complete	100%
BSG NoC	4x4 Half/Full Ruche	Complete	100%
BSG NoC	8x8 & 16x16 Mesh	Partial (Proof of Scalability)	N/A
Floo NoC	4x4 Simple Mesh	Partial (Proof of Scalability)	N/A

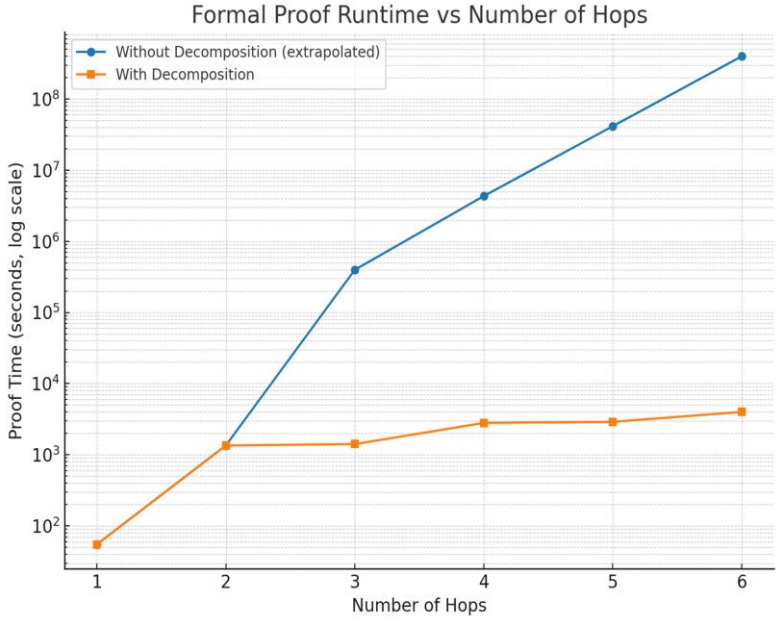
Bug Examples

Exhaustive formal verification revealed corner-case bugs missed by simulation in NoCs under testing.(Safe statement) (Mutation Analysis)

- **Packet dropping bug:**
  - Improper synchronization between FIFO readiness and output valid signals caused packets to be dequeued but never forwarded
- **Arbitration bug:**
  - Rare overlap of route computation and buffer arbitration led to incorrect output port selection and packet misrouting
- Formal counterexamples provided exact destination coordinates and cycle-accurate timing, enabling targeted fixes in flow-control and arbitration logic

Benefits of this approach

- Enables scalable formal sign-off for large NoCs by conquering state-space explosion
- Reduces exponential verification complexity to near-linear scaling problem
- Delivers complete end-to-end guarantees for deadlock freedom, livelock freedom, and packet integrity
- Achieves **~10^{5x}X** proof speedup and **~99%** formal coverage on standard workstation hardware



Future Works

- Extend the framework to other NoC topologies
- Support adaptive and non-deterministic routing algorithms
- Automate assumption generation and abstraction refinement for improved usability
- Apply the methodology to heterogeneous and hierarchical NoC architectures

2026
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SANTA CLARA, CA, USA
MARCH 2 - 5, 2026

Q&A