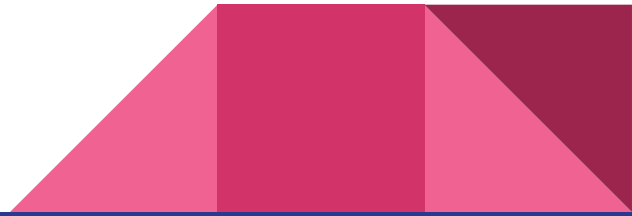
Sparse Hamming Graph: A Customizable Network-on-Chip Topology

Patrick Iff, Maciej Besta, Matheus Cavalcante, Tim Fischer, Luca Benini and Torsten Hoefler

Scalable Networks-on-chip (NoCs)

- Customization can lead to diverse design goals of different chips
- Develop a toolchain that can deliver fast and accurate cost and performance predictions
- Sparse Hamming graph
- Four fundamental NoC topology design principles
  - Low cost
    - Low radix topologies
    - Design for link routability
  - High performance
    - Minimize the network diameter
    - Minimize the physical path length



Design Principles

- Assume a chip is organized as an R xC grid of identical building blocks (tiles)
  - Contains one or more endpoints and a local router where all endpoints are connected
- NoC is used to provide connectivity between tiles
- NoC's links are attached to the tile's local routers
- If a link is too long to be operated at target frequency, insert as many registers as necessary to meet the frequency
- Tiles occupy all available metal layers which disallows routing an inter-tile link between two tiles A and B over a third tile C

Reduce Cost

- Use low-radix topologies
  - Area is 4 times the router radix
  - A higher radix implies a higher overall number of links
- Design for routability
  - Short links
  - Aligned Links
  - Uniform Link Density
  - Optimized Port Placement

Boost Performance

- Minimize the network diameter
  - Diameter = max number of routers-to-router hops that a flit takes on the path from its source-tile to its destination tile
  - Flit traversing a router lead to a delay
    - Minimizing the network diameter reduces latency
  - Congestion also gets reduced
- Minimize the Physical Path Length
  - Minimizing the physical distance that a flit travels is crucial to achieve low latency
  - Needs paths that provide minimal physical distance
    - Contain links that provide physically minimal paths
    - Be co-designed with the routing algorithm to use these paths without reducing the throughput

The Sparse Hamming Graph NoC Topology: Concept

- Customizable sparse Hamming graph topology
  - Based on four Noc topology design principles
    - Provides low router radix
    - Design for routability
    - Result in low latency
    - Unfortunately has a high network diameter
      - Add additional links to the topology
      - Depends on the two input parameters - enable an effortless adjustment of its cost performance trade off
        - The way links are added ensures a good routability of links is maintained
        - Flattened butterfly (low network diameter and excellent performance)

Overview of Prediction Toolchain

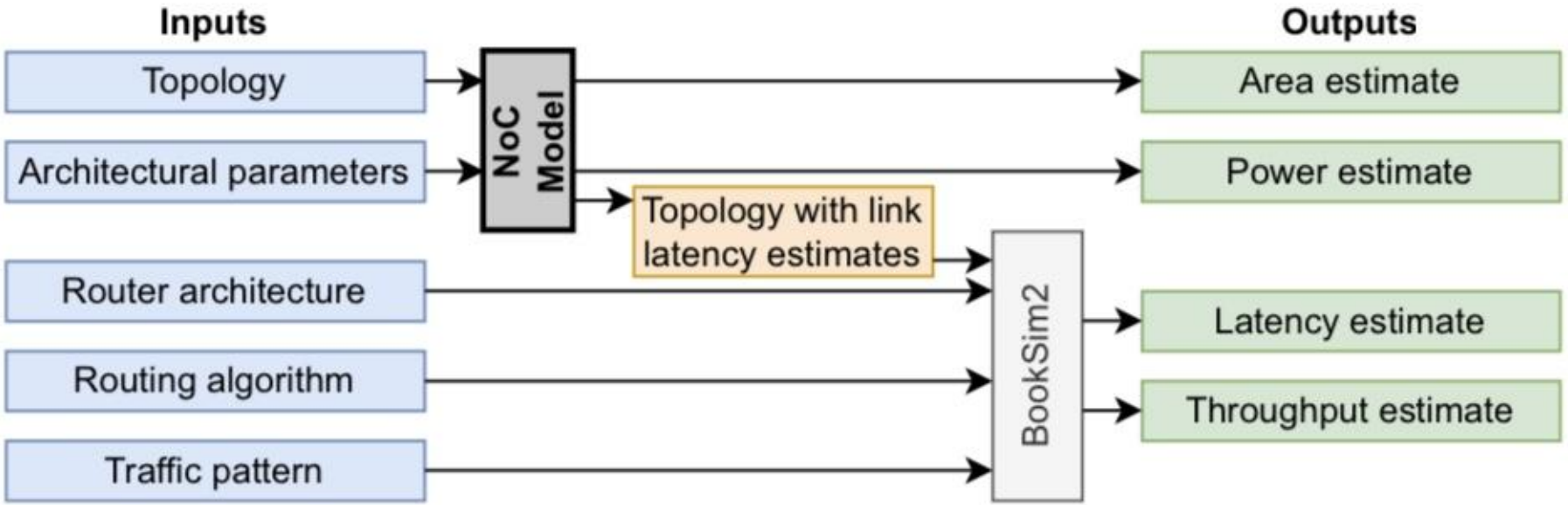


Fig. 3: Toolchain to predict performance and cost metrics of a NoC.

Prediction Model

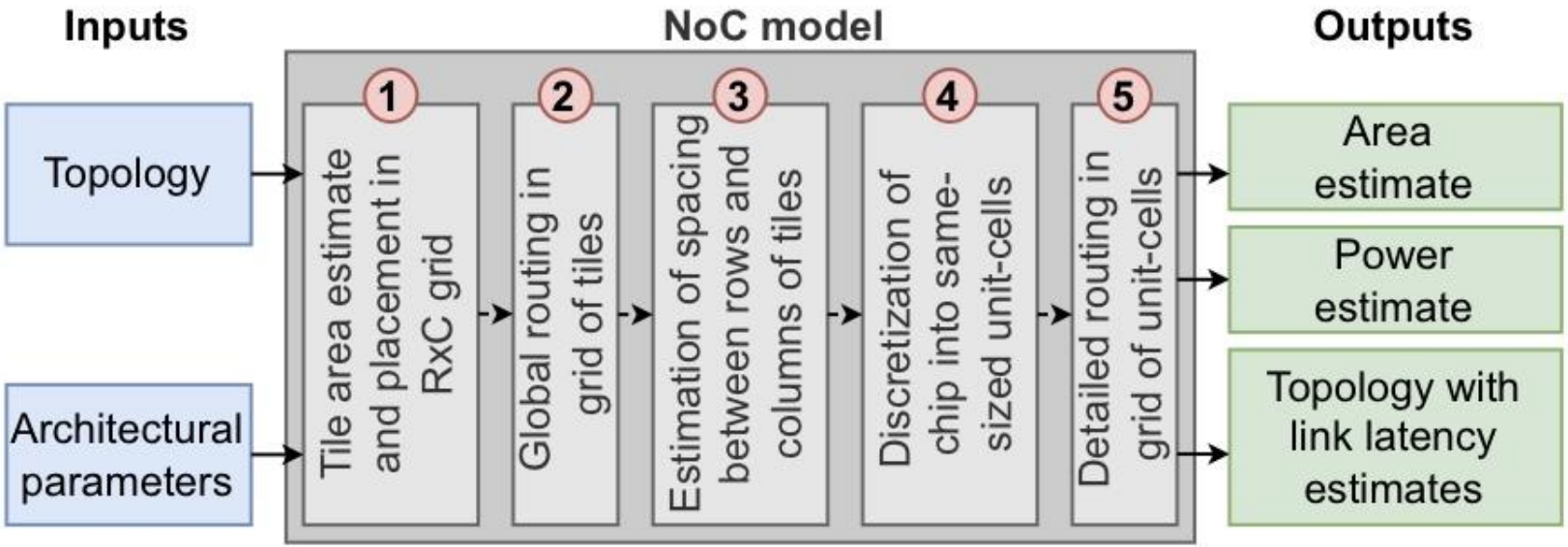


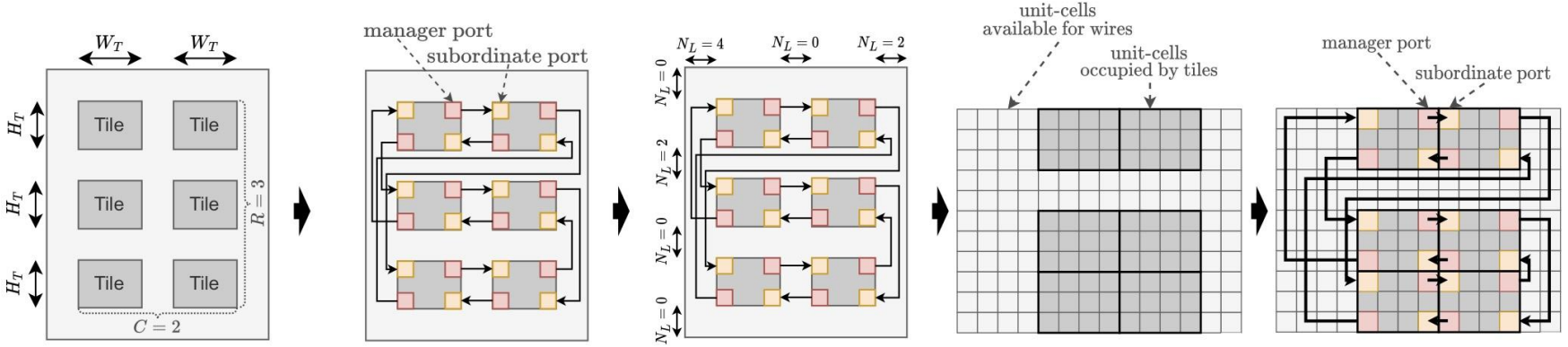
Fig. 4: Our model to predict area overhead, power consumption, and link latencies of a NoC.

TABLE II: Architectural Parameters Needed as Model Inputs.

Parameters describing the chip design	
N_T	Number of tiles
A_E	Combined area of all endpoints (cores + local memories) in a tile in gate equivalent (GE)
R_T	Aspect ratio of a tile (height:width)
Parameters describing the NoC	
F	Frequency at which the NoC is run
B	Bandwidth of each router-to-router link
Parameters describing the technology node	
$f_{\text{GE} \to \text{mm}^2}(x)$	Function; area (in mm^2) needed to synthesize x GE of logic.
$f^H_{\text{wires} \to \text{mm}}(x)$	Function; space (in mm) needed to manufacture x parallel, horizontal wires.
$f^V_{\text{wires} \to \text{mm}}(x)$	Function; space (in mm) needed to manufacture x parallel, vertical wires.
$f^L_{\text{mm}^2 \to \text{W}}(x)$	Function; approximate power consumption (in W) of x mm^2 of logic-dominated area.
$f^W_{\text{mm}^2 \to \text{W}}(x)$	Function; approximate power consumption (in W) of x mm^2 of wire-dominated area.
$f_{\text{mm} \to \text{s}}(x)$	Function; time (in s) it takes a signal to travel a distance of x mm along a buffered wire.
Parameters describing the on-chip transport protocol	
$f_{\text{bw} \to \text{wires}}(x)$	Function; number of wires needed to build a link with a bandwidth of x bits/cycle
$f_{A_R}(m, s, B)$	Function; area (in GE) of a NoC router with m manager ports, s subordinate ports and bandwidth B

Architectural Parameters

Model Construction



(a) Step 1: Tile Area Estimate and Placement in $R \times C$ Grid.

(b) Step 2: Global routing in grid of tiles.

(c) Step 3: Estimation of spacing between rows and columns of tiles.

(d) Step 4: Discretization of chip into same-sized unit-cells.

(e) Step 5: Detailed routing in the grid of unit-cells.

Fig. 5: The five steps in our model for predictions of area, power consumption, and link latencies.

Toolchain Evaluation

TABLE III: Cost and Performance results and predictions of the MemPool Architecture [37].

Metric	Correct Value	Prediction	Prediction Error
Area	21.16 mm^2	24.26 mm^2	15%
Power	1.55 W	1.447 W	7%
Latency	5 cycles	10 cycles	100%
Throughput	38%	25%	34%

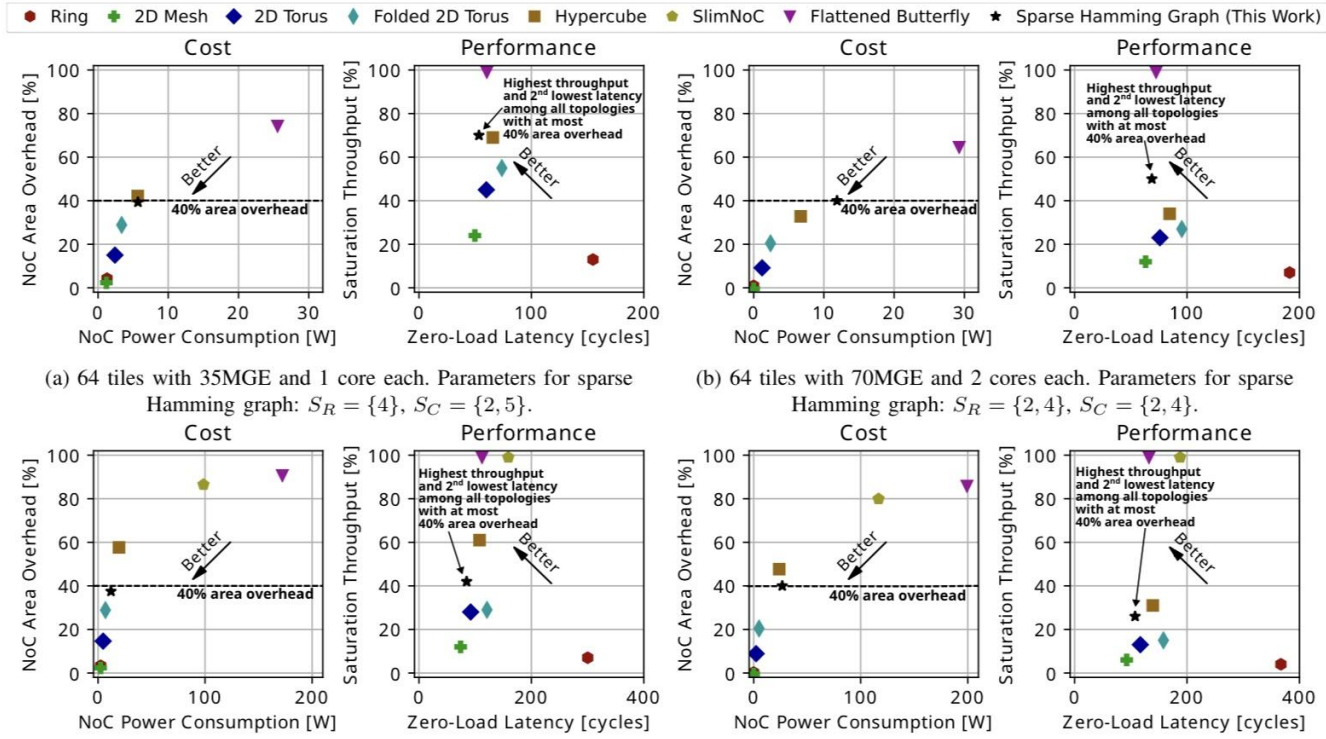
NoC Topology Customization Strategy

- **Step 1**: Start with the simplest sparse Hamming graph topology, which is a mesh ($S_R = \{\}$, $S_C = \{\}$).
- **Step 2**: Use prediction toolchain to estimate performance of and cost of current topology if applied to the target architecture with its unique architecture parameters.
- **Step 3**: Compare the estimates from step 2 to design goals to identify insufficiencies of current topology.
- **Step 4**: Follow design principles to change the parameters S_R and S_C to eliminate the insufficiencies identified in step 3.
- **Step 5**: Go back to step 2 and repeat until finding the satisfied performance and cost.

Target Architectures & Evaluation Results

ASSUMED DESIGN GOAL

Maximize the throughput and minimize the latency without exceeding an area overhead of 40%



(a) 64 tiles with 35MGE and 1 core each. Parameters for sparse Hamming graph: $S_R = \{4\}$, $S_C = \{2, 5\}$.

(b) 64 tiles with 70MGE and 2 cores each. Parameters for sparse Hamming graph: $S_R = \{2, 4\}$, $S_C = \{2, 4\}$.

(c) 128 tiles with 35MGE and 1 core each. Parameters for sparse Hamming graph: $S_R = \{3\}$, $S_C = \{2, 5\}$.

(d) 128 tiles with 70MGE and 2 cores each. Parameters for sparse Hamming graph: $S_R = \{2, 4\}$, $S_C = \{2, 4\}$.

Fig. 6: Comparison of various topologies for four different scenarios. The comparison is performed using our prediction toolchain with a random uniform traffic pattern and a routing algorithm that minimizes the number of router-to-router hops. SlimNoC is only applicable for scenarios c) and d) because it requires the number of tiles N_T to be $N_T = 2p^2$ for a prime power p.

Contributions

1) A set of NoC topology design principles that reveal how to influence the NoC's cost and performance
2) The customizable sparse Hamming graph topology with and adjustable cost-performance trade-off
3) A fast and easy-to-use toolchain for predicting the NoC's performance and cost featuring our custom model for area power and link latency estimates

THANK YOU!

References:

Iff, Patrick & Besta, Maciej & Cavalcante, Matheus & Fischer, Tim & Benini, Luca & Hoefler, Torsten. (2023). *Sparse Hamming Graph: A Customizable Network-on-Chip Topology*.