3D Torus with & Elevator Mapping

Vivek Lasi
University of Waterloo
Waterloo, Canada
vlasi@uwaterloo.ca

Gavin Lusby
University of Waterloo
Waterloo, Canada
gdplusby@uwaterloo.ca

Damir Gazizullin
University of Waterloo
Waterloo, Canada
dgazizul@uwaterloo.ca

***Abstract*—This report introduces a flexible and user-friendly way of quantifying and improving the positioning of vertical channels (elevators) in a 3D Torus. The tool allows changing the number and placement of elevators, configuring latency penalties of vertical channels, and quantifying performance based on throughput, latency and maximum sustainable injection rate.**

I. Introduction

With the rise of chiplets and 3D chip design in the context of AI Accelerators and parallel computing, 3D NoCs have become more important than ever before. A lot of research has already been done on the topic, but it tends to neglect the unquestionable prestige of the basic torus [1]. The cost per performance of the torus is indeed unquestionable, but modern applications prefer high performance. Thus, we propose a bidirectional mesh only for the Z-direction to reduce the effects of bottlenecks in 3D torus NoCs while maintaining the cost benefits of the basic torus.

II. Background & Proposal

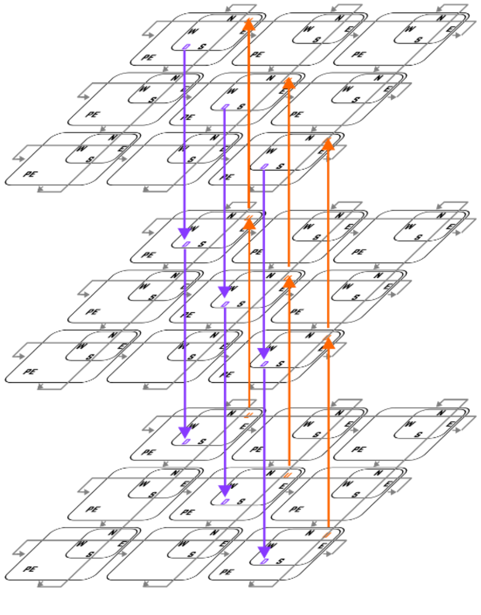
In the labs, we implemented a credit-based flow control system using XY DOR routing with unidirectional toruses and wrap-around. We wanted to extend this to the third dimension due to the greater feasibility of chiplets, thanks to TIV and evolutionary hybrid bonding techniques. The main problem here is that TIVs are at least greater than ~1um [2] and usually do not justify the area and delays. Our proposal is to alter the original torus to enable bidirectional meshing in the third dimension, as the dimension's latency cost is sufficiently high to justify adding more complex logic and area for the NoC. The industry gets around this issue by exploiting locality and reducing the number of elevators. Thus, the second goal of this paper is to identify the optimal elevator location and develop an effective elevator mapping system.

III. Implementation

A. Booksim Implementation

Booksim2 was modified to accept an alternative network topology, allowing for a different topology in the z direction. The routing method used is similar to the lab's DOR X-first routing, where Y has the highest priority, followed by X. The change here is that first X-first is used to reach the elevator, then Z has priority, and then X-first is used again to reach the destination. This means that Z has more priority than Y, which has more priority than X. This is called elevator first routing, which is known to be livelock and deadlock-free [3]. The difference here is that the Z dimension is now a bidirectional mesh, allowing for more ports and increased bandwidth. A DOR Allocator had to be implemented in Booksim, as it did not initially support a unidirectional torus. The connectivity and elevator mapping were tested using our implementation of a single packet traffic pattern to verify hop counts.

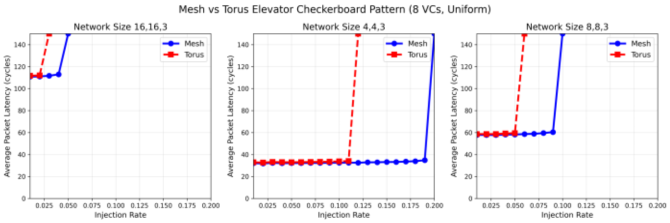
Furthermore, we have modelled the TIV as having a 3-cycle wait instead of the usual 1-cycle so that the clocks do not have to be decreased to allow a 3rd dimension to the NoC. In the experiment, we present the throughput versus injection rates for different network sizes. This is done for both the general 3d unidirectional torus case and our proposed bidirectional mesh elevator. The VC count was set to 8, and credit-based flow control is being used. Note that the Z dimension is 3, as that is the more practical scenario with 3.5D processes having a TIV layer for chiplets and a base interposer for 2D SiPs [4] Note that in the elevator dimension having two floors, the mesh and torus are the same.

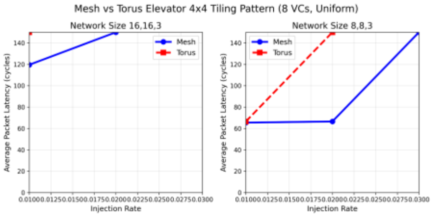


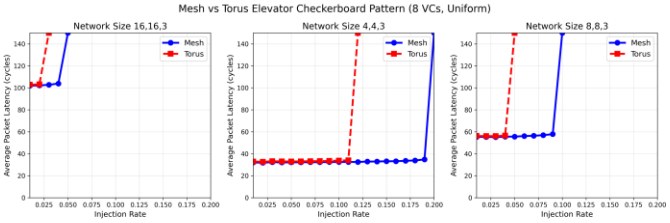
B. Elevator Patterns and Mapping Algorithms

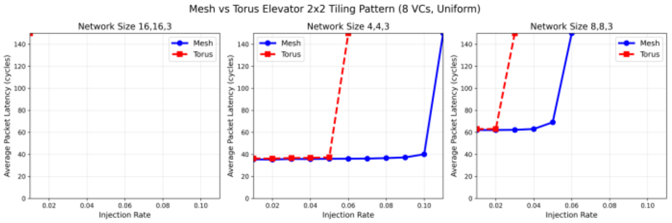
Since we are using DOR the mapping between a given switch with X and Y coordinates (x,y) and the elevator with X Y coordinates (u,v) needs to be predetermined. For this, we created a script that allows those trying to model elevator patterns/mappings to enter in potential elevator patterns, as well as their own custom function to choose which of the provided elevators a given switch with X and Y coordinates (x,y) should map to. The default algorithm provided searches for the nearest elevator by # of hops in the +X and +Y directions. We tried some basic patterns to model different densities of elevators, including some basic patterns to model different densities of elevators, including a diagonal line, a checkerboard pattern, and breaking the chip into sub-tiles, where each sub-tile receives one router

IV. Results









Since we are using DOR, the mapping between a given switch with X and Y coordinates (x, y) and the elevator with X and Y coordinates (u, v) must be predetermined. For this, we created a script that allows those trying to model elevator patterns/mappings to enter in potential elevator patterns, as well as their own custom function to choose which of the provided elevators a given switch with X and Y coordinates (x,y) should map to. The default algorithm provided searches for the nearest elevator by # of hops in the +X and +Y directions. We attempted a diagonal line, a checkerboard pattern, and breaking the chip into sub-tiles, where each sub-tile gets one router.

The results indicate that addressing the critical delays in the NoC can nearly double performance. By adding a Mesh in the Z direction, we can double the max throughput. Multiple elevator patterns verify it, and it is not unique to a network size. In terms of cost, the ports have been doubled for the Z direction and that explains the doubled throughput compared to the original 3D Torus. Area has not been doubled though, as it only has doubled for the Z direction and assuming that's 1/3 of the area, the area has only increased by 33%. The relationship between added elevators and increased throughput is not strictly linear. Notice that for an 8x8 torus with checkerboard elevators (1:1 ratio of elevators to switches), the maximum injection rate is roughly five times better than that of the 4x4 tiling pattern (1:16 ratio), despite having eight times as many elevators. However, the 4x4 tiling pattern may not be of much use anyway, as it seems to offer a maximum injection rate of just 0.02, which may not be sufficient for many use cases.

V. Conclusion and Future Work

To carry on this experiment and results, the following can be done:

- Experiment with other routing algorithms like Adaptive XYZ, Fault tolerance, etc.
- Deadlock and worst-case latency proof instead of references
- More optimal elevator mapping based on application statistics
- Determine the relationship between the number of elevators and maximum throughput

VI. References

[1] R. G. Kunthara, N. K, R. K. James and S. Z. Sleeba, "Interleaved Edge Routing in Buffered 3D Mesh & CMesh NoC," in *2021 8th International Conference on Smart Computing and Communications (ICSCC)*, Kochi, Kerala, 2021.

[2] G. Gao, X. Miao and H. Xuan, "Analysis of 3D stacking technology and TSV technology," in *2023 24th International Conference on Electronic Packaging Technology (ICEPT)*, Shihezi City, 2023.

[3] F. Dubois, A. Sheibanyrad, F. Pétrot and M. Bahmani, "Elevator-First: A Deadlock-Free Distributed Routing Algorithm for Vertically Partially Connected 3D-NoCs," *IEEE Transactions on Computers,* vol. 62, no. 3, pp. 609-615, 2013.

[4] D. Million, C. Fuguet, A. Evans, R. El Cheikh, A. Monemi, J. Balkind and F. Petrot, "Depth-First: A Deterministic and Scalable NoC Routing Protocol for 3.5D Packaged Architectures," *IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS,* 2025.