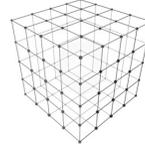




Oregon State
University

Three-Dimensional Network-on-Chip Architectures for Cycle Accurate Full-System Simulator



Presented By:

Akash Agarwal

agarwaak@oregonstate.edu

Under the Supervision of

Dr. Lizhong Chen

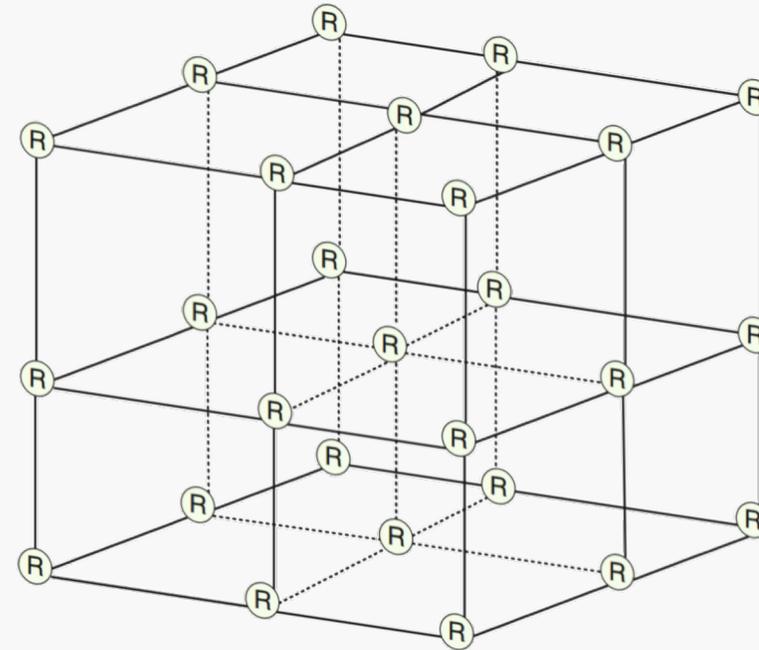
Assistant Professor

EECS, Oregon State University

Contents



- Introduction
- Related Work
 - 3D Die-Stacking Techniques
 - 3D Architectures for Multi-Core
 - Through Silicon Vias Placement
- Garnet Network Model
- Three-dimensional NoC Implementation
 - 3D Mesh NoC
 - 3D-stacked Mesh NoC
 - Optimal TSV-Link Placement
 - Program Flow
- Simulation Results
- Conclusion and Future Work
- References



Introduction



Moore's Observation +
Dennard Scaling

Rise of Multi-core
Architectures

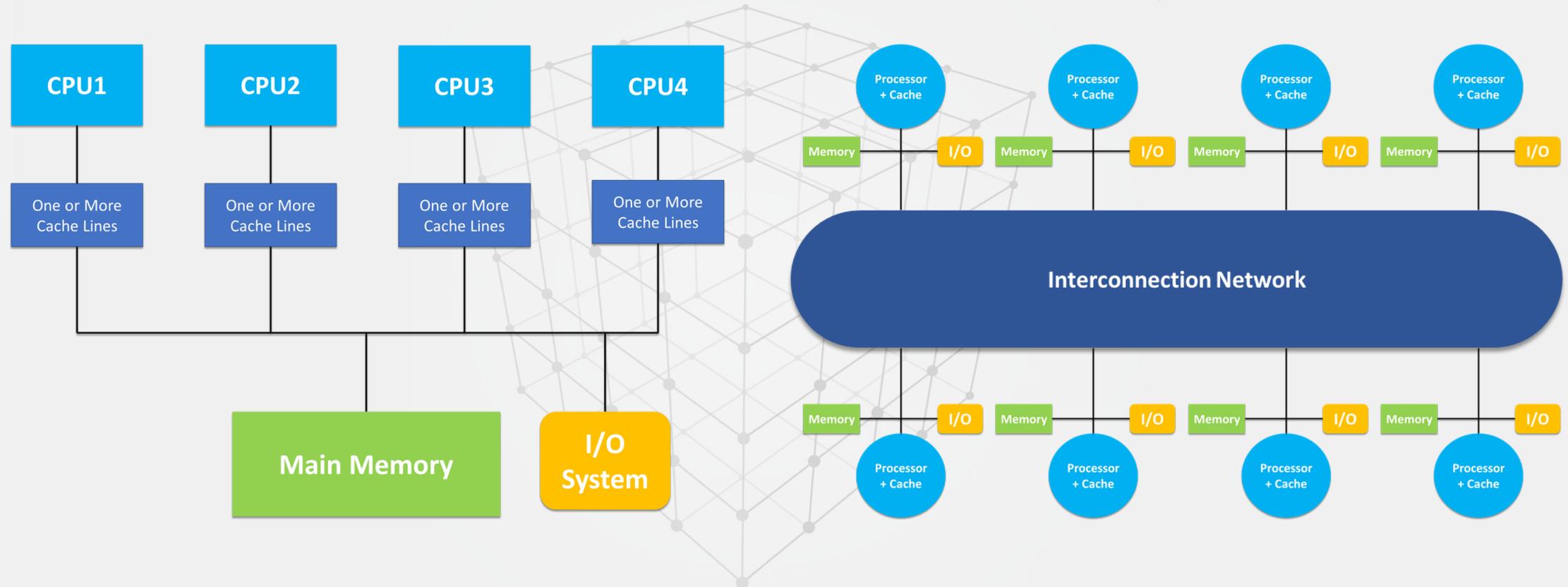
Comparing Memory
Architectures

- Many-core research led to the development of:

Shared-Memory Multiprocessor and **Distributed Memory** architectures.

- Multiple CPUs “sharing” the same main memory.
- Each process can read and write a data item simply using load and store operations, and process communication is through shared memory
- Systems with no shared memory.
- Each CPU must have its own copy of the operating system, and processes can only communicate through message passing

Shared Memory Multiprocessors



Uniform Memory Access (UMA) systems

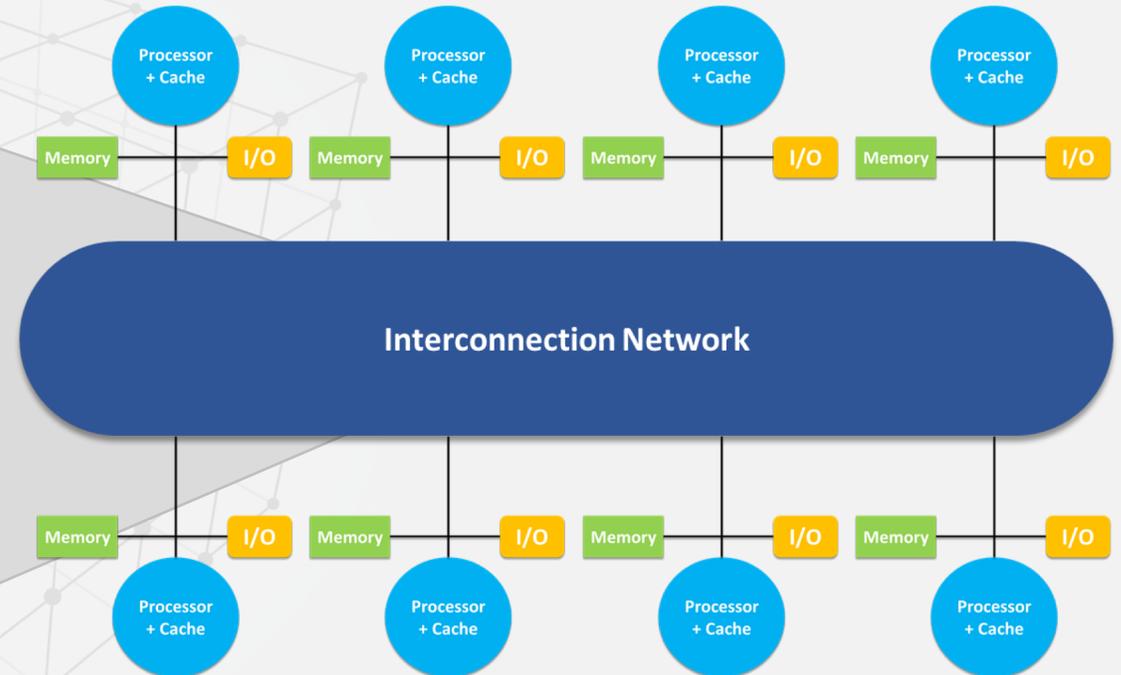
- All processors share a unique centralized primary memory.
- CPUs have same memory access time.
- Less Scalable.

Non-Uniform Memory Access (NUMA) systems

- Physical memory is distributed among CPUs.
- Access time to data depends on data position, in local or in a remote memory.
- Concept used in Chip-Multiprocessors (CMPs).



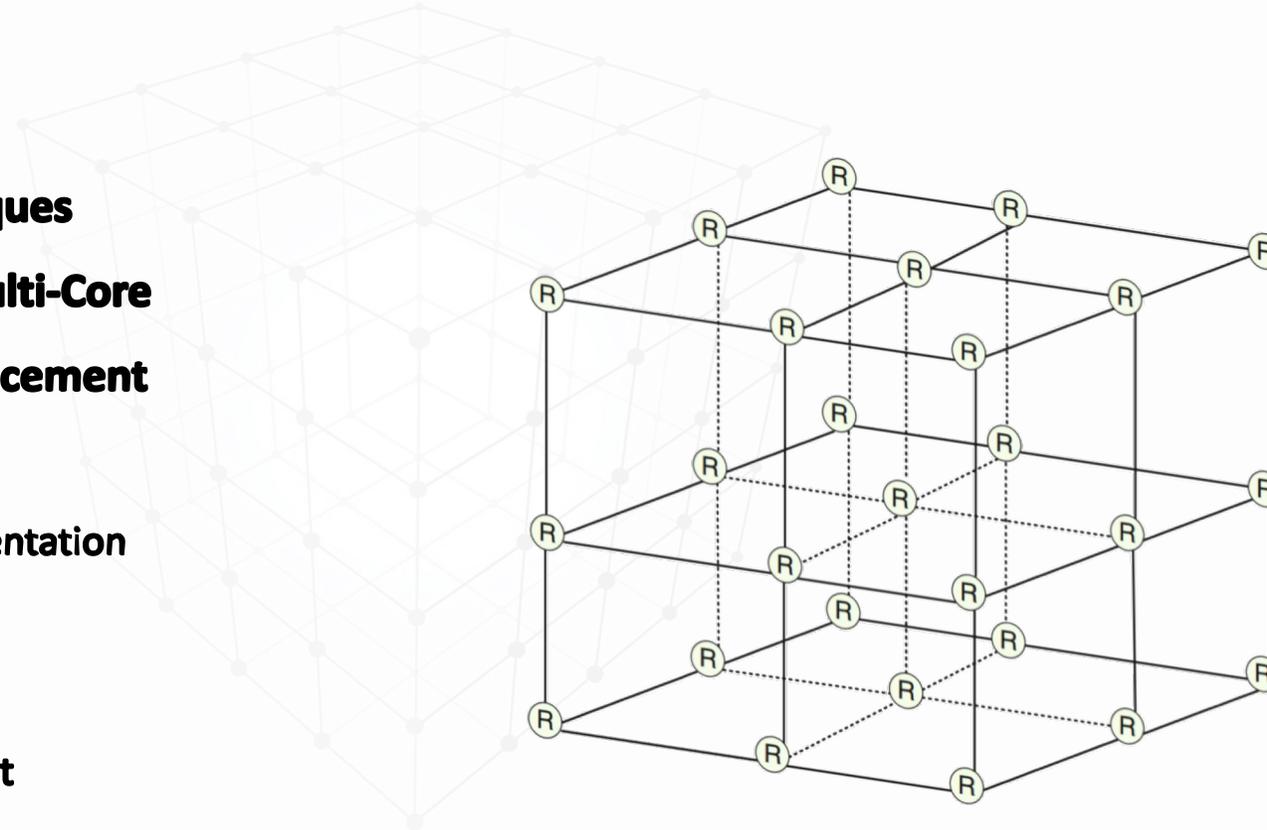
- CMPs: Multiple cores on single chip, connected using **on-chip Interconnect**.
- Interconnections Network became bottleneck in performance.
- One Aspect for NoC Performance Improvement: **NoC Topology**.
- Recent Research in 3D Topologies.
- D2D low latency vias in 3D NoCs help to increase the memory locality even further and hence reduces the cache access time, thus improving performance.



Contents



- Introduction
- **Related Work**
 - **3D Die-Stacking Techniques**
 - **3D Architectures for Multi-Core**
 - **Through Silicon Vias Placement**
- **Garnet Network Model**
- **Three-dimensional NoC Implementation**
 - **3D Mesh NoC**
 - **3D-stacked Mesh NoC**
 - **Optimal TSV-Link Placement**
 - **Program Flow**
- **Simulation Results**
- **Conclusion and Future Work**
- **References**

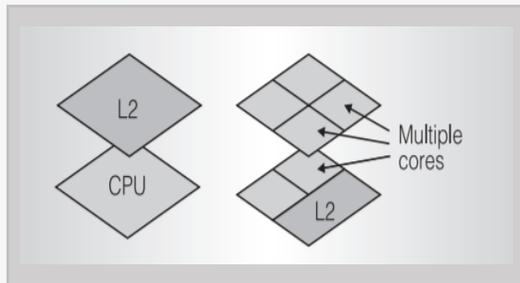


Related Work



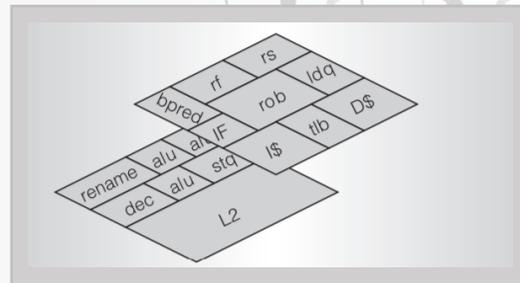
3D Die-Stacking Techniques

Stacking Macroscopic Blocks



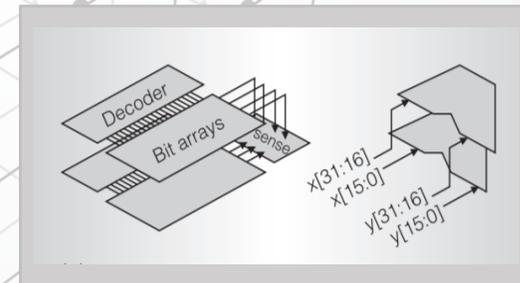
- **Benefits:** reduction in a multicore processor's core-to-core interaction paths
- **Redesign effort:** Reuse existing 2D
- **Example:** Core-on-core, Cache-on-core

Stacking Functional Unit Blocks (FUBs)



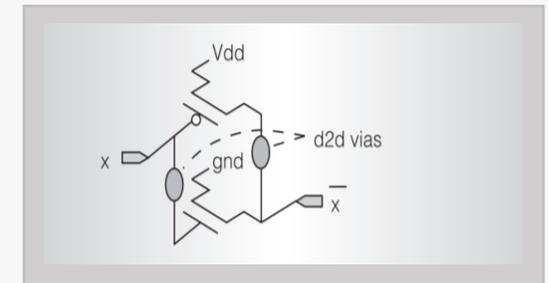
- **Benefits:** Reduced latency and power of global routes
- **Redesign effort:** Re-floorplan and retime paths
- **Example:** ALU-on-ALU

Stacked Logic Gates



- **Benefits:** Elimination of intra-block wiring
- **Redesign effort:** 3D circuit designs and layout tool development
- **Example:** Cache splitting, ALU bit splitting

Stacked Transistors

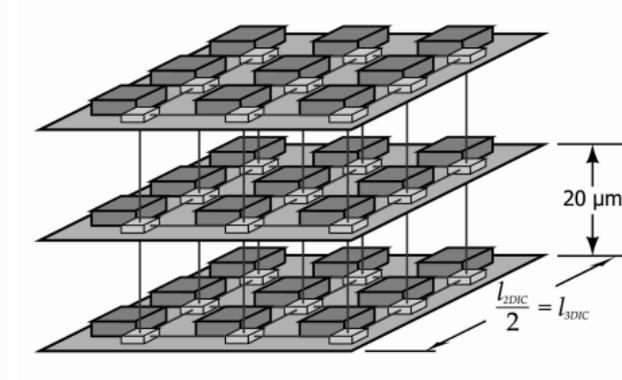


- **Benefits:** reductions in areas, latency and power consumptions (max benefits)
- **Redesign effort:** Almost no reuse
- **Example:** NMOS/PMOS partitioning

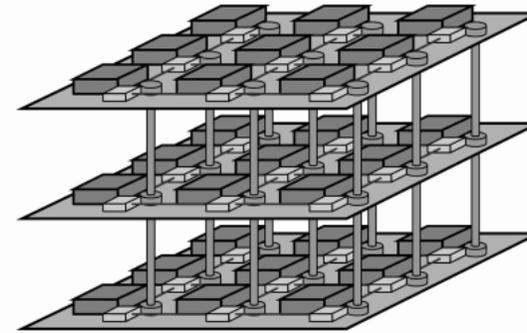
Related Work



3D Architectures for Multi-Core



3D Mesh



3D-stacked Mesh

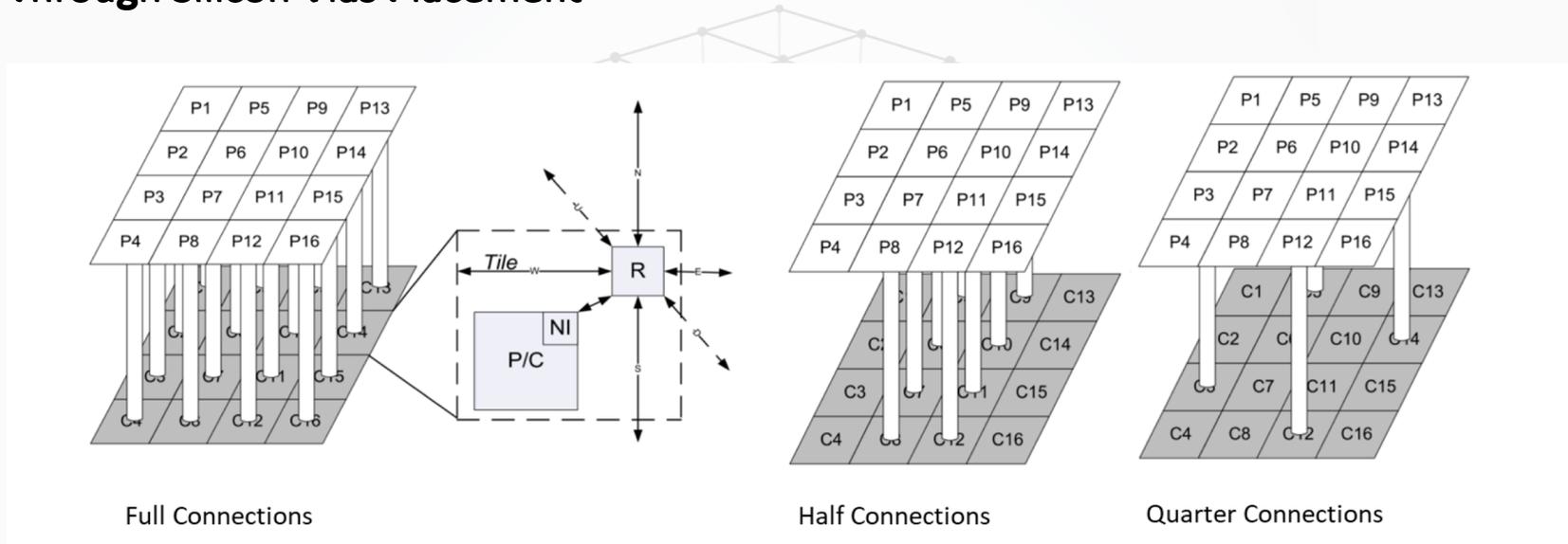
- An extension of its 2D predecessor. Multiple layers connected using vertical links same as the intra-layer links.
- The design generally consists of processing cores uniformly distributed across the $N_1 \times N_2 \times N_3$ 3D mesh. (external controllers directly connected to each node)
- $2 \times 2 \times 2$ 3D Mesh \Rightarrow
 - num_cpus = 8
 - num_rows = 2
 - num_cols = 2
 - num_layers = 2

- It integrates multiple layers of 2D Mesh networks by connecting them with a bus spanning (low-latency TSV links) the entire vertical distance of the chip.
- The design generally consists of stacked layers of different node-types (e.g. stacking a shared-cache).
- Area of die: $\frac{1}{2}$ cores, $\frac{1}{2}$ network components
 $\therefore 2 \times 2 \times 3$ 3D Mesh \Rightarrow
 - num_cpus = 4
 - num_rows = 2
 - num_cols = 2
 - num_layers = 3

Related Work

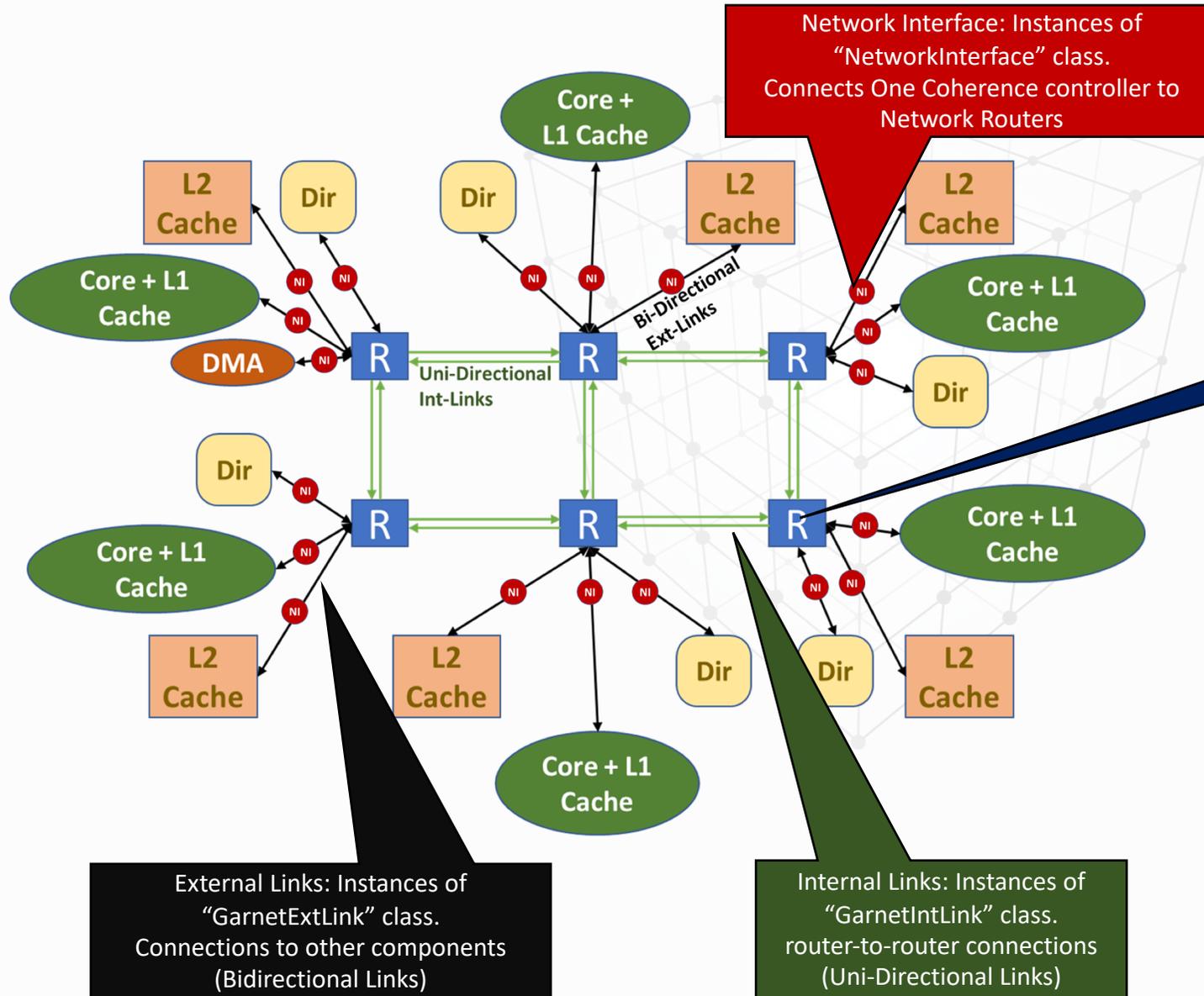


■ Through Silicon Vias Placement



- Max Performance by **full “layer-layer” connections**.
- For 8x8 Mesh:
 - 1.83e+18 possible **half “layer-layer” connections**.
 - 4.88e+14 possible **quarter “layer-layer” connections**.
- **Approach:** Search entire Design-space to choose most optimal TSV Placement configuration.
 - divide the NoC into smaller sub-sections to reduce the design space

Garnet Network Model



The "GarnetNetwork" class generates a network object which instantiates objects of other network components (network interface, routers, links).

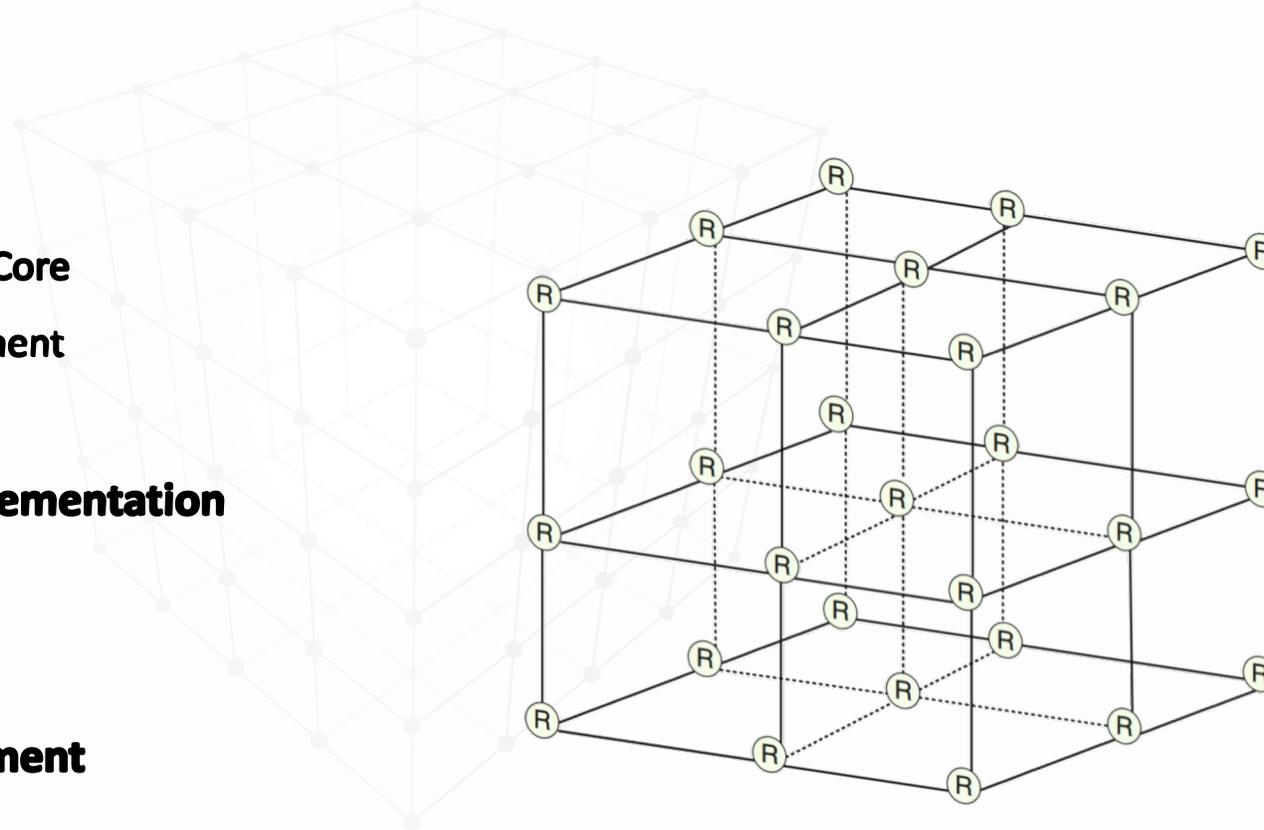
Garnet's "Topology" class enables heterogeneous topology designs

Topology class calls the methods to attach external and internal links for "Network Interface-routers" and "routers-routers" respectively according to user's NoC layout

Contents

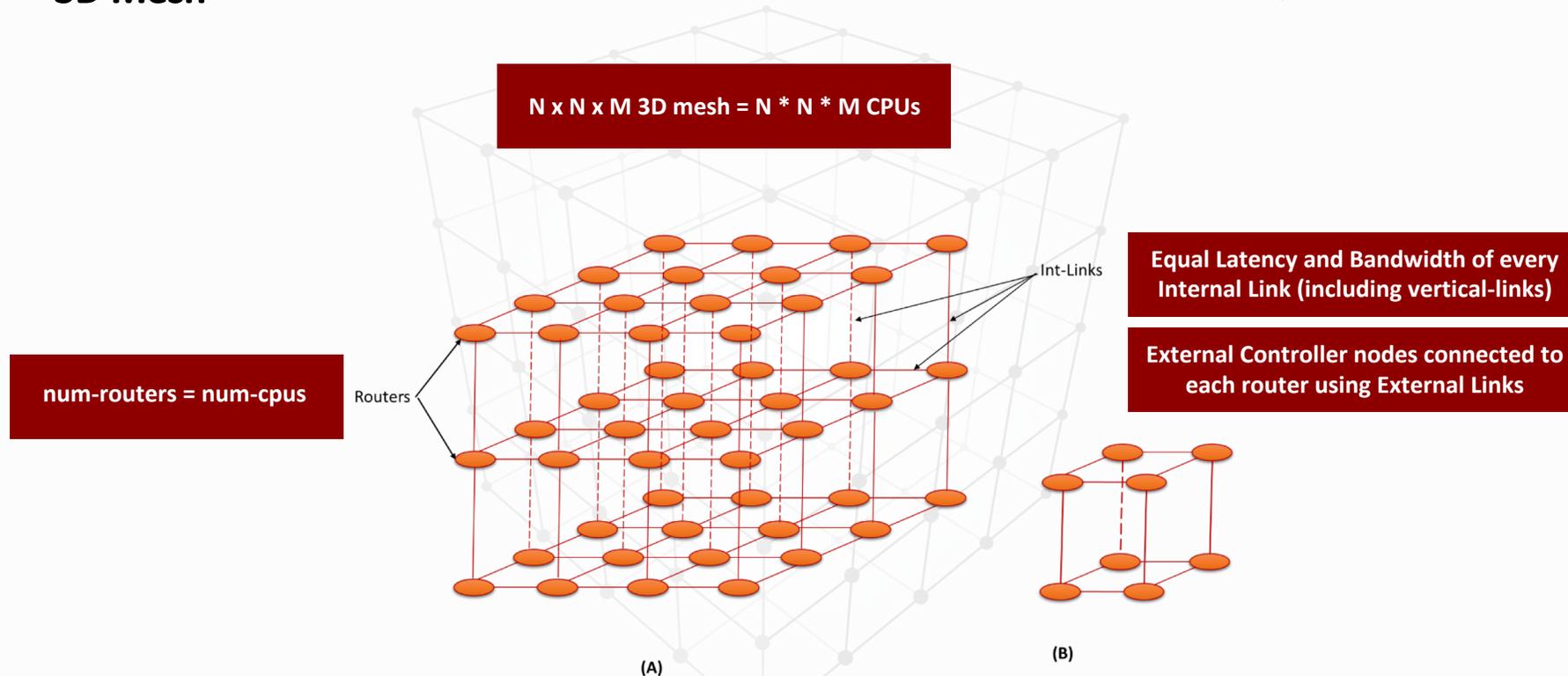


- Introduction
- Related Work
 - 3D Die-Stacking Techniques
 - 3D Architectures for Multi-Core
 - Through Silicon Vias Placement
- Garnet Network Model
- **Three-dimensional NoC Implementation**
 - 3D Mesh NoC
 - 3D-stacked Mesh NoC
 - Optimal TSV-Link Placement
 - Program Flow
- Simulation Results
- Conclusion and Future Work
- References



NoC Implementation

- 3D Mesh

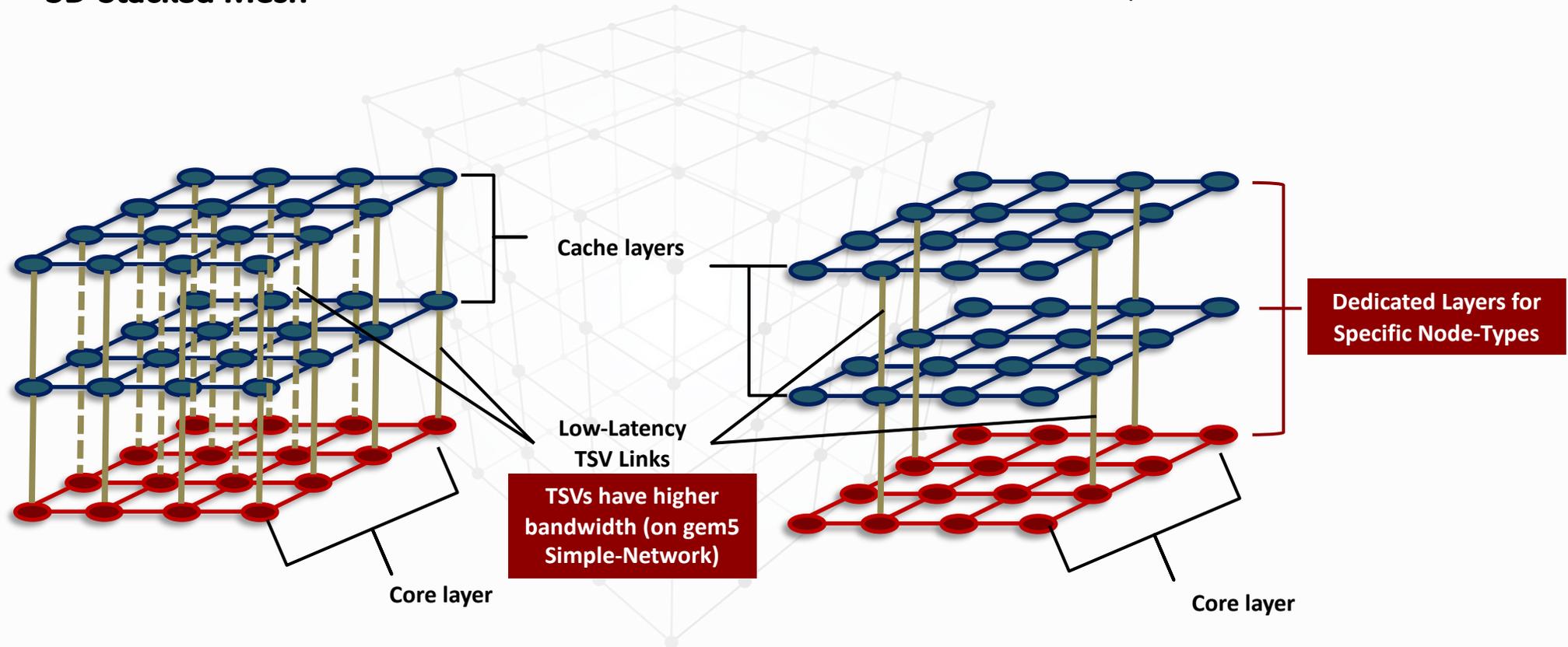


3D-Mesh NoC Designs. (A) 4x4x3 3D-Mesh (B) 2x2x2 3D-Mesh

Symmetric by design, but
Easy to design Asymmetric 3D-Mesh

NoC Implementation

▪ 3D-Stacked Mesh



An Abstract Layout of a 4x4x3 3D-stacked Mesh Network-on-Chip Interconnect. (A) All low-latency TSV Links are connected. (B) Optimal Number of low-latency TSV Links (four, in this case) are connected.

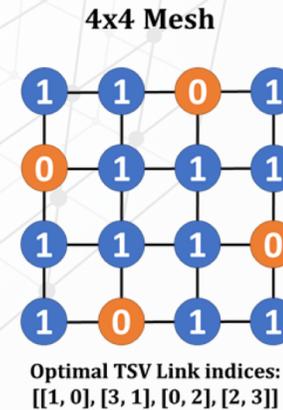
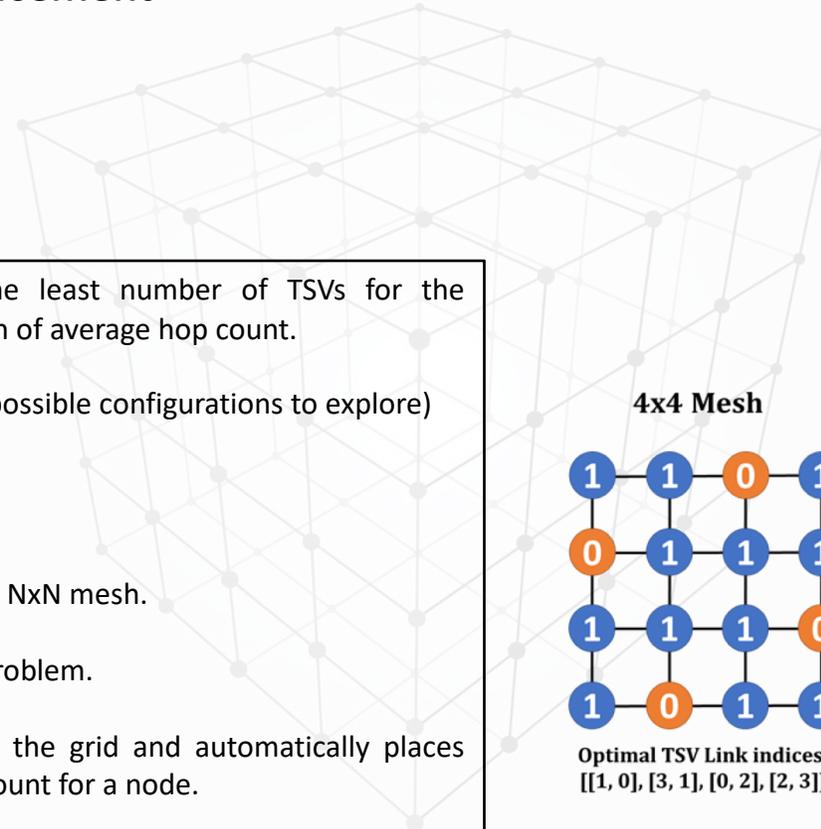
- Number of layers = 3; by design due to practical limitations (temperature & die-area).
- Calculates Optimal TSV-Links by calling sub-routine.

NoC Implementation

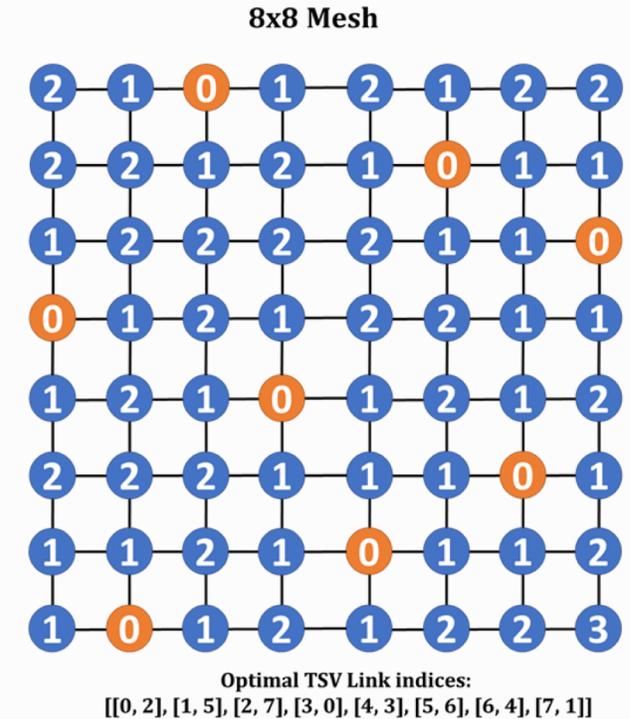
Optimal TSV Link Placement



- **Space Exploration Optimization problem:** The least number of TSVs for the maximum possible performance \Rightarrow minimization of average hop count.
- **Assumptions:** (dramatically shortens the no. of possible configurations to explore)
 - No two TSVs in same row.
 - No two TSVs in same column.
 - No two TSVs in same diagonal.
- Outputs a fixed configuration for value of 'N' in a NxN mesh.
- Branch and Bound approach akin to N-Queens Problem.
- This approach restricts the number of TSVs in the grid and automatically places them in the position of minimum possible hop count for a node.
- Flexible code written to accommodate other methods of Optimal TSV placement.



(A)



(B)

Packet Hop values from each Network-Node to a nearest TSV link in (A) 4x4 Mesh and (B) 8x8 Mesh.

NoC Implementation

■ Program Flow



```
function setTopology():
{
    Initialize network nodes list
    Set number of mesh-rows
    Initialize number of routers    # for Mesh-3D: num-routers = num-cpus
                                    # for Mesh-stacked: num-routers = l1 + l2 ctrls

    Initialize num-cols
    Initialize num-layers

    Define general link and router latencies

    sanity check for user input-parameters

    For num-routers, declare instances of Garnet Router class

    Setup external links    # each router is connected with external controllers
                            # (cache controllers, directory nodes, DMA controllers)

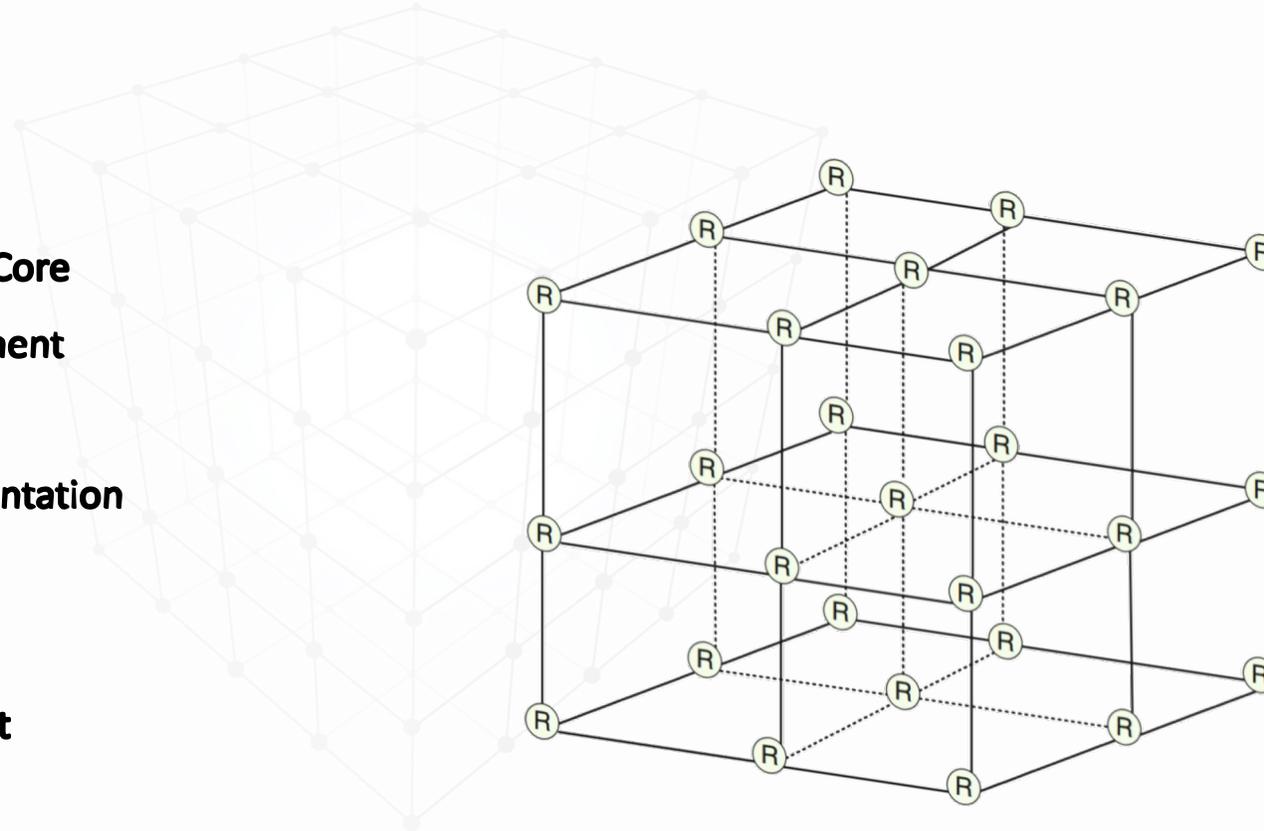
    # Set Network Layout
    Setup layer-wise Internal links
    Generate TSV link positions
    Setup Intra-layer links

    Send router and external/internal links to gem5
}
```

Contents



- Introduction
- Related Work
 - 3D Die-Stacking Techniques
 - 3D Architectures for Multi-Core
 - Through Silicon Vias Placement
- Garnet Network Model
- Three-dimensional NoC Implementation
 - 3D Mesh NoC
 - 3D-stacked Mesh NoC
 - Optimal TSV-Link Placement
 - Program Flow
- Simulation Results
- Conclusion and Future Work
- References





Evaluation Methodology

```
./build/x86_MOESI_CMP_directory/gem5.opt \  
--outdir=outputDirPath/ \  
-r ./configs/example/fs.py \  
--kernel= x86_64-vmlinux-2.6.22.9.smp \  
--disk=x86root-parsec.img \  
--ruby \  
--num-cpus=16 \  
--cpu-type=DerivO3CPU \  
--num-dir=16 \  
--num-l2-caches=32 \  
--l1d_size=32kB \  
--l1i_size=32kB \  
--l1d_assoc=4 \  
--l1i_assoc=4 \  
--l2_size=512kB \  
--l2_assoc=8 \  
--script=blackscholes.rcs \  
--network=garnet2.0 \  
--mesh-rows=4 \  
--topology=Mesh_stacked_OptimalTSVs
```

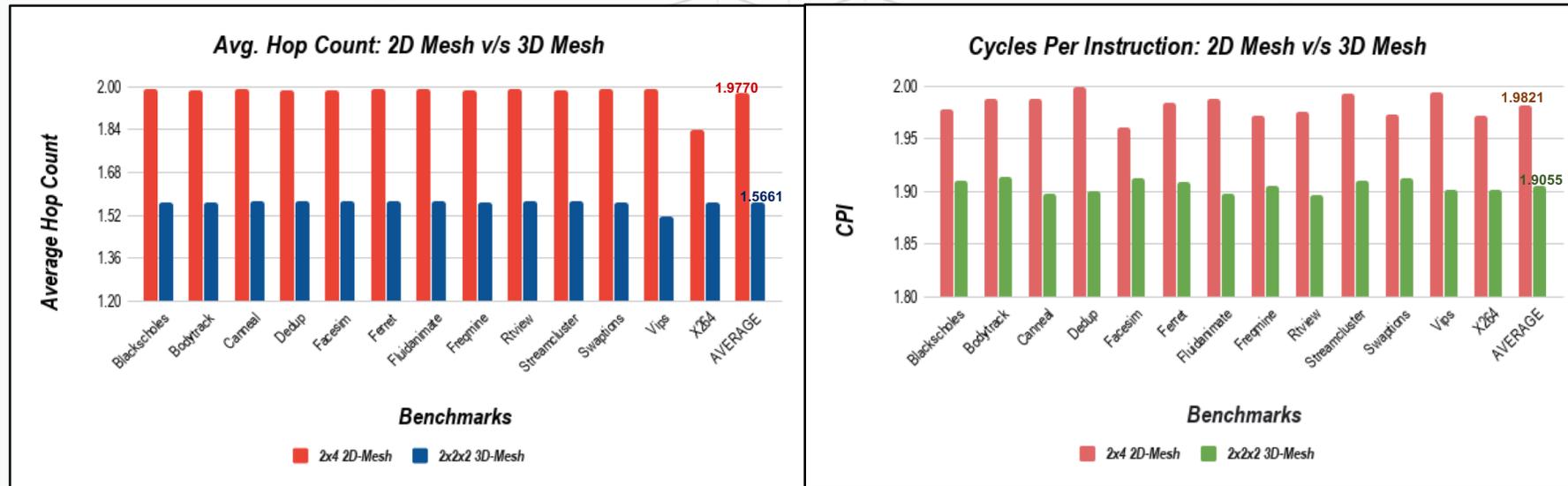
Parameters to set the Number of Network-Nodes to be generated

Sets the Default Network-on-Chip framework to be used by gem5 simulator
Options: "garnet2.0" & "simple"

- Set the Network Topology to be used by the gem5 simulation.
 - Can be 2D or 3D.
 - 3D Topology options: "Mesh_3D", "Mesh_stacked" & "Mesh_stacked_OptimalTSVs"

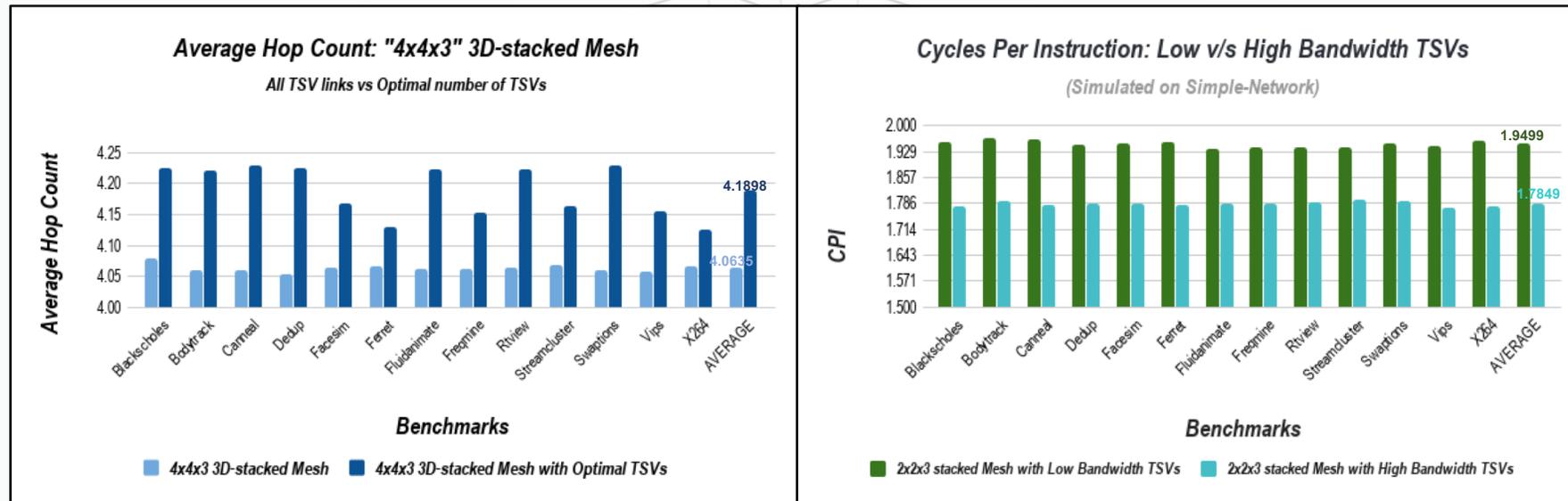
- Run Full-system simulations by changing the network-defining parameters.
- Collect "network.average_hops", "sim_ticks" and "sim_insts" from "stats.txt".
 - Average CPI = (sim_ticks/sim_insts)
 - Average Hop Count = network.average_hops

Simulation Results



- Evaluated on PARSEC Benchmark suite.
- Total number of internal links (in one direction) in the model = $N_1N_2(N_3 - 1) + N_1N_3(N_2 - 1) + N_2N_3(N_1 - 1)$
- 2x2x2 3D Mesh v/s 2x4 2D Mesh:
 - About 21% on average reduction in Average hop-count.
 - Reduced values of average CPI for all benchmarks. Signifies better performance.
- Values of Hop-count follow the metric $hops_{NoC} = \frac{n_1n_2n_3(n_1 + n_2 + n_3) - n_3(n_1 + n_2) - n_1n_2}{3(n_1n_2n_3 - 1)}$, with 5-8% smaller values.

Simulation Results (cont.)



- Around 3-4% increment in the average hop-count values compared to the naïve 3D-stacked Mesh.
- Very slight increment in the hop count reflects the efficiency of the approach.
- High-Bandwidth TSV model: thrice the bandwidth than the rest of the internal links.
- 8-9% average reduction in Average CPI values. Signifies Greater performance of High-Bandwidth model.
- Simulated on gem5 Simulator's "SimpleNetwork" framework.

Conclusion & Future Work



In Summary

- Demonstrated need for Network-on-Chips and discussed state-of-the-art research on 3D NoC Designs.
- Implemented three variations of 3D-mesh NoC designs: **3D mesh**, **3D-stacked Mesh** and discussed the subtle differences in their structure.
- *Discussed a fast approach to extract Optimal TSV indices for the implementation of **3D-stacked Mesh with Optimal TSV links**.*
- Discussed and analysed simulation results to verify correct working of the model.

Future Work

- More decisive methods to validate 3D models.
- Simulations for large NoC sizes.
- Integration with Power models.
- More Sophisticated approaches for Optimal TSV placement problem.
- Implementation of other 3D NoC designs (other than Mesh designs).

References



1. N. E. Jerger, T. Krishna and L.-S. Peh, "**On-Chip Networks**," in *On-Chip Networks*, Morgan & Claypool, 2009.
2. G. H. Loh, Y. Xie and B. Black, "**Processor Design in 3D Die-Stacking Technologies**," *IEEE Micro*, vol. 27, no. 3, pp. 31-48, May-June 2007.
3. F. Denneman, "**frankdenneman.nl**," 7 July 2016. [Online]. Available: <https://frankdenneman.nl/2016/07/07/numa-deep-dive-part-1-uma-numa/>. [Accessed Friday May 2020].
4. B. S. Feero and P. P. Pande, "**Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation**," *IEEE Transactions on Computers*, vol. 58, no. 1, pp. 32-45, Jan 2009.
5. J. Lowe-Power, "**gem5 Documentation**," gem5, [Online]. Available: <http://www.gem5.org/documentation/>. [Accessed Tuesday May 2020].
6. T. Krishna, "**A Detailed On-Chip Network Model Inside a Full-System Simulator**," Monday, September 2017. [Online]. Available: <https://pdfs.semanticscholar.org/c1e9/0beac857ce1af1a531b6538804e71efdef05.pdf>. [Accessed Tuesday May 2020].
7. T. C. Xu, P. Liljeberg and H. Tenhunen, "**A study of Through Silicon Via impact to 3D Network-on-Chip design**," in 2010 International Conference on Electronics and Information Engineering, Kyoto, 2010.

...



THANK YOU