System design exploration with fully customizable NoC

Tushar Garg, Ranjan Mahajan
Synopsys India Pvt. Ltd.
Noida, UP, India
{Tushar.Garg2|Ranjan.Mahajan}@synopsys.com

Abstract- In this paper, we present a novel solution for a fully customizable interconnect which enables system exploration and optimization with its various options and customizable interfaces like arbiter, router and decoder etc. As a key component in system, it's important to analyze power and performance of Interconnect in full use-case based scenario and in turn analyzing the whole system. Network on Chip architecture used in this paper is implemented as Transaction Level Model (TLM) in SystemC. The Transaction Level Model is designed with flexibility at its core, allowing it to adapt to various configurations and requirements. It can mimic a wide range of Network on Chips (NoC), providing a versatile solution for diverse applications. With its inbuilt topologies like mesh and ring together with customizable topologies, it can be used to create any NoC architecture with its network of routers. It also provides customizable interfaces to mimic specific interconnect behavior which includes user defined arbiter, user defined delay block, user defined decoder, user defined router and user defined replicator for transaction replication. It also provides detailed analysis views for root cause analysis and optimization. Proposed solution can be used for System on Chip (SOC) architecture exploration and optimization with these customized options in the model.

Keywords—SystemC; Network on Chip (NoC); Transaction Level Model (TLM); System on Chip (SOC); Simulation; Integrated Circuit (IC)

I. INTRODUCTION

A NoC[5] consists of a structure of routers and links implementing a packet-switched communication fabric. Now a days, there are so many different NoCs present in market that it is difficult to decide on which NoC to be used on a particular SoC. A NoC can be described by its topology[3] which can be mesh, ring, torus or any custom topology and the strategies employed for routing, switching, flow control, arbitration and buffering. Various topologies are shown in Figure 1.a, 1.b, 1.c.

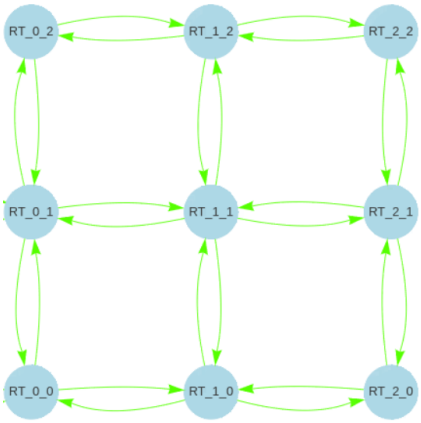


Figure 1.a: Mesh

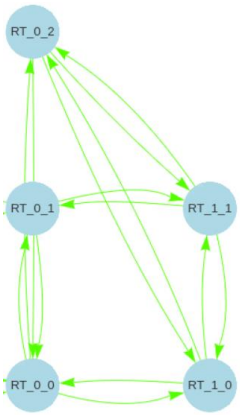


Figure 1.b: Custom

Figure 1.c: Crossbar

In modern systems, optimizing Network-on-Chip (NoC) design is crucial for enhancing Chip-to-Chip (C2C) communication, given the bandwidth limitations of ICs. NoCs are vital for the efficient operation of multi-core processors, SoCs, and other complex ICs.

Current NoC TLM models face several challenges which include support for few topologies which may not suit all applications. There are Transaction Level Models available in which different topologies were explored but each model offers a single topology like ring, mesh, torus etc. In these models, there is limited capability to customize the architecture of the NoC to effectively explore system design and exploration. Also, there are scalability issues with performance bottlenecks as the number of nodes increases. There is limited ability to customize routing algorithms, arbitration mechanisms, and other parameters. Also, there is no support for high-speed C2C communication, crucial for multi-chip systems. Addressing these challenges is vital for applications like HPC, data centers, and AI accelerators, where NoC performance significantly impacts overall system efficiency. In our solution there is no restriction as any NoC architecture can be created with its own characteristics.

Our model offers several key features:

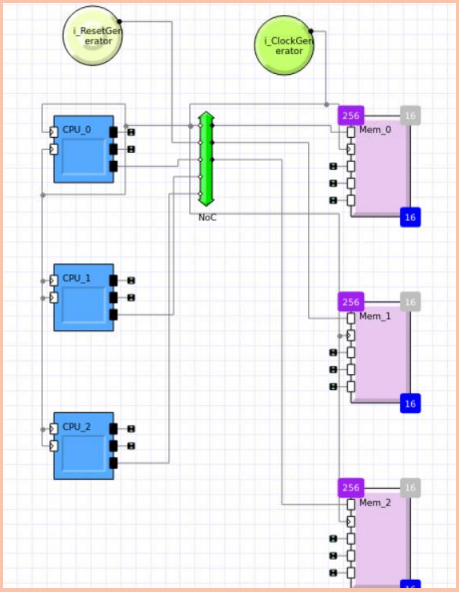
1. Flexibility in Configuration: Designed with adaptability at its core, the model can meet various configuration requirements.
2. Versatility in Mimicking NoCs: Capable of emulating a wide range of NoC architectures[4], making it suitable for diverse applications.
3. Support for Multiple Topologies: Accommodates various network topologies, including mesh, torus, ring, and star topologies.
4. Unique Interface Offering: Currently, no other model provides an interface that enables communication between chips with different protocols, such as PCIe, Ethernet, and proprietary protocols.

II. Proposed Solution

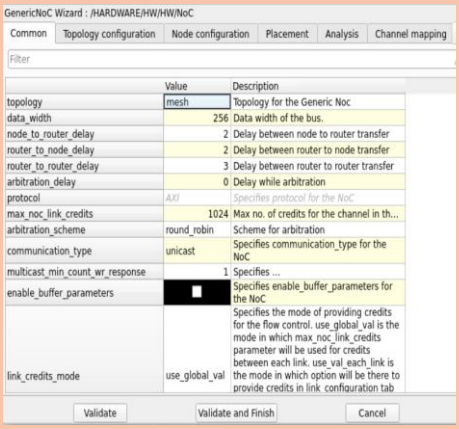
In this solution, a user will be able to generate a NoC TLM model which can support multiple topologies and configurable parameters as shown in Figure 2. User will start creating a system with multiple cores and memories(nodes) and the bus placeholder. For the generation of the bus, user will select topology like ring, mesh or custom. After selecting the topology, user will provide various delay options related to routing and arbitration. User will also select the number of routers and its connections based on topology. After providing network of routers, user will provide routing mechanisms like shortest, clockwise, x-axis first based on topology. User can also provide their custom routing algo. Arbitration scheme must be provided for arbitration between different nodes. After selection of these options, user will decide the placement of nodes on particular router for system exploration and optimization. User will validate these options and then finish the creation of bus. Then user will run simulations and analyse performance using various analysis views, including latency, throughput, power consumption, and thermal analysis.

Let's take a look at creation of NoC in below flow chart:

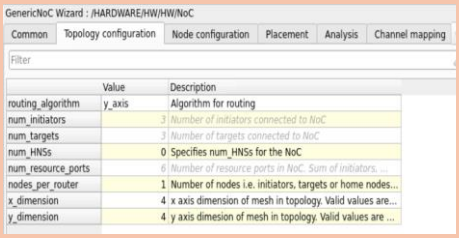
Create a System with cores, memory subsystem and bus placeholder



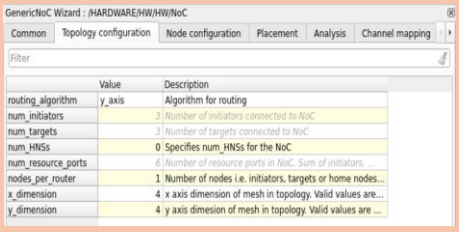
Configure NoC with topology, delays, arbitration algorithm and other options related to flow control



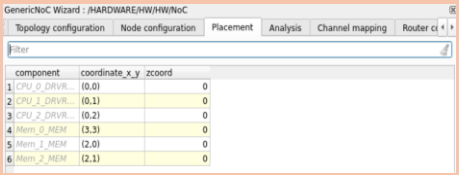
Select routing algorithm, number of routers based on topology using x and y dimensions



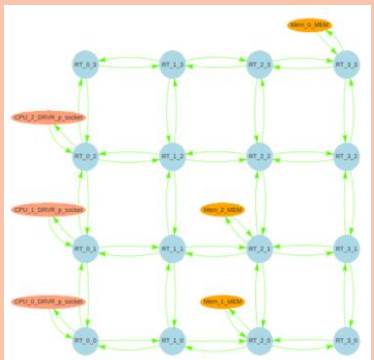
Configure options related to nodes connected to the NoC



Decide placement of nodes on routers



Validate and Finish



During development, accommodating different customer requirements in terms of topology, routing algorithms, and multicasting revealed surprising traffic flow patterns. These insights highlighted the need for enhanced analysis capabilities within the NoC.

III. RESULT

With the proposed solution, analysis of the system was done for exploration and optimization. We made a sample system with Coherent NoC with multiple cores, caches and memories. Various delays were provided in NoC configuration to see the throughput of NoC.

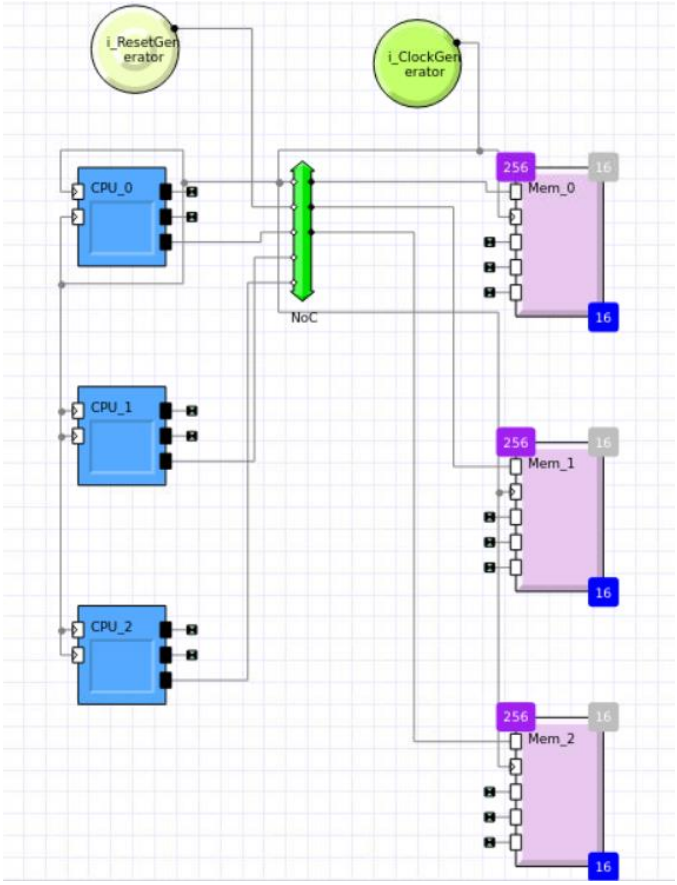


Figure 3: System with NoC

After configuring the NoC with mesh topology, node to router delays, round robin arbitration scheme and y-axis routing algorithm with flow control, placement of nodes, we can analyze[2] the NoC performance with various analysis views.
Below we can see the packet latency on each link for write data packets. We can also see the latency for each channel and get an understanding of NoC design and make better design decisions.

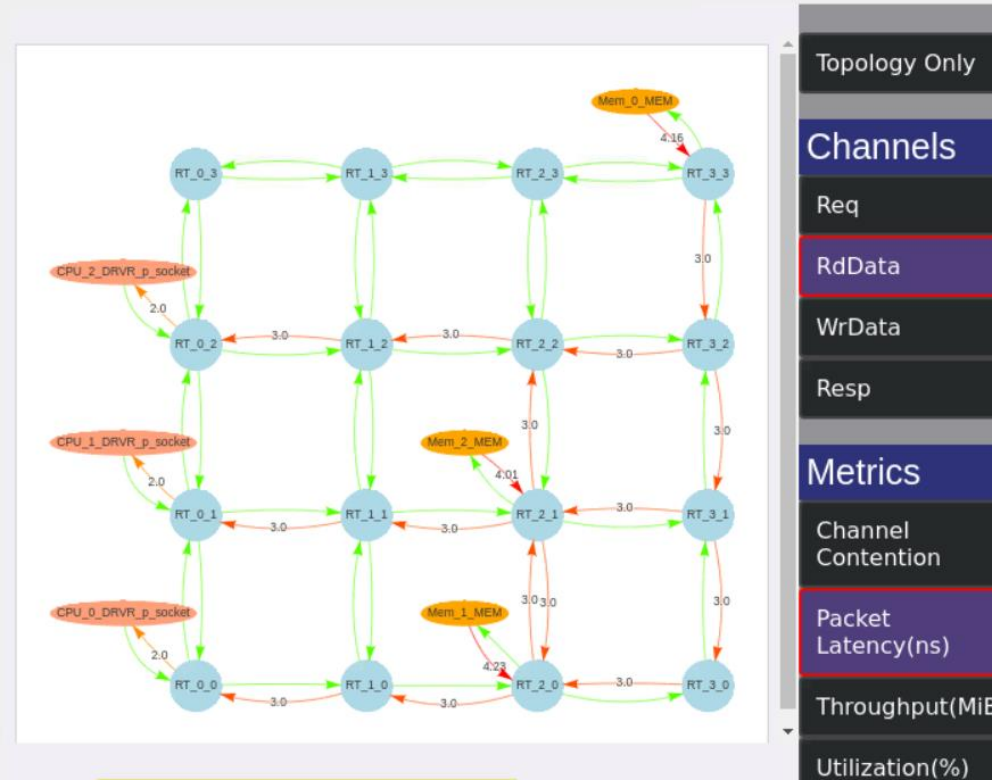


Figure 4: NoC Topology with nodes on routers (before)

If we investigate the topology, we will see that Requester Node (RN) is too far from the Subordinate Nodes (SN). Topology is also too big. With the router-to-router delay provided, there will be too much delay when the transaction reaches from RN to SN. Similarly, from SN to RN, there will be too much delay.
To confirm the above findings, we can check the analysis views available for NoC.

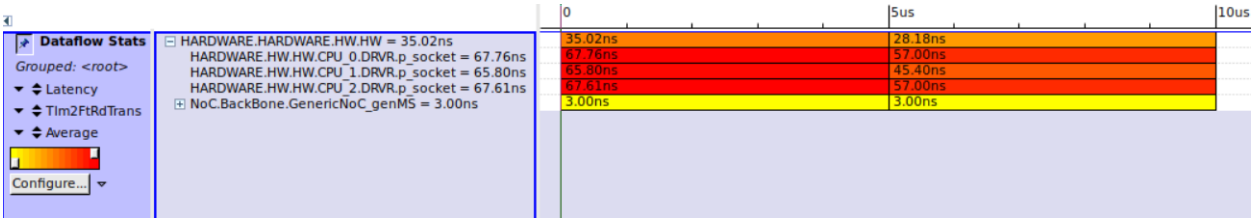


Figure 5: Total Latency(before)

We will see that the latency is too much, now analysing link latency for more understanding:

Figure 6: Link Latency(before)

In this way we can check the link latency of each link and find out where the bottleneck is. Now changing the placement of the nodes on the routers for reducing the overall latency like below:

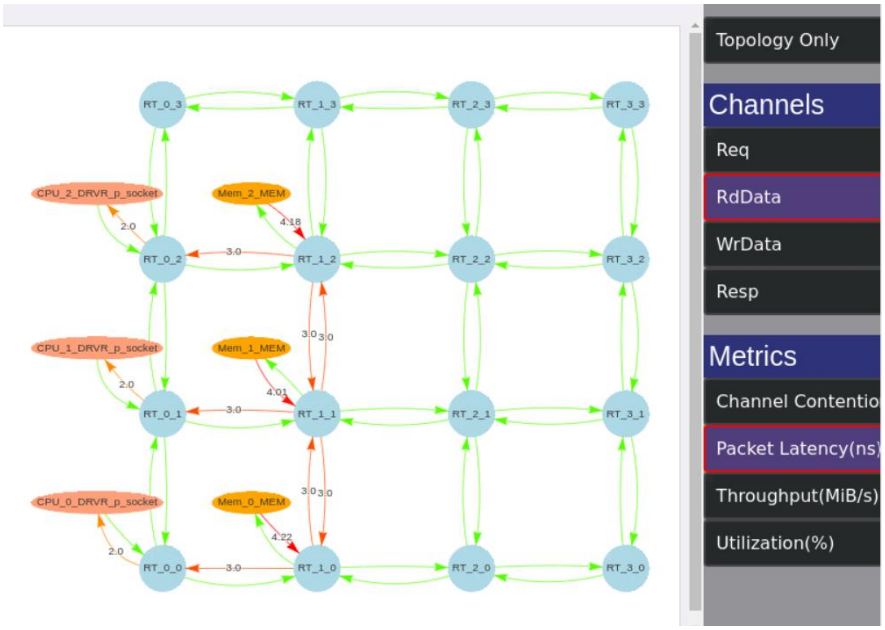


Figure 7: NoC Topology with nodes on routers (after)

We can see in the topology that now the path of the packets will be different, and time taken for 1 packet from node to another node will be less[1] resulting in reduction of latency which we can see in below analysis.

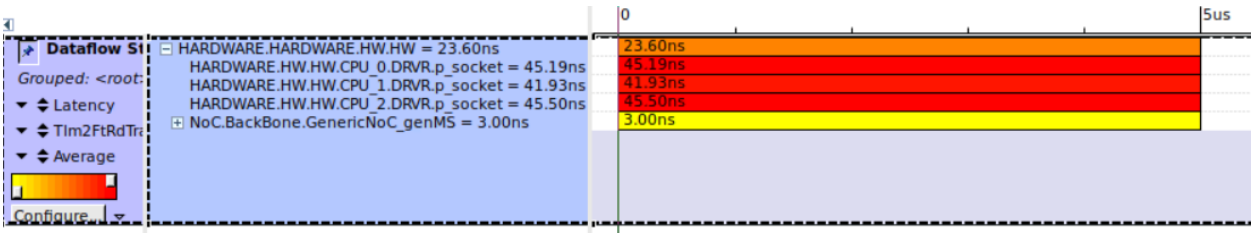


Figure 8: Total Latency(after)

IV. CONCLUSION

Generic model of NoC to provide capabilities to mimic typical NoC architectures which support both coherent and non-coherent interfaces with selectable topology options to explore design options. Multiple configuration knobs are available to explore design space for NoC architectures by having customization options to implement specific NoC logic. It also offers out of box solution to tune typical NoC parameters There are detailed analysis views available for system optimization opportunity identification

REFERENCES

[1] Van den Brand, J. W. and Ciordas, C. and Goossens, K. and Basten, T., 'Congestion-Controlled Best-Effort Communication for Networks-on-Chip.', Design, Automation Test in Europe Conference Exhibition, 2007, pp. 1 – 6.
[2] Atienza, D., Angiolini, F., Murali, S., Pullini, A., Benini, L., De Micheli, G., 'Network-on-Chip design and synthesis outlook.', The VLSI journal, Elsevier, 2008, 41, pp. 340 – 359.
[3] Alakesh Kalita & Kaushik Ray & Abhijit Biswas, 'A Topology for Network-on-Chip'.
[4] Duato, J. and Yalamanchili, S., Ni, L., 'Interconnection Networks.', Morgan-Kayfmann, 2002.
[5] Songwen Pei, Xinyi Wang, "Shuttle: A Novel Topology Model for 3D NoC" Journal of Convergence Information Technology(JCIT) Volume 7, Number 19, Oct 2012.