

Enhancing Cache Coherent Interconnects to support Space Systems

AHSEN EJAZ, Chalmers University of Technology, Sweden
BHAVISHYA GOEL, Chalmers University of Technology, Sweden
MADHAVAN MANIVANNAN, Chalmers University of Technology, Sweden
MEHRZAD NEJAT, Chalmers University of Technology, Sweden
IOANNIS SOURDIS, Chalmers University of Technology, Sweden
PER STENSTRÖM, Chalmers University of Technology, Sweden

Abstract: As multi-core designs become prevalent in space systems, the role played by cache coherent interconnects is becoming increasingly crucial for scaling to larger systems. This paper outlines techniques to enhance a cache coherent interconnection network based on AMBA CHI specification, developed originally for high-performance chips, to support the needs of a space System-on-Chip. The objective is to add reliability and performance isolation capabilities through the proposed techniques.

1 Introduction

The growth of RISC-V highlights the demand for an open standard for Instruction Set Architectures (ISA) that provides customization, flexibility, and reduces reliance on proprietary ISAs. Its adoption has expanded from its initial focus on microcontrollers in embedded systems to now encompass high-performance microprocessors used in servers and accelerator-based systems [20, 21, 24, 25]. As the industry transitions to high performance RISC-V based microprocessors, design offerings with a large number of processor cores are becoming more prevalent. These designs typically implement a shared memory model with hardware-managed cache coherence, a feature that simplifies the task for software since it doesn't have to explicitly manage on-chip communication. The complexity introduced by the addition of more cores coupled with the increasing number of other IP blocks in designs, places an increasing emphasis on the scalable cache coherent interconnect for multi-core processors.

The evolution of standardized interconnect protocols has closely followed the development of microprocessor architecture, transitioning from simple bus-based systems gradually to complex Networks-on-Chip (NoCs). Traditionally, in single-core systems and simple multi-core systems with a few cores, bus-based interconnects featuring separate buses for different purposes, such as the Processor Bus and the Peripheral Bus, were mostly sufficient to meet the communication requirements. As microprocessor designs with an increasing number of cores and IP blocks become more prevalent, standardized interconnect protocols such as AMBA5 Coherent Hub Interconnect (CHI) [22] and SiFive's TileLink [23] have emerged, utilizing complex interconnection networks and communication protocols to support parallel high-speed data transfers and enable efficient inter-core communication. This shift has accelerated the utilization of NoC-based cache coherent interconnect due to its ability to offer better scalability. Furthermore, these standards enable fabless design houses to create reusable components that can be integrated across various designs or to even be licensed for external use, thereby fostering adoption, accelerating innovation, and reducing design effort.

Historically, there has been a performance gap between space-based and terrestrial systems, primarily due to conservative design and manufacturing choices driven by harsh conditions of space, stringent reliability requirements, and rigorous testing standards. However, the growing demand for high-speed on-board processing in space, coupled with an increase in private sector involvement, is expected to increase performance requirements and reduce this gap. The demand for increased computational performance in space systems, like in terrestrial systems, is being met by integrating multiple cores into a single system-on-chip (SoC). The adoption of open standards like RISC-V coupled with increasing system integration helps reduce system costs by consolidating functions, which previously required multiple chips, onto a single SoC. In addition to adopting multi-core designs, with the number of cores expected to rise, such SoCs are also utilizing advanced processing units, like AI accelerators and FPGAs, for enhanced processing efficiency. In a nutshell, the complexity of space SoC design is starting to mirror that of terrestrial systems [12] and this consequently underscores the importance of scalable cache coherent interconnect in space SoC designs. While the performance requirements from a cache coherent interconnect used in space SoCs may be relaxed in comparison to terrestrial systems, there is stricter requirements in terms of reliability and performance isolation.

53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104

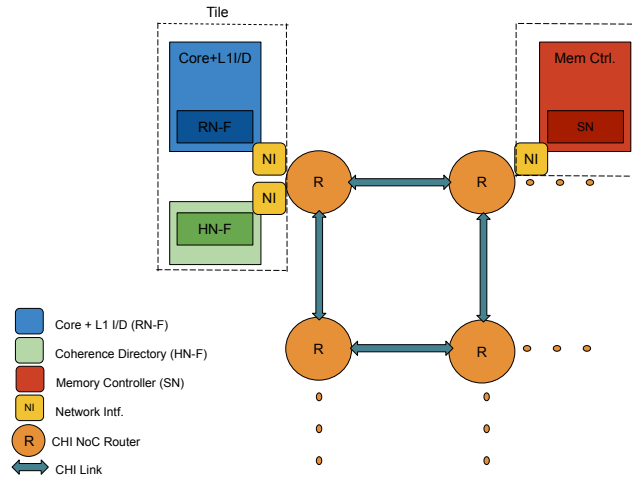


Fig. 1. System Architecture Overview showing key components based on AMBA CHI specification

The team at Chalmers has been working on the design and implementation of a cache coherent interconnect based on the AMBA CHI standard in the context of multiple research projects like EPI [13], EUPilot [17], and eProcessor [16], under the EuroHPC umbrella. The preliminary version of this design has been taped-out and bring-up efforts are currently in progress. While the current design targets high performance use cases, we are exploring techniques to make it amenable for operation in environments that pose reliability and performance isolation requirements. In this paper, we outline the architectural and design techniques we are considering to achieve this objective.

The rest of the paper is organized as follows: Section 2 first provides an overview of the system architecture wherein the key components of the cache coherent interconnect are introduced. Section 3 outlines their functions and describes the techniques that we are evaluating for use in the different components before we conclude in Section 4.

2 System Architecture Overview

An overview of the multi-core system architecture is shown in Figure 1. For the sake of clarity we focus only on the key components that are part of the AMBA CHI specification. The design comprises multiple tiles connected together typically using a mesh interconnection network. Each tile comprises a core with single/multiple levels of private caches, referred to as a fully-coherent requester node (RN-F) that acts as a source of requests in the CHI network, as well as a slice of the coherence directory, referred to as a fully-coherent HomeNode (HN-F) that acts as a point of coherence in the CHI network. Both the RN-F and the HN-F, that are part of a tile, have interfaces to connect to the NoC router. Tiles may optionally comprise a main memory controller that serves read and write requests directed to main memory, also referred to as a Subordinate Node (SN) in CHI terminology. The CHI NoC uses separate channels for request, response, data and snoop traffic classes. RN-F, HN-F and SN comprise Tx/Rx interfaces to connect to the necessary channels in the NoC. The interface and the communication protocol between the different nodes are based on the AMBA CHI specification. We collectively refer to HN-F and the CHI NoC as the cache coherent interconnect.

3 Design Overview

In this section, we first provide an overview of the functionality of key components in the cache coherent interconnect. We then describe strategies that we are evaluating to enhance the reliability for each of these components. Note that the focus of this discussion is primarily on mitigating the impact of transient faults because they represent the most likely type of fault in space electronics triggered because of interaction with high-energy particles in space environment.

3.1 Network-on-Chip

Design Overview: An on-chip interconnection network is composed of three main components [6]: i) network interfaces (NI), which act as bridges between the connected IP and the network and translate messages to network packets and vice versa, ii) network routers to route and control the flow of packets in the network and iii) link wires to

transfer packets from one router to the next. Routers and links are connected to create a network topology, typically a 2D mesh topology (as shown in Figure 1) because its regular structure simplifies layout, placement and routing during physical implementation. Efficient design of NoC routers is vital to ensure scalable and high-performance communication in multicore chips.

The router datapath, which carries packets across a router, is composed of two stages: i) Switch Traversal (ST) and ii) Link Traversal (LT), separated by input Virtual Channel (VC) buffers and an output register. The router control logic, which manages the flow of packets over the datapath, consists of Next-Route Computation (NRC), Virtual Channel Selection (VS) and Switch Allocation (SA) blocks. Most modern NoC routers allow flits to hop from one router to the next in 3 clock cycles, one cycle for NRC, VS and SA working in parallel and one clock cycle for each ST and LT stages [2, 7].

Enhancements for the Space domain: NoC performance is crucial to the overall performance of applications running on multicore systems. Performance of an application may be sensitive to the network latency (e.g., real-time applications) [19] or to the maximum network throughput (e.g., concurrent scale-out applications) [14, 19]. NoC latency can be reduced with low overhead, from 3 to 2 clock cycles per hop, by bypassing control stages (VS and SA) in a router, when there is no contention and downstream buffer resources are available [2, 9]. Further reduction to a single clock cycle per hop is possible by bypassing the ST stage as well [10]. NoC throughput is improved by using higher-radix topologies [11, 19], balancing the logic delay of all pipeline stages and ensuring that the datapath, rather than the control logic, defines the timing critical path of the NoC router. This allows the full clock period to be utilized to move the flit forward over the datapath, maximizing performance [8].

Providing latency and throughput guarantees for certain traffic flows in the NoC is also important for many real-time space applications. This can be achieved by configuring the NoC to permanently allocate specific VCs and ST timeslots to traffic flows requiring performance guarantees, while allowing the remaining network resources to handle other traffic in a best-effort manner [18]. These mechanisms can also be used to isolate specific traffic flows and mitigate timing side-channel attacks by limiting their access to shared network resources [5].

In multicore systems, NoCs typically occupy around 10% of chip area, with the datapath accounting for more than 80% of this, depending on the number and size of VCs [3, 4, 7]. Given that radiation in space can induce transient faults, ensuring fault tolerance in the datapath is crucial. This can be achieved by adding ECC bits to flits and packets, allowing error detection and correction. For packets with uncorrectable errors, packet retransmission can be enabled to ensure reliability. Although packet retransmission can have a significant performance overhead, it can be minimized by enabling it only for traffic classes that are sensitive to errors.

3.2 Coherence Directory

Design Overview: The coherence directory is responsible for tracking the location and state of data across the various caches. The operation of the coherence directory is akin to a state machine, in that it utilizes the presence state bits (used to track which private caches have a copy of the data), the coherence state bits (indicating the state of the line depending on the underlying protocol used) and the current request state bits to determine the next state for the cache line and the request/response/snoop/data packets that need to be generated. Our implementation features a multi-stage pipelined design that is capable of handling one request and response in each cycle and supports handling coherence state for multiple outstanding in-flight requests concurrently. In addition, the design supports coherence optimizations that can reduce coherence traffic and request processing latency [15].

Enhancements for the Space domain: We envision a combination of different techniques in order to ensure reliable operation of the coherence directory. The first is to use redundancy in cases where the overheads involved is considerably small. For instance the combinatorial logic in the directory state machine could be replicated with little overhead. Likewise, the coherence state bits typically require two to three bits per cache line and can be replicated with relatively low overhead. An alternative option is to group state information across different cache lines within a set together and then using error correction codes (ECC). As for the presence bits adopting ECC is the most favorable approach since using redundancy for this can lead to prohibitive overheads. Another option is to consider the use of radiation hardened flip flops and memory for structures which retain state.

The discussion until now assumes that the network is completely reliable and that errors can occur only when incoming packets are processed at the directory controller. However, in reality, the NoC, using the mechanisms described in the previous section, cannot guarantee error freedom for packet communication over the network layer.

This consequently requires error detection and error handling mechanisms at the protocol level as well. Typical strategies to achieve this include using timeout intervals, retransmitting packets and incorporating extra states in the state machine to handle redundant messages [1].

4 Conclusion

The transition to RISC-V-based multi-core processors for space necessitates scalable cache coherent interconnect design that can meet reliability and performance isolation requirements. Our work on adapting the AMBA CHI-based cache coherent interconnect developed for high performance use case to space domain represents a step in this direction. By implementing the architectural and design techniques outlined in this paper, we aim to create a cache coherent interconnect solution that satisfies the needs of the space industry.

Acknowledgments

This work has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No. 956702 (eProcessor), under Framework Partnership Agreement No. 800928 and Specific Grant Agreement No. 101036168 (EPI SGA2), and under grant agreement No. 101034126 (The European PILOT). The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Sweden, Greece, Italy, France, Germany. Additionally, this work has also received funding from the Swedish Foundation for Strategic Research (SSF), the Swedish Research Council (VR), Vinnova and the Swedish National Space Agency (SNSA).

References

- [1] Konstantinos Aisopos and Li-Shiuan Peh. 2011. A systematic methodology to develop resilient cache coherence protocols. In *44rd Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 2011, Porto Alegre, Brazil, December 3-7, 2011*, Carlo Galuzzi, Luigi Carro, Andreas Moshovos, and Milos Prvulovic (Eds.). ACM, 47–58. doi:10.1145/2155620.2155627
- [2] Chrysostomos Nicopoulos Anastasios Psarras, Ioannis Seitanidis and Giorgos Dimitrakopoulos. 2016. ShortPath: A Network-on-Chip Router with Fine-Grained Pipeline Bypassing. *IEEE Trans. on Computers* 65, 10 (2016), 3136–3147.
- [3] Brent Bohnenstiehl, Aaron Stillmaker, Jon Pimentel, Timothy Andreas, Bin Liu, Anh Tran, Emmanuel Adeagbo, and Bevan Baas. 2017. KiloCore: A Fine-Grained 1,000-Processor Array for Task-Parallel Applications. *IEEE Micro* 37, 2 (2017), 63–69. doi:10.1109/MM.2017.34
- [4] B. Bohnenstiehl, A. Stillmaker, J. J. Pimentel, T. Andreas, B. Liu, A. T. Tran, E. Adeagbo, and B. M. Baas. 2017. KiloCore: A 32-nm 1000-Processor Computational Array. *IEEE Journal of Solid-State Circuits* 52, 4 (April 2017), 891–902. doi:10.1109/JSSC.2016.2638459
- [5] Subodha Charles and Prabhat Mishra. 2021. A Survey of Network-on-Chip Security Attacks and Countermeasures. *ACM Comput. Surv.* 54, 5, Article 101 (May 2021), 36 pages. doi:10.1145/3450964
- [6] W.J. Dally and B. Towles. 2004. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers.
- [7] Bhavya K. Daya, Chia-Hsin Owen Chen, Suvinay Subramanian, Woo-Cheol Kwon, Sunghyun Park, Tushar Krishna, Jim Holt, Anantha P. Chandrakasan, and Li-Shiuan Peh. 2014. SCORPIO: A 36-core Research Chip Demonstrating Snoopy Coherence on a Scalable Mesh NoC with In-network Ordering. In *Int. Symp. on Computer Architecture (ISCA '14)*. 25–36.
- [8] A. Ejaz, V. Papaefstathiou, and I. Sourdis. 2018. DDRNoC: Dual Data-Rate Network-on-Chip. *ACM Trans. Archit. Code Optim.* 15, 2 (2018).
- [9] Ahsen Ejaz, Vassilis Papaefstathiou, and Ioannis Sourdis. 2021. HighwayNoC: Approaching Ideal NoC Performance With Dual Data Rate Routers. *IEEE/ACM Transactions on Networking* 29, 1 (2021). doi:10.1109/TNET.2020.3034581
- [10] Ahsen Ejaz and Ioannis Sourdis. 2022. FastTrackNoC: A NoC with FastTrack Router Datapaths. In *IEEE International Symposium on High-Performance Computer Architecture, HPCA 2022, Seoul, South Korea, April 2-6, 2022*. IEEE, 971–985. doi:10.1109/HPCA53966.2022.00075
- [11] B. Grot, J. Hestness, S. W. Keckler, and O. Mutlu. 2009. Express Cube Topologies for on-Chip Interconnects. In *HPCA*. 163–174.
- [12] High Performance Spaceflight Computer (HPSC). 2024. NASA. <https://www.nasa.gov/wp-content/uploads/2024/07/hpsc-white-paper-tmg-26jun2024-final.pdf>
- [13] European Processor Initiative. 2019. EPI. <https://www.european-processor-initiative.eu/>
- [14] P. Lotfi-Kamran, B. Grot, and B. Falsafi. 2012. NOC-Out: Microarchitecting a Scale-Out Processor. In *2012 45th Annual IEEE/ACM Int. Symp. on Microarchitecture*. 177–187.
- [15] Madhavan Manivannan and Per Stenström. 2014. Runtime-Guided Cache Coherence Optimizations in Multi-core Architectures. In *2014 IEEE 28th International Parallel and Distributed Processing Symposium, Phoenix, AZ, USA, May 19-23, 2014*. IEEE Computer Society, 625–636. doi:10.1109/IPDPS.2014.71
- [16] The European Pilot. 2020. Eprocessor. <https://eprocessor.eu/e-processor-in-a-nutshell/>
- [17] The European Pilot. 2022. EUPilot. <https://eupilot.eu/>
- [18] A. Psarras, I. Seitanidis, C. Nicopoulos, and G. Dimitrakopoulos. 2015. PhaseNoC: TDM scheduling at the virtual-channel level for efficient network traffic isolation. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*. 1090–1095. doi:10.7873/DATE.2015.0418
- [19] Antonis Psathakis, Vassilis Papaefstathiou, Nikolaos Chrysos, Fabien Chaix, Evangelos Vasilakis, Dionisios Pnevmatikatos, and Manolis Katevenis. 2015. A Systematic Evaluation of Emerging Mesh-like CMP NoCs. In *ANCS*. 159–170.
- [20] Akeana 5000 Series. 2024. Akeana. <https://www.akeana.com/product/akeana-5000-series/>
- [21] Sophon SG2042. 2024. Sophgo. <https://en.sophgo.com/sophon-u/product/introduce/sg2042.html>
- [22] AMBA5 CHI Specification. 2024. ARM. <https://www.arm.com/architecture/system-architectures/amba/amba-5>
- [23] TileLink Specification. 2024. SiFive. <https://www.sifive.com/document-file/tilelink-spec-1.9.3>
- [24] TT-Ascalon. 2024. Tenstorrent. <https://tenstorrent.com/ip/tt-ascalon>
- [25] Veyron V1 and V2. 2024. Ventana Microsystem. <https://www.ventanamicro.com/technology/risc-v-cpu-ip/>