Exploring the Potential of Wireless-enabled Multi-Chip AI Accelerators

Emmanuel Irabor, Mariam Musavi, Abhijit Das and Sergi Abadal

Universitat Politècnica de Catalunya, Barcelona, Spain

emmanuel.onyekachukwu.irabor@estudiantat.upc.edu

Abstract—The insatiable appetite of Artificial Intelligence (AI) workloads for computing power is pushing the industry to develop faster and more efficient accelerators. The rigidity of custom hardware, however, conflicts with the need for scalable and versatile architectures capable of catering to the needs of the evolving and heterogeneous pool of Machine Learning (ML) models in the literature. In this context, multi-chiplet architectures assembling multiple (perhaps heterogeneous) accelerators are an appealing option that is unfortunately hindered by the still rigid and inefficient chip-to-chip interconnects. In this paper, we explore the potential of wireless technology as a complement to existing wired interconnects in this multi-chiplet approach. Using an evaluation framework from the state-of-the-art, we show that wireless interconnects can lead to speedups of 10% on average and 20% maximum. We also highlight the importance of load balancing between the wired and wireless interconnects, which will be further explored in future work.

Index Terms—AI Accelerators, Multi-Chiplet Accelerators, Wireless Interconnects, Network-on-Package (NoP).

I. INTRODUCTION

AI has revolutionized a wide range of fields thanks to its superhuman classification, discrimination, and generative capabilities [1]–[3]. However, the continuous advancements made in the different ML models that sustain such a revolution also lead to a constant increase in their computational requirements. Indeed, due to their evolving size and diversity, modern ML models urgently call for faster, more efficient, and more flexible computing platforms [4].

To address the speed and efficiency issues, a wide range of specialized hardware accelerators have been presented in the last decade [5]–[7]. These accelerators are typically composed of a large array of processing elements (PEs, generally in the form of multiply-accumulate units) implementing a given fixed dataflow through a dense Network-on-Chip (NoC) [8]. These architectures go beyond the performance and energy efficiency of Graphical Processing Units (GPUs), yet at the cost of a loss of generality or versatility. Indeed, it is extremely challenging to scale and reconfigure such AI accelerators to execute ever-growing and heterogeneous AI workloads without sacrificing performance and efficiency [9], [10].

Chiplet technology is a promising approach that could enable the creation of scalable and versatile AI accelerators, by combining together multiple specialized (and potentially
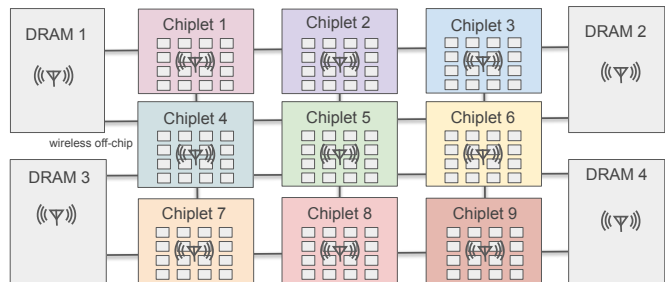
Figure 1: A schematic architecture of wireless-enabled multi-chip AI accelerator with 3×3 chiplets and 4 DRAMs. An antenna and transceiver are integrated at the center of each DRAM and compute chiplet.

heterogeneous) AI accelerator chiplets in a single computing platform, as illustrated in Figure 1. These chiplets are interconnected among themselves and to memory via on-package links, typically through silicon interposers or organic substrates, in order to create a Network-on-Package (NoP) [11]–[13]. Hence, chiplet-based architectures show a promising path to address the scaling challenge of computing platforms, as hinted in multiple works, including SIMBA [14] or WIENNA [15].

One of the main drawbacks of multi-chiplet architectures is the reduction of the interconnect performance and efficiency, which is crucial when serving workloads that imply significant data movement across the architecture [16], [17]. This is not only due to the limited speed of memory modules but, more critically, the relatively slow chiplet-to-chiplet data transfers. These data transfers often dominate the system energy as they rely on traversing the long interconnects. The problem is further intensified by the use of collective (i.e. multicast and reduction) communication in many dataflows employed by AI accelerators [4], [15], [18].

To illustrate the point made above, we simulated multiple AI workloads over a 144-TOPS accelerator broken down in 3×3 chiplets using the methods described later in Section III and summarized in Table 1. We recorded, for each of the layers of the workload, its execution time and which is the performance bottleneck. Figure 2 summarizes the results, showing the percentage of the time that each element of the architecture is a bottleneck. The results clearly indicate that the chiplet interconnects (i.e. the NoP) can be a very significant limiting
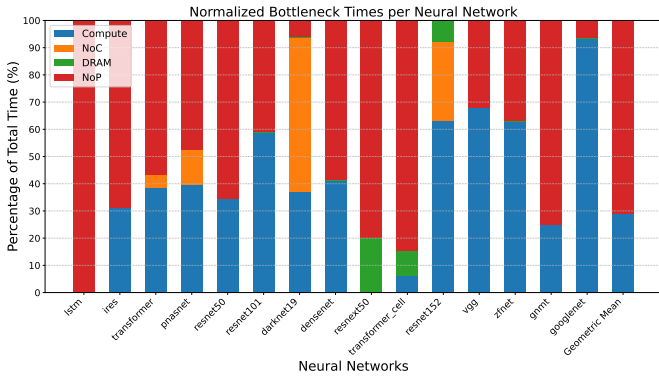
Figure 2: Percentage of time where each of the elements of a 144-TOPS 3×3 multi-chip AI accelerator is the performance bottleneck.

factor hindering the performance and efficiency of multi-chip AI accelerator architectures.

Given that multicast traffic is one of the sources of inefficiency in this context [15], [18], wireless technology appears as a promising candidate to complement existing chiplet interconnects due to its low latency, reconfigurability, and inherent broadcast nature [19]. Antennas and transceivers operating at millimeter-wave frequencies can occupy less than 1 mm^2, operate at ~1 pJ/bit and reach speeds in excess of 100 Gb/s [20]–[22]. In this context, building multi-chip AI accelerator architectures with wireless interconnects is expected to relieve the NoP bottleneck of existing accelerators in an effective and versatile way [19]. However, state-of-the-art developments either study how to improve a particular dataflow [23] or optimize the architecture template [15] without making sure that (i) the mapping of the workloads on the architectures are optimal, and that (ii) the wireless link is used judiciously.

In this paper, we further delve into the study of cost-effective scalable wireless-enabled multi-chip AI accelerators with the aid of GEMINI [24], which allows finding optimal mappings in multi-chiplet architectures. In particular, we modify the code of GEMINI to: (i) perform a brief study of the bottleneck of several AI workloads optimally mapped in multi-chiplet architectures, (ii) propose a reconfigurable wireless-enabled architecture capable of alleviating the NoP bottleneck across workloads, and (iii) assess the potential of our approach to increase the performance of multi-chiplet AI accelerators. We show an average (max) speedup of around 10% (20%) in a 3×3 multi-chip architecture as well as the importance of balancing the load between the wired and wireless planes.

The remainder of the paper is organized as follows. In Section II we present the related work. Section III describes the proposed architecture to mitigate the workload-specific data movement bottleneck of existing architectures and outlines the methodology to evaluate it. In Section IV, we discuss the results and in Section V we conclude the paper.

II. RELATED WORK

A. Design space exploration in multi-chip AI accelerators

Multi-chip AI accelerators have been proposed in multiple works. In SIMBA [14], authors developed a multi-chip AI inference accelerator with a hierarchical interconnect architecture on a package. The objective was to enhance the energy efficiency and reduce the accelerator's inference latency by partitioning the non-uniform workload, considering communication-aware data placement, and implementing cross-layer pipelining.

In GEMINI [24], authors developed a scalable multi-chip AI inference accelerator design framework that explores the design space to deliver architectures for a particular workload and minimizes the monetary cost and the Energy-Delay Product (EDP). In more recent work, the authors in SCAR [25] developed a multi-chip multi-model AI inference accelerator that is scalable under heterogeneous traffic models (i.e., data center and AR/VR) with the objective of minimizing the EDP of the overall system.

Undoubtedly, these and similar multi-chip accelerator architectures have demonstrated excellent scalability and efficiency [26]. However, a significant gap remains in exploring the performance improvements achievable through wireless interconnects, which could address communication bottlenecks, improve energy efficiency, and enable more versatile designs.

B. Wireless-enabled multi-chip AI accelerators

There are a few works that investigated wireless NoP to achieve performance enhancements in multi-chip AI accelerators. In [23], the authors explored the use of wireless-enabled NoP in chiplet-based DNN accelerators to address the bottlenecks caused due to inter-chiplet communication. By leveraging single-hop communication and broadcast capabilities, wireless-enabled NoP significantly reduces latency and energy consumption, achieving superior EDP performance compared to traditional wired NoP. In WIENNA [15], the authors presented an NoP-based 2.5D DNN accelerator that addresses the bandwidth and scalability challenges of interposer-based designs. By utilizing wireless NoPs for high-bandwidth multicasting, WIENNA demonstrated significant improvements in both throughput and energy efficiency, providing a scalable solution for DNN acceleration. These works illustrated the potential of wireless-enabled AI accelerators but left several unexplored gaps in terms of optimal mapping or load balancing between wired and wireless interconnection networks.

In [27], authors explore the potential of in-package wireless communication as a scalable solution for multi-chiplet designs, reducing design complexity and freeing package resources by eliminating physical chiplet-to-substrate connections. Through simulations, it demonstrates that wireless interconnects can match or outperform wired alternatives, particularly for workloads like Convolutional Neural Networks (CNNs), with performance influenced by the wireless protocols and application mapping strategies. In this case, however, the AI workloads were mapped in general-purpose architectures.
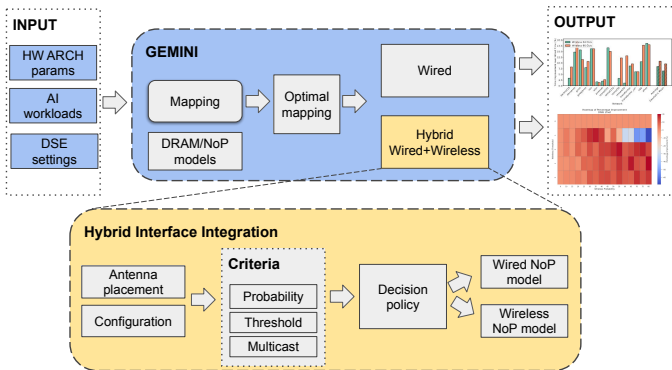
Figure 3: Overall methodology. GEMINI is augmented with a wireless communication model and a wireless interface model. This allows to assess the impact of wireless interconnects on optimally mapped workloads in multi-chip AI accelerators.

C. Workload Mapping on multi-chip AI accelerators

The mapper in SIMBA employed a layer-sequential mapping-first approach to reduce memory access overheads and data replication, with workload partitioning handled by Timeloop [28] for latency estimation and Accelergy [29] for energy evaluation. The mapper employed in GEMINI [24] uses spatial-temporal mapping and inter-layer pipelining using SET [30] mapper tool, and a cost model was customized to evaluate the cost of architecture. These enhancements significantly improved the overall performance of the system compared to the baseline mapping variants. The mapper used in SCAR is also based on SET [30], and the hybrid cost model is customized using MAESTRO [31]. These works, particularly GEMINI, are leveraged in this paper to study the impact of wireless interconnects in optimally mapped AI workloads in multi-chip AI accelerators.

III. METHODOLOGY

In this work, we extend the GEMINI framework [24] by integrating wireless communication channels to alleviate the NoP load and improve the overall latency. This section outlines the modifications that have been made to the GEMINI simulator and the decision criteria to opt for the wireless channel, as illustrated in Figure 3. Also, Table 1 summarizes the values of the main simulation parameters.

A. Architecture

GEMINI is a mapping and architecture co-exploration framework for DNN inference chiplet accelerators. The original GEMINI architecture consists of multiple chiplets, each containing a mesh of PEs, and DRAM modules connected through wired communication links. Data transfer between chiplets and DRAM modules is done via the Die-to-Die (D2D) links, which can become a bottleneck for large-scale DNN workloads due to high communication overhead. On top of this architecture, illustrated in Figure 1, we integrate wireless communication capabilities as described in the subsequent text.

Table 1: Simulation Parameters

Number of Chiplets	3×3
DRAM Configuration	4 Chiplets, 16 GB/s per chiplet
NoP Configuration	XY Mesh, 32 Gb/s per side
NoC Configuration	XY Mesh, 64 Gb/s per port
Wireless Bandwidth	64 Gb/s, 96 Gb/s
Distance Threshold	1, 2, 3, 4 NoP hops
Injection Probability	10% to 80% with step-size of 5%
AI Workloads	Darknet19, DenseNet, ZfNet, GNMT, Vgg, LSTM, ResNet50, ResNet101, ResNet152, ResNeXt50, PNasNet, Transformer, Transformer Cell, iRESNet, GoogleNet

B. Integration of Wireless Communication

To address the communication bottleneck, we introduce a wireless communication channel into the GEMINI framework. Specifically, a wireless antenna and transceiver are placed on each chiplet as well as all DRAM modules. The antennas enable direct communication among chiplets and between chiplets and DRAMs, potentially reducing the number of hops required for message transfer and, consequently, the communication latency.

1) Antenna Placement and Configuration: The GEMINI simulator has been modified to include an antenna module representing wireless transceivers. We have strategically placed antennas at the center of each chiplet and DRAM module. The coordinates of the antennas are calculated based on the chiplet and DRAM positions to accurately model the physical layout. Therefore, the total number of antennas is equal to the sum of chiplets and DRAM modules. If a message needs to be wirelessly transmitted, it is routed through the NoC to the central routers of the chip, which are connected to the wireless interface.

2) Wireless Communication Decision Criteria: To decide whether to use wireless or wired communication for a given message, we implemented multiple configurable decision functions that consider the following factors:

- **Multi-chip Multicast:** First, the network interface analyzes the destinations of a particular message. If the message is a multicast and there is at least one destination in a different chiplet than the source, wireless communication is used to exploit the broadcast nature of wireless channels.
- **Distance Threshold:** Second, a distance threshold is set to assess the minimum number of NoP hops required to consider wireless communication beneficial. If the number of chip-to-chip hops between the source and destination(s) exceeds this threshold, wireless communication is preferred.
- **Injection Probability:** Finally, a configurable probability is implemented to prevent a potential overuse of wireless channels. This probabilistic parameter ensures that the wireless channel is not saturated and became a source of bottleneck for workloads with a higher number of multi-chip and long-range multicasts.

3) Wireless Communication Modeling: When a message is designated for wireless transmission, it is loaded onto the source antenna and sent directly to the destination antennas, thereby inherently implementing broadcast or multicast functionality by virtue of wireless. The simulator updates the total wireless transmission and reception volumes. The impact on latency and energy consumption is computed based on these parameters.

C. Simulator Modifications for Performance Evaluation

GEMINI is not a cycle-accurate simulator rather employs certain approximations to speed up the simulation and allows fast mapping and exhaustive design space explorations. In particular, GEMINI calculates, layer by layer, computing time for each PE, the memory times for each DRAM chiplet, and interconnect times for each link i.e., NoC and NoP in an aggregated form. Then, it analyzes which element is the bottleneck for each layer. The total execution time is the sum of the maximum latency (i.e., that of the bottleneck) across all the layers of the workload. We also note that, GEMINI is not cycle-accurate, it does not take into consideration factors such as the contention in the NoP/NoC routers or in the DRAM chiplets.

To evaluate the benefits of wireless communication without altering the original simulation and mapping strategy of GEMINI, we simulate both wired and wireless communication paths for each message that qualifies for wireless transmission. We calculate the NoC and NoP hops for both cases and compute the difference to assess the performance improvements of having wireless, as described in subsequent section.

1) Wired Communication Path Simulation: For messages that are sent via wireless channels, we also simulate the wired communication path to determine the number of hops and latency that would have occurred without wireless communication. This involves calling the standard unicast or multicast functions and accumulating the total hops and latency. This approach allows us to quantify the benefits of wireless communication by comparing against the baseline wired communication.

2) Wireless Communication Path Simulation: We simulate the wireless communication by updating the wireless-specific counters and latency calculations. This includes the total wireless transmission and reception volumes, as well as the wireless NoC hops. The simulator tracks the data sent and received via each antenna and models the wireless communication time by dividing the total volume of traffic by the wireless link bandwidth, similar to how GEMINI handles NoP/NoC times.

IV. Performance Evaluation

A. Experimental Setup

We conducted experiments by varying key parameters affecting wireless communication to evaluate its impact on the overall performance. The simulation parameters are summarized in Table 1. The additional parameters introduced for the wireless communication are the wireless bandwidth, with
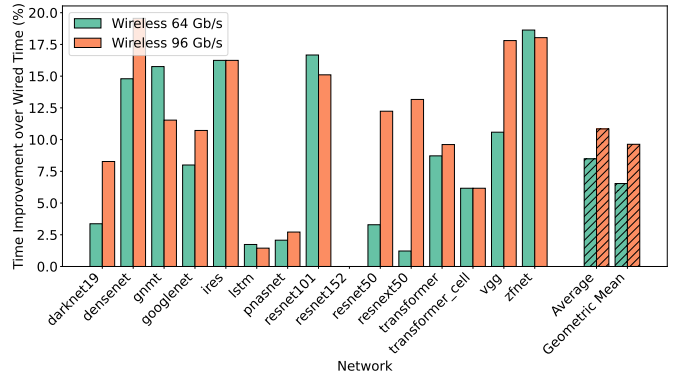

Figure 4: Speedup of the proposed approach over a wired baseline in a 3×3 multi-chip accelerator across the different evaluated AI workloads and for two different wireless bandwidths.

values that commensurate with state-of-the-art transceivers [20], [22], the distance threshold, which is swept from 1 to 4 NoP hops, and the injection probability, which is swept from 10% to 80% with step-size of 5% to assess the importance of load balancing in this scenario.

We tested the modified GEMINI simulator using a set of representative DNN workloads that stress different aspects of the communication infrastructure. We choose benchmark models that contain multi-branch classic residual (e.g. ResNet50, ResNet152, GoogleNet, Transformer (TF), TF Cell) and inception (e.g. iRES) structures with more intricate dependencies which are prevalent and widely used in various scenarios such as image classification and language processing. These workloads differ in size and communication requirements, ensuring that the evaluation captures a wide range of realistic scenarios.

B. Results

We compare the performance of the original GEMINI architecture with the wireless-enhanced version by analyzing the reduction in communication latency and energy consumption. The updated total hops and latency are computed by subtracting the wired communication metrics that were replaced by wireless communication. In our exploration, we sweep the distance threshold and injection probability parameters until finding a near-optimal value for each workload; this way, we assess the acceleration potential of the proposed approach.

Figure 4 shows the exploratory results for the two assumed wireless bandwidths. In particular, we plot the improvement of the hybrid wired-wireless architecture with respect to the wired baseline. There are several observations to be made from this figure:

- The proposed approach improves performance across the board, except in the case of resnet152, which is mostly compute and NoC bound, as observed in Figure 2.
- The average speedups are around 7.5% and 10% for a wireless bandwidth of 64 Gb/s and 96 Gb/s, respectively, with maximum values of almost 20%.
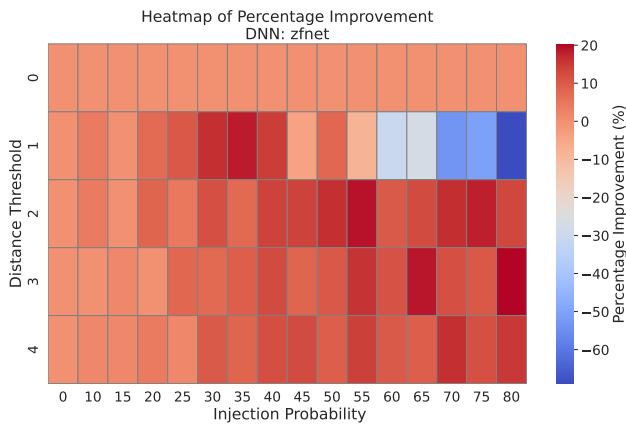
Figure 5: Impact of the distance threshold and injection probability on the performance of the proposed approach for the *zfnet* workload. Hotter colors indicate higher speedups whereas colder colors indicate performance degradations.

- In some cases, an increase in wireless bandwidth does not directly translate to an increase in the speedup. This could be due to the coarse exploration of the distance threshold and injection probability values, leading to a sub-optimal result for higher bandwidth values. This also suggests that the maximum attainable speedups might be higher than the ones shown here.

To illustrate the importance of preventing the wireless network from becoming the bottleneck and the role of the distance threshold and injection probability in this regard, Figure 5 shows the performance improvement (positive values) or degradation (negative values) in the *zfnet* workload as a function of the wireless configuration parameters. Looking from left to right and top to bottom, the figure shows how increasing the load on the wireless link (by maintaining a low distance threshold and increasing the injection probability) can lead to a higher reward. However, the advantage is negated and turns into performance degradation for injection probabilities over 50%. In that scenario, increasing the distance threshold can reduce the pressure on the wireless link and help regain the advantage of the wireless approach. This result, which changes from workload to workload, underscores the need for a mechanism to balance the load between the wired and wireless planes of the accelerator.

V. Conclusion

In this paper, we have showcased the potential of wireless interconnects for improving the performance and flexibility of multi-chip AI accelerators. We have first seen how the NoP can become a bottleneck in these scenarios. Based on our previous works, this is due to multicast patterns leading to congested bisection links, a situation that an overlaid wireless can help prevent. We have then seen how speedups of 20% are achievable but contingent on finding a suitable load-balancing mechanism to prevent the wireless link from

becoming saturated. In future work, we will investigate ways to proactively configure the wireless interface based on offline profiling of AI workloads and investigate alternative mapping methods capable to optimally exploit the advantages of the wireless interconnects.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI open*, vol. 3, pp. 111–132, 2022.

[3] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark *et al.*, "Training compute-optimal large language models," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022, pp. 30 016–30 030.

[4] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, *Efficient processing of deep neural networks*. Springer, 2020.

[5] Y.-H. Chen, T.-J. Yang, J. Emer, and V. Sze, "Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 2, pp. 292–308, 2019.

[6] H. Kwon, L. Lai, M. Pellauer, T. Krishna, Y.-H. Chen, and V. Chandra, "Heterogeneous dataflow accelerators for multi-dnn workloads," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2021, pp. 71–83.

[7] S. Abadal, A. Jain, R. Guirado, J. López-Alonso, and E. Alarcón, "Computing graph neural networks: A survey from algorithms to accelerators," *ACM Computing Surveys (CSUR)*, vol. 54, no. 9, pp. 1–38, 2021.

[8] P. Chatarasi, H. Kwon, A. Parashar, M. Pellauer, T. Krishna, and V. Sarkar, "Marvel: A data-centric approach for mapping deep learning operators on spatial accelerators," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 19, no. 1, pp. 1–26, 2021.

[9] H. Kwon, A. Samajdar, and T. Krishna, "Maeri: Enabling flexible dataflow mapping over dnn accelerators via reconfigurable interconnects," *ACM SIGPLAN Notices*, vol. 53, no. 2, pp. 461–475, 2018.

[10] J. Yang, H. Zheng, and A. Louri, "Versa-dnn: A versatile architecture enabling high-performance and energy-efficient multi-dnn acceleration," *IEEE Transactions on Parallel and Distributed Systems*, 2023.

[11] N. Beck, S. White, M. Paraschou, and S. Naffziger, "'Zeppelin': An SoC for multichip architectures," in *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2018, pp. 40–42.

[12] A. Kannan, N. E. Jerger, and G. H. Loh, "Enabling interposer-based disintegration of multi-core processors," in *Proceedings of the 48th international symposium on Microarchitecture*, 2015, pp. 546–558.

[13] P. Vivet, E. Guthmuller, Y. Thonnart, G. Pillonnet, G. Moritz, I. Miro-Panadès, C. Fuguet, J. Durupt, C. Bernard, D. Varreau *et al.*, "2.3 a 220gops 96-core processor with 6 chiplets 3d-stacked on an active interposer offering 0.6 ns/mm latency, 3tb/s/mm 2 inter-chiplet interconnects and 156mw/mm 2@ 82%-peak-efficiency dc-dc converters," in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2020, pp. 46–48.

[14] Y. S. Shao, J. Clemons, R. Venkatesan, B. Zimmer, M. Fojtik, N. Jiang, B. Keller, A. Klinefelter, N. Pinckney, P. Raina *et al.*, "Simba: Scaling deep-learning inference with multi-chip-module-based architecture," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2019, pp. 14–27.

[15] R. Guirado, H. Kwon, S. Abadal, E. Alarcón, and T. Krishna, "Dataflow-architecture co-design for 2.5D DNN accelerators using wireless network-on-package," in *2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2021, pp. 806–812.

[16] A. Boroumand, S. Ghose, Y. Kim, R. Ausavarungnirun, E. Shiu, R. Thakur, D. Kim, A. Kuusela, A. Knies, P. Ranganathan *et al.*, "Google workloads for consumer devices: Mitigating data movement bottlenecks," in *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2018, pp. 316–331.

[17] A. Das, M. Palesi, J. Kim, and P. P. Pande, "Chip and package-scale interconnects for general-purpose, domain-specific and quantum computing systems-overview, challenges and opportunities," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2024.

[18] M. Musavi, E. Irabor, A. Das, E. Alarcon, and S. Abadal, "Communication characterization of ai workloads for large-scale multi-chiplet accelerators," *arXiv preprint arXiv:2410.22262*, 2024.

[19] S. Abadal, R. Guirado, H. Taghvaee, A. Jain, E. P. de Santana, P. H. Bolívar, M. Saeed, R. Negra, Z. Wang, K.-T. Wang *et al.*, "Graphene-based wireless agile interconnects for massive heterogeneous multi-chip processors," *IEEE Wireless Communications*, 2022.

[20] X. Yu, J. Baylon, P. Wettin, D. Heo, P. P. Pande, and S. Mirabbasi, "Architecture and design of multichannel millimeter-wave wireless noc," *IEEE Design & Test*, vol. 31, no. 6, pp. 19–28, 2014.

[21] K. K. Tokgoz, S. Maki, J. Pang, N. Nagashima, I. Abdo, S. Kawai, T. Fujimura, Y. Kawano, T. Suzuki, T. Iwai *et al.*, "A 120gb/s 16qam cmos millimeter-wave wireless transceiver," in *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2018, pp. 168–170.

[22] C. Yi, D. Kim, S. Solanki, J.-H. Kwon, M. Kim, S. Jeon, Y.-C. Ko, and I. Lee, "Design and performance analysis of thz wireless communication systems for chip-to-chip and personal area networks applications," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 6, pp. 1785–1796, 2021.

[23] M. Palesi, E. Russo, A. Das, and J. Jose, "Wireless enabled inter-chiplet communication in dnn hardware accelerators," in *2023 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2023, pp. 477–483.

[24] J. Cai, Z. Wu, S. Peng, Y. Wei, Z. Tan, G. Shi, M. Gao, and K. Ma, "Gemini: Mapping and Architecture Co-exploration for Large-scale DNN Chiplet Accelerators," in *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2024, pp. 156–171.

[25] M. Odema, L. Chen, H. Kwon, and M. A. A. Faruque, "SCAR: Scheduling Multi-Model AI Workloads on Heterogeneous Multi-Chiplet Module Accelerators," *arXiv preprint arXiv:2405.00790*, 2024.

[26] A. Das, E. Russo, and M. Palesi, "Multi-objective hardware-mapping co-optimisation for multi-dnn workloads on chiplet-based accelerators," *IEEE Transactions on Computers*, 2024.

[27] R. Medina, J. Kein, G. Ansaloni, M. Zapater, S. Abadal, E. Alarcón, and D. Atienza, "System-level exploration of in-package wireless communication for multi-chiplet platforms," in *Proceedings of the 28th Asia and South Pacific Design Automation Conference*, 2023, pp. 561–566.

[28] A. Parashar, P. Raina, Y. S. Shao, Y.-H. Chen, V. A. Ying, A. Mukkara, R. Venkatesan, B. Khailany, S. W. Keckler, and J. Emer, "Timeloop: A systematic approach to DNN accelerator evaluation," in *2019 IEEE international symposium on performance analysis of systems and software (ISPASS)*. IEEE, 2019, pp. 304–315.

[29] Y. N. Wu, J. S. Emer, and V. Sze, "Accelergy: An architecture-level energy estimation methodology for accelerator designs," in *2019 IEEE/ACM International Conference on Computer-Aided Design (IC-CAD)*. IEEE, 2019, pp. 1–8.

[30] J. Cai, Y. Wei, Z. Wu, S. Peng, and K. Ma, "Inter-layer scheduling space definition and exploration for tiled accelerators," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, 2023, pp. 1–17.

[31] H. Kwon, P. Chatarasi, V. Sarkar, T. Krishna, M. Pellauer, and A. Parashar, "Maestro: A data-centric approach to understand reuse, performance, and hardware cost of DNN mappings," *IEEE micro*, vol. 40, no. 3, pp. 20–29, 2020.