Wireless enabled Inter-Chiplet Communication in DNN Hardware Accelerators

Maurizio Palesi[1], Enrico Russo[1], Abhijit Das[2] and John Jose[3]
[1] *University of Catania, Italy*
[2] *Univ Rennes, Inria, France*
[3] *Indian Institute of Technology Guwahati, India*
maurizio.palesi@unict.it, enrico.russo@phd.unict.it, abhijit.a.das@inria.fr, johnjose@iitg.ac.in

***Abstract*—Inter-chiplet communication is a fundamental bottleneck in scale-out Homogeneous Multi-Chip-Module-based Hardware Accelerators (HMCMHAs). This paper focuses on the problem of many-to-many communication traffic generated when dispatching output feature map tiles among chiplets. Such traffic has a strong impact on the latency and energy metrics of the HMCMHAs as it exposes the limitations of the existing wire-based Network-on-Package (NoP). This paper investigates augmenting the existing NoP with emerging wireless in-package communication links. The intrinsic single-hop and broadcast-capable technology is exploited to tackle the many-to-many communication traffic in question. We show that the proposed wireless-enabled NoP can significantly improve the latency and energy of Deep Neural Network (DNN) inference on HMCMHAs.**

***Index Terms*—Deep Neural Network (DNN) Hardware Accelerator, Domain Specific Architecture (DSA), Multi-Chip-Module (MCM), Network-on-Package (NoP), Wireless-enabled NoP.**

I. Introduction

Many daily applications we interact with are powered by Deep Learning (DL) techniques at their core. This includes features such as natural language processing and speech recognition in intelligent personal assistants, advanced driver assistance systems in cars, and user behaviour prediction and data security in smart homes. These DL techniques are implemented using Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), Graph Neural Networks (GNNs), etc. The good news is that these techniques are easy to parallelise and scalable, making general-purpose Graphics Processing Units (GPUs) the preferred platform for their execution. However, limited memory bandwidth, low performance-per-watt, and area overheads of GPUs are leading to a paradigm shift towards Domain Specific Architectures (DSAs) [6] [7] to improve efficiency[1] and avoid the so-called "Turing Tariff".

In the context of DNNs, industry and academia have proposed various DSAs, aka DNN hardware accelerators. Despite differences in tasks, such as training or inference, or targets, like Internet-of-Things (IoT), edge, or cloud, these accelerators share a similar architecture where the communication subsystem is critical and plays a significant role. The parallel processing units that constitute the architecture must have a steady supply of data to avoid underutilisation of resources and loss of efficiency. The communication subsystem, together with the memory subsystem, manages the dispatch and retrieval of data and determines the overall latency and energy consumption.

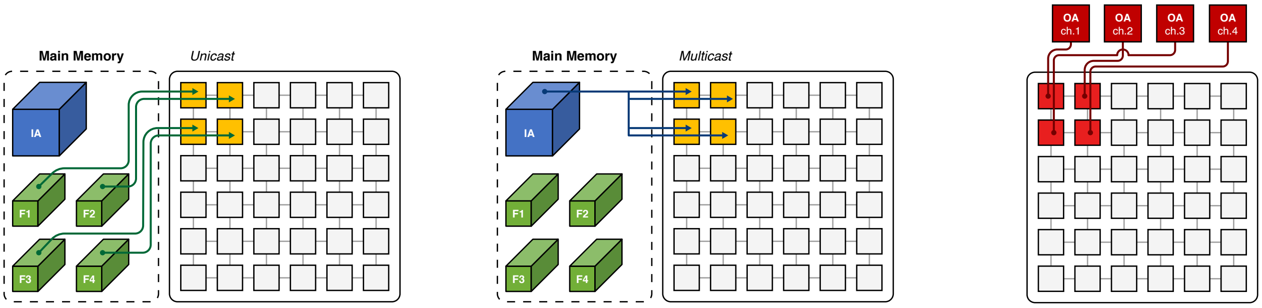
This paper focuses on scale-out Homogeneous Multi-Chip-Module-based Hardware Accelerators (HMCMHAs), where the chiplets are interconnected with a communication subsystem called Network-on-Package (NoP). Existing literature shows that NoP plays a significant role in determining the overall performance of the HMCMHAs [20]. The challenge we address is the transfer of the output feature map from one set of chiplets that execute a layer of the DNN to another set of chiplets that execute the next layer. This transfer is multicast in nature and hence induces heavy traffic volume into the NoP. We investigate how the latency and energy metrics scale when the NoP is extended to support in-package wireless communication using short-range millimeter-wave (mm-Wave) technology [17]. We advocate that the naturally single-hop and broadcast-capable wireless-enabled NoP could be a potential solution to improve the efficiency of the HMCMHAs.

II. Background

Many hardware accelerators are proposed for the efficient execution of DL models, and in particular, for the DNNs [20] [5] [13]. These accelerators typically have a memory hierarchy, interconnection networks, and units capable of performing Multiply-And-Accumulate (MAC) operations, which are fundamental to DL algorithms. Lower-level buffers in the memory hierarchy have limited storage capacity, high transfer bandwidth and lower access energy, while higher-level buffers have bigger storage capacity but are slower and energy-hungry. Most of these accelerators employ a spatial architecture where MAC units and memories are replicated as multiple instances to parallelise workloads and speed up the execution of models.

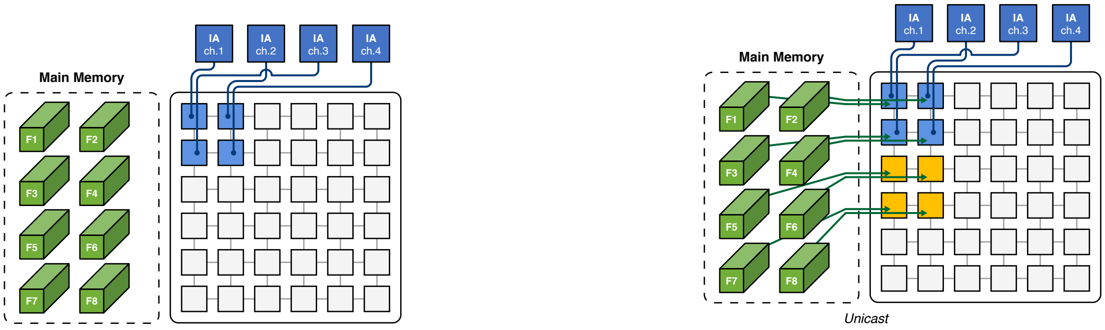
Recently, the use of 2.5D integration, which involves integrating multiple small chips (i.e., chiplets) on an interposer, is increasingly gaining popularity as it represents an effective solution to improve yield and reduce fabrication costs. Moreover, 2.5D integration also facilitates heterogeneous integration [22] and allows for compact scale-out implementations of emerging larger and complex DL models [21]. A notable example in this context is Simba [20], a Multi-Chip-Module (MCM) based multi-accelerator system, where each chiplet is a DNN accelerator (called sub-accelerator in the bigger context) interconnected by an NoP. Unfortunately, scalable

[1] Efficiency in this paper refers to reducing latency and energy consumption.



(a) (b) (c)

Figure 1: Fetching the filters and *IA*, performing the convolution and generating the *OA* for the current layer L_i.



(a) (b)

Figure 2: Fetching the filters for the next layer L_{i+1}.

communication between chiplets is particularly challenging due to relatively large physical distances between chiplets, poor technology scaling of electrical wires, and shrinking power budgets. These issues get worsened in the case of DNN accelerators as they exert tremendous pressure on the NoP.

In-package wireless communication using short-range mm-Wave technology [17] opens many interesting scenarios in the context of efficient and scalable HMCMHAs. Several wireless transceiver designs [4] [15] [10] and system-level simulations [1] [19] [3] [2] have shown that wireless interconnects in the mm-Wave bands can reduce energy consumption and increase the bandwidth of chip-to-chip communication significantly as compared to the traditional metallic wired interconnects. Thus, this paper investigates the impact on performance[2] and energy figures when the traditional wired NoP in HMCMHAs is replaced with a wireless-enabled NoP.

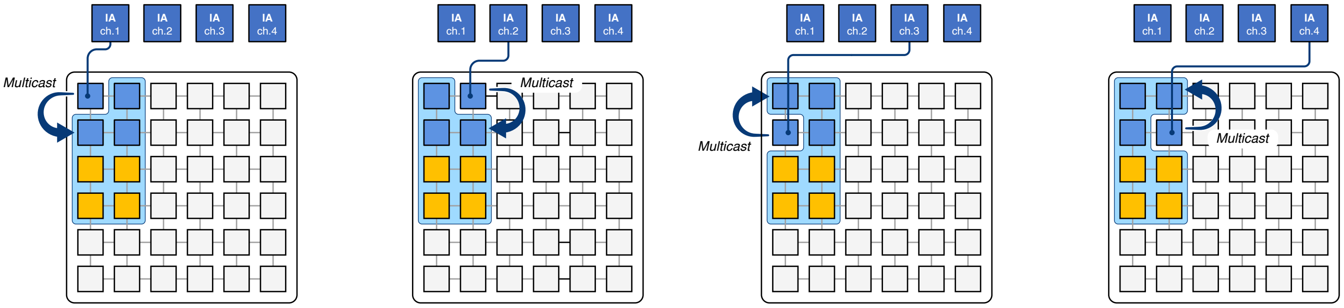
III. Motivation by Example

The communication issues in DNN accelerators have become a fundamental bottleneck for performance and energy figures. The impact of this bottleneck intensifies in MCM-based designs where the latency and energy-per-bit of NoP links are significantly imbalanced (in a negative way) compared to those of Network-on-Chip (NoC) links. For example, it is reported in the literature [20] that the interconnect latency in NoP is 20ns/hop, while in NoC, it is half at just 10ns/hop. Similarly, the interconnect energy in NoP is 1.75pJ/bit, while in NoC, it is significantly lower ($<$ one-fourth) at just 0.4pJ/bit.

To understand the specific problem, we present an illustrative example. Consider an HMCMHA with 36 chiplets and a weight-stationary dataflow executing a CNN workload. A given layer L_i of the workload uses 4 filters, *F1*, *F2*, *F3* and *F4*, for its execution, as shown in Figure 1a. Suppose L_i is mapped into 4 chiplets, each devoted to performing a convolution between the corresponding filter and the input feature map (*IA*). As shown in Figure 1a, each filter is fetched from the main memory into the corresponding chiplet using unicast communication. Whereas the *IA* is fetched and distributed among the involved chiplets using multicast communication, as shown in Figure 1b. Each chiplet performs the convolution and produces the activations of 4 corresponding channels of the output feature map (*OA*), as shown in Figure 1c.

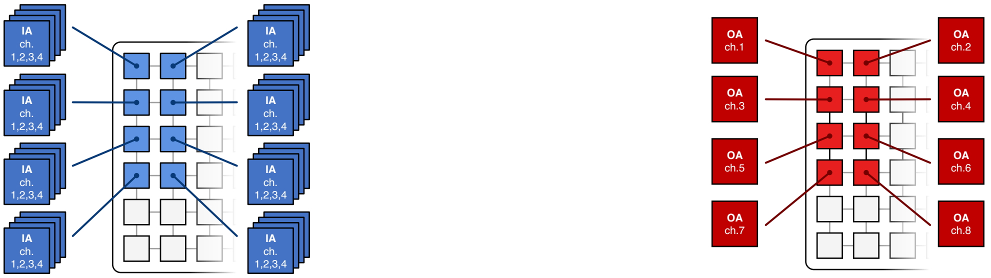
The *OA* of layer L_i becomes the *IA* for the next layer L_{i+1} of the workload. As shown in Figure 2a, L_{i+1} uses 8 filters for its execution and is mapped into 8 chiplets, each devoted to applying a filter to its *IA*. The filters are fetched into the corresponding chiplets using unicast communication, as shown in Figure 2b. Suppose 4 of these chiplets (shown in blue) were involved in the execution of the previous layer L_i. Hence, the blue chiplets have one channel each of the current *IA*, whereas the other 4 chiplets (shown in yellow) have nothing. Nevertheless, none of the chiplets can initiate the convolution as they either have partial or no *IA*. Each of the blue chiplets

[2] Performance in this paper refers to latency, hence used interchangeably.



(a) (b) (c) (d)

Figure 3: Disseminating the partial *OA* of layer L_i to build the *IA* for layer L_{i+1}.



(a) (b)

Figure 4: Performing the convolution and generating the *OA* for the layer L_{i+1}.

must multicast its *IA* channel to the other chiplets involved in the execution of L_{i+1}, as shown in Figures 3a through 3d.

At the end of this massive all-to-all communication, each of the involved chiplets (now all shown in blue) has the complete *IA*, as shown in Figure 4a. Each of these chiplets performs the convolution and produces the activations of 8 corresponding channels of the new *OA*, as shown in Figure 4b.

The dissemination of the partial *OA* from the involved chiplets of layer L_i to build the *IA* in the involved chiplets of layer L_{i+1} requires intense inter-chiplet communication. This stresses the NoP and consequently has a strong impact on both performance and energy. For example, Figure 5 shows the computation and communication latency and energy breakdown for some popular CNN-based DL models. The inferences are run on an HMCMHA platform considered for the experiments of this work (refer Section V). As it can be observed, on average, the contribution of the partial output feature maps dissemination traffic (communication) accounts for 83% and 80% of inference energy and latency, respectively.

In the remainder of the paper, we investigate wireless-enabled inter-chiplet communication for the dissemination of partial output feature maps. The intrinsically broadcast-capable wireless links can tackle this massive all-to-all communication.

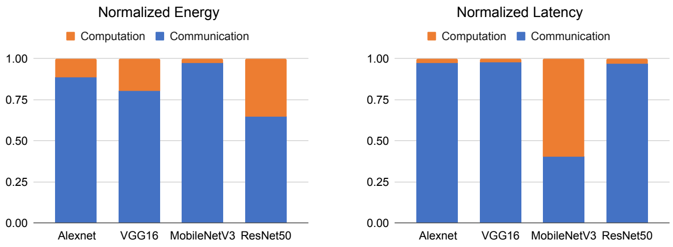


Figure 5: Energy and latency breakdown of some DL models.

IV. Reference Platform and Mapping Selection

A. *HMCMHA Platform*

We consider Simba [20] as a reference HMCMHA platform. Simba is an MCM-based multi-accelerator system that features 36 homogeneous chiplets. A wired NoP interconnects these chiplets (or sub-accelerators). Each of them is a DNN accelerator and has an array of 16 Processing Elements (PEs) interconnected by an NoC, and a shared Global Buffer (GB). Each PE houses 8 vector MAC units (each capable of performing 8:1 dot-product) that fetch data from an Input Buffer (IB) and a Weight Buffer (WB) and store results into an Accumulation Buffer (AB). GB collects weights and activations from the main memory through the NoP and distributes them to IB through the NoC. Similarly, outputs are written from IB to GB and then to memory through NoC and NoP, respectively. Figure 6 shows the block diagram of the reference Simba and the proposed wireless-enabled version we call *WSimba*. The only difference is in the NoP, where Simba employs a wired NoP and WSimba employs a wireless-enabled NoP. Each chiplet in WSimba is augmented with a wireless interface.

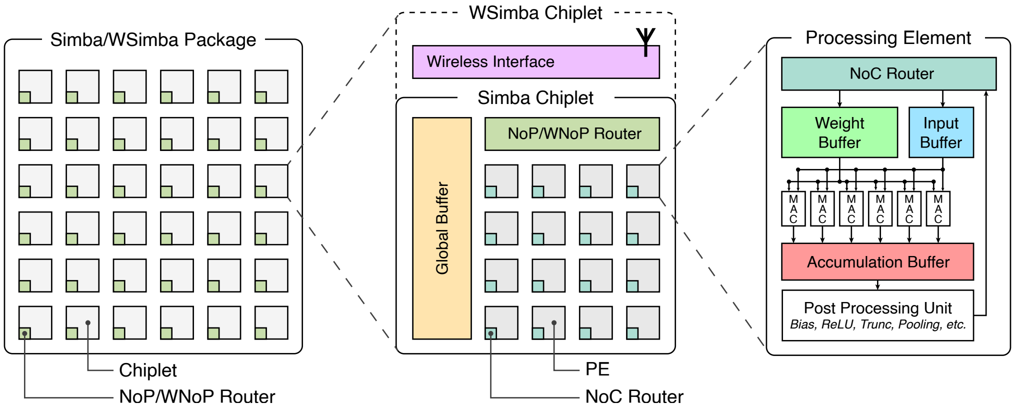


Figure 6: Reference Simba and proposed WSimba architectures.

B. Dataflow and Mapping

We consider a weight-stationary dataflow as shown in Figure 7 in a loop-nest form. As it can be observed, for each level of the system hierarchy, weights are uniformly partitioned spatially (along the input channel C and the output channel K), while the remaining dimensions are tiled temporally.

For mapping a layer with dimensions P, Q, R, S, C, and K (dimensions H and W are derived from others), we proceed as follows: We determine the bounds $C0$, $C1$, $C2$, $C3$, $K0$, $K1$, $K2$, $K3$, $P1$, $P2$, $P3$, $Q1$, $Q2$, $Q3$ in such a way to:

$$\max C0 \times K0 \tag{1}$$

$$\max C2 \times K2 \tag{2}$$

$$\max C3 \times K3 \tag{3}$$

$$\min P3 \times Q3 \tag{4}$$

Where Equation (1) aims to maximise MAC unit utilisation, Equation (2) aims to maximise PE utilisation, Equation (3) aims to maximise chiplet utilisation, and Equation (4) aims to minimise the number of tiles used to partition the output feature map, thereby reducing the number of iterations at the package level. This problem has the following constraints:

$$\frac{K \times P \times Q}{K3 \times C3 \times P3 \times Q3} \leq \mathit{GBSize} \tag{5}$$

$$K3 \times C3 \leq \mathit{NumChiplets} \tag{6}$$

$$K2 \times C2 \leq \mathit{NumPEperChiplet} \tag{7}$$

$$K0 \times C0 \leq \mathit{NumMACperPE} \tag{8}$$

$$K0 \times K1 \times K2 \times K3 = K \tag{9}$$

$$C0 \times C1 \times C2 \times C3 = C \tag{10}$$

$$P1 \times P2 \times P3 = P \tag{11}$$

$$Q1 \times Q2 \times Q3 = Q \tag{12}$$

$$C0, C1, C2, C3, K0, K1, K2, K3, \\ P1, P2, P3, Q1, Q2, Q3 \in \mathbb{N} \tag{13}$$

Where Equation (5) states that the output feature map size computed by a chiplet cannot exceed the GB size (*GBSize*), Equation (6), (7), and (8) states that the parallelisation degree over chiplets, PEs, and MAC units cannot exceed the number of chiplets (*NumChiplets*), the number of PEs per chiplet (*NumPEperChiplet*), and the number of MAC units per PE (*NumMACperPE*), respectively. Equation (9) through (12) states that K, C, P, and Q dimensions must be factorised considering their respective tiling upper bounds at each level of the system hierarchy, and finally, Equation (13) states that all the considered dimension variables must be positive integers.

```
//Package level
for p3=[0:P3):
  for q3=[0:Q3):
    parallel_for k3=[0:K3):
      parallel_for c3=[0:C3):
// Chiplet level
for p2=[0:P2):
  for q2=[0:Q2):
    parallel_for k2=[0:K2):
      parallel_for c2=[0:C2):
// PE level
for r=[0:R):
  for s=[0:S):
    for k1=[0:K1):
      for c1=[0:C1):
        for p1=[0:P1):
          for q1=[0:Q1):
// Vector-MAC level
parallel_for k0=[0:K0):
  parallel_for c0=[0:C0):
    p = (p3 * P2 + p2) * P1 + p1
    q = (q3 * Q2 + q2) * Q1 + q1
    k = ((k3 * K2 + k2) * K1 + k1) * K0 + k0
    c = ((c3 * C2 + c2) * C1 + c1) * C0 + c0
    OA[p,q,k] += IA[p-1+r,q-1+s,c] * W[r,s,c,k]
```

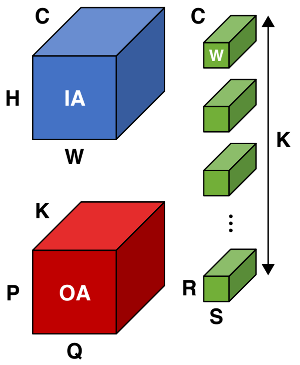


Figure 7: Weight-stationary dataflow loop-nest representation.

For each layer of the considered DNN workload, we solve the above nonlinear multi-objective optimisation problem with nonlinear integer constraints by using a multi-objective Genetic Algorithm (GA) based on NSGA-II [9]. We sort the solutions of the Pareto set using Equation (1) through (4), in that exact order. We select the first solution to map the layer.

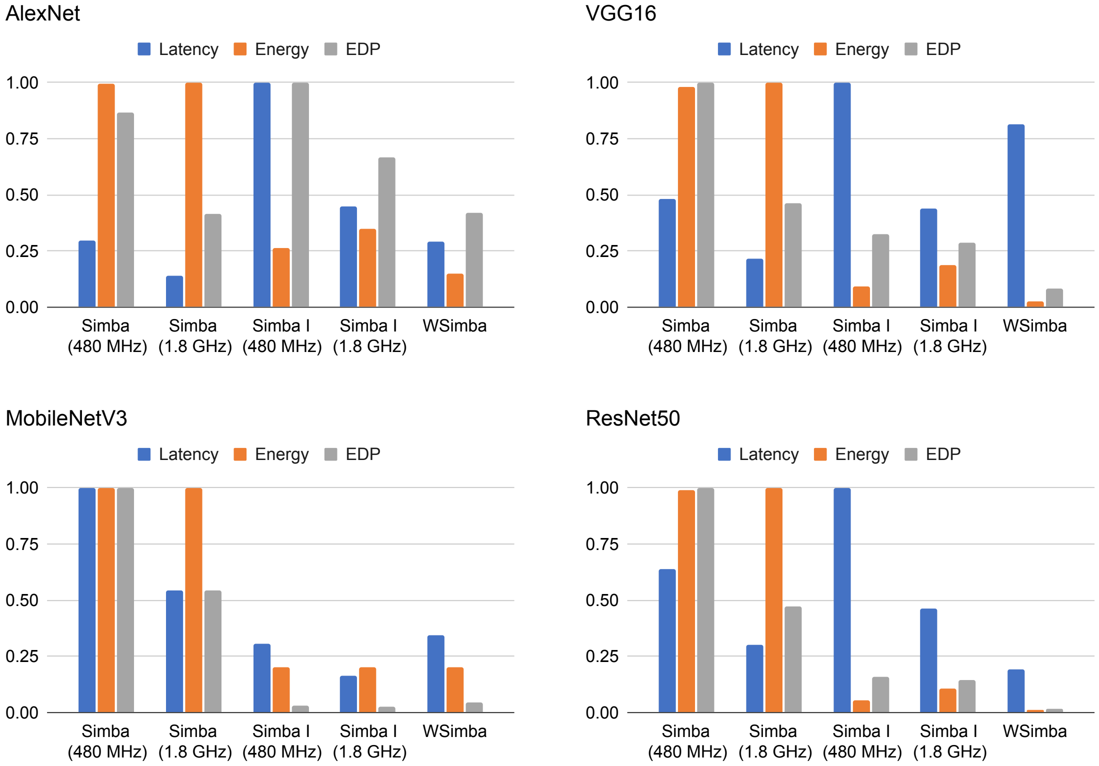


Figure 8: Inference latency, energy and EDP for different DNN-based DL models.

V. Experiments

A. Simulation Setup and System Configurations

We use LAMBDA [18] as the simulation platform to run the experiments. We consider 3 different system configurations: *Simba*, *Simba I*, and *WSimba*. In Simba, for each layer, the input feature map is fetched from the external DRAM (main memory), and the output feature map is stored back in the main memory. In Simba I, the first input feature map is fetched from the main memory, and the output feature map is kept in the GB of the chiplets to be used as the input feature map for the next layer as discussed in Section III and shown in Figure 1. The output feature map of the last layer is stored in the main memory. Both Simba and Simba I use a wired 2D Mesh NoP that employs a deterministic XY routing algorithm. WSimba extends Simba I by augmenting chiplets with a wireless interface. The transceiver design from [16] is used, which considers low-power design aspects at the architecture level. Non-coherent On-Off Keying (OOK) modulation is used for simple and low-power circuit implementation.

Table 1 reports the values of bandwidth, latency and energy parameters for the main memory and NoP. For the wired NoP, we consider two design points at 480 MHz and 1.8 GHz [20]. The remaining simulation parameters used for deriving the mapping as discussed in Sub-Section IV-B are given these values: *GBSize* = 64K, *NumChiplets* = 36, *NumPEperChiplet* = 16, *NumMACperPE* = 64. Input/weight precision is 8 bits.

Table 1: Architecture parameters

Main Memory			
DRAM bandwidth	12 GB/s		
DRAM latency	15 ns		
DRAM energy per bit	1 nJ/bit		
Network-on-Package			
Design points	wired 480 MHz	wired 1.8 GHz	wireless
Bandwidth (Gb/s)	44	100	20
Energy per bit (pJ/bit)	0.82	1.75	1.00

B. Experimental Results

We assess the above system configurations under 4 DNN-based DL models, namely, AlexNet [14], VGG16 [21], MobileNetV3 [12], and ResNet50 [11]. The comparison is carried out considering 3 figures of merit: inference latency, inference energy and their product, popularly called Energy-Delay-Product (EDP). Normalised results are shown in Figure 8.

In general observation, WSimba outperforms others in terms of energy efficiency across all the considered workloads. Despite the energy per bit being similar between the wired and wireless-enabled NoPs, the single-hop and broadcast-capable infrastructure of WSimba makes a significant difference.

In terms of latency, the results are more diverse. WSimba

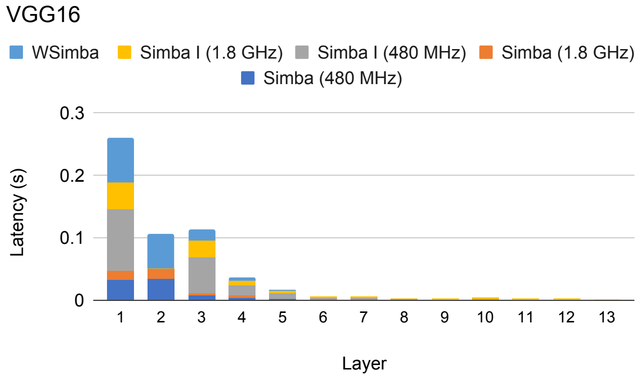


Figure 9: Communication latency at each layer of VGG16.

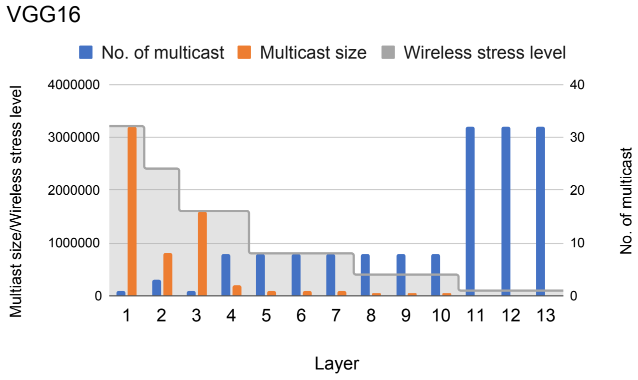


Figure 10: Wireless stress level at each layer of VGG16.

exhibits higher latency values compared to the average only for VGG16, mainly in its initial layers. Figure 9 shows the communication latency with different system configurations at each layer of VGG16. The output feature map size in the initial layers of VGG16 is much bigger than in the deeper layers. Due to lower bandwidth in wireless-enabled NoP compared to the wired counterparts, the massive initial traffic in VGG16 stresses WSimba. This is due to the inefficient utilisation of the wireless infrastructure. For example, Figure 10 shows the number of multicasts, wireless data volume per multicast and their product, called wireless stress level, at each layer of VGG16. It can be observed that the wireless stress level is much higher in the initial layers than in the deeper layers, resulting in higher latency values. The number of multicasts is correlated with the number of active chiplets in subsequent layers, which is determined by the chosen mapping. Hence, better mapping strategies that consider inter-layer information may improve the effectiveness of the wireless-enabled NoP.

In terms of EDP, WSimba outperforms Simba and Simba I.

Finally, we assess WSimba when the wireless-enabled NoP is designed using different physical layers. Apart from the traditional frequency range of mm-Wave using the zigzag antenna with OOK modulation scheme, we explore using Sub-THz with Amplitude-Shift Keying (ASK) modulation scheme and THz using Multi-Walled Carbon NanoTube (MWCNT) antennas with non-coherent OOK modulation scheme. Bandwidth and energy per bit values are derived from [8]. Figure 11 shows the normalised latency and energy for different workloads executed by WSimba when different wireless physical layers are employed. In terms of latency, the higher bandwidth offered by Sub-THz and THz (320 Gb/s and 240 Gb/s, respectively) leads to significant performance improvements. The graph displays the percentage of latency attributable to the main memory, which contributes no more than 11% of the total latency. Advanced physical layers do not significantly improve energy consumption compared to the traditional mm-Wave. Despite the lower energy per bit of the THz-based physical layer compared to the Sub-THz and mm-Wave, overall energy savings are not notable, as the main memory dominates total energy consumption. Except for VGG16, main memory accounts for more than 60% of the total energy consumption.

VI. Conclusion

In this paper, we investigate the use of wireless-enabled NoP in chiplet-based DNN hardware accelerators. We show that inter-chiplet communication accounts for a relevant fraction of the total latency and energy; thus, it is a natural candidate for optimisation. The use of in-package wireless technology, thanks to its single-hop communication and intrinsic broadcast capabilities, is an excellent solution to address the scale-out needs of current and future accelerators and to alleviate the high multicast traffic volume in the critical phase of output feature map dispatching among the chiplets. We show that, compared to the use of traditional wired NoP for inter-chiplet communication, wireless-enabled NoP provides essential energy savings. Notable latency savings are also observed in the majority of the considered workloads. Overall, when the EDP is used as the figure of merit for ranking the different considered implementations, using a wireless-enabled NoP provides the best results in all the considered workloads.

We also investigate using physical layers operating at different frequency ranges: mm-Wave, Sub-THz, and THz. As the energy consumption is dominated by the external DRAM, no relevant differences are noticed in terms of energy. However, the higher bandwidths provided by Sub-THz and THz significantly reduce the latency in all the workloads.

The evaluation is carried out considering two different approaches for managing the delivery of the feature map from one layer to the subsequent one. We observe that avoiding the continuous swap of the input/output feature maps between the main memory and the accelerators in favour of keeping feature maps in the accelerator improves EDP. However, it is essential to define new mapping strategies that consider the inter-layer communication effect among subsequent layers. This aspect is not considered in the paper and is a future research direction.

Acknowledgment

This work was supported by the Research Grants from: MIUR PO FESR SICILIA 2014-2020 within the project

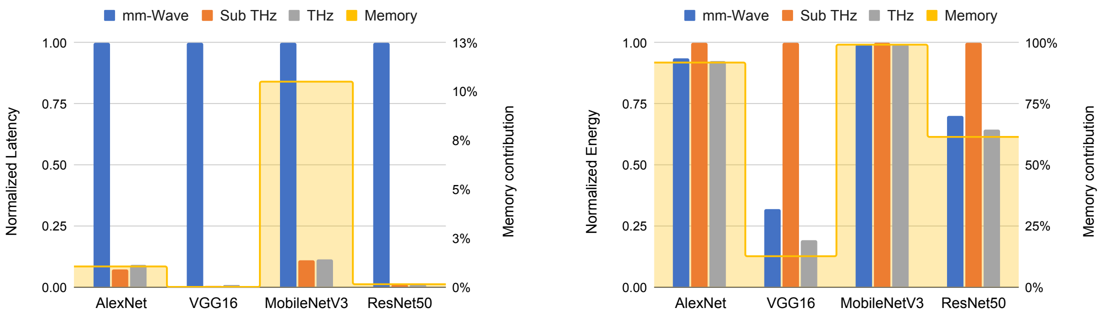


Figure 11: Normalised latency and energy using different wireless physical layers.

"PRE-CUBE: PREdizione, PREvenzione, PREdisposizione", ref. 086201000304; the University of Catania - Piaceri 2020-2022 - Linea 2, within the project MANGO.

References

[1] S. Abadal, A. Mestres, M. Nemirovsky, H. Lee, A. González, E. Alarcón, and A. Cabellos-Aparicio, "Scalability of broadcast performance in wireless network-on-chip," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 12, pp. 3631–3645, 2016.

[2] G. Ascia, V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, "Improving inference latency and energy of dnns through wireless enabled multi-chip-module-based architectures and model parameters compression," in *IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, 2020.

[3] G. Ascia, V. Catania, S. Monteleone, M. Palesi, D. Patti, J. Jose, and V. M. Salerno, "Exploiting data resilience in wireless network-on-chip architectures," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 16, no. 2, apr 2020.

[4] J. Baylon, X. Yu, S. Gopal, R. Molavi, S. Mirabbasi, P. P. Pande, and D. Heo, "A 16-gb/s low-power inductorless wideband gain-boosted baseband amplifier with skewed differential topology for wireless network-on-chip," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 11, pp. 2406–2418, 2018.

[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] Y. Chi, W. Qiao, A. Sohrabizadeh, J. Wang, and J. Cong, "Democratizing domain-specific computing," *Communications of the ACM*, vol. 66, no. 1, pp. 74–85, 2022.

[7] W. J. Dally, Y. Turakhia, and S. Han, "Domain-specific hardware accelerators," *Communications of the ACM*, vol. 63, no. 7, pp. 48–57, Jun. 2020.

[8] D. Deb, A. Ganguly, P. P. Pande, B. Belzer, and D. Heo, "Wireless noc as interconnection backbone for multicore chips: Promises and challenges," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 2, pp. 228–239, 2012.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[10] A. Ganguly, M. M. Ahmed, R. S. Narde, A. Vashist, M. S. Shamim, N. Mansoor, T. Shinde, S. Subramaniam, S. Saxena, J. Venkataraman, and M. Indovina, "The advances, challenges and future possibilities of millimeter-wave chip-to-chip interconnections for multi-chip systems," *Journal of Low Power Electronics and Applications*, vol. 8, no. 1, 2018.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the Annual IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[12] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF Intl. Conference on Computer Vision*, 2019, pp. 1314–1324.

[13] B. Keller, R. Venkatesan, S. Dai, S. G. Tell, B. Zimmer, C. Sakr, W. J. Dally, C. T. Gray, and B. Khailany, "A 95.6-tops/w deep learning inference accelerator with per-vector scaled 4-bit quantization in 5 nm," *IEEE Journal of Solid-State Circuits*, 2023.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proceedings of the Annual Conference on Neural Information Processing Systems*, 2012, pp. 1097—-1105.

[15] S. Laha, S. Kaya, D. W. Matolak, W. Rayess, D. DiTomaso, and A. Kodi, "A new frontier in ultralow power wireless links: Network-on-chip and chip-to-chip interconnects," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 2, pp. 186–198, 2015.

[16] N. Mansoor, P. J. S. Iruthayaraj, and A. Ganguly, "Design methodology for a robust and energy-efficient millimeter-wave wireless network-on-chip," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 1, no. 1, pp. 33–45, 2015.

[17] R. S. Narde, J. Venkataraman, A. Ganguly, and I. Puchades, "Intra- and inter-chip transmission of millimeter-wave interconnects in noc-based multi-chip systems," *IEEE Access*, vol. 7, 2019.

[18] E. Russo, M. Palesi, S. Monteleone, D. Patti, G. Ascia, and V. Catania, "Lambda: An open framework for deep neural network accelerators simulation," in *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2021, pp. 161–166.

[19] M. S. Shamim, N. Mansoor, R. S. Narde, V. Kothandapani, A. Ganguly, and J. Venkataraman, "A wireless interconnection framework for seamless inter and intra-chip communication in multichip systems," *IEEE Transactions on Computers*, vol. 66, no. 3, pp. 389–402, 2017.

[20] 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.

[21] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[22] D. Stow, I. Akgun, R. Barnes, P. Gu, and Y. Xie, "Cost analysis and cost-driven ip reuse methodology for soc design based on 2.5d/3d integration," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016.