Exploring interconnect structure and wiring optimization in FPGA-based many-core architectures

Masaru Nishimura, ○**Shintaro Kawasaki**, Yoshiki Yamaguchi
Graduate School of Science and Technology, University of Tsukuba

2023 RISC-V Days Tokyo Winter @UTokyo, 1/16

1. Introduction

Our mission encompasses three core facets: ① *Issue Identification: We embark on a meticulous exploration to unearth the fundamental issues that plague many-core architectures when implemented using FPGA technology.* ② *Innovative Solutions: To surmount these challenges, we present a novel Wiring Reduction Architecture designed to alleviate the wiring complexity conundrum and enhance overall system performance.* ③ *Empirical Validation: Armed with real-world hardware, we present empirical evidence derived from comprehensive experiments.* Through these findings, we not only substantiate the viability of our proposed methodology but also provide valuable insights that contribute to the broader landscape of computer architecture.

2. Challenges in Implementing Many-Core Architectures on FPGAs

Many-core architectures demand extensive interconnections between many processing elements (PEs) and memory components. This becomes particularly problematic when implementing these complex interconnections in field-programmable gate arrays (FPGAs), which inherently have limited wiring resources. Moreover, integrating PEs with external memory requires additional wiring, exacerbating the wiring complexity issue. As wiring length increases, it introduces concerns about latency, potentially affecting overall system performance. In pursuit of enhanced computational efficiency, traditional approaches have often focused on fully segregating computation and memory elements. For example, higher-order topology was well-studied. However, it leads to inflexibility in configuration and a notable increase in the required wiring connections, which is not good for FPGAs.

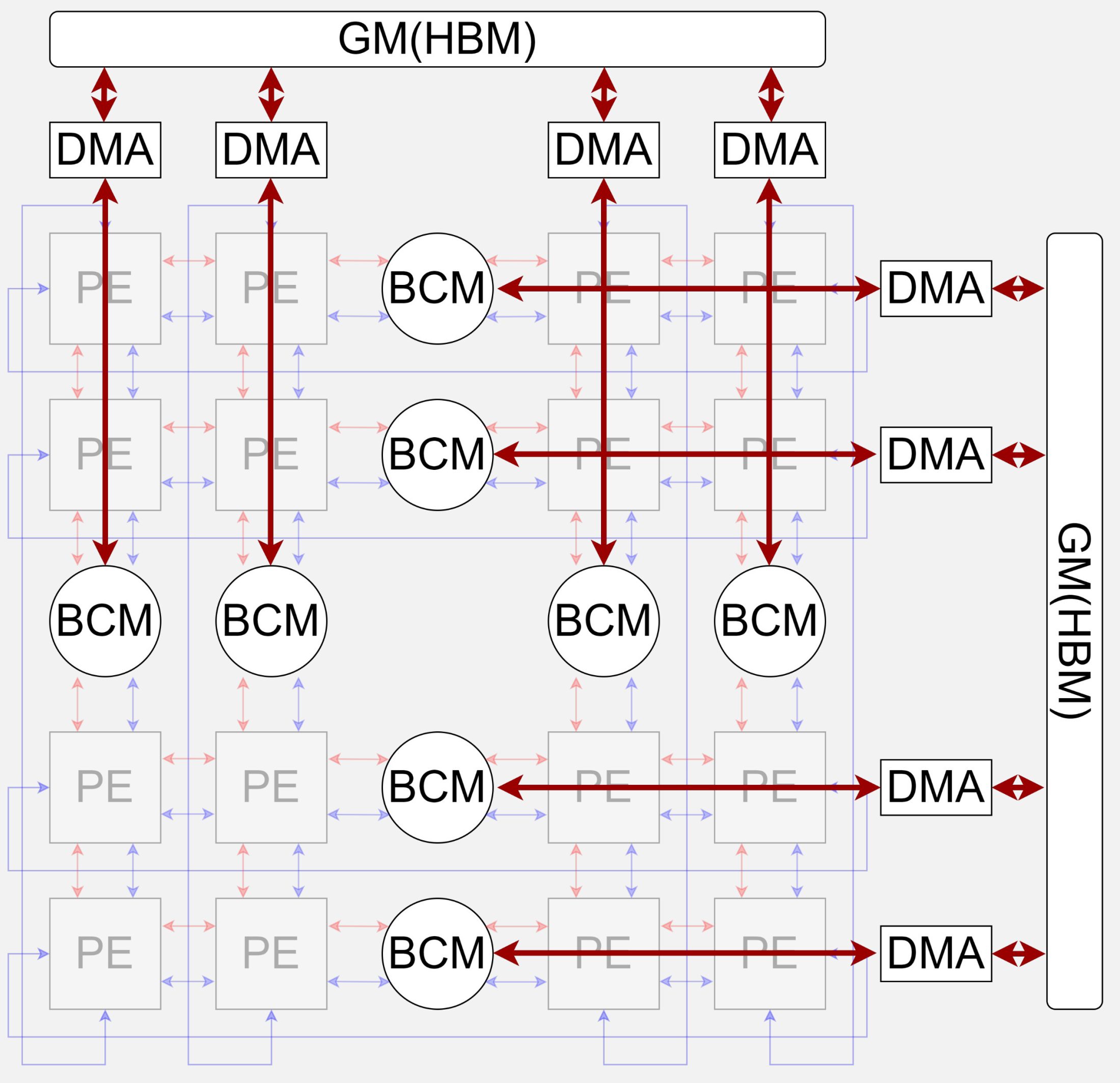
3. Architecture of the Interconnect and memory hierarchy

The following memory hierarchy is introduced in our proposed processor, consisting of three components: **Global Memory**, **Broadcast Memory**, and **Local Memory**. It is pivotal in optimizing data access and management within the processor, enhancing its overall performance and efficiency.

Global Memory (GM): used for data exchange between host computer and each PE.

Broadcast Memory (BCM): Positioned between GM and LM. Execute BM operation and mask codes.

Local Memory (LM): Scratchpad memory embedded in each PE.



Address line (Simplex Mesh), Data line (Duplex Torus)

4. Shared-memory integration in a manycore architecture

In conventional many-cores as shown in Figure 4A, there was a clear separation of computation and memory, driven by the constraints imposed by device-level wiring. However, under the assumption of FPGA implementation, integrating on-chip memory as shared memory among processing elements (PEs) becomes achievable. Consequently, we aim to explore an architecture that seamlessly combines both the physical and functional aspects of computation and memory. Our investigation is motivated by simplifying wiring complexity and mitigating wiring delays. The proposed architecture is shown in Figure 4B.

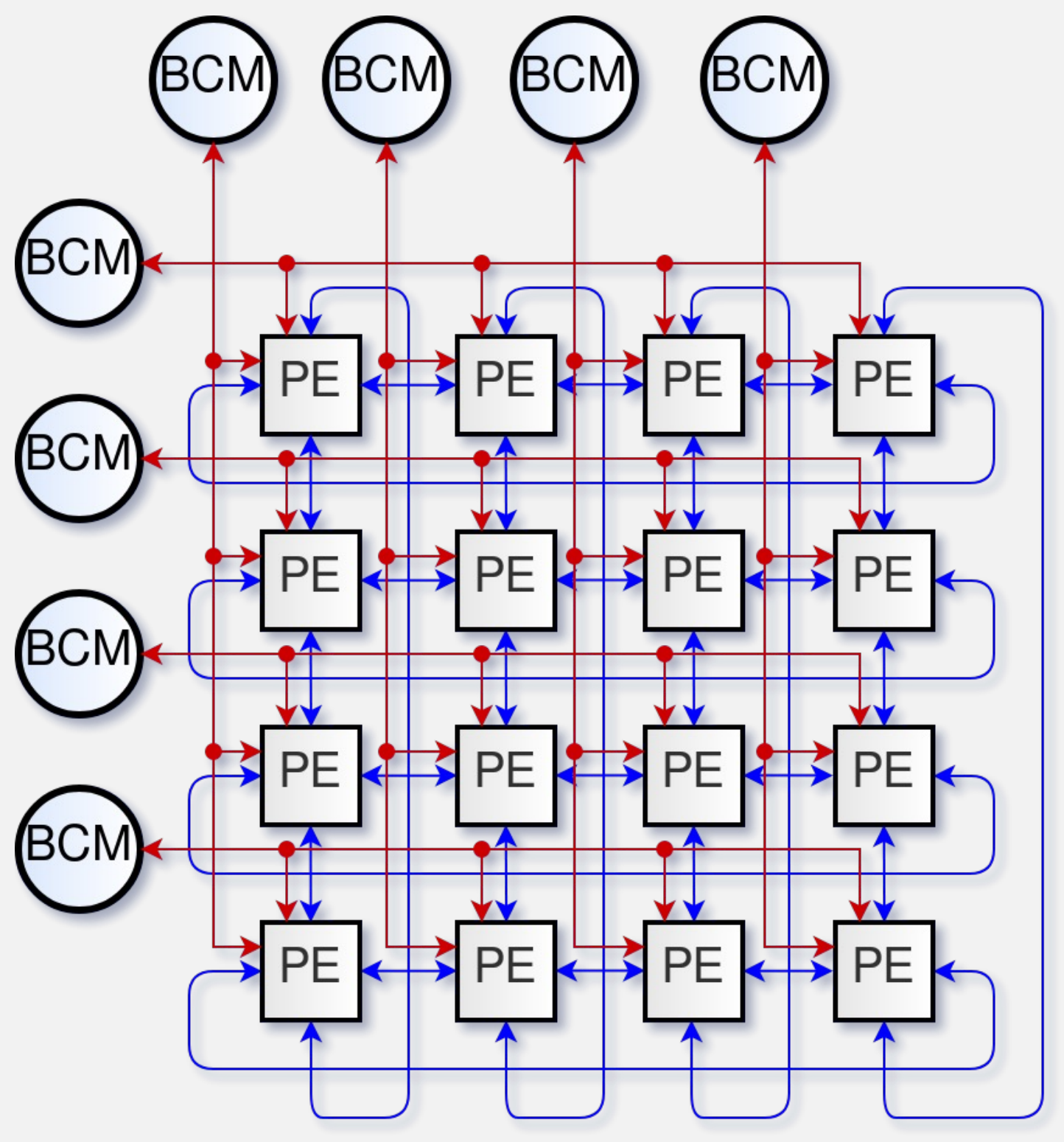


Figure 4A: Segmented architecture based on functional separation

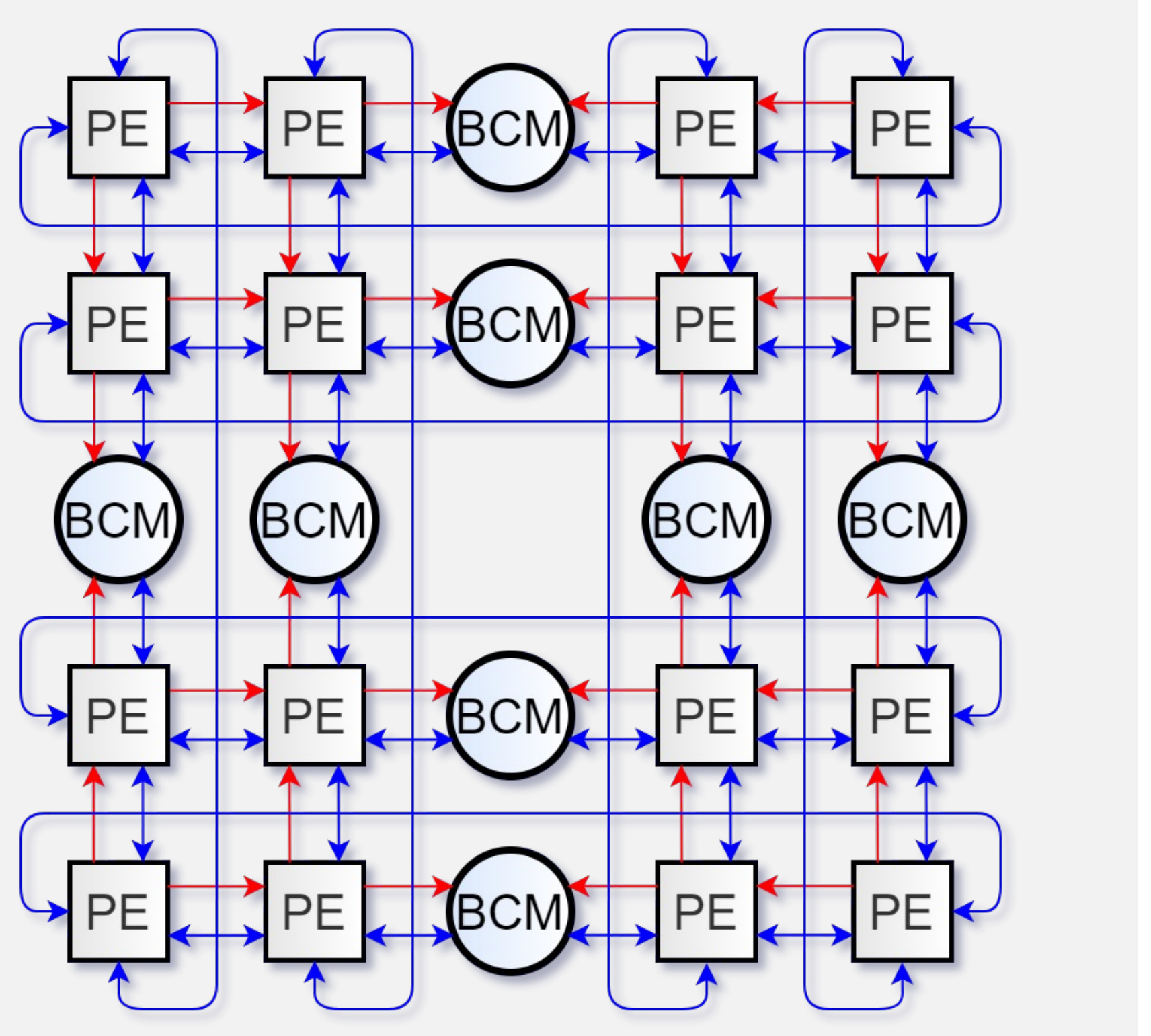
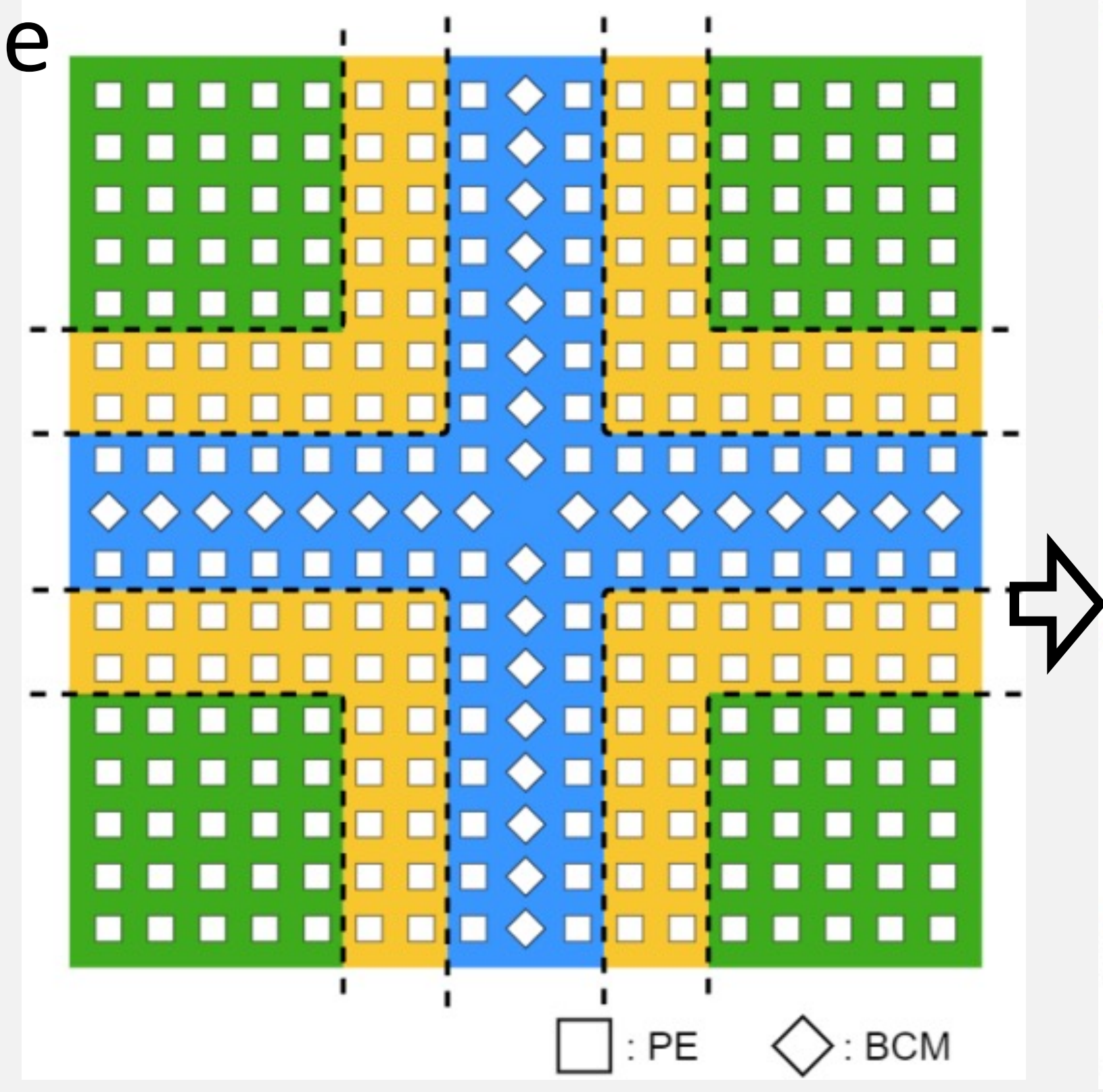


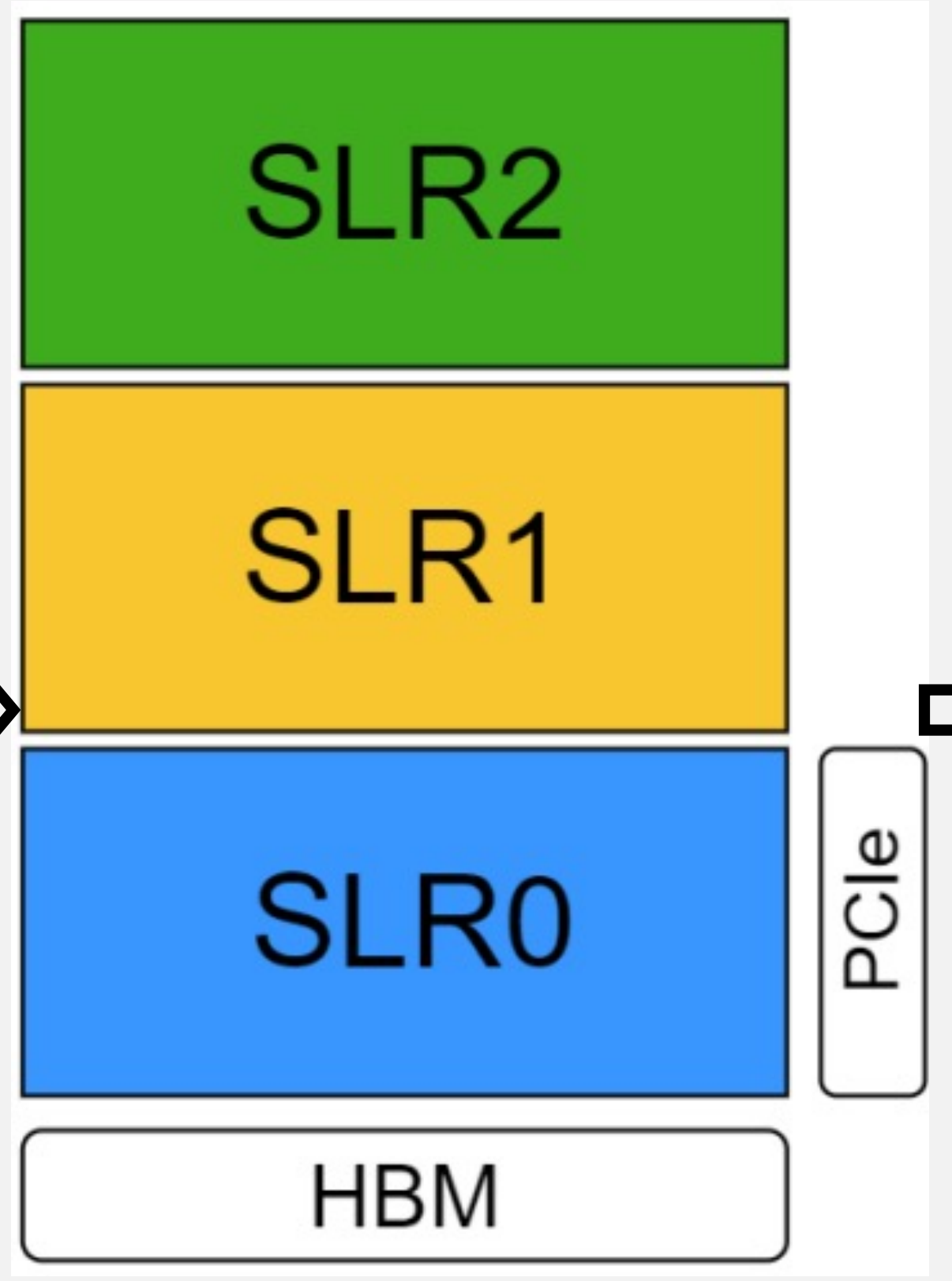
Figure 4B: Shared memory integration in a manycore architecture

5. FPGA floor plan design considering multi-die package

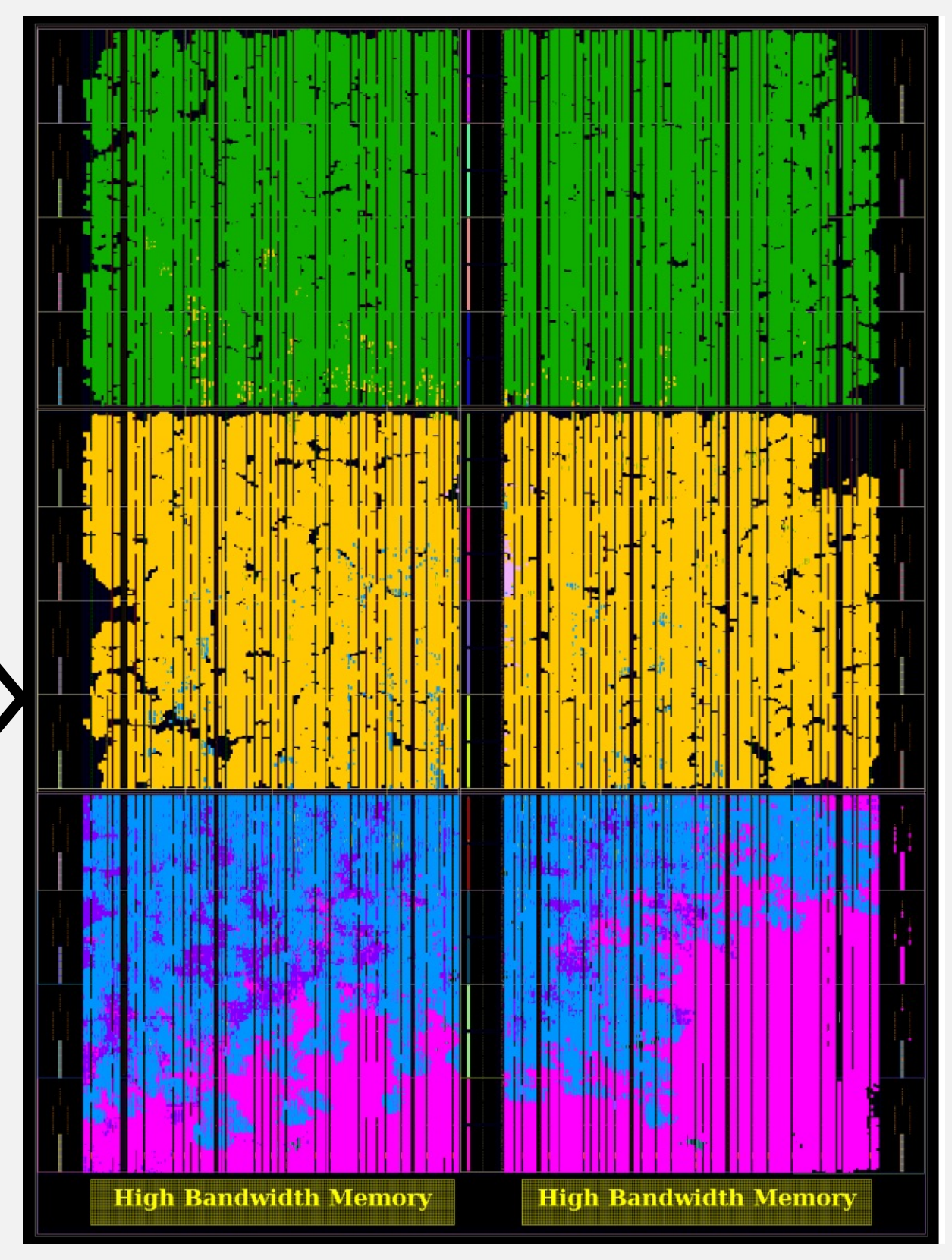
As the size of FPGAs increases, multi-die FPGAs by silicon interposers become popular. It introduces significant variations in delay times among the elementary components of FPGAs, necessitating meticulous attention to placement and routing strategies. Our research focuses on enhancing scalability and performance in large-scale FPGAs by addressing wiring complexity reduction and efficient control of inter-die connectivity. We verified our implementation using the Xilinx FPGA evaluation board, Alveo U280, which incorporates four dies comprising three Super Logic Regions (SLRs) and one High Bandwidth Memory (HBM). The accompanying diagram on the right provides an overview of our approach to placement and routing considerations for this configuration.



Interconnect Topology



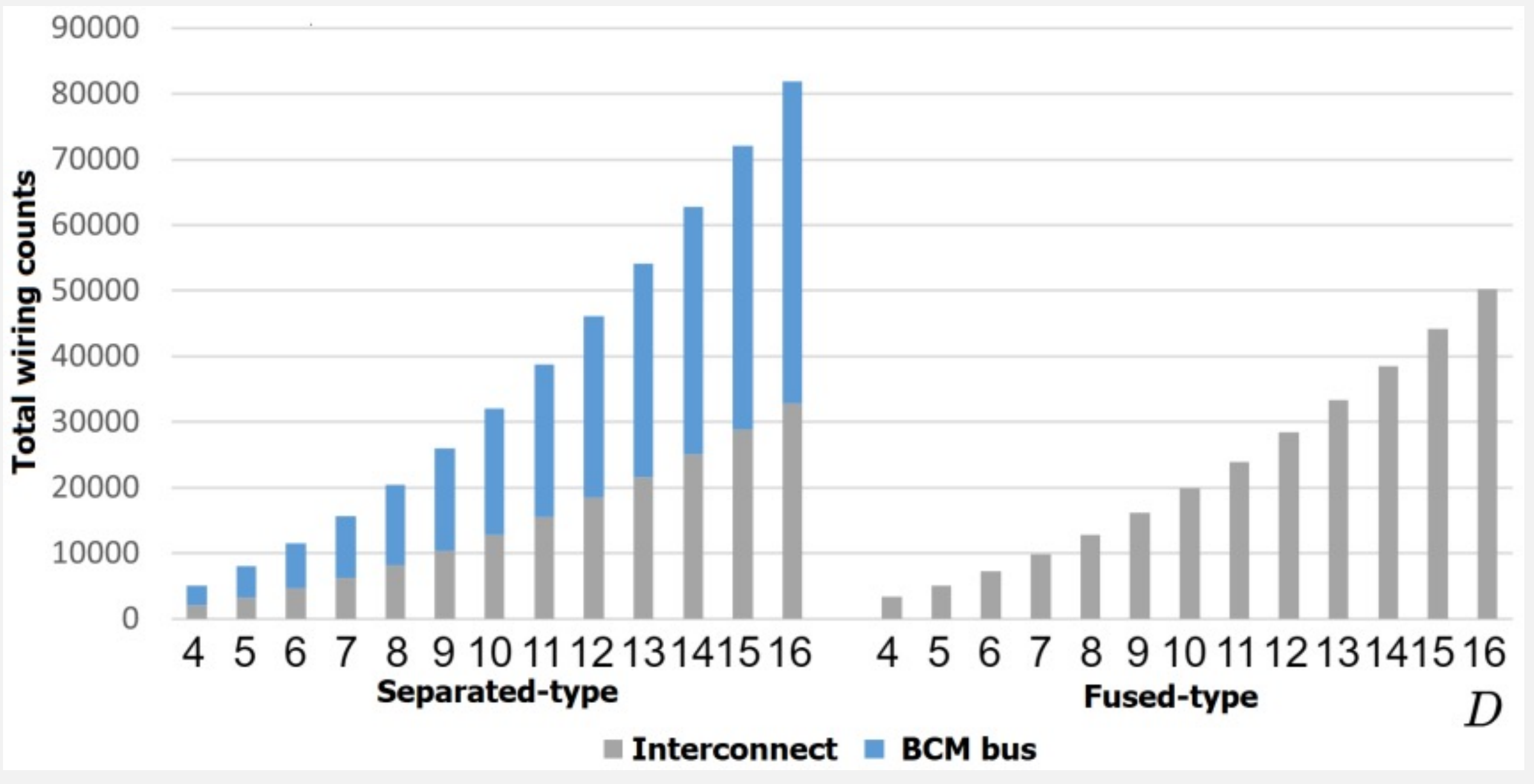
SLR layout



Place and Route

6. Diameter (D) modification impact on wiring count

Our proposed approach, called fused-type architecture, reduces the wire count dramatically compared to traditional design called segmented architecture. It achieves approximately 35% of the wiring count reduction, which leads to ease of PE placements and enhances the working frequency because of reducing the wiring latency.



Variation in wiring count resulting from diameter modification (D)

7. Clock cycle count of proposed architecture

The proposed architecture increase a larger number of data exchange instructions with BCM compared to the traditional architecture. However, it can allow direct transmission wiring and expect to eliminate additional latency.

Clock cycle counts in 16-point FFT computation

Inst. type	Separated-type[1]	Fused-type
Data-exchange (PE ⇔ BCM) inst.	22	34(155%)
Data-exchange (PE ⇔ PE) inst.	23	23(100%)
fadd, fsub, fmul	8	8(100%)
Etc.	11	11(100%)

[1] T. Yuxi et al., "A Tightly-connected RISC-V Manycore Processor in a SIMD Manner," IEICE Technical report, vol. 120, no. 36, p. 754–766, May 2020.

8. Conclusion

In conclusion, our proposed fused-type architecture, which integrates a mutual connection network with the BCM memory bus, represents a significant step forward in mitigating wiring congestion in many-core architectures. On an FPGA evaluation board, the approach has demonstrated a reduction in wiring complexity when compared to the conventional separated-type interconnect architecture. As a next step, our future endeavors will focus on the development of an automated design methodology, aiming to streamline the placement process and further enhance the efficiency of our microarchitecture.

9. Acknowledgement

This work was supported in part by MEXT as "Next-Generation High-Performance Computing Infrastructures and Applications R\&D Program" (Development of Computing-Communication Unified Supercomputer in Next Generation), also funded by TIA collaborative research program "Kakehashi" between annual 2019-2023, and by JSPS KAKENHI Grant Number JP17H01707, JP18H03246 and JP21H04869. We also thank the Xilinx University Program for the kind donation of the software tools.