



HAL
open science

Efficient Application Execution Framework for CGRAs

Christie Sajitha Sajan, Satyajit Das, Kevin Martin, Philippe Coussy

► **To cite this version:**

Christie Sajitha Sajan, Satyajit Das, Kevin Martin, Philippe Coussy. Efficient Application Execution Framework for CGRAs. Colloque du GDR SOC2, Jun 2024, Toulouse (FRANCE), France. hal-04651635

HAL Id: hal-04651635

<https://hal.science/hal-04651635v1>

Submitted on 17 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

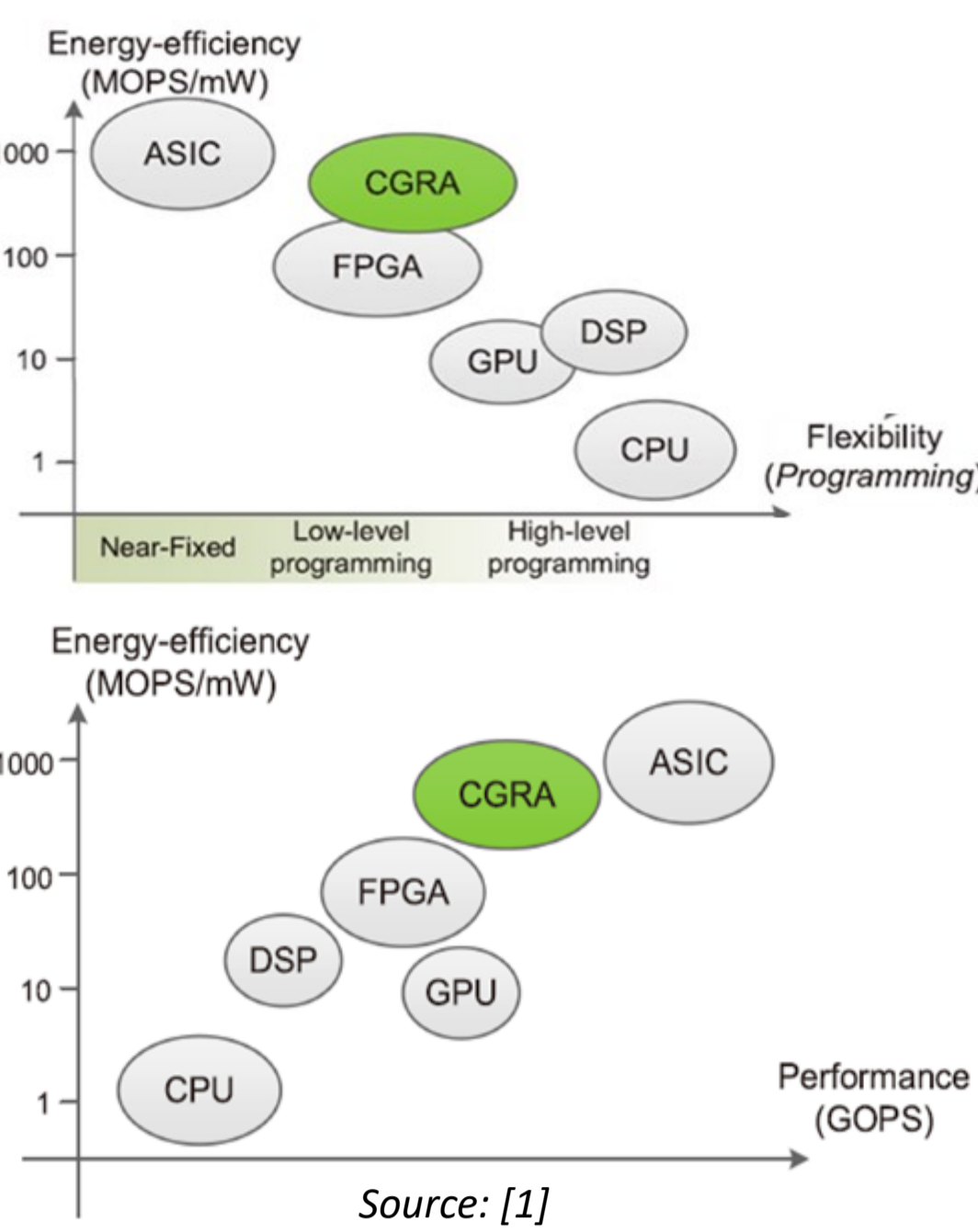
Coarse-Grained Reconfigurable Array (CGRA)

- Array of interconnected processing elements
- Combines both temporal and spatial computation

- Examples**
- **Commercial**
 - Sambanova Datascale
 - Intel CSA
 - Cerebras Wafer Scale
 - **Academic**
 - IPA
 - HyCUBE
 - SNAFU

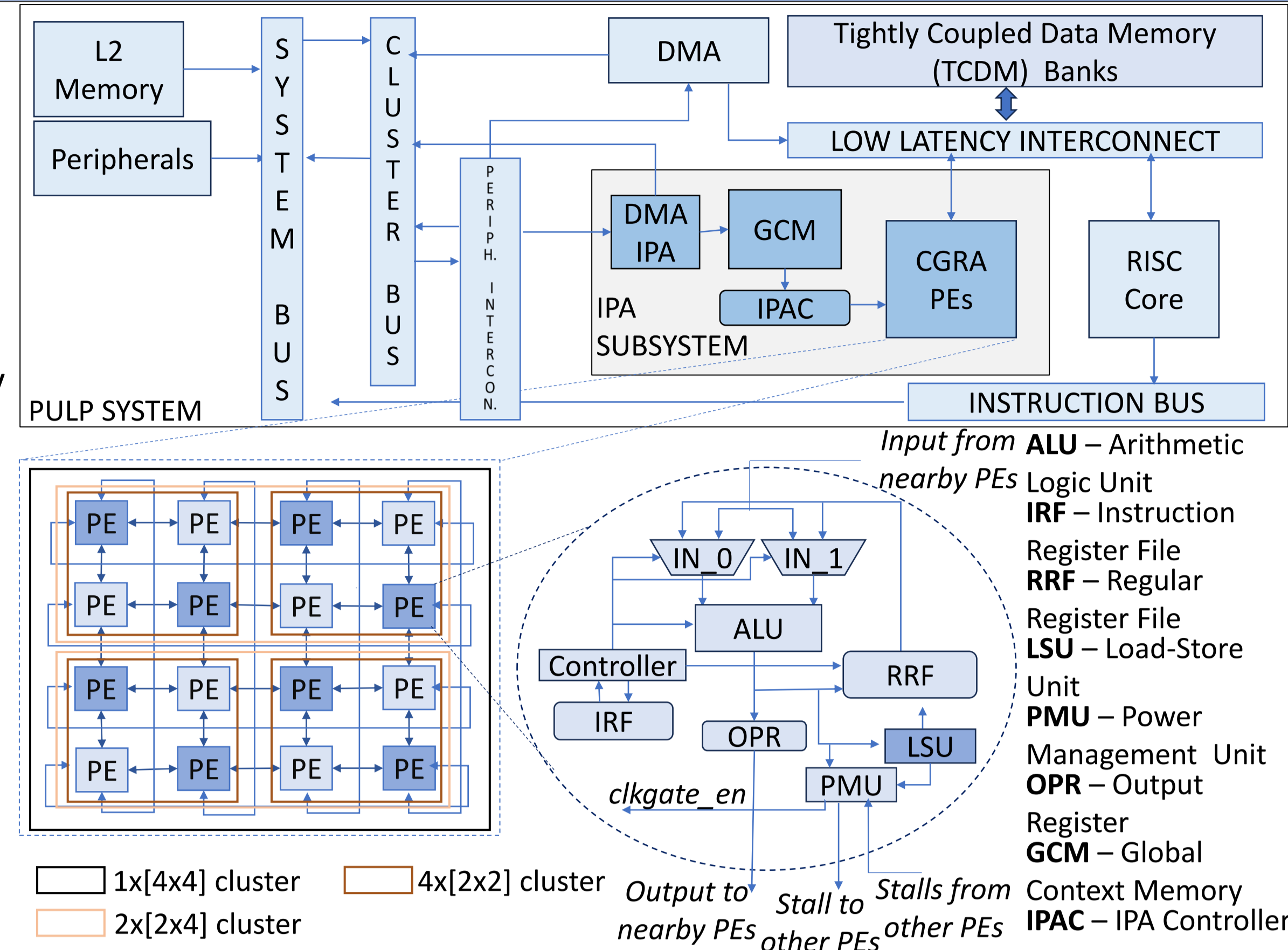
- Why CGRA?**
- Lesser power consumption than CPU, GPU, FPGA and DSP
 - Highly flexible compared to ASIC
 - Lesser reconfigurable time than FPGA

- Applications**
- Signal Processing - image, video, speech
 - Security - cryptography
 - Artificial Intelligence - PCA, CNN, Transformers



State-of-the-Art CGRA - Integrated Programmable Array (IPA)

- Focus**
- Ultra-low power
- Interconnect**
- Mesh-torus interconnect - Data
 - Bus based network - Instruction
 - Logarithmic Interconnect - Memory
- Processing Elements (PEs)**
- 32 bits ALU
 - 8x32 Regular Register File
 - 64x20 Instruction Register File
- Cluster-based CGRA**
- PEs grouped as PE clusters
 - Clusters work independently



Modulo Scheduling (MS)

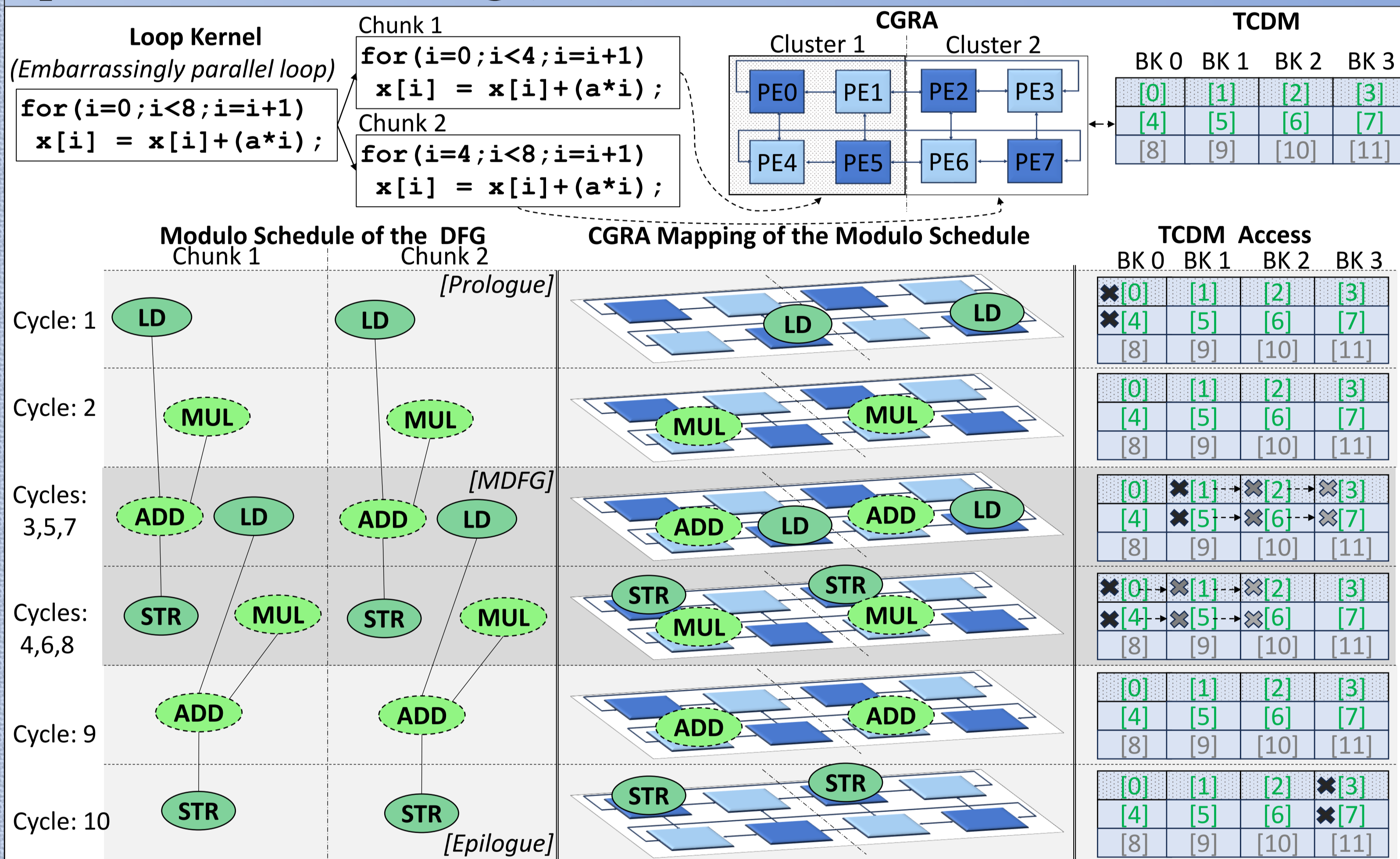
- Commonly used software pipelining technique to accelerate loops on CGRA
- Achieves Instruction Level Parallelism (ILP)
- **Objective:**
 - Generate schedule for one iteration of the loop
 - This same schedule is repeated at regular interval without violating dependency constraints

Motivation

- State-of-the-art technique: Unrolling + MS [3]
- Disadvantages of unrolling + MS on CGRAs : Difficulty in finding mapping solution
- Iterations in embarrassingly parallel loops can be executed in parallel

How to exploit Data Level Parallelism in embarrassingly parallel loops?

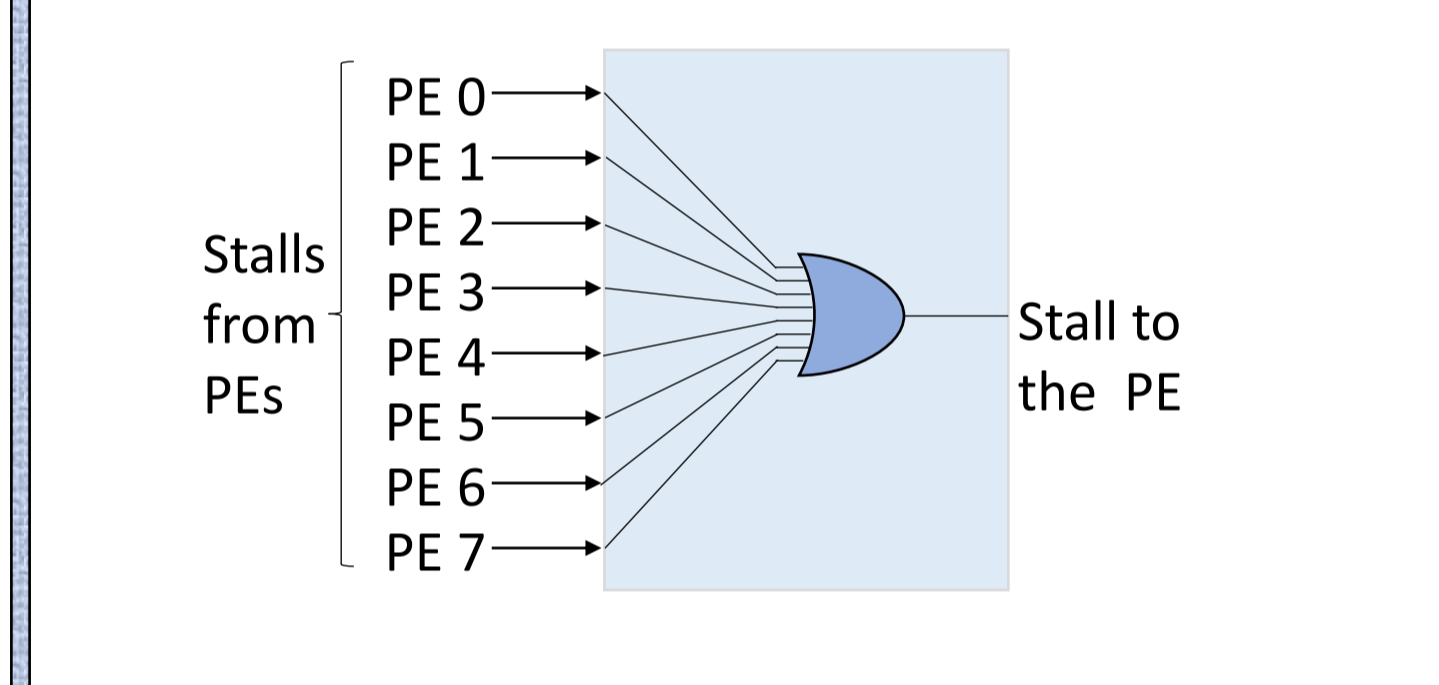
Split Modulo Scheduling (SMS)



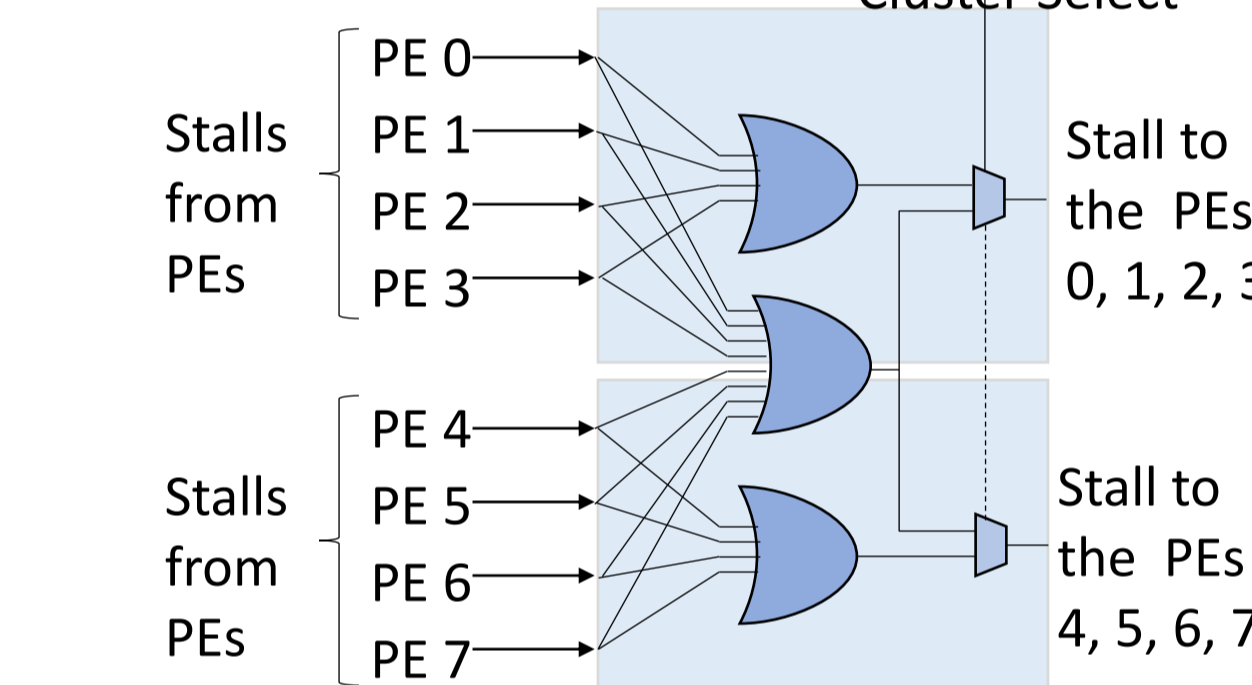
Cluster-Based Execution

- **Global Freeze Mechanism (GFM)** maintains the schedule across all PEs
- **Cluster-level Freeze Mechanism (CFM)** maintains the schedule within a PE cluster

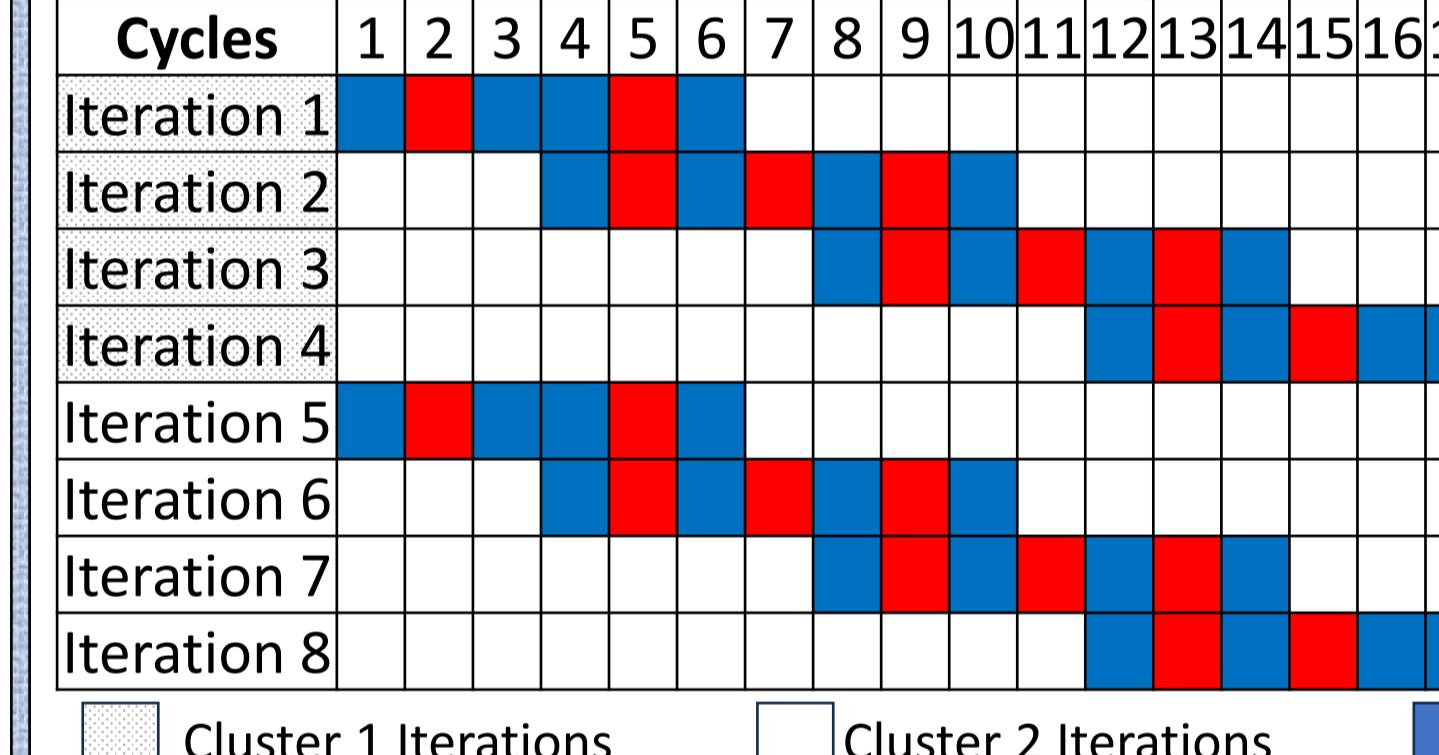
GFM of 2x4 CGRA



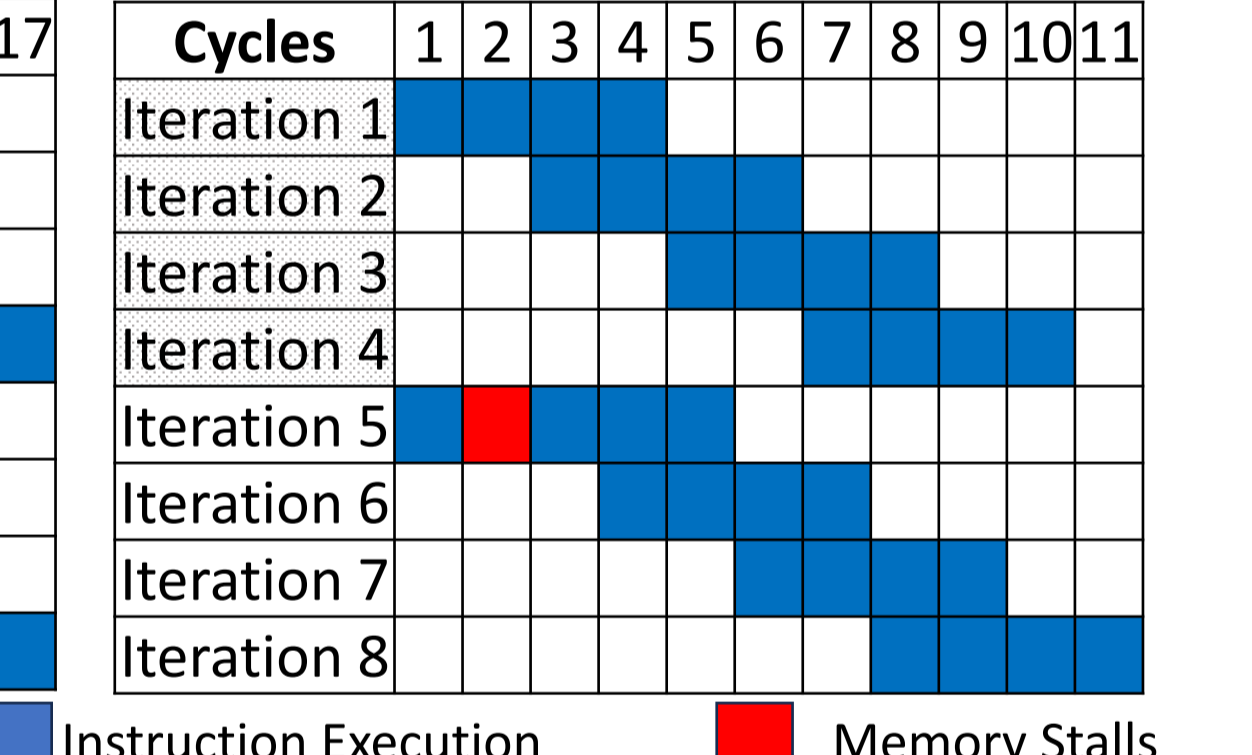
CFM of 2x4 CGRA



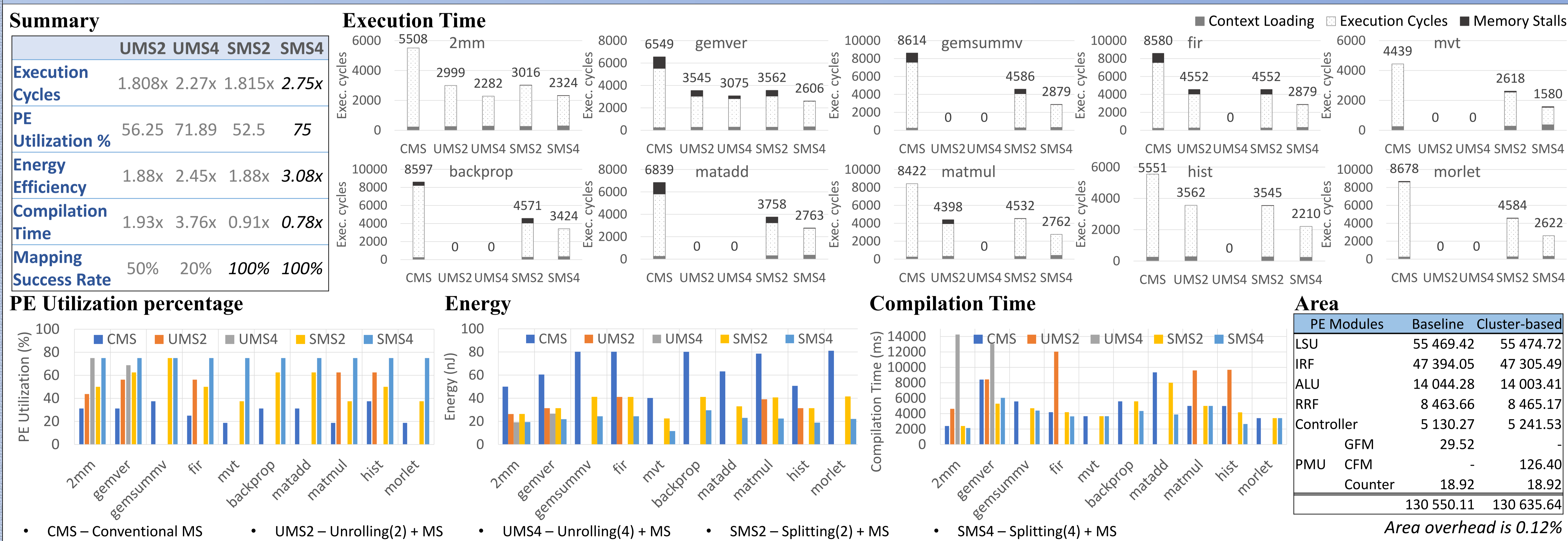
Execution of 2 chunks with GFM



Execution of 2 chunks with CFM



Results



Conclusion

- Accelerate loop chunks that can be executed in parallel
- Mapping becomes easier in Loop Splitting + MS than Loop Unrolling + MS
- Better PE Utilization
- Better Energy Efficiency
- Better Execution Efficiency
- Better Compilation time
- Highly Scalable

References

- [1] Leibo Liu, Jianfeng Zhu, Zhaoshi Li, Yanan Lu, Yangdong Deng, Jie Han, Shouyi Yin, and Shaojun Wei. 2019. A Survey of Coarse-Grained Reconfigurable Architecture and Design: Taxonomy, Challenges, and Applications. ACM Comput. Surv.
- [2] Das Satyajit, Kevin JM Martin, Davide Rossi, Philippe Coussy, and Luca Benini. "An energy-efficient integrated programmable array accelerator and compilation flow for near-sensor ultralow power processing." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems
- [3] Balasubramanian, Mahesh, and Aviral Shrivastava. "CRIMSON: Compute-intensive loop acceleration by randomized iterative modulo scheduling and optimized mapping on CGRAs." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 39, no. 11 (2020).
- [4] B. R. Rau, "Iterative modulo scheduling: An algorithm for software pipelining loops," in Proceedings of the 27th annual international symposium on Microarchitecture. ACM, 1994.