OpenGeMM: A High-Utilization GeMM Accelerator Generator with Lightweight RISC-V Control and Tight Memory Coupling

ASP-DAC 2025

Xiaoling Yi[1], Ryan Antonio[1], Joren Dumoulin[1], Jiacong Sun[1], Josse Van Delm[1], Guilherme Paim[1,2], Marian Verhelst[1]
[1]MICAS-ESAT, KU Leuven, Belgium,
[2]INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal

23rd January 2025

Outline

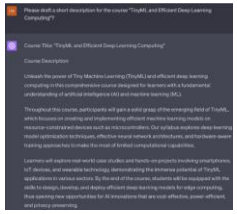
- Edge AI Computing Background and Motivation
- OpenGeMM System Architecture
  - Overview
  - GeMM Accelerator Generator
  - Mechanisms for High Utilization
  - Reusability and Flexibility Summary
- Experimental Results and SotA Comparison
- Conclusion and Future Work

Edge AI Computing - Necessity

- DNN models become pervasive while evolving rapidly

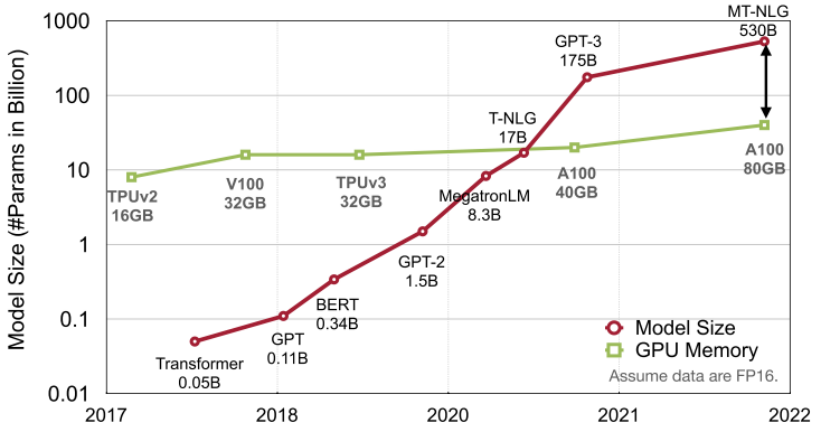
Image Classification

Video Generation



Language Assistance

Intelligent Robotics



Model size of language models

Edge AI Computing - Necessity

- DNN models become pervasive while evolving rapidly
- Edge DNN deployment challenges
  - 1) High performance and energy efficiency
    - Real-time application
    - Low battery capacity

Edge AI Computing - Necessity

- DNN models become pervasive while evolving rapidly
- Edge DNN deployment challenges
  - 1) High performance and energy efficiency
  - 2) Flexibility
    - Reusable across DNN models

Edge AI Computing - Necessity

- DNN models become pervasive while evolving rapidly
- Edge DNN deployment challenges
  - 1) High performance and energy efficiency
  - 2) Flexibility
  - 3) Underutilization
    - Low effective computation

Edge AI Computing – SotA Works

- Efficiency vs. Flexibility/Reusability

- Higher efficiency
- Tailored to specific workloads
- Limited reusability and programmability

- Higher flexibility
- Reusable for diverse workloads
- Low efficiency and high control overhead

Edge AI Computing – SotA Works

- Efficiency vs. Flexibility/Reusability

RISC-V AI platforms:

Gemmini [2], RedMule [3]

- Flexible GeMM accelerator
- Programmability with RISC-V
- High control overhead
- Low utilization

Edge AI Computing – SotA Works

- Efficiency vs. Flexibility/Reusability

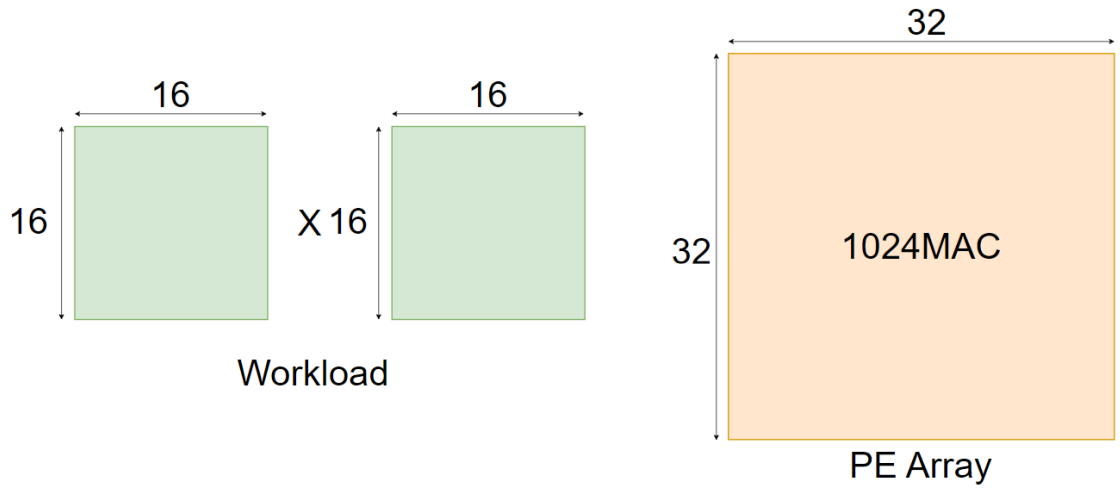
- Flexible GeMM accelerator
- Programmability with RISC-V
- Low control overhead
- High utilization

Edge AI Computing – Utilization Issue

- PE array underutilization – performance killer
  - Spatial underutilization
  - Temporal underutilization

Edge AI Computing – Utilization Issue

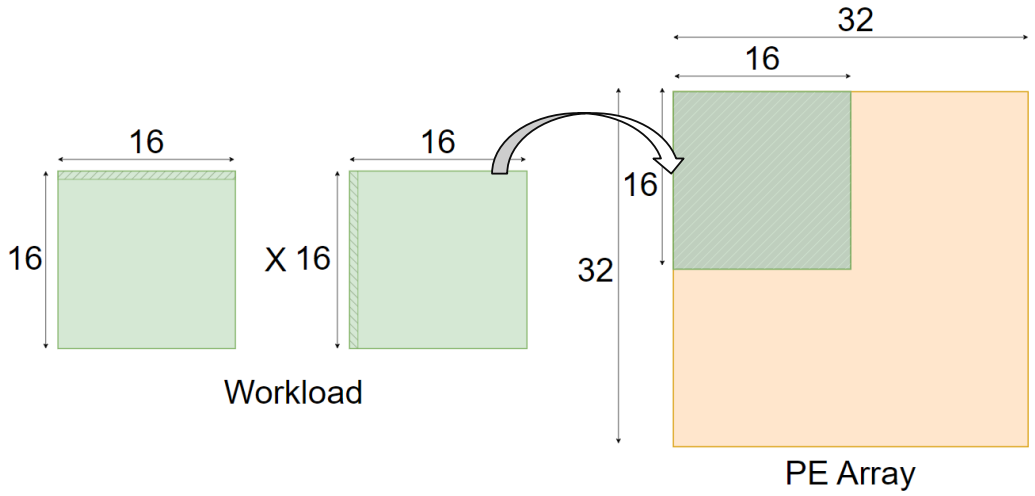
- Performacne killer – PE array underutilization
  - Spatial underutilization
    - Insufficient algorithm parallelism



  - Temporal underutilization

Edge AI Computing – Utilization Issue

- Performacne killer – PE array underutilization
  - Spatial underutilization
    - Insufficient algorithm parallelism



Utilization = 25%

Underutilization = 75%

  - Temporal underutilization

Edge AI Computing – Utilization Issue

- Performacne killer – PE array underutilization
  - Spatial underutilization
    - Insufficient algorithm parallelism
  - Temporal underutilization
    - Long configuration time
    - Memory stall

Edge AI Computing – Utilization Issue

- Performacne killer – PE array underutilization
  - Spatial underutilization
    - Insufficient algorithm parallelism
  - Temporal underutilization
    - Long configuration time
    - Memory stall

Edge AI Computing – Utilization Issue

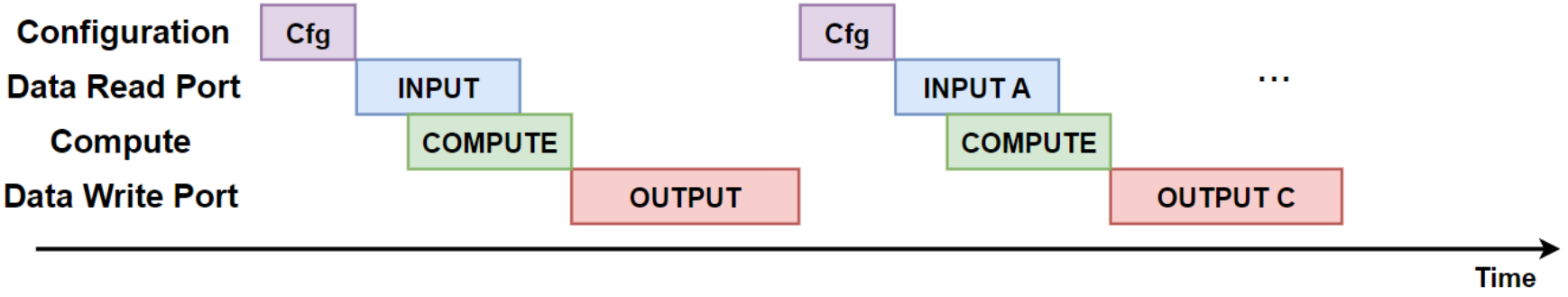
- Performacne killer – PE array underutilization
  - Spatial underutilization
    - Insufficient algorithm parallelism
  - Temporal underutilization
    - Long configuration time
    - Memory stall

Edge AI Computing – Utilization Issue

- Performacne killer – PE array underutilization
  - Spatial underutilization
    - Insufficient algorithm parallelism
  - Temporal underutilization
    - Long configuration time
    - Memory stall

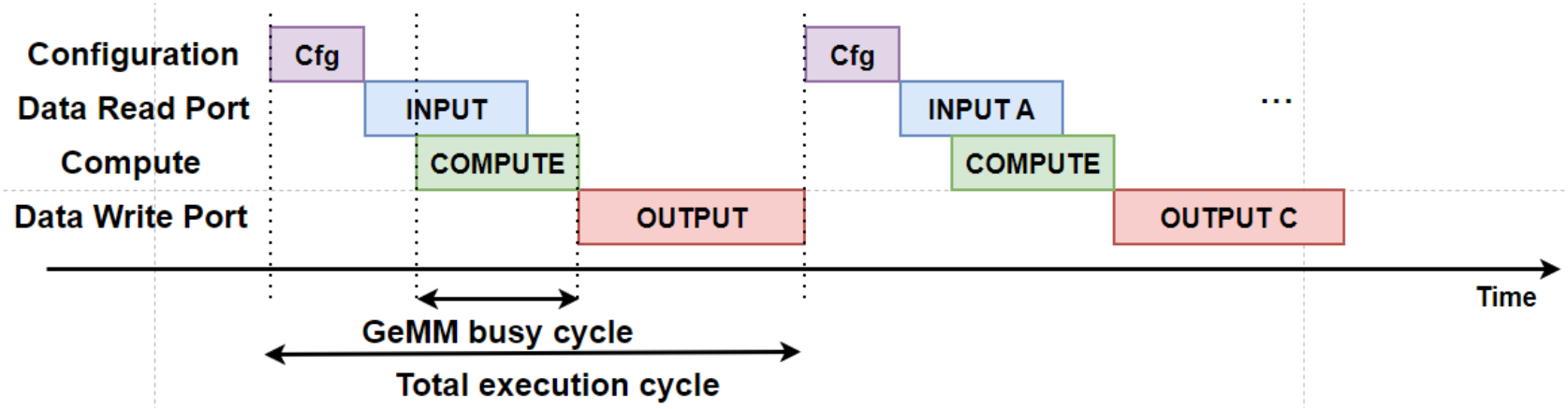
Edge AI Computing – Utilization Issue

- Performacne killer – PE array underutilization
  - Spatial underutilization
    - Insufficient algorithm parallelism
  - Temporal underutilization
    - Long configuration time
    - Memory stall



Edge AI Computing – Utilization Issue

- Performacne killer – PE array underutilization
  - Spatial underutilization
    - Insufficient algorithm parallelism
  - Temporal underutilization
    - Long configuration time
    - Memory stall

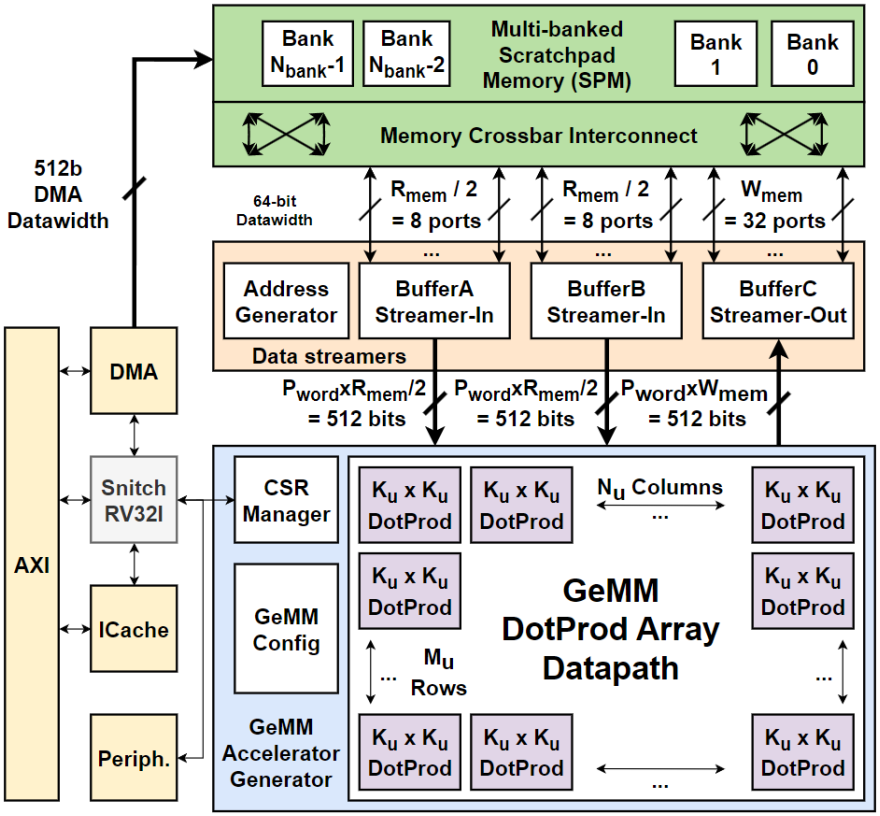


Outline

- Edge AI Computing Background and Motivation
- OpenGeMM System Architecture
  - Overview
  - GeMM Accelerator Generator
  - Mechanisms for High Utilization
  - Reusability and Flexibility Summary
- Experimental Results and SotA Comparison
- Conclusion and Future Work

Proposed OpenGeMM Overall

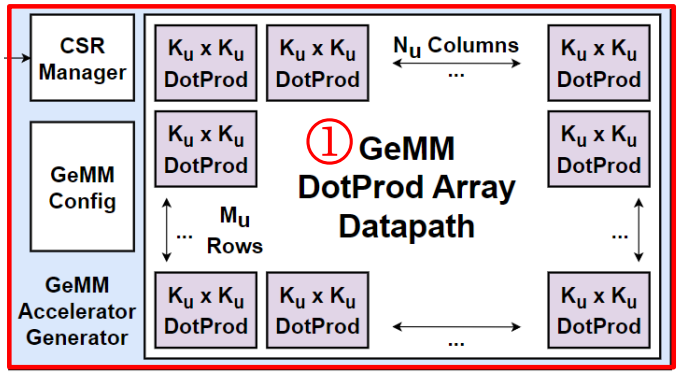
- An open-source GeMM acceleration platform
  - Perfectly balance efficiency and flexibility



OpenGeMM Arch. Overview

Proposed OpenGeMM Overall

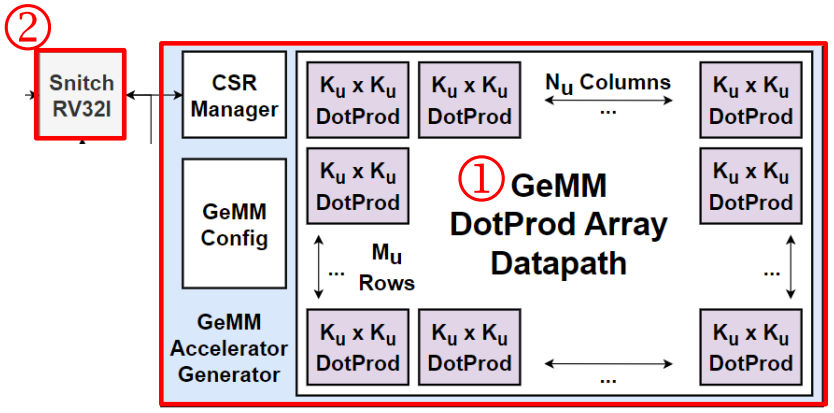
- An open-source GeMM acceleration platform
  - ① Programmable GeMM hardware generator
    - 3D spatial unrolling for high utilization
    - Design-time and run time flexibility



OpenGeMM Arch. Overview

Proposed OpenGeMM Overall

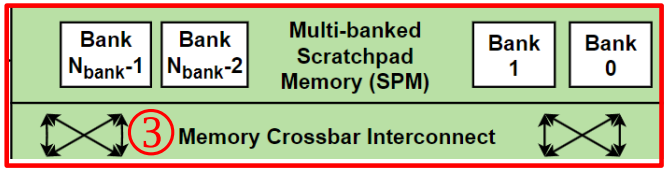
- An open-source GeMM acceleration platform
  - ① Programmable GeMM hardware generator
  - ②A lightweight RISC-V host processor
    - Configuration and Status Registers (CSR) instructions for Acc. Ctrl.
    - High configuration bandwidth (32 bits/cycle)

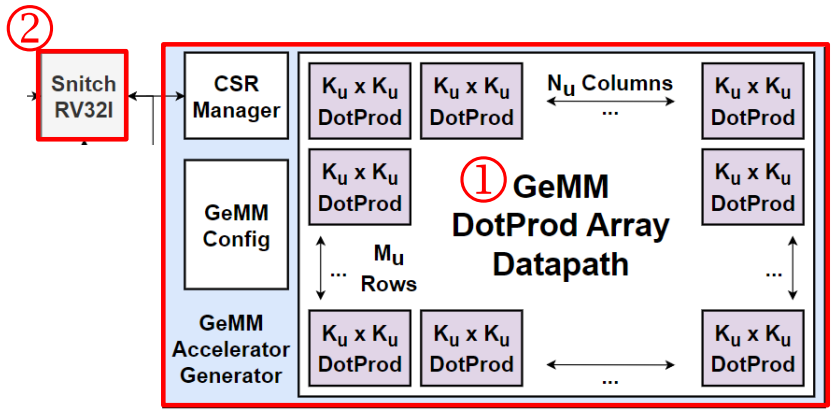


OpenGeMM Arch. Overview

Proposed OpenGeMM Overall

- An open-source GeMM acceleration platform
  - ① Programmable GeMM hardware generator
  - ②A lightweight RISC-V host processor
  - ③A tightly coupled memory subsystem

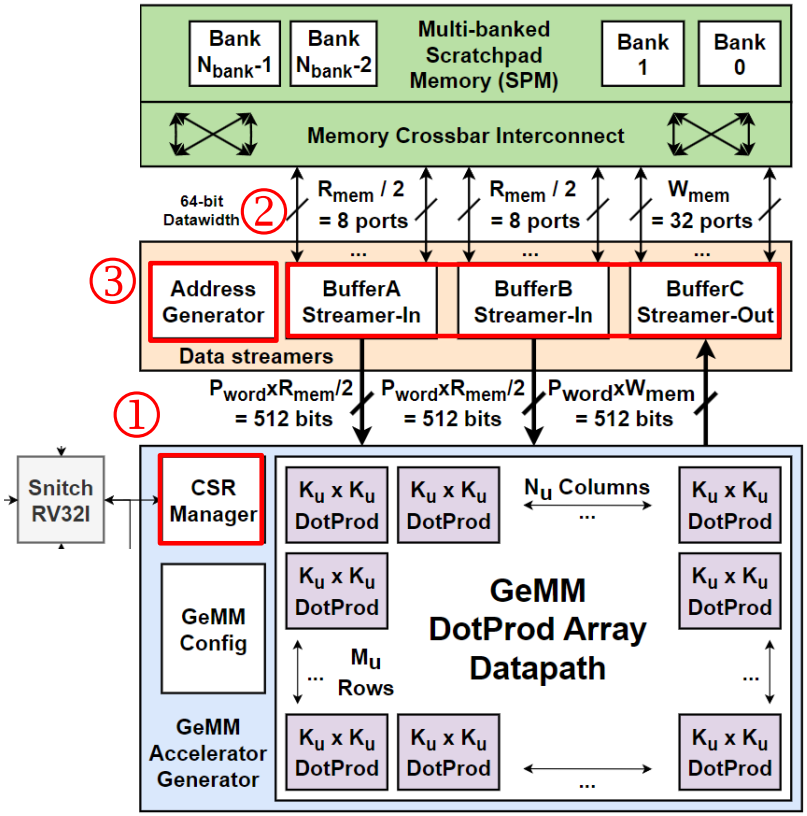




OpenGeMM Arch. Overview

Proposed OpenGeMM Overall

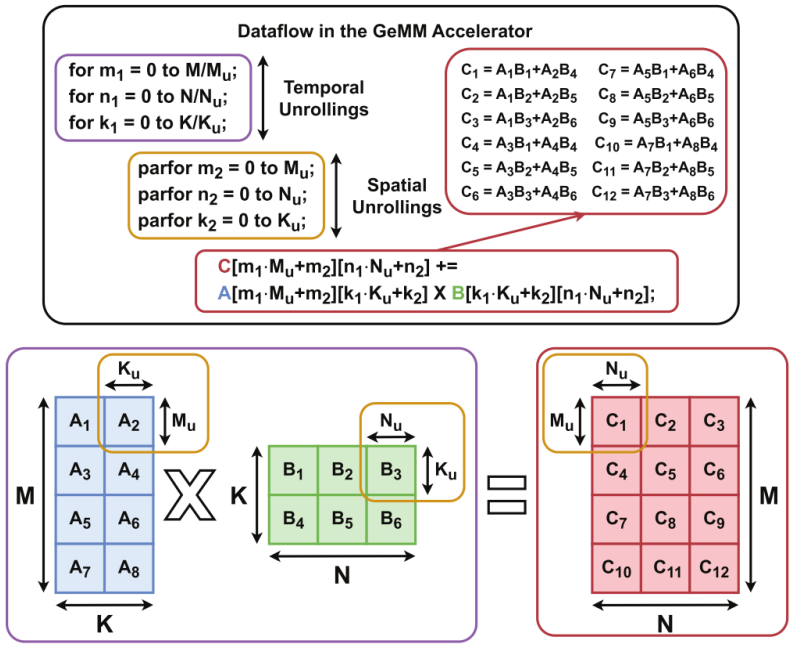
- Three mechanisms for high temporal utilization
  - ① Configuration pre-loading
  - ②Input pre-fetching with output buffering
  - ③Programmable strided memory access



OpenGeMM Arch. Overview

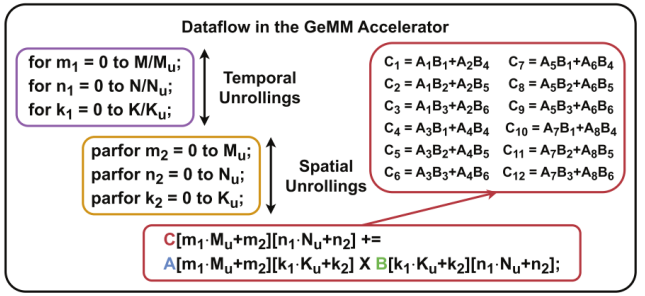
GeMM Accelerator Dataflow

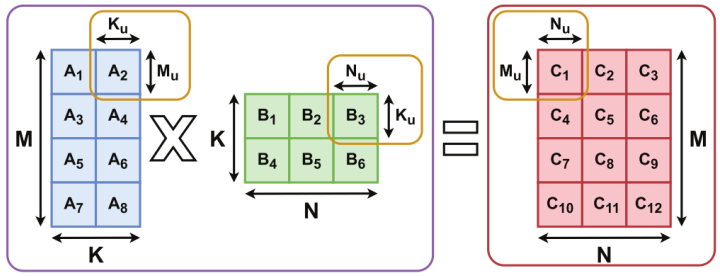
- Blocked GeMM acceleration
  - 3D spatial unrolling (SU)
  - 3D temporal unrolling (TU)

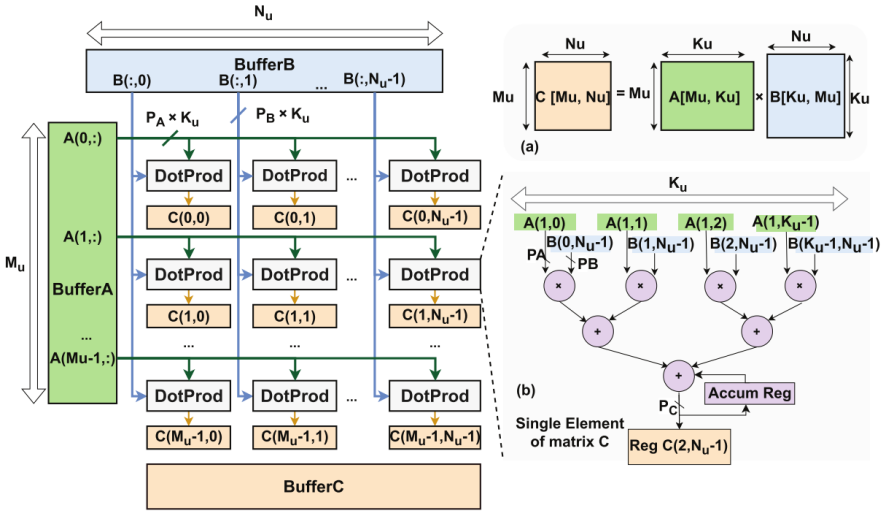


GeMM Accelerator Architecture

- Blocked GeMM acceleration
  - 3D spatial unrolling (SU)
  - 3D temporal unrolling (TU)
- 3D MAC array
  - (Mu, Nu)-sized mesh of Ku-sized vector dot product units (DotProd)
  - Spatially reuse each row of A and each column of B
  - Spatially reduction of partial sum
  - Design time configurable
    - (Mu, Ku, Nu)
    - Data precision

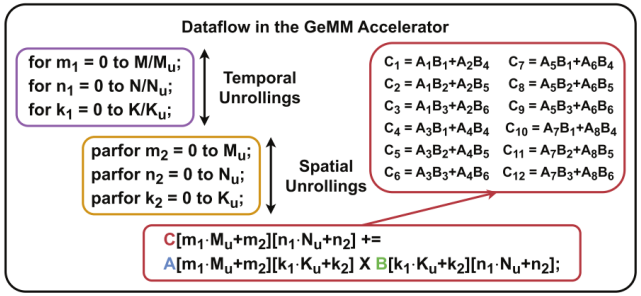


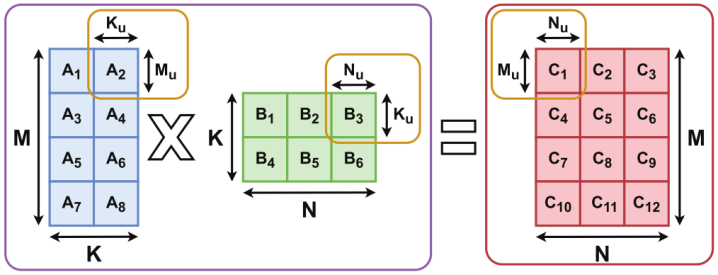


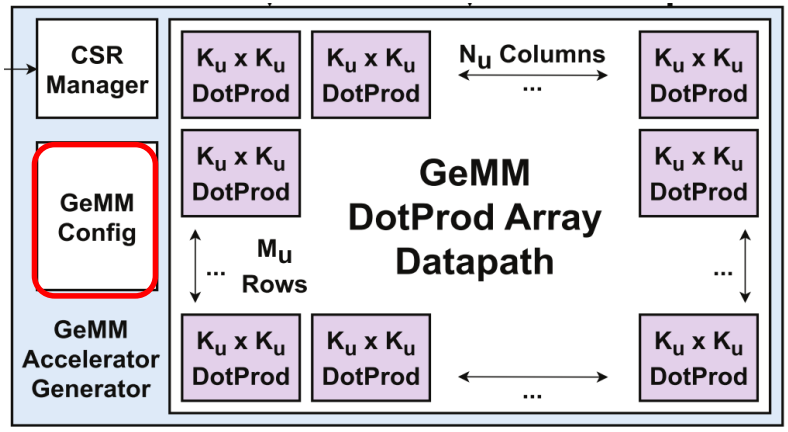


GeMM Accelerator Architecture

- Blocked GeMM acceleration
  - 3D spatial unrolling (SU)
  - 3D temporal unrolling (TU)
- 3D MAC array
  - (Mu,Nu)-sized mesh of Ku-sized vector dot product units (DotProd)
  - Design time configurable
    - (Mu,Ku,Nu)
    - Data precision
- Output-stationary
  - Temporally store partial sum
- Progammable loop bounds
  - M/Mu, N/Nu, K/Ku

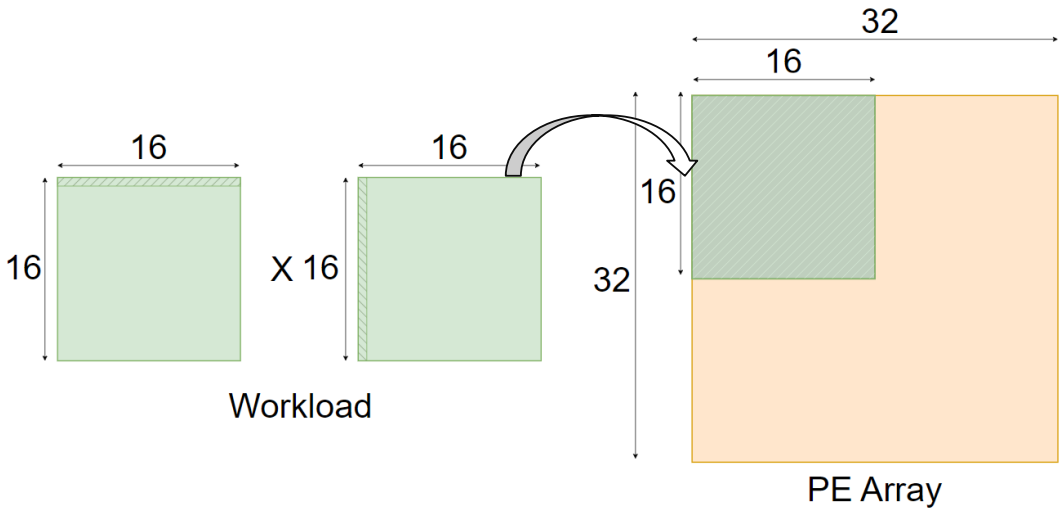






3D SU for High Spatial Utilization

- 3D SU balances unrolling of each dimension
- 16x16 matrix mapped on 2D 32x32 (1024 MAC) PE array

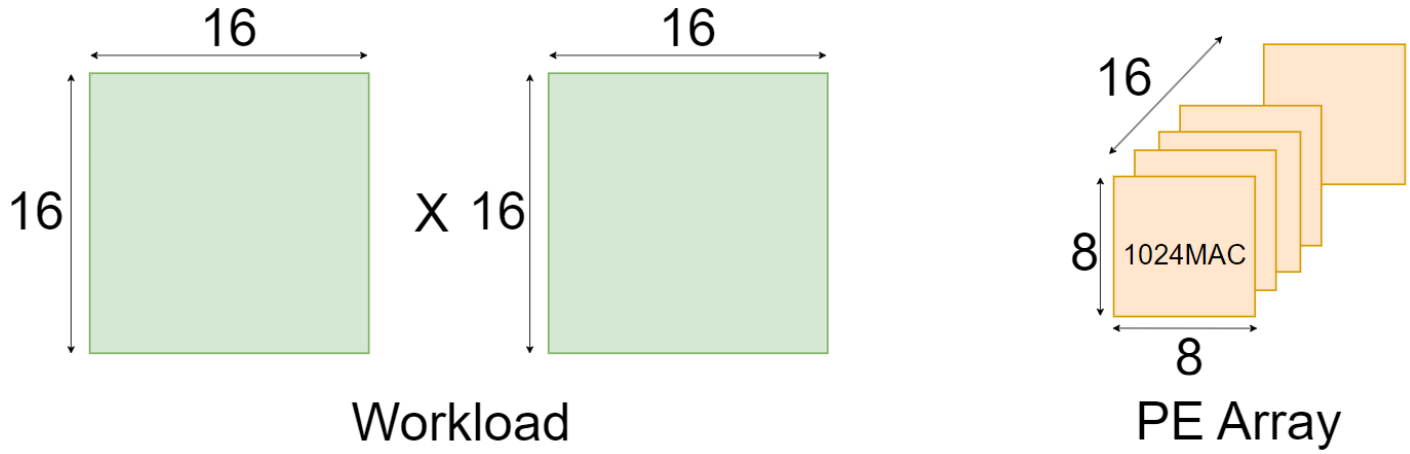


Utilization = 25%

Underutilization = 75%

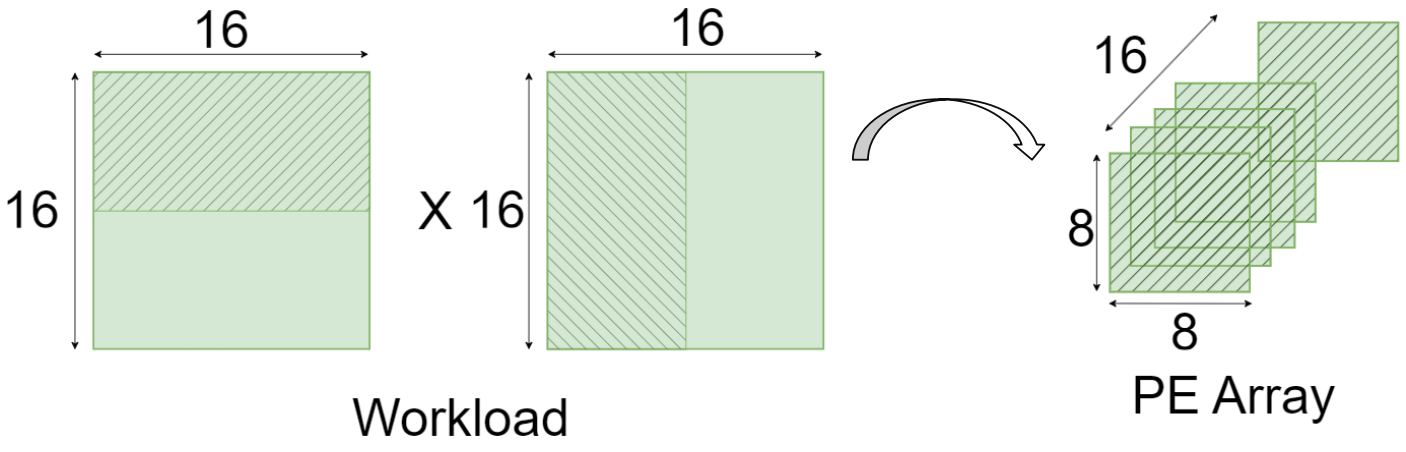
3D SU for High Spatial Utilization

- 3D SU balances unrolling of each dimension
- 16x16 matrix mapped on 2D 32x32 (1024 MAC) PE array: utilization = 25%
- 16x16 matrix mapped on 3D 8x8x16 (1024 MAC) PE array



3D SU for High Spatial Utilization

- 3D SU balances unrolling of each dimension
- 16x16 matrix mapped on 2D 32x32 (1024 MAC) PE array: utilization = 25%
- 16x16 matrix mapped on 3D 8x8x16 (1024 MAC) PE array: utilization = 100%

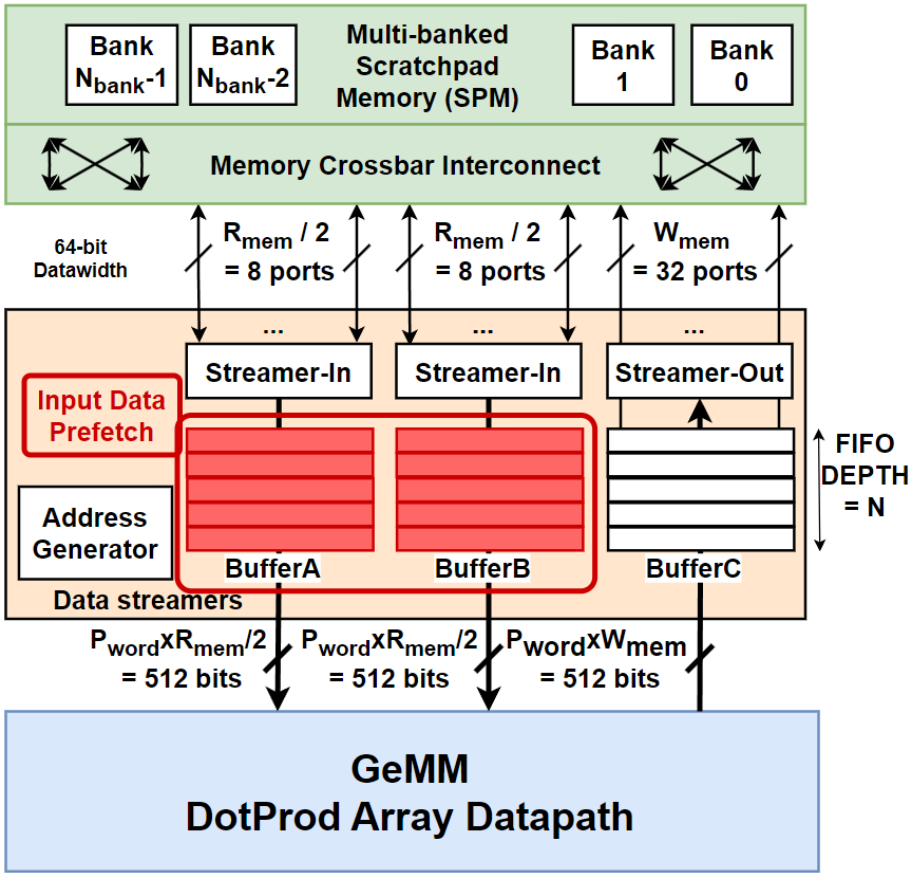


Mechanisms for High Temporal Utilization

- (1) Configuration Pre-loading at control panel
  - Double buffered configuration
    - Overlap configuration and computation

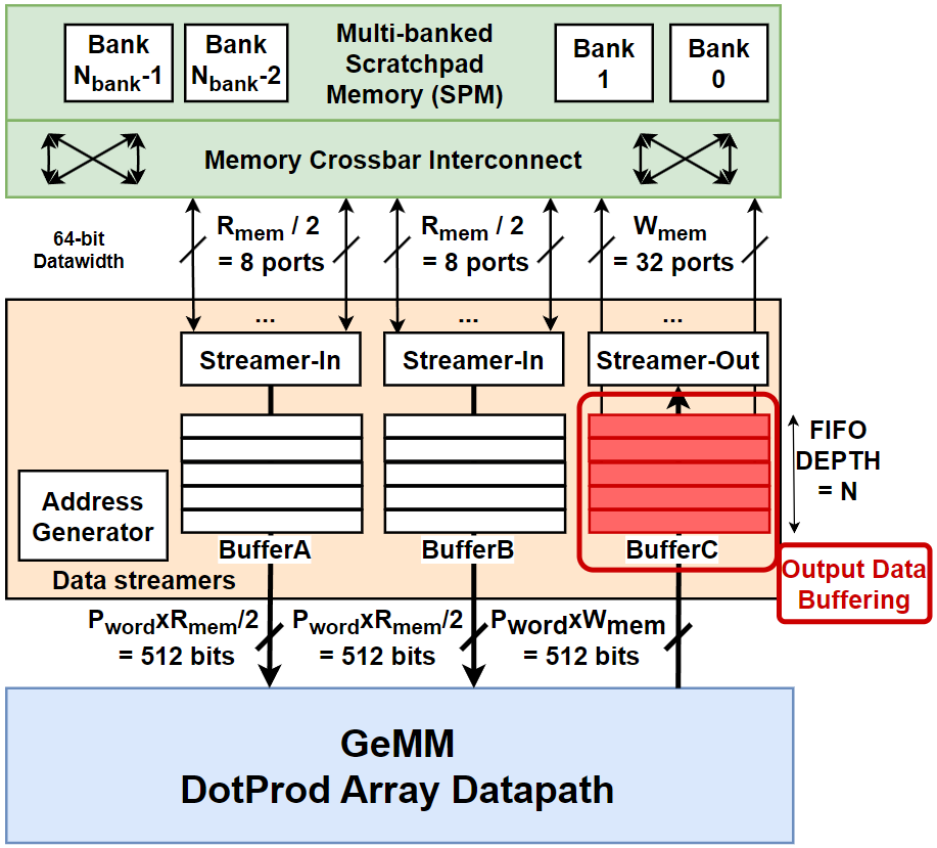
Mechanisms for High Temporal Utilization

- (1) Configuration Pre-loading at control panel
  - Double buffered configuration
    - Overlap configuration and computation
- (2) Data Pre-fetch and Buffering
  - Input data prefetch



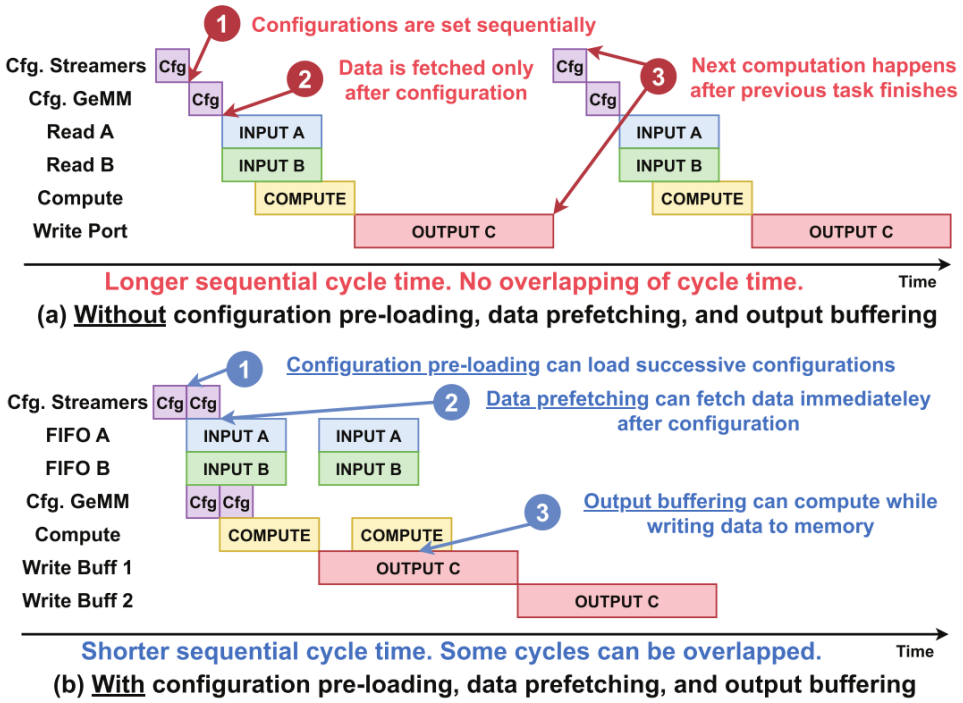
Mechanisms for High Temporal Utilization

- (1) Configuration Pre-loading at control panel
  - Double buffered configuration
    - Overlap configuration and computation
- (2) Data Pre-fetch and Buffering
  - Input data prefetch
  - Output data buffering



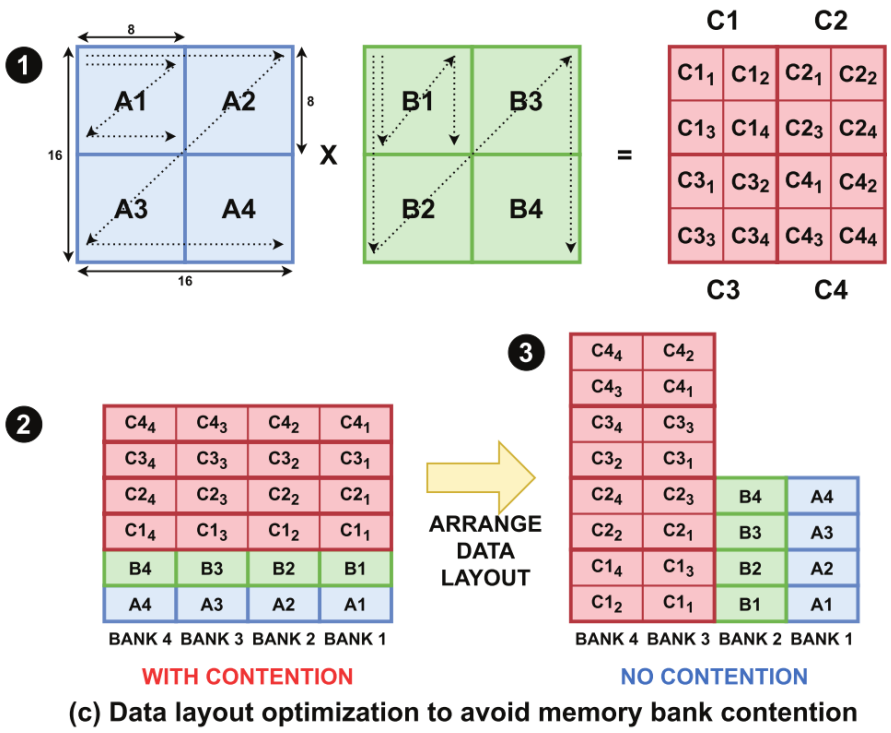
Mechanisms for High Temporal Utilization

- (1) Configuration Pre-loading at control panel
  - Double buffered configuration
    - Overlap configuration and computation
- (2) Data Pre-fetch and Buffering
  - Input data prefetch
  - Output data buffering



Mechanisms for High Temporal Utilization

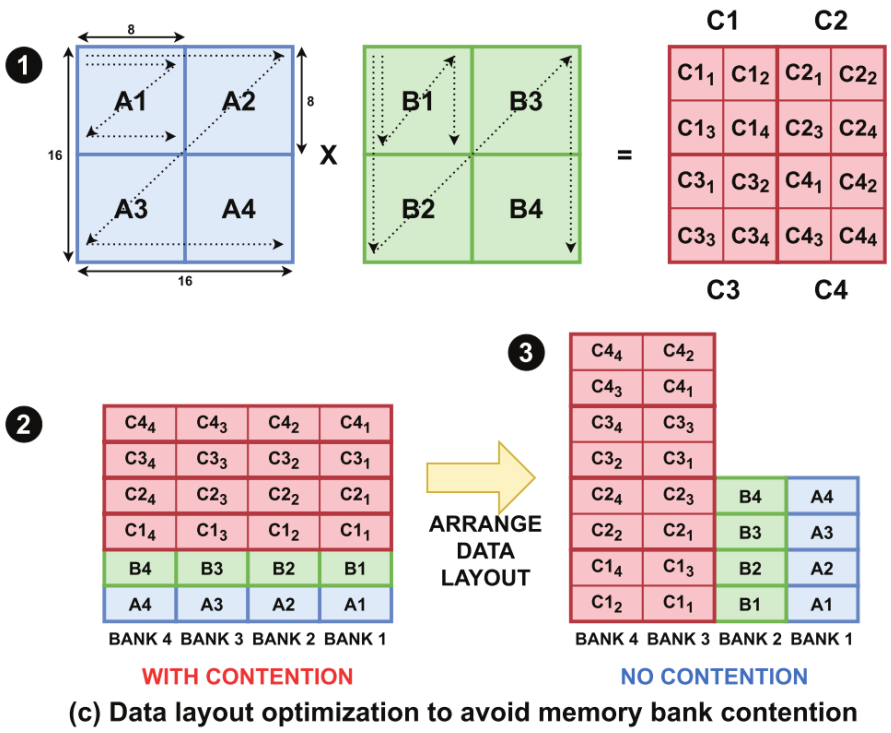
- (3) Strided memory access for bank contention alleviation
  - Banked data layout
  - Flexible address generation



(c) Data layout optimization to avoid memory bank contention

Mechanisms for High Temporal Utilization

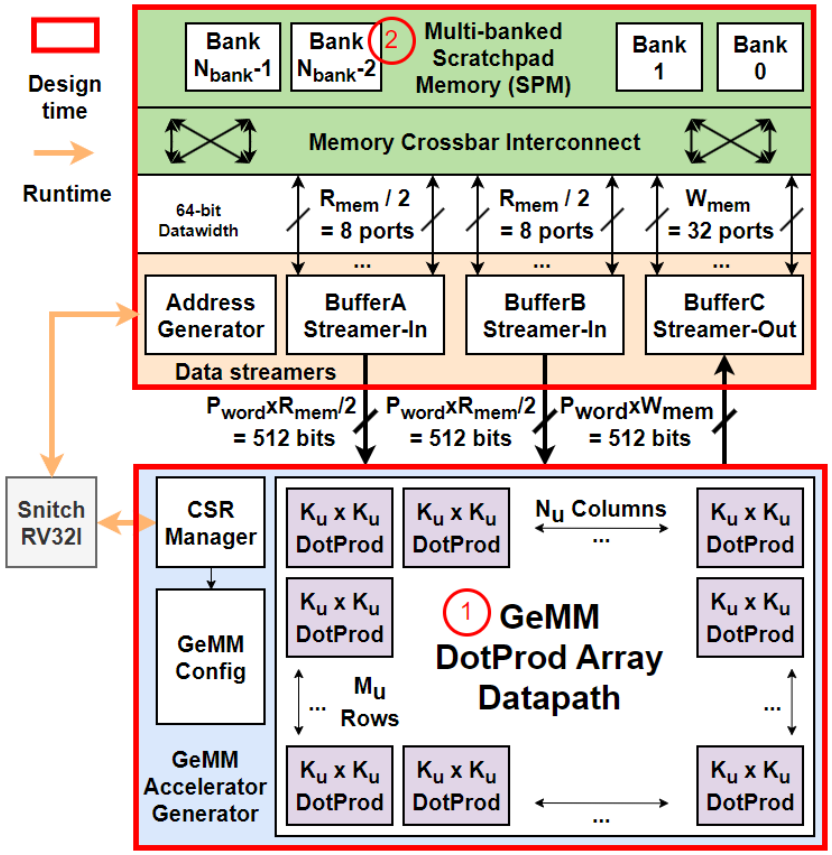
- (3) Strided memory access for bank contention alleviation
  - Banked data layout
  - Flexible address generation
  - Configurable strided address generation unit (AGU)
    - Design time
      - Hardware loop number
    - Runtime
      - Loop bounds, base addresses, address strides



OpenGeMM Reusability and Flexibility Summary

Table 1: Customizable *OpenGeMM* design-time parameters and runtime programmable configurations.

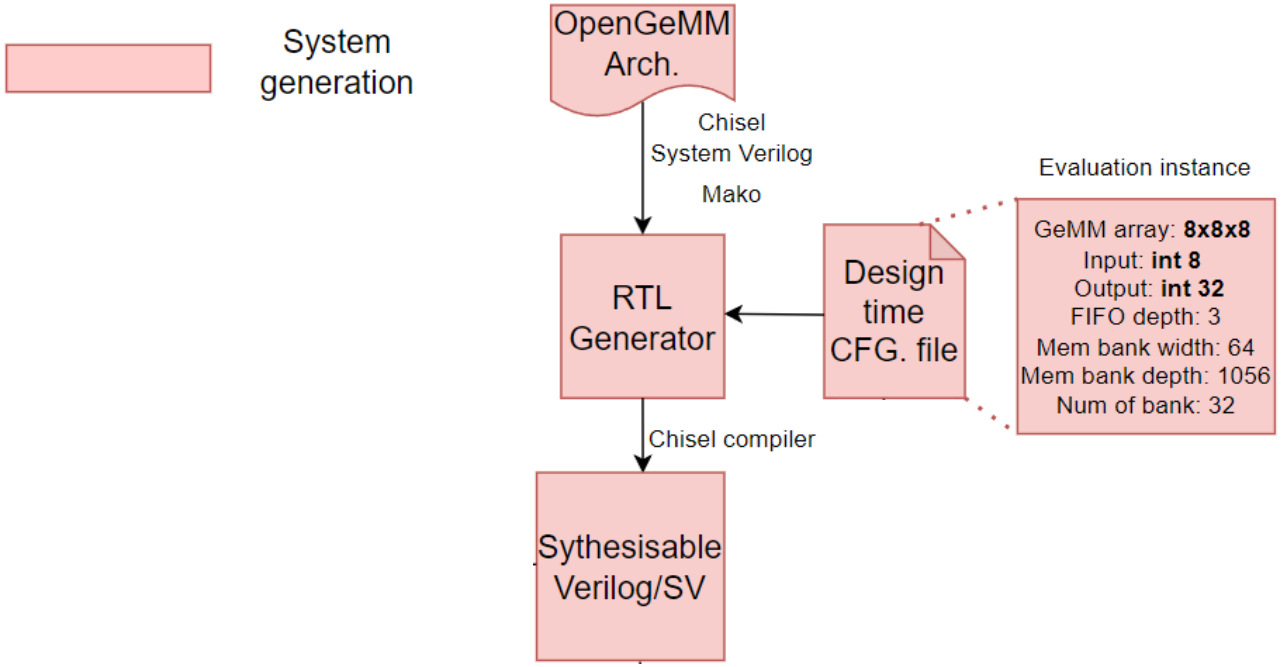
Type	Parameters	Meaning
	① **Parameters for GeMM core**	
Design time	M_u	Number of rows of the array
	N_u	Number of columns of the array
	K_u	Size of each DotProd
	P_A	Integer bit precision of A
	P_B	Integer bit precision of B
	P_C	Integer bit precision of C
Run time	M/M_u	Temporal loop bound of M dimension
	N/N_u	Temporal loop bound of N dimension
	K/K_u	Temporal loop bound of K dimension
	② **Parameters for memory system**	
Design time	$D_{streamer}$	Pre-fetch buffer and output buffer depth in the streamer
	R_{mem}	Input memory ports
	W_{mem}	Output memory ports
	P_{word}	Memory port data width
	N_{bank}	Number of banks
	D_{mem}	Bank depth
Run time	$LB_{streamer}$	Loop bounds for the address generation
	$BA_{streamer}$	Base address for the address generation
	$S_{streamer}$	Strides for the address generation



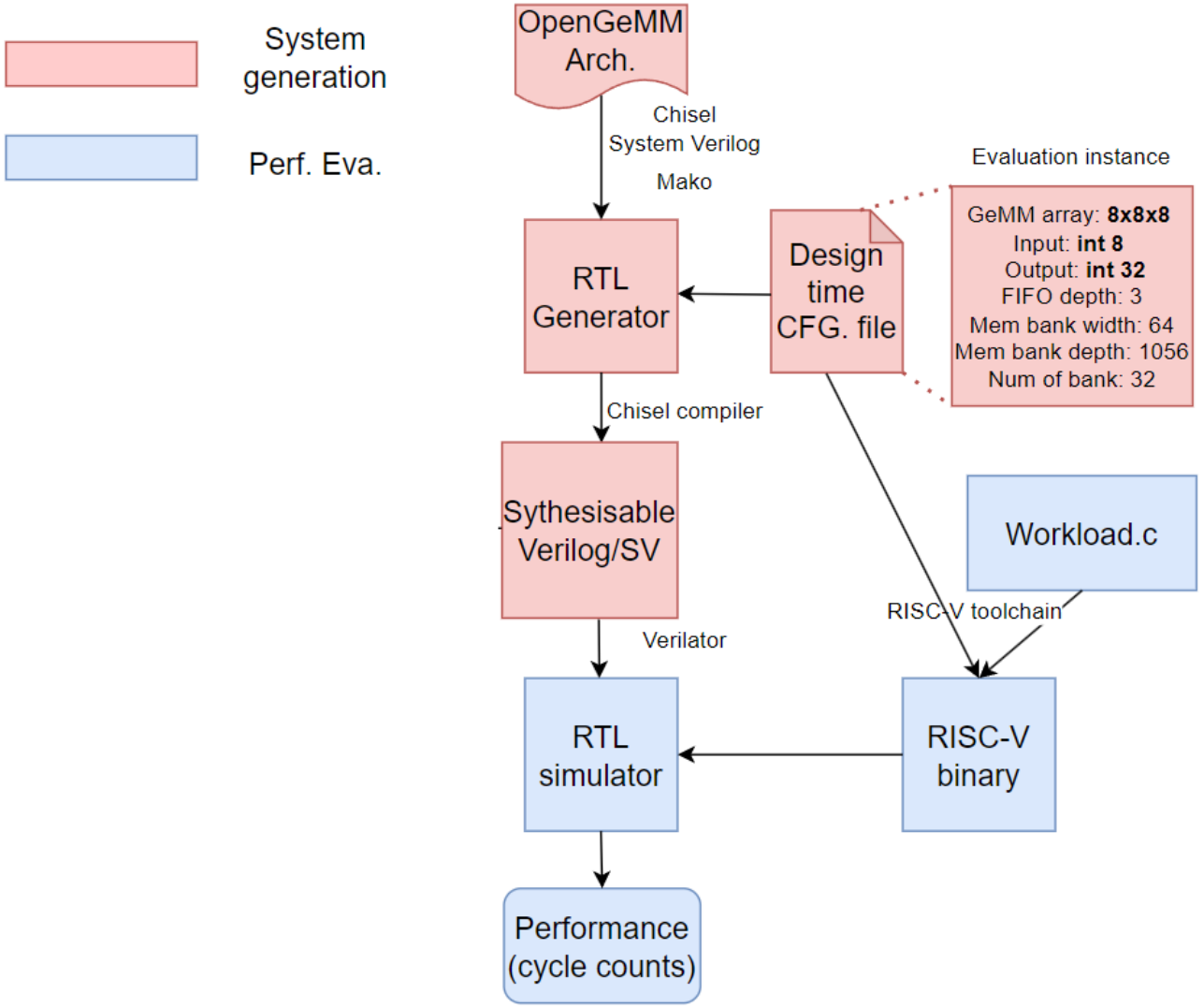
Outline

- Edge AI Computing Background and Motivation
- OpenGeMM System Architecture
  - Overview
  - GeMM Accelerator Generator
  - Mechanisms for High Utilization
  - Reusability and Flexibility Summary
- Experimental Results and SotA Comparison
- Conclusion and Future Work

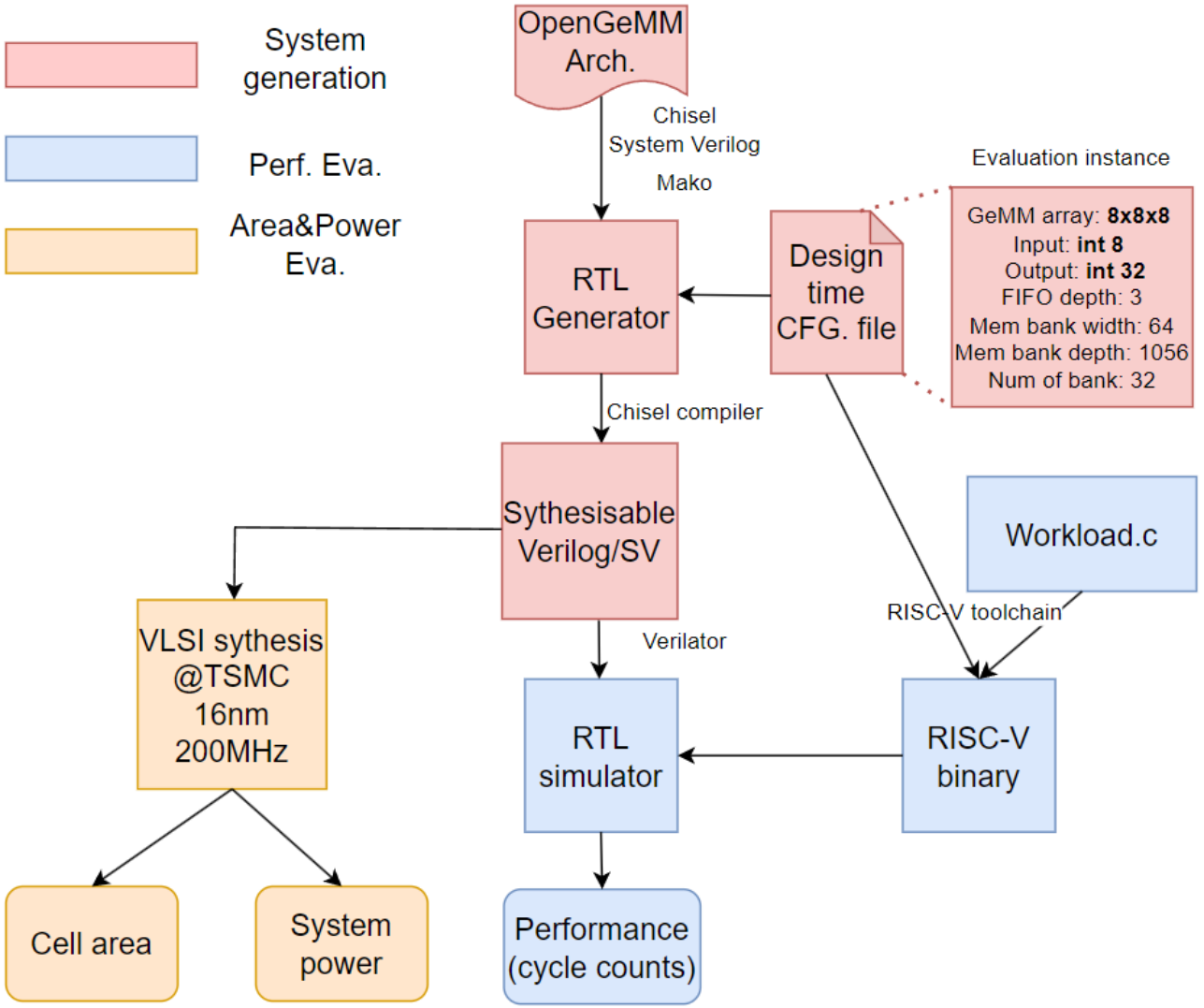
Experimental Setup – OpenGeMM Hardware Generation



Experimental Setup – Performance

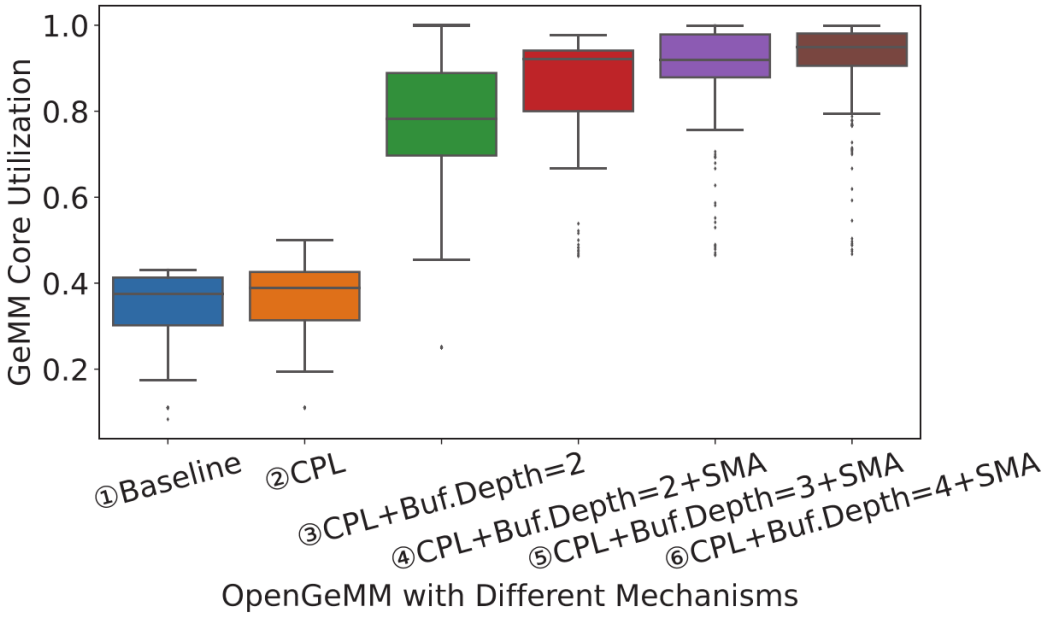


Experimental Setup – Area and Power



GeMM Core Utilization Analysis

- 500 different computational matrix sizes (M, K, N) where M, K, N $\in$ {8, 16, 24,. . . , 256}
- Configuration pre-loading (CPL): 1.4×
- Input pre-fetching and output buffering: 2.02×
- Strided memory access (SMA): 1.18×



Real-life DNNs Benchmarking

- Real-life CNNs and Transformers
- Temporal utilization: 93.74-99.80%
- Spatial utilization: 87.36-99.54%
  - Im2col-ed matrix irregularity in MobileNetV2
- Overall utilization: 81.89% to 99.34%

Table 2: Utilization (in %) and performance (in cycles) of *OpenGeMM* on real DNN workloads.

	MobileNetV2	ResNet18	ViT-B-16	BERT-Base
SU*	87.36	96.01	98.41	99.54
TU†	93.74	99.72	99.75	99.80
OU‡	81.89	95.74	98.16	99.34
CC§	3.33×10^8	9.29×10^8	1.79×10^{10}	4.93×10^{10}

[∗] Spatial utilization. [†] Temporal utilization.
[‡] Overall utilization. [§] Cycle count.

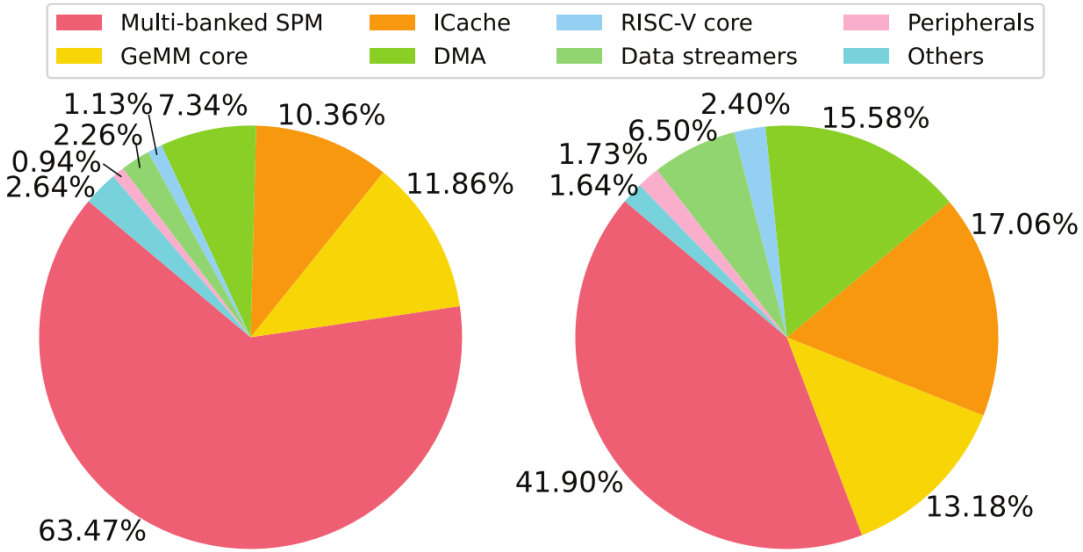
DRAM and on-chip memory communication cycles are not counted.

Area and Power Evaluation

- 0.531 $mm2$ total cell area @TSMC 16nm FFC 200MHZ
- 43.8 mW system power under (32, 32, 32) GeMM workload
- 204.8 GOPS peak performance, 4.68 TOPS/W efficiency

Area and Power Evaluation

- 0.531 $mm2$ total cell area @TSMC 16nm FFC 200MHZ
- 43.8 mW system power under (32, 32, 32) GeMM workload
- 204.8 GOPS peak performance, 4.68 TOPS/W efficiency

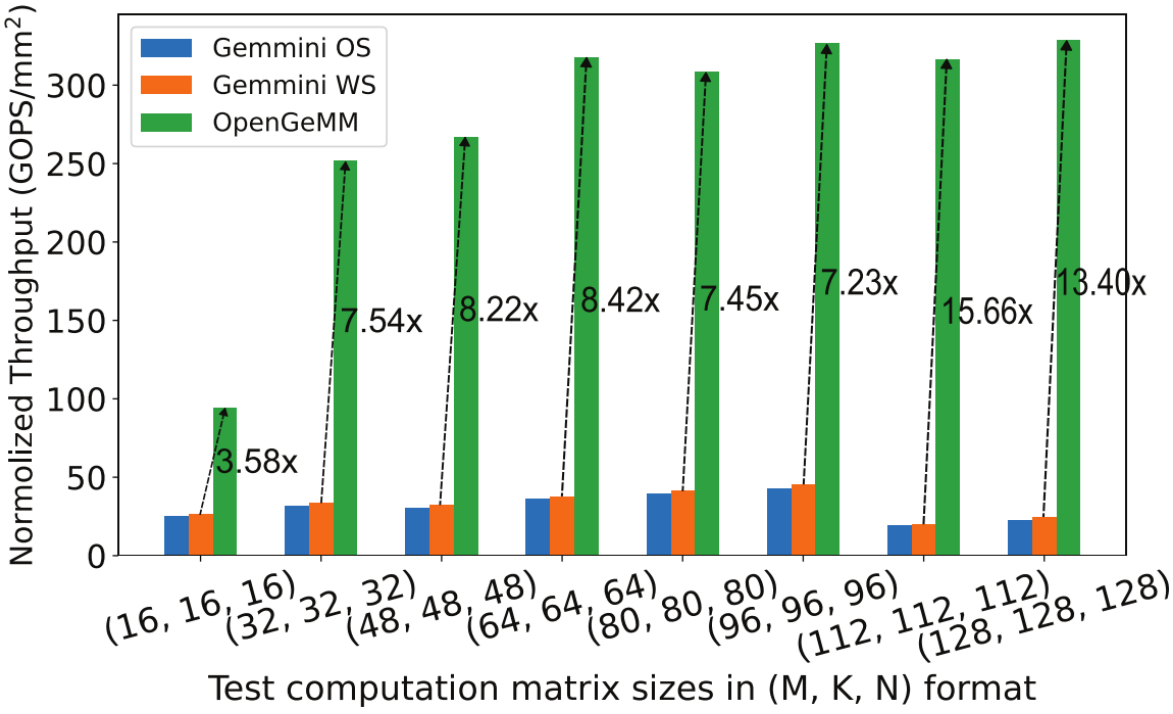


(a) Cell Area Breakdown

(b) Total Power Breakdown

State-of-the-Art Comparison - Utilization

- OpenGeMM vs. Gemmini [2] with output-stationary (OS) and weight-stationary (WS) mode
- 3.75× to 16.40× better vs. Gemmini OS
- 3.58× to 15.66× better vs. Gemmini WS



State-of-the-Art Comparison - Overall

Table 3: State-of-the-Art Comparison. The efficiency metrics shown for each platform are system efficiencies.

Accelerator	SIGMA [26]	CONNA [13]	Gemmini [12]	DIANA [19]	RBE [11]	RedMule [15]	*OpenGeMM* This Work
Tech (nm)	28	65	22	22	22	22	16
Area (mm2)	65	2.36	1.03	8.91	2.42	0.73	0.62†
Memory (KiB)	6,000	144	256	512	128	128	270
Freq (MHz)	500	200	1000	280	420	470	200
Peak Perf. (GOPS)	16,000	102.4	512	224 (Dig.) 40 (AIMC)	637 (2b) 91 (8b)	89	204(8b)
Peak Eff. (TOPS/W)	0.48	0.856	-	1.7 (Dig.) 4 (AIMC)	12.4 (2b) 0.74 (8b)	1.6	4.68(8b)
Peak Perf./Area (GOPS/mm2)	246	43	497	25 (Dig.) 4.5 (AIMC)	263 (2b) 37 (8b)	121	329(8b)†
Op-Area-Eff. (TOPS/W/mm2)	0.0073	0.363	-	0.2 (Dig.) 0.44 (AIMC)	5.12 (2b) 0.31 (8b)	2.2	7.55(8b)†
Supported Precision	BFP 16, FP 32	INT 4, 8, 16, 32	INT8	INT 8	INT 2, 4, 8	FP 8, 16	INT 2, 4, 8§
Open Source	✓	×	✓	✓	✓	✓	✓
Generated Arch.*	×	✓	✓	×	×	×	✓
Design- and Run- time Config.¶	×	✓	✓	×	✓	✓	✓

* Means the design comes from a hardware generator like Chisel.
¶ Has parameters configurable during design-time (or hardware design configurations) and run-time (or programmable).
† After placement and routing layout area estimation with 60% cell density according to [27].
§ Design-time configurable.

Conclusion and Future Work

Conclusion

- OpenGeMM: an open-source GeMM acceleration platform targeting edge AI applications
  - GeMM accelerator generator, a lightweight RISC-V processor, and a tightly coupled memory system
- Three mechanisms for high hardware utilization at the system level
  - Configuration pre-loading, input pre-fetching with output buffering, and programmable strided memory access

Future Work

- More flexible/sparse GeMM core generation, multi-core computing cluster…
- Maping more emerging workloads

References

- [1] Frans Sijstermans. The nvidia deep learning accelerator. In Hot Chips, volume 30, pages 19–21, 2018.
- [2] Hasan Genc, Seah Kim, Alon Amid, Ameer Haj-Ali, Vighnesh Iyer, Pranav Prakash, Jerry Zhao, Daniel Grubb, Harrison Liew, Howard Mao, et al. Gemmini: Enabling systematic deep-learning architecture evaluation via full-stack integration. In 2021 58th ACM/IEEE Design Automation Conference (DAC), pages 769–774. IEEE, 2021.
- [3] Yvan Tortorella, Luca Bertaccini, Luca Benini, Davide Rossi, and Francesco Conti. Redmule: A mixed-precision matrix-matrix operation engine for flexible and energy-efficient on-chip linear algebra and tinyml training acceleration. arXiv preprint arXiv:2301.03904, 2023.
- [4] Pouya Houshmand, Giuseppe M. Sarda, Vikram Jain, Kodai Ueyoshi, Ioannis A. Papistas, Man Shi, Qilin Zheng, Debjyoti Bhattacharjee, Arindam Mallik, Peter Debacker, Diederik Verkest, and Marian Verhelst. Diana: An end-to-end hybrid digital and analog neural network soc for the edge. IEEE Journal of Solid-State Circuits, 58(1):203–215, 2023. doi: 10.1109/JSSC.2022.3214064.
- [5] Linyan Mei, Pouya Houshmand, Vikram Jain, Sebastian Giraldo, and Marian Verhelst. Zigzag: Enlarging joint architecture-mapping design space exploration for dnn accelerators. IEEE Transactions on Computers, 70(8):1160–1174, 2021.
- [6] Kim, Hyungyo, Gaohan Ye, Nachuan Wang, Amir Yazdanbakhsh, and Nam Sung Kim. "Exploiting Intel® Advanced Matrix Extensions (AMX) for Large Language Model Inference." IEEE Computer Architecture Letters (2024).
- [7] Florian Zaruba, Fabian Schuiki, Torsten Hoefler, and Luca Benini. Snitch: A tiny pseudo dual-issue processor for area and energy efficient execution of floating point intensive workloads. IEEE Transactions on Computers, 70(11):1845–1860, 2020.
- [8] K. Goetschalckx and M. Verhelst, “DepFiN: A 12nm, 3.8TOPs depthfirst CNN processor for high res. image processing,” in 2021 Symposium on VLSI Circuits, 2021, pp. 1–2.
- [9] Kim, Hyungyo, Gaohan Ye, Nachuan Wang, Amir Yazdanbakhsh, and Nam Sung Kim. "Exploiting Intel® Advanced Matrix Extensions (AMX) for Large Language Model Inference." IEEE Computer Architecture Letters (2024).
- [10] Gonzalez, Abraham, Jerry Zhao, Ben Korpan, Hasan Genc, Colin Schmidt, John Wright, Ayan Biswas et al. "A 16mm 2 106.1 GOPS/W heterogeneous RISC-V multi-core multi-accelerator SoC in low-power 22nm FinFET." In *ESSCIRC 2021-IEEE 47th European Solid State Circuits Conference (ESSCIRC)*, pp. 259-262. IEEE, 2021.

GitHub Repository

- Standalone GeMM core generator
  - https://github.com/KULeuven-MICAS/snax-gemm
- System platform
  - https://github.com/KULeuven-MICAS/snax_cluster

Q&A

Thank you for listening!