Multiply-and-Fire: An Event-Driven Sparse Neural Network Accelerator

19 Jan 2024 @ HiPEAC, Munich

Miao Yu*[1], Tingting Xiang*[1], Venkata Pavan Kumar Miriyala[1], Trevor E. Carlson[1]

[1]National University of Singapore

*Co-first Author

Overview

- Deep Neural Network Models -> Ubiquitous
- To support resource & power constraint edge devices
- Neural Network Hardware Accelerator
  - Sparse Convolutional Neural Networks (CNNs)
  - Accelerates network inference

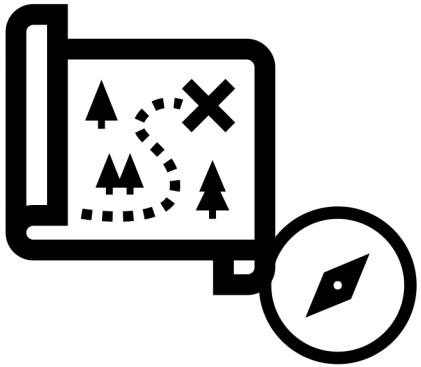
AI Accelerator

GOAL : High Energy Efficiency

- ➢ Novel event-driven sparsity-aware accelerator
- ➢ Efficiently exploits sparsity
- ➢ Reduces memory access

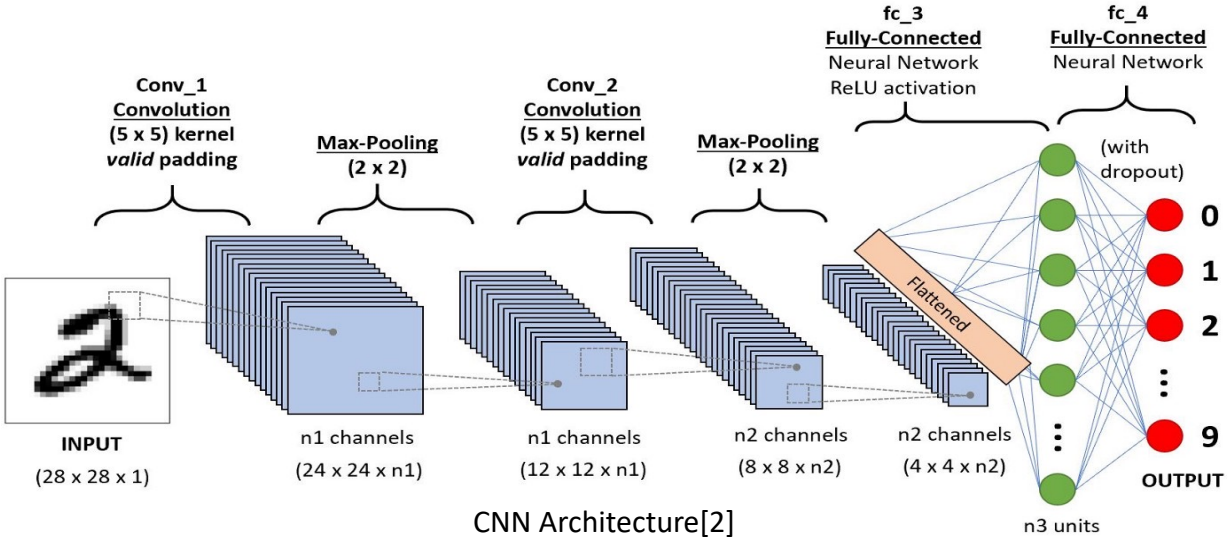
Image retrieved from :https://bigdata-madesimple.com/innovative-ai-chips-to-make-the-world-faster-and-more-efficient/

Outline



- **Introduction**
  - Background on CNN & Sparsity
- **Prior Work & Motivation**
- **MnF Methodology**
  - Event-driven Dataflow
  - Hardware Design
- **Evaluation Setup & Results**
- **Conclusion**

Convolutional Neural Network (CNN)



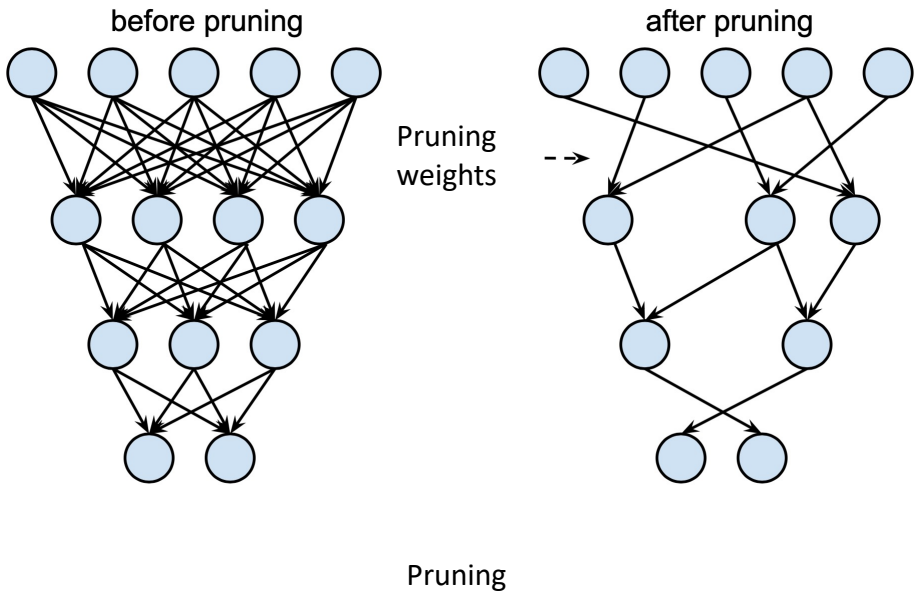
CNN Architecture[2]

- High level of parallelism
- Typically transformed into Matrix multiplication (BUT we do it in a different way!)

[1] Image retrieved from : https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/blocks/2d-convolution
[2] Image retrieved from : https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Sparsity – Weight Pruning

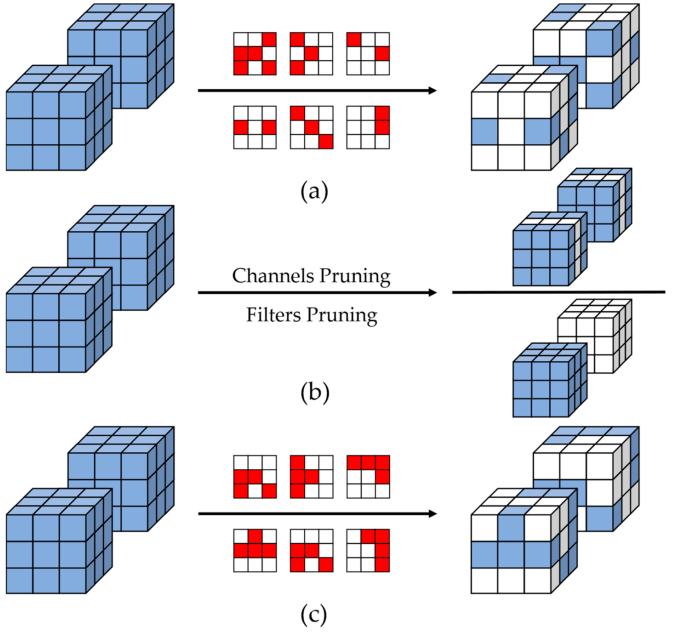
- Pruning weights of low importance
- No/Minimum drop in accuracy
- Benefits
  - Reduce the number of computations
  - Reduce the storage
  - Reduce the memory access
  - Better performance on hardware



Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15, page 1135-1143, Cambridge, MA, USA, 2015. MIT Press.

Sparsity – Weight Pruning

- Unstructured pruning
  - Weights are pruned irregularly without any pattern
  - Unstructured sparsity
  - Exploited by: SCNN, Eyeriss v2, GoSPA
- Structured pruning
  - Weight pruned regularly following a particular pattern
  - Structure sparsity
  - Exploited by: CambriconS, Nvidia A100, S2TA



Different pruning techniques

Sui, X.; Lv, Q.; Zhi, L.; Zhu, B.; Yang, Y.; Zhang, Y.; Tan, Z. A Hardware-Friendly High-Precision CNN Pruning Method and Its FPGA Implementation. *Sensors* 2023, *23*, 824. https://doi.org/10.3390/s23020824

Sparsity – Weight Pruning

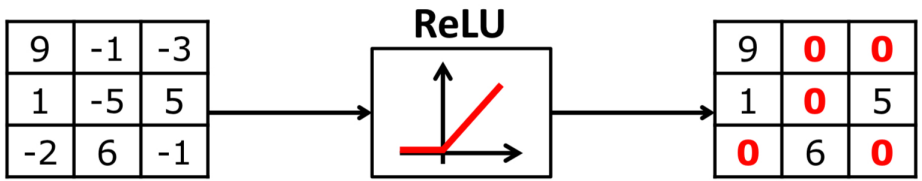
Unstructured	Structured
High compression ratio	Lower compression ratio
Able to maintain accuracy	Able to maintain accuracy
Unstructured sparsity	Structure sparsity
Low storage requirement	Less storage requirement

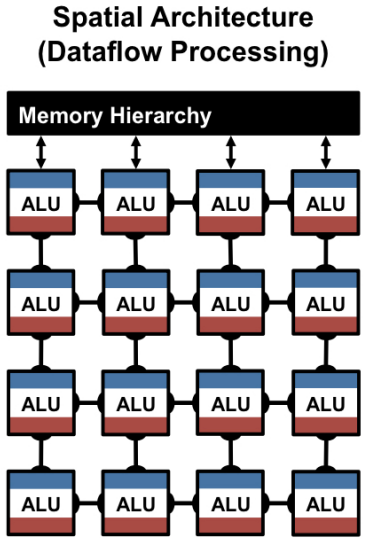
- Compression ratio of structured pruning is ~ 1.14x to 2.56x lower
- Storage requirement of structurally pruned models is 1.7x to 3x less than or comparable (1.09x and 1.35x more) to unstructurally pruned models.

Xiaolong Ma, Sheng Lin, Shaokai Ye, Zhezhi He, Linfeng Zhang, Geng Yuan, Sia Huat Tan, Zhengang Li, Deliang Fan, Xuehai Qian, et al. 2021.Non-Structured DNN Weight Pruning–Is It Beneficial in Any Platform? IEEE Transactions on Neural Networks and Learning Systems (2021).

Sparsity – Activation

- Activation is the input to a layer & output of a layer
- ReLU Function
  - Max(0, x)
  - Result in irregular data
- Typically processed in groups

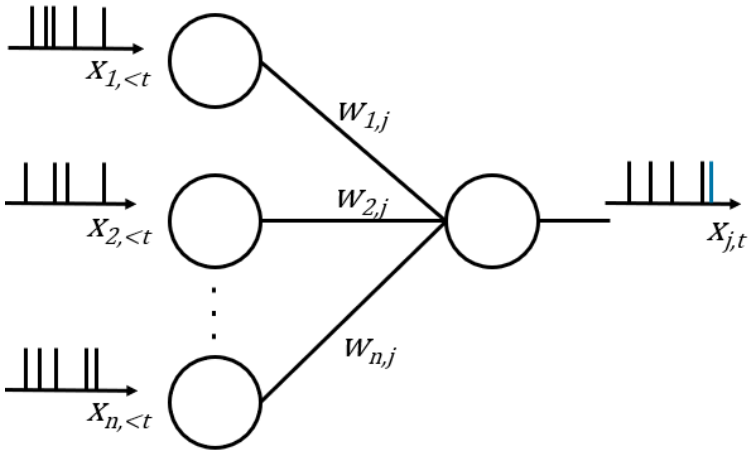


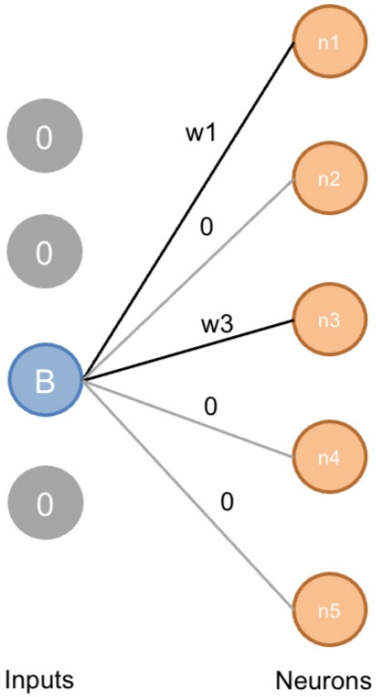


Images retrieved from : http://eyeriss.mit.edu/2019_neurips_tutorial.pdf

Sparsity – Activation

- Activation sparsity is ~50%
- How about process it individually like SNN ?





Images retrieved from : http://eyeriss.mit.edu/2019_neurips_tutorial.pdf

Perspective On Sparse CNNs

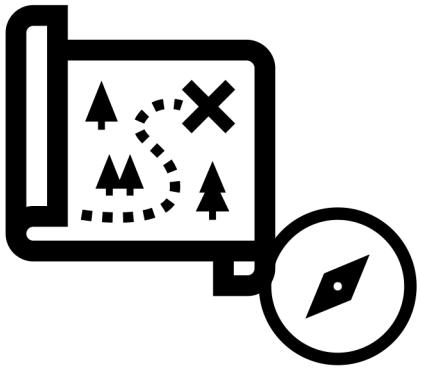
- Benefits
  - Reduce the number of computations
  - Reduce the storage
  - Reduce the memory access
- Challenges to hardware accelerators
  - Supporting irregular data sparsity
  - Avoid multiplications with zero operands
  - Ensure high utilization of MAC units on hardware
  - Avoid high number of accesses to DRAM and global SRAM

Operation	Energy [pJ]	Relative Cost
32 bit int ADD	0.1	1
32 bit float ADD	0.9	9
32 bit Register File	1	10
32 bit int MULT	3.1	31
32 bit float MULT	3.7	37
32 bit SRAM Cache	5	50
32 bit DRAM Memory	**640**	**6400**

Energy Table

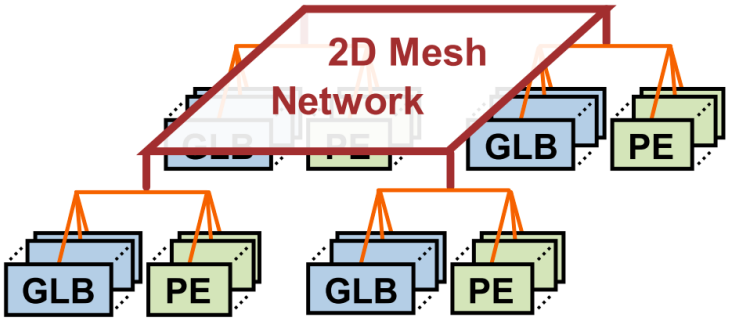
Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15, page 1135-1143, Cambridge, MA, USA, 2015. MIT Press.

Outline



- Introduction
  - Background on CNN & Sparsity
- **Prior Work & Motivation**
- MnF Methodology
  - Event-driven Dataflow
  - Hardware Design
- Evaluation Setup & Results
- Conclusion

Existing accelerators : Eyeriss v2



Eyeriss v2 architecture

- Exploits unstructured pruning, stores data in the CSC format

✓ Avoids zero-valued (pruned weights/ zero-valued activation) storage and access

✓ Skips zero-valued multiplications

✗ Storage format memory overhead

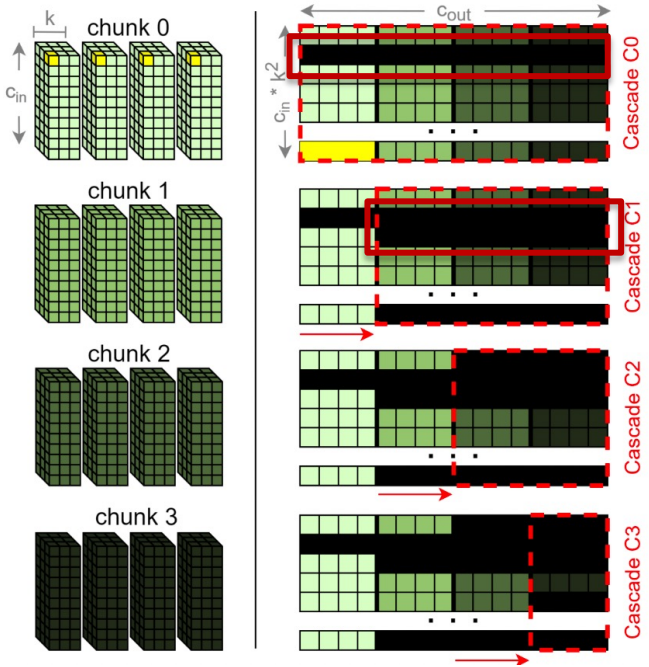
✗ Read dependencies

✗ Unnecessary input access

✗ 73% area overhead , 25% power overhead

Yu-Hsin Chen, Tushar Krishna, Joel S. Emer, and Vivienne Sze. 2017. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. IEEE J. Solid-State Circ. 52, 1 (2017), 127–138.

Existing accelerators : CSP



CSP pruning technique

- Hardware-software co-design
- Exploits structured pruning, and stores data in the customized compressed format

✓ Avoids complex logic to handle irregular weight sparsity

✓ Skips zero-valued weight multiplications

✗ Activation sparsity not exploited

✗ Unnecessary input access

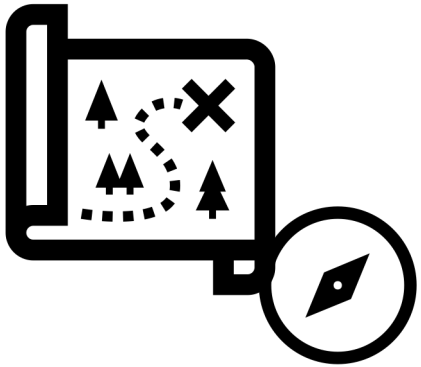
✗ Computes ineffectual multiplications

Edward Hanson, Shiyu Li, Hai 'Helen' Li, and Yiran Chen. 2022. Cascading structured pruning: Enabling high data reuse for sparse DNN accelerators. In Proceedings of the 49th Annual International Symposium on Computer Architecture (ISCA'22). 522–535.

Proposed Work

- Prior works
  - Memory overhead
  - Complex and expensive hardware logic
  - No support for activation sparsity
  - Performs ineffectual multiplications
- **Our work**
  - **Leverages structured pruning in Conv and FC layers, unstructured pruning in FC layers**
  - **Novel event-driven dataflow**
  - **Efficiently exploits activation sparsity**
  - **Without complex and high logic overhead**

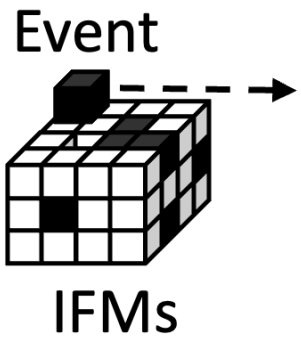
Outline

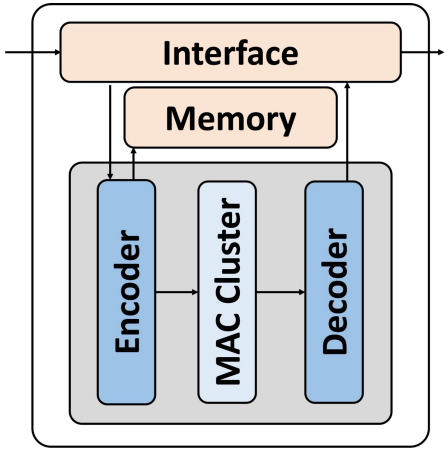


- Introduction
  - Background on CNN & Sparsity
- Prior Work & Motivation
- **MnF Methodology**
  - Event-driven Dataflow
  - Hardware Design
- Evaluation Setup & Results
- Conclusion

Multiply-and-Fire

- A novel event-driven dataflow to handle activation sparsity efficiently without complex, high-overhead logic.
- An accelerator enables a highly parallel computation of each activation with high energy efficiency and performance.
- Support sparsity in convolutional neural networks.

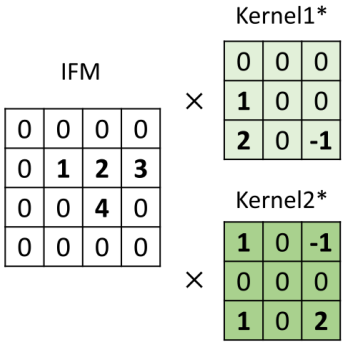




Methodology – Event-driven Dataflow

✓ Utilize structured pruning to save storage

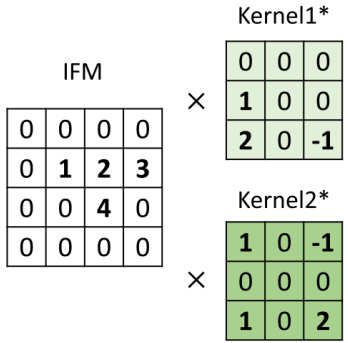
Eyeriss V2
SCNN
Sparten
GoSPA
…



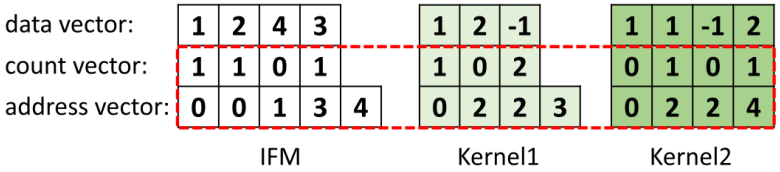
Methodology – Event-driven Dataflow

✓ Utilize structured pruning to save storage

Eyeriss V2
SCNN
Sparten
GoSPA
...



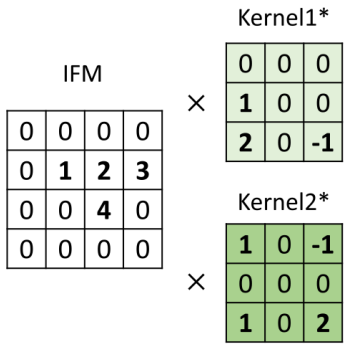
*Activations and kernels with unstructured sparsity are stored and accessed in CSC format



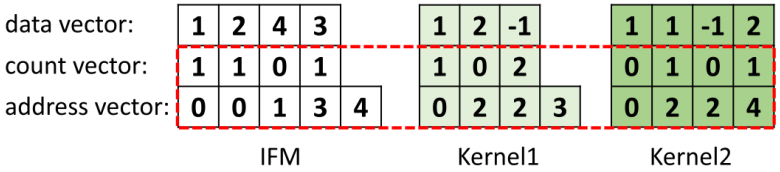
Methodology – Event-driven Dataflow

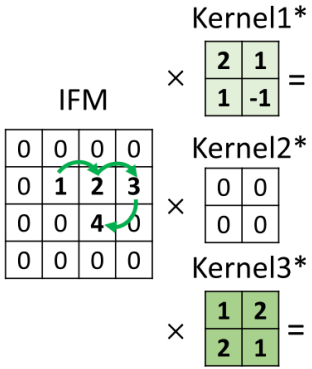
✓ Utilize structured pruning to save storage

Eyeriss V2
SCNN
Sparten
GoSPA
...



*Activations and kernels with unstructured sparsity are stored and accessed in CSC format



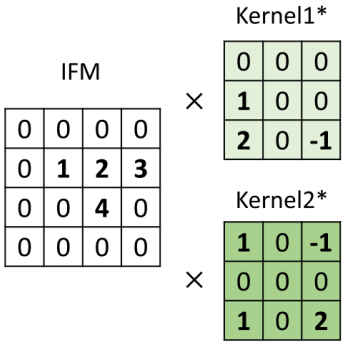


MnF

Methodology – Event-driven Dataflow

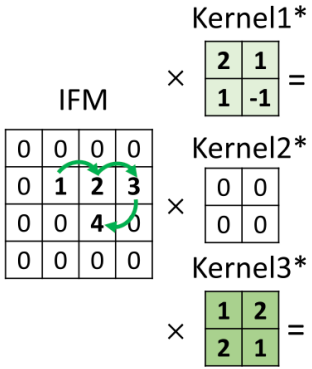
✓ Utilize structured pruning to save storage

Eyeriss V2
SCNN
Sparten
GoSPA
...



*Activations and kernels with unstructured sparsity are stored and accessed in CSC format

	IFM	Kernel1	Kernel2
data vector:	1 2 4 3	1 2 -1	1 1 -1 2
count vector:	1 1 0 1	1 0 2	0 1 0 1
address vector:	0 0 1 3 4	0 2 2 3	0 2 2 4



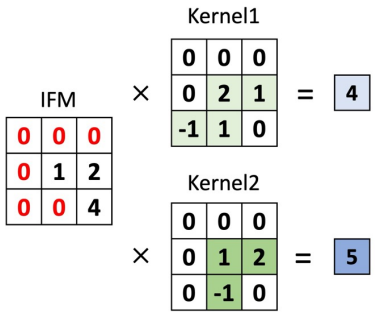
MnF

*Kernels with structured sparsity are stored and accessed in dense format

2	1	1	-1	1	2	2	1

Methodology – Event-driven Dataflow

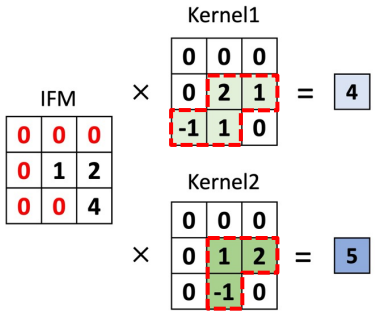
- ✓ Utilize structured pruning to save storage
- ✓ Utilize event-driven dataflow to highly reuse activation



CSP, SPOTS, …

Methodology – Event-driven Dataflow

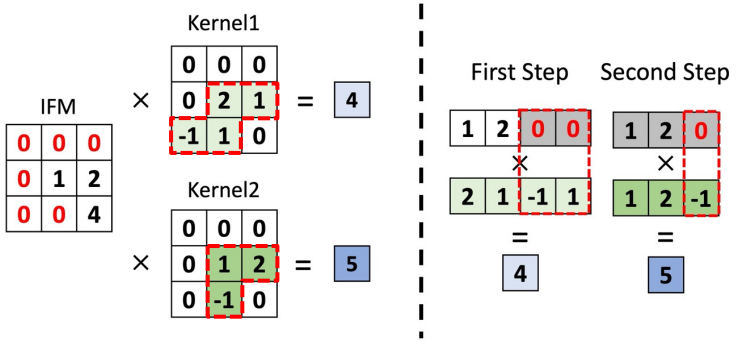
- ✓ Utilize structured pruning to save storage
- ✓ Utilize event-driven dataflow to highly reuse activation



CSP, SPOTS, …

Methodology – Event-driven Dataflow

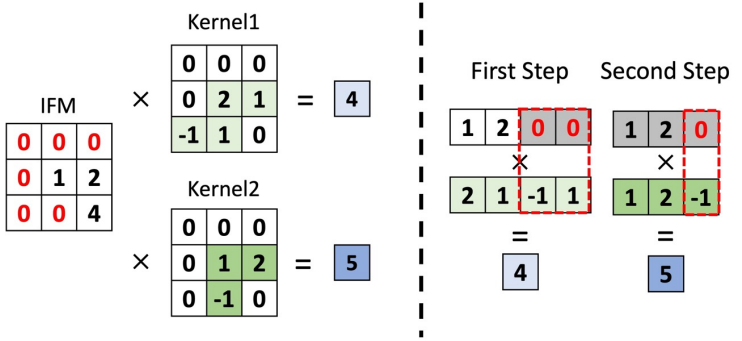
- ✓ Utilize structured pruning to save storage
- ✓ Utilize event-driven dataflow to highly reuse activation



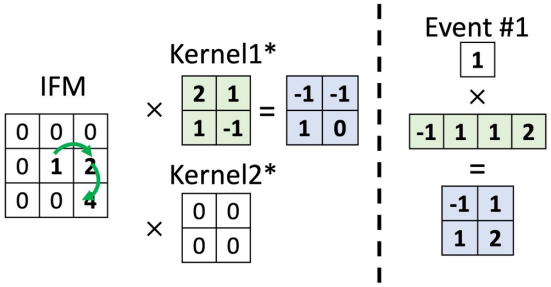
CSP, SPOTS, …

Methodology – Event-driven Dataflow

- ✓ Utilize structured pruning to save storage
- ✓ Utilize event-driven dataflow to highly reuse activation



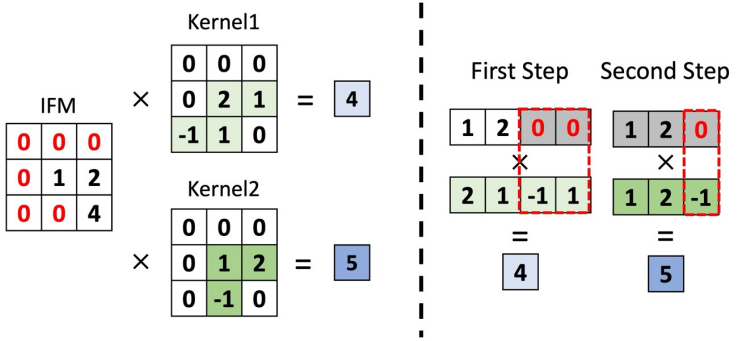
CSP, SPOTS, …



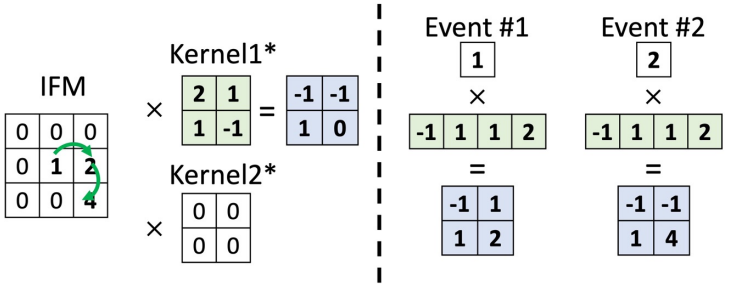
MnF

Methodology – Event-driven Dataflow

- ✓ Utilize structured pruning to save storage
- ✓ Utilize event-driven dataflow to highly reuse activation



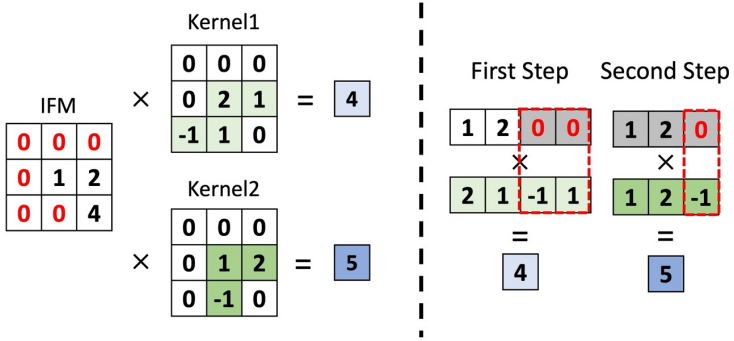
CSP, SPOTS, …



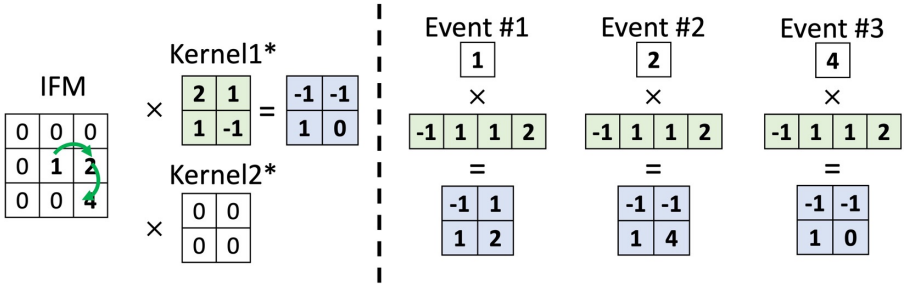
MnF

Methodology – Event-driven Dataflow

- ✓ Utilize structured pruning to save storage
- ✓ Utilize event-driven dataflow to highly reuse activation



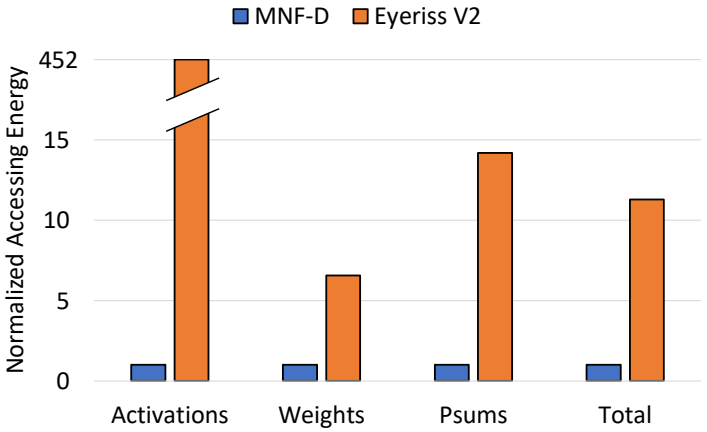
CSP, SPOTS, …

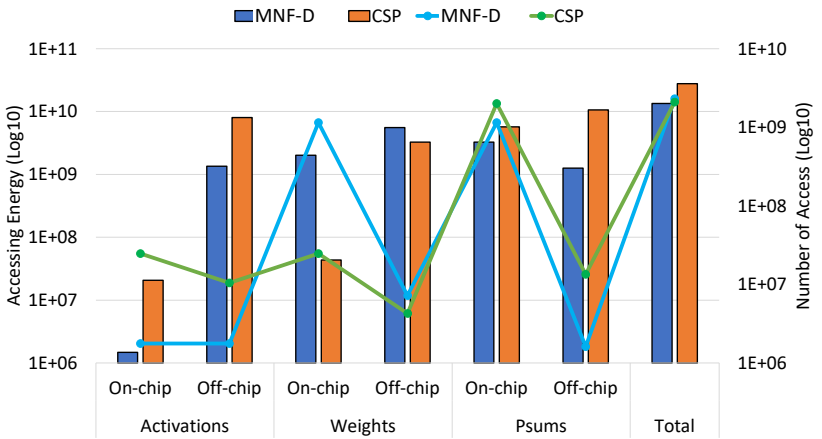


MnF

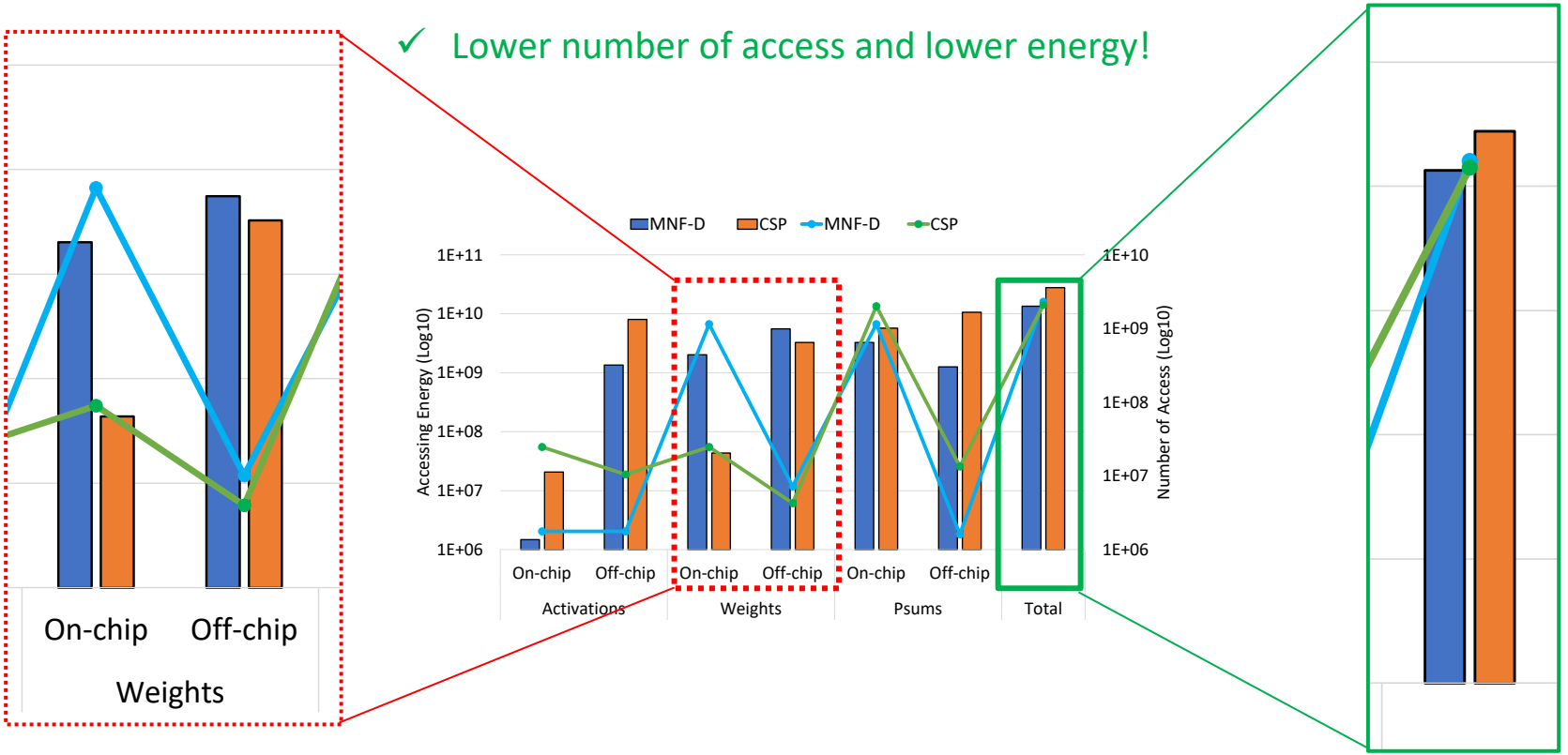
Methodology – Event-driven Dataflow

✓ Lower number of access and lower energy!



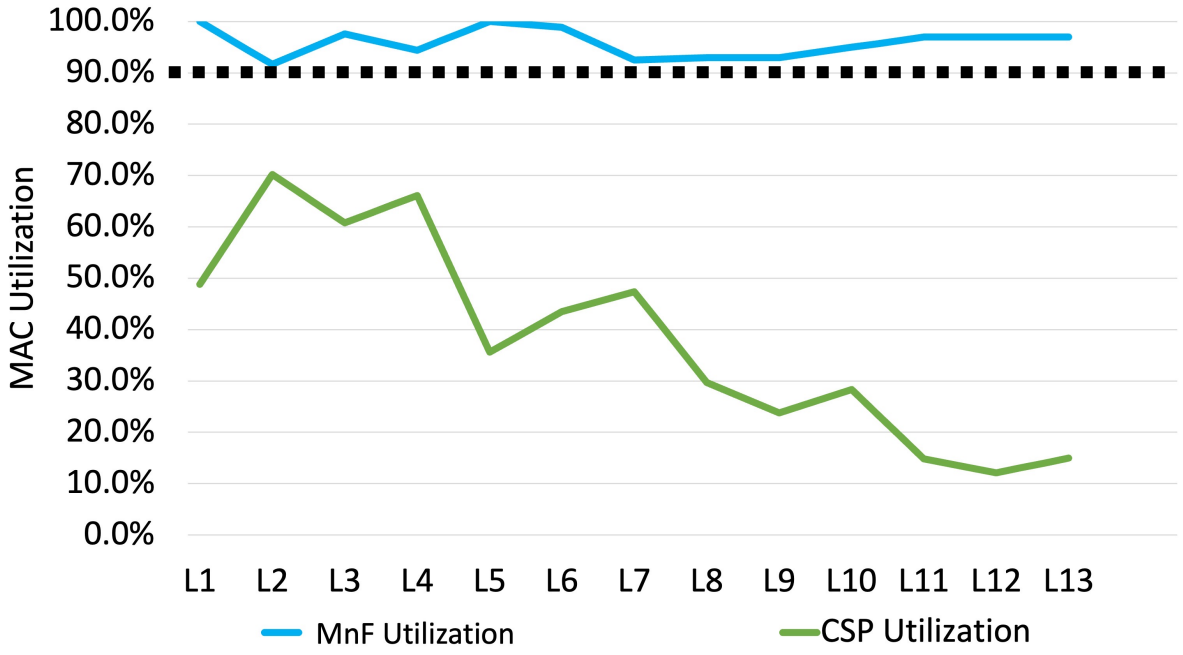


Methodology – Event-driven Dataflow

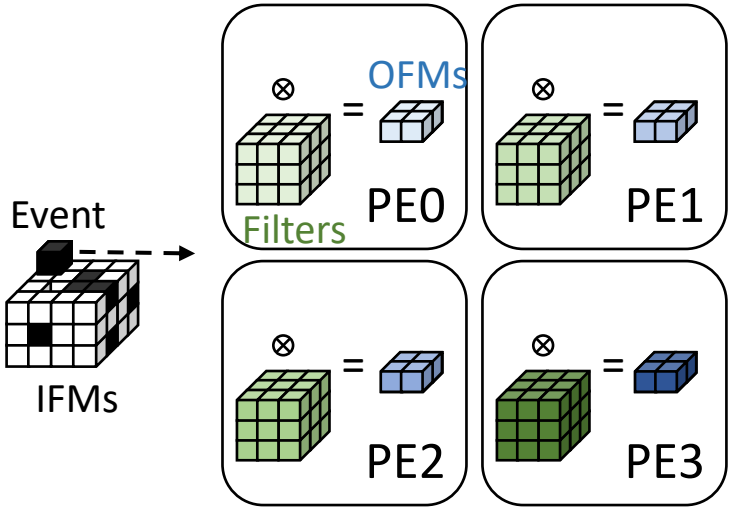


Methodology – Event-driven Dataflow

✓ Higher utilization!

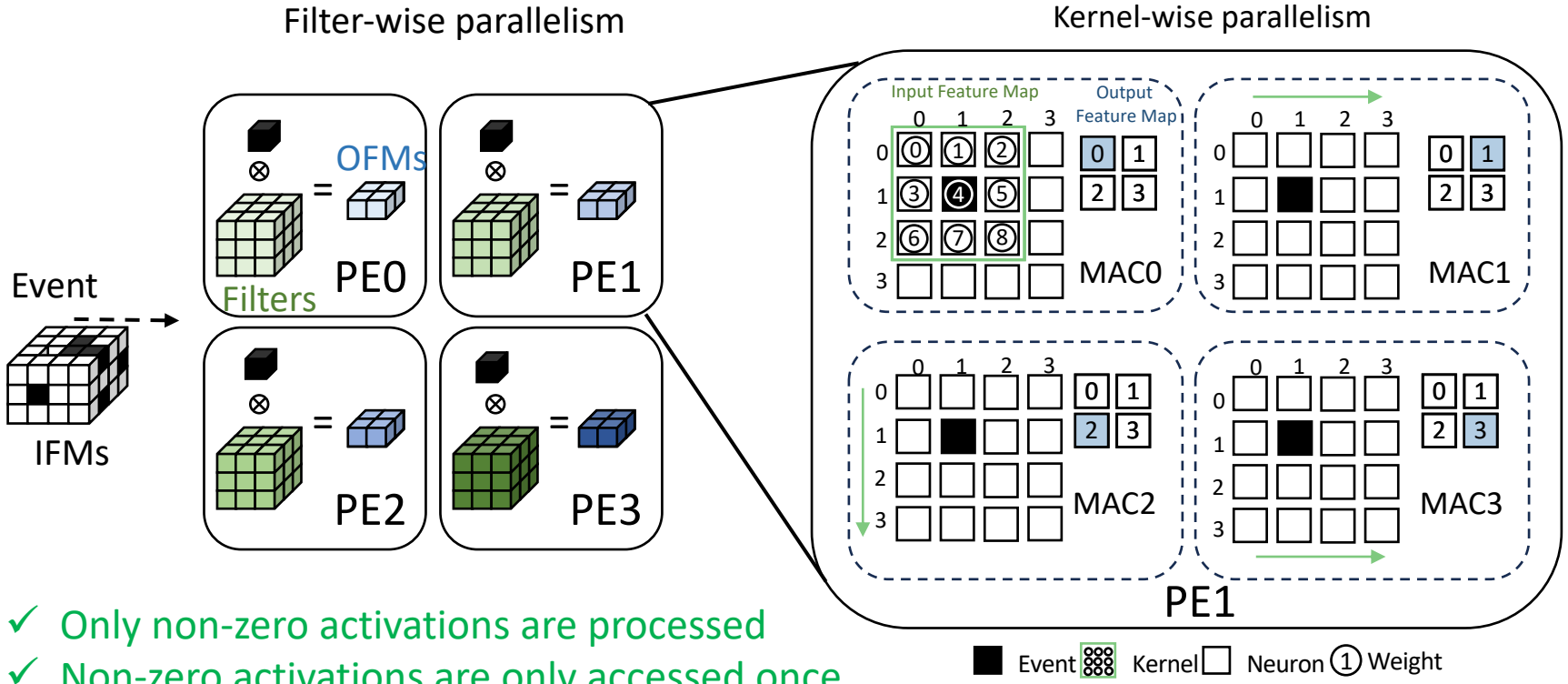


Methodology – Event-driven Dataflow



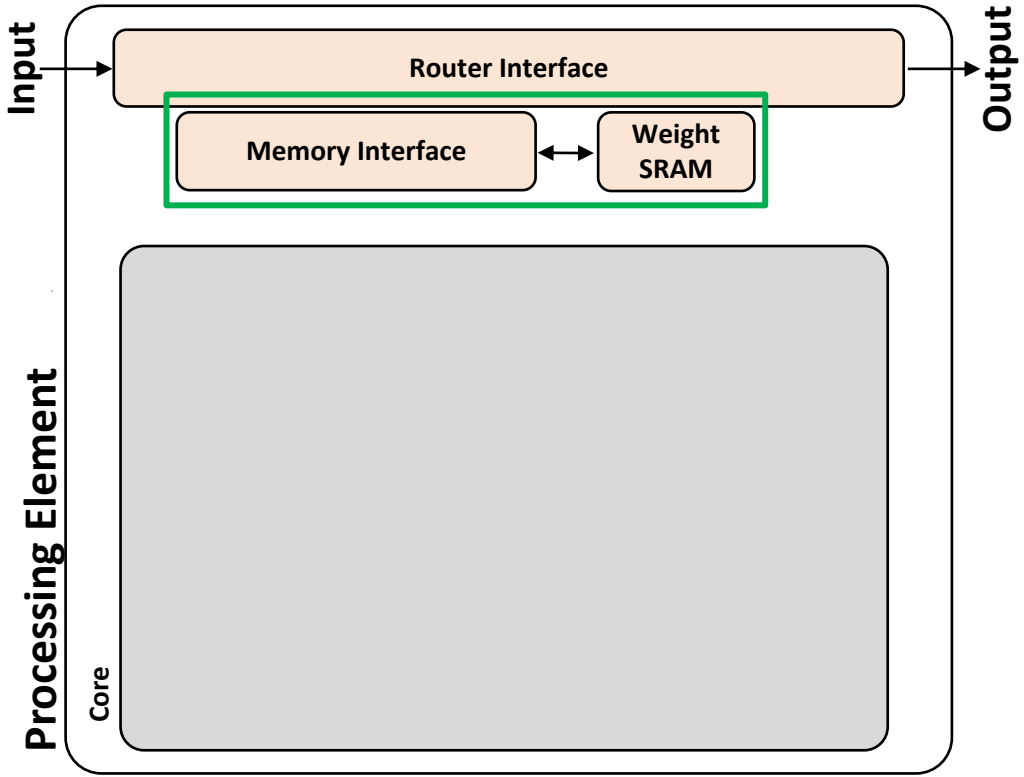
- ✓ Only non-zero activations are processed
- ✓ Non-zero activations are only accessed once.

Methodology – Event-driven Dataflow



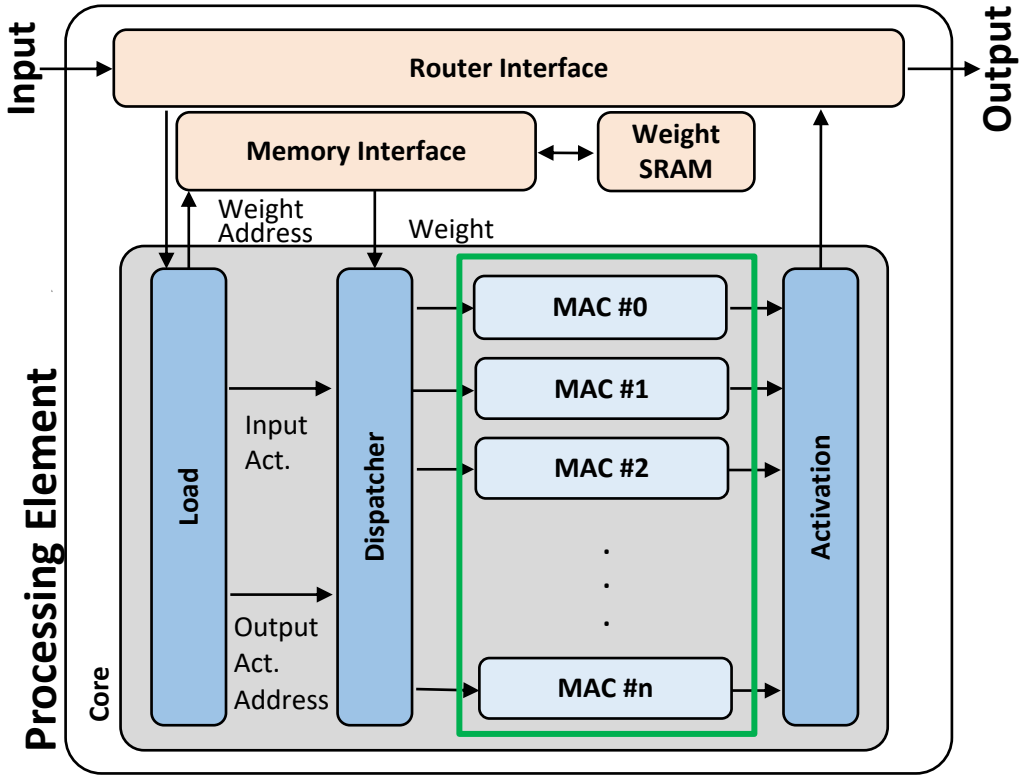
- ✓ Only non-zero activations are processed
- ✓ Non-zero activations are only accessed once.
- ✓ Access weight regularly

Methodology – Hardware



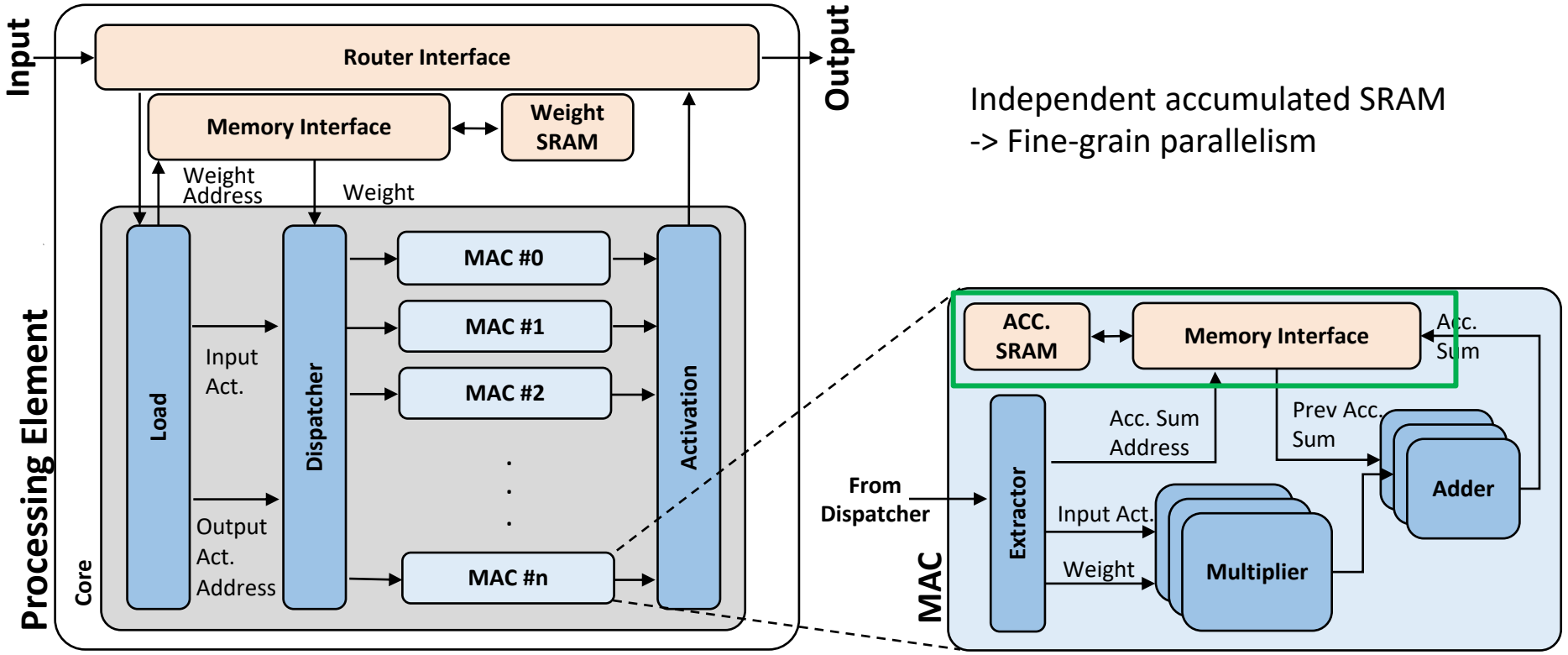
The large size of on-chip SRAM
-> Low access energy

Methodology – Hardware



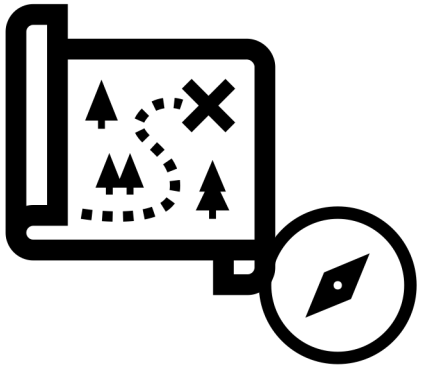
A large number of computing units
-> High degree of parallelism

Methodology – Hardware



Independent accumulated SRAM
-> Fine-grain parallelism

Outline



- Introduction
  - Background on CNN & Sparsity
- Prior Work & Motivation
- MnF Methodology
  - Event-driven Dataflow
  - Hardware Design
- **Evaluation Setup & Results**
- Conclusion

Evaluation - Setup

- Synopsys Design Compiler version P-2019.03, targeting the 22-nm technology node.
- Clock gating of the inactive SRAMs is implemented with latches and included in the synthesis.
- Gate-level simulations are performed using Synopsys VCS-MX K- 2015.09, and power analysis is performed with Synopsys PrimePower version P-2019.03.
- All simulations and performance analyses of MnF hardware are carried out at a frequency of 200 MHz.

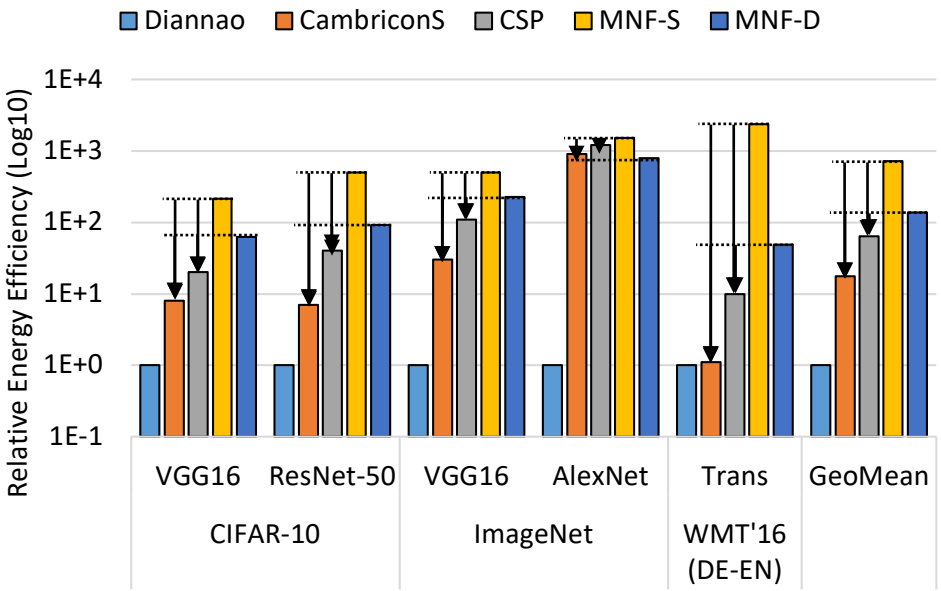
	MnF-DRAM	MnF-SRAM
MAC Cluster Size	9	9
Multiplier per PE	27	27
Weight SRAM per PE	10.1 KB	648 KB
Acc SRAM MAC Cluster	4.69 KB	51.6 KB
Frequency (MHz)	200	200
Bit Precision	Weight/Activation: 8 bits Psum: 25 bits	

MnF-DRAM: implementation with off-chip memory

MnF-SRAM: our target design with only on-chip local memory access

Evaluation – Structured Accelerators

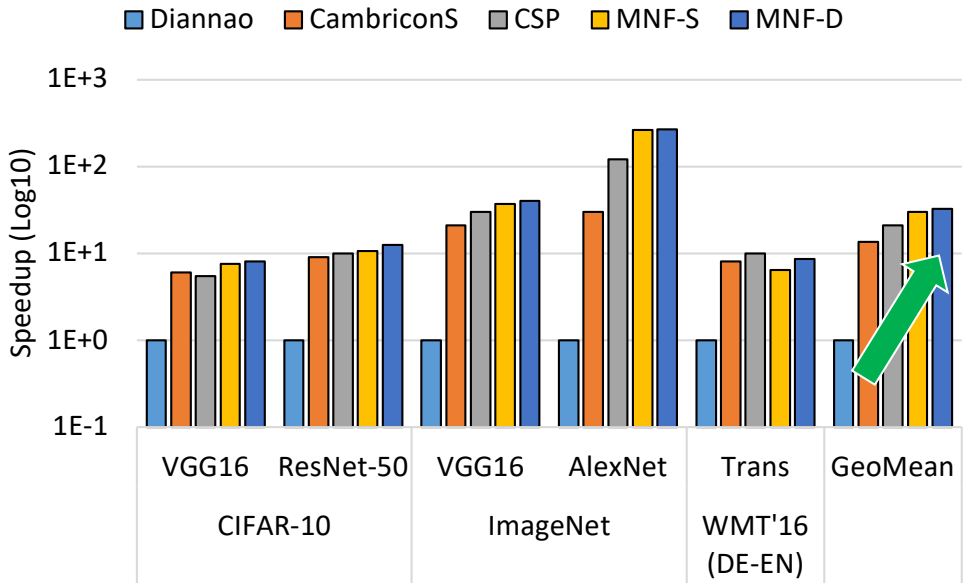
- Comparison with Structured Sparsity-aware DNN Accelerators



- MnF-S is **40.7×** and **11.2×** more energy efficient than Cambricon-S and CSP, respectively.
- MnF-D is **7.8×** and **2.2×** more energy efficient than Cambricon-S and CSP, respectively.

Evaluation – Structured Accelerators

- Comparison with Structured Sparsity-aware DNN Accelerators



- MnF-S is overall **2.19×** and **1.41×** better than Cambricon-S and CSP, respectively.
- MnF-D achieves a **2.38×** and **1.53×** faster speed than Cambricon-S and CSP, respectively.

Evaluation

- Comparison with Unstructured Sparsity-aware DNN Accelerators (Power and Frames/J are scaled to 28 nm)

Design		Eyeriss	Eyeriss V2	NullHop	GoSPA	MnF-S	MnF-D
Frames/S	AlexNet	35	342	-	460	473	473
	VGG-16	1	-	14	30	41	42
	MobileNet	-	1471	-	1868	2894	3180
Frames/J	AlexNet	169	843	-	1587	**2682**	**1700**
	VGG-16	5	-	53	107	**225**	**135**
	MobileNet	-	1708	-	4473	**12426**	**4678**

- Our targeted energy-saving design, MnF-S achieves **28.2×, 4.81×, 4.21×,** and **2.14×** better energy efficiency than Eyeriss, Eyeriss v2, NullHop and GoSPA, respectively.

Evaluation

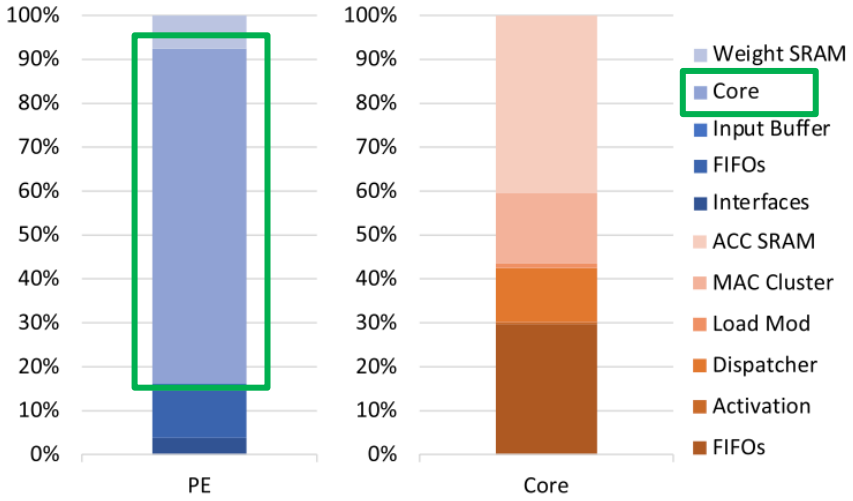
- Comparison with Unstructured Sparsity-aware DNN Accelerators (Power and Frames/J are scaled to 28 nm)

Design		Eyeriss	Eyeriss V2	NullHop	GoSPA	MnF-S	MnF-D
Frames/S	AlexNet	35	342	-	460	**473**	**473**
	VGG-16	1	-	14	30	**41**	**42**
	MobileNet	-	1471	-	1868	**2894**	**3180**
Frames/J	AlexNet	169	843	-	1587	2682	1700
	VGG-16	5	-	53	107	225	135
	MobileNet	-	1708	-	4473	12426	4678

- In terms of speedup, compared to the Eyeriss series, NullHop, and the state-of-the-art design GoSPA, MnF (considering both MnF-S and MnF-D) is at least **1.39×** faster on all the evaluated networks except for AlexNet.

Evaluation – Power & Area Breakdown

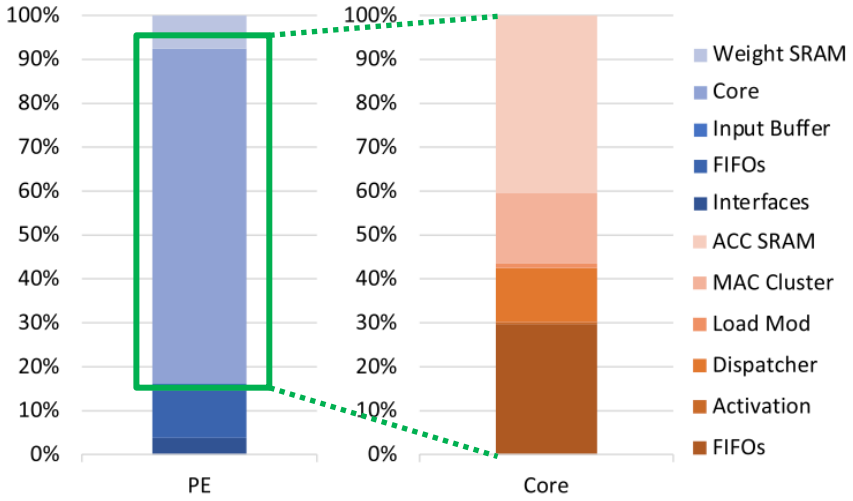
- On-chip memory, FIFOs, and buffer consume 73.5% of the power



(a) Power breakdown of the MnF-S/D processing element.

Evaluation – Power & Area Breakdown

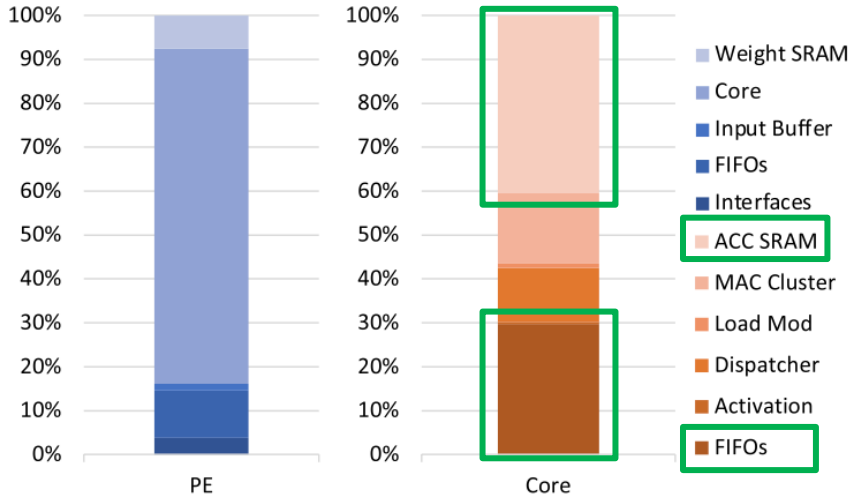
- On-chip memory, FIFOs, and buffer consume 73.5% of the power



(a) Power breakdown of the MnF-S/D processing element.

Evaluation – Power & Area Breakdown

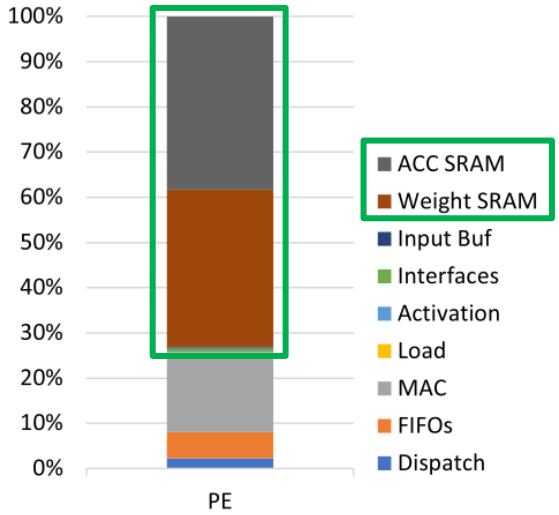
- On-chip memory, FIFOs, and buffer consume 73.5% of the power



(a) Power breakdown of the MnF-S/D processing element.

Evaluation – Power & Area Breakdown

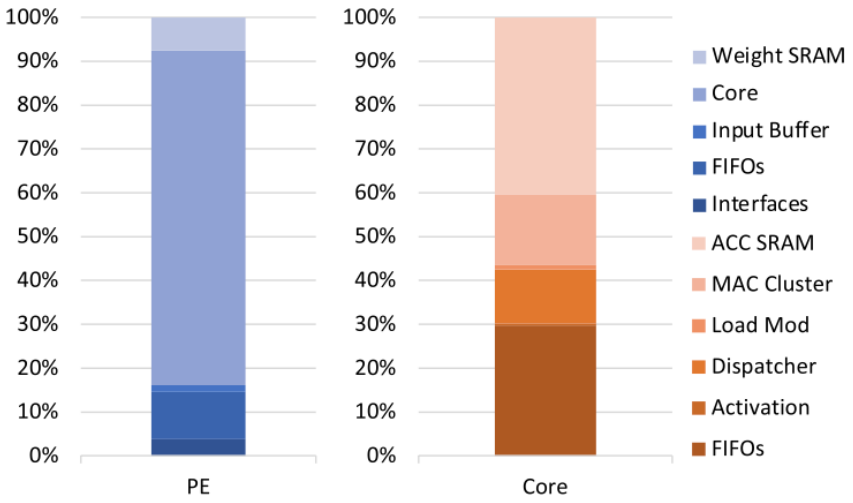
- On-chip memory consumes 73.5% of the total area in the MnF-D PE design



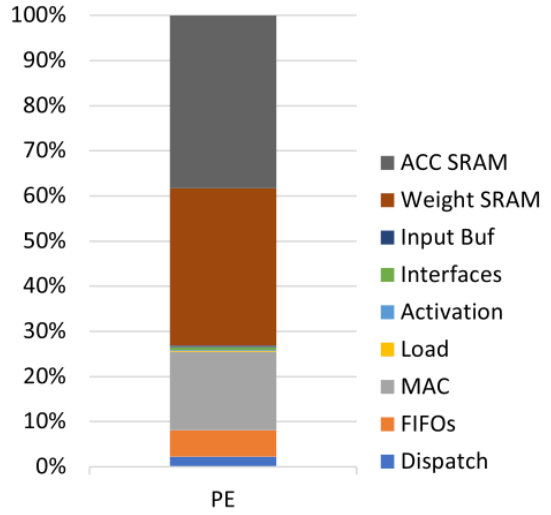
(b) Area breakdown of the MnF-D processing element.

Evaluation – Power & Area Breakdown

- On-chip memory dominates the area and power.

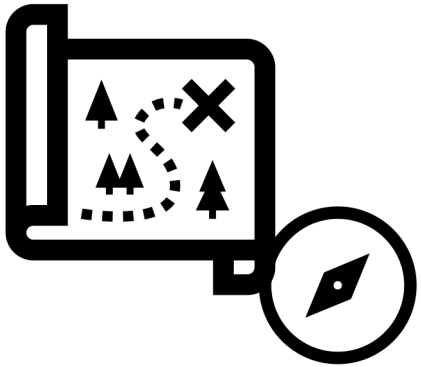


(a) Power breakdown of the MnF-S/D processing element.



(b) Area breakdown of the MnF-D processing element.

Outline



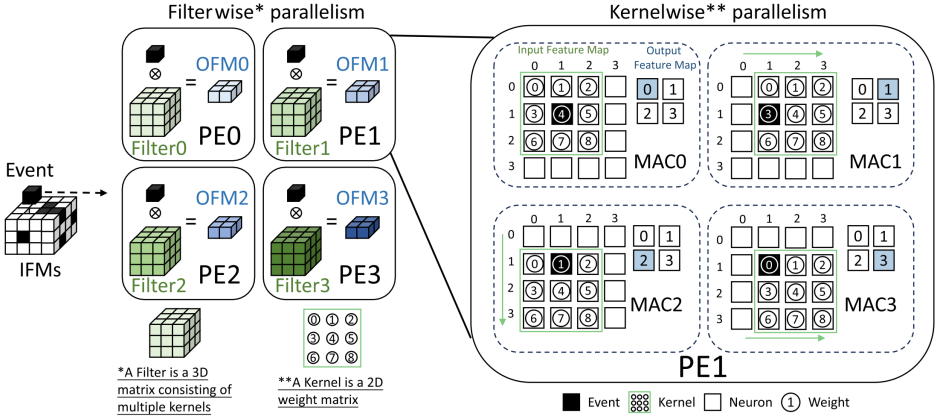
- Introduction
  - Background on CNN & Sparsity
- Prior Work & Motivation
- MnF Methodology
  - Event-driven Dataflow
  - Hardware Design
- Evaluation Setup & Results
- **Conclusion**

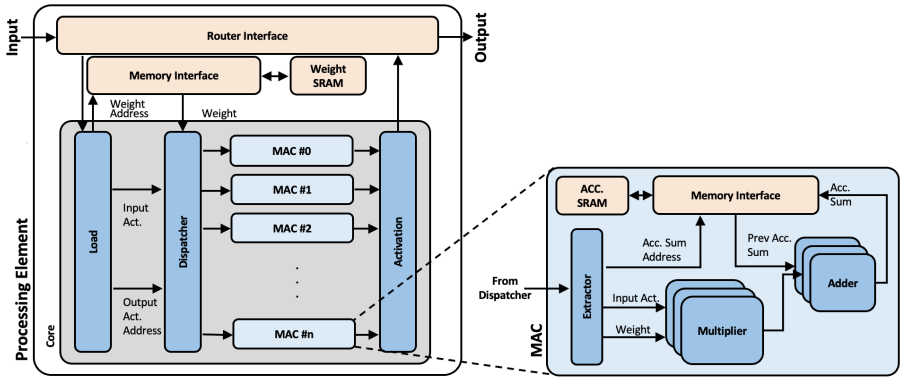
Conclusion – Multiply-and-Fire (MnF)

- A novel event-driven dataflow and an energy-efficient hardware accelerator for sparse DNN inference workloads
  - ✓ Only non-zero activations are processed
  - ✓ Non-zero activations are only accessed once.
  - ✓ Access weights regularly

- A geometric mean of **11.2×** more **energy efficiency** (inferences/J) on all evaluated models and **1.4× speedup** (inferences/second) on most of the evaluated models compared with the latest sparsity-aware DNN accelerator, CSP.

Q&A





- Exploiting activation sparsity
- Maximizing the reuse of activation data
- Designing an energy-efficient, high-performance sparsity-aware DNN accelerator
- Achieving 11.2× more energy efficiency and 1.41× speedup

Thanks!