MAERI 2.0

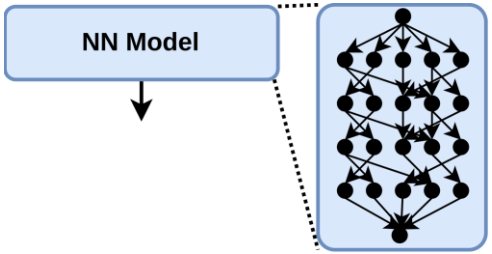
End-to-end framework to explore architecture design space on FPGA

Jianming Tong, Yangyu Chen, Yue Pan, Abhimanyu Bambhaniya, Taekyung Heo, Tushar Krishna
Georgia Institute of Technology
jianming.tong@gatech.edu

High-level Overview

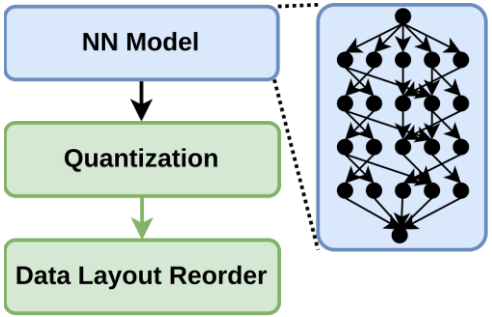
Overview: end2end framework enable NN inference on FPGA running MAERI 2.0

High-level Overview



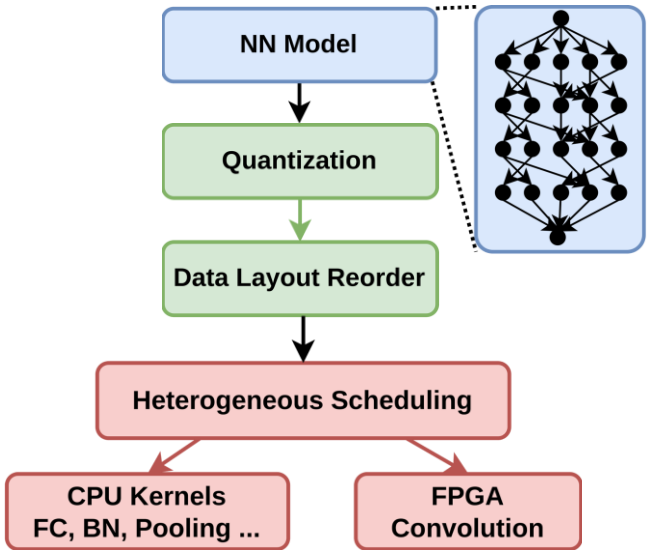
Overview: end2end framework enable NN inference on FPGA running MAERI 2.0

High-level Overview



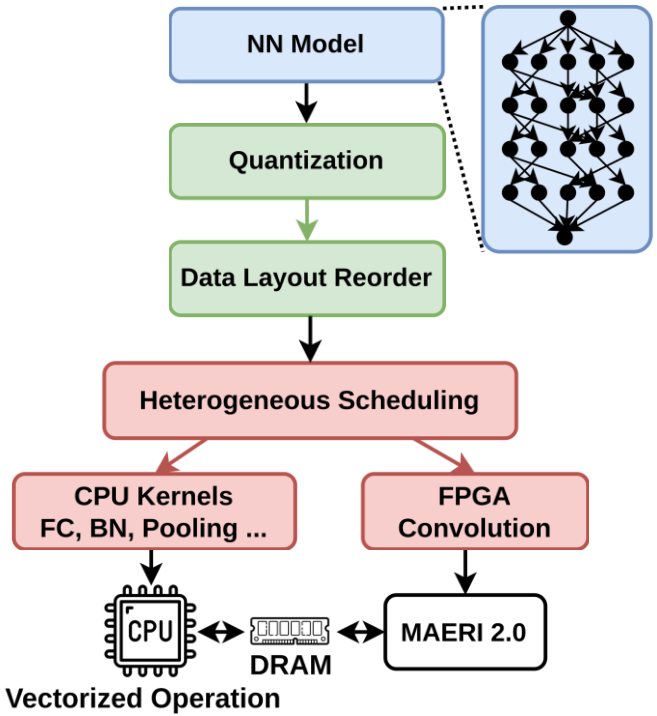
Overview: end2end framework enable NN inference on FPGA running MAERI 2.0

High-level Overview



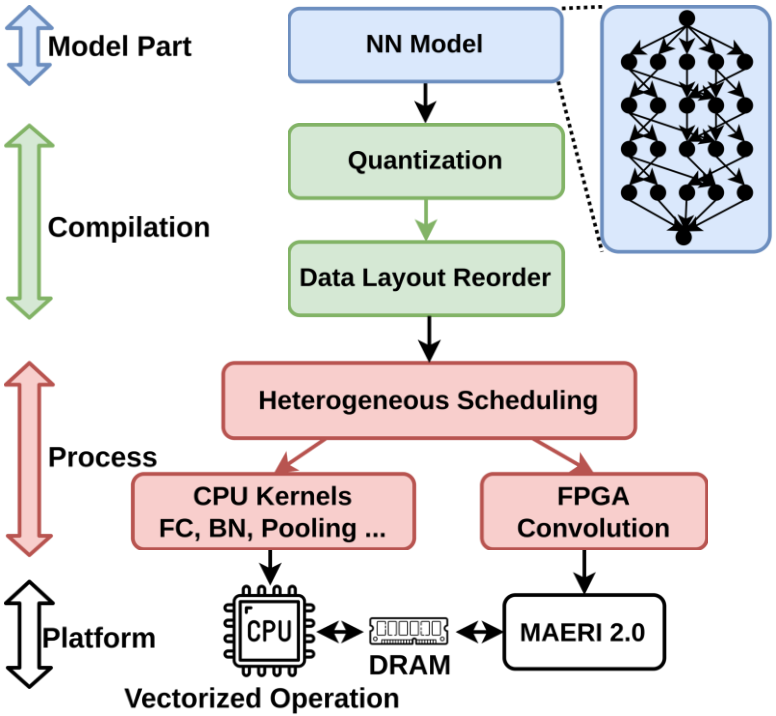
Overview: end2end framework enable NN inference on FPGA running MAERI 2.0

High-level Overview



Overview: end2end framework enable NN inference on FPGA running MAERI 2.0

High-level Overview

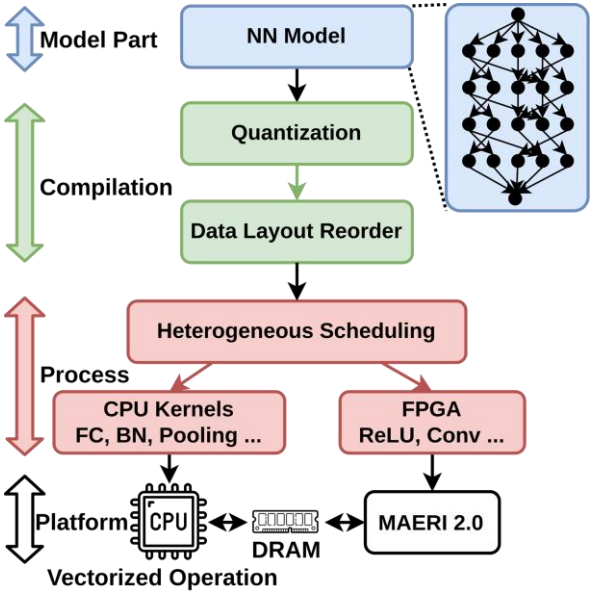


Overview: end2end framework enable NN inference on FPGA running MAERI 2.0

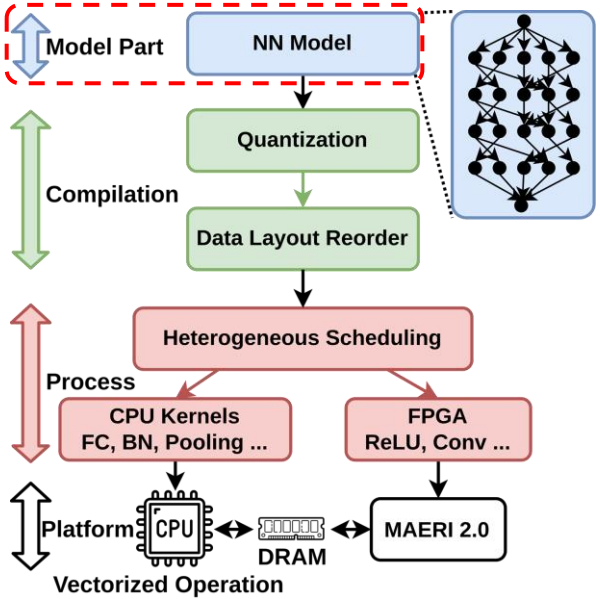
Outlines

- **Supported Neural Network Model**
- Quantization Flow
- Memory Layout
- Heterogeneous Scheduling
- MAERI 2.0 Microarchitecture
- DEMO

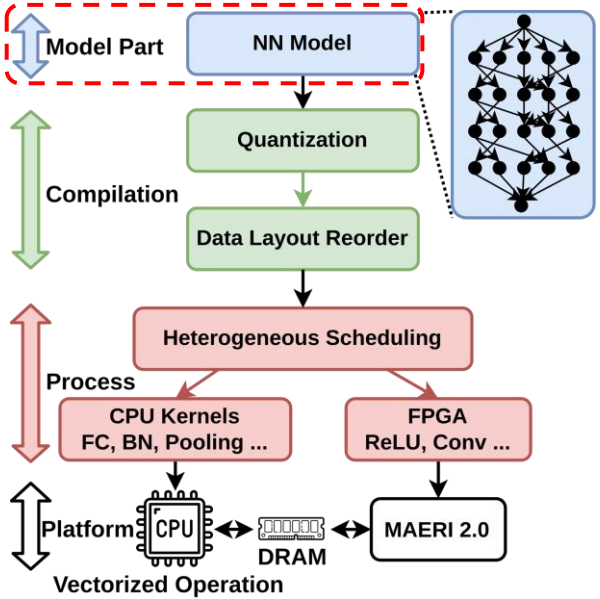
MAERI 2.0 Supported Neural Network Model



MAERI 2.0 Supported Neural Network Model

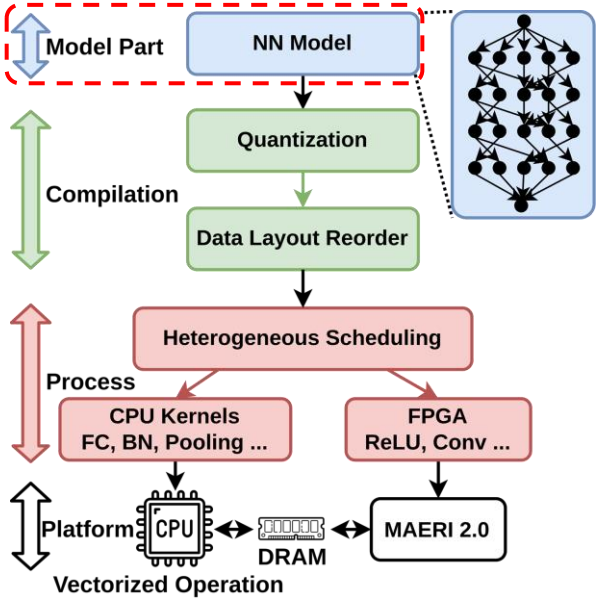


MAERI 2.0 Supported Neural Network Model



- Supported Models from **PyTorch** Framework

MAERI 2.0 Supported Neural Network Model



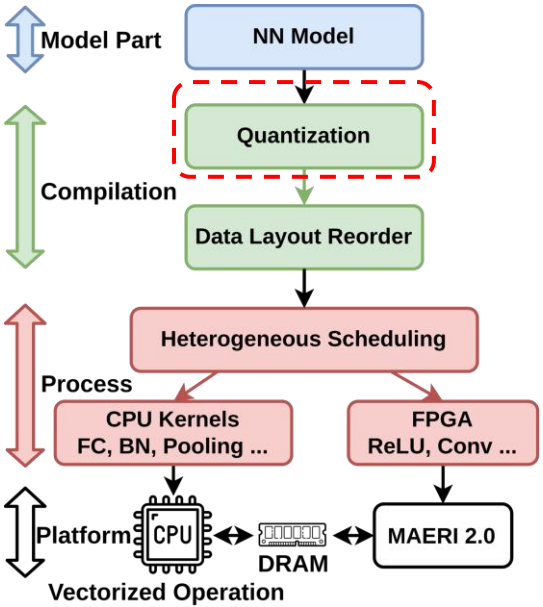
- Supported Models from **PyTorch** Framework

Layer/Feature	Attribute	Range
Convolution	Kernel Sizes	w, h: 1, 3
	Strides	w, h: 1, 2
	Padding	w: [0, kernel_w-1] h: [0, kernel_h-1]
	Input Size	Arbitrary
	Input Channel	Arbitrary
	Output Channel	Arbitrary
	Activation	ReLU, ReLU6 and LeakyReLU
	Dilation	Future Work
Max Pooling/Average Pooling	Kernel Sizes	Arbitrary
	Strides	Arbitrary
	Padding	Arbitrary
Fully Connected	Input_channel	Arbitrary
	Output_channel	Arbitrary
Skip Add	Distance	Arbitrary

Outlines

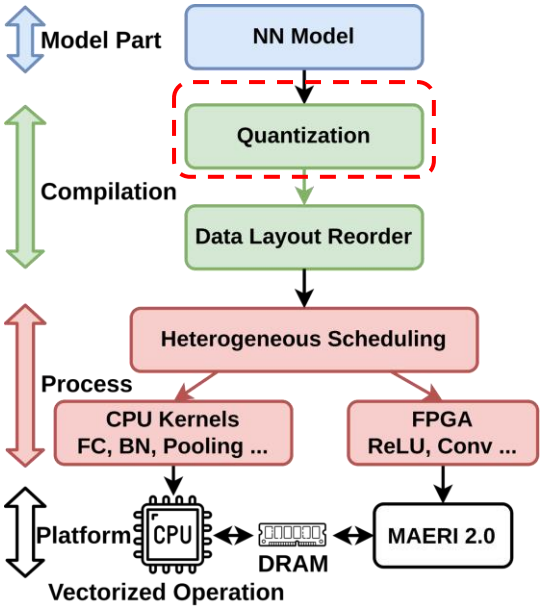
- Supported Neural Network Model
- **Quantization Flow**
- Memory Layout
- Heterogeneous Scheduling
- MAERI 2.0 Microarchitecture
- DEMO

MAERI 2.0 Quantization Illustration



[1] C. Torres-Huitzil and B. Girau, "Fault and Error Tolerance in Neural Networks: A Review," in IEEE Access, vol. 5, pp. 17322-17341, 2017, doi: 10.1109/ACCESS.2017.2742698.
[2] Jacob, Benoit, et al. ”Quantization and training of neural networks for efficient integer-arithmetic-only inference.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2018

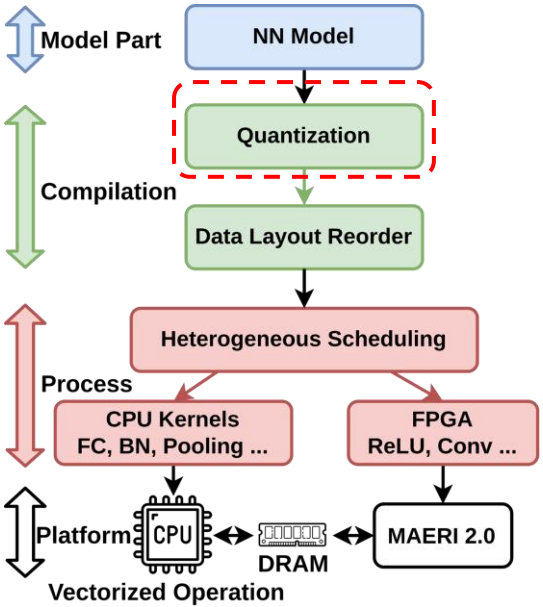
MAERI 2.0 Quantization Illustration



- Neural Network are error-tolerant [1]
  - Use less precision with little scarifice of accuracy - save compute [2]

[1] C. Torres-Huitzil and B. Girau, "Fault and Error Tolerance in Neural Networks: A Review," in IEEE Access, vol. 5, pp. 17322-17341, 2017, doi: 10.1109/ACCESS.2017.2742698.
[2] Jacob, Benoit, et al. ”Quantization and training of neural networks for efficient integer-arithmetic-only inference.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2018

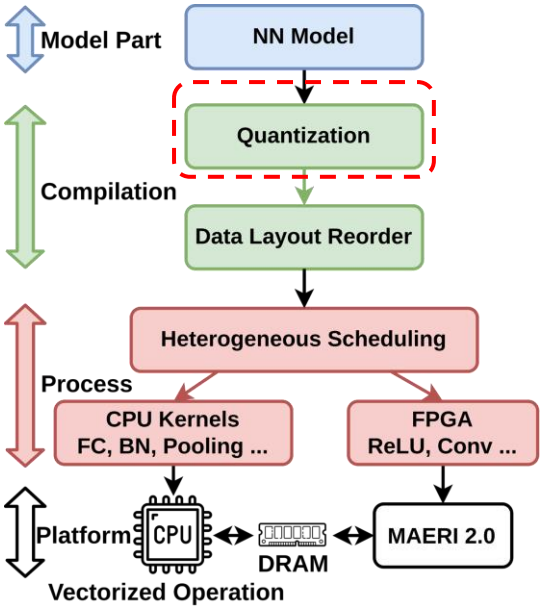
MAERI 2.0 Quantization Illustration



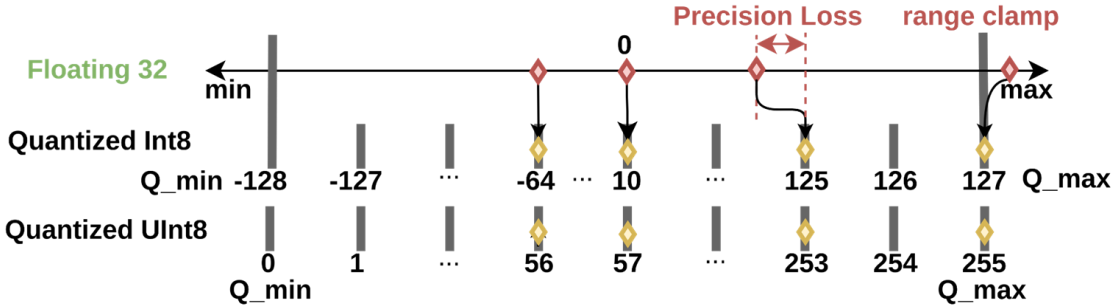
- Neural Network are error-tolerant [1]
  - Use less precision with little scarifice of accuracy - save compute [2]

[1] C. Torres-Huitzil and B. Girau, "Fault and Error Tolerance in Neural Networks: A Review," in IEEE Access, vol. 5, pp. 17322-17341, 2017, doi: 10.1109/ACCESS.2017.2742698.
[2] Jacob, Benoit, et al. ”Quantization and training of neural networks for efficient integer-arithmetic-only inference.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2018

MAERI 2.0 Quantization Illustration

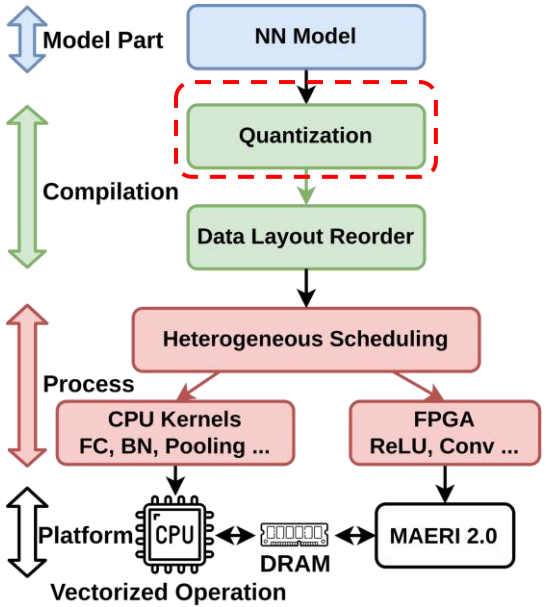


- Neural Network are error-tolerant [1]
  - Use less precision with little scarifice of accuracy - save compute [2]

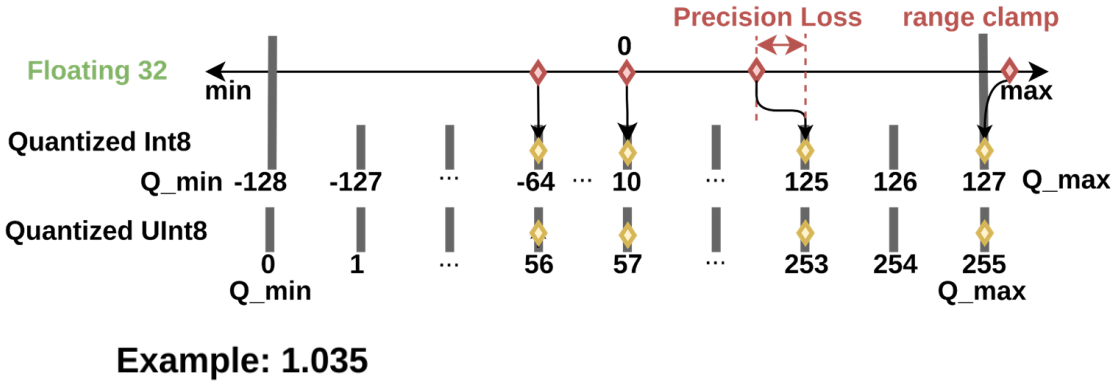


[1] C. Torres-Huitzil and B. Girau, "Fault and Error Tolerance in Neural Networks: A Review," in IEEE Access, vol. 5, pp. 17322-17341, 2017, doi: 10.1109/ACCESS.2017.2742698.
[2] Jacob, Benoit, et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018

MAERI 2.0 Quantization Illustration



- Neural Network are error-tolerant [1]
  - Use less precision with little scarifice of accuracy - save compute [2]

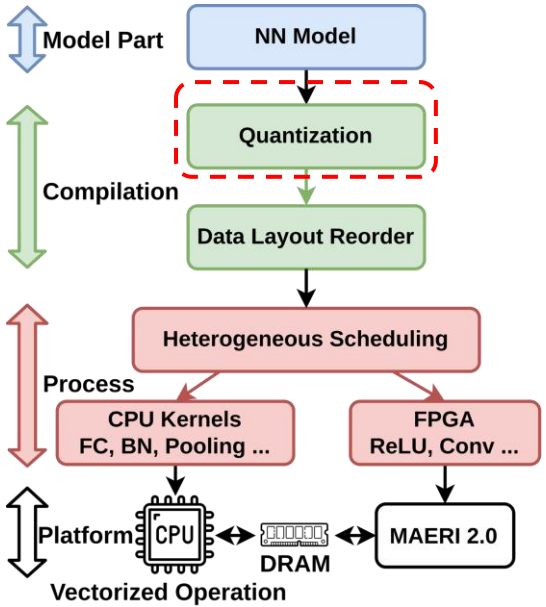


Example: 1.035

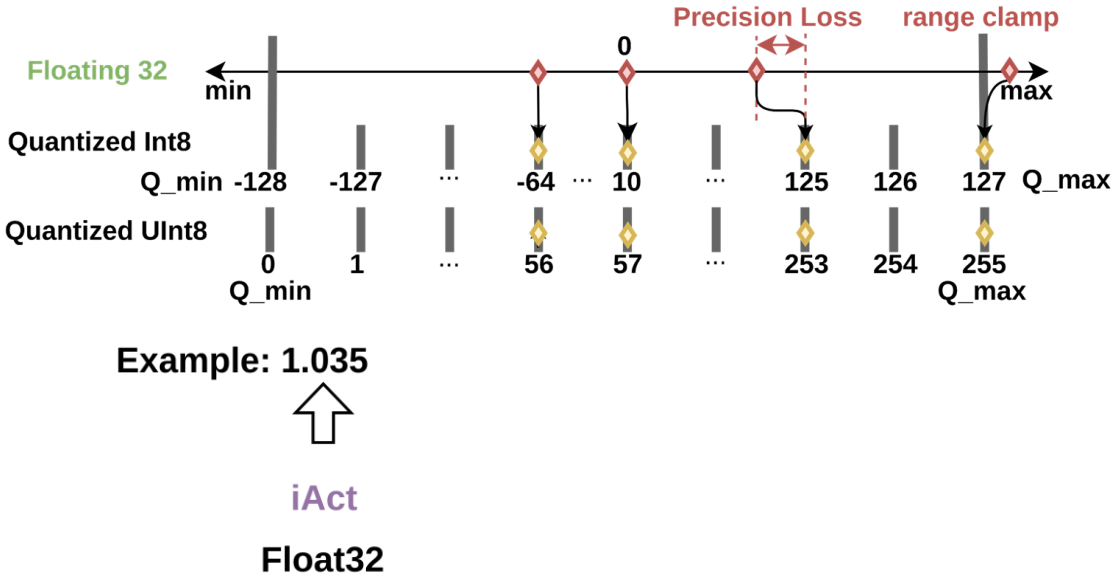
[1] C. Torres-Huitzil and B. Girau, "Fault and Error Tolerance in Neural Networks: A Review," in IEEE Access, vol. 5, pp. 17322-17341, 2017, doi: 10.1109/ACCESS.2017.2742698.

[2] Jacob, Benoit, et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018

MAERI 2.0 Quantization Illustration



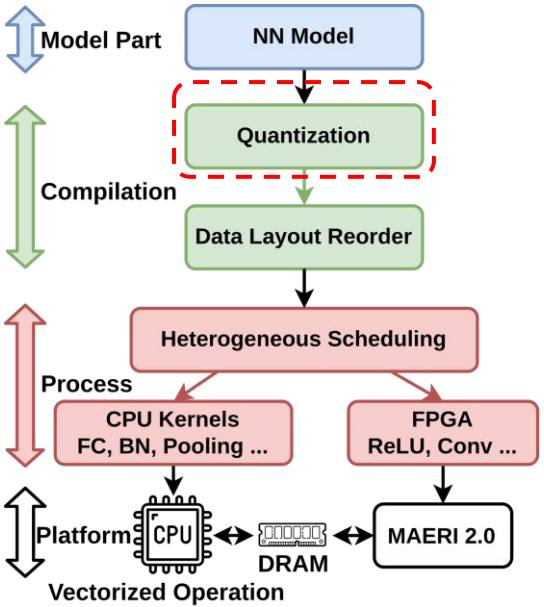
- Neural Network are error-tolerant [1]
  - Use less precision with little scarifice of accuracy - save compute [2]



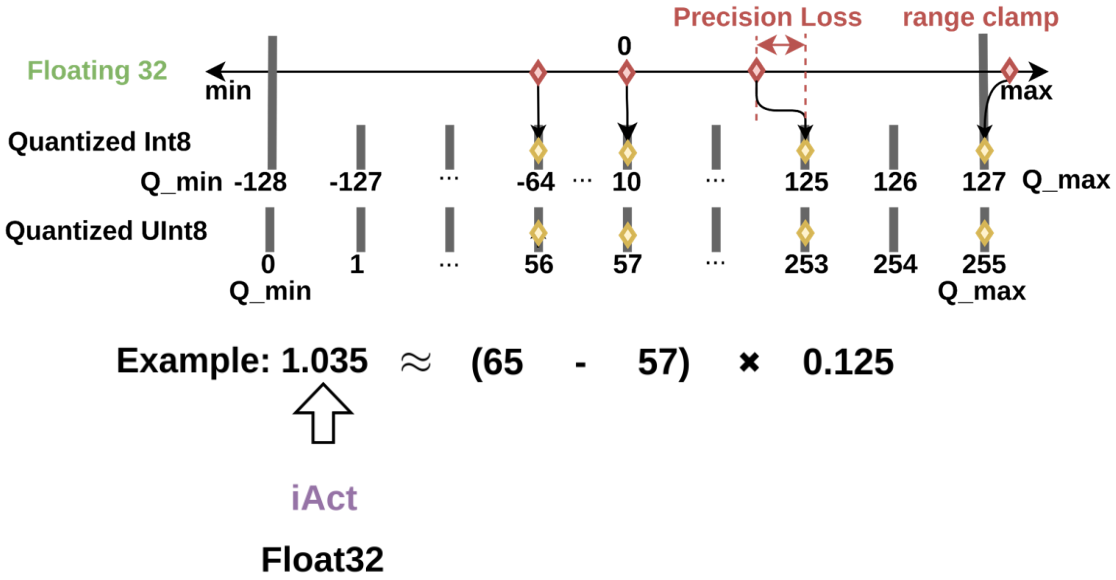
[1] C. Torres-Huitzil and B. Girau, "Fault and Error Tolerance in Neural Networks: A Review," in IEEE Access, vol. 5, pp. 17322-17341, 2017, doi: 10.1109/ACCESS.2017.2742698.

[2] Jacob, Benoit, et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018

MAERI 2.0 Quantization Illustration



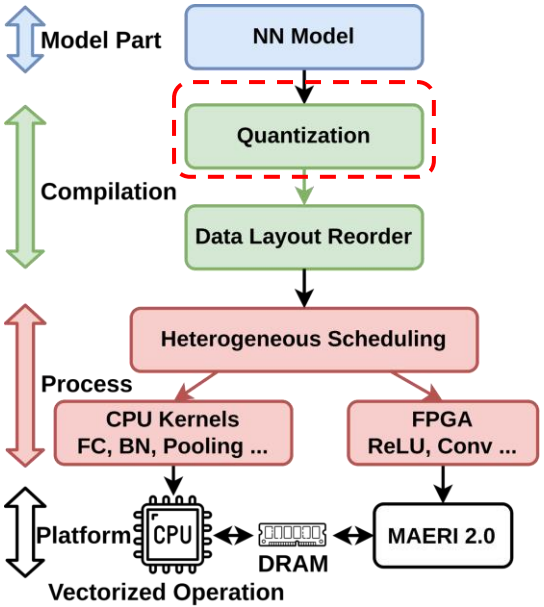
- Neural Network are error-tolerant [1]
  - Use less precision with little scarifice of accuracy - save compute [2]



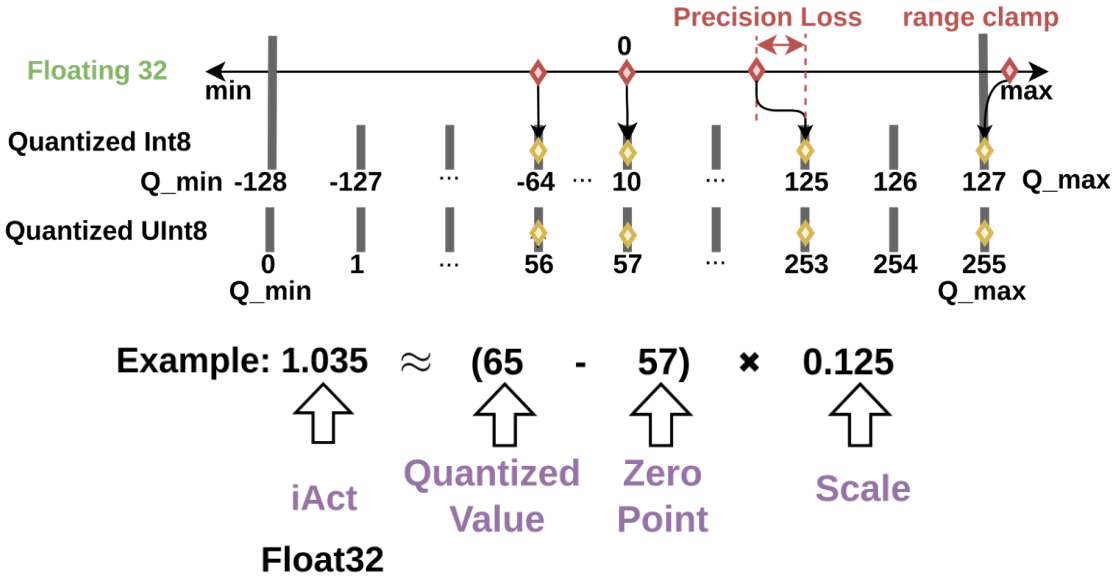
[1] C. Torres-Huitzil and B. Girau, "Fault and Error Tolerance in Neural Networks: A Review," in IEEE Access, vol. 5, pp. 17322-17341, 2017, doi: 10.1109/ACCESS.2017.2742698.

[2] Jacob, Benoit, et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018

MAERI 2.0 Quantization Illustration



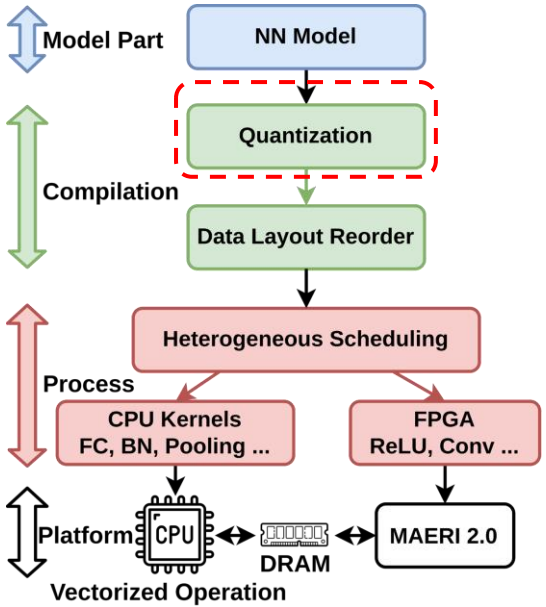
- Neural Network are error-tolerant [1]
  - Use less precision with little scarifice of accuracy - save compute [2]



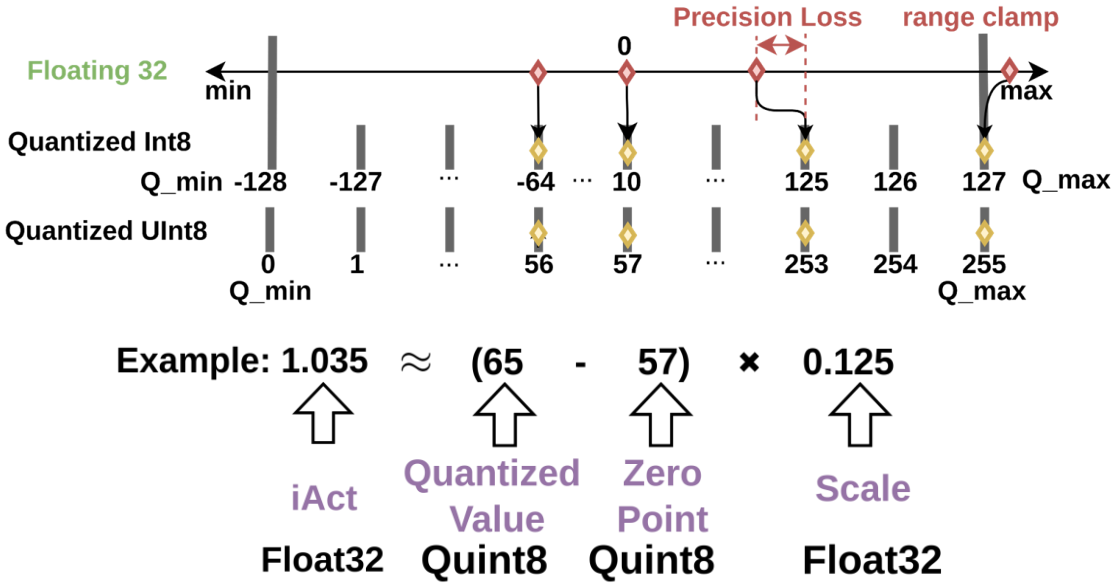
[1] C. Torres-Huitzil and B. Girau, "Fault and Error Tolerance in Neural Networks: A Review," in IEEE Access, vol. 5, pp. 17322-17341, 2017, doi: 10.1109/ACCESS.2017.2742698.

[2] Jacob, Benoit, et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018

MAERI 2.0 Quantization Illustration

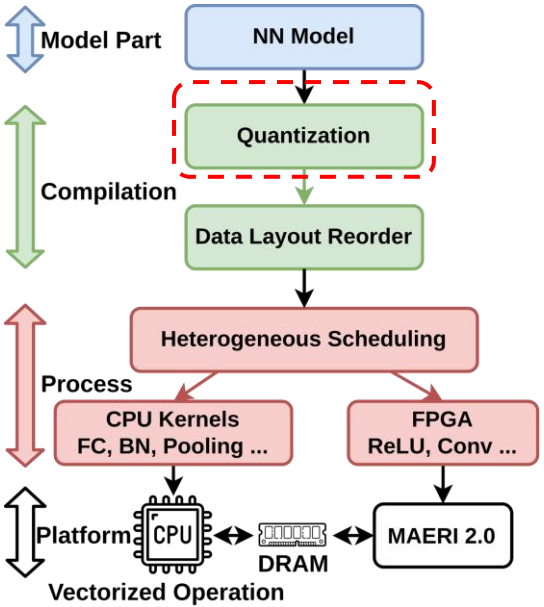


- Neural Network are error-tolerant [1]
  - Use less precision with little scarifice of accuracy - save compute [2]

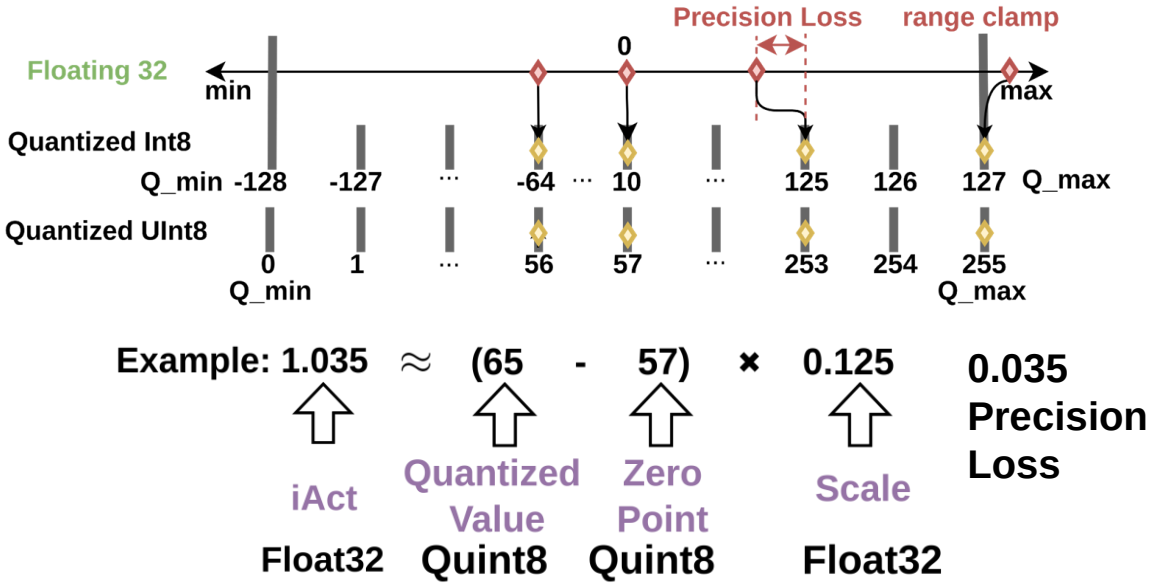


[1] C. Torres-Huitzil and B. Girau, "Fault and Error Tolerance in Neural Networks: A Review," in IEEE Access, vol. 5, pp. 17322-17341, 2017, doi: 10.1109/ACCESS.2017.2742698.
[2] Jacob, Benoit, et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018

MAERI 2.0 Quantization Illustration



- Neural Network are error-tolerant [1]
  - Use less precision with little scarifice of accuracy - save compute [2]



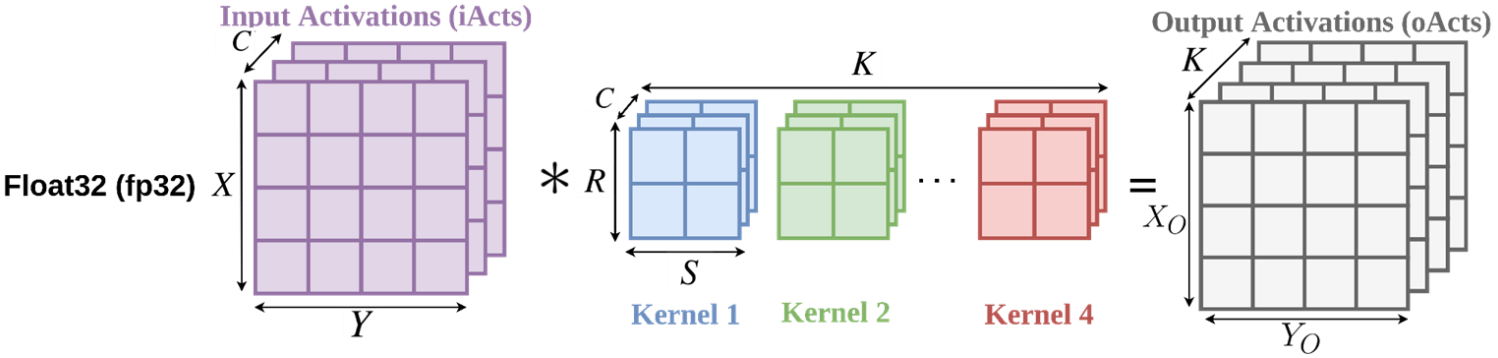
[1] C. Torres-Huitzil and B. Girau, "Fault and Error Tolerance in Neural Networks: A Review," in IEEE Access, vol. 5, pp. 17322-17341, 2017, doi: 10.1109/ACCESS.2017.2742698.
[2] Jacob, Benoit, et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018

Outlines

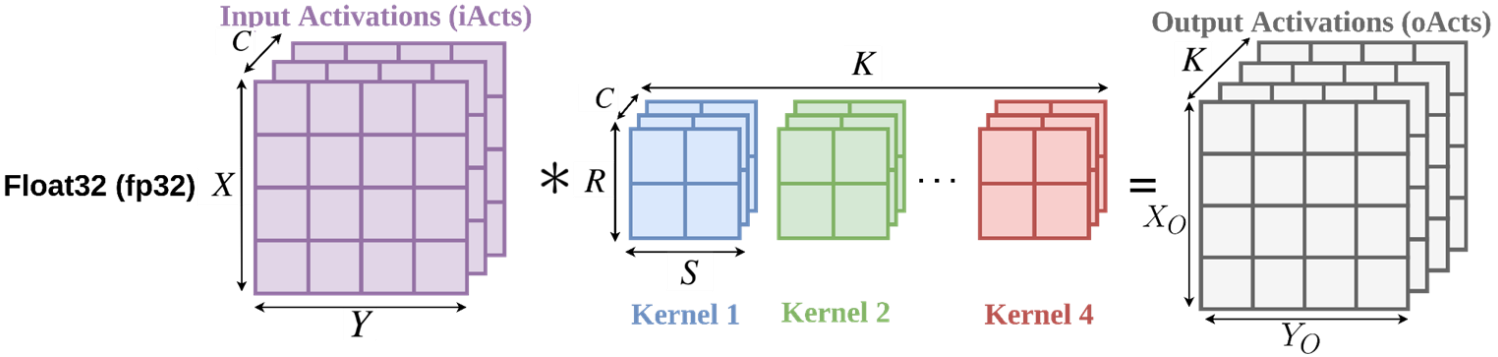
- Supported Neural Network Model
- Quantization Flow
- **Memory Layout**
- Heterogeneous Scheduling
- MAERI 2.0 Microarchitecture
- DEMO

MAERI 2.0 Model Terminology

MAERI 2.0 Model Terminology

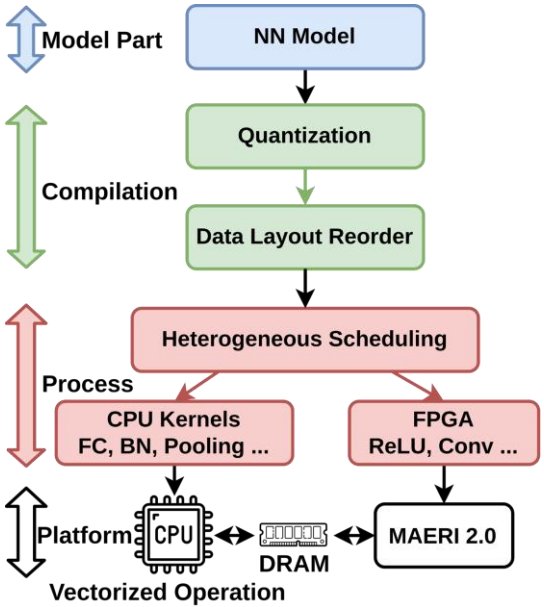


MAERI 2.0 Model Terminology

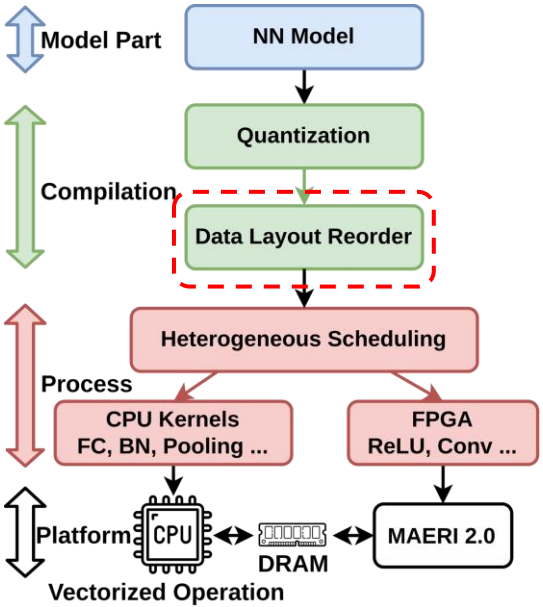


Terminology	Overall
Input Channel	C
Input Height	X
Input Width	Y
Kernel Number	K
Weight Height	R
Weight Width	S
Output Height	X_O
Output Width	Y_O

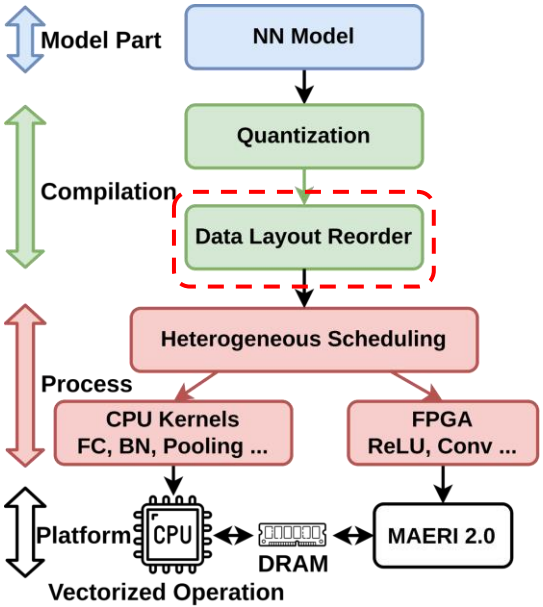
MAERI 2.0 Data Layout



MAERI 2.0 Data Layout

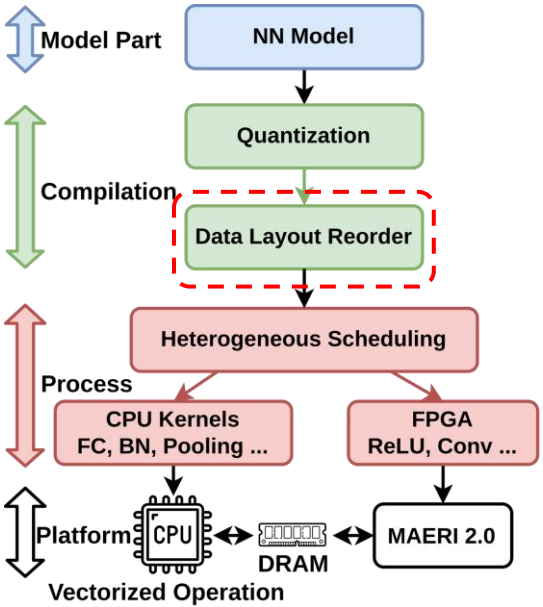


MAERI 2.0 Data Layout

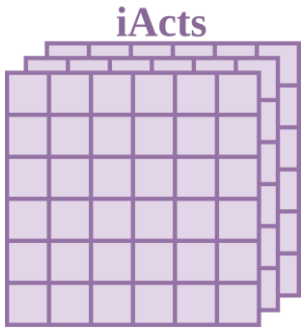


- DRAM is 1D -> each address refers to a single data.

MAERI 2.0 Data Layout

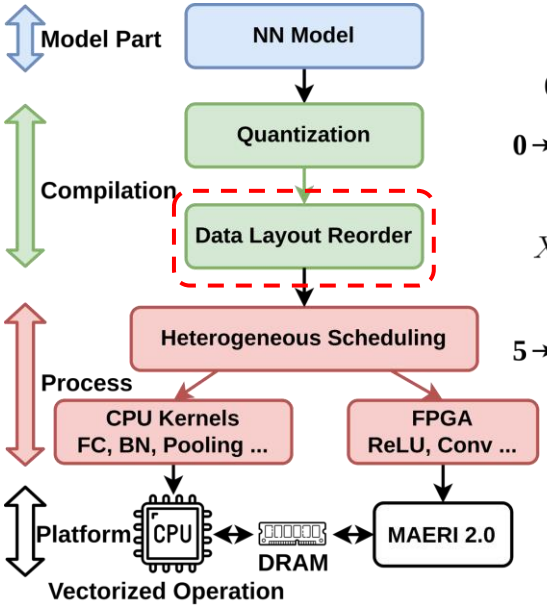


- DRAM is 1D -> each address refers to a single data.

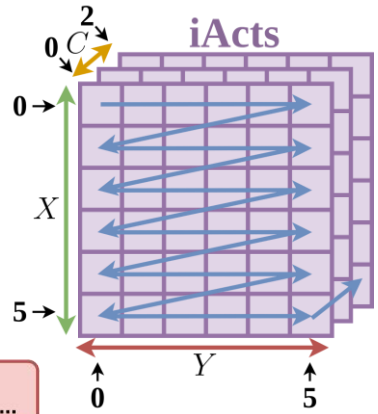


iAct Layout in DRAM

MAERI 2.0 Data Layout

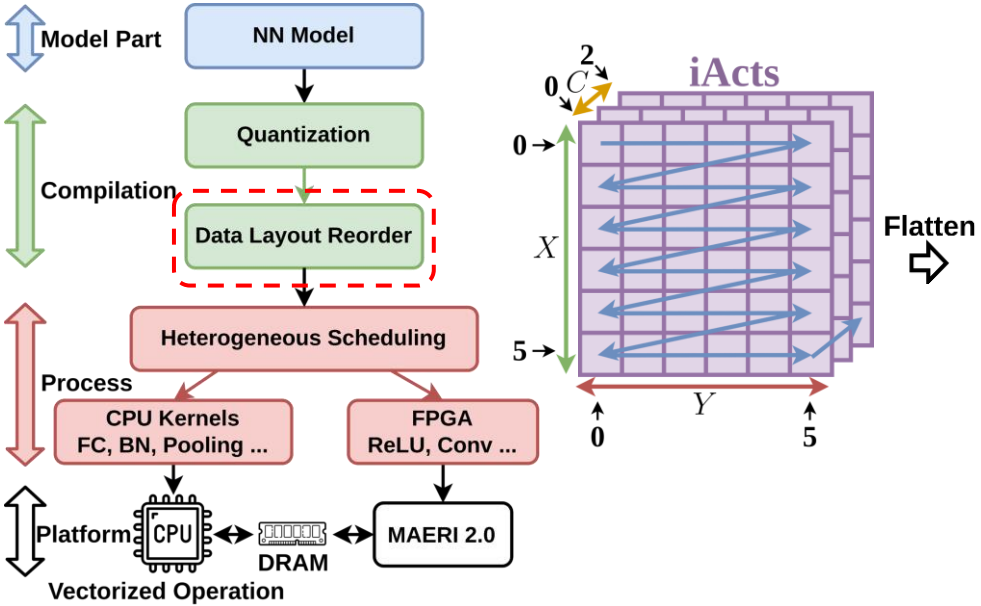


- DRAM is 1D -> each address refers to a single data.



iAct Layout in DRAM

MAERI 2.0 Data Layout

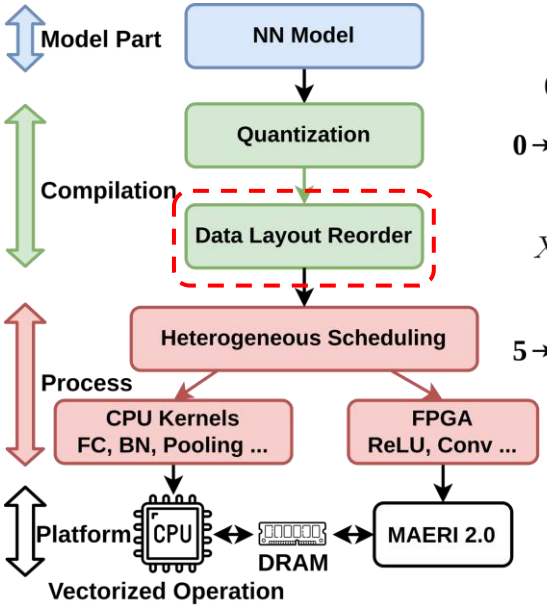


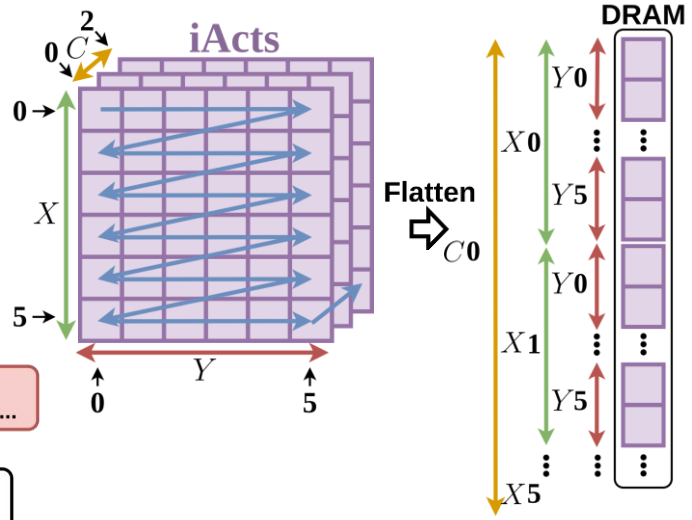
- DRAM is 1D -> each address refers to a single data.

iAct Layout in DRAM

MAERI 2.0 Data Layout

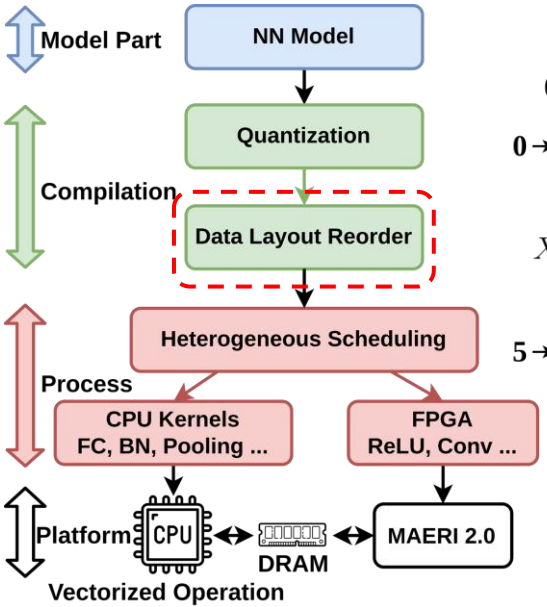
- DRAM is 1D -> each address refers to a single data.



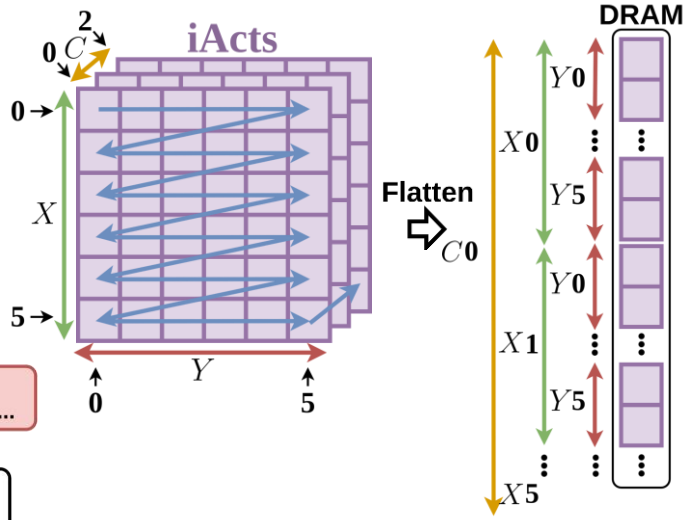


iAct Layout in DRAM

MAERI 2.0 Data Layout

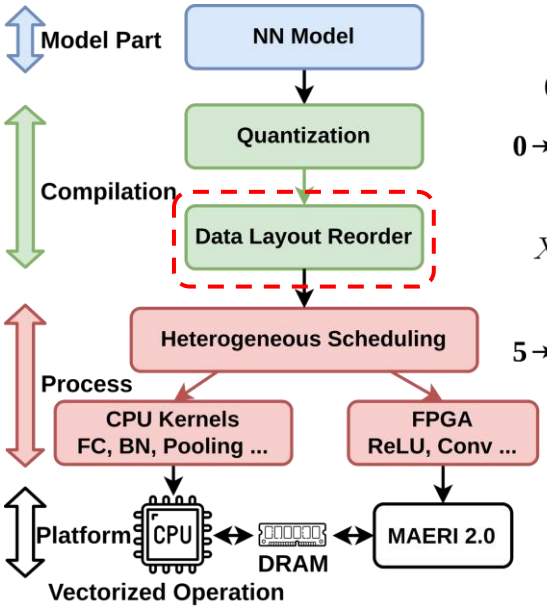


- DRAM is 1D -> each address refers to a single data.

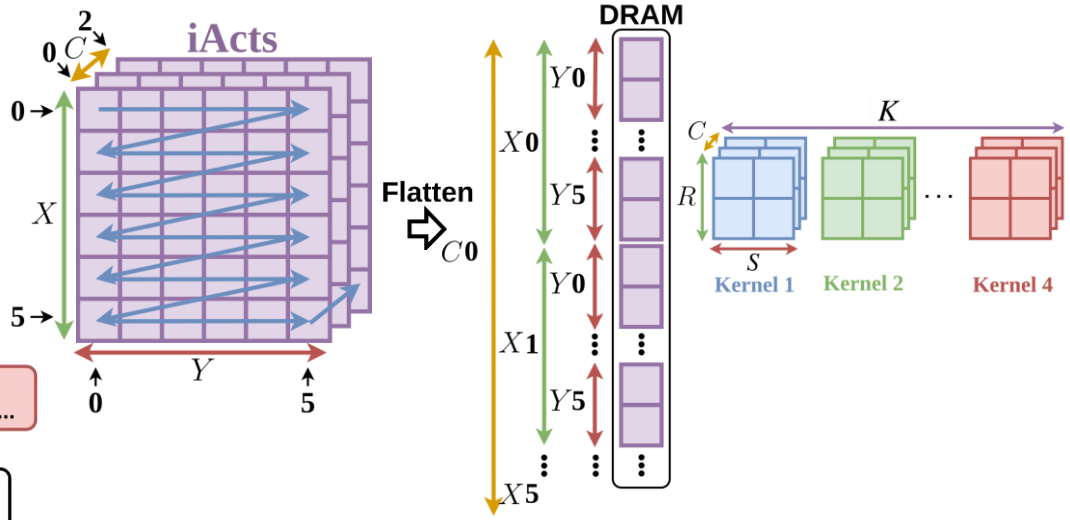


iAct Layout in DRAM **Weights Layout in DRAM**

MAERI 2.0 Data Layout

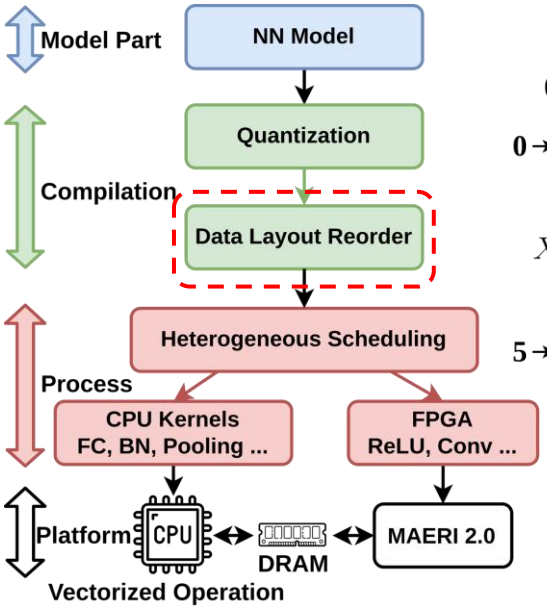


- DRAM is 1D -> each address refers to a single data.

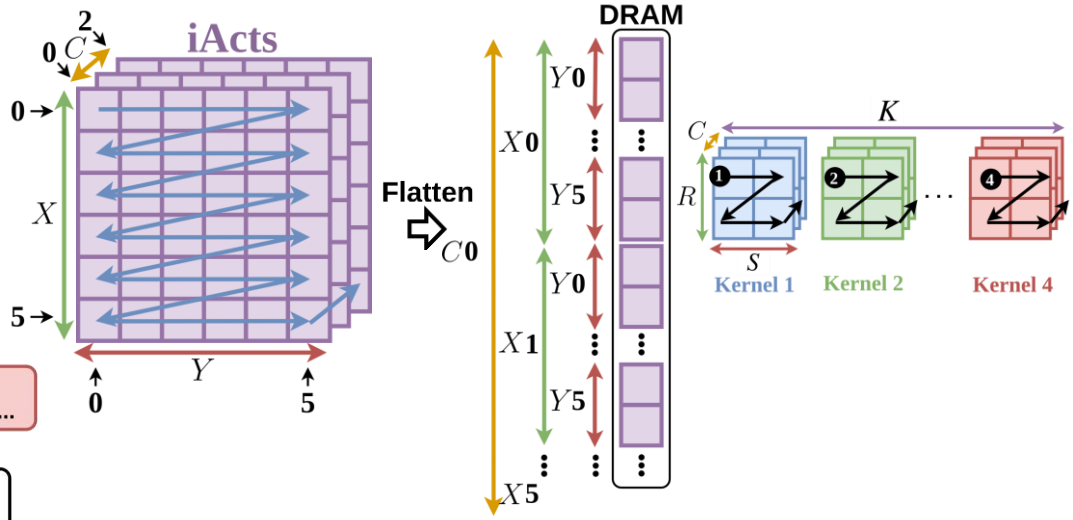


iAct Layout in DRAM **Weights Layout in DRAM**

MAERI 2.0 Data Layout

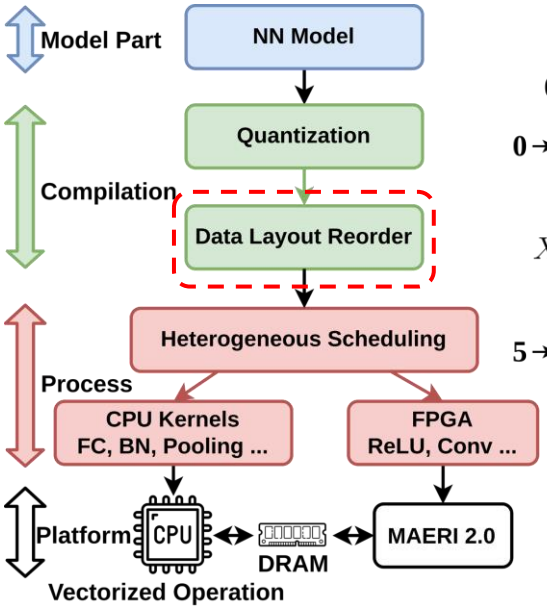


- DRAM is 1D -> each address refers to a single data.

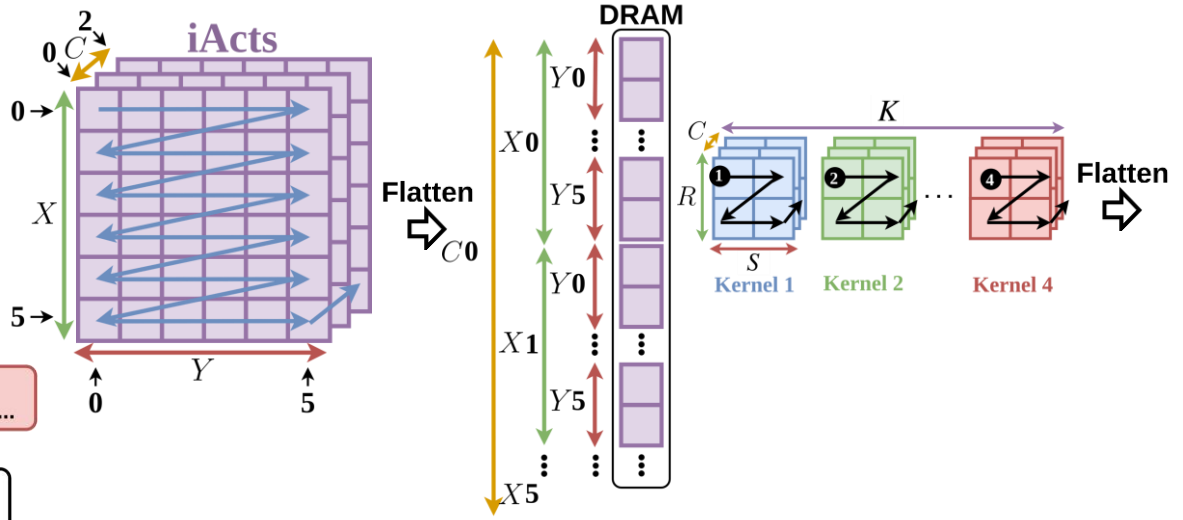


iAct Layout in DRAM **Weights Layout in DRAM**

MAERI 2.0 Data Layout



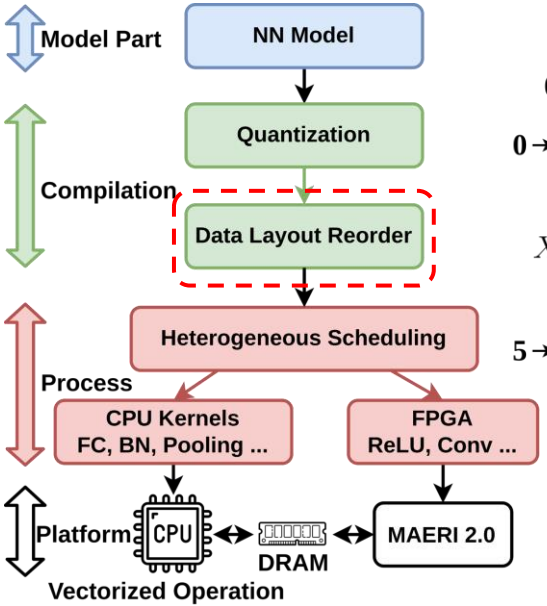
- DRAM is 1D -> each address refers to a single data.

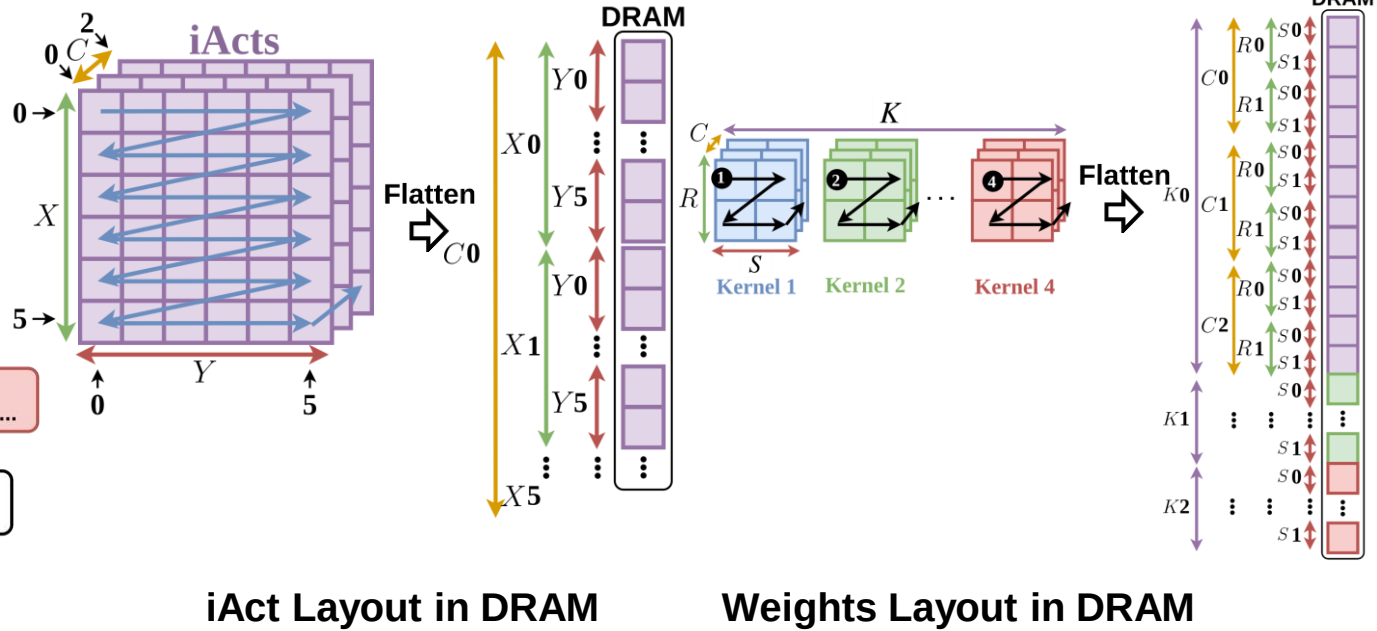


iAct Layout in DRAM **Weights Layout in DRAM**

MAERI 2.0 Data Layout

- DRAM is 1D -> each address refers to a single data.



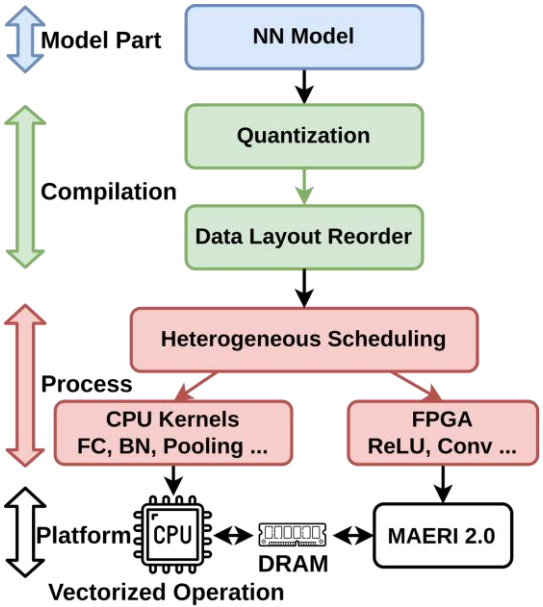


iAct Layout in DRAM **Weights Layout in DRAM**

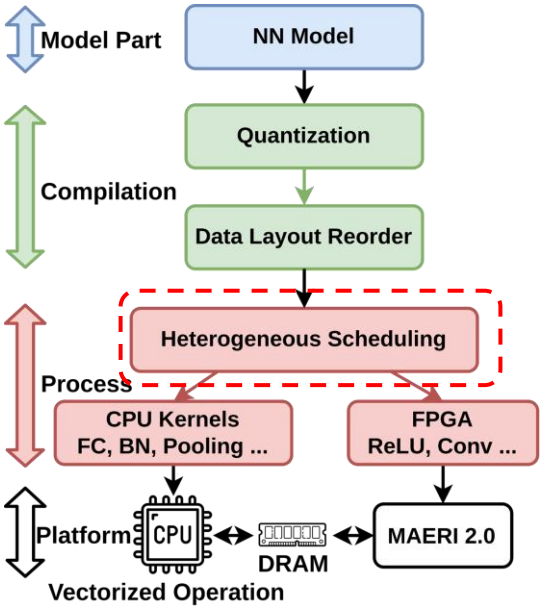
Outlines

- Supported Neural Network Model
- Quantization Flow
- Memory Layout
- **Heterogeneous Scheduling**
- MAERI 2.0 Microarchitecture
- DEMO

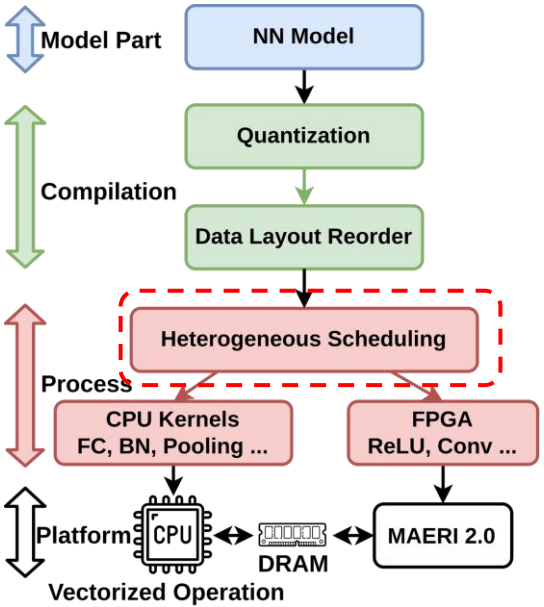
MAERI 2.0 Operators Scheduling



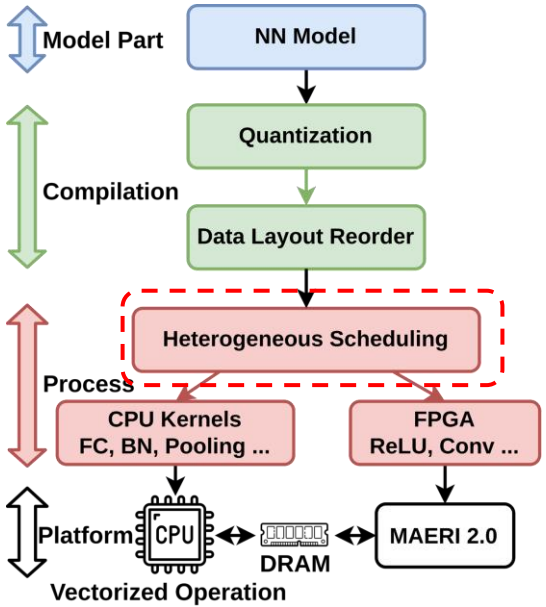
MAERI 2.0 Operators Scheduling



MAERI 2.0 Operators Scheduling

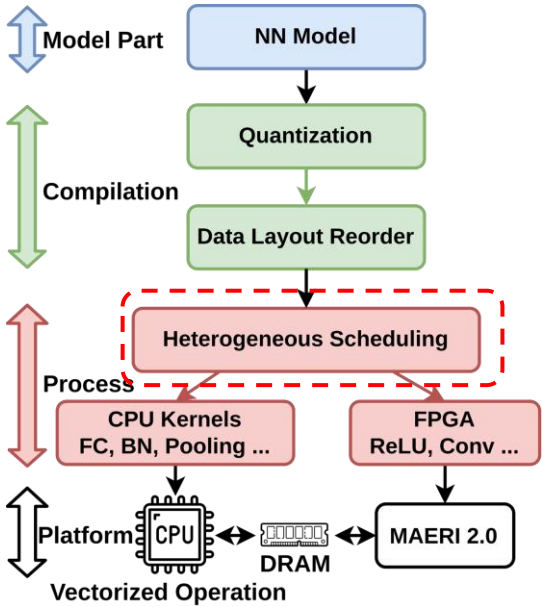


MAERI 2.0 Operators Scheduling

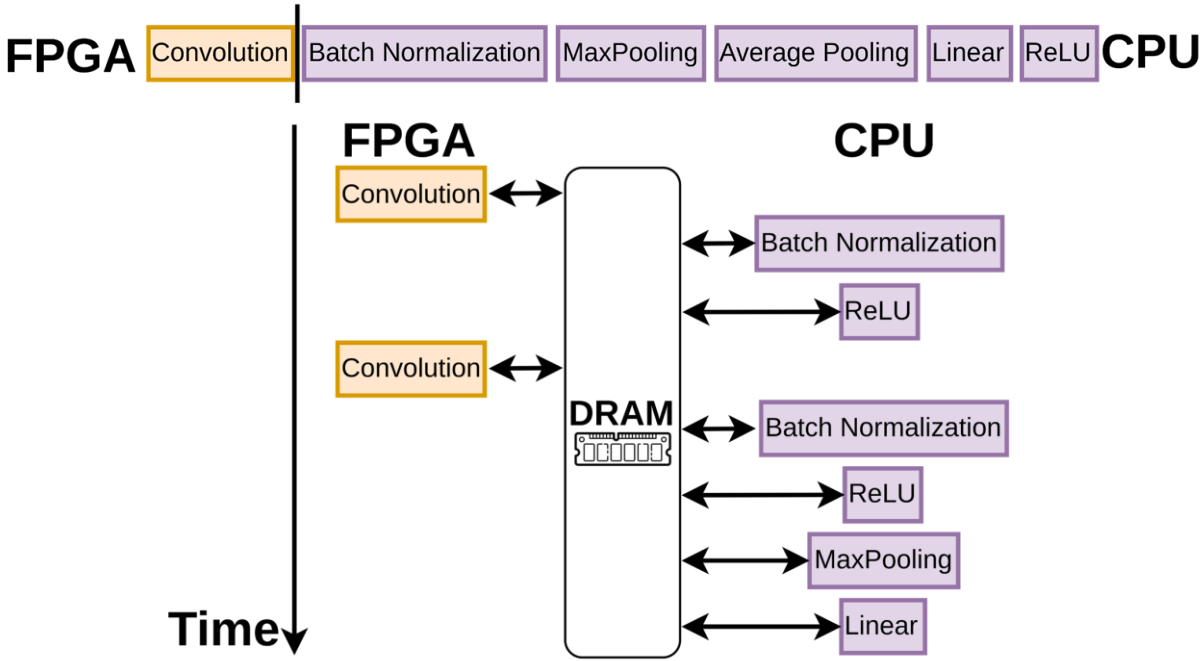


- Heterogeneous Scheduling

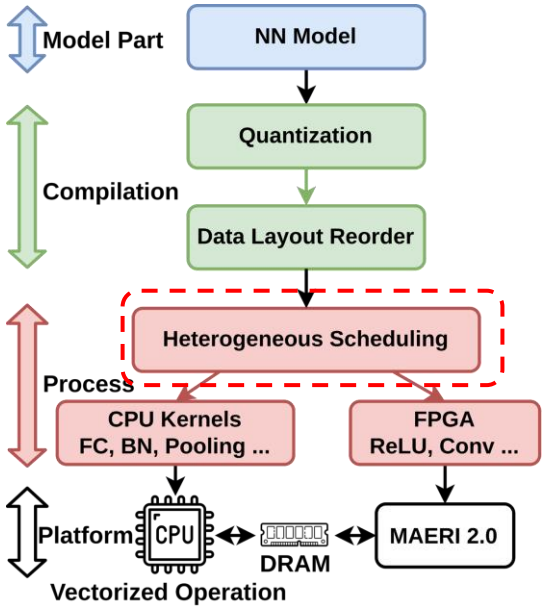
MAERI 2.0 Operators Scheduling



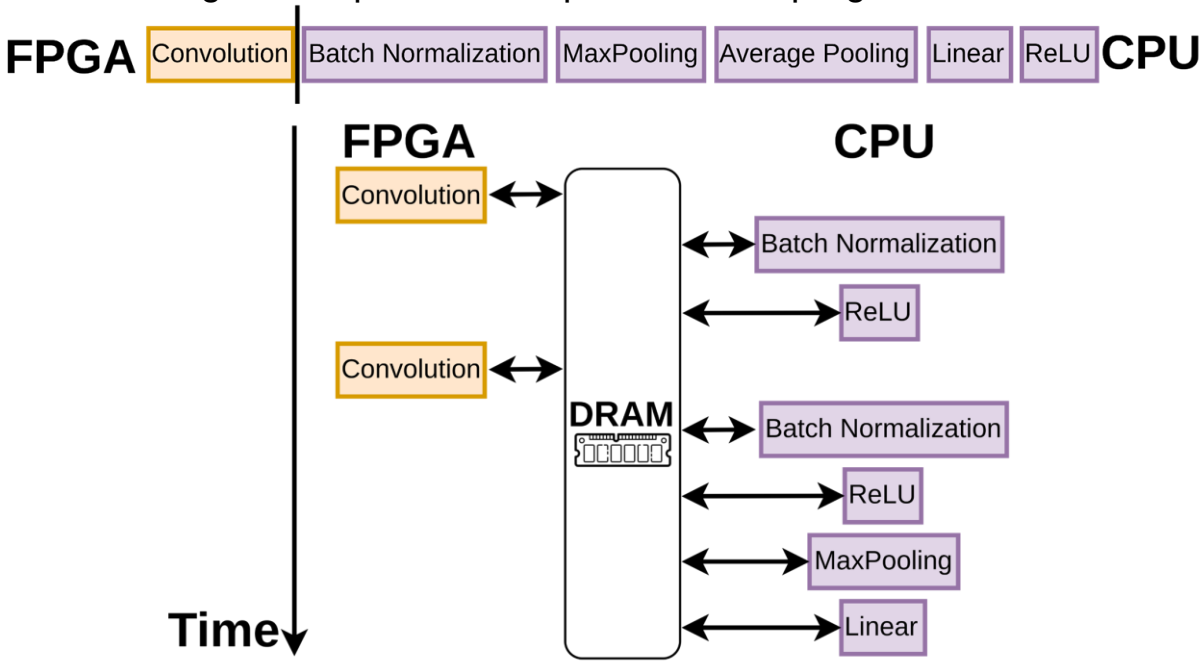
- Heterogeneous Scheduling



MAERI 2.0 Operators Scheduling



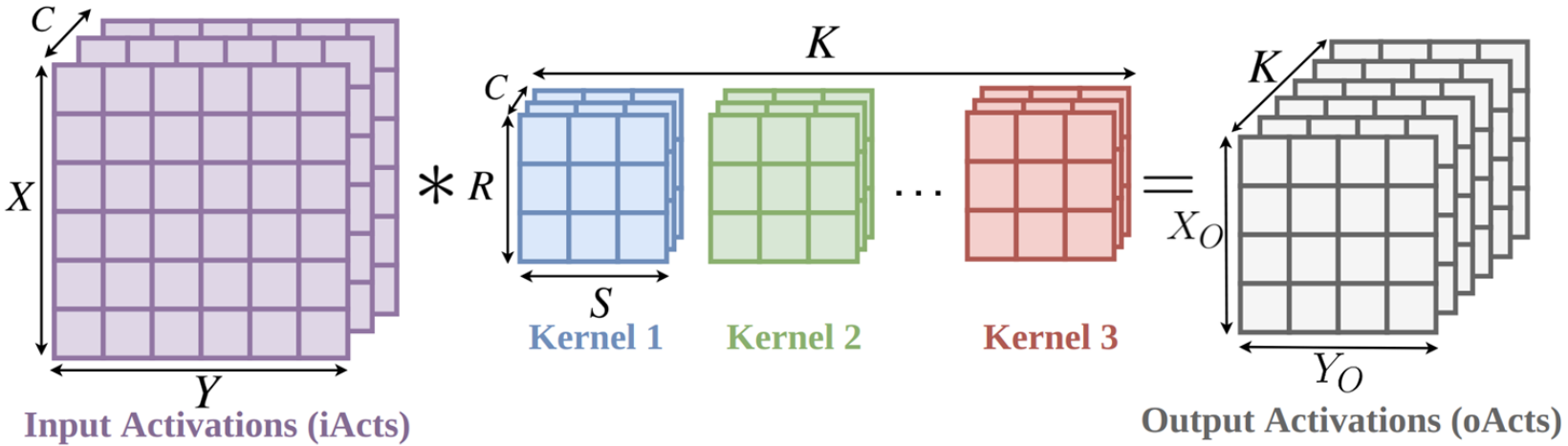
- Heterogeneous Scheduling
  - Heterogeneous parallelism optimization in progress



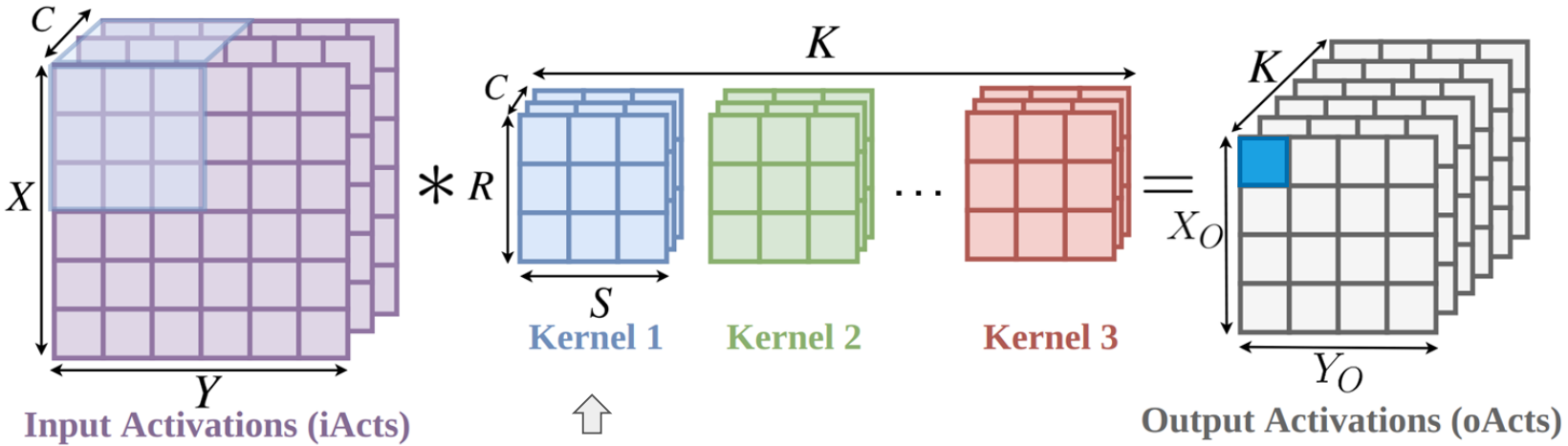
Outlines

- Supported Neural Network Model
- Quantization Flow
- Memory Layout
- Heterogeneous Scheduling
- **MAERI 2.0 Microarchitecture**
  - **Data Processing Order**
  - Microarchitecture
- DEMO

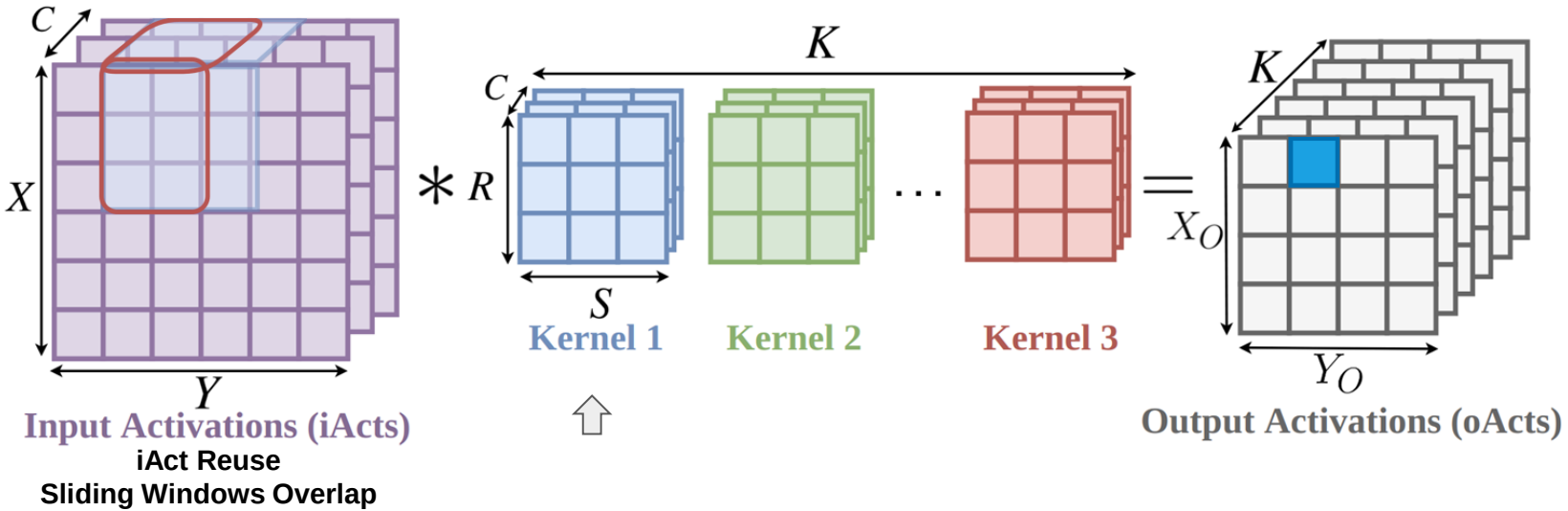
Data Processing Order - Under Weights Stationary



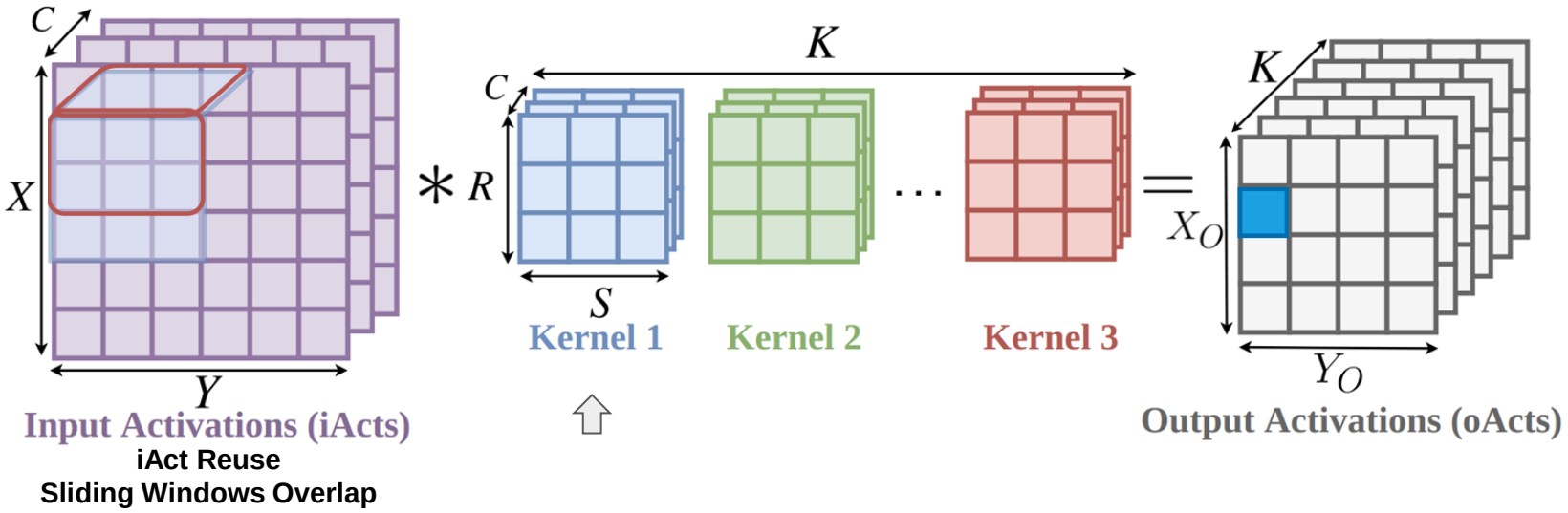
Data Processing Order - Under Weights Stationary



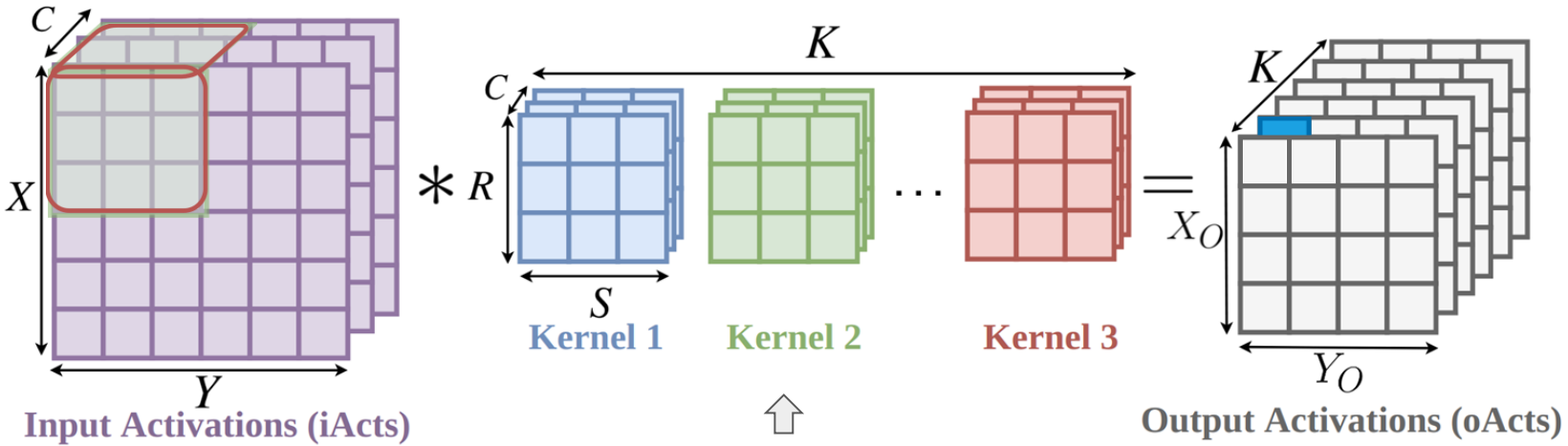
Data Processing Order - Under Weights Stationary



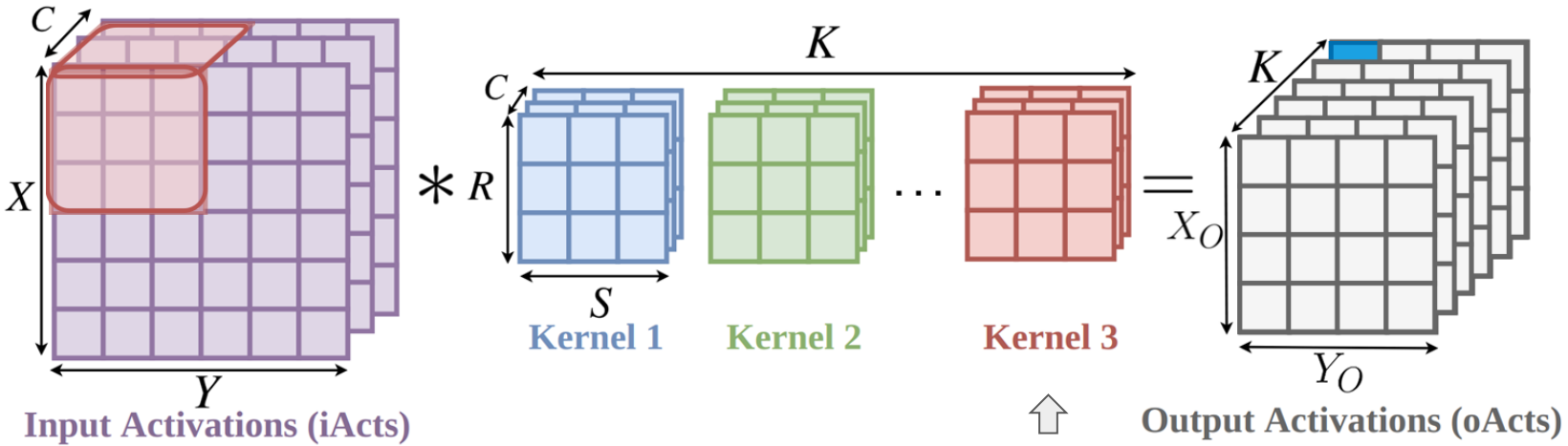
Data Processing Order - Under Weights Stationary



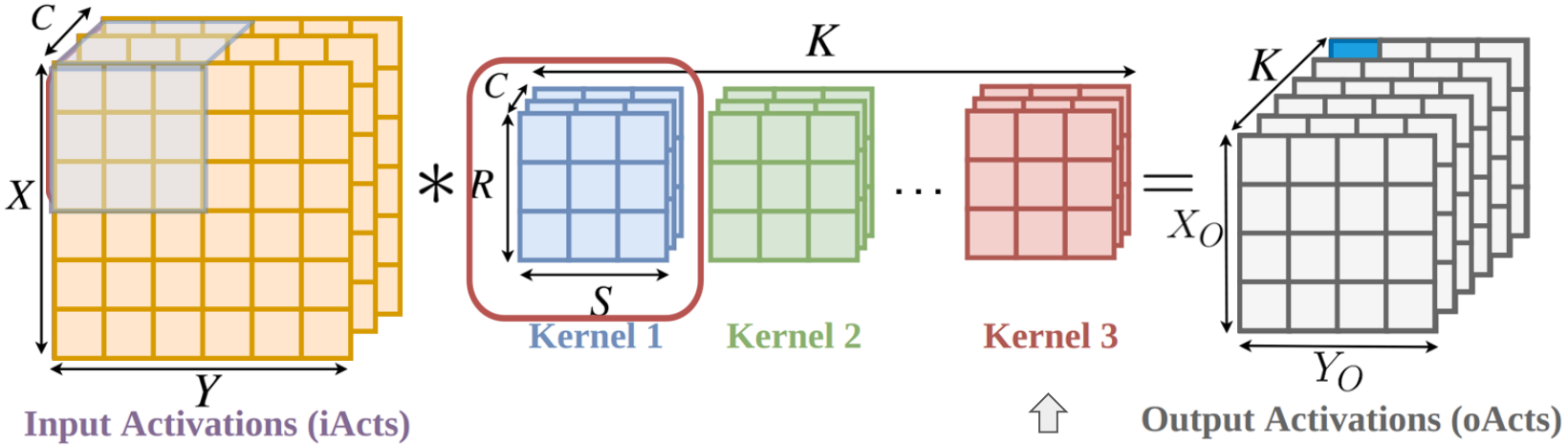
Data Processing Order - Under Weights Stationary



Data Processing Order - Under Weights Stationary



Data Processing Order - Under Weights Stationary

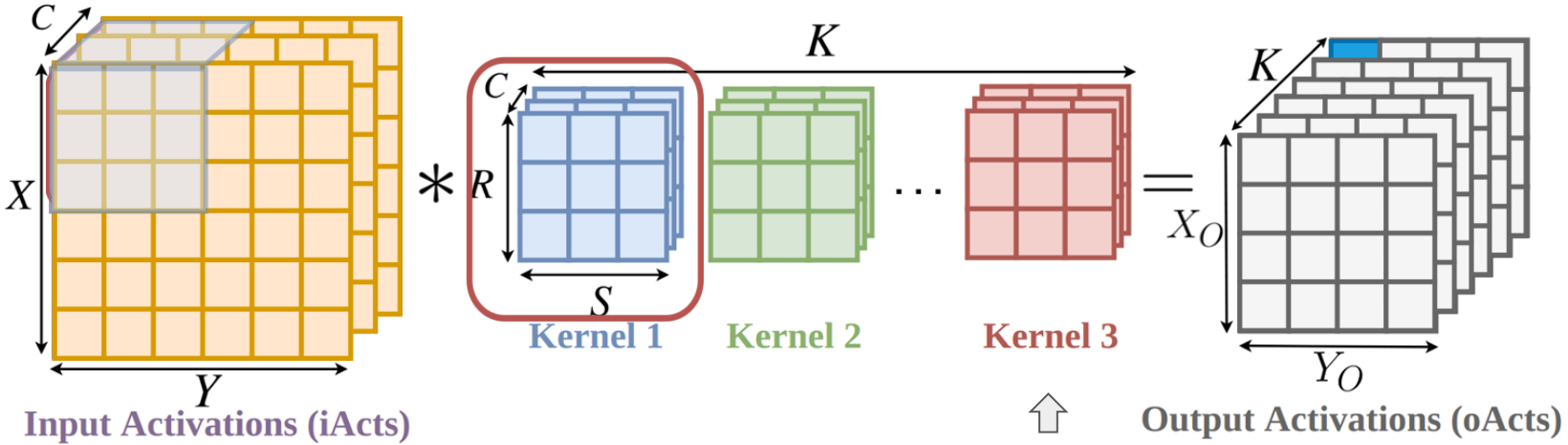


iAct Reuse
Sliding Windows Overlap

iAct Reuse
Multi-Kernel

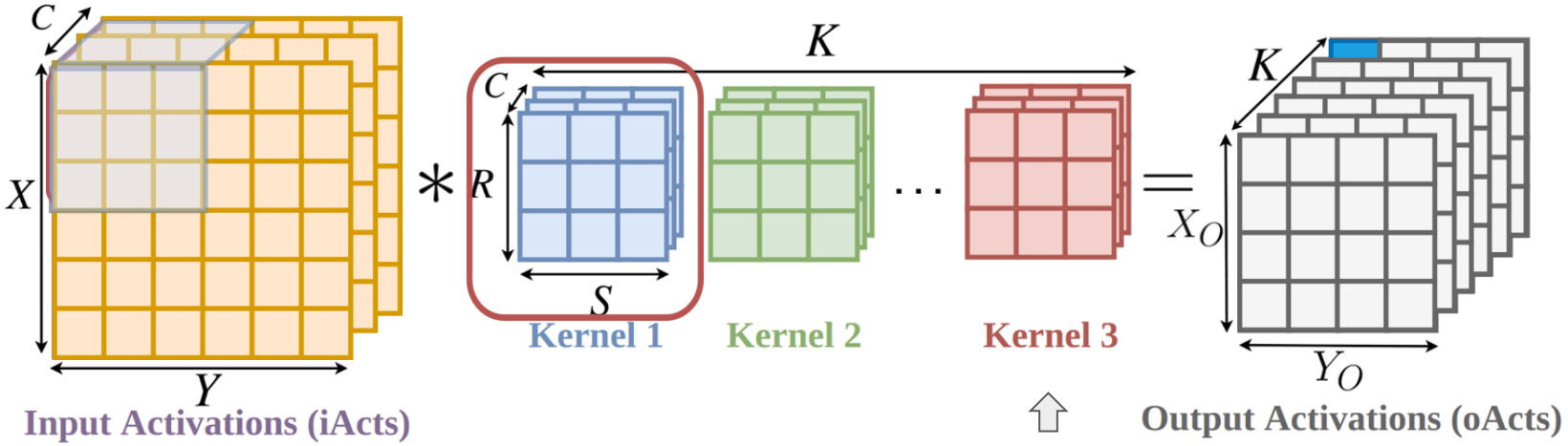
Weights Reuse
Multi-iAct Tiling

Data Processing Order - Under Weights Stationary



Insight 1: iAct and weights are reused.

Data Processing Order - Under Weights Stationary



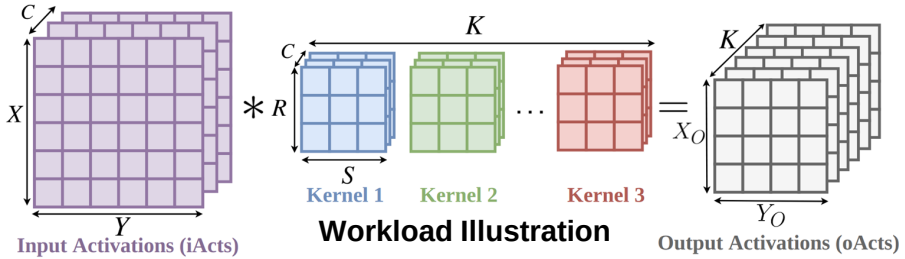
Insight 1: iAct and weights are reused.

Insight 2: iAct access are not continuous, weights access are continuous.

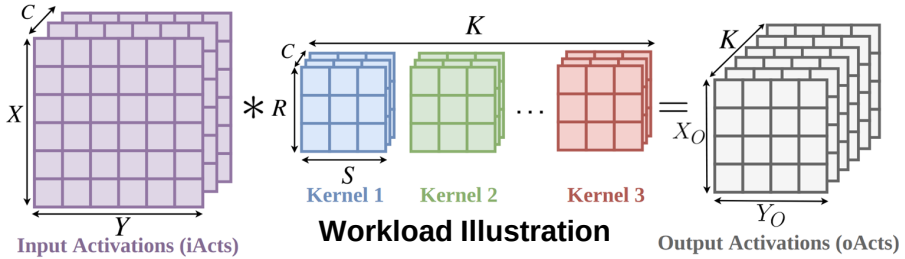
Outlines

- Supported Neural Network Model
- Quantization Flow
- Memory Layout
- Heterogeneous Scheduling
- **MAERI 2.0 Microarchitecture**
  - Data Processing Order
  - **Challenges and Proposed Microarchitecture**
- DEMO

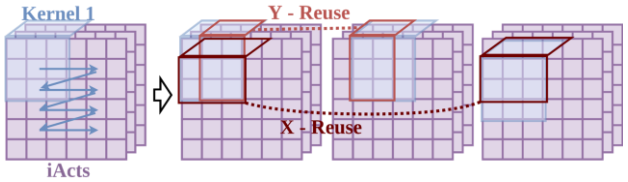
Challenge 1 - How to Leverage Various Data Reuse



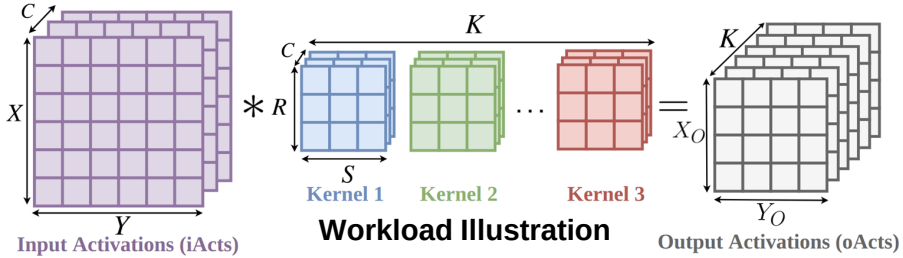
Challenge 1 - How to Leverage Various Data Reuse



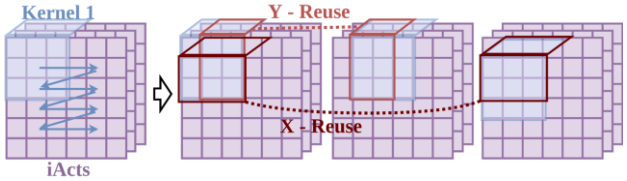
iAct Reuse
Sliding Windows Overlap



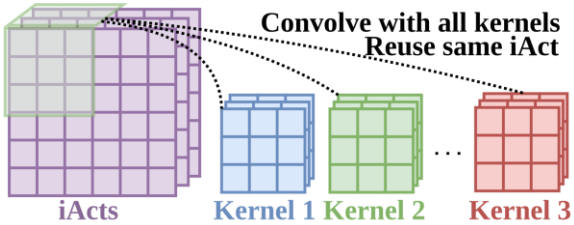
Challenge 1 - How to Leverage Various Data Reuse



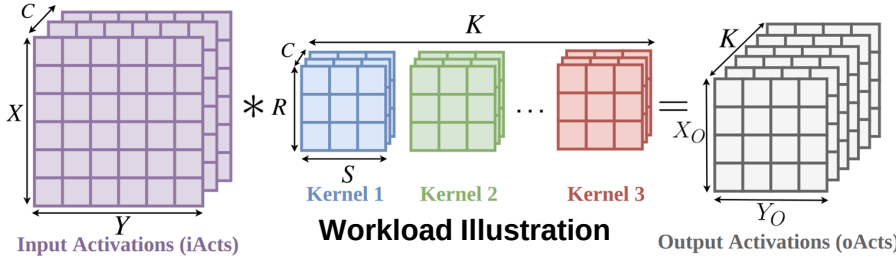
iAct Reuse
Sliding Windows Overlap



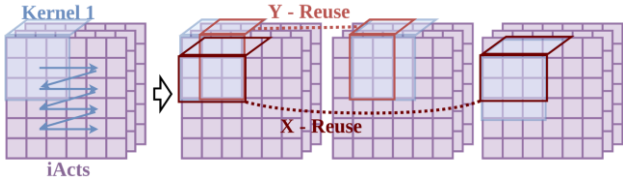
iAct Reuse
Multi-Kernel



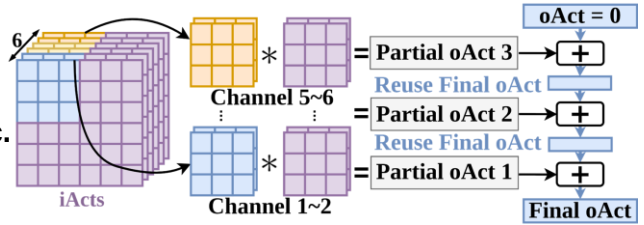
Challenge 1 - How to Leverage Various Data Reuse



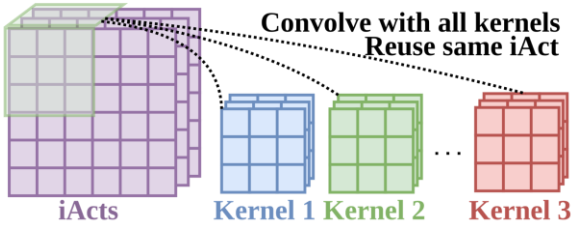
iAct Reuse
Sliding Windows Overlap



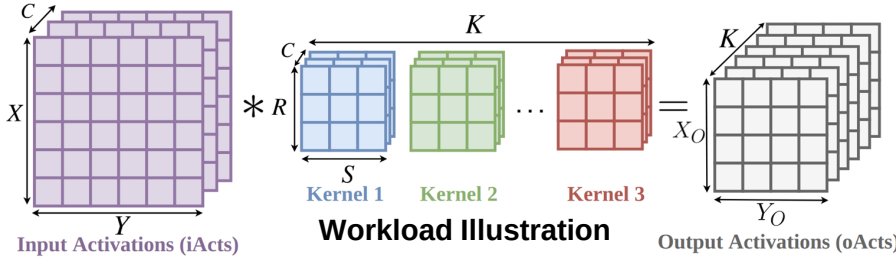
oAct Reuse
Partial Sum Acc.



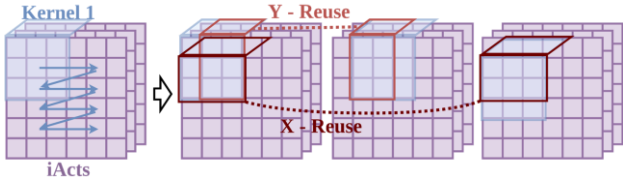
iAct Reuse
Multi-Kernel



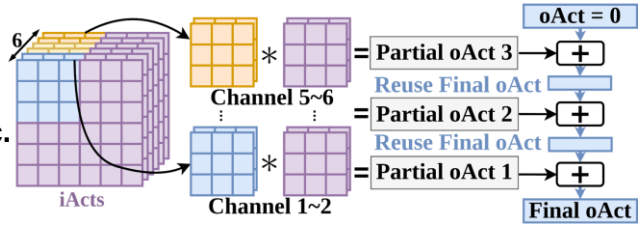
Challenge 1 - How to Leverage Various Data Reuse



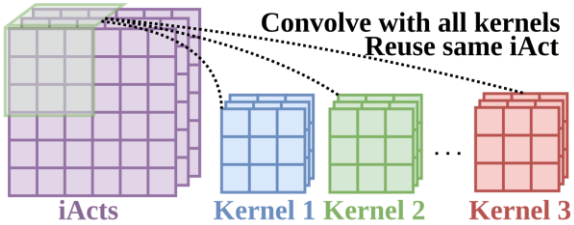
iAct Reuse
Sliding Windows Overlap



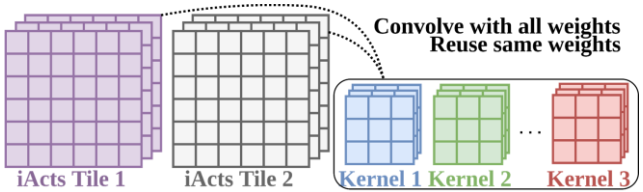
oAct Reuse
Partial Sum Acc.



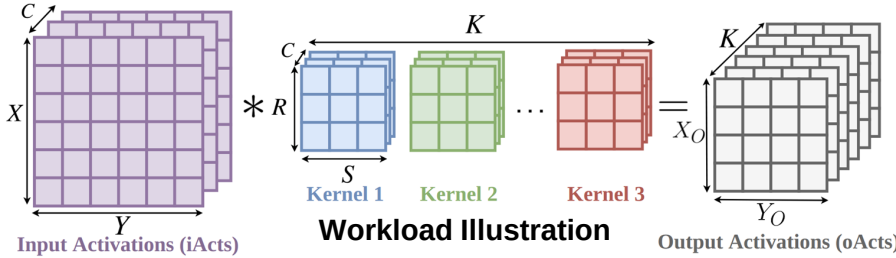
iAct Reuse
Multi-Kernel



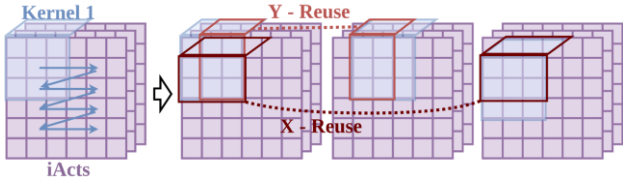
Weights Reuse
Multi-iAct Tiling



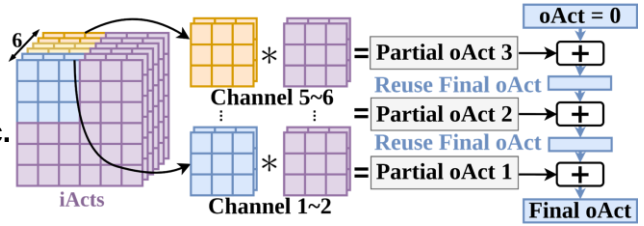
Challenge 1 - How to Leverage Various Data Reuse



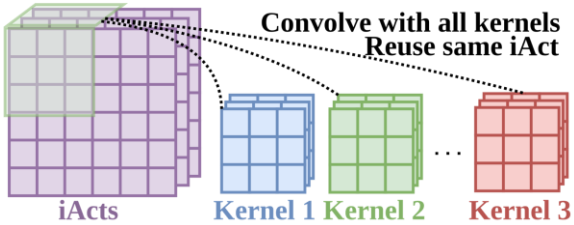
iAct Reuse
Sliding Windows Overlap



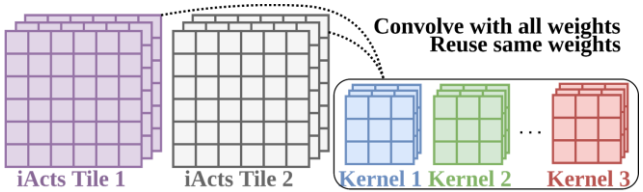
oAct Reuse
Partial Sum Acc.



iAct Reuse
Multi-Kernel

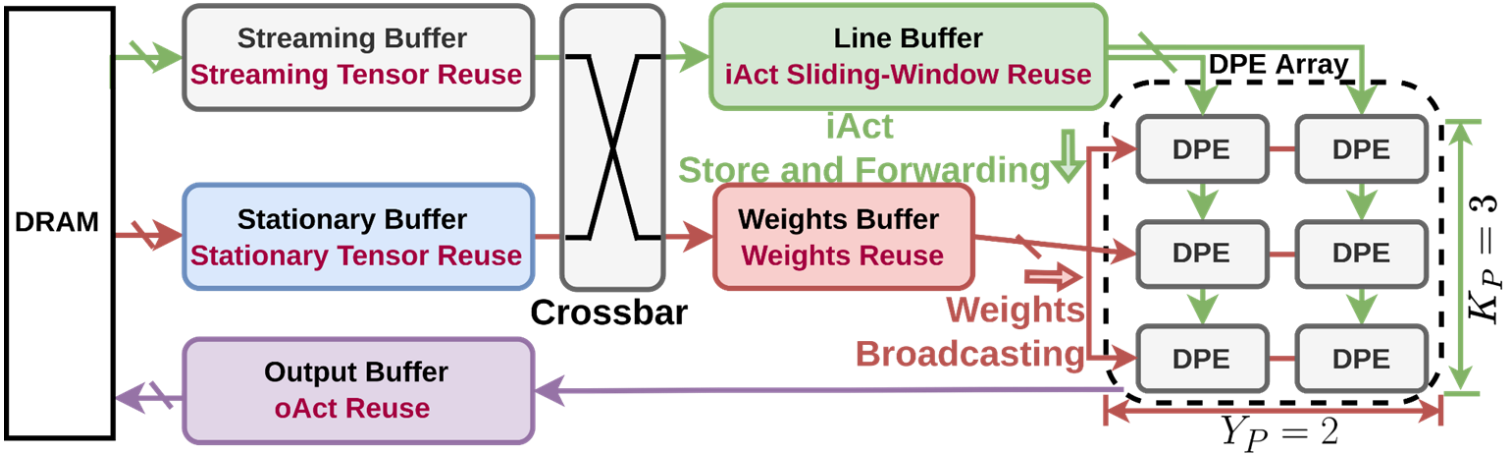


Weights Reuse
Multi-iAct Tiling

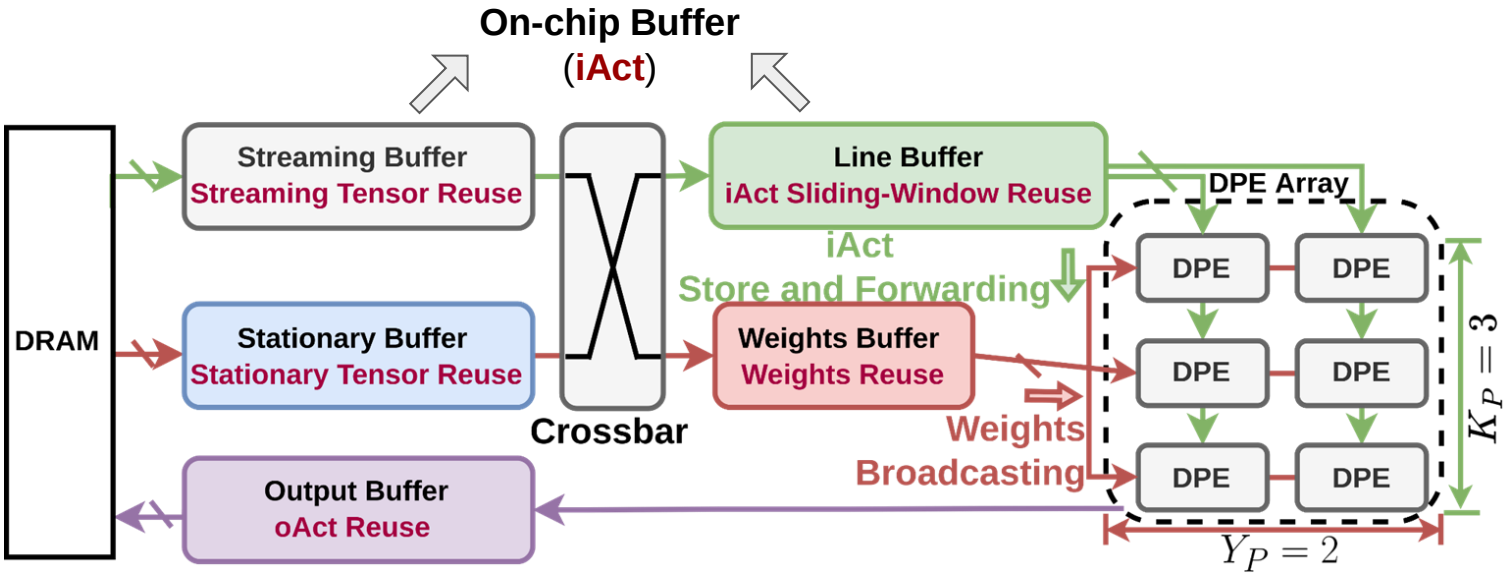


Insight 1: need on-chip buffer to store data for leveraging reuse.

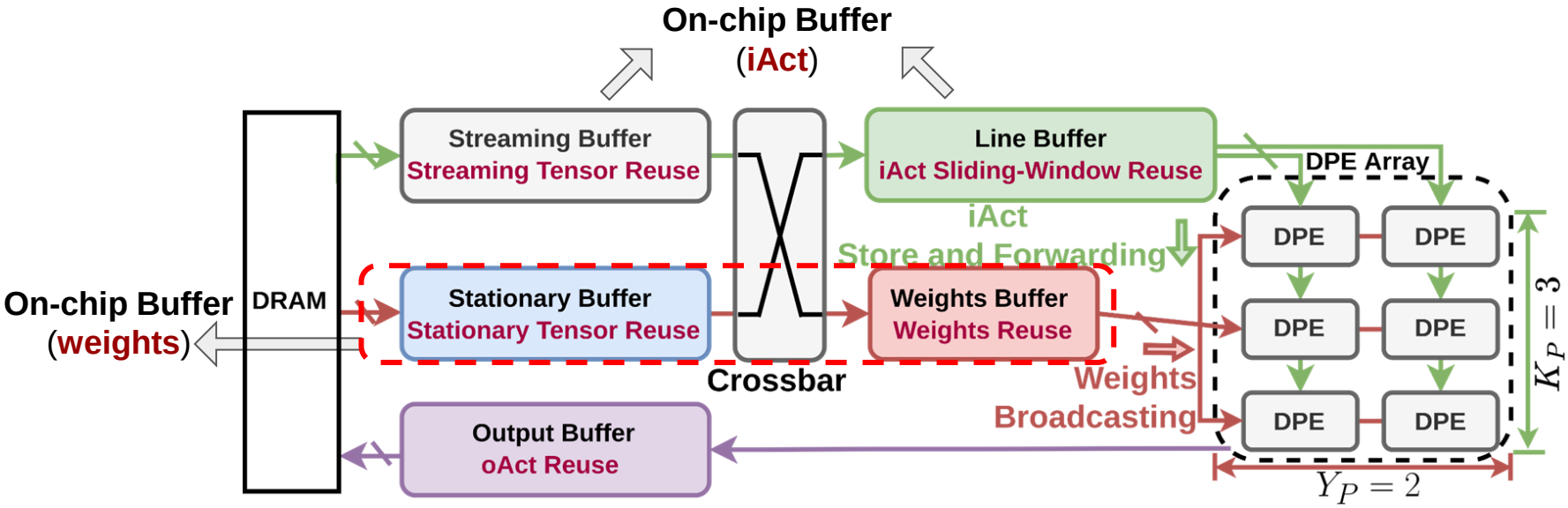
MAERI 2.0 Micro-architecture - Overview



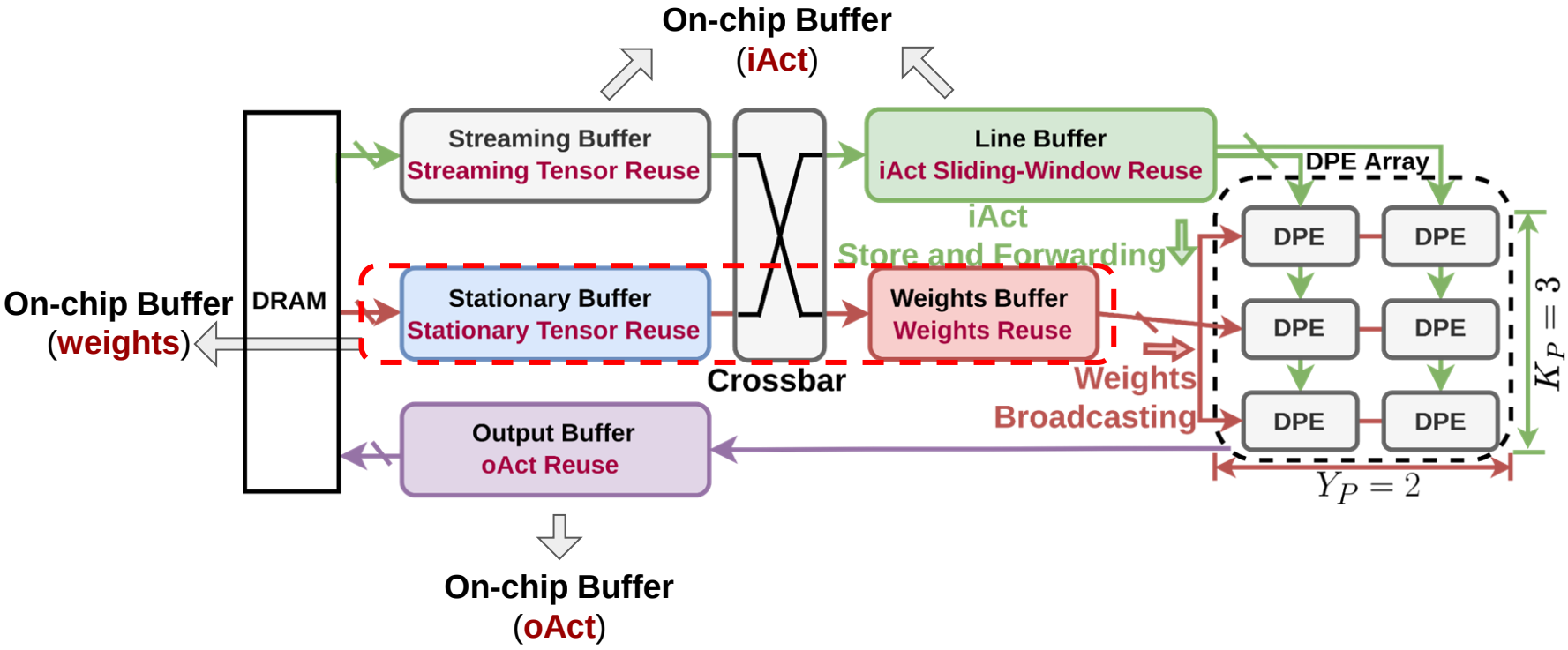
MAERI 2.0 Micro-architecture - Overview



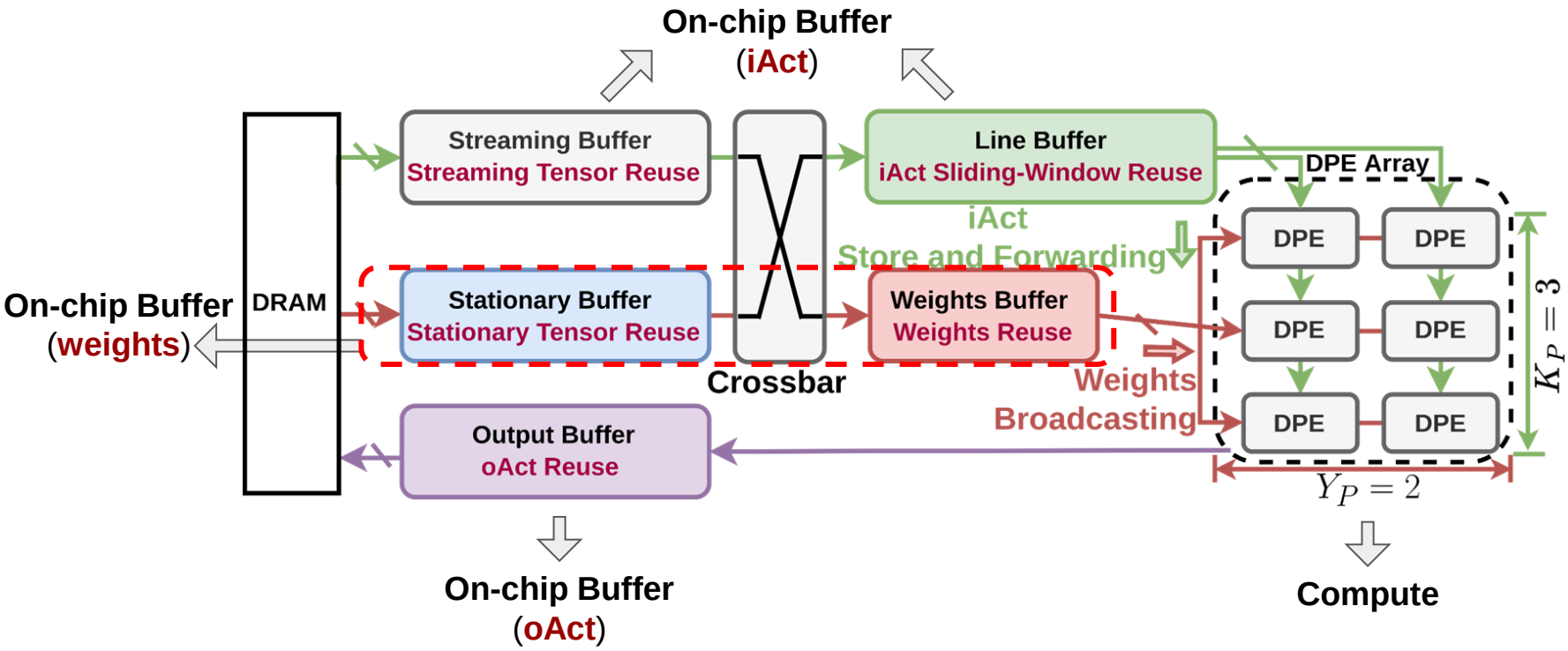
MAERI 2.0 Micro-architecture - Overview



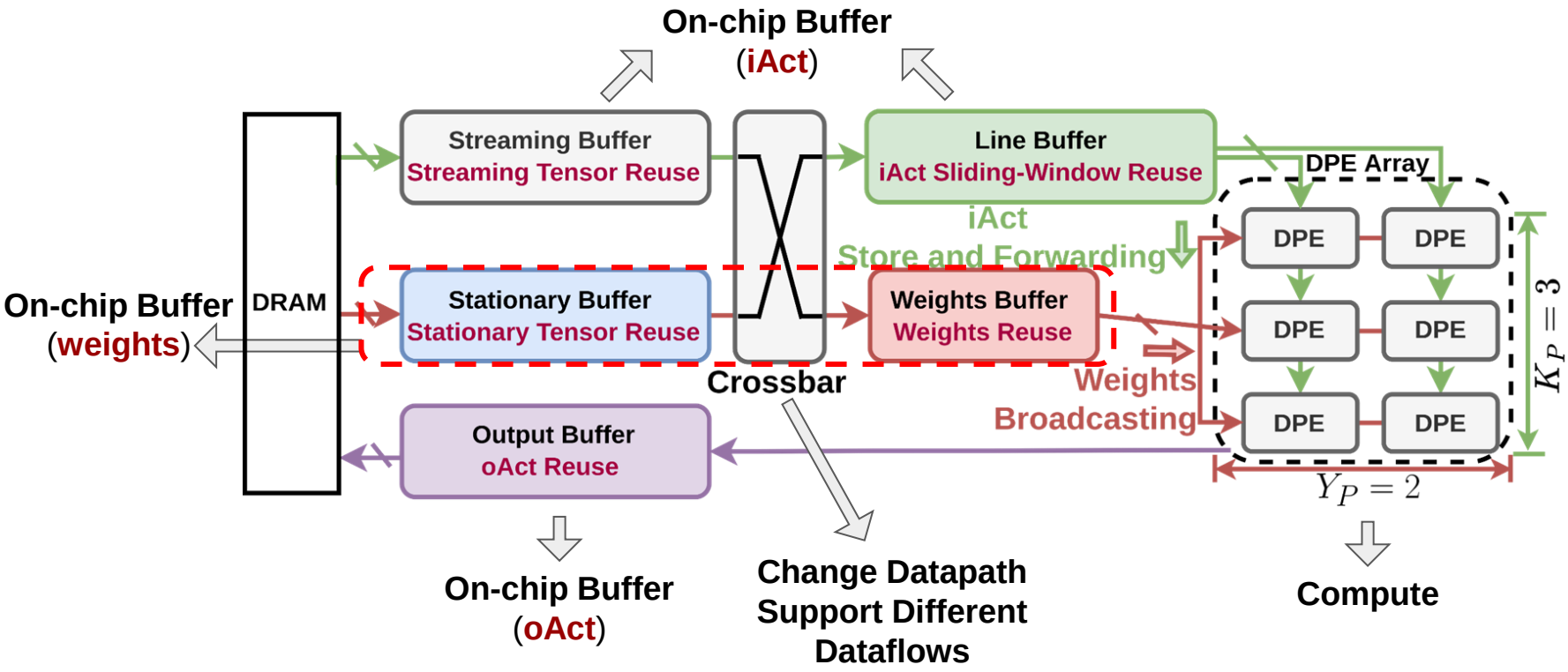
MAERI 2.0 Micro-architecture - Overview



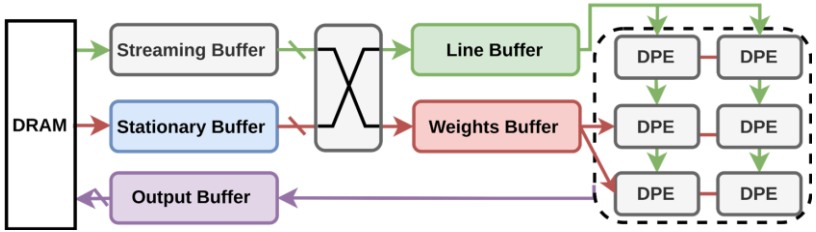
MAERI 2.0 Micro-architecture - Overview



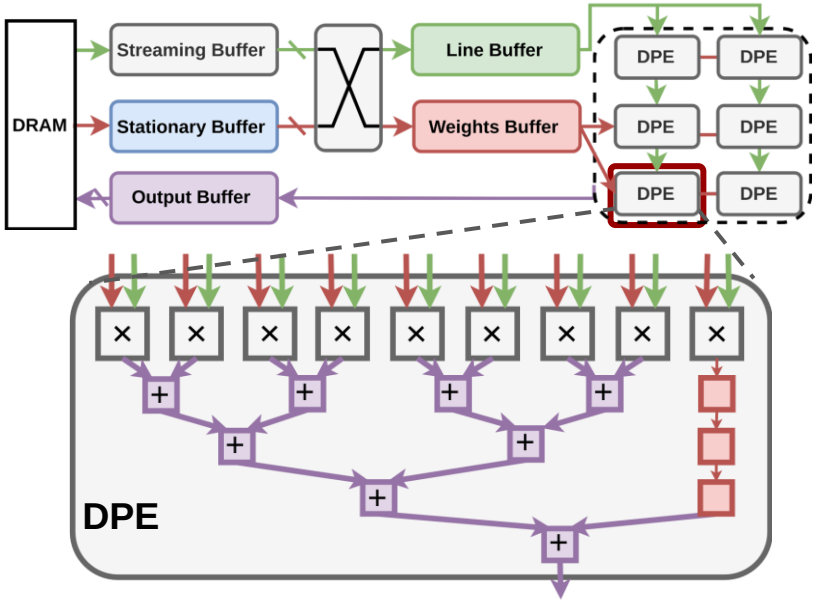
MAERI 2.0 Micro-architecture - Overview



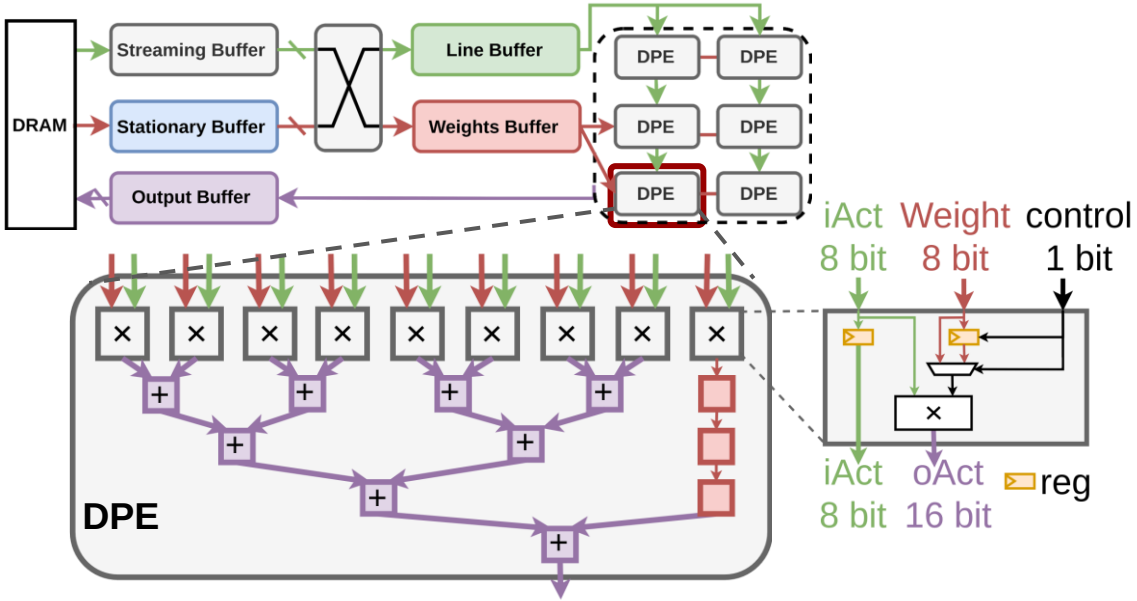
MAERI 2.0 Micro-architecture - Computation



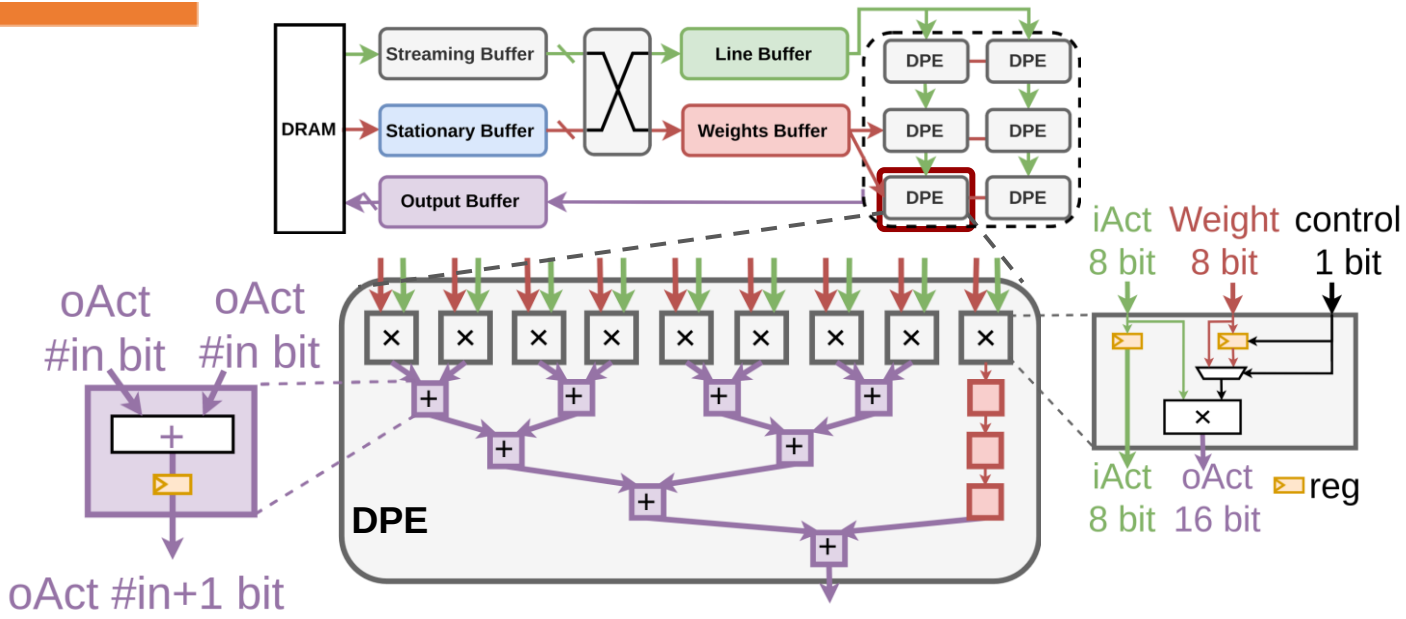
MAERI 2.0 Micro-architecture - Computation



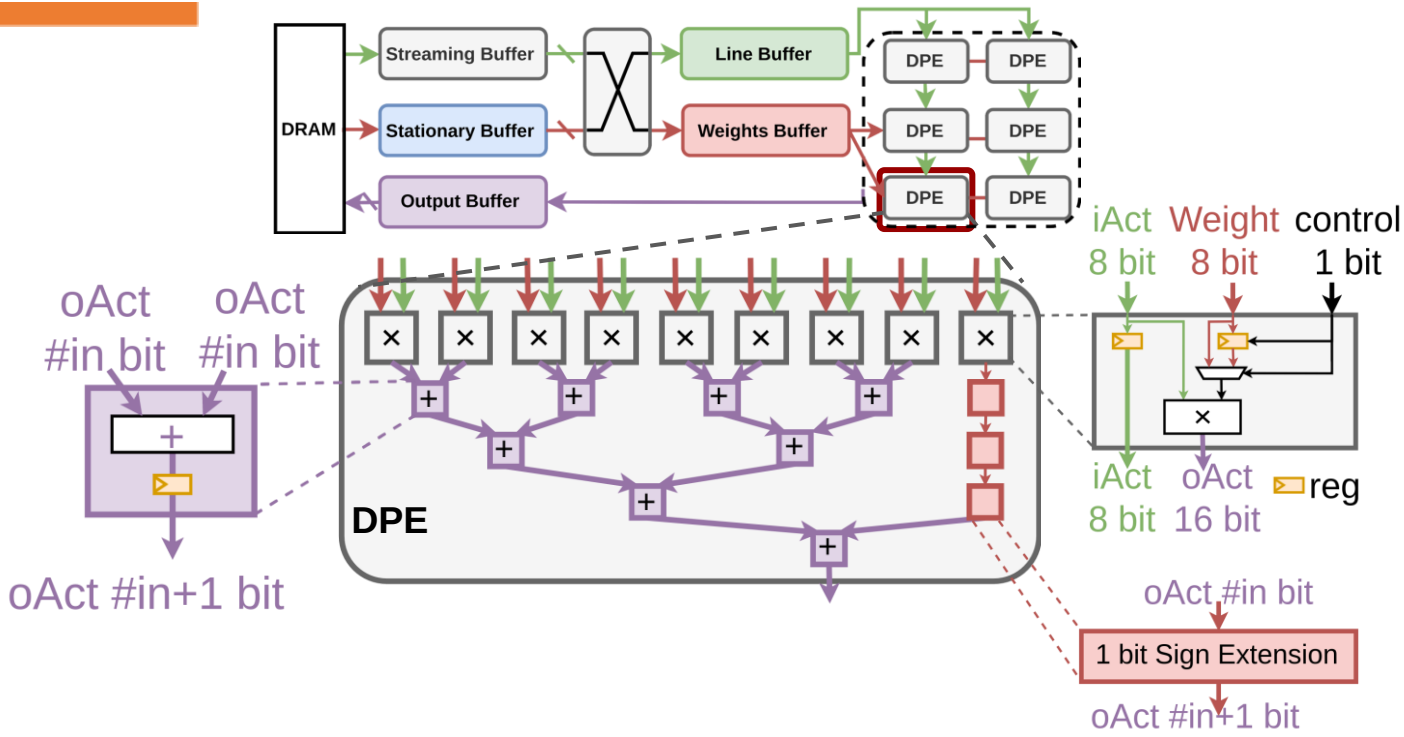
MAERI 2.0 Micro-architecture - Computation



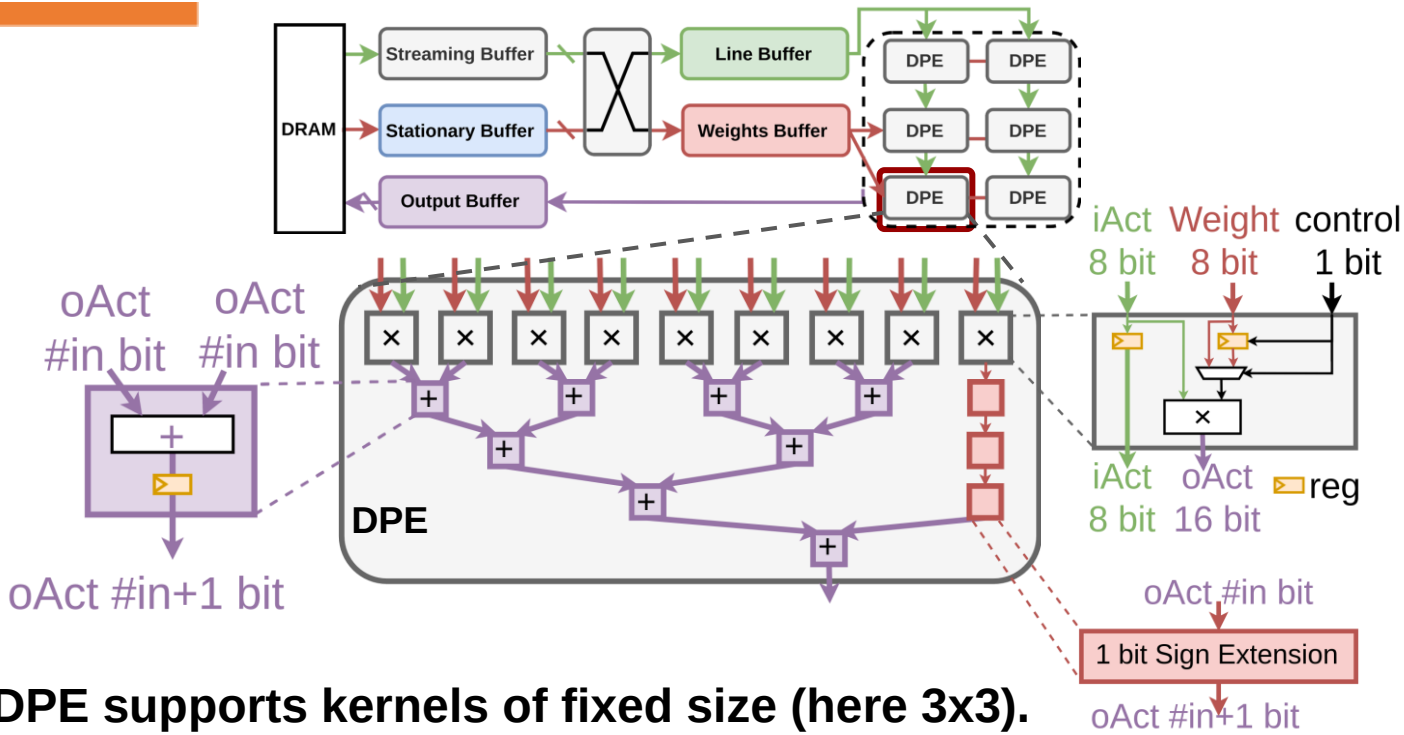
MAERI 2.0 Micro-architecture - Computation



MAERI 2.0 Micro-architecture - Computation

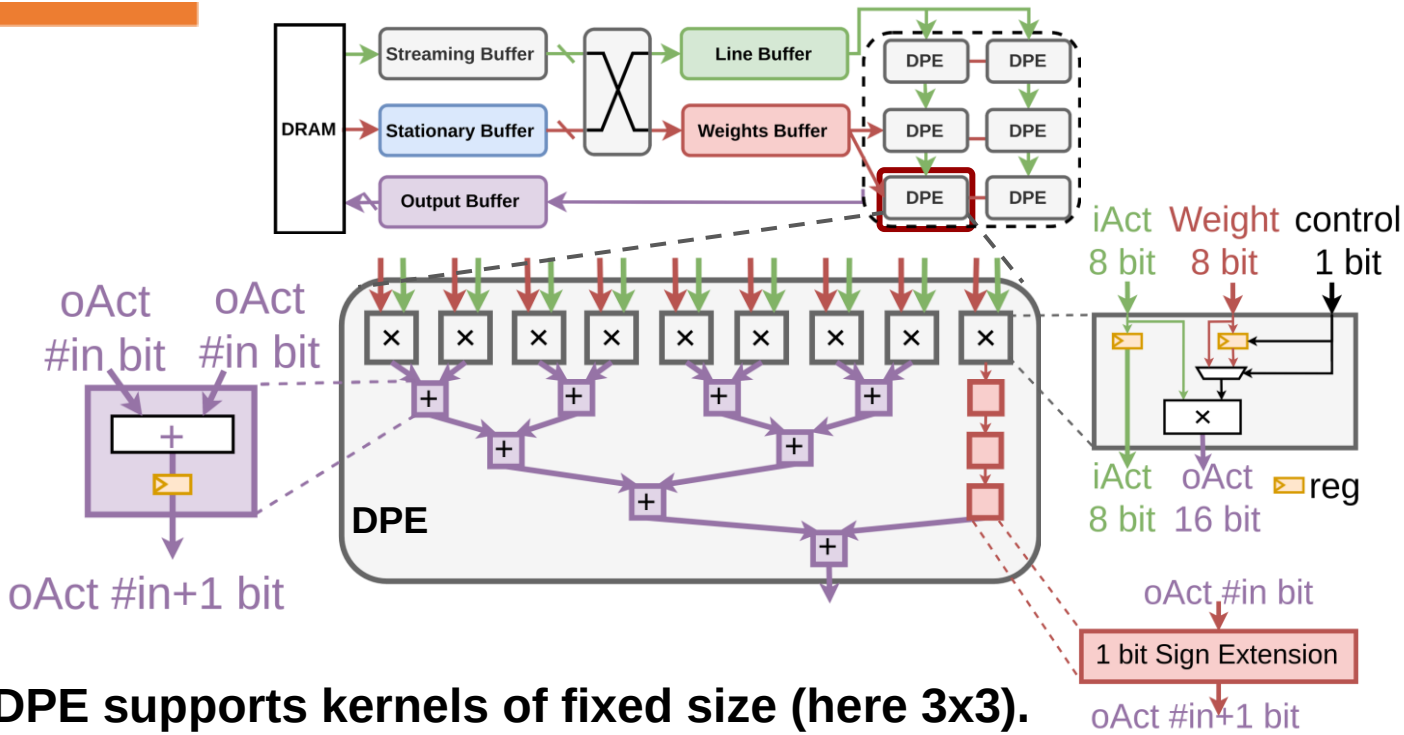


MAERI 2.0 Micro-architecture - Computation



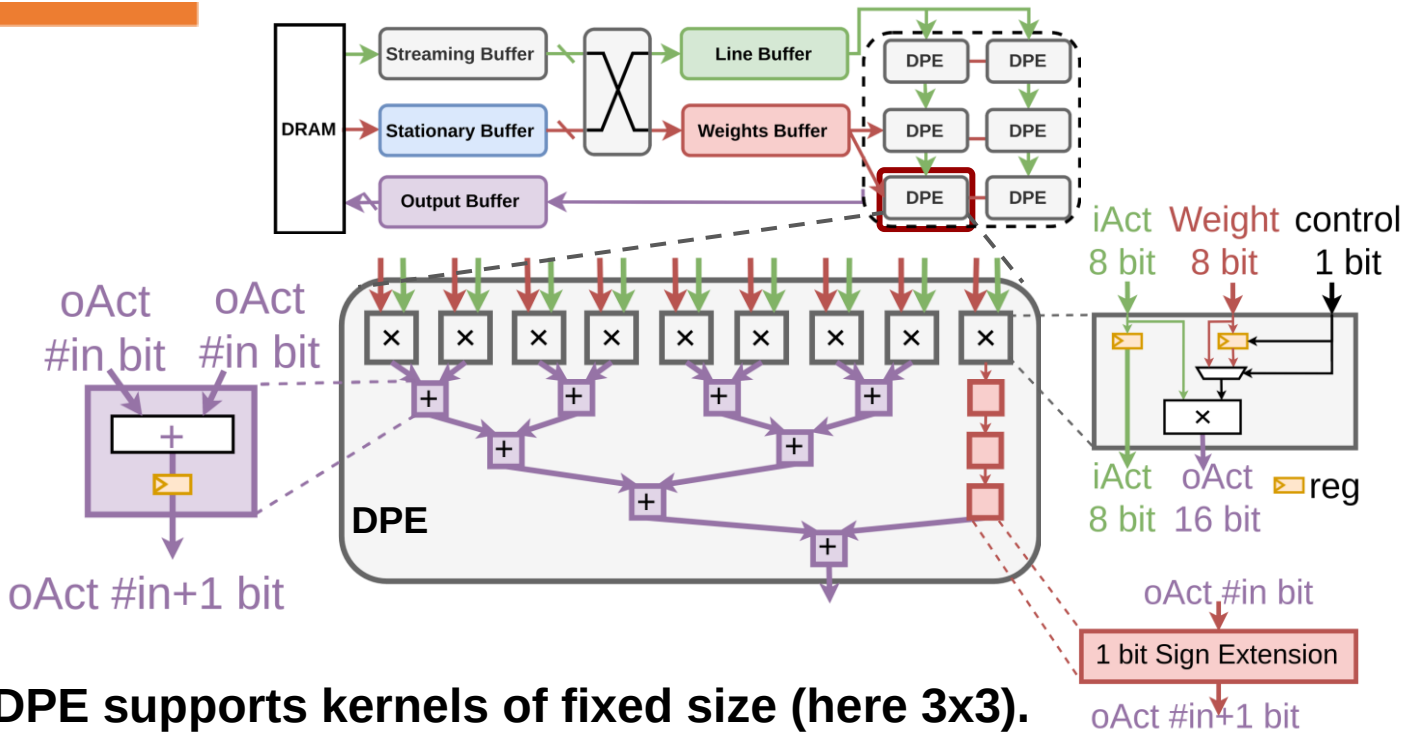
- **DPE supports kernels of fixed size (here 3x3).**
  - Large Kernel could be decomposed into serial of 3x3.
    - Save computation & storage
  - Small Kernel directly put into design.

MAERI 2.0 Micro-architecture - Computation



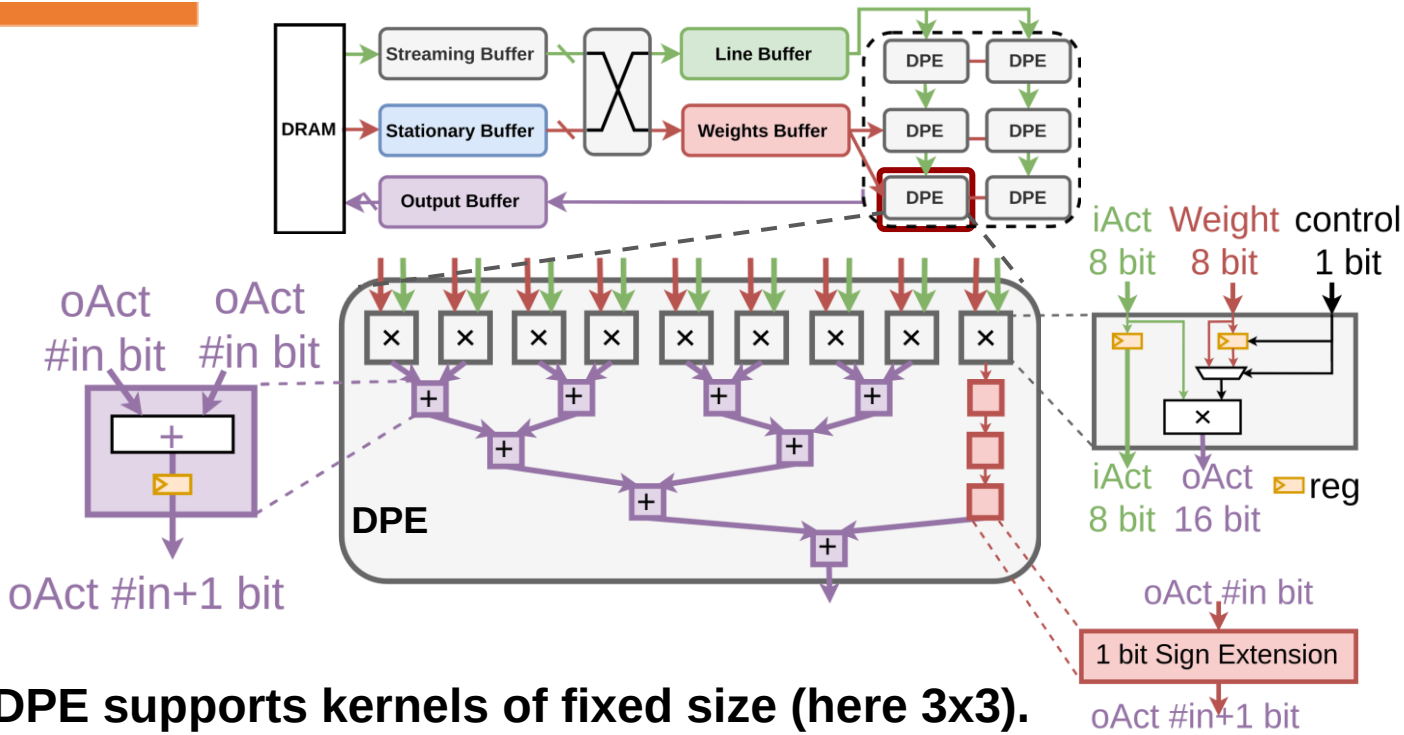
- **DPE supports kernels of fixed size (here 3x3).**
  - Large Kernel could be decomposed into serial of 3x3.
    - Save computation & storage
  - Small Kernel directly put into design.

MAERI 2.0 Micro-architecture - Computation



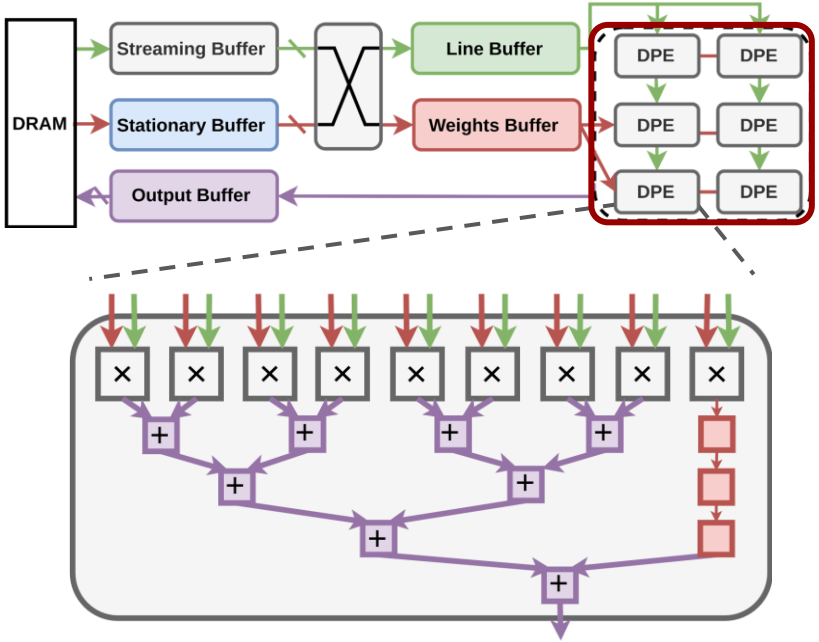
- **DPE supports kernels of fixed size (here 3x3).**
  - Large Kernel could be decomposed into serial of 3x3.
    - Save computation & storage
  - Small Kernel directly put into design.

MAERI 2.0 Micro-architecture - Computation

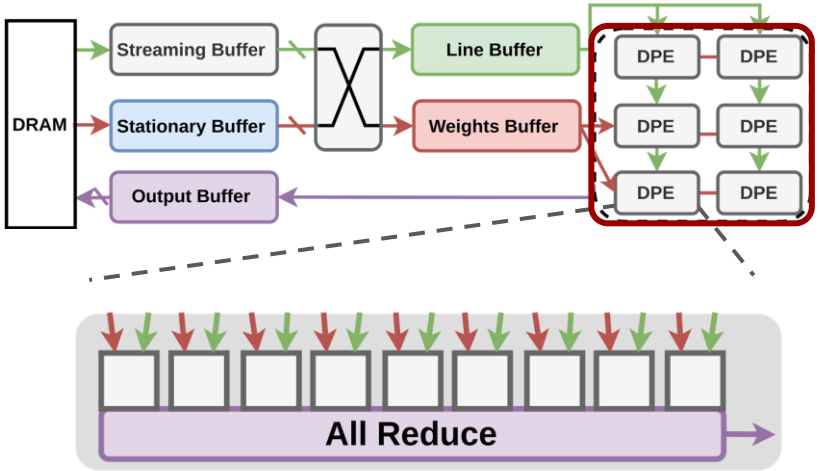


- **DPE supports kernels of fixed size (here 3x3).**
  - Large Kernel could be decomposed into serial of 3x3.
    - Save computation & storage
  - Small Kernel directly put into design.

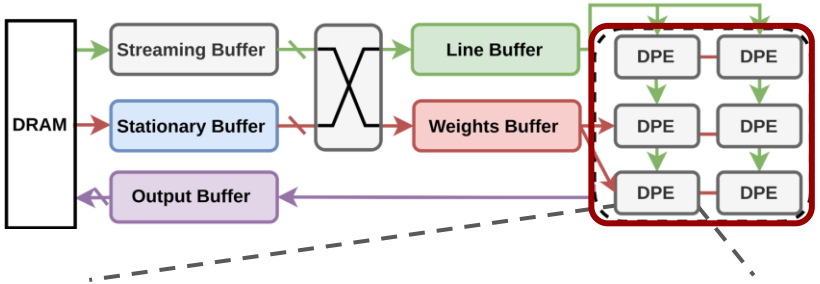
MAERI 2.0 Micro-architecture - Computation

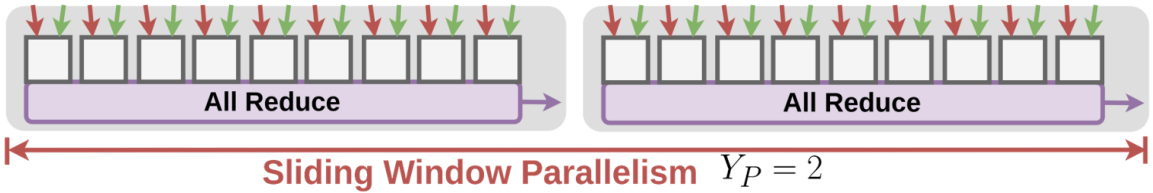


MAERI 2.0 Micro-architecture - Computation



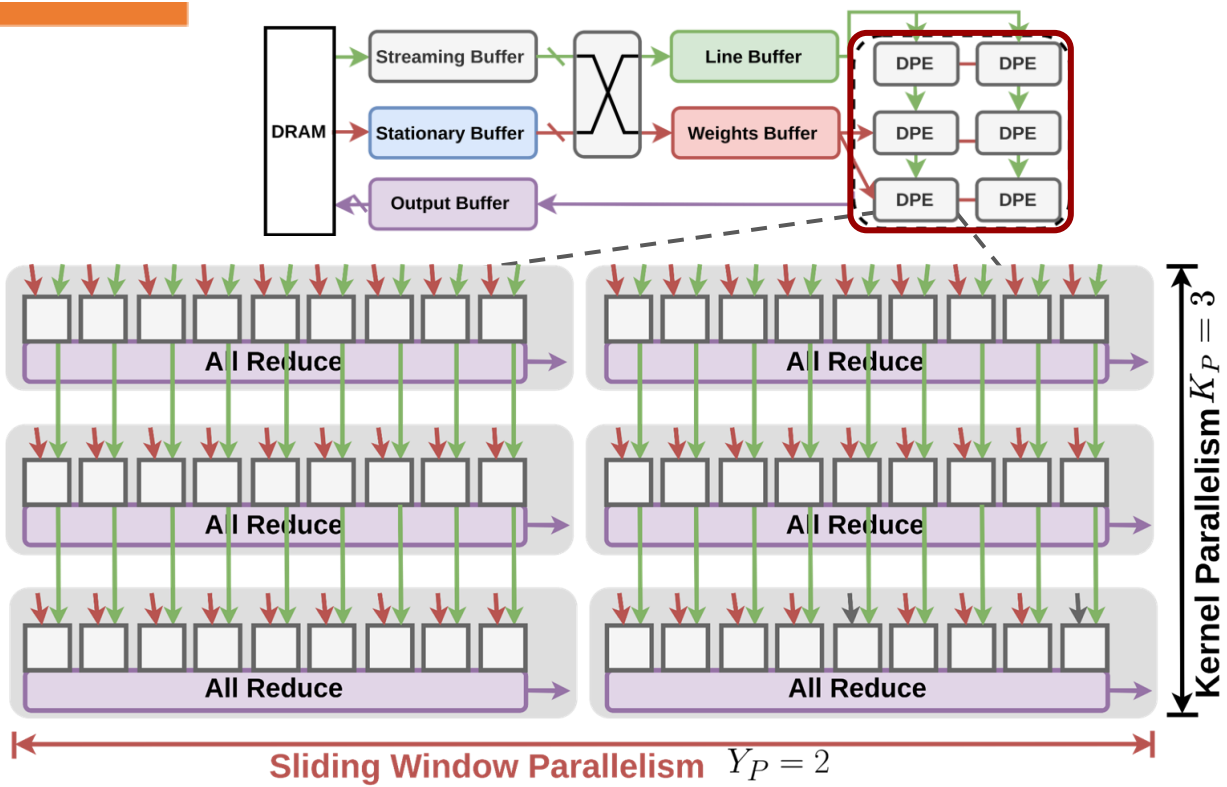
MAERI 2.0 Micro-architecture - Computation





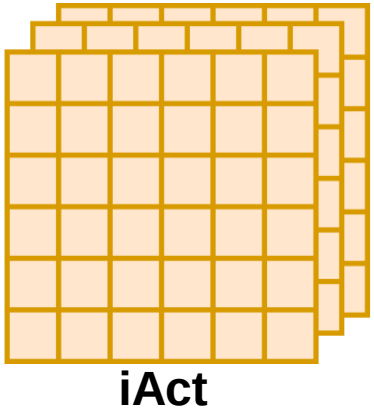
- **Sliding Windows Parallelism.**

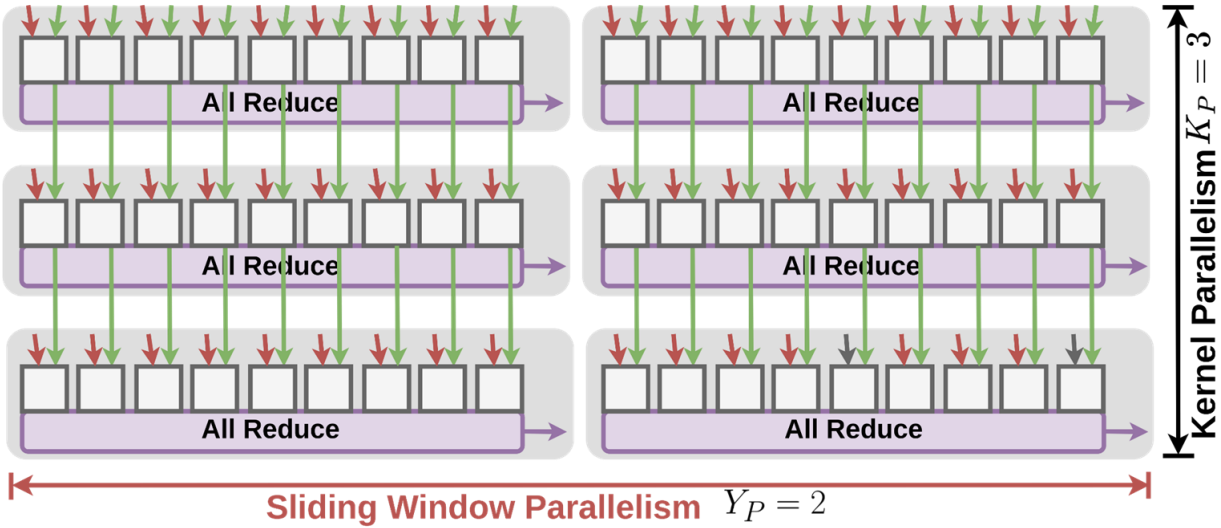
MAERI 2.0 Micro-architecture - Computation



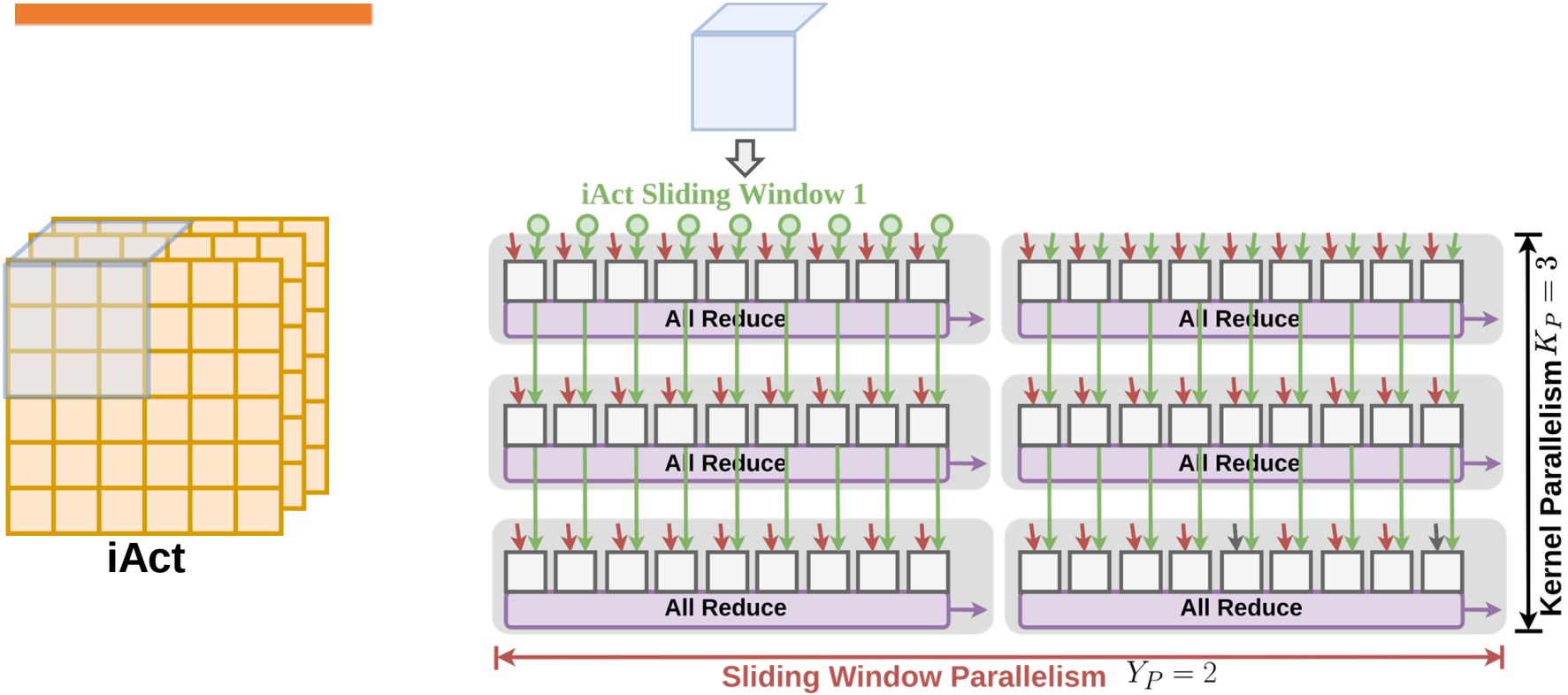
- **Sliding Windows Parallelism.**
- **Kernel Parallelism.**

MAERI 2.0 Micro-architecture - Computation

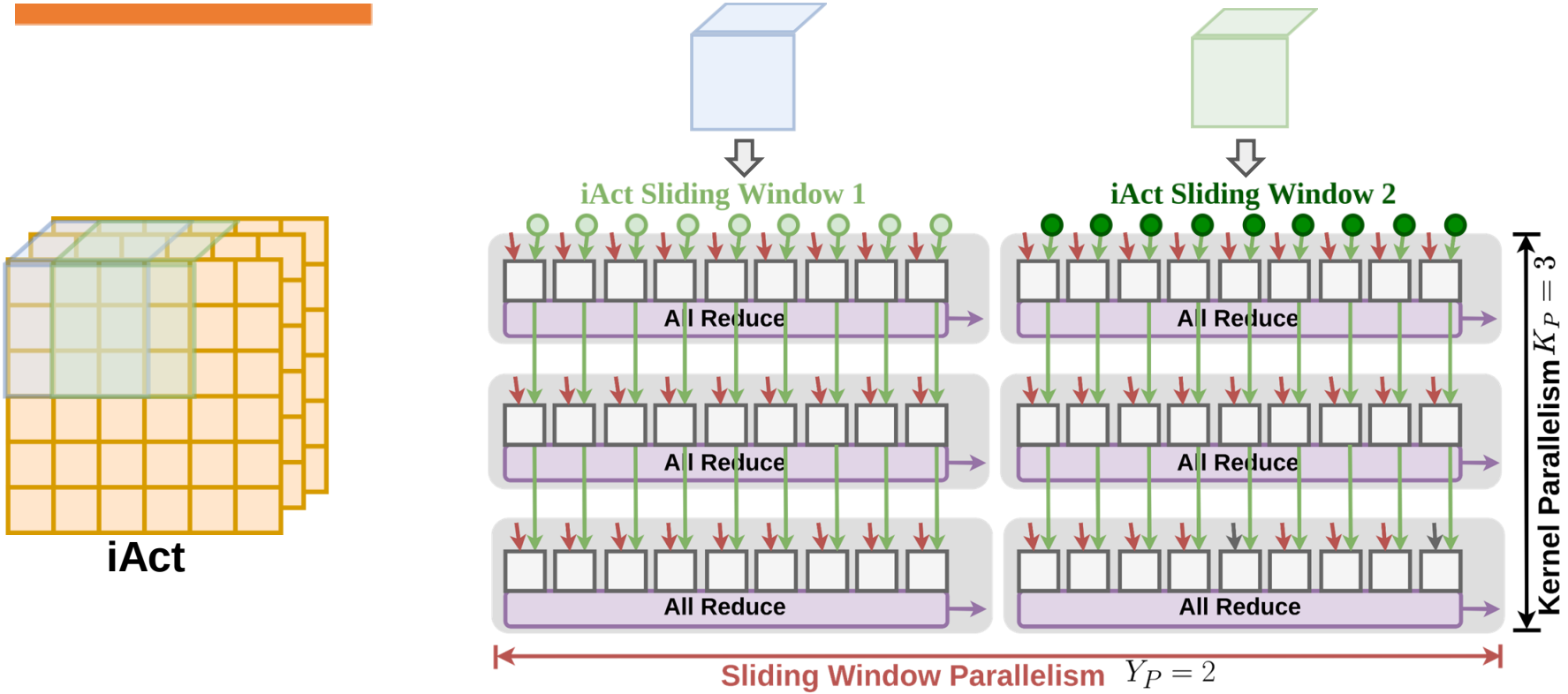




MAERI 2.0 Micro-architecture - Computation

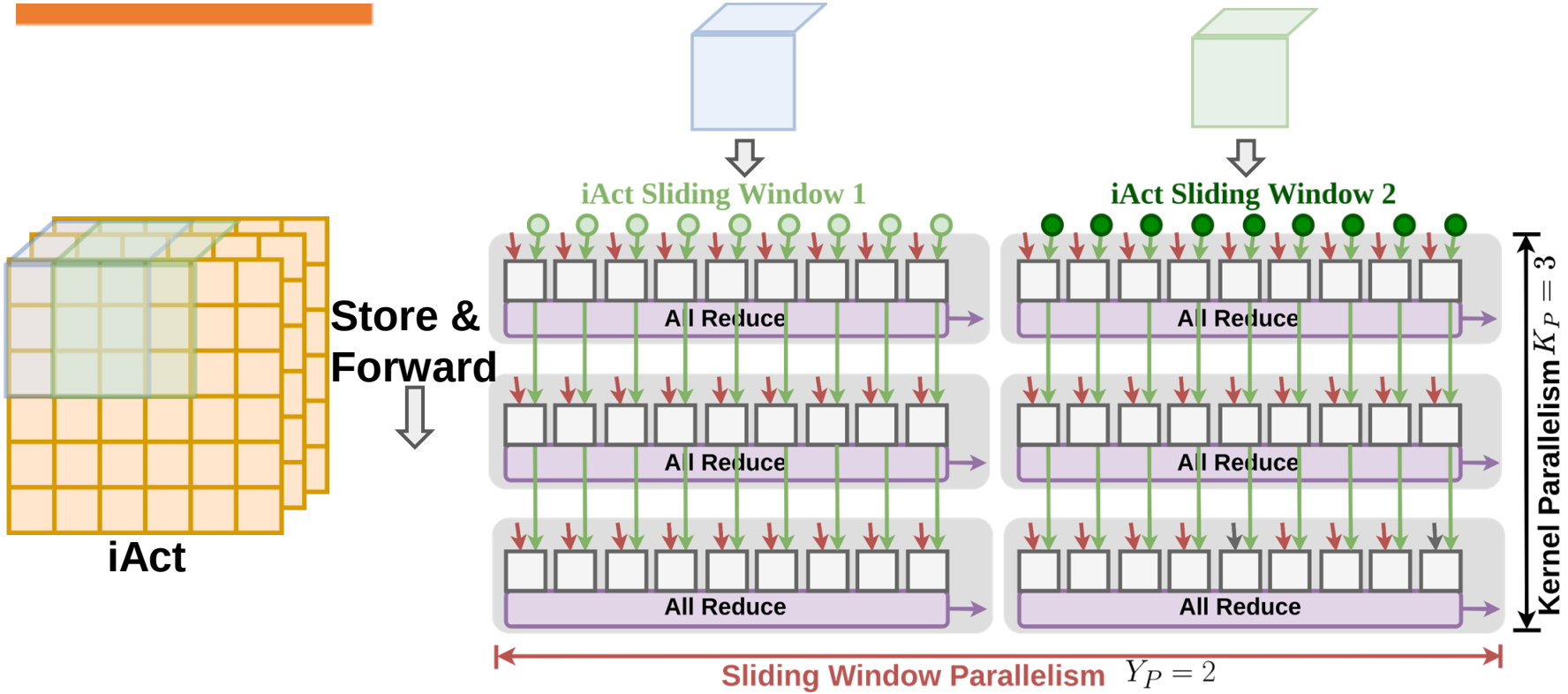


MAERI 2.0 Micro-architecture - Computation



- **Process $Y_P = 2$ Sliding Windows**

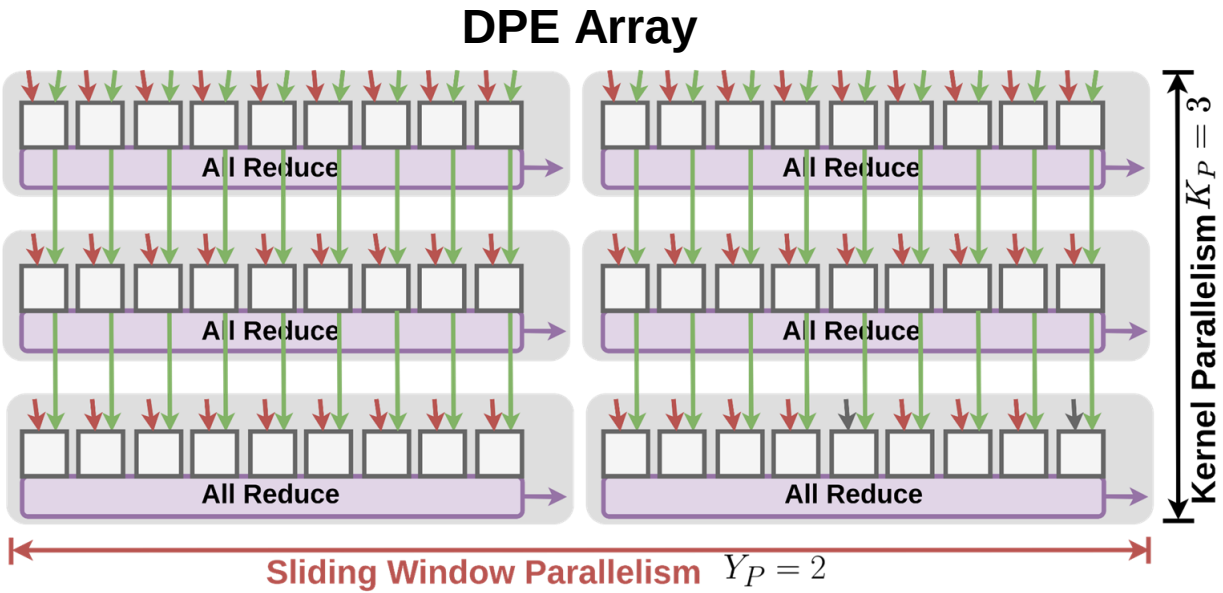
MAERI 2.0 Micro-architecture - Computation



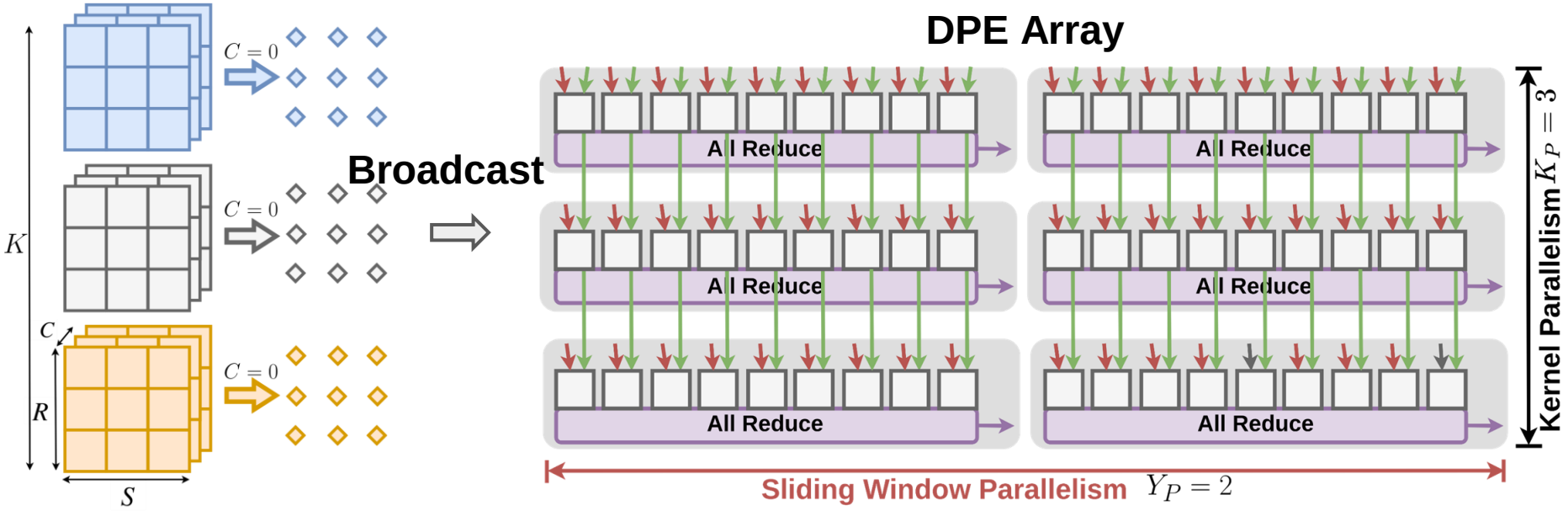
- **Process $Y_P = 2$ Sliding Windows**

MAERI 2.0 Micro-architecture - Computation

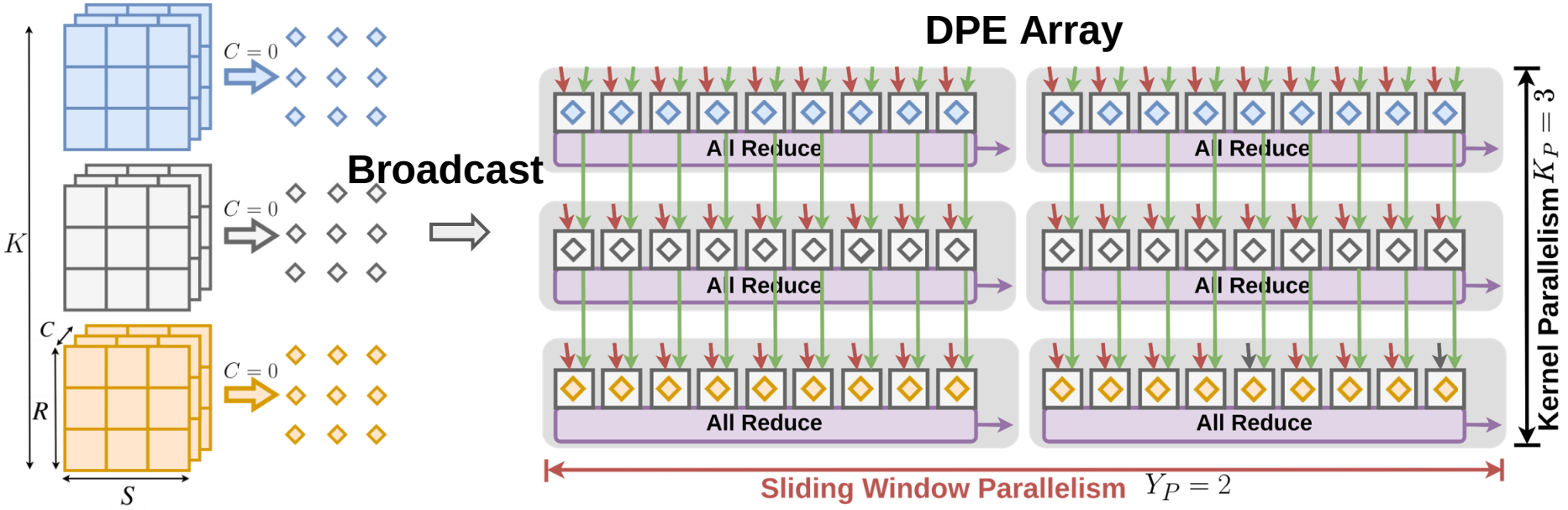
Weights



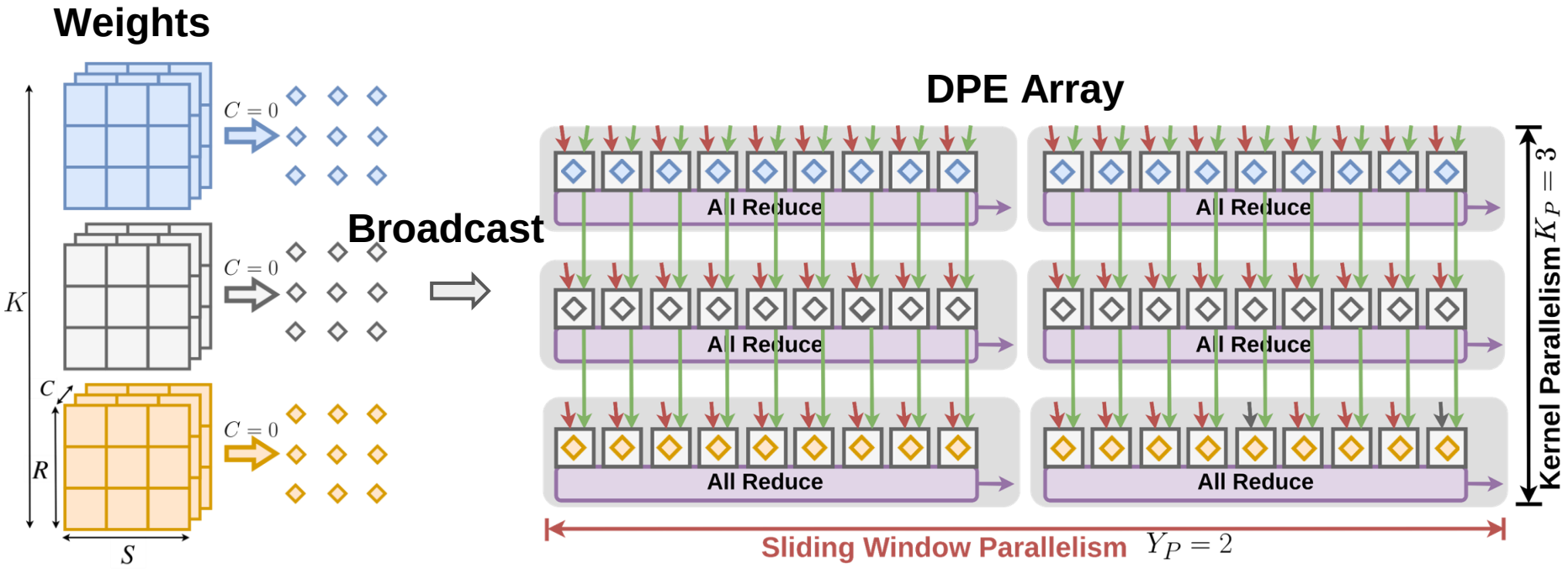
MAERI 2.0 Micro-architecture - Computation



MAERI 2.0 Micro-architecture - Computation

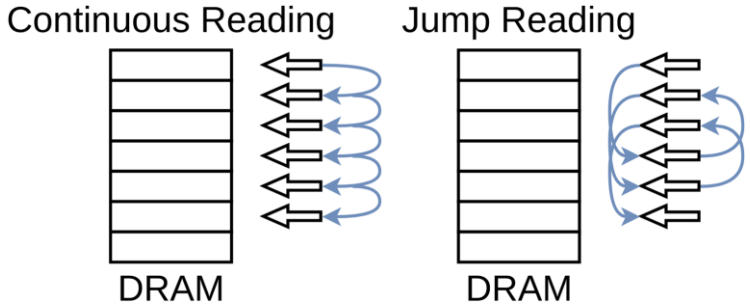


MAERI 2.0 Micro-architecture - Computation

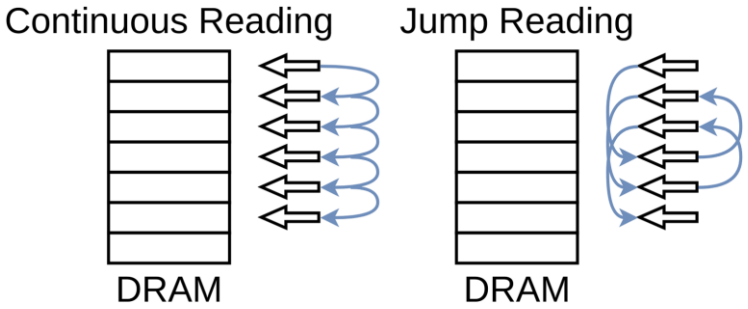


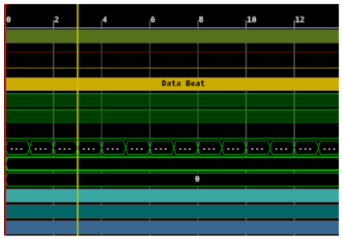
- **Process $K_P = 3$ kernels in parallel**

Challenge 2: Continuous DRAM access

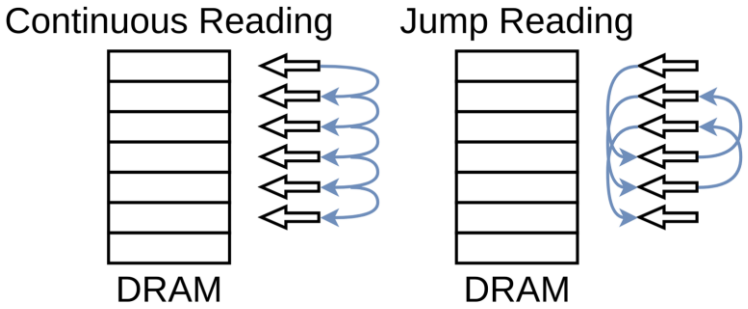


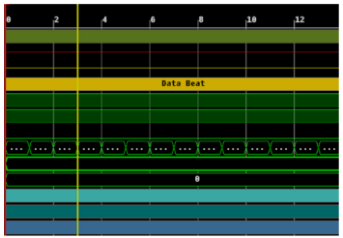
Challenge 2: Continuous DRAM access

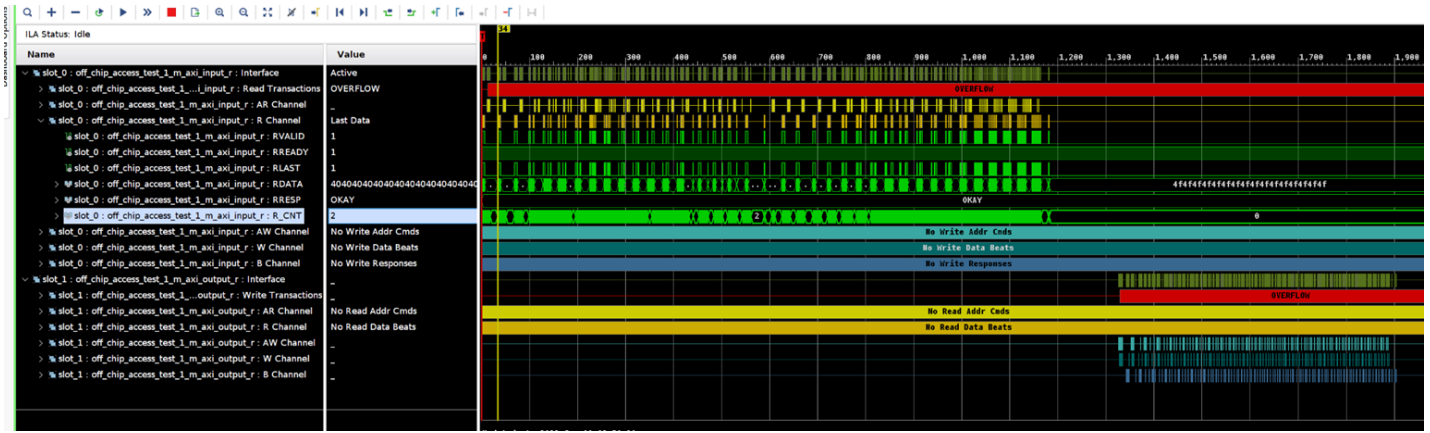




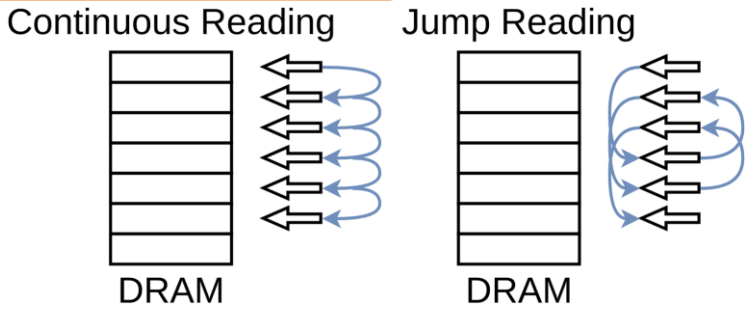
Challenge 2: Continuous DRAM access



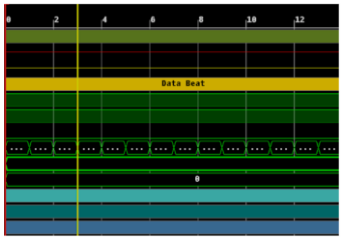


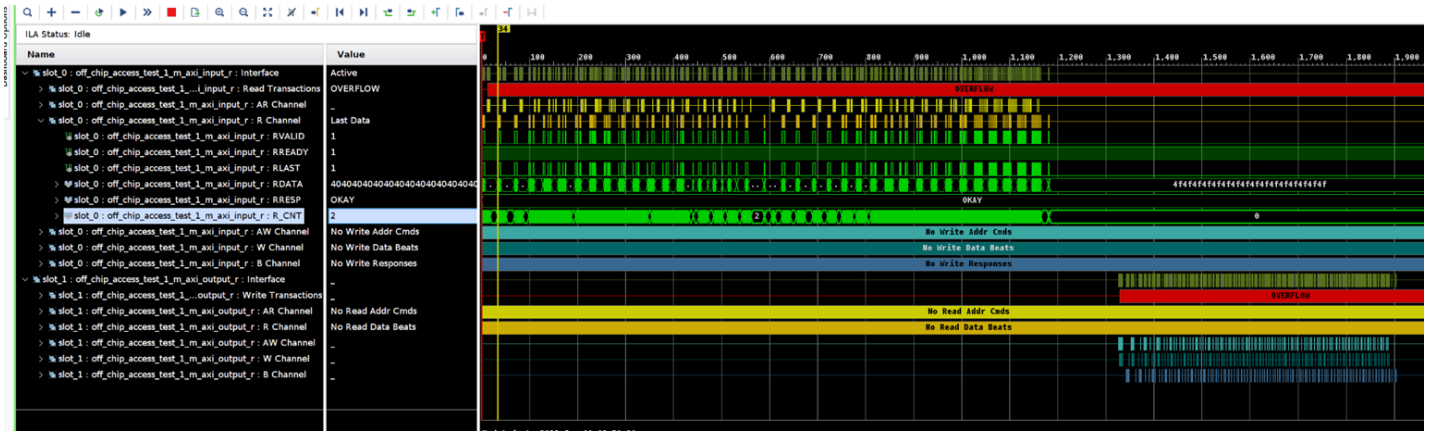


Challenge 2: Continuous DRAM access

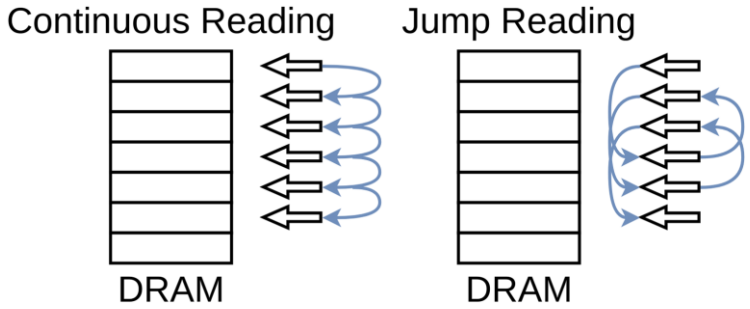


Latency Operation	Continuous	Jump Mode
Read 256 data (128 bit)	256	1182
Write 128 data (128 bit)	128	576

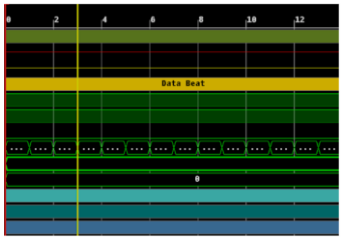


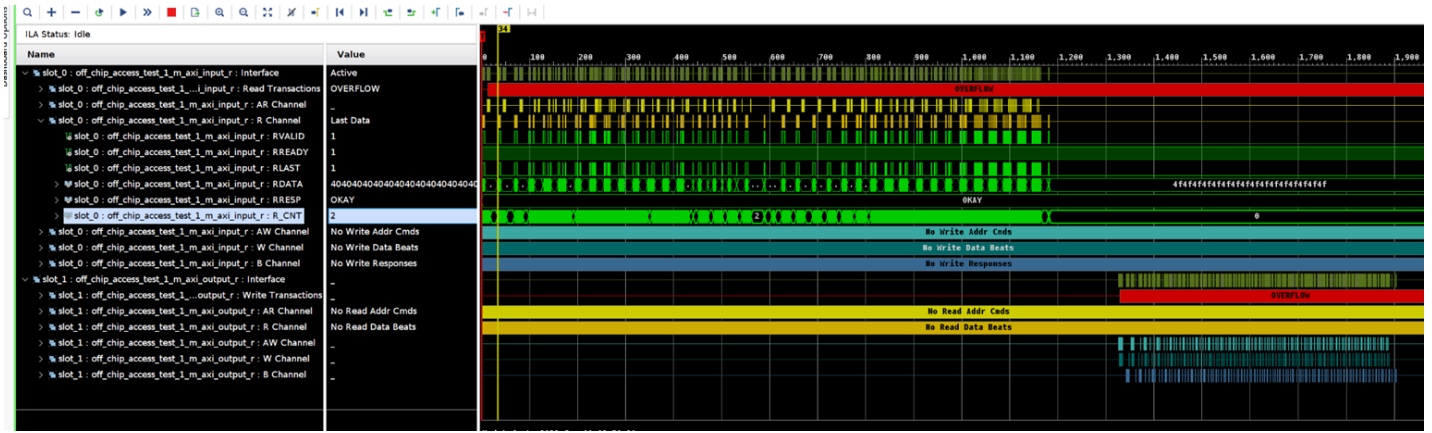


Challenge 2: Continuous DRAM access



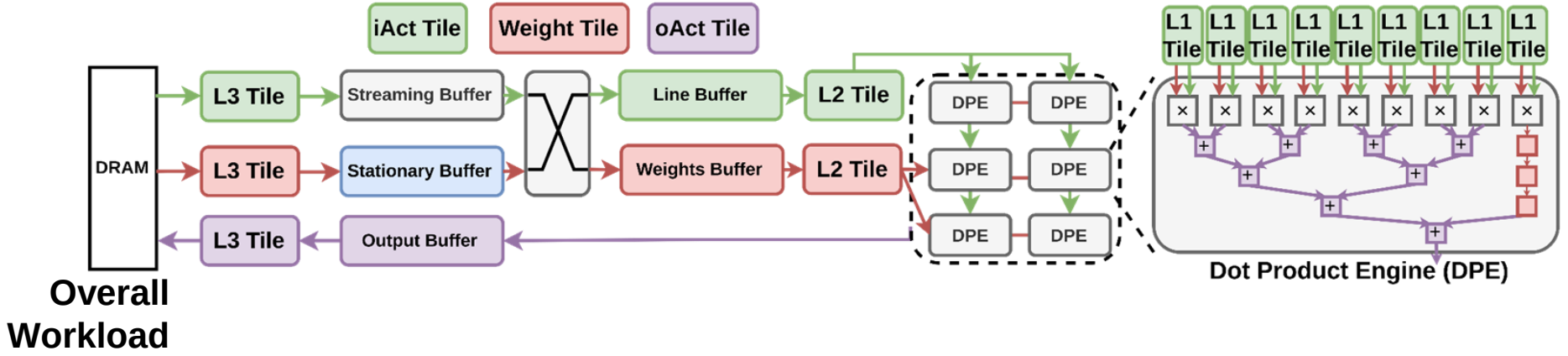
Latency Operation	Continuous	Jump Mode
Read 256 data (128 bit)	256	1182
Write 128 data (128 bit)	128	576



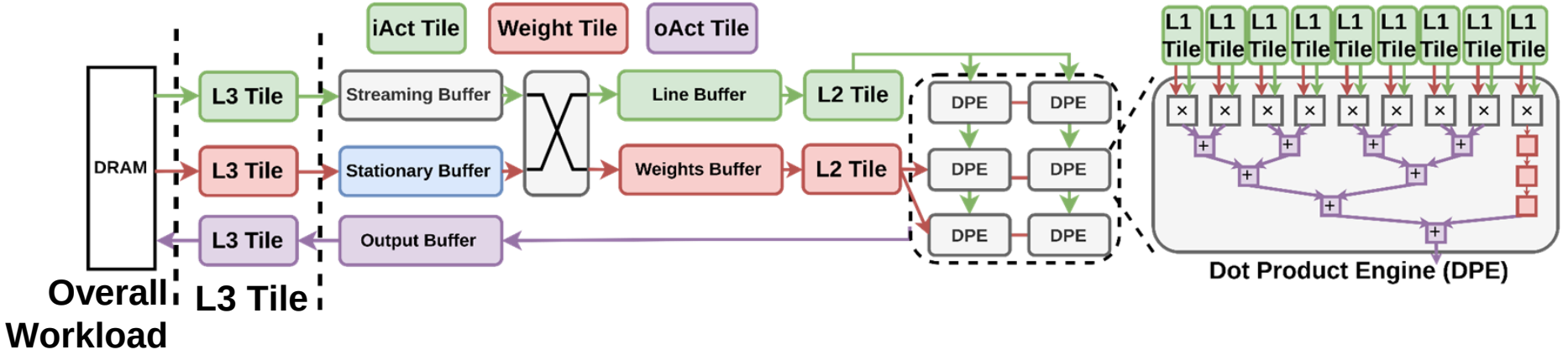


Insight 2: need multi-level tiling to delivery continuous DRAM access.

MAERI 2.0 Micro-architecture - Multi-level Tiling Terminology

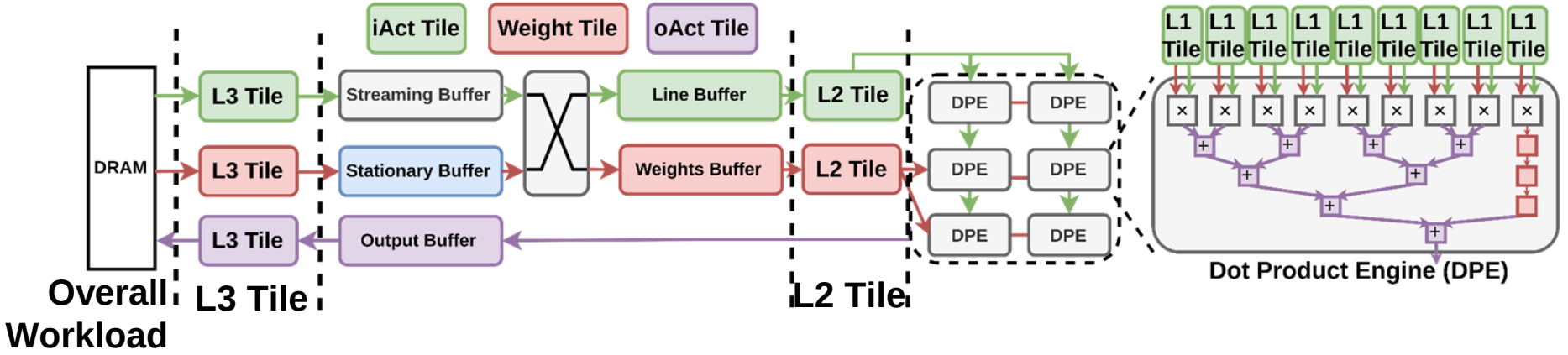


MAERI 2.0 Micro-architecture - Multi-level Tiling Terminology



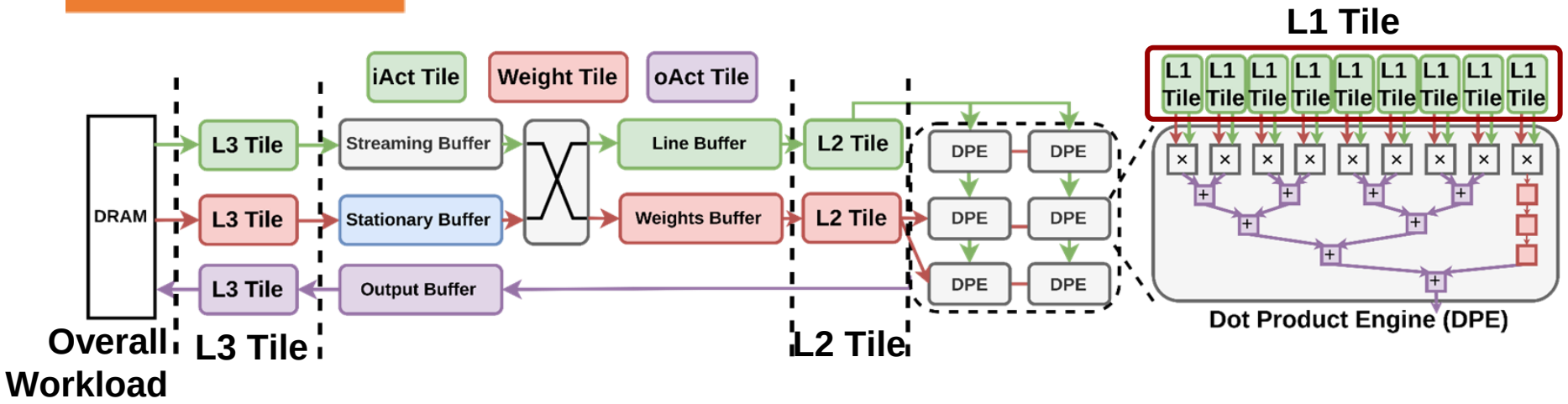
- L3 Tile: Transferred Data from DRAM to achieve continuous data access.

MAERI 2.0 Micro-architecture - Multi-level Tiling Terminology



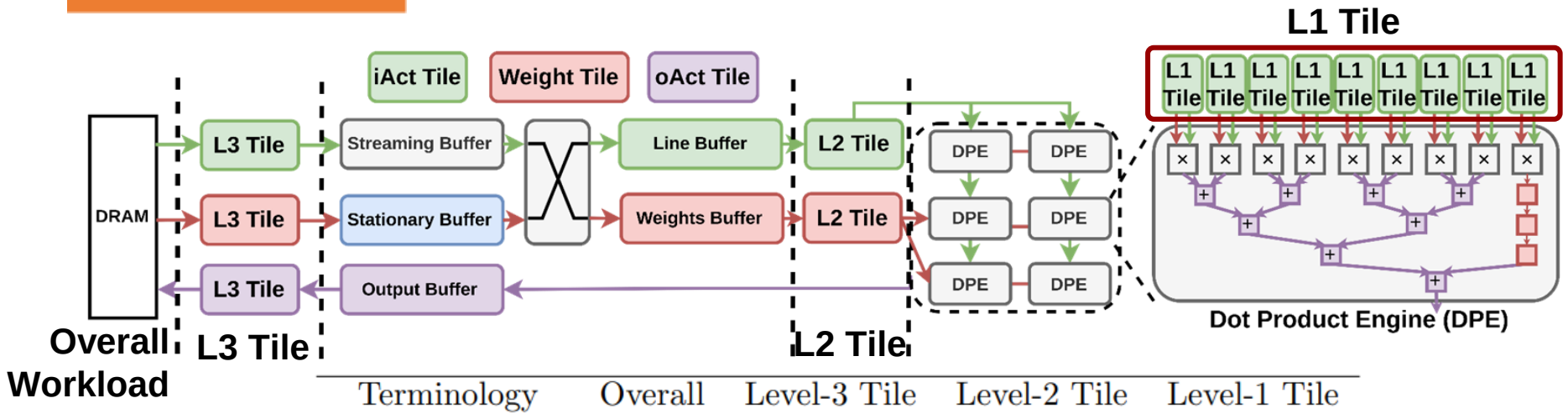
- L3 Tile: Transferred Data from DRAM to achieve continuous data access.
- L2 Tile: Data the entire DPE Array requires every cycle.

MAERI 2.0 Micro-architecture - Multi-level Tiling Terminology



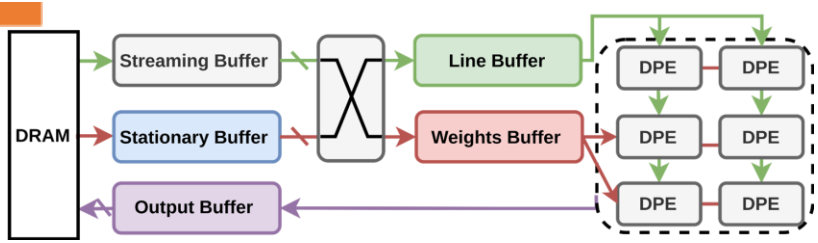
- L3 Tile: Transferred Data from DRAM to achieve continuous data access.
- L2 Tile: Data the entire DPE Array requires every cycle.
- L1 Tile: The data each single PE requires.

MAERI 2.0 Micro-architecture - Multi-level Tiling Terminology

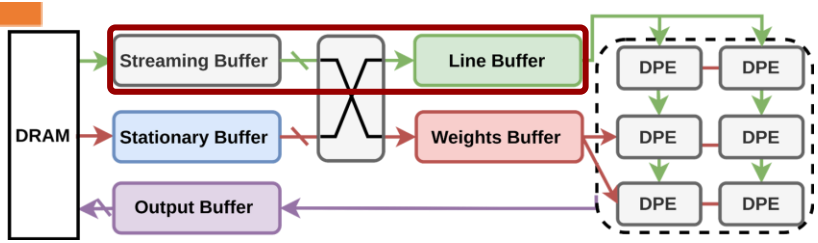


Terminology	Overall	Level-3 Tile	Level-2 Tile	Level-1 Tile
Input Channel	C	T_C	1	1
Input Height	X	T_X	1	1
Input Width	Y	T_Y	Y_P	1
Kernel Number	K	T_K	K_P	1
Weight Height	R	R	R	1
Weight Width	S	S	S	1
Output Height	X_O	T_{X_O}	1	1
Output Width	Y_O	T_{Y_O}	$Y_P/Stride$	1

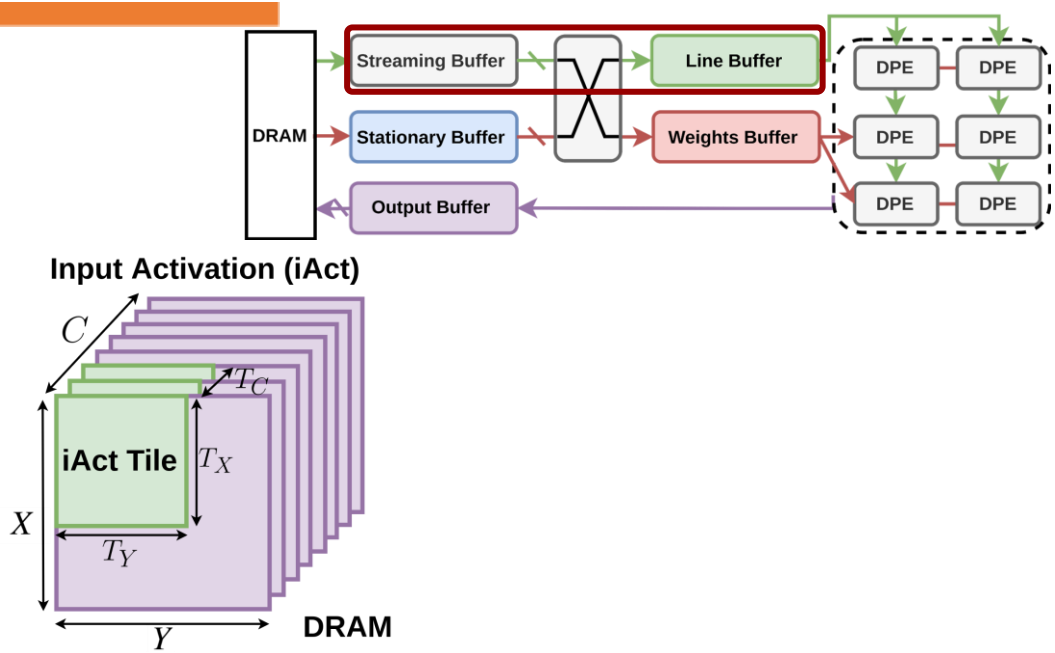
MAERI 2.0 Micro-architecture - Buffers for iAct



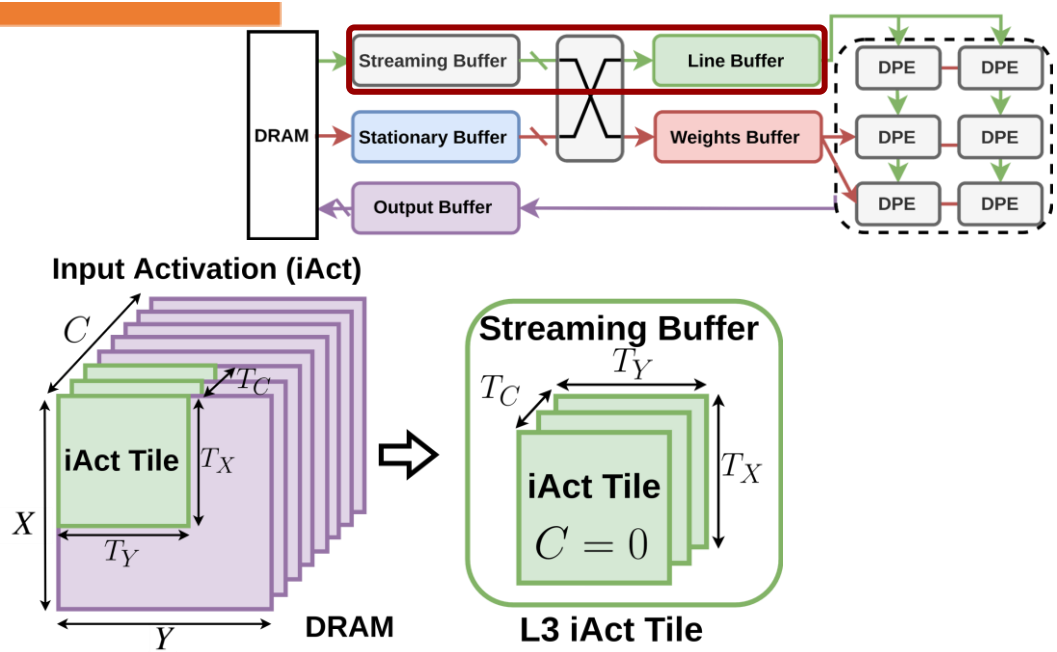
MAERI 2.0 Micro-architecture - Buffers for iAct



MAERI 2.0 Micro-architecture - Buffers for iAct

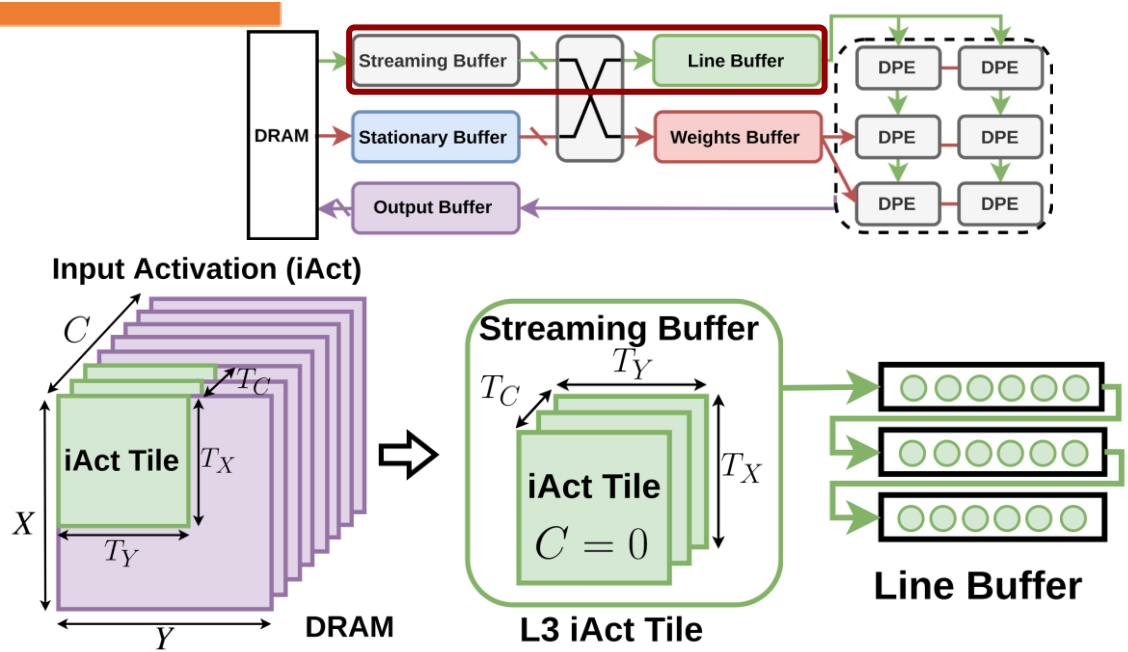


MAERI 2.0 Micro-architecture - Buffers for iAct



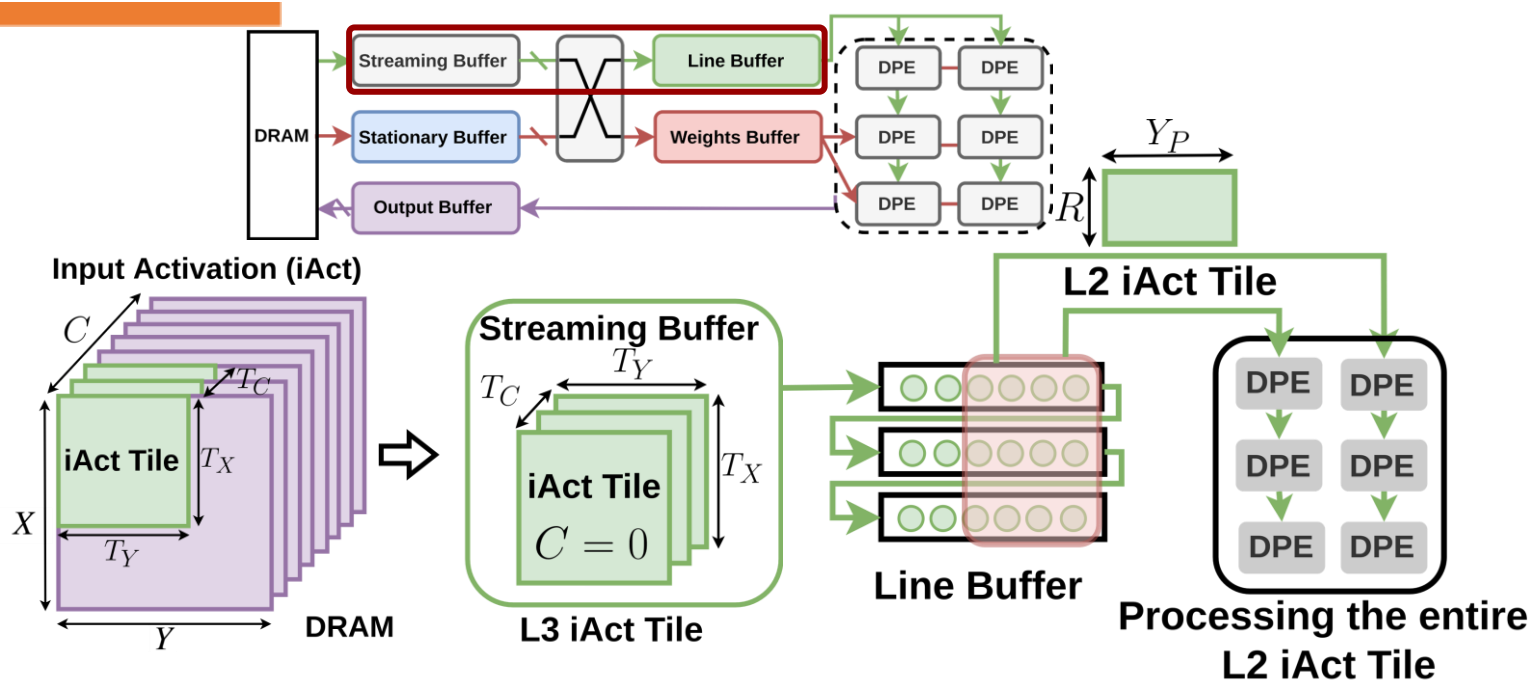
- **Streaming Buffer → Reuse L2 iAct tiles for multiple kernels**

MAERI 2.0 Micro-architecture - Buffers for iAct



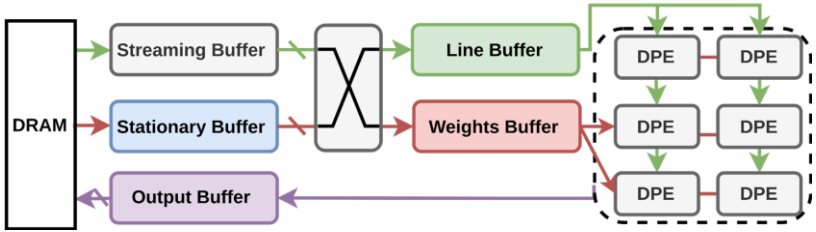
- **Streaming Buffer → Reuse L2 iAct tiles for multiple kernels**
- **Line Buffer →Reuse overlapped iAct for sliding windows.**

MAERI 2.0 Micro-architecture - Buffers for iAct

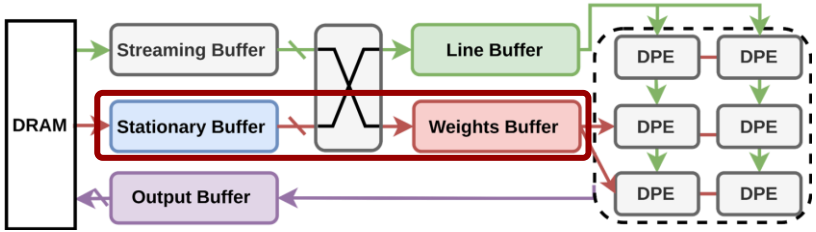


- **Streaming Buffer → Reuse L2 iAct tiles for multiple kernels**
- **Line Buffer →Reuse overlapped iAct for sliding windows.**
- **iAct Reuse in different DPE rows → Reuse iAct by multiple kernels**

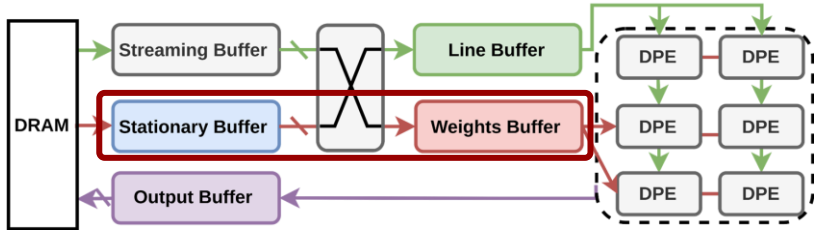
MAERI 2.0 Micro-architecture - Buffers for weights

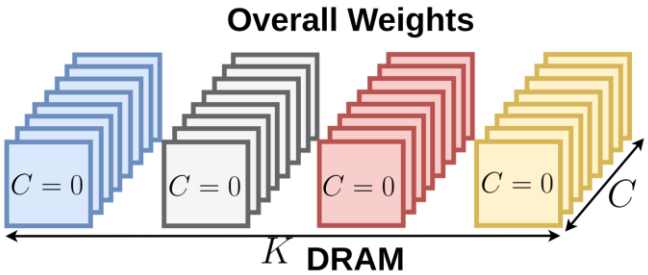


MAERI 2.0 Micro-architecture - Buffers for weights

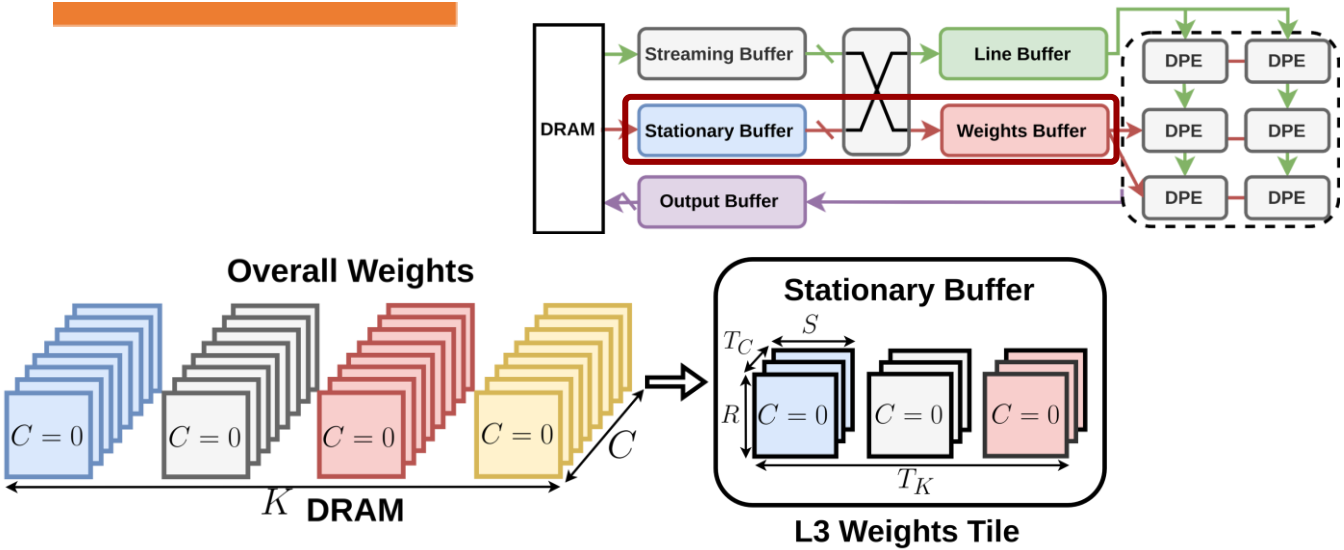


MAERI 2.0 Micro-architecture - Buffers for weights



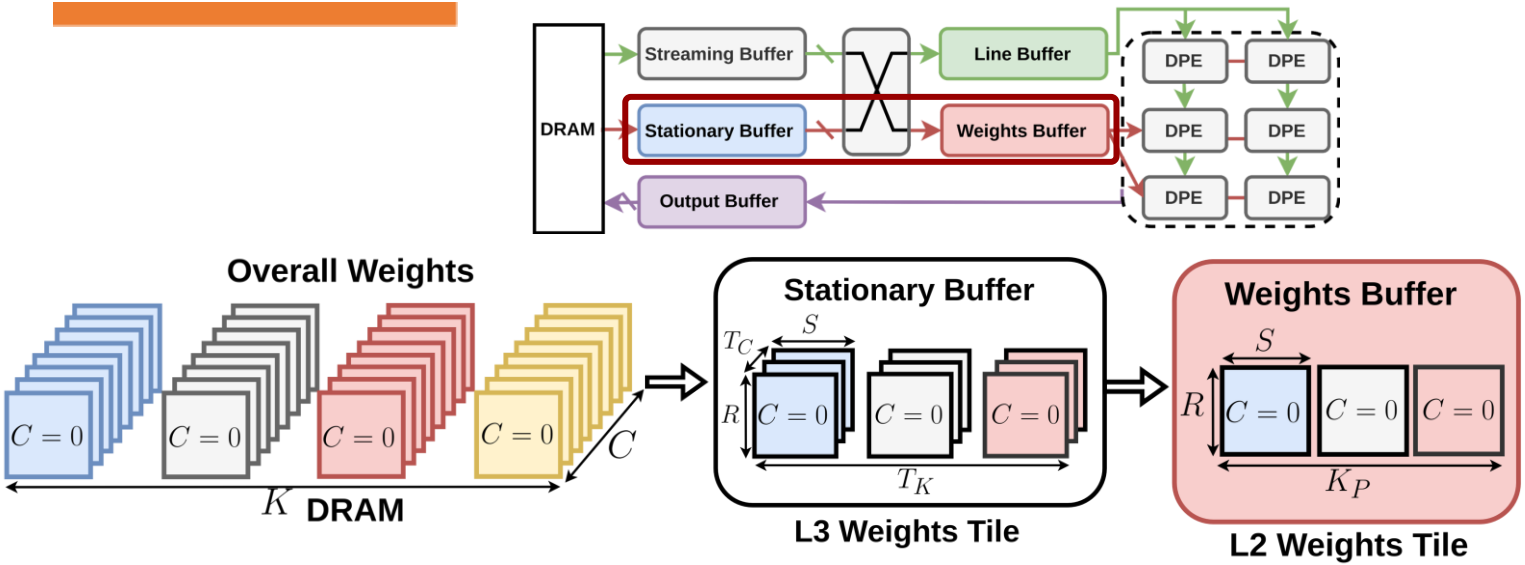


MAERI 2.0 Micro-architecture - Buffers for weights



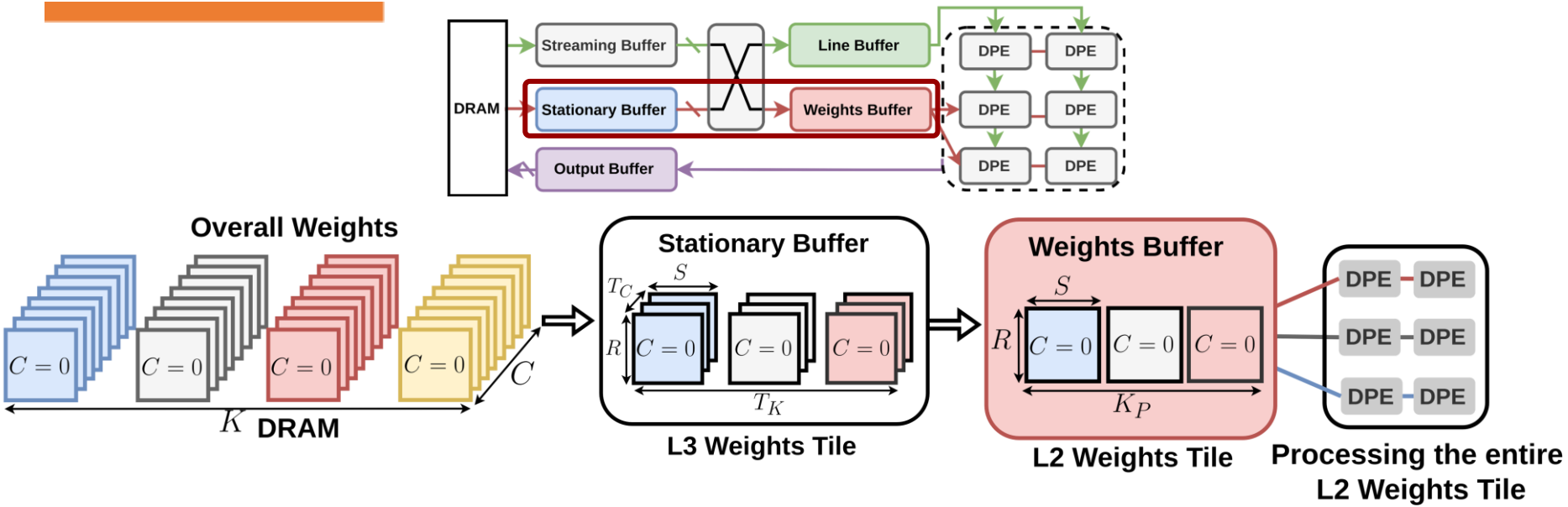
- **Stationary Buffer → Enable weights reuse on different L2 weights tile**
- **Weights Buffer → Double buffer for L2 weights tile.**

MAERI 2.0 Micro-architecture - Buffers for weights



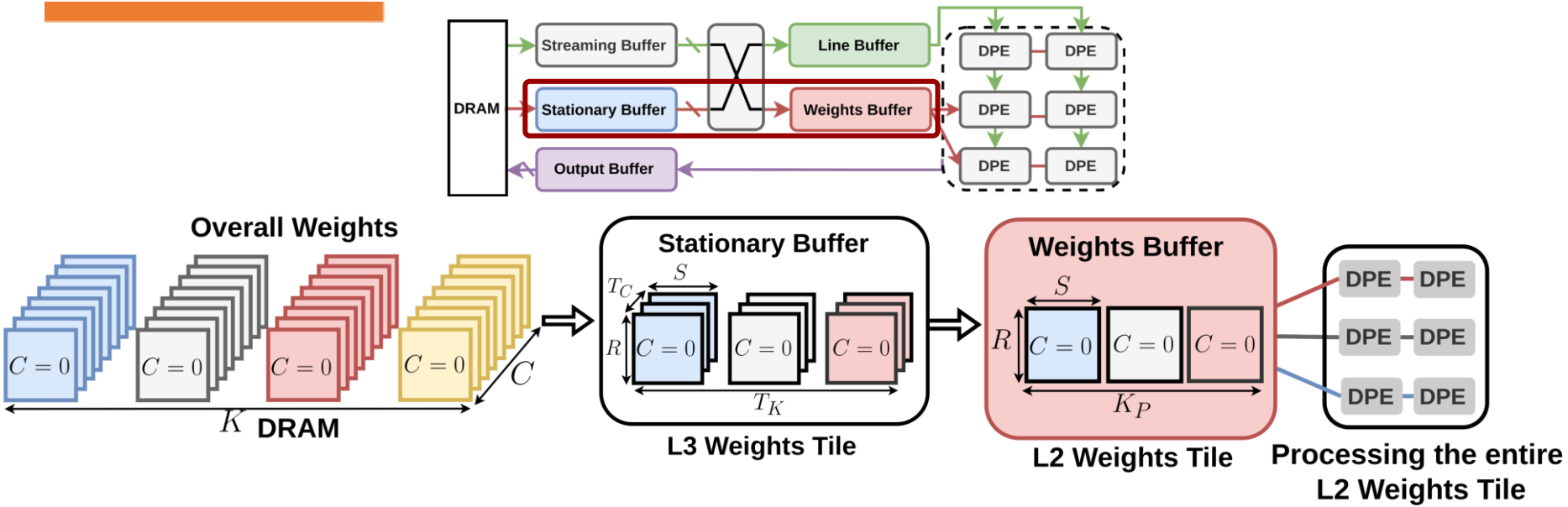
- **Stationary Buffer → Enable weights reuse on different L2 weights tile**
- **Weights Buffer → Double buffer for L2 weights tile.**
- **Weights are broadcasted in different DPE columns**

MAERI 2.0 Micro-architecture - Buffers for weights



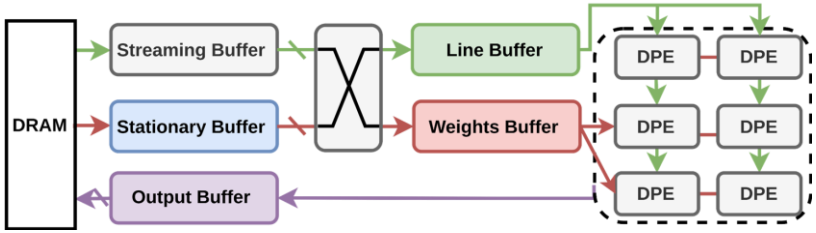
- **Stationary Buffer → Enable weights reuse on different L2 weights tile**
- **Weights Buffer → Double buffer for L2 weights tile.**
- **Weights are broadcasted in different DPE columns**

MAERI 2.0 Micro-architecture - Buffers for weights

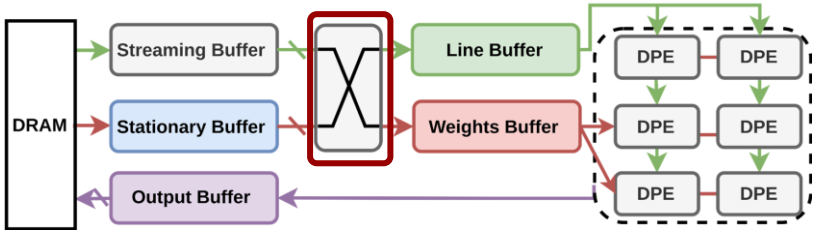


- **Stationary Buffer → Enable weights reuse on different L2 weights tile**
- **Weights Buffer → Double buffer for L2 weights tile.**
- **Weights are broadcasted in different DPE columns**
- **Buffer Write and Data Forward happen in parallel for fetching first weights**

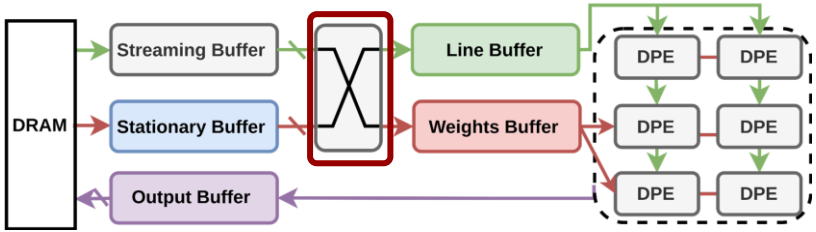
MAERI 2.0 Micro-arch: Crossbar Weights Stationary



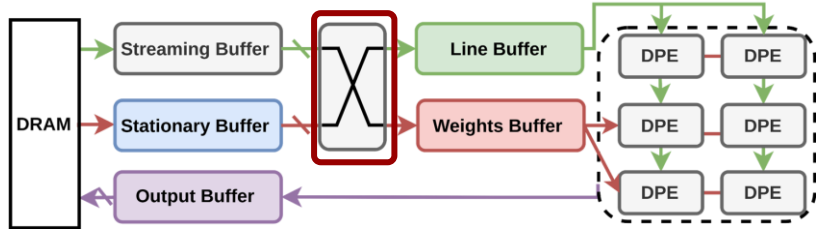
MAERI 2.0 Micro-arch: Crossbar Weights Stationary

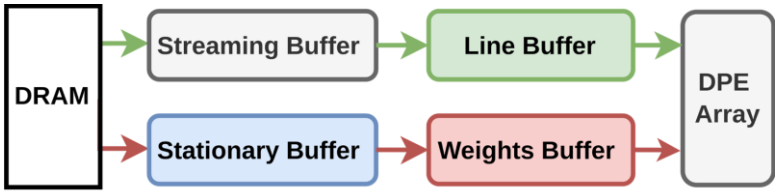


MAERI 2.0 Micro-arch: Crossbar Weights Stationary

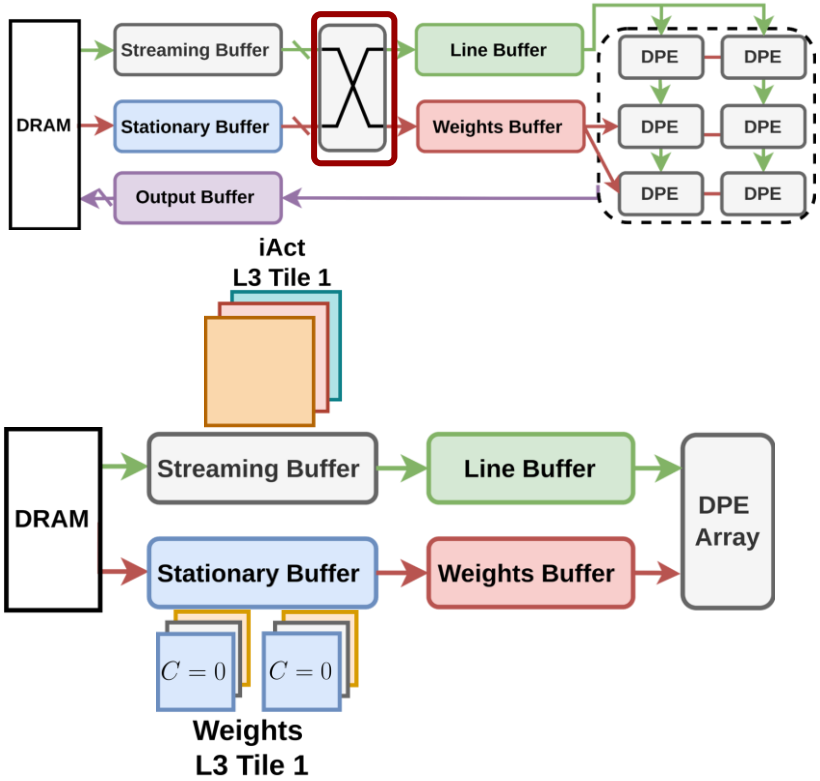


MAERI 2.0 Micro-arch: Crossbar Weights Stationary

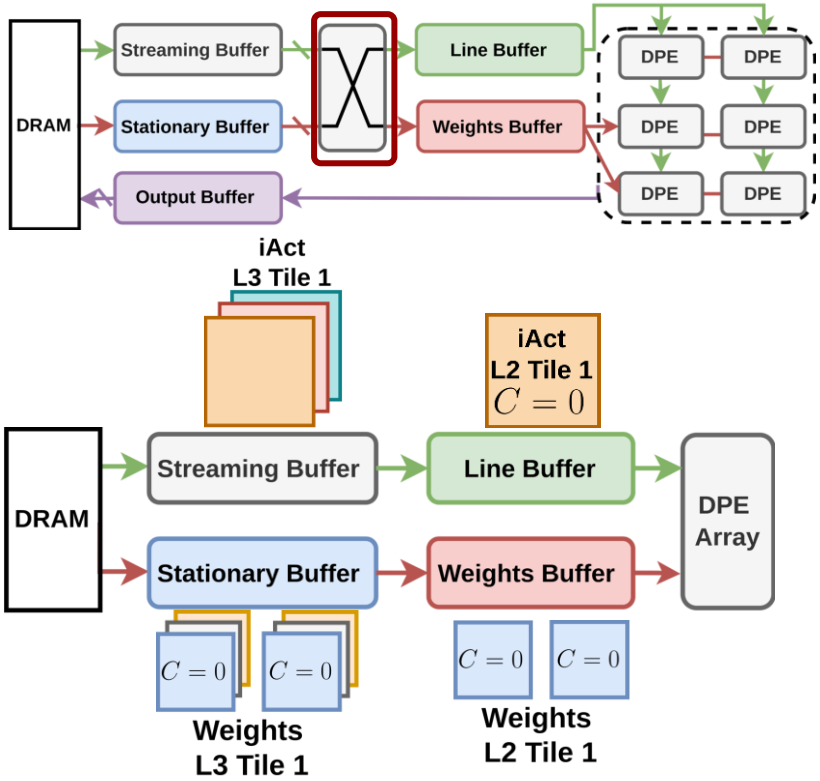




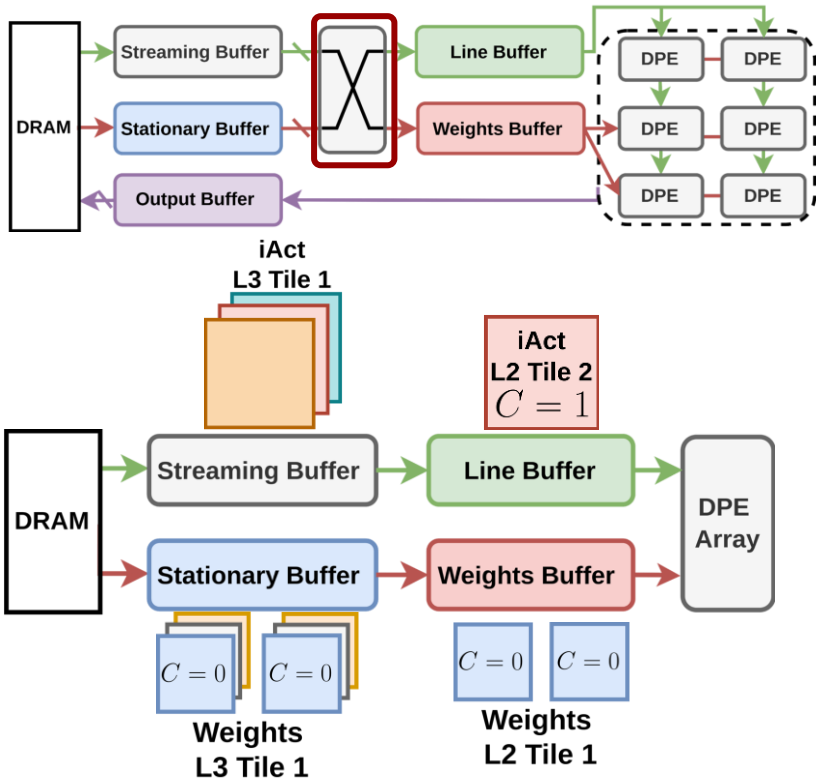
MAERI 2.0 Micro-arch: Crossbar Weights Stationary



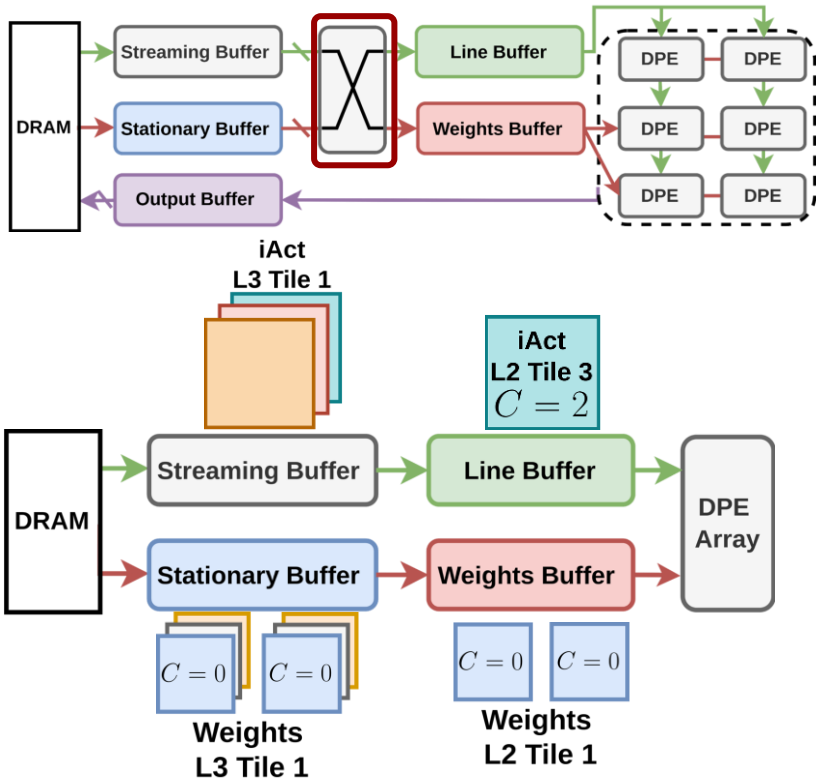
MAERI 2.0 Micro-arch: Crossbar Weights Stationary



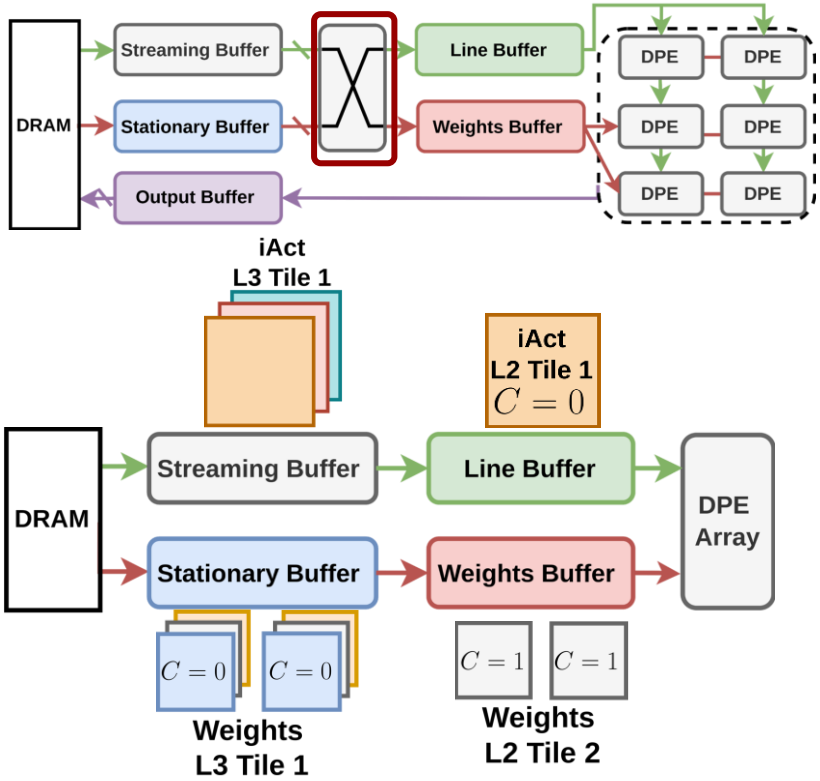
MAERI 2.0 Micro-arch: Crossbar Weights Stationary



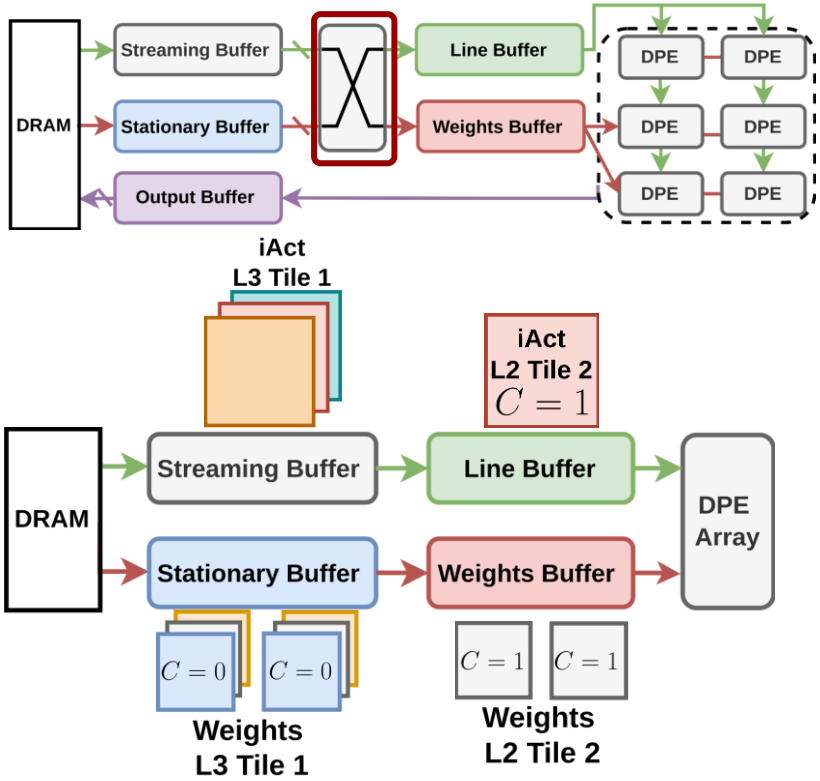
MAERI 2.0 Micro-arch: Crossbar Weights Stationary



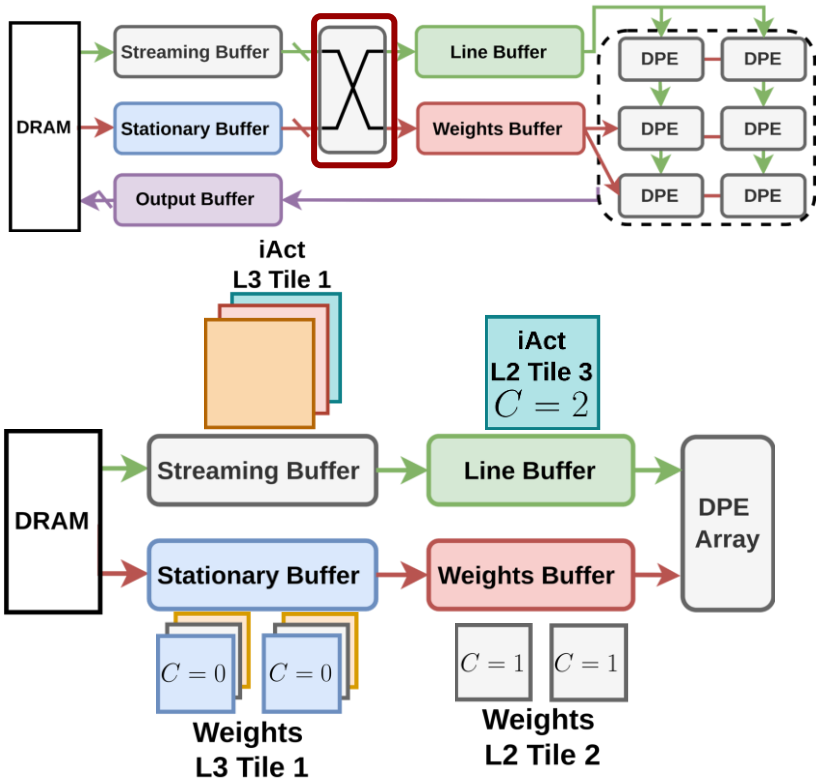
MAERI 2.0 Micro-arch: Crossbar Weights Stationary



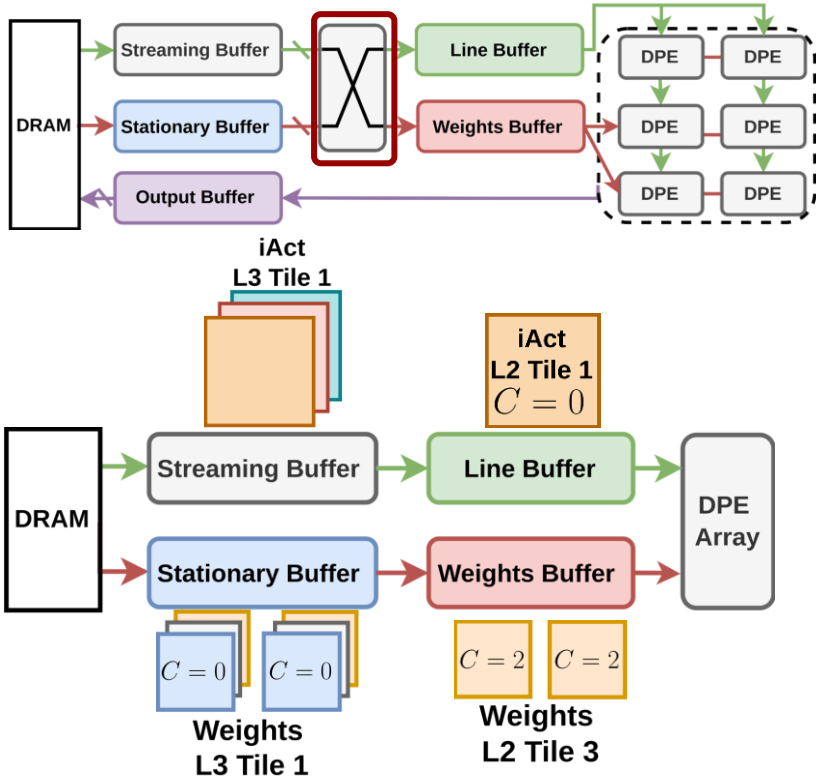
MAERI 2.0 Micro-arch: Crossbar Weights Stationary



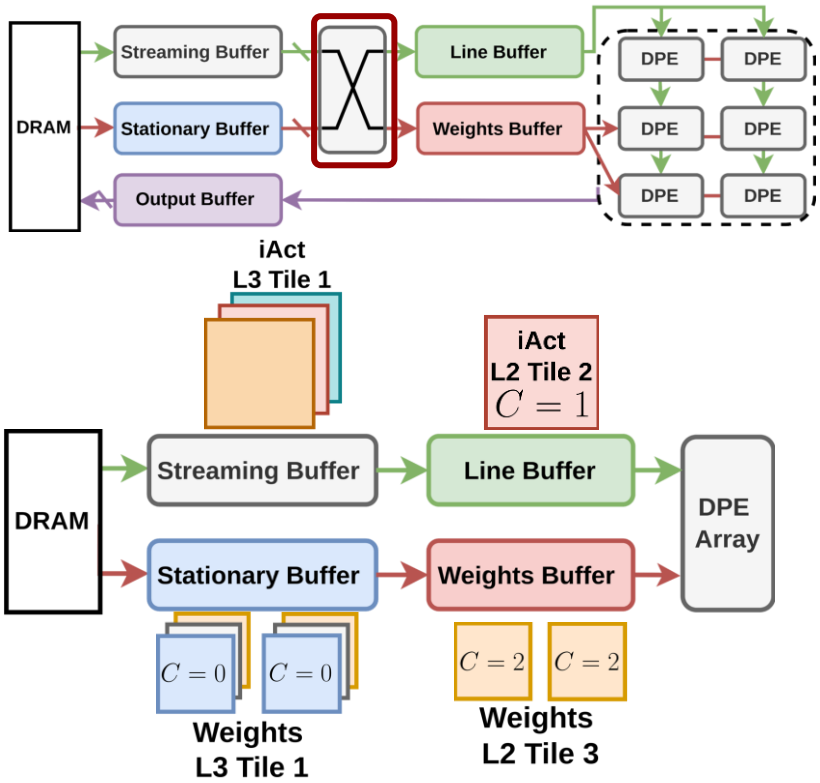
MAERI 2.0 Micro-arch: Crossbar Weights Stationary



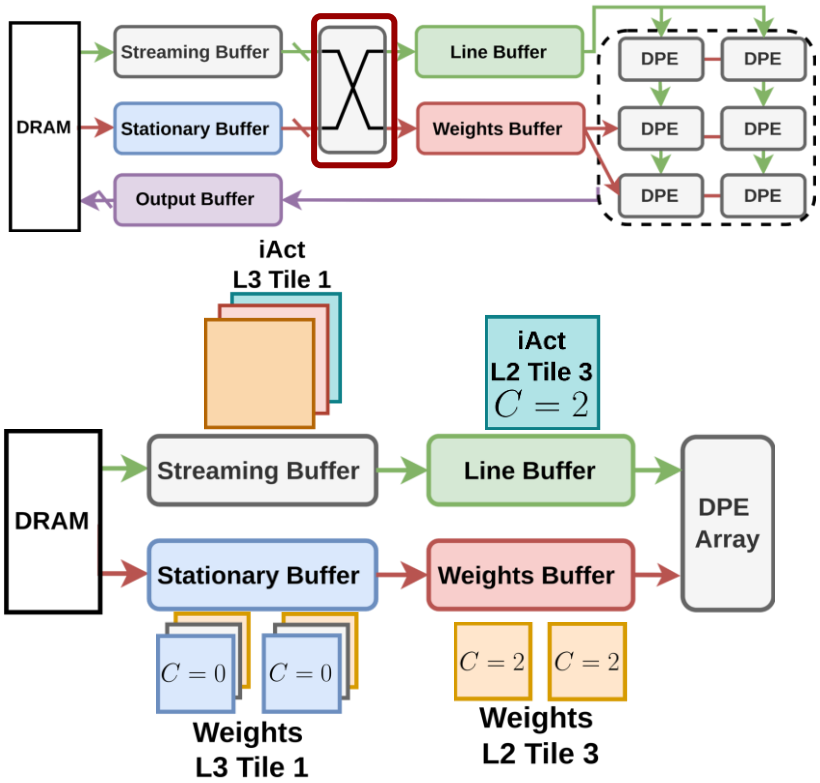
MAERI 2.0 Micro-arch: Crossbar Weights Stationary



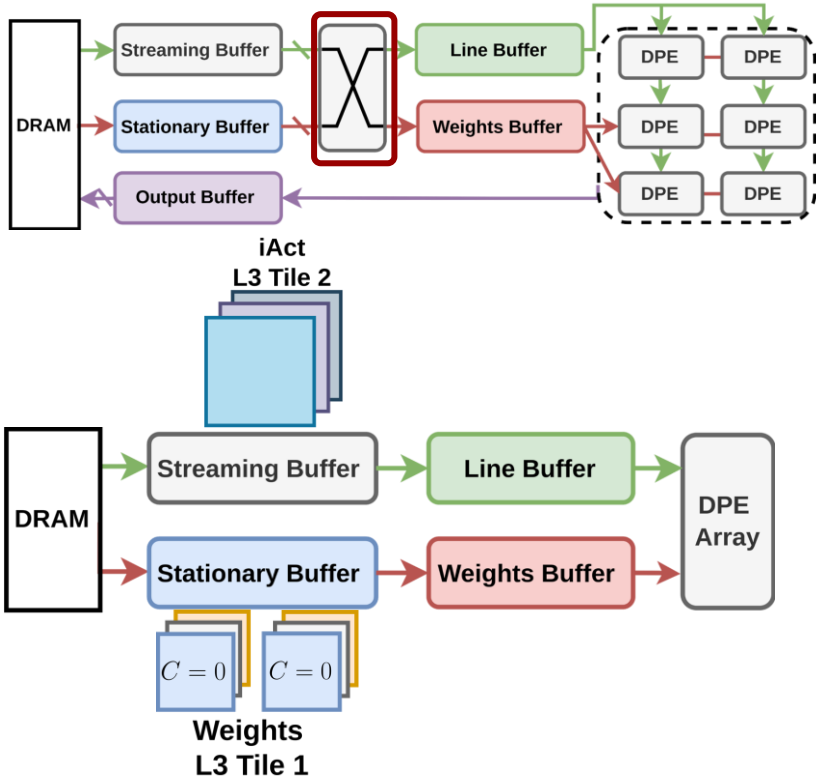
MAERI 2.0 Micro-arch: Crossbar Weights Stationary



MAERI 2.0 Micro-arch: Crossbar Weights Stationary

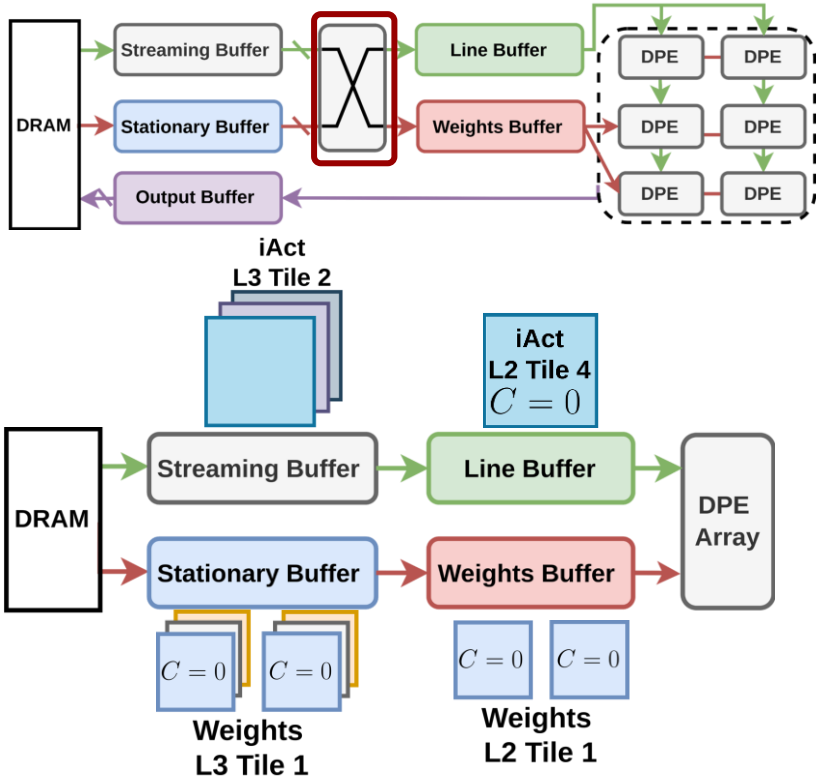


MAERI 2.0 Micro-arch: Crossbar Weights Stationary



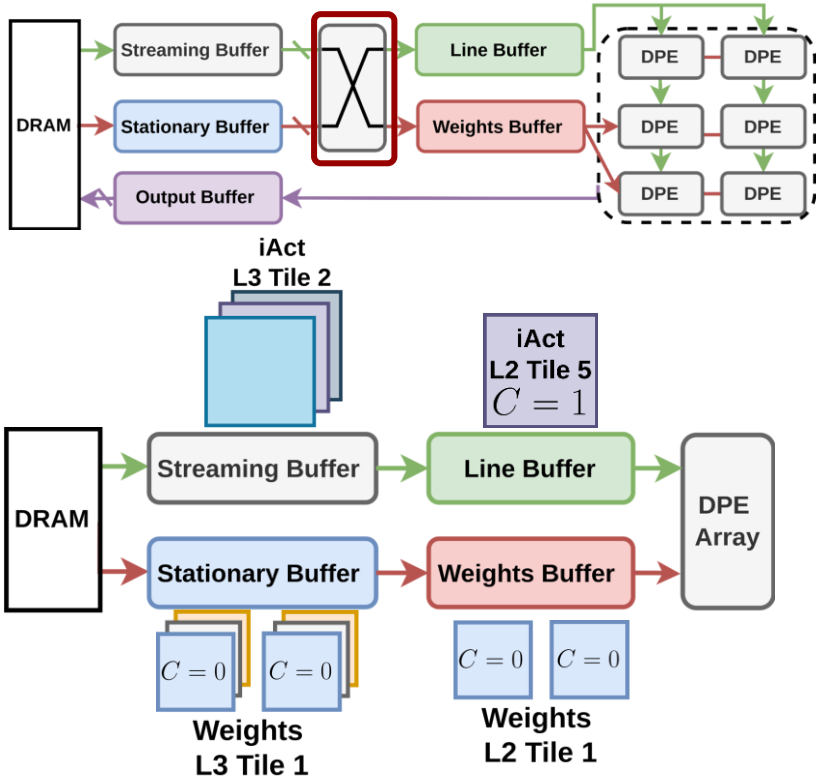
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



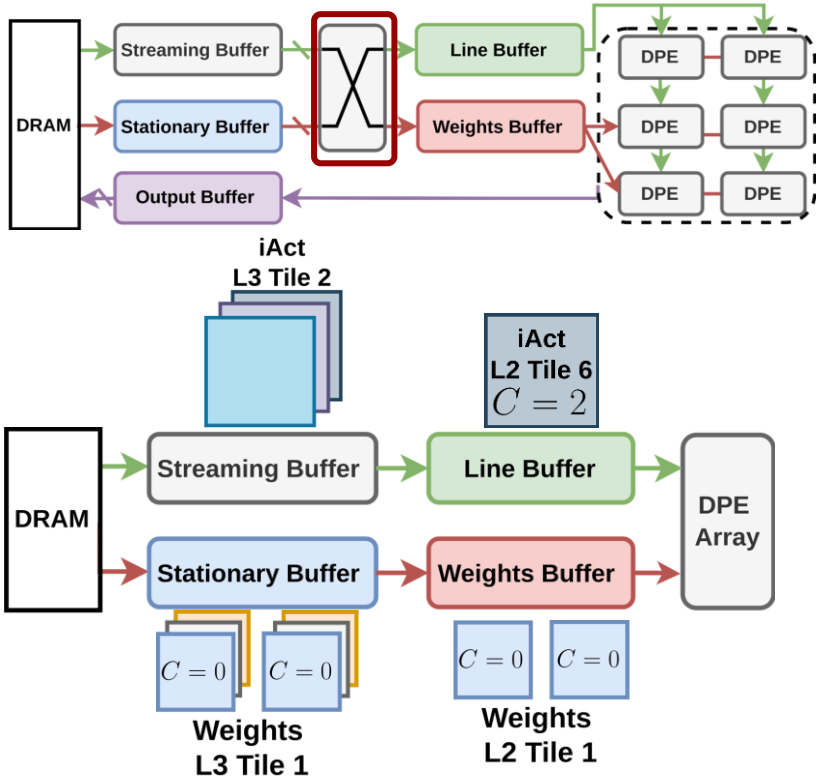
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



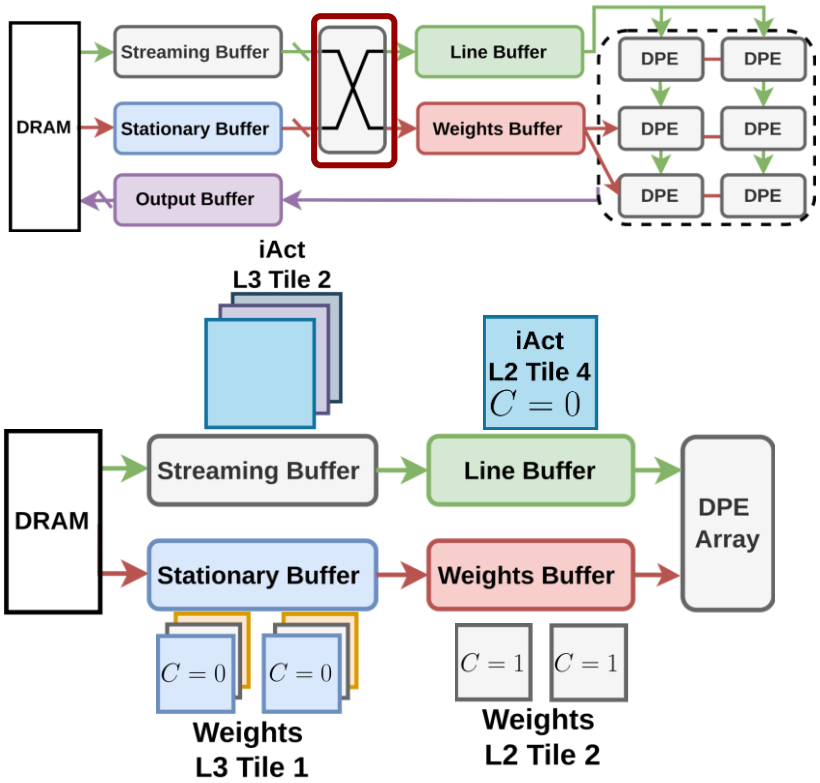
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



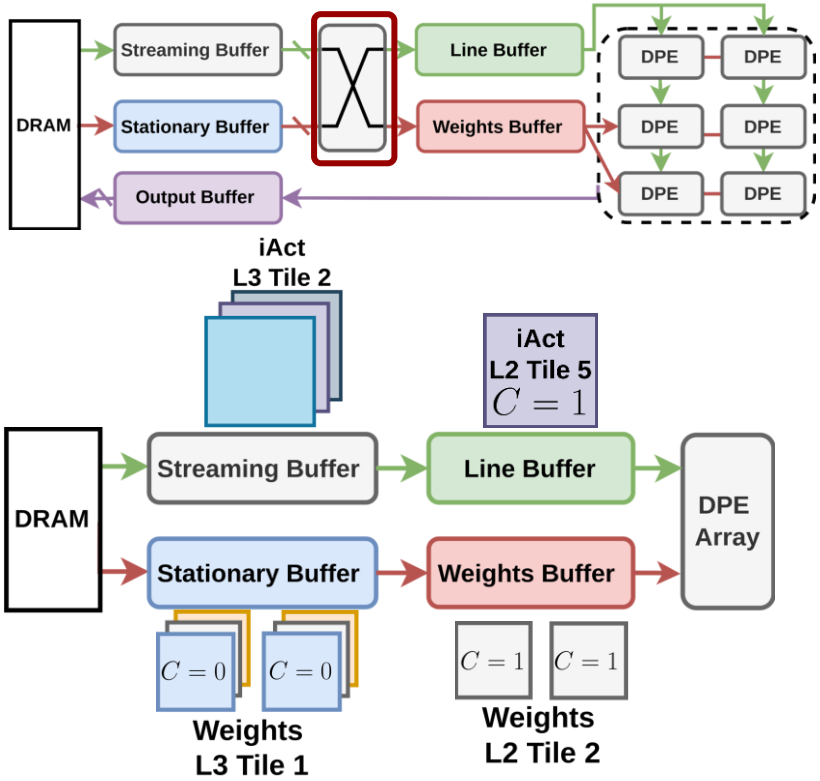
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



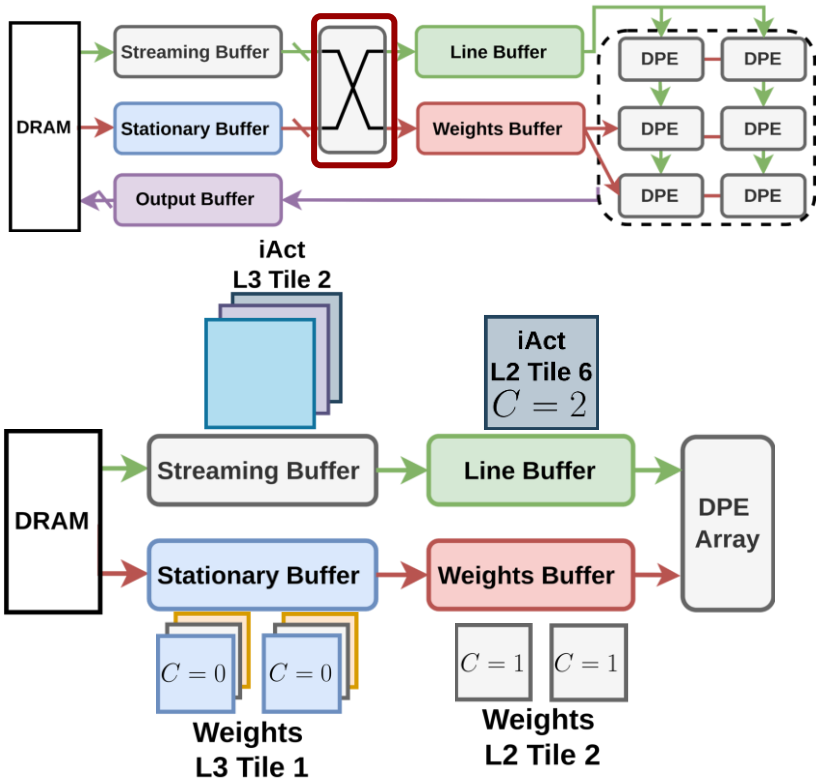
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



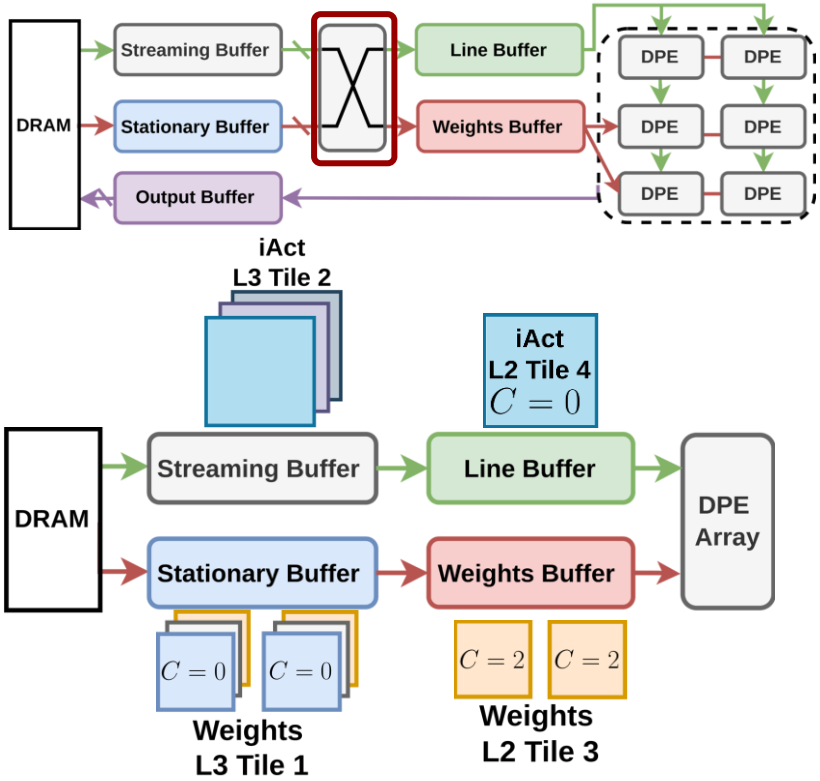
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



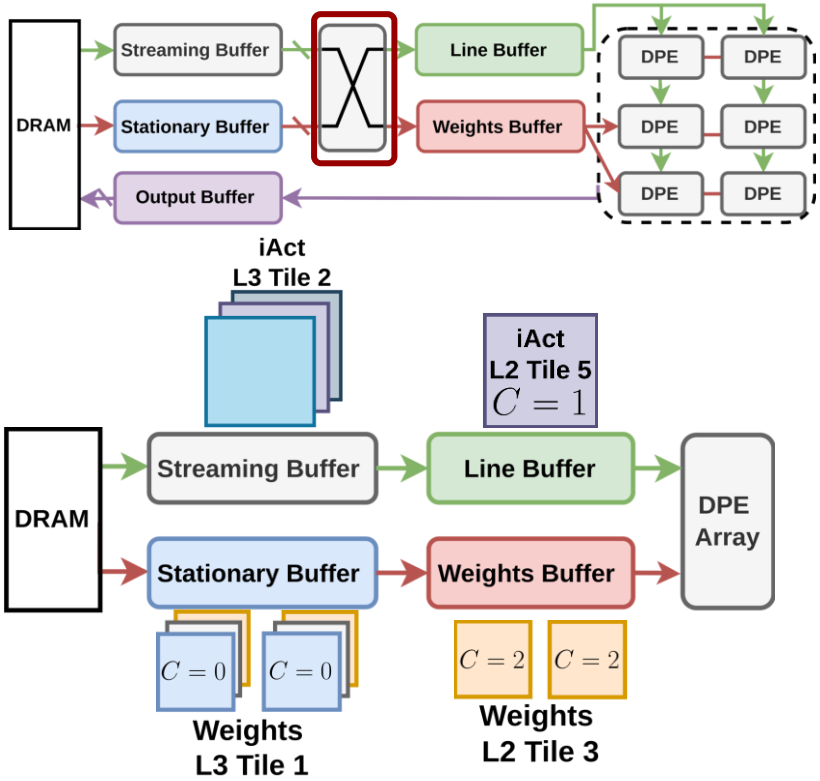
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



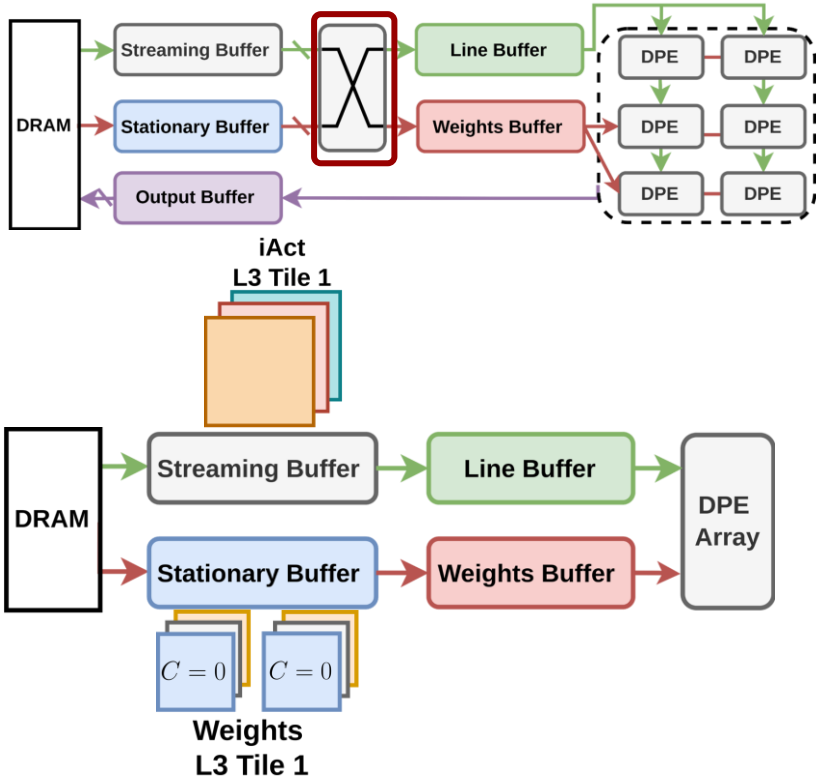
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



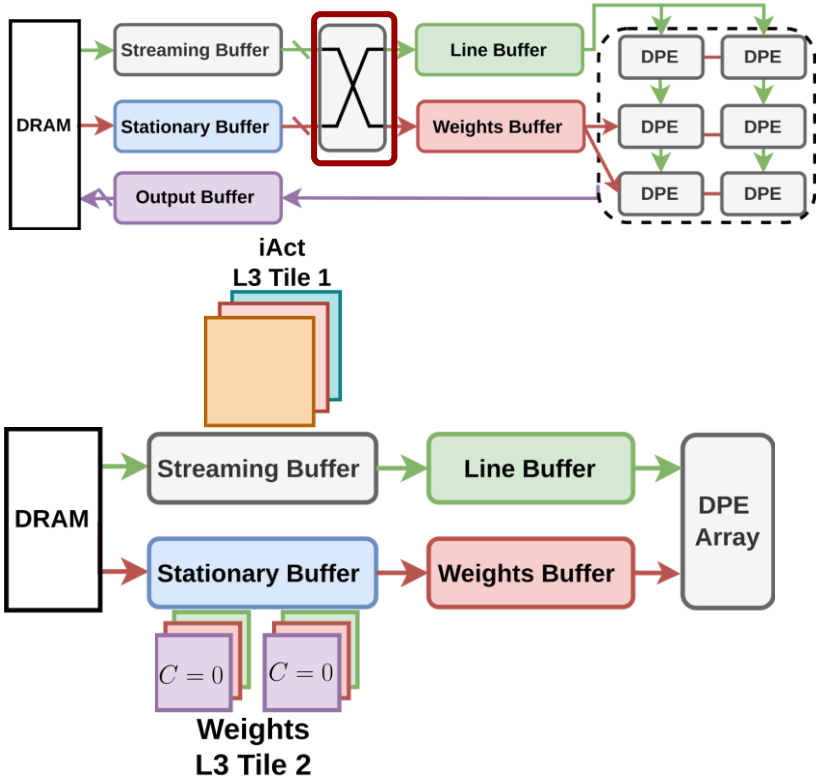
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



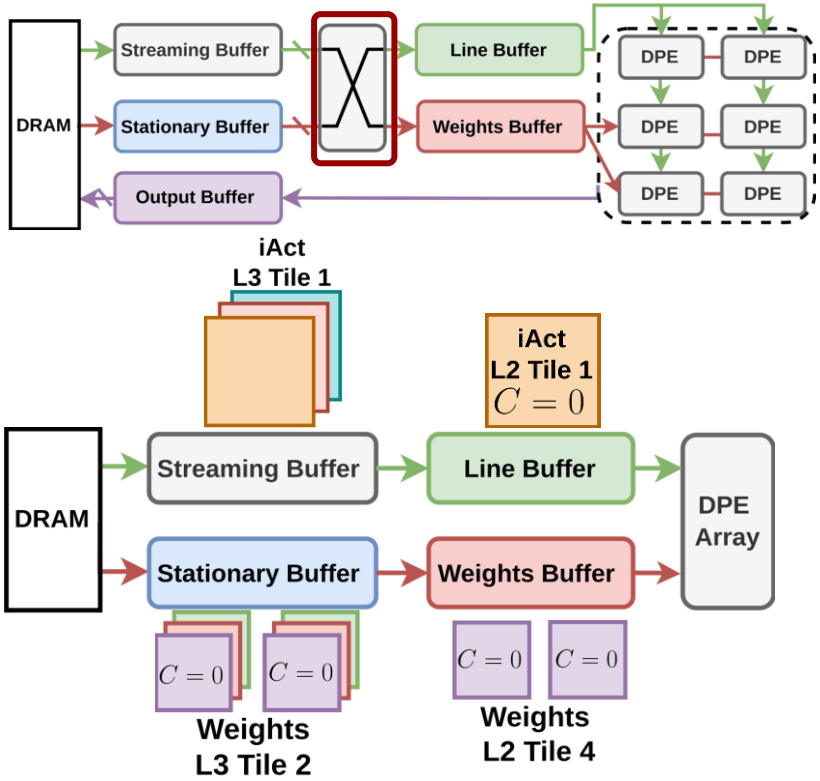
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



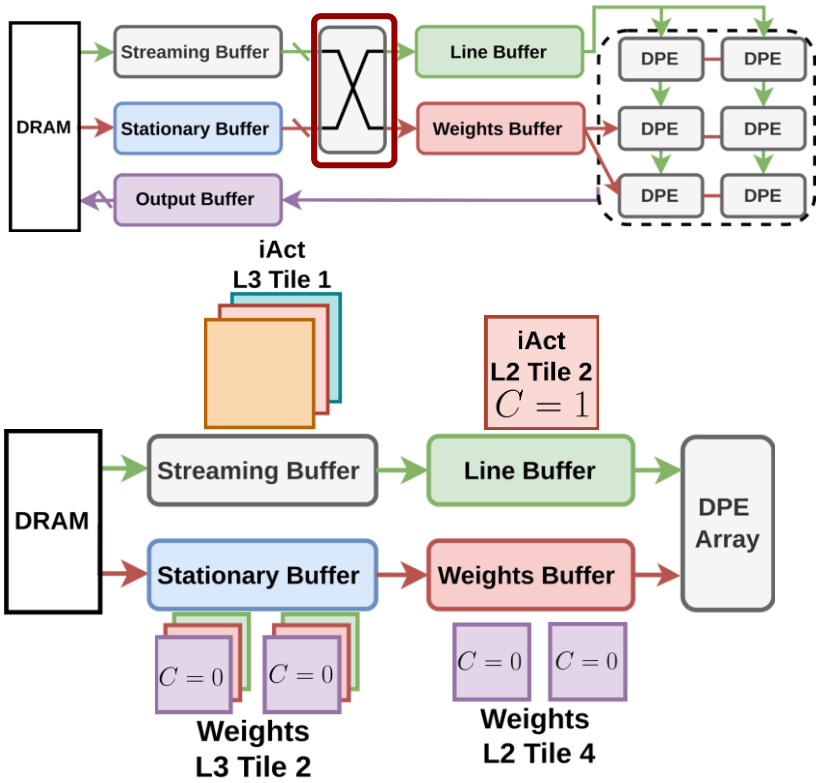
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



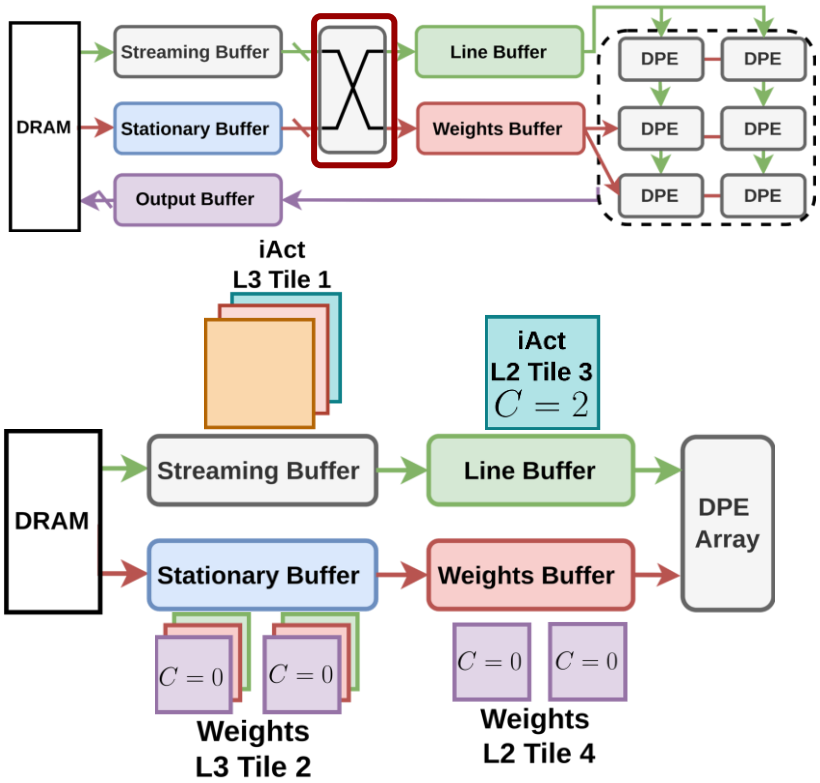
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



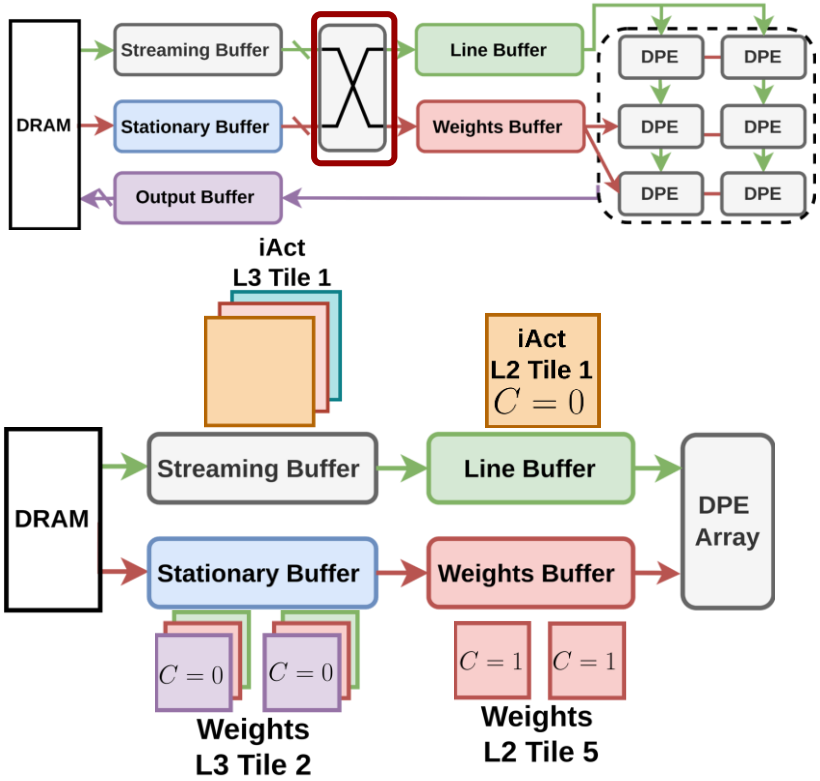
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



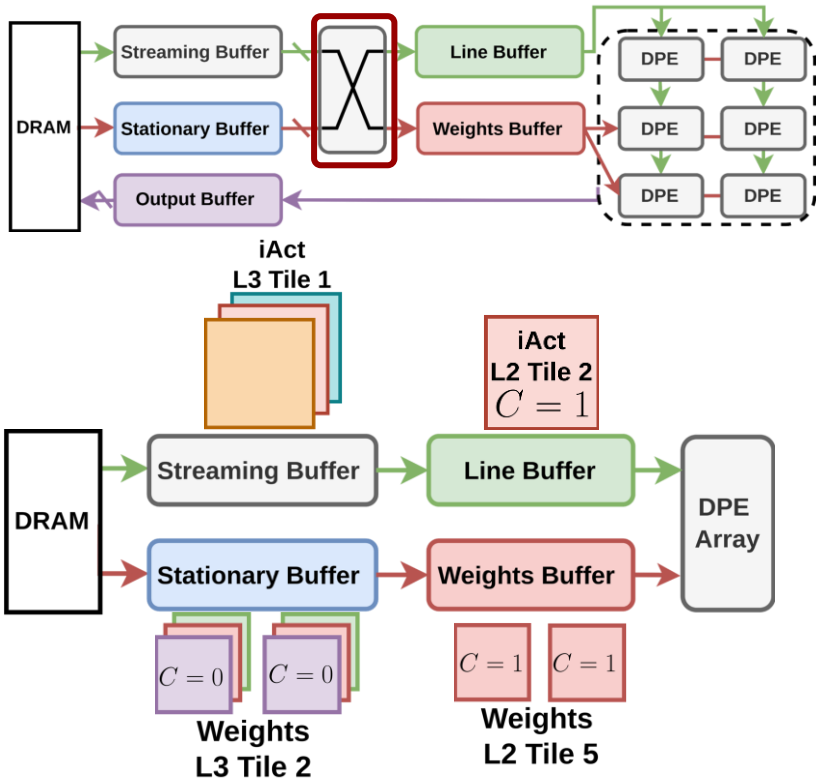
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



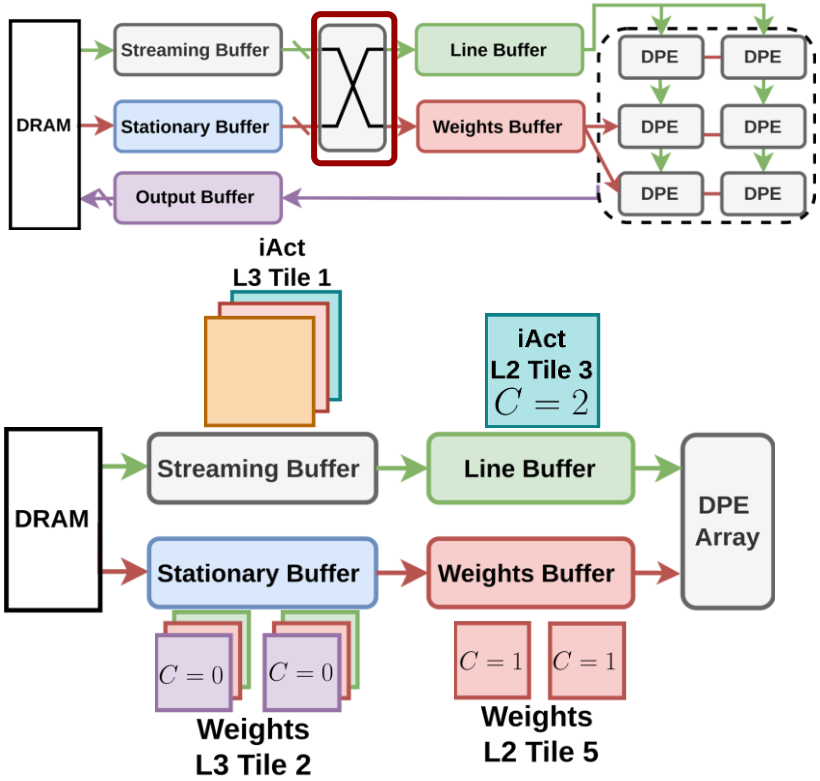
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



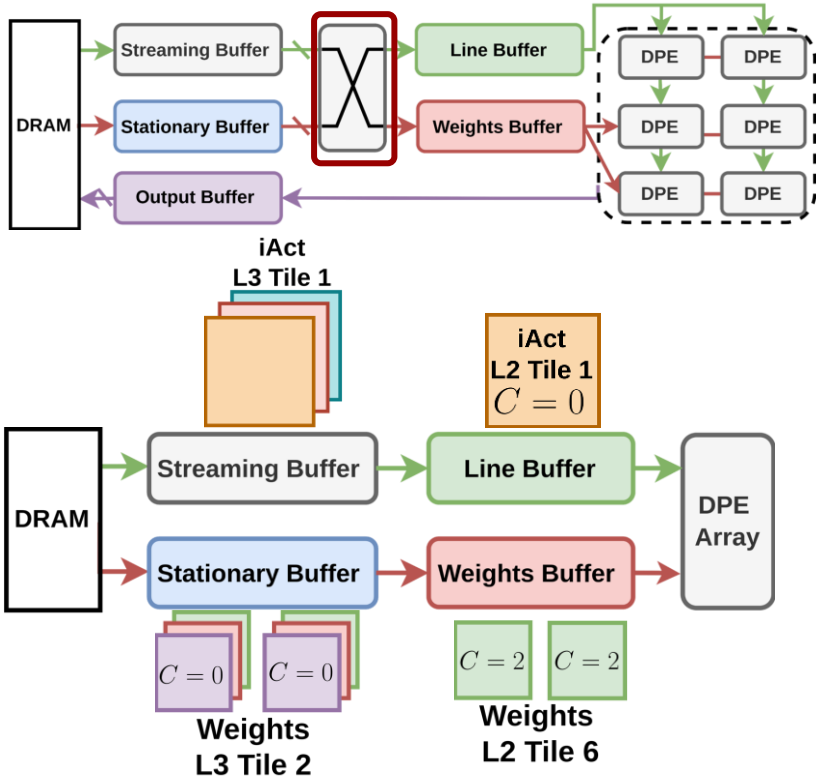
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



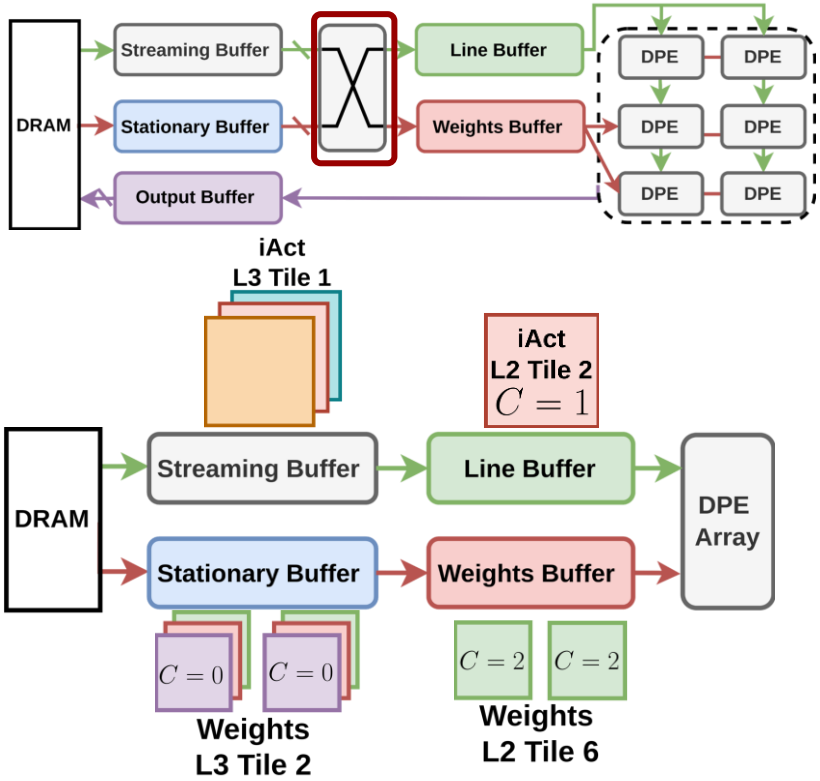
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



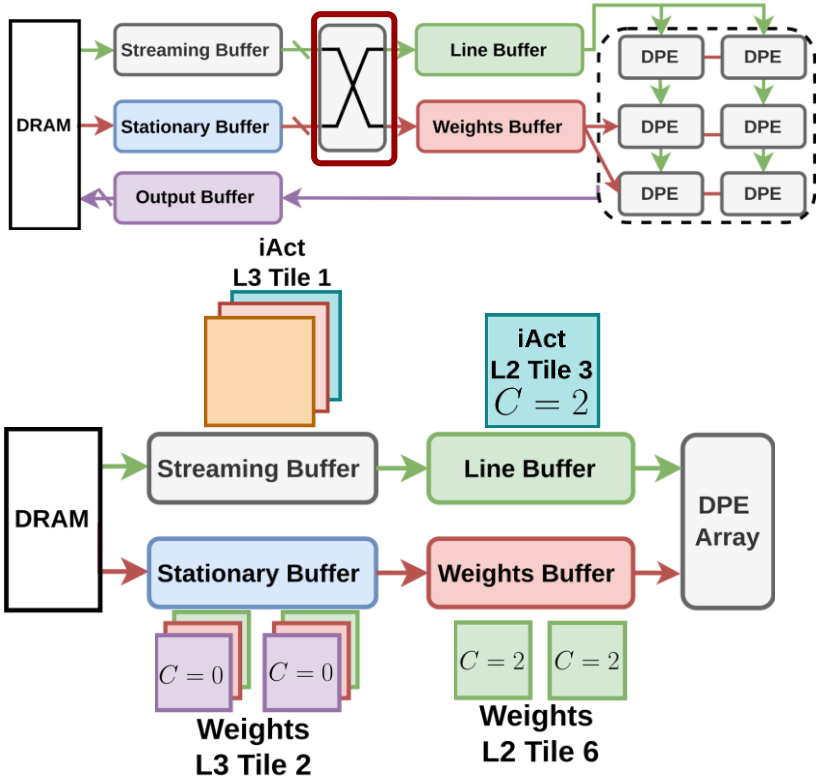
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



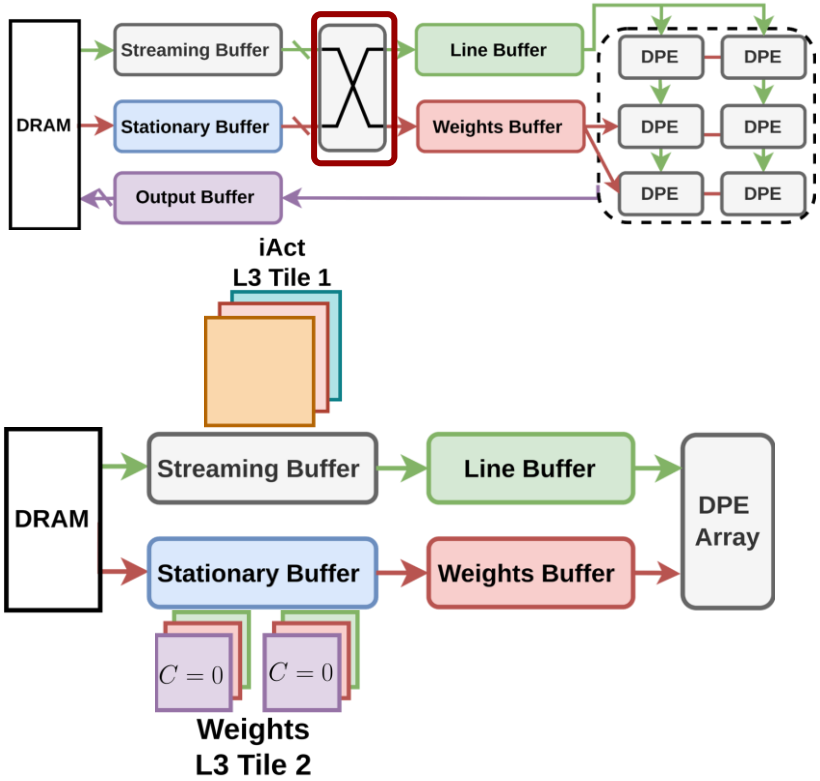
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



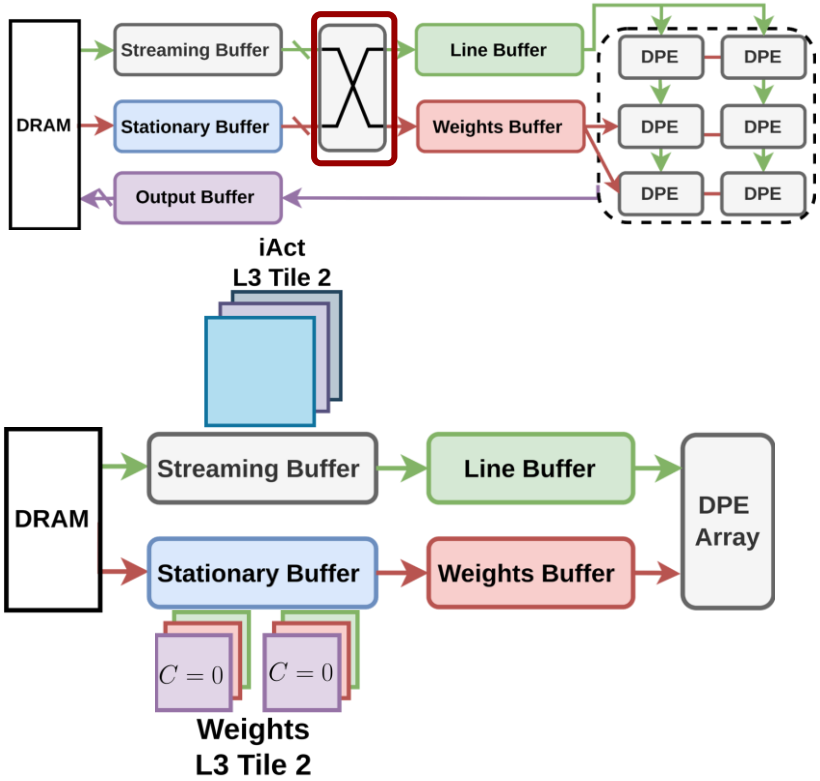
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



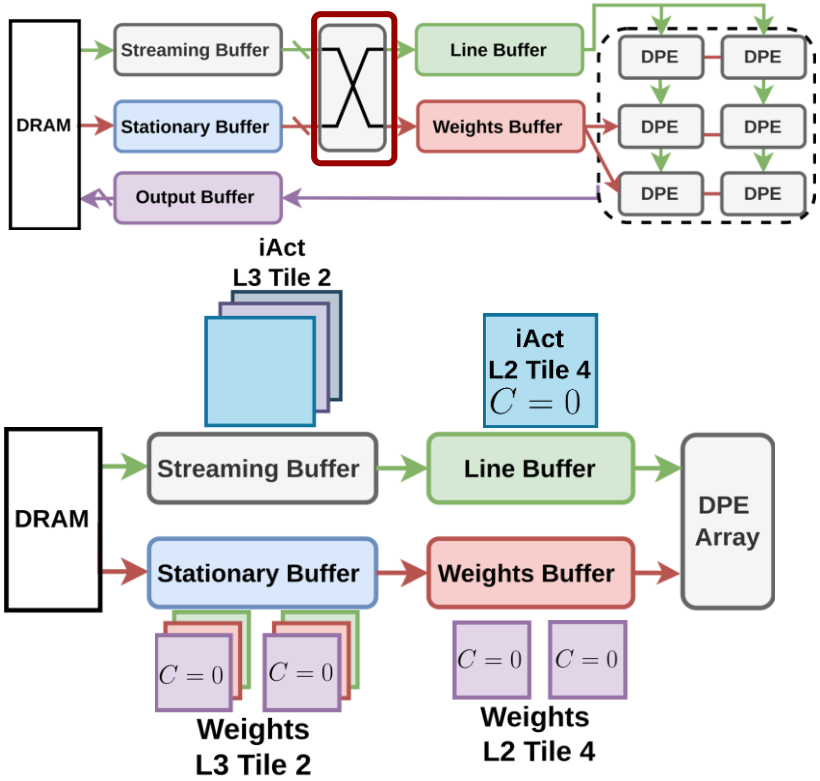
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



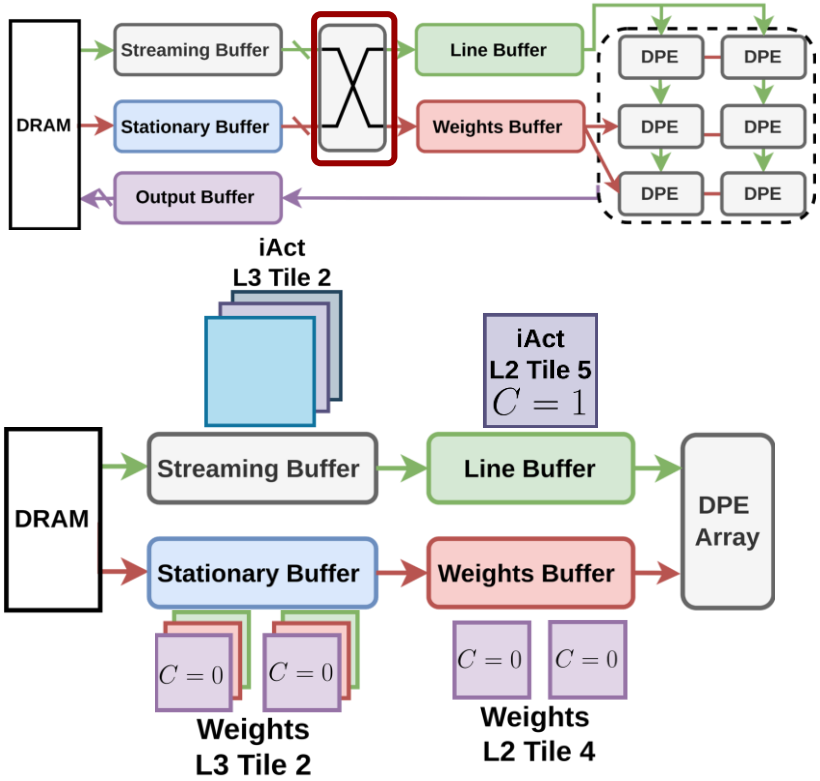
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



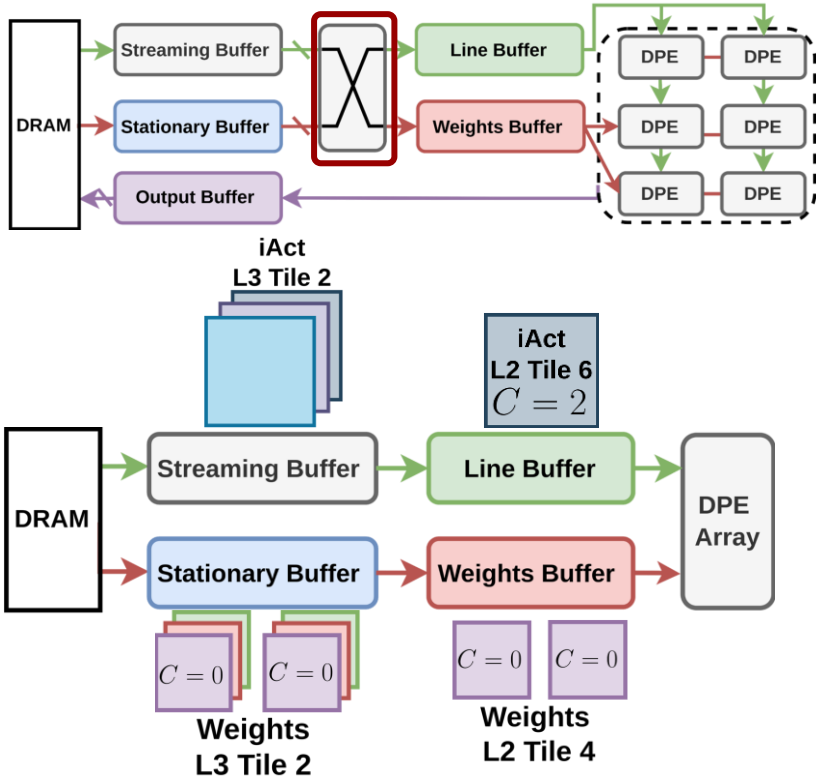
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



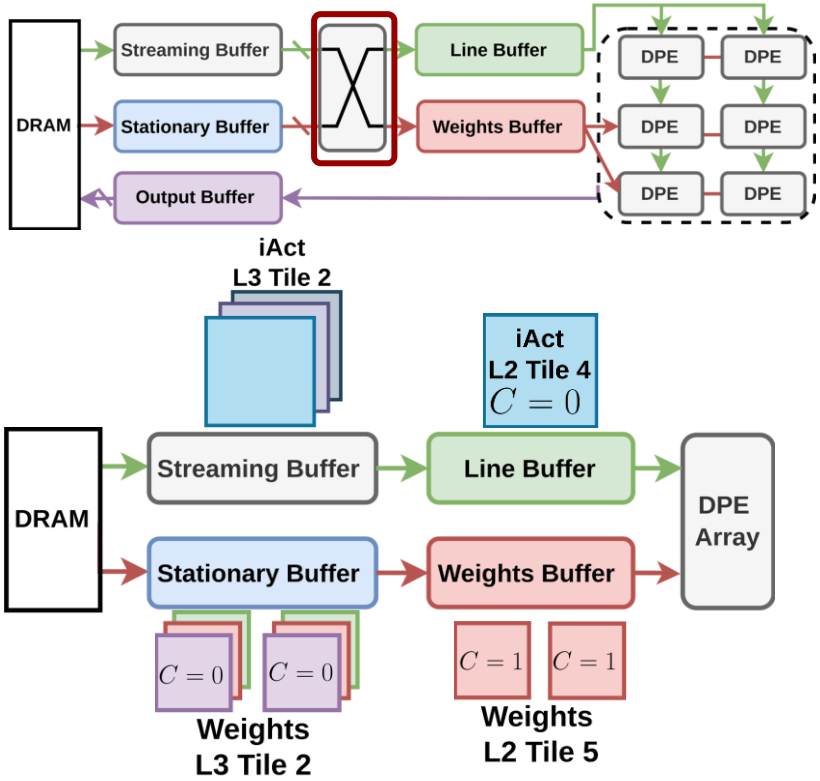
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



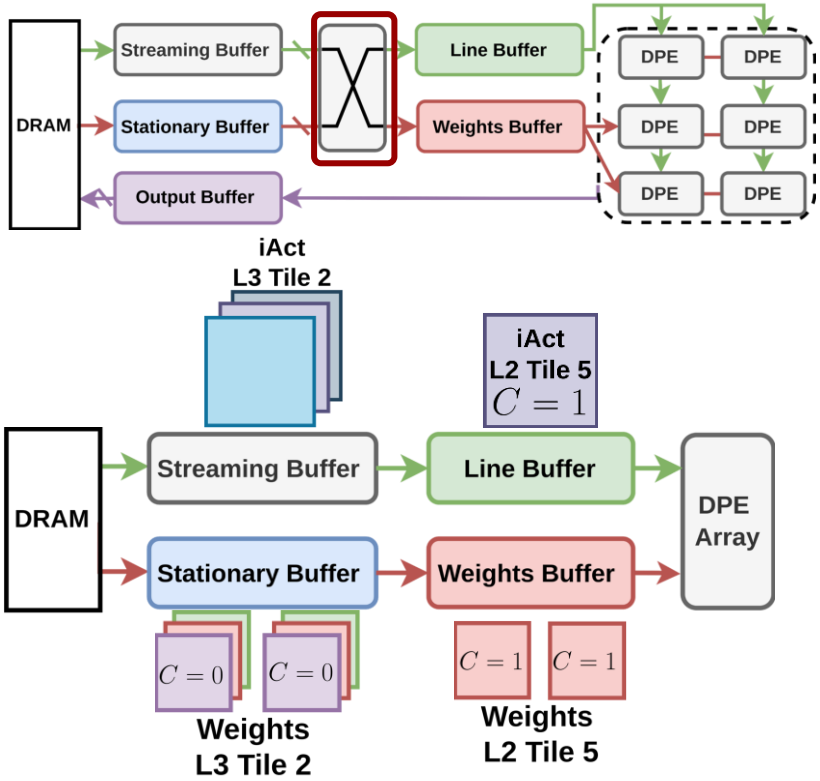
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



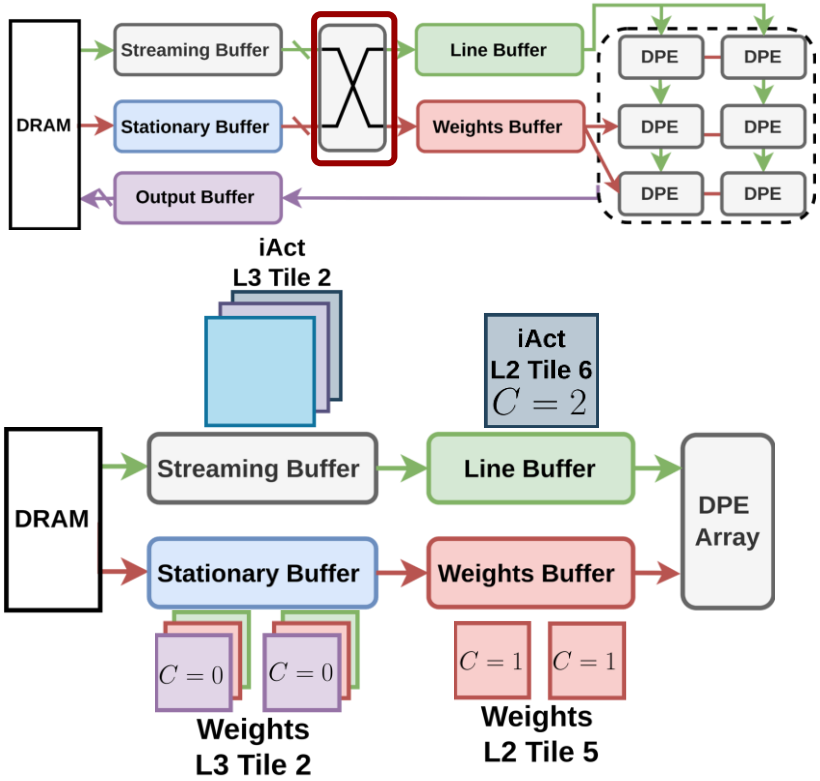
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



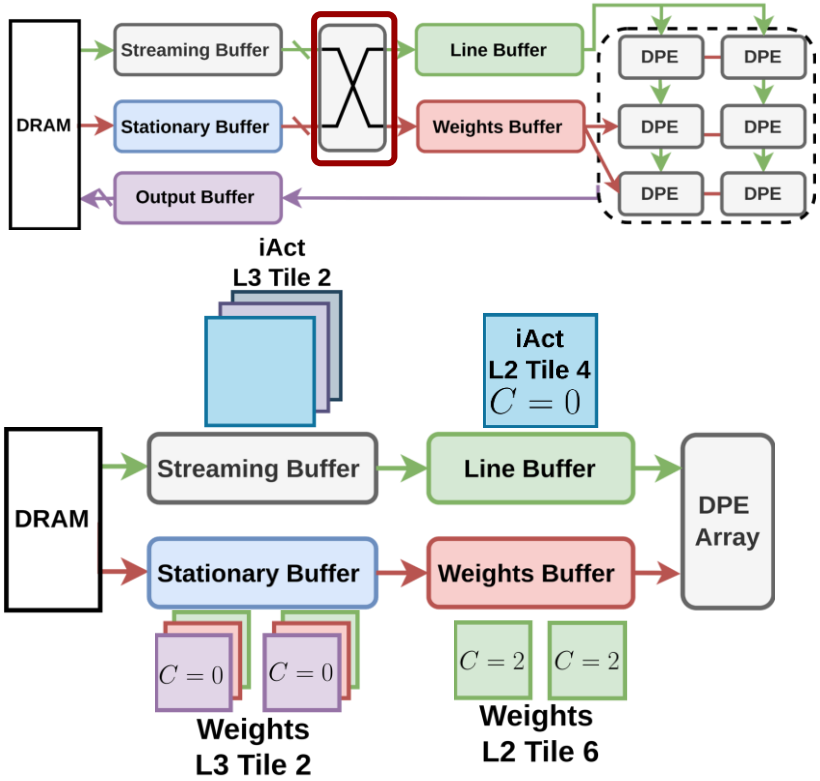
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



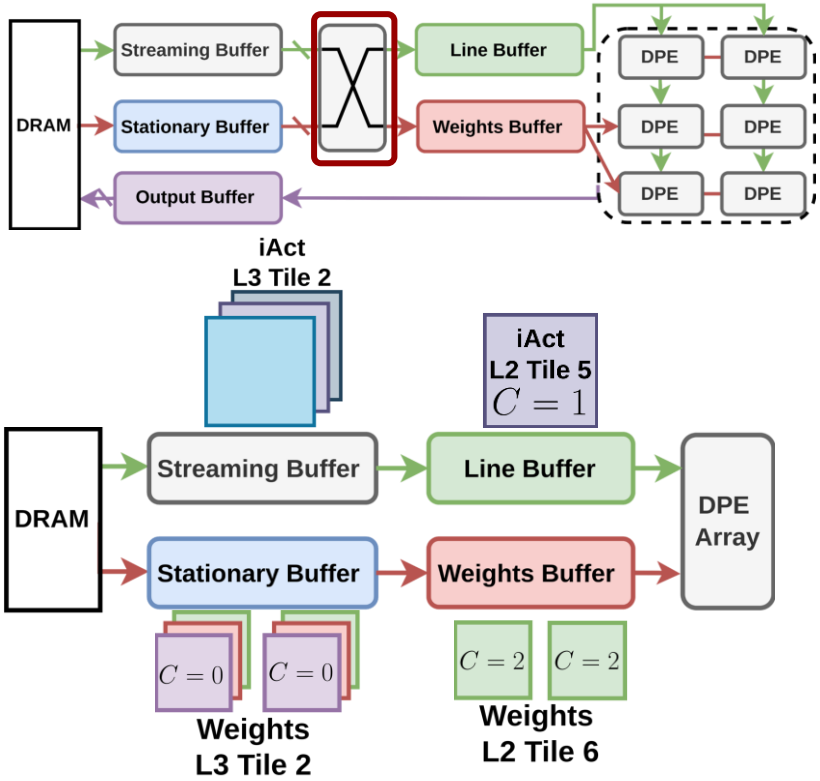
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



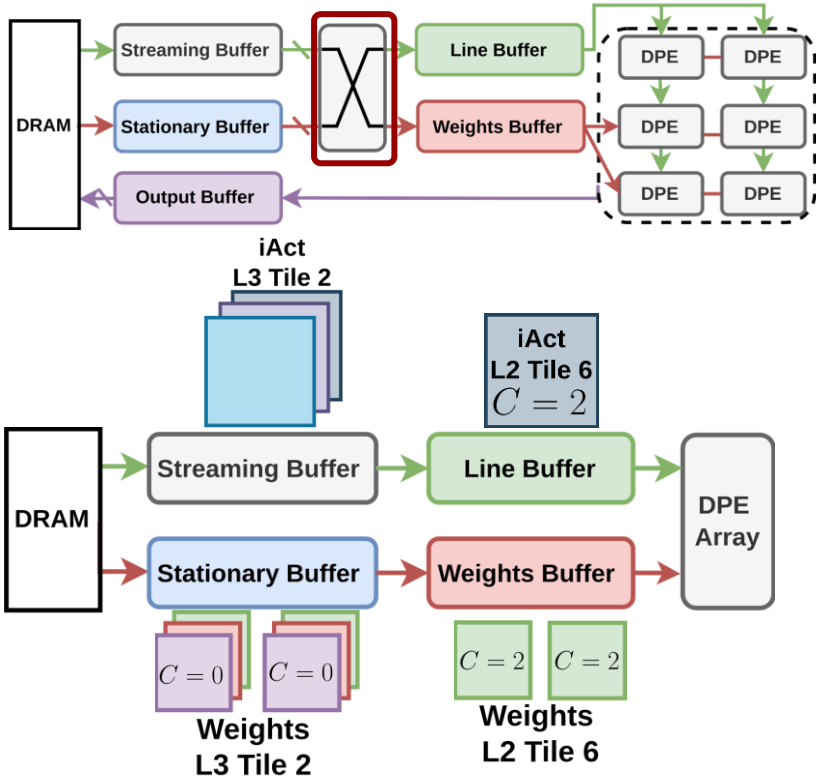
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



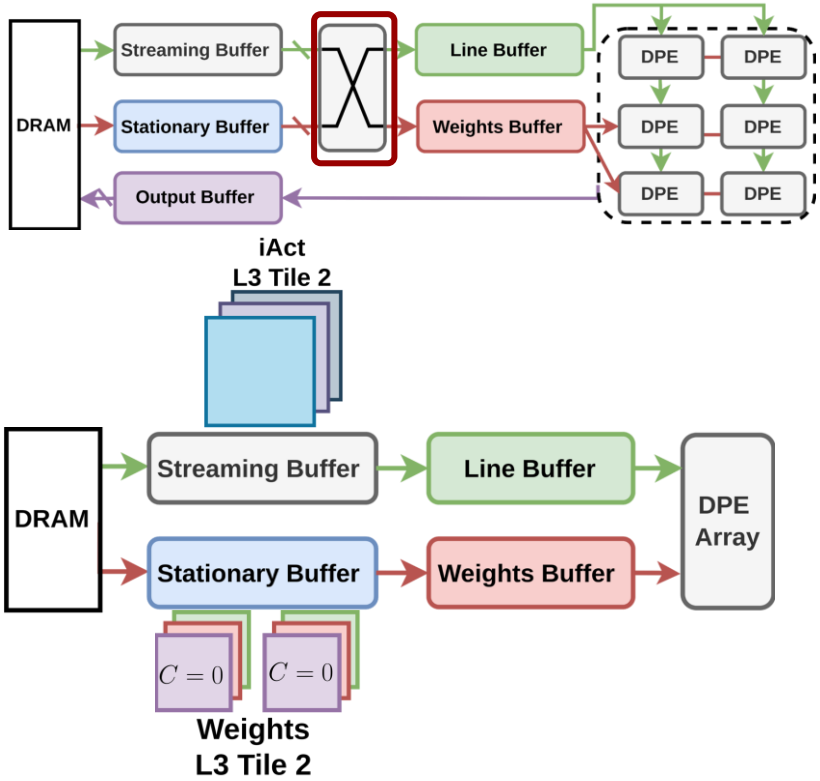
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary



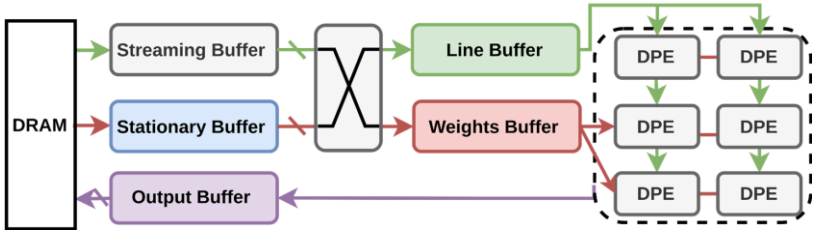
Weight Stay Stationary

MAERI 2.0 Micro-arch: Crossbar Weights Stationary

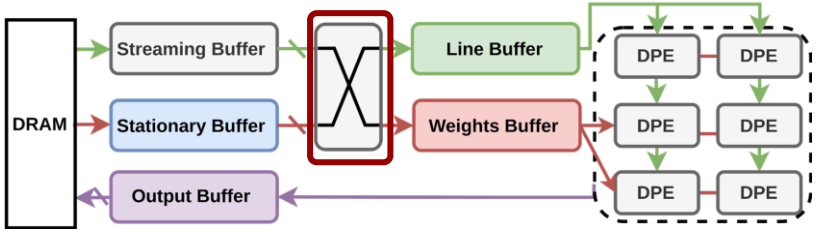


Weight Stay Stationary

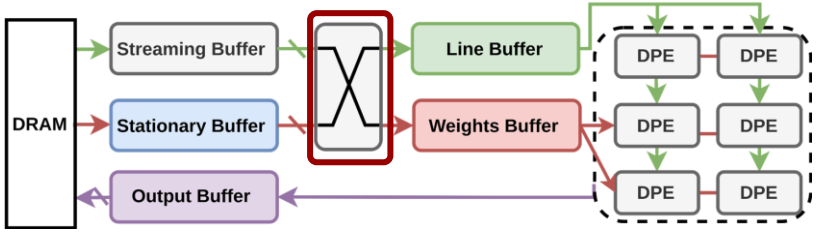
MAERI 2.0 Micro-arch: Crossbar iAct Stationary

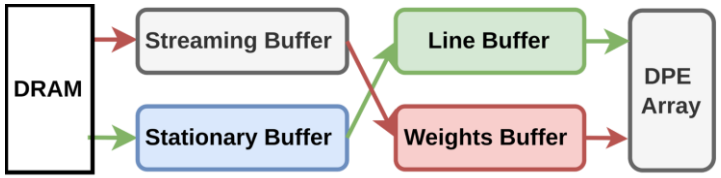


MAERI 2.0 Micro-arch: Crossbar iAct Stationary

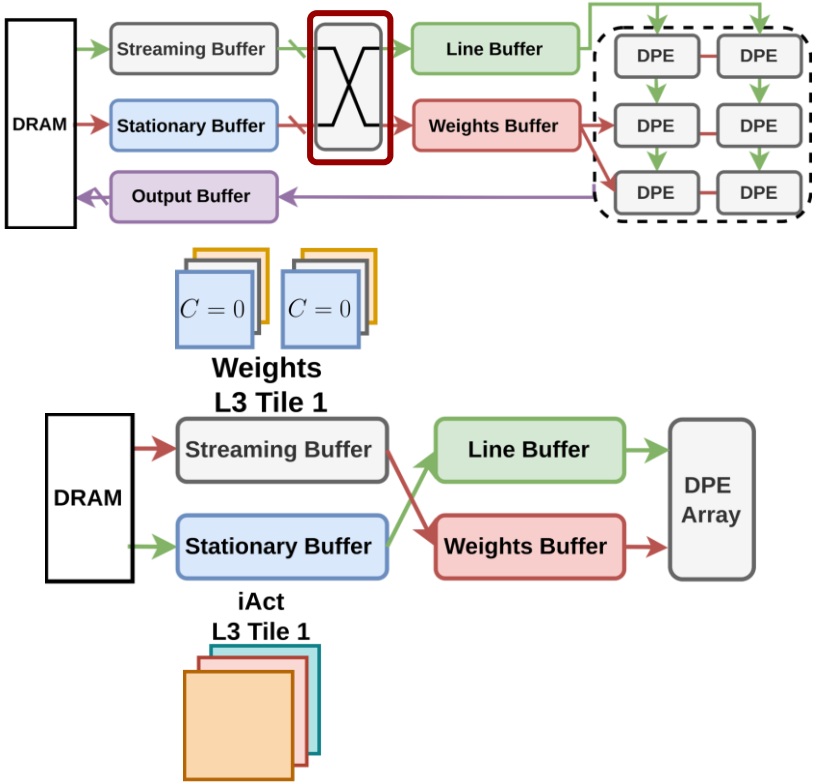


MAERI 2.0 Micro-arch: Crossbar iAct Stationary

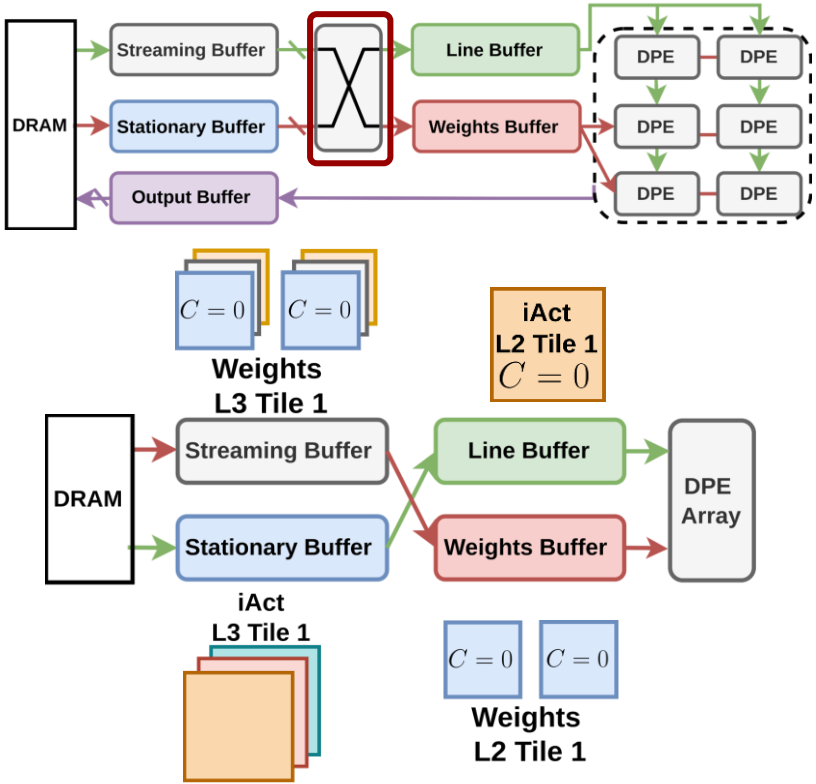




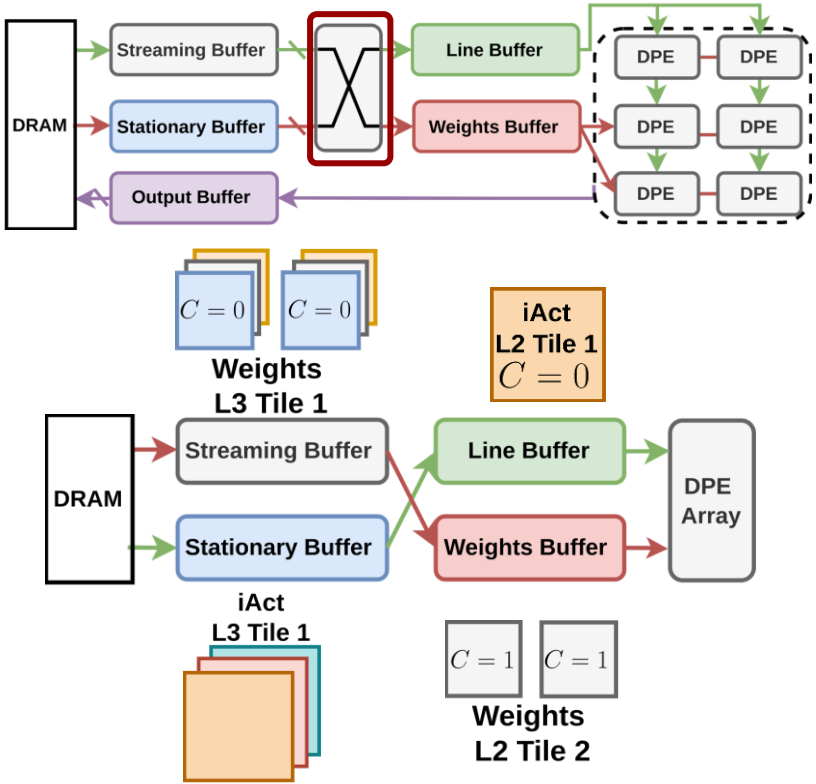
MAERI 2.0 Micro-arch: Crossbar iAct Stationary



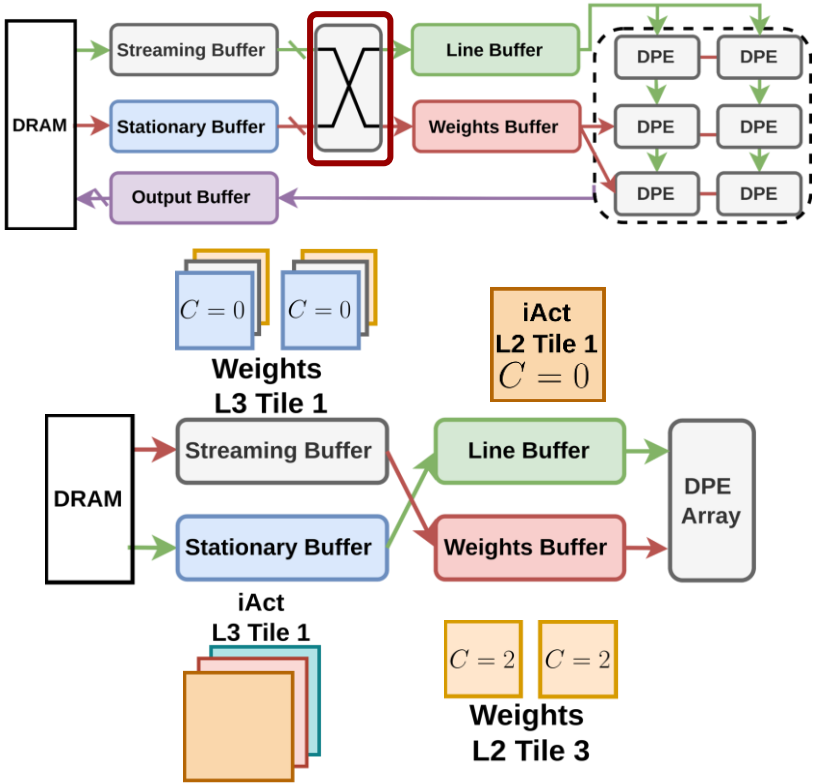
MAERI 2.0 Micro-arch: Crossbar iAct Stationary



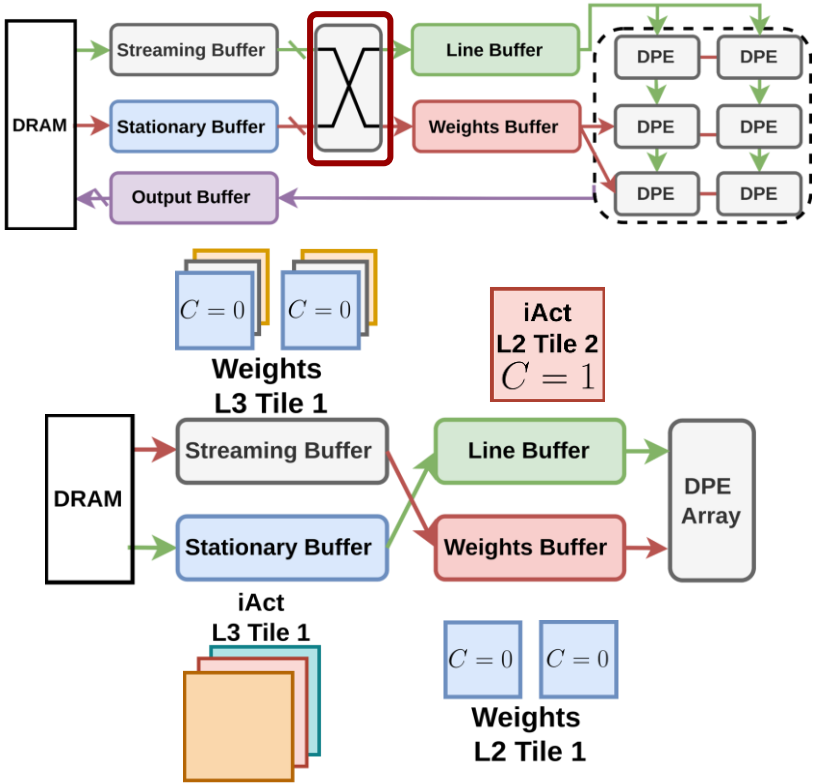
MAERI 2.0 Micro-arch: Crossbar iAct Stationary



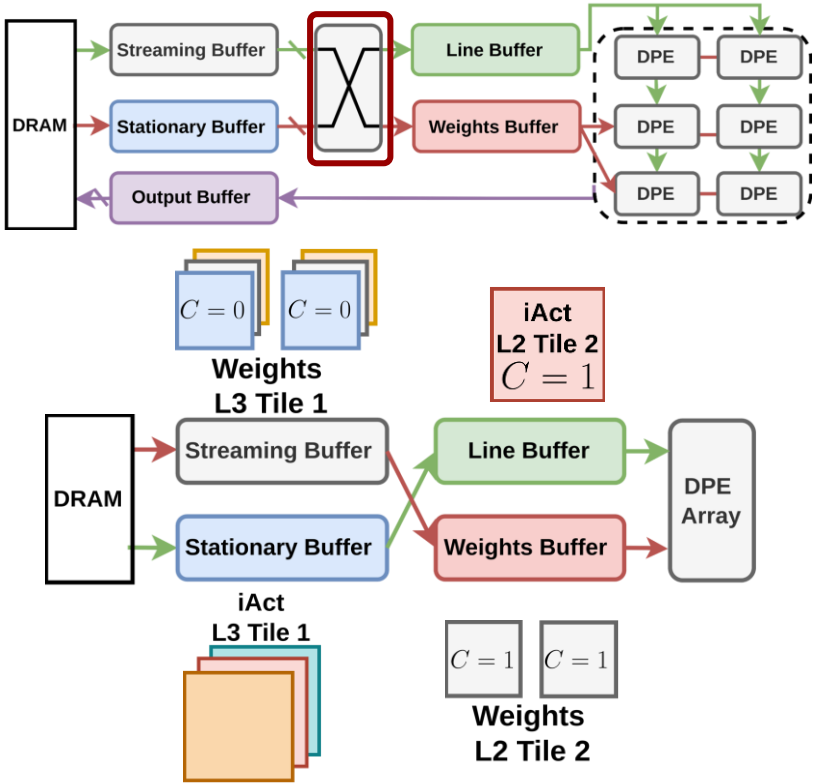
MAERI 2.0 Micro-arch: Crossbar iAct Stationary



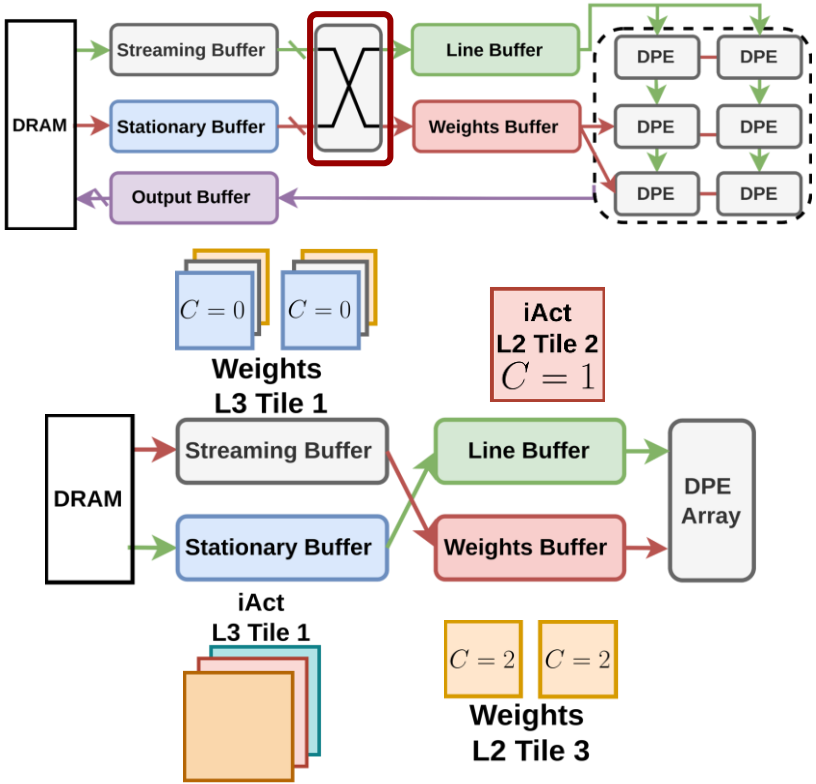
MAERI 2.0 Micro-arch: Crossbar iAct Stationary



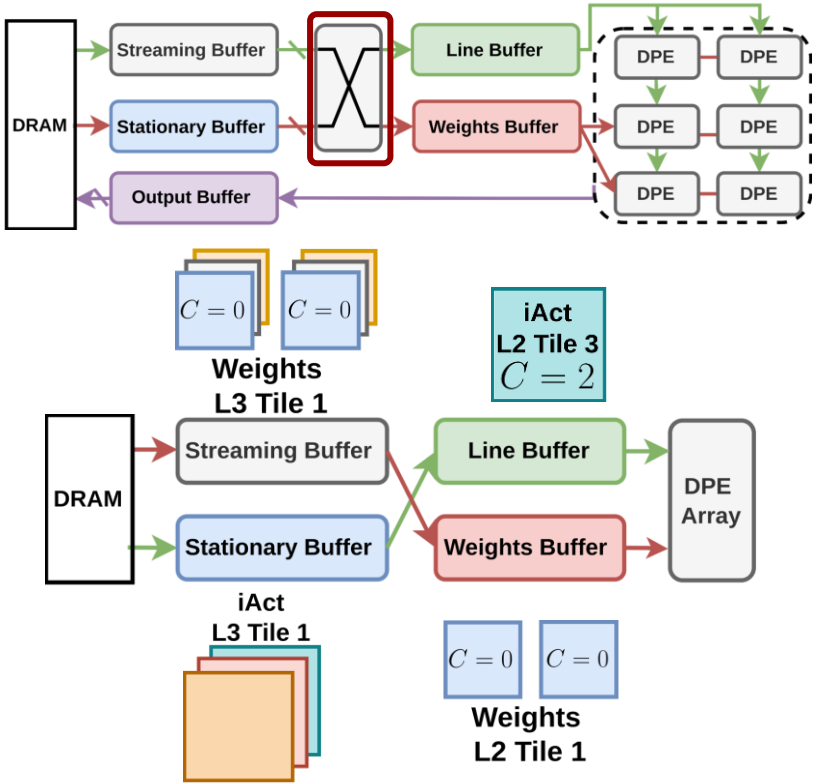
MAERI 2.0 Micro-arch: Crossbar iAct Stationary



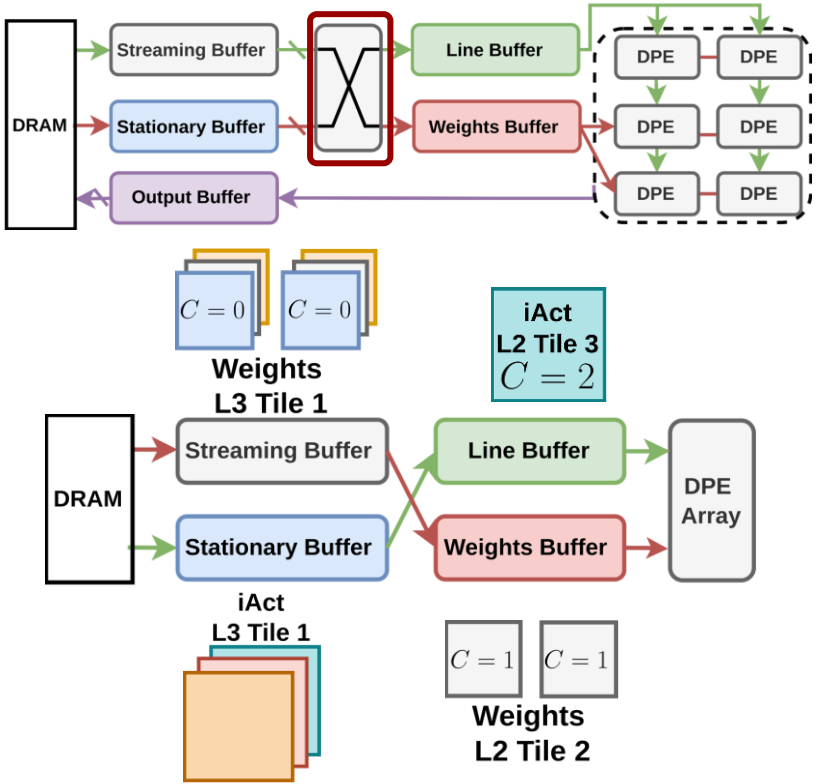
MAERI 2.0 Micro-arch: Crossbar iAct Stationary



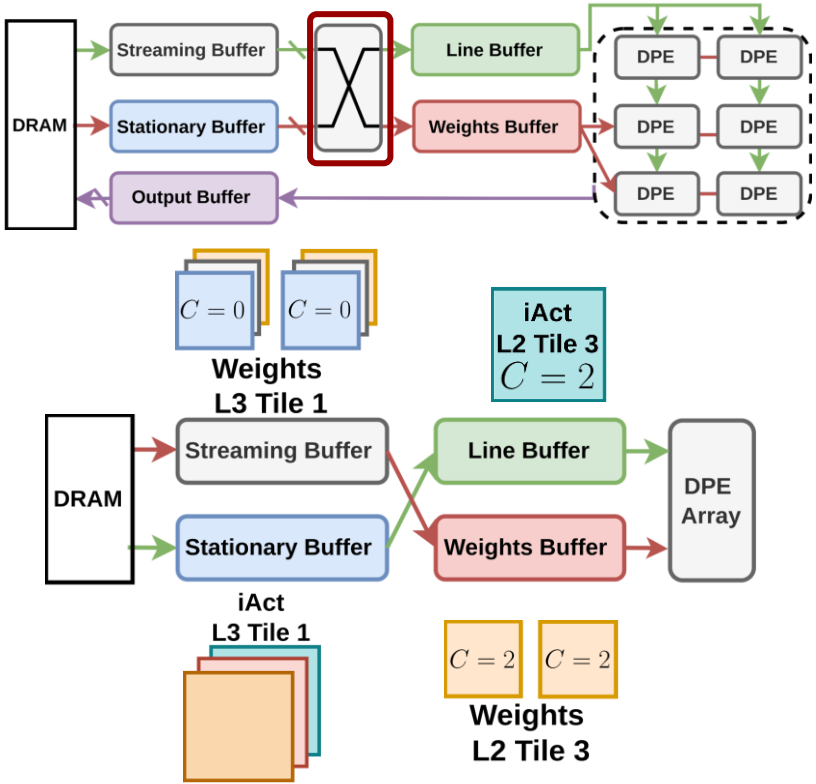
MAERI 2.0 Micro-arch: Crossbar iAct Stationary



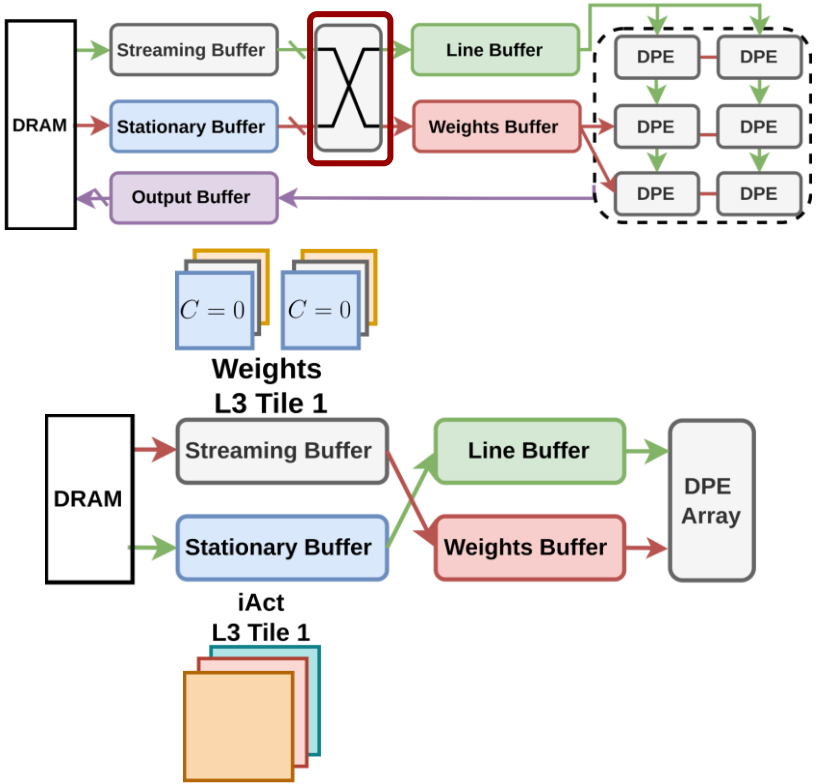
MAERI 2.0 Micro-arch: Crossbar iAct Stationary



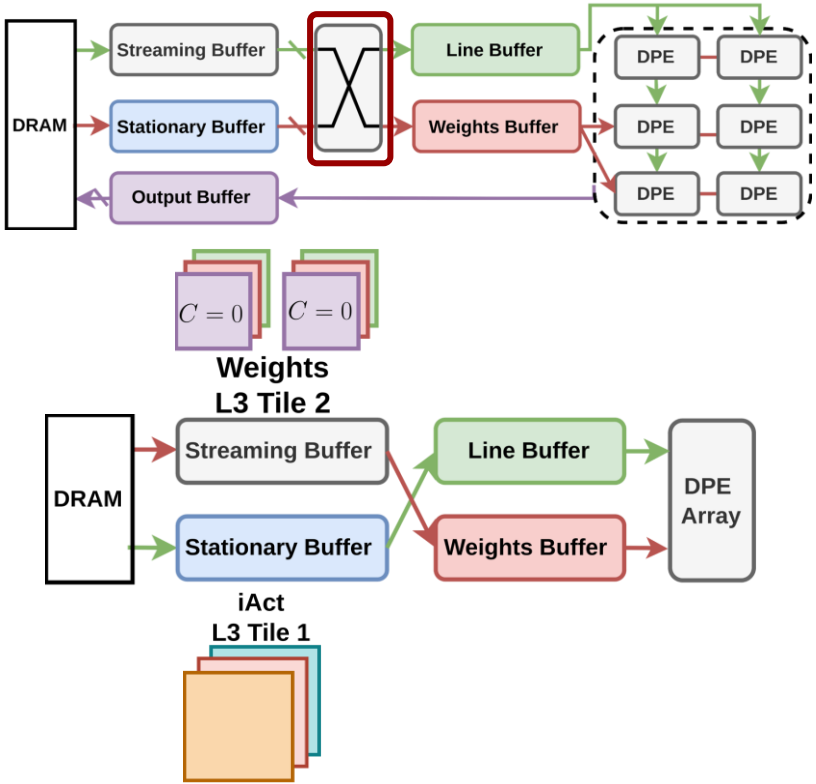
MAERI 2.0 Micro-arch: Crossbar iAct Stationary



MAERI 2.0 Micro-arch: Crossbar iAct Stationary

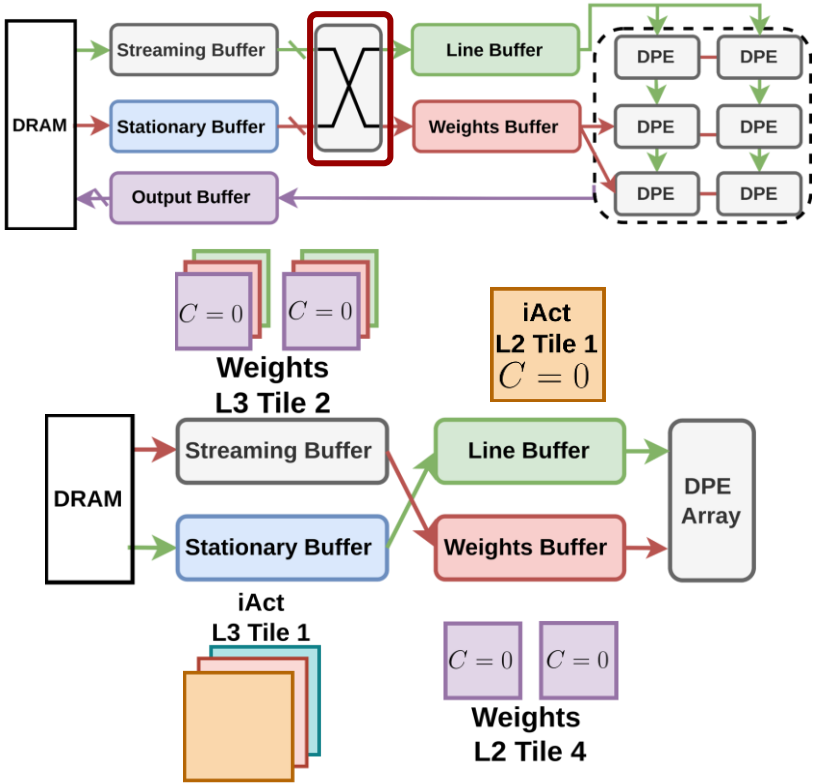


MAERI 2.0 Micro-arch: Crossbar iAct Stationary



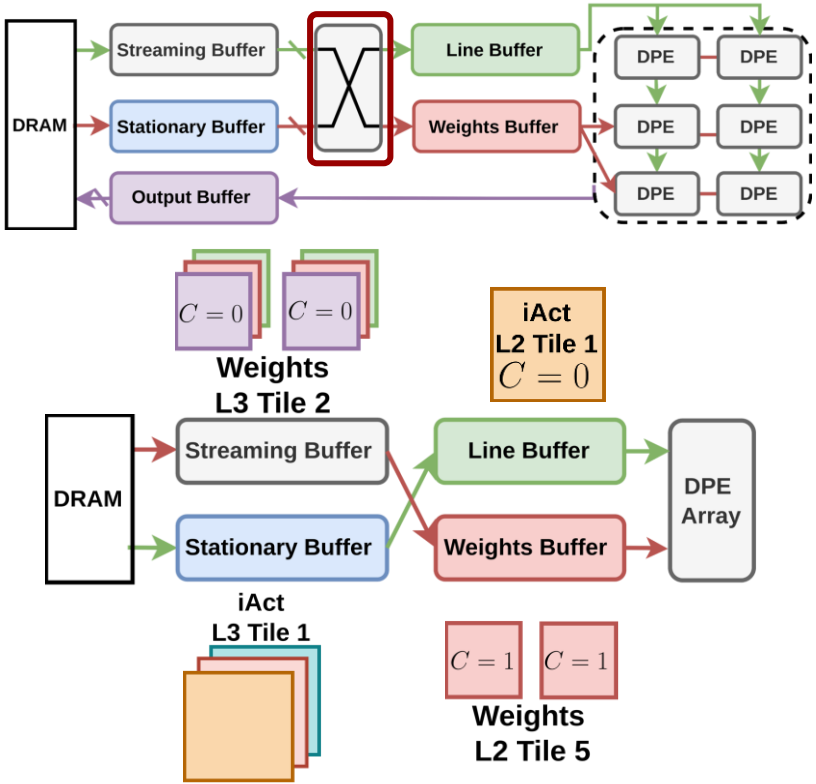
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



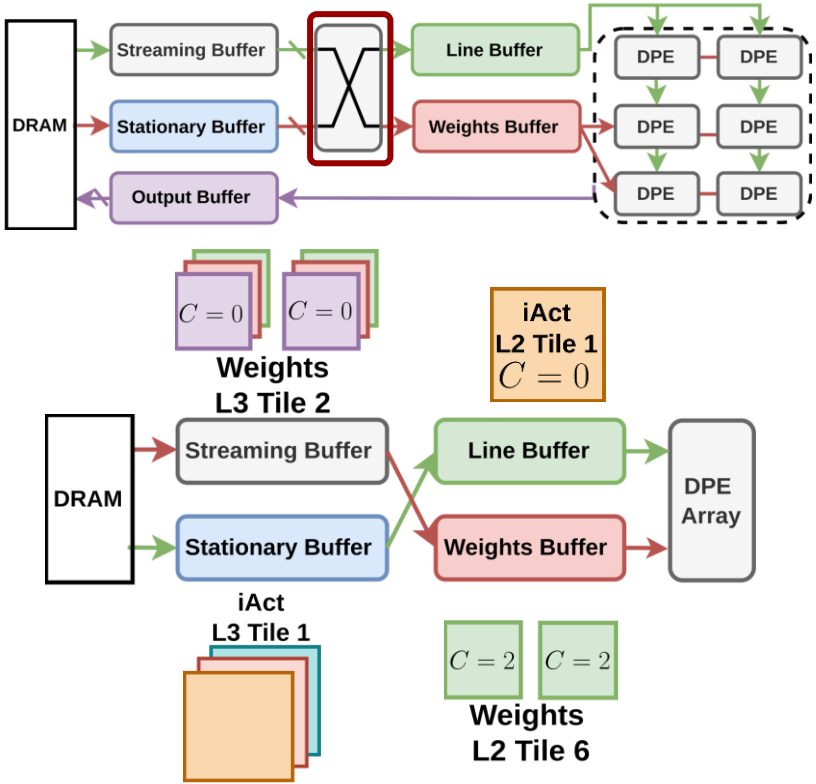
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



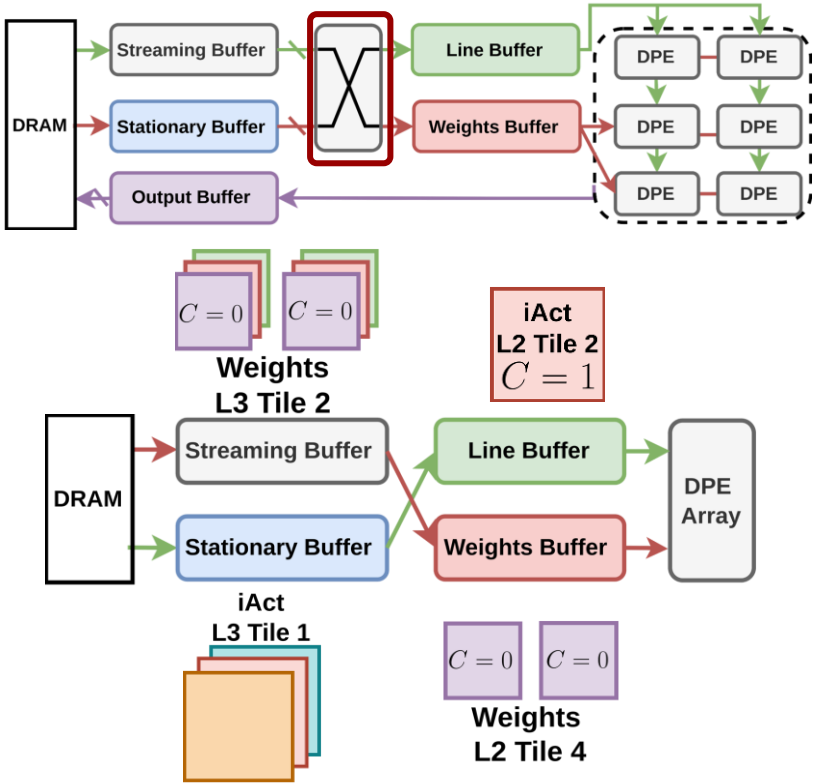
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



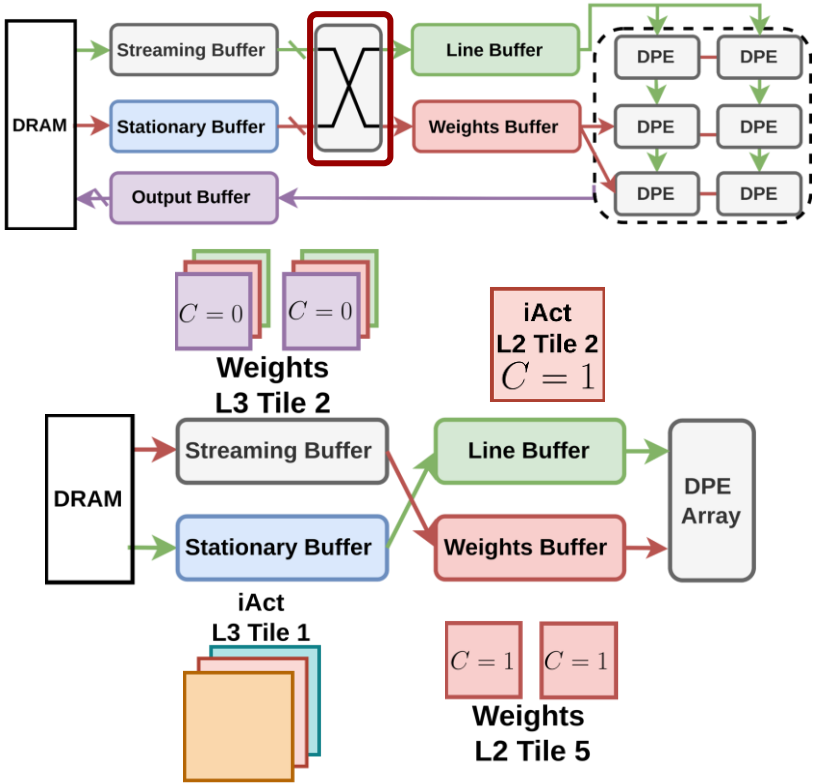
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



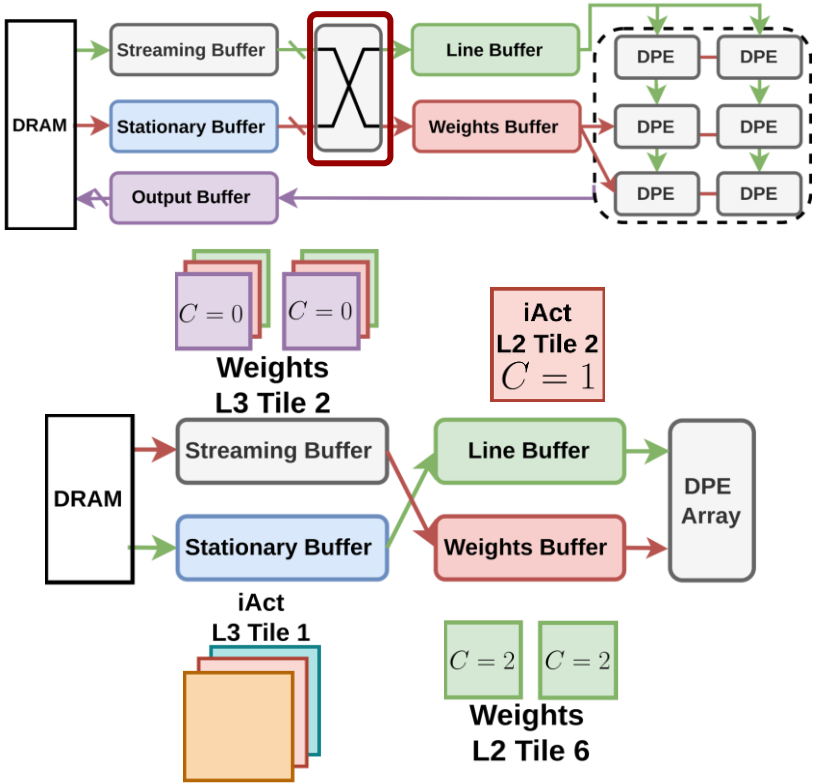
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



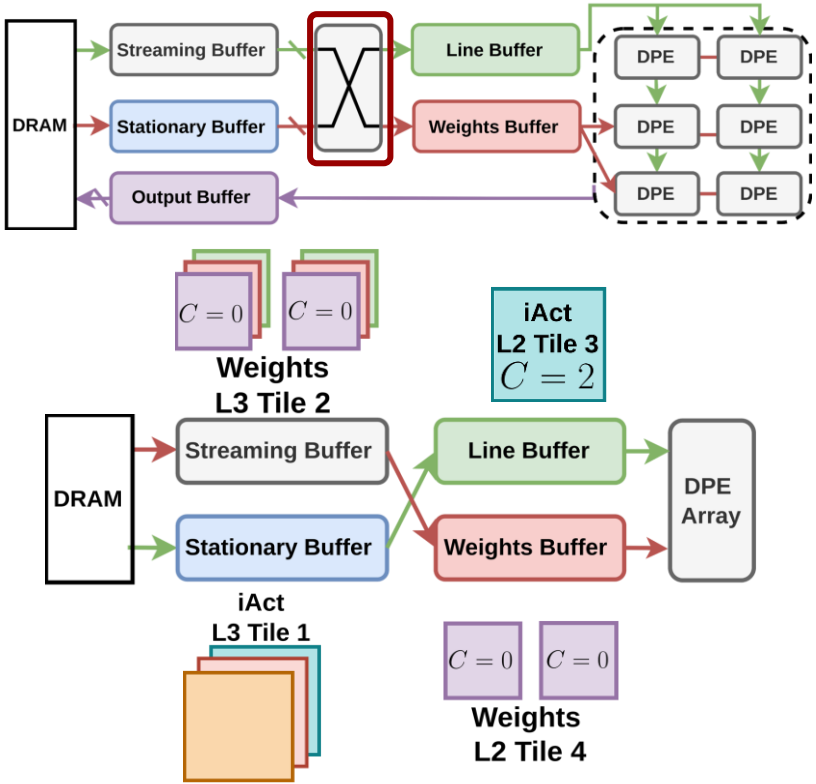
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



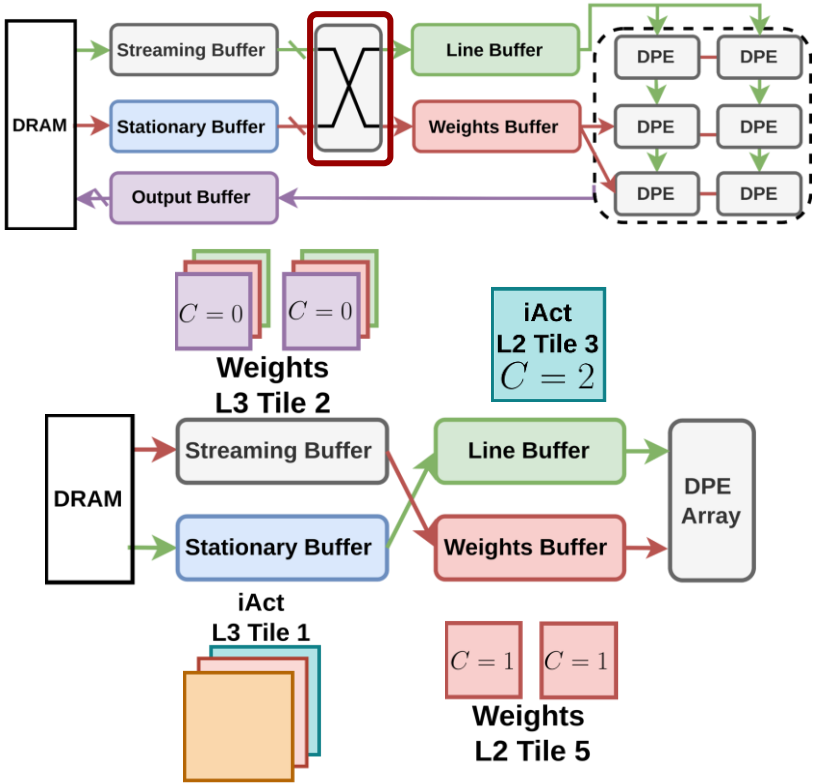
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



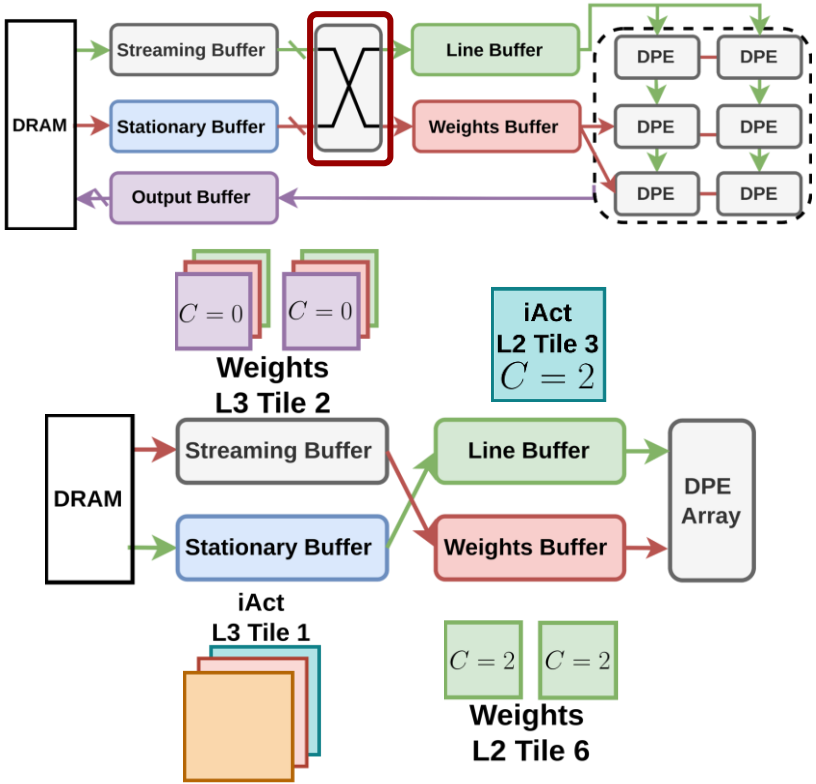
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



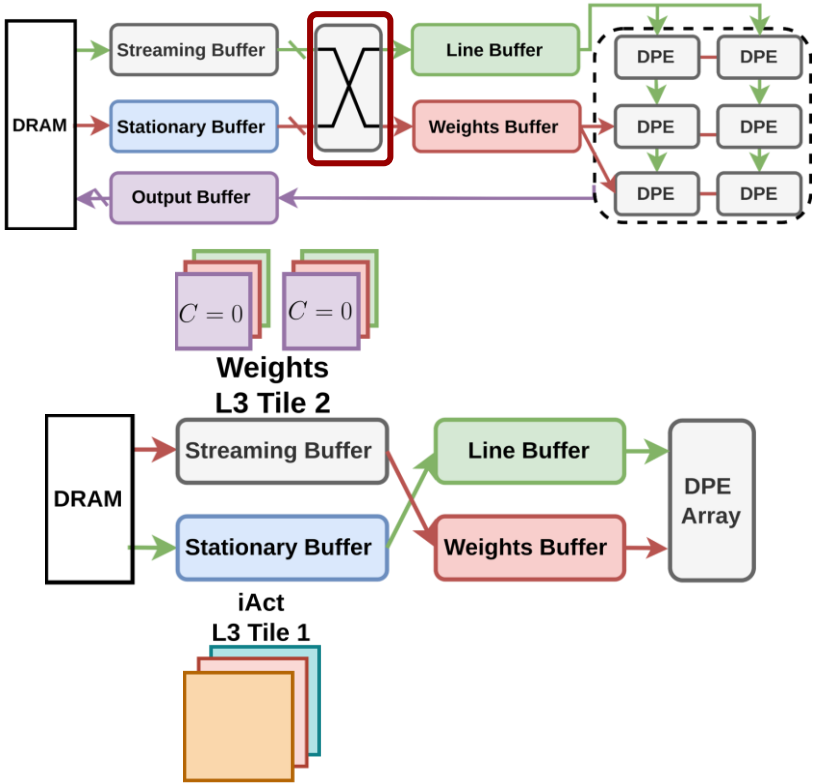
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



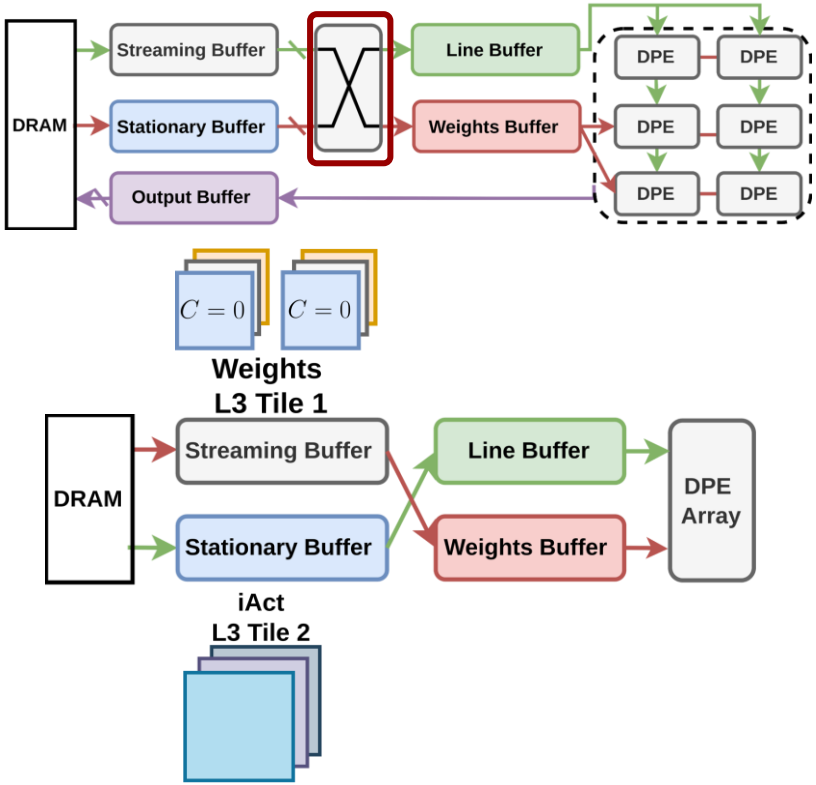
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



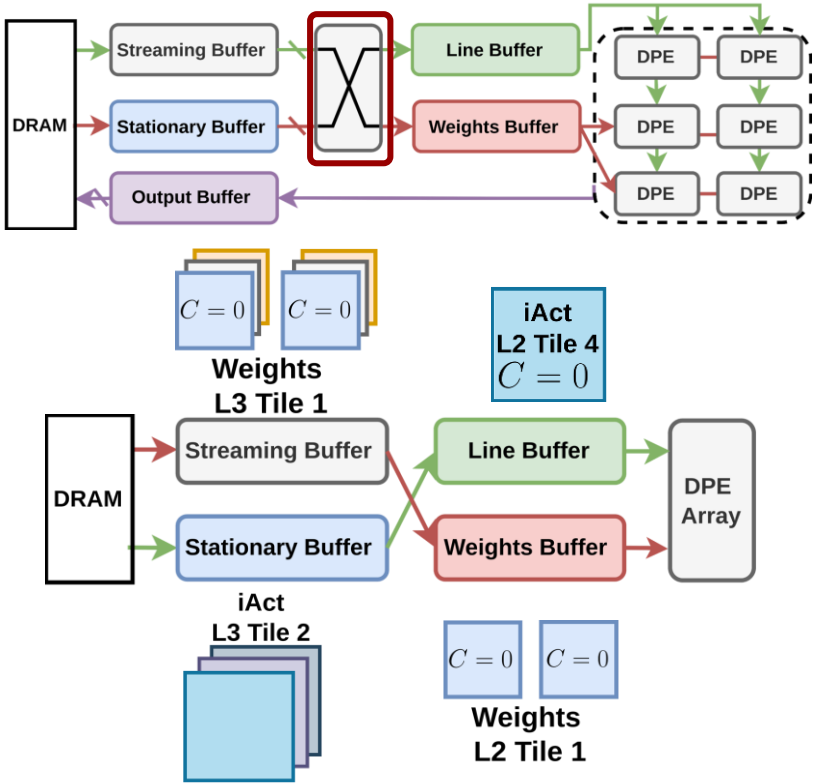
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



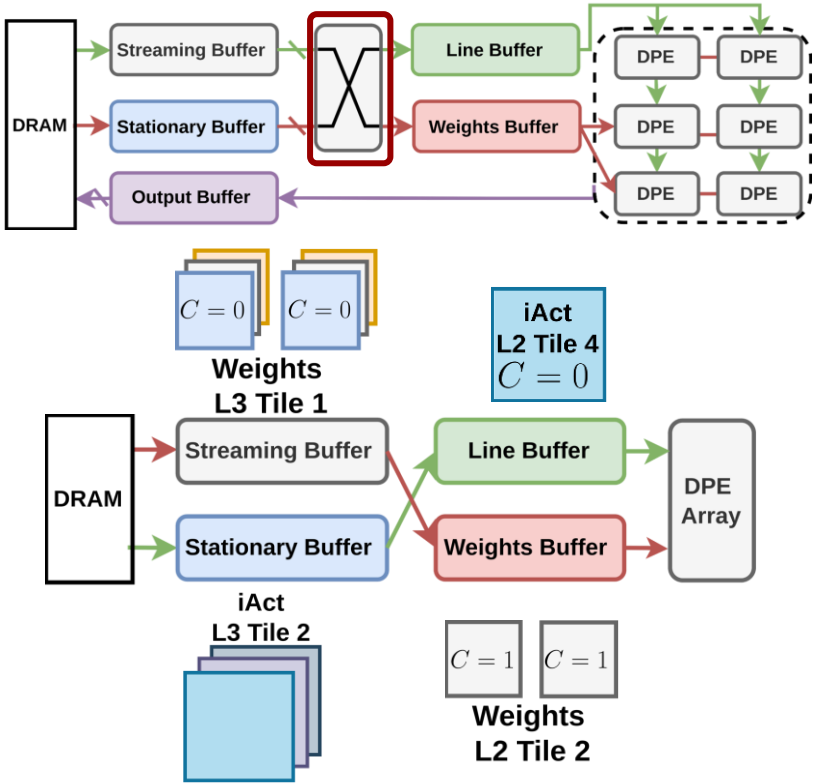
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



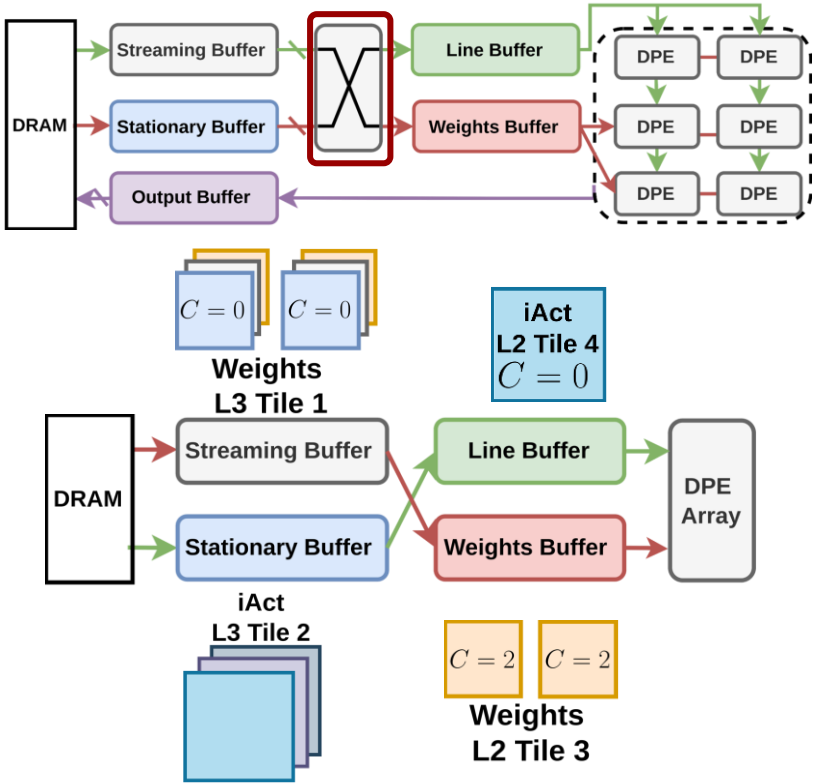
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



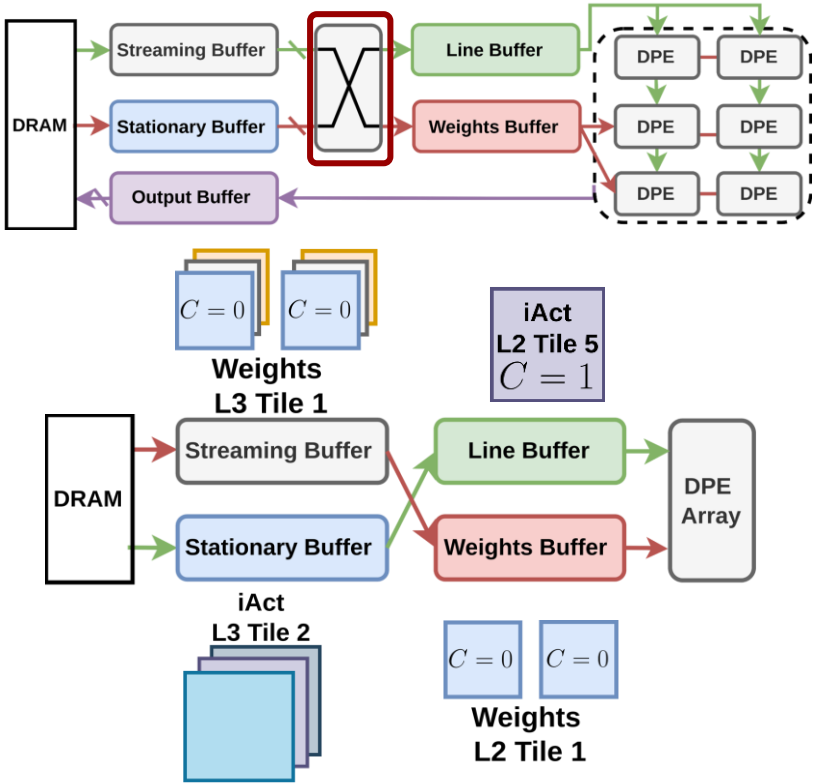
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



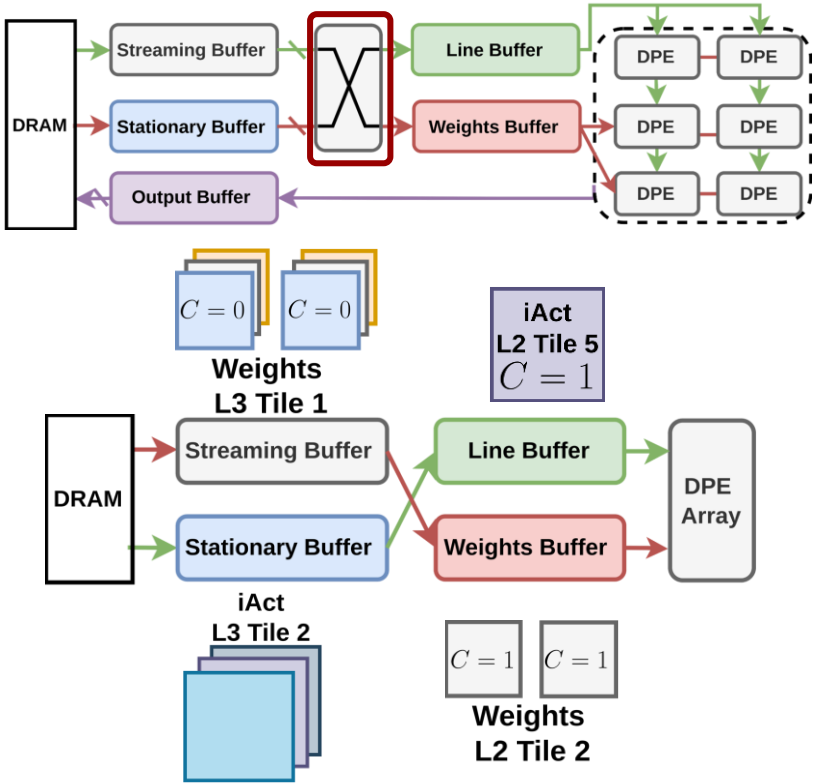
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



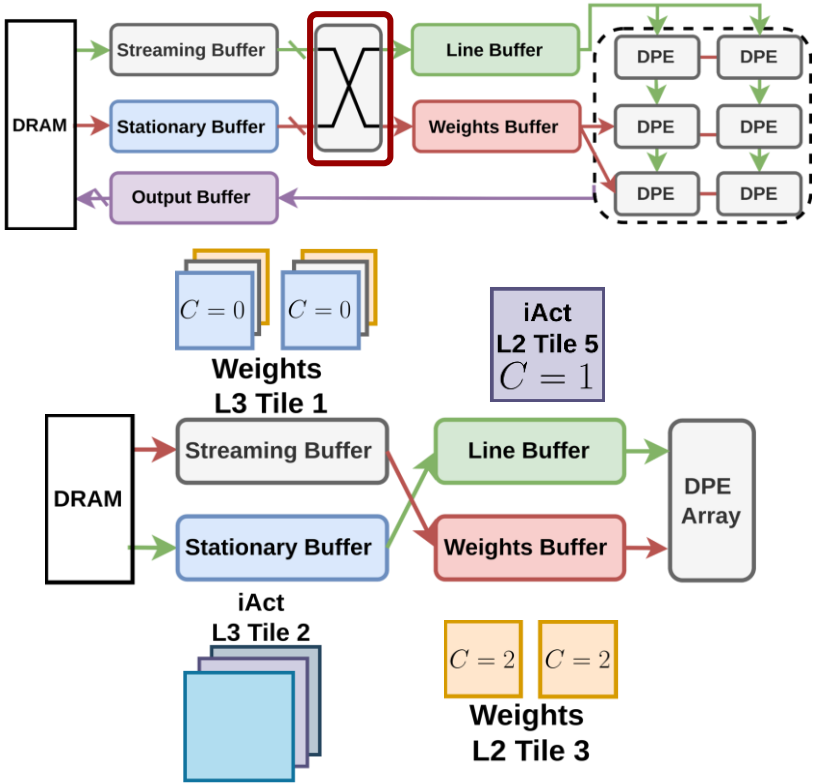
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



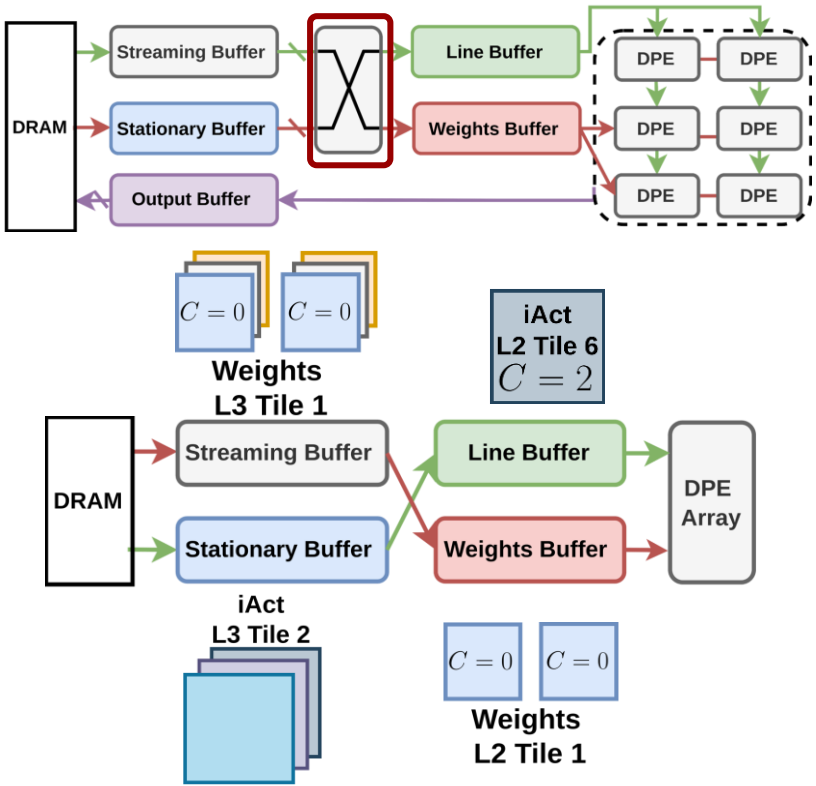
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



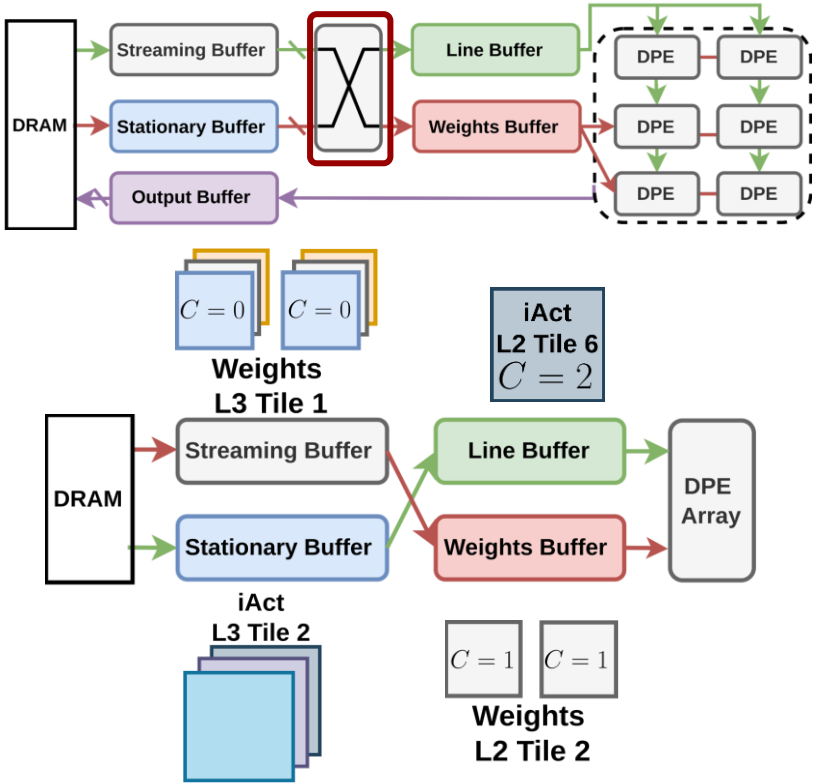
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



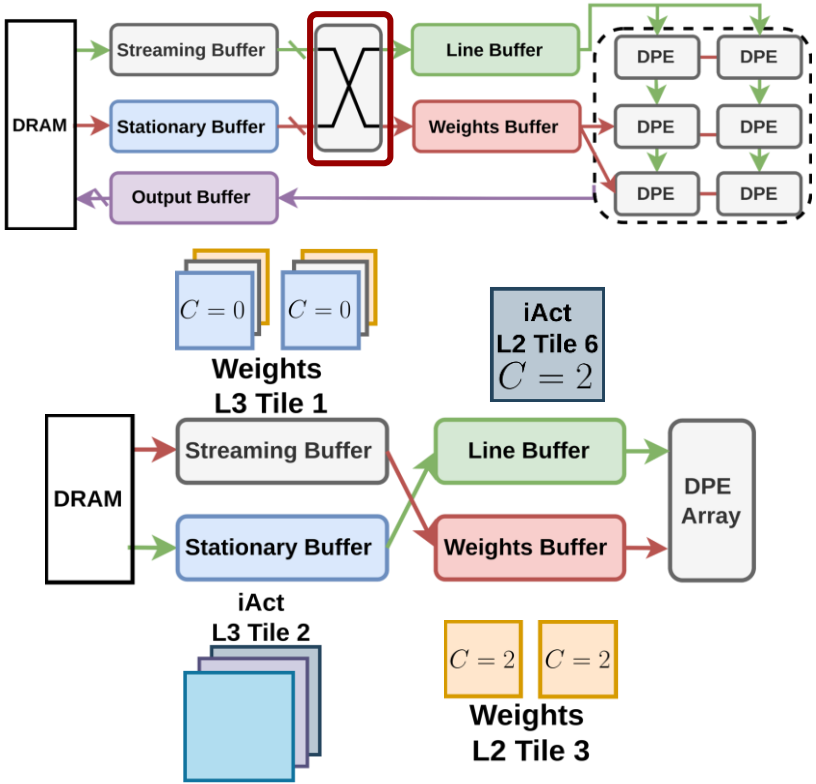
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



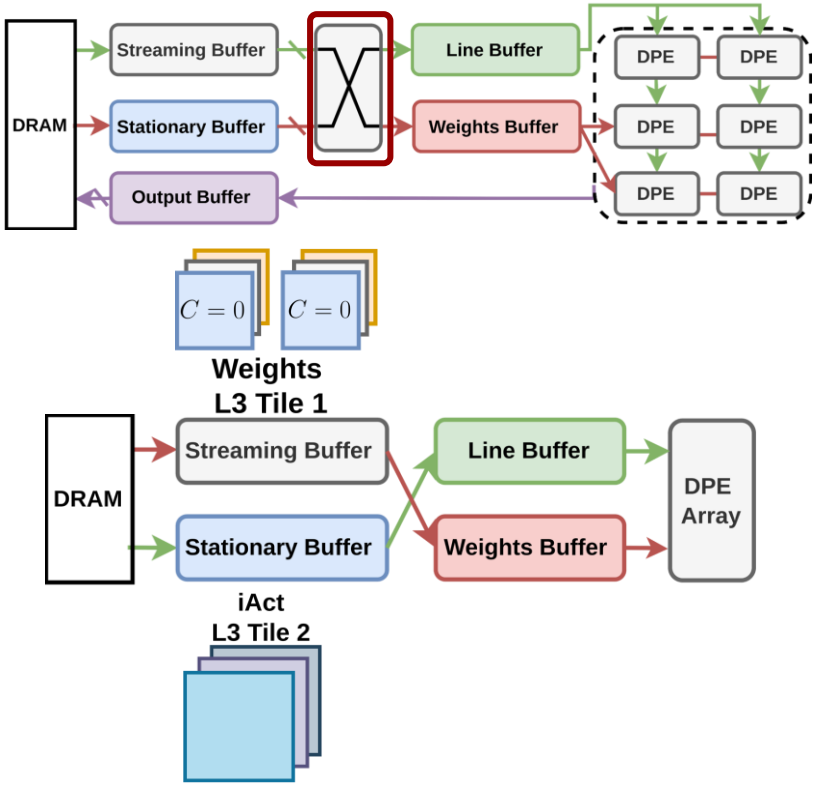
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



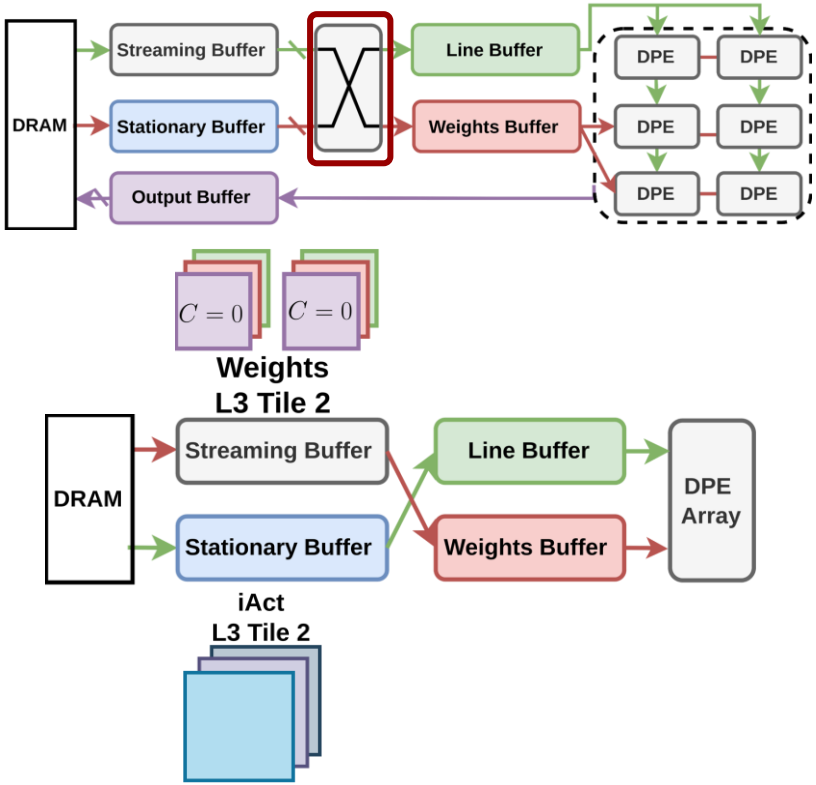
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



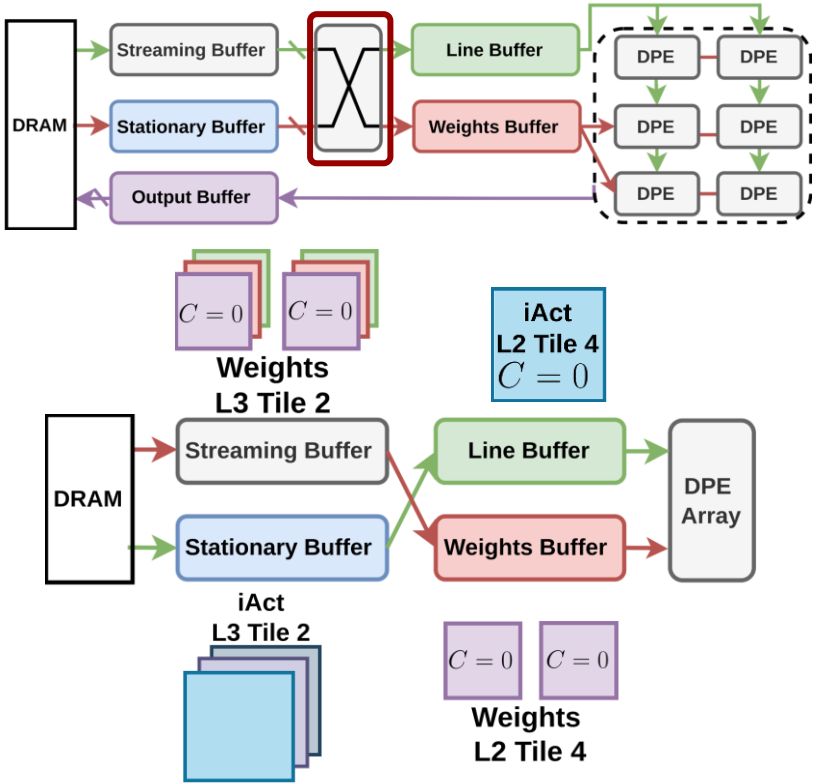
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



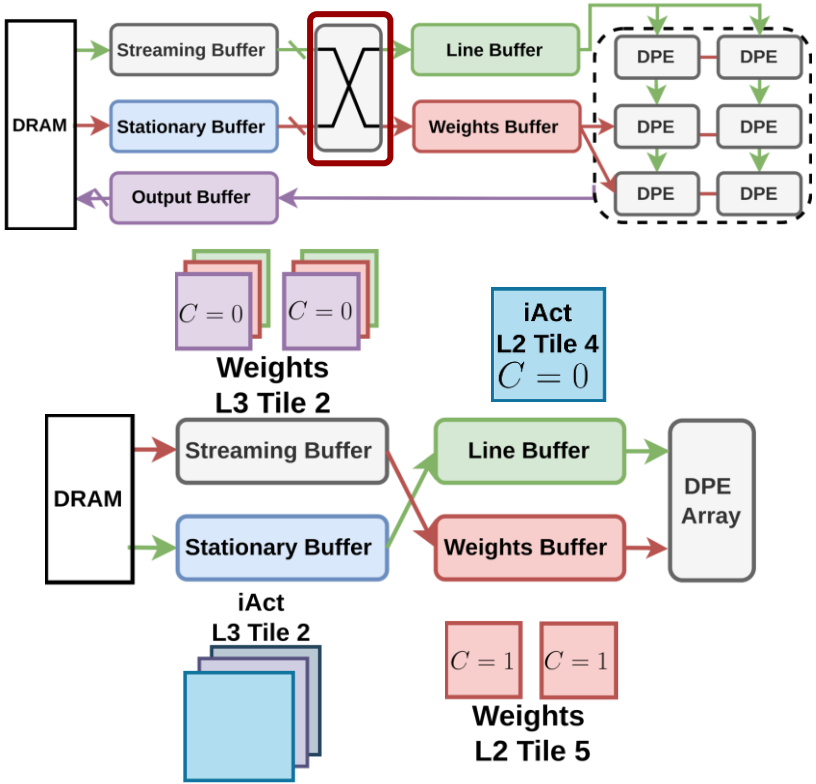
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



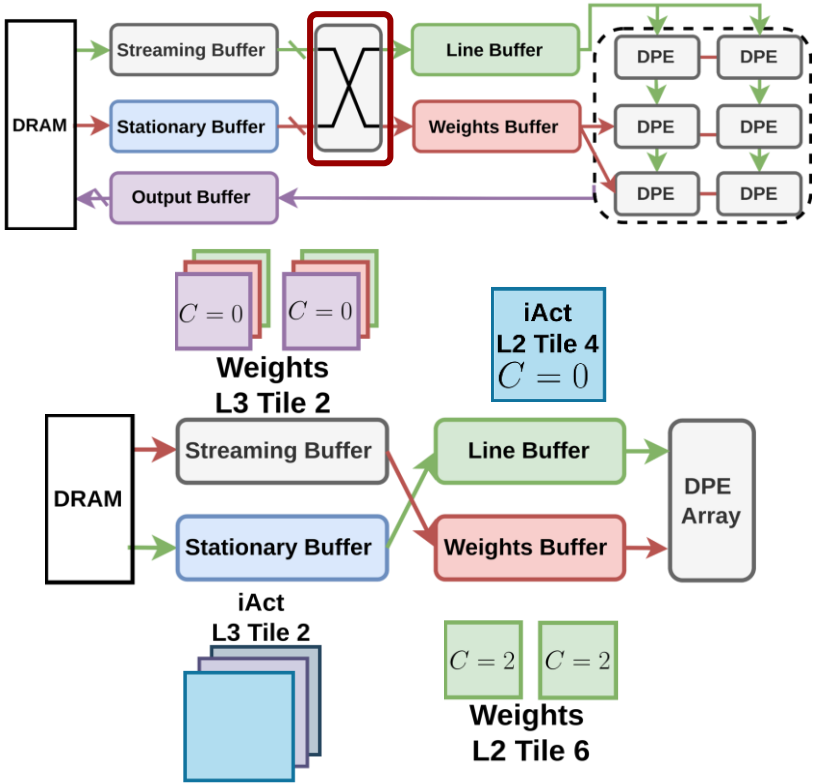
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



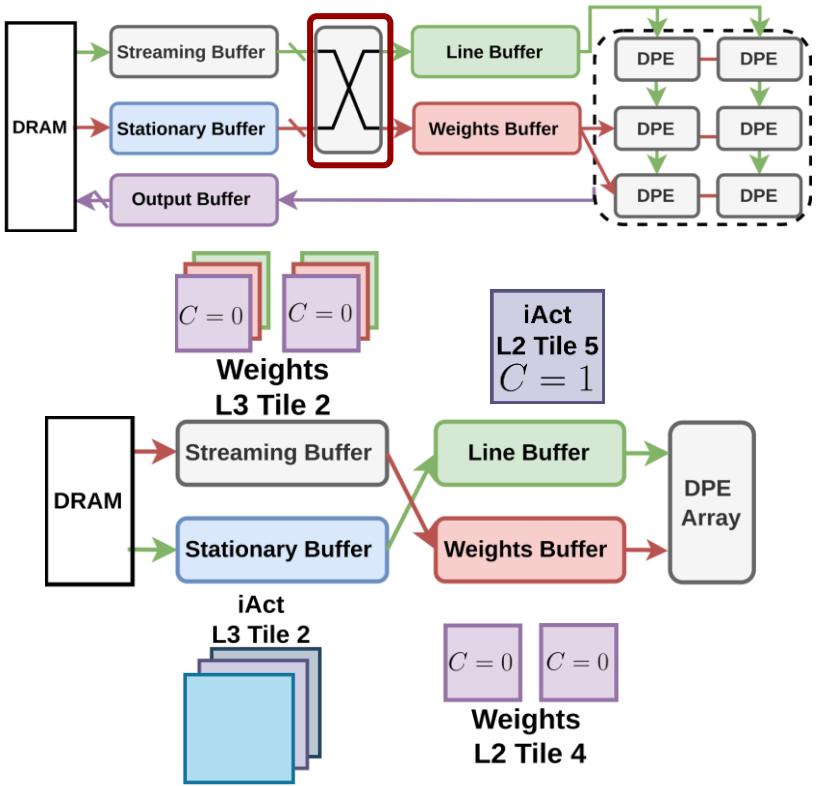
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



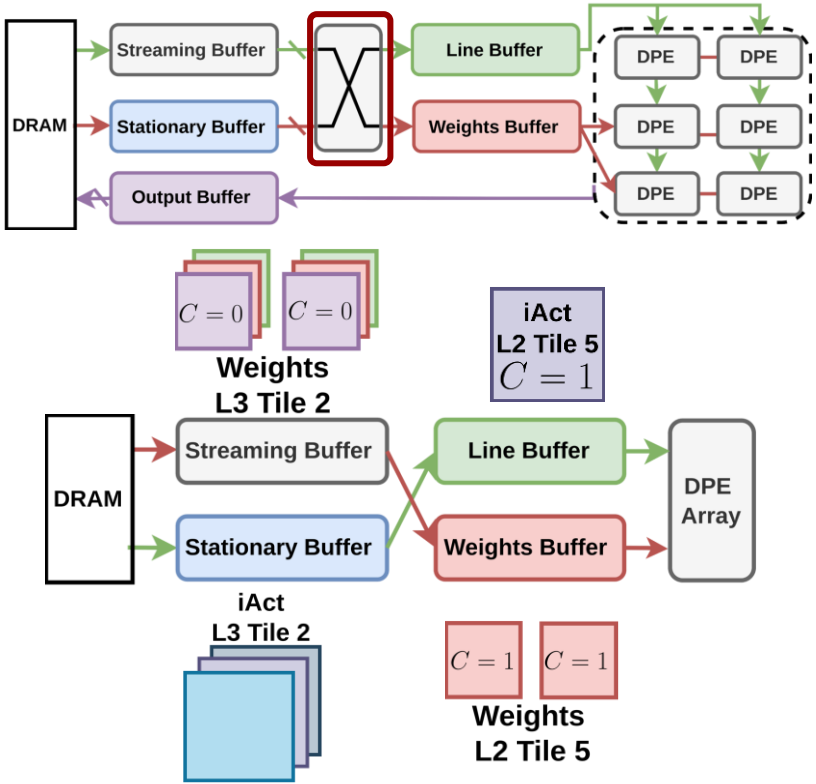
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



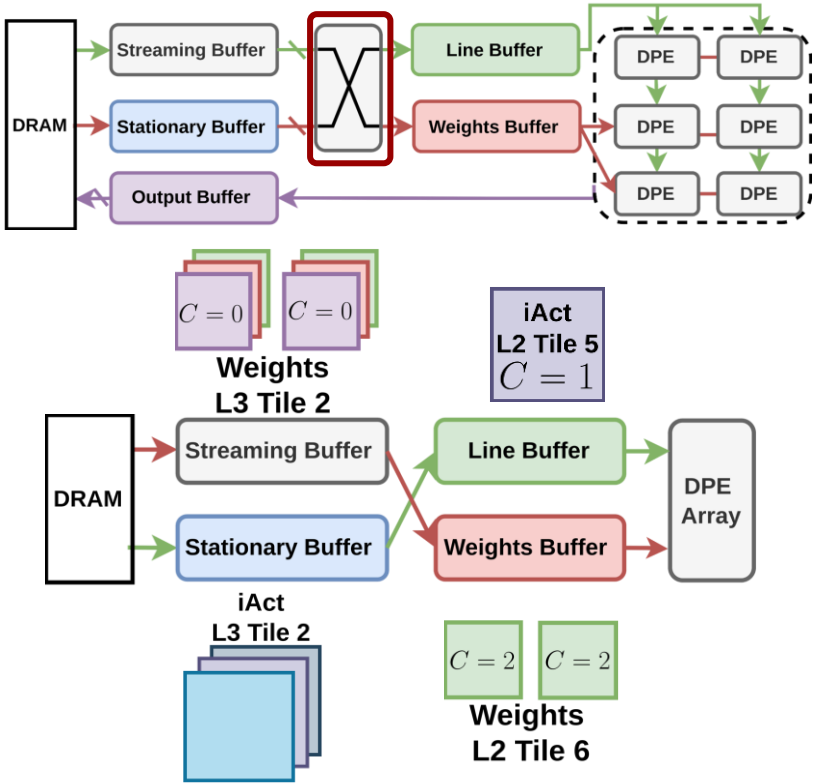
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



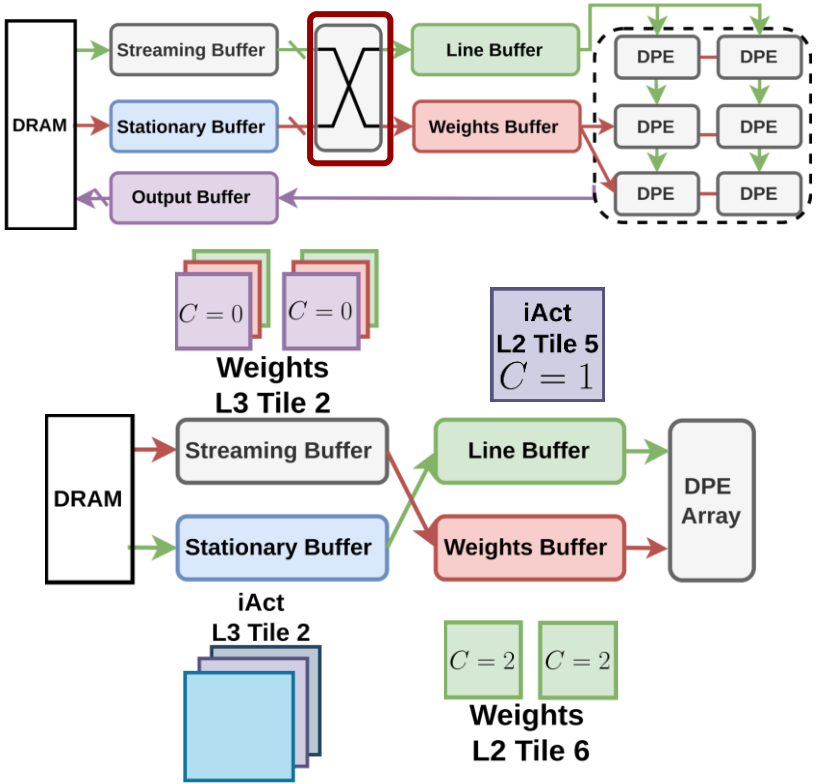
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



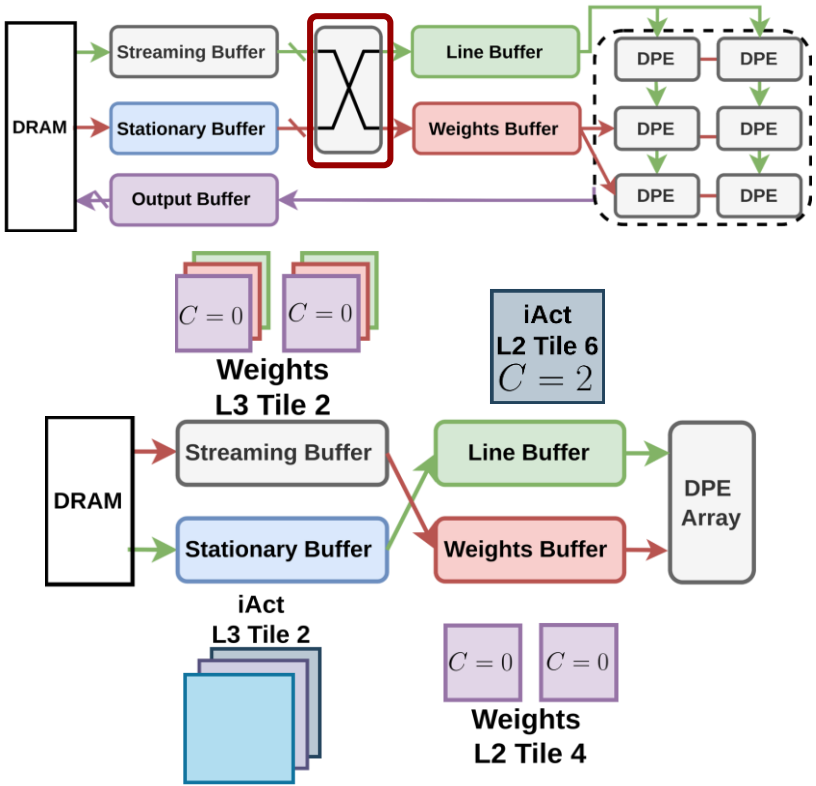
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



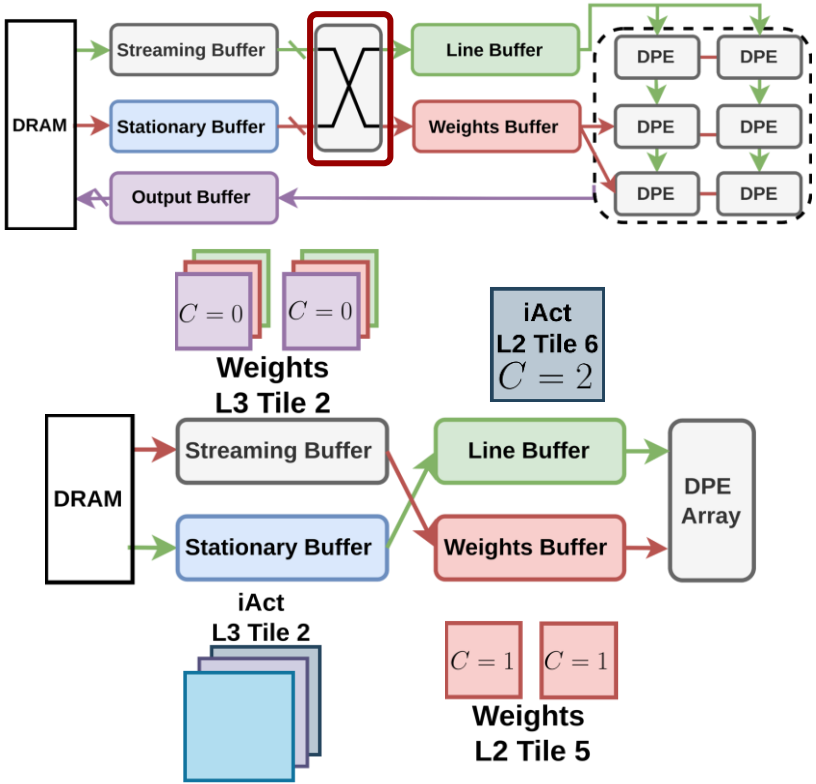
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



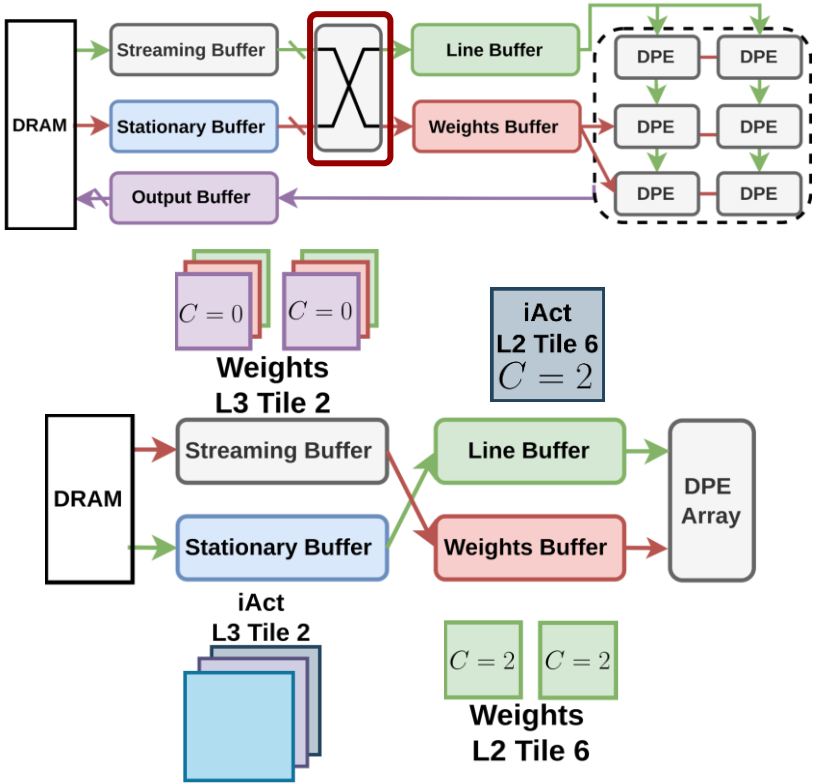
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



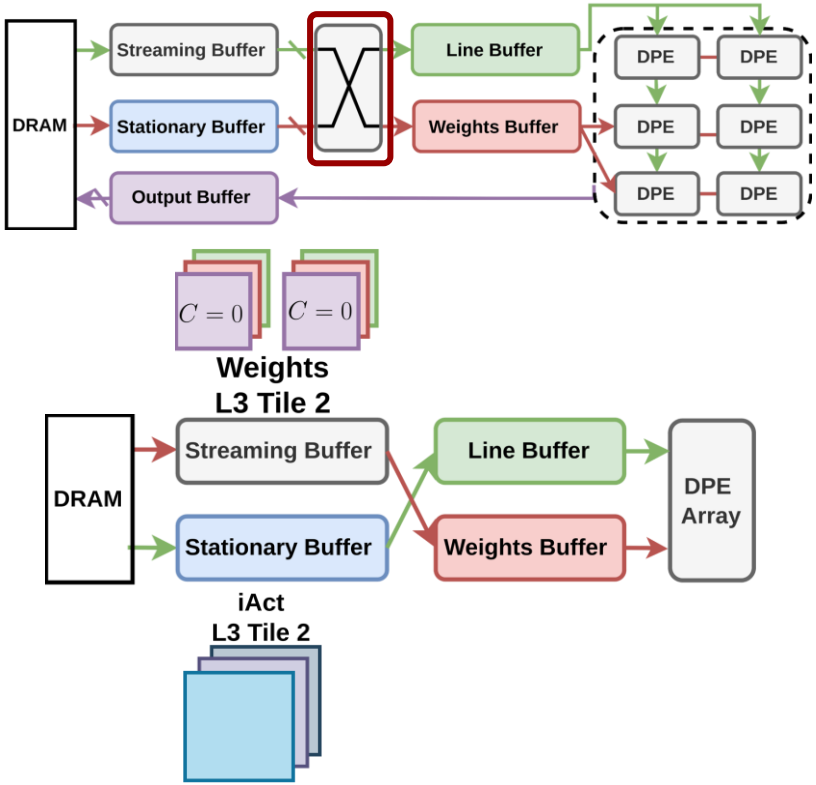
iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



iAct Stay Stationary

MAERI 2.0 Micro-arch: Crossbar iAct Stationary



iAct Stay Stationary

Outlines

- Supported Neural Network Model
- Quantization Flow
- Memory Layout
- Heterogeneous Scheduling
- MAERI 2.0 Microarchitecture
- **DEMO**
  - **Entire Flow Demonstration**
  - Walk Through Example of deploying ResNet50

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

Goal of DEMO 1: Demonstrate the entire MAERI 2.0 Flow

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

Goal of DEMO 1: Demonstrate the entire MAERI 2.0 Flow

- **Workload: ResNet 50**

Layer Type	Convolution 2d	BatchNorm 2d	ReLU	Skip Add	Average Pooling	Fully Connected
Number of Layers	53	53	49	16	1	1

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

Goal of DEMO 1: Demonstrate the entire MAERI 2.0 Flow

- **Workload: ResNet 50**

Layer Type	Convolution 2d	BatchNorm 2d	ReLU	Skip Add	Average Pooling	Fully Connected
Number of Layers	53	53	49	16	1	1

- **Platform: zcu 104** 

Name	OS	python	PyTorch	Quantization Scheme	Dataset	CPU
Version	pynqlinux v2.6 (18.04)	3.6.5	1.8.1	qnnpack	imagenet 1k	Dual-core Arm Cortex-R5F

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

Goal of DEMO 1: Demonstrate the entire MAERI 2.0 Flow

- **Workload: ResNet 50**

Layer Type	Convolution 2d	BatchNorm 2d	ReLU	Skip Add	Average Pooling	Fully Connected
Number of Layers	53	53	49	16	1	1

- **Platform: zcu 104** 

Name	OS	python	PyTorch	Quantization Scheme	Dataset	CPU
Version	pynqlinux v2.6 (18.04)	3.6.5	1.8.1	qnnpack	imagenet 1k	Dual-core Arm Cortex-R5F

- **DEMO 1 (Pre-recorded)**
  - Setup Environment
  - Model Quantization
  - Data Layout Reorder (PyTorch Default order)
  - Custom Inference

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

Goal of DEMO 1: Demonstrate the entire MAERI 2.0 Flow

- **Workload: ResNet 50**

Layer Type	Convolution 2d	BatchNorm 2d	ReLU	Skip Add	Average Pooling	Fully Connected
Number of Layers	53	53	49	16	1	1

- **Platform: zcu 104** 

Name	OS	python	PyTorch	Quantization Scheme	Dataset	CPU
Version	pynqlinux v2.6 (18.04)	3.6.5	1.8.1	qnnpack	imagenet 1k	Dual-core Arm Cortex-R5F

- **DEMO 1 (Pre-recorded)**
  - Setup Environment
  - Model Quantization
  - Data Layout Reorder (PyTorch Default order)
  - Custom Inference

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

Goal of DEMO 1: Demonstrate the entire MAERI 2.0 Flow

- **Workload: ResNet 50**

Layer Type	Convolution 2d	BatchNorm 2d	ReLU	Skip Add	Average Pooling	Fully Connected
Number of Layers	53	53	49	16	1	1

- **Platform: zcu 104** 

Name	OS	python	PyTorch	Quantization Scheme	Dataset	CPU
Version	pynqlinux v2.6 (18.04)	3.6.5	1.8.1	qnnpack	imagenet 1k	Dual-core Arm Cortex-R5F

- **DEMO 1 (Pre-recorded)**
  - Setup Environment
  - Model Quantization
  - Data Layout Reorder (PyTorch Default order)
  - Custom Inference

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

Goal of DEMO 1: Demonstrate the entire MAERI 2.0 Flow

- **Workload: ResNet 50**

Layer Type	Convolution 2d	BatchNorm 2d	ReLU	Skip Add	Average Pooling	Fully Connected
Number of Layers	53	53	49	16	1	1

- **Platform: zcu 104** 

Name	OS	python	PyTorch	Quantization Scheme	Dataset	CPU
Version	pynqlinux v2.6 (18.04)	3.6.5	1.8.1	qnnpack	imagenet 1k	Dual-core Arm Cortex-R5F

- **DEMO 1 (Pre-recorded)**
  - Setup Environment
  - Model Quantization
  - Data Layout Reorder (PyTorch Default order)
  - Custom Inference

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

Goal of DEMO 1: Demonstrate the entire MAERI 2.0 Flow

- **Workload: ResNet 50**

Layer Type	Convolution 2d	BatchNorm 2d	ReLU	Skip Add	Average Pooling	Fully Connected
Number of Layers	53	53	49	16	1	1

- **Platform: zcu 104** 

Name	OS	python	PyTorch	Quantization Scheme	Dataset	CPU
Version	pynqlinux v2.6 (18.04)	3.6.5	1.8.1	qnnpack	imagenet 1k	Dual-core Arm Cortex-R5F

- **DEMO 1 (Pre-recorded)**
  - Setup Environment
  - Model Quantization
  - Data Layout Reorder (PyTorch Default order)
  - Custom Inference

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

- **DEMO 1 (Pre-run offline)**
  - Setup Environment
  - Model Quantization
  - Data Layout Reorder (PyTorch Default order)

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

- **DEMO 1 (Pre-run offline)**
  - Setup Environment
  - Model Quantization
  - Data Layout Reorder (PyTorch Default order)

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

- **DEMO 1 (Pre-run offline)**
  - Setup Environment
  - Model Quantization
  - Data Layout Reorder (PyTorch Default order)

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

- **DEMO 1 (Pre-run offline)**
  - Setup Environment
  - Model Quantization
  - Data Layout Reorder (PyTorch Default order)

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

- **DEMO 1 (Pre-run offline)**
  - Setup Environment

```
Validate: 100%|██████████| 250/250 [24:13<00:00,  5.78s/it, loss=0.909, top1=76.7, top5=93, img_size=224]
Results: loss=0.90904,   top1=76.7,      top5=93.0
```

DEMO 1 - MAERI 2.0 Entire Flow Demonstration

- **DEMO 1 (Pre-run offline)**
  - Setup Environment

```
Validate: 100%|██████████| 250/250 [24:13<00:00,  5.78s/it, loss=0.909, top1=76.7, top5=93, img_size=224]

Results: loss=0.90904,   top1=76.7,      top5=93.0
```

  - Custom Inference

```
In [13]: import cv2
image = cv2.imread(image_folder.imgs[fp_input_labels][0])
_, ax = plt.subplots(1)
_ = ax.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
```


```
In [14]: import label as label_list
print(f"the inferenced category of the image is:    '{label_list.IMAGENET_LABELS[int(torch.argmax(sample_input).deta

the inferenced category of the image is:    'horned viper, cerastes, sand viper, horned asp, Cerastes cornutus'
```

Outlines

- Supported Neural Network Model
- Quantization Flow
- Memory Layout
- Heterogeneous Scheduling
- MAERI 2.0 Microarchitecture
- **DEMO**
  - Entire Flow Demonstration
  - **Performance Evaluation of Conv Accel.**

DEMO 2 - Performance Evaluation of Conv Accel.

DEMO 2 - Performance Evaluation of Conv Accel.

Goal of DEMO: Evaluate convolution performance of MAERI 2.0 Accel.

DEMO 2 - Performance Evaluation of Conv Accel.

Goal of DEMO: Evaluate convolution performance of MAERI 2.0 Accel.

- **Workload:** Once-for-all ResNet50 [1]
  - 36 Convolution Layers

DEMO 2 - Performance Evaluation of Conv Accel.

Goal of DEMO: Evaluate convolution performance of MAERI 2.0 Accel.

- **Workload:** Once-for-all ResNet50 [1]
  - 36 Convolution Layers

DEMO 2 - Performance Evaluation of Conv Accel.

Goal of DEMO: Evaluate convolution performance of MAERI 2.0 Accel.

- **Workload:** Once-for-all ResNet50 [1]
  - 36 Convolution Layers
- **Platform: zcu 104**
  - DRAM-PL bandwidth (High Performance Interface): use 512 bits out of 768-bit

DEMO 2 - Performance Evaluation of Conv Accel.

Goal of DEMO: Evaluate convolution performance of MAERI 2.0 Accel.

- **Workload:** Once-for-all ResNet50 [1]
  - 36 Convolution Layers
- **Platform: zcu 104**
  - DRAM-PL bandwidth (High Performance Interface): use 512 bits out of 768-bit

DEMO 2 - Performance Evaluation of Conv Accel.

Goal of DEMO: Evaluate convolution performance of MAERI 2.0 Accel.

- **Workload:** Once-for-all ResNet50 [1]
  - 36 Convolution Layers
- **Platform: zcu 104**
  - DRAM-PL bandwidth (High Performance Interface): use 512 bits out of 768-bit
    - Bandwidth restricted parallelism: $K_P = 3$ (3 kernels)

DEMO 2 - Performance Evaluation of Conv Accel.

Goal of DEMO: Evaluate convolution performance of MAERI 2.0 Accel.

- **Workload:** Once-for-all ResNet50 [1]
  - 36 Convolution Layers
- **Platform: zcu 104**
  - DRAM-PL bandwidth (High Performance Interface): use 512 bits out of 768-bit
    - Bandwidth restricted parallelism: $K_P = 3$ (3 kernels)
    - Workload Preferred parallelism: $Y_P = 4$ (4 Sliding Windows)
  - L2 tile size (base on Parallelism):

DEMO 2 - Performance Evaluation of Conv Accel.

Goal of DEMO: Evaluate convolution performance of MAERI 2.0 Accel.

- **Workload:** Once-for-all ResNet50 [1]
  - 36 Convolution Layers
- **Platform: zcu 104**
  - DRAM-PL bandwidth (High Performance Interface): use 512 bits out of 768-bit
    - Bandwidth restricted parallelism: $K_P = 3$ (3 kernels)
    - Workload Preferred parallelism: $Y_P = 4$ (4 Sliding Windows)
  - L2 tile size (base on Parallelism):
  - L3 tile size (Based on DSE Tool): $(T_K, T_C, R, S, T_X, T_Y) = (12, 32, 3, 3, 104, 226)$

DEMO 2 - Performance Evaluation of Conv Accel.

Goal of DEMO: Evaluate convolution performance of MAERI 2.0 Accel.

- **Workload:** Once-for-all ResNet50 [1]
  - 36 Convolution Layers
- **Platform: zcu 104**
  - DRAM-PL bandwidth (High Performance Interface): use 512 bits out of 768-bit
    - Bandwidth restricted parallelism: $K_P = 3$ (3 kernels)
    - Workload Preferred parallelism: $Y_P = 4$ (4 Sliding Windows)
  - L2 tile size (base on Parallelism):
  - L3 tile size (Based on DSE Tool): $(T_K, T_C, R, S, T_X, T_Y) = (12, 32, 3, 3, 104, 226)$

- **MAERI 2.0 Configuration**
  - Streaming Buffer: 85696 x 64-bit
  - Stationary Buffer: 8192 x 80-bit
  - Line Buffer: 88 x 64-bit
  - No Weights Buffer (Weights Stationary)
  - Output Buffer: 1326 x 512-bit

DEMO 2 - Performance Evaluation of Conv Accel.

Goal of DEMO: Evaluate convolution performance of MAERI 2.0 Accel.

- **Workload:** Once-for-all ResNet50 [1]
  - 36 Convolution Layers
- **Platform: zcu 104**
  - DRAM-PL bandwidth (High Performance Interface): use 512 bits out of 768-bit
    - Bandwidth restricted parallelism: $K_P = 3$ (3 kernels)
    - Workload Preferred parallelism: $Y_P = 4$ (4 Sliding Windows)
  - L2 tile size (base on Parallelism):
  - L3 tile size (Based on DSE Tool): $(T_K, T_C, R, S, T_X, T_Y) = (12, 32, 3, 3, 104, 226)$

- **MAERI 2.0 Configuration**
  - Streaming Buffer: 85696 x 64-bit
  - Stationary Buffer: 8192 x 80-bit
  - Line Buffer: 88 x 64-bit
  - No Weights Buffer (Weights Stationary)
  - Output Buffer: 1326 x 512-bit

DEMO 2 - Performance Evaluation of Conv Accel.

Goal of DEMO: Evaluate convolution performance of MAERI 2.0 Accel.

- **Workload:** Once-for-all ResNet50 [1]
  - 36 Convolution Layers
- **Platform: zcu 104**
  - DRAM-PL bandwidth (High Performance Interface): use 512 bits out of 768-bit
    - Bandwidth restricted parallelism: $K_P = 3$ (3 kernels)
    - Workload Preferred parallelism: $Y_P = 4$ (4 Sliding Windows)
  - L2 tile size (base on Parallelism):
  - L3 tile size (Based on DSE Tool): $(T_K, T_C, R, S, T_X, T_Y) = (12, 32, 3, 3, 104, 226)$

- **MAERI 2.0 Configuration**
  - Streaming Buffer: 85696 x 64-bit
  - Stationary Buffer: 8192 x 80-bit
  - Line Buffer: 88 x 64-bit
  - No Weights Buffer (Weights Stationary)
  - Output Buffer: 1326 x 512-bit

DEMO 2 - Performance Evaluation of Conv Accel.

Goal of DEMO: Evaluate convolution performance of MAERI 2.0 Accel.

- **Workload:** Once-for-all ResNet50 [1]
  - 36 Convolution Layers
- **Platform: zcu 104**
  - DRAM-PL bandwidth (High Performance Interface): use 512 bits out of 768-bit
    - Bandwidth restricted parallelism: $K_P = 3$ (3 kernels)
    - Workload Preferred parallelism: $Y_P = 4$ (4 Sliding Windows)
  - L2 tile size (base on Parallelism):
  - L3 tile size (Based on DSE Tool): $(T_K, T_C, R, S, T_X, T_Y) = (12, 32, 3, 3, 104, 226)$

- **MAERI 2.0 Configuration**
  - Streaming Buffer: 85696 x 64-bit
  - Stationary Buffer: 8192 x 80-bit
  - Line Buffer: 88 x 64-bit
  - No Weights Buffer (Weights Stationary)
  - Output Buffer: 1326 x 512-bit

DEMO 2 - Performance Evaluation of Conv Accel.

Goal of DEMO: Evaluate convolution performance of MAERI 2.0 Accel.

- **Workload:** Once-for-all ResNet50 [1]
  - 36 Convolution Layers
- **Platform: zcu 104**
  - DRAM-PL bandwidth (High Performance Interface): use 512 bits out of 768-bit
    - Bandwidth restricted parallelism: $K_P = 3$ (3 kernels)
    - Workload Preferred parallelism: $Y_P = 4$ (4 Sliding Windows)
  - L2 tile size (base on Parallelism):
  - L3 tile size (Based on DSE Tool): $(T_K, T_C, R, S, T_X, T_Y) = (12, 32, 3, 3, 104, 226)$

- **MAERI 2.0 Configuration**
  - Streaming Buffer: 85696 x 64-bit
  - Stationary Buffer: 8192 x 80-bit
  - Line Buffer: 88 x 64-bit
  - No Weights Buffer (Weights Stationary)
  - Output Buffer: 1326 x 512-bit

DEMO 2 - Performance Evaluation of Conv Accel.

Goal of DEMO: Evaluate convolution performance of MAERI 2.0 Accel.

- **Workload:** Once-for-all ResNet50 [1]
  - 36 Convolution Layers
- **Platform: zcu 104**
  - DRAM-PL bandwidth (High Performance Interface): use 512 bits out of 768-bit
    - Bandwidth restricted parallelism: $K_P = 3$ (3 kernels)
    - Workload Preferred parallelism: $Y_P = 4$ (4 Sliding Windows)
  - L2 tile size (base on Parallelism):
  - L3 tile size (Based on DSE Tool): $(T_K, T_C, R, S, T_X, T_Y) = (12, 32, 3, 3, 104, 226)$

- **MAERI 2.0 Configuration**
  - Streaming Buffer: 85696 x 64-bit
  - Stationary Buffer: 8192 x 80-bit
  - Line Buffer: 88 x 64-bit
  - No Weights Buffer (Weights Stationary)
  - Output Buffer: 1326 x 512-bit

DEMO 2 - Pre-run Results

DEMO 2 - Pre-run Results

- Run DSE tool to get best hardware configuration
  - 30-mins on 10-th Core I7 10750H.
- Generate tiling strategy for specific workload
- Run on the Xilinx ZCU 104 (DEMO 2 today)
  - Load Pre-generated Tiling Strategy
  - Configure FPGA with MAERI 2.0 bitstream
  - Run the network inference.

DEMO 2 - Pre-run Results

- Run DSE tool to get best hardware configuration
  - 30-mins on 10-th Core I7 10750H.
- Generate tiling strategy for specific workload
- Run on the Xilinx ZCU 104 (DEMO 2 today)
  - Load Pre-generated Tiling Strategy
  - Configure FPGA with MAERI 2.0 bitstream
  - Run the network inference.

DEMO 2 - Pre-run Results

- Run DSE tool to get best hardware configuration
  - 30-mins on 10-th Core I7 10750H.
- Generate tiling strategy for specific workload
- Run on the Xilinx ZCU 104 (DEMO 2 today)
  - Load Pre-generated Tiling Strategy
  - Configure FPGA with MAERI 2.0 bitstream
  - Run the network inference.

DEMO 2 - Pre-run Results

- Run DSE tool to get best hardware configuration
  - 30-mins on 10-th Core I7 10750H.
- Generate tiling strategy for specific workload
- Run on the Xilinx ZCU 104 (DEMO 2 today)
  - Load Pre-generated Tiling Strategy
  - Configure FPGA with MAERI 2.0 bitstream
  - Run the network inference.

DEMO 2 - Pre-run Results

- Run DSE tool to get best hardware configuration
  - 30-mins on 10-th Core I7 10750H.
- Generate tiling strategy for specific workload
- Run on the Xilinx ZCU 104 (DEMO 2 today)
  - Load Pre-generated Tiling Strategy
  - Configure FPGA with MAERI 2.0 bitstream
  - Run the network inference.

DEMO 2 - Pre-run Results

- Run DSE tool to get best hardware configuration
  - 30-mins on 10-th Core I7 10750H.
- Generate tiling strategy for specific workload
- Run on the Xilinx ZCU 104 (DEMO 2 today)
  - Load Pre-generated Tiling Strategy
  - Configure FPGA with MAERI 2.0 bitstream
  - Run the network inference.

DEMO 2 - Pre-run Results

- Run DSE tool to get best hardware configuration
  - 30-mins on 10-th Core I7 10750H.
- Generate tiling strategy for specific workload
- Run on the Xilinx ZCU 104 (DEMO 2 today)
  - Load Pre-generated Tiling Strategy
  - Configure FPGA with MAERI 2.0 bitstream
  - Run the network inference.

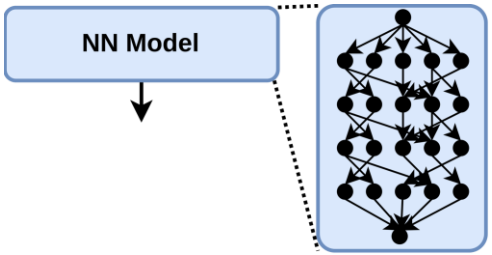
DEMO 2 - Pre-run Results

- Run DSE tool to get best hardware configuration
  - 30-mins on 10-th Core I7 10750H.
- Generate tiling strategy for specific workload
- Run on the Xilinx ZCU 104 (DEMO 2 today)
  - Load Pre-generated Tiling Strategy
  - Configure FPGA with MAERI 2.0 bitstream
  - Run the network inference.

```
Conv Layer Index: 0
Conv Layer Index: 1
Conv Layer Index: 2
Conv Layer Index: 3
Conv Layer Index: 4
Conv Layer Index: 5
Conv Layer Index: 6
Conv Layer Index: 7
Conv Layer Index: 8
Conv Layer Index: 9
Conv Layer Index: 10
Conv Layer Index: 11
Conv Layer Index: 12
Conv Layer Index: 13
Conv Layer Index: 14
Conv Layer Index: 15
Conv Layer Index: 16
Conv Layer Index: 17
Conv Layer Index: 18
Conv Layer Index: 19
Conv Layer Index: 20
Conv Layer Index: 21
Conv Layer Index: 22
Conv Layer Index: 23
Conv Layer Index: 24
Conv Layer Index: 25
Conv Layer Index: 26
Conv Layer Index: 27
Conv Layer Index: 28
Conv Layer Index: 29
Conv Layer Index: 30
Conv Layer Index: 31
Conv Layer Index: 32
Conv Layer Index: 33
Conv Layer Index: 34
Conv Layer Index: 35
overall latency of running all layers = 103.56596040725708 seconds
```

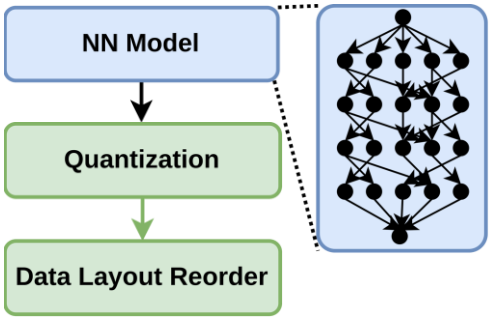
Summary

Summary



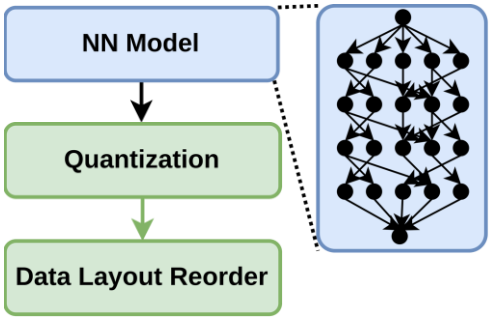
- takes PyTorch NN model

Summary



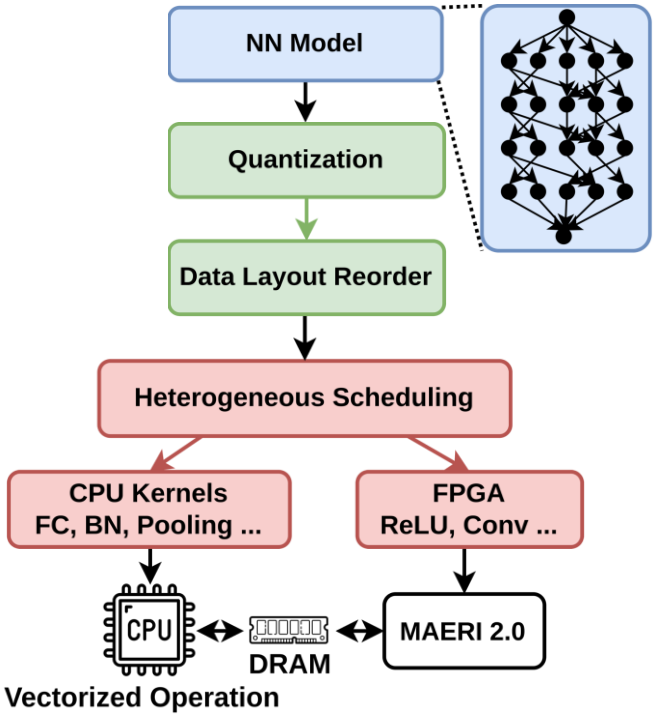
- takes PyTorch NN model
- Quint8 iAct, Qint Weights, Quint8 oAct

Summary



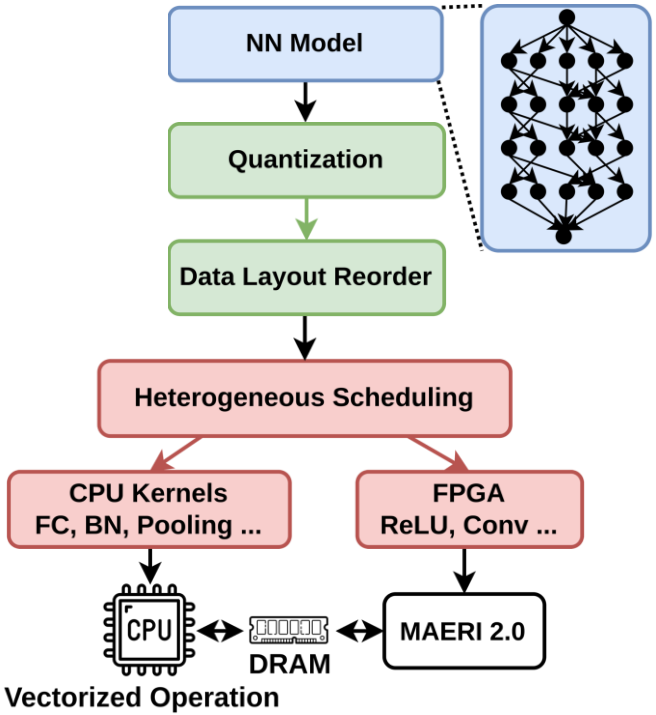
- takes PyTorch NN model
- Quint8 iAct, Qint Weights, Quint8 oAct
- PyTorch default data layout

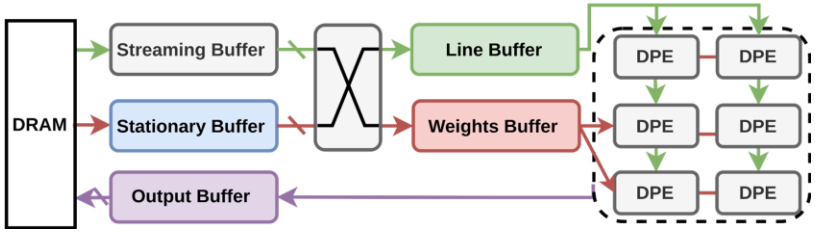
Summary



- takes PyTorch NN model
- Quint8 iAct, Qint Weights, Quint8 oAct
- PyTorch default data layout
- Accelerate Conv on MAERI 2.0 (FPGA)

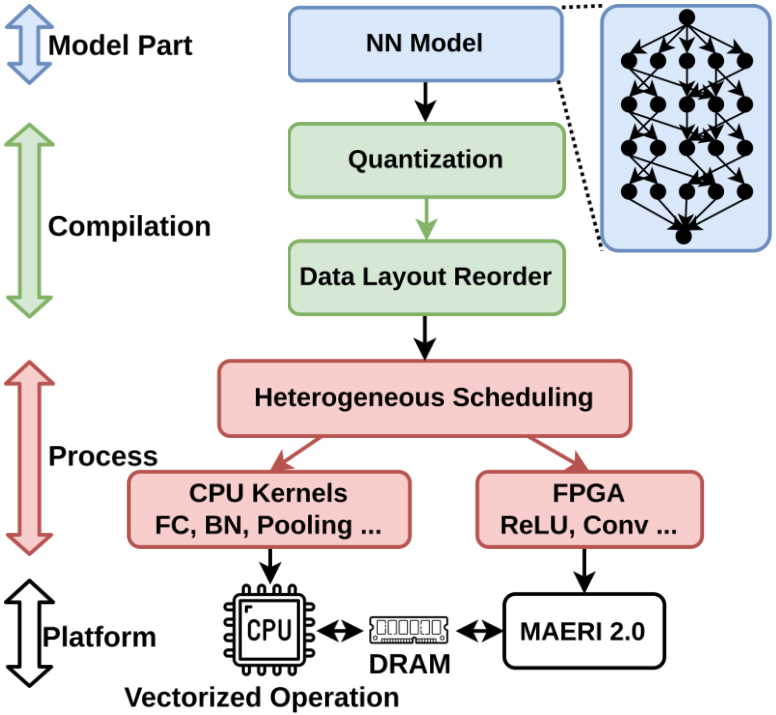
Summary

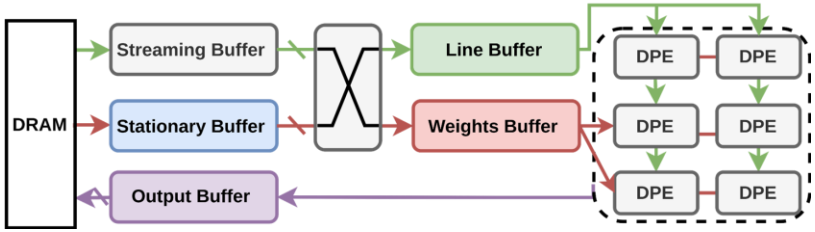




- takes PyTorch NN model
- Quint8 iAct, Qint Weights, Quint8 oAct
- PyTorch default data layout
- Accelerate Conv on MAERI 2.0 (FPGA)
- Multi-tiling memory hierarchy

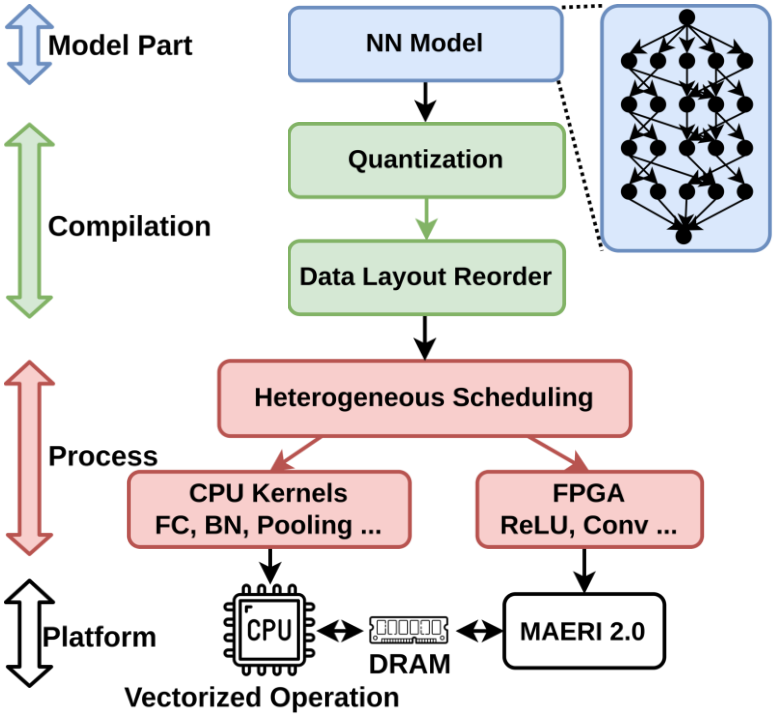
Summary

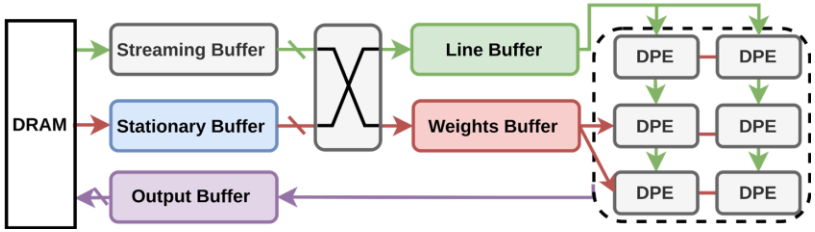




- takes PyTorch NN model
- Quint8 iAct, Qint Weights, Quint8 oAct
- PyTorch default data layout
- Accelerate Conv on MAERI 2.0 (FPGA)
- Multi-tiling memory hierarchy

Summary





- takes PyTorch NN model
- Quint8 iAct, Qint Weights, Quint8 oAct
- PyTorch default data layout
- Accelerate Conv on MAERI 2.0 (FPGA)
- Multi-tiling memory hierarchy
- Design optimization in progress

Summary

Thank You!
Welcome for Questions!
https://maeri-project.github.io/

Join us to build a better framework for researcher!

- Multi-tiling memory hierarchy
- Design optimization in progress