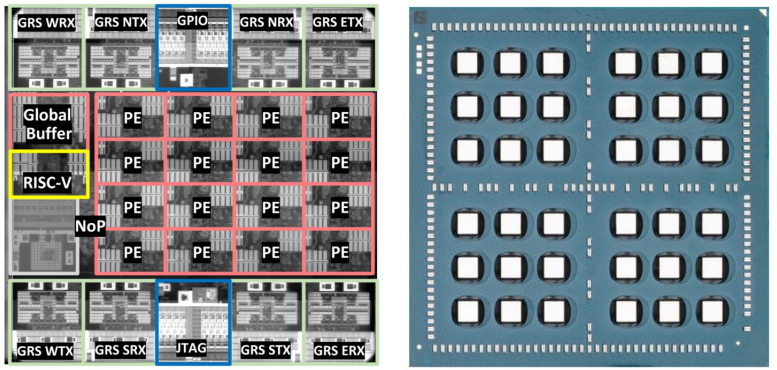
DNN Accelerators Architectures

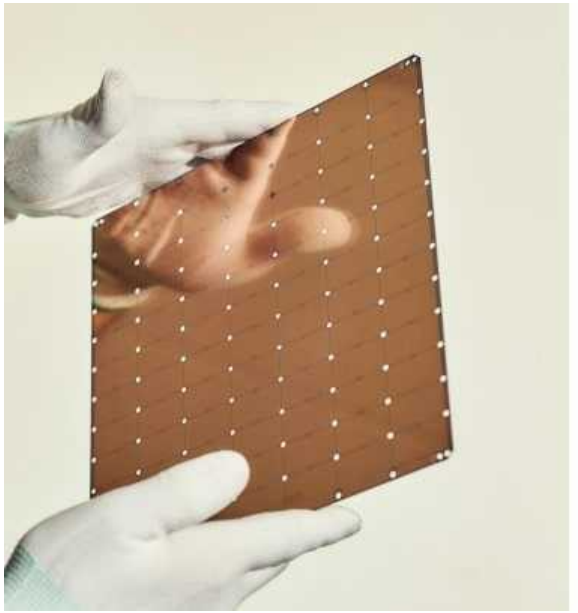
Workshop Agenda

- Lecture 1: Domain Specific Architectures
- Lecture 2: Kernel computation
- Lecture 3: Data-flow techniques
- Lecture 4: DNN accelerators architectures

DNN Accelerators



Simba
36 chiplets x 16 PEs

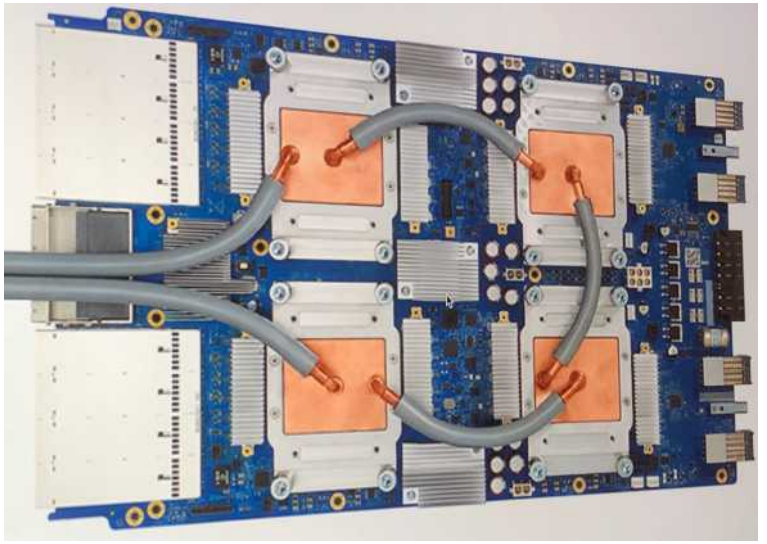


Cerebras Wafer Scale Engine (WSE)

The Most Powerful Processor for AI

400,000 AI-optimized cores
46,225 mm² silicon
1.2 trillion transistors
18 Gigabytes of On-chip Memory
9 PByte/s memory bandwidth
100 Pbit/s fabric bandwidth
TSMC 16nm process

Eyeriss
168 PEs



Google TPU

Agenda

- Simba MCM DNN HW Inference Accelerator
- Improving Energy/Performance with Compression

Multi-Chip-Module Packaging

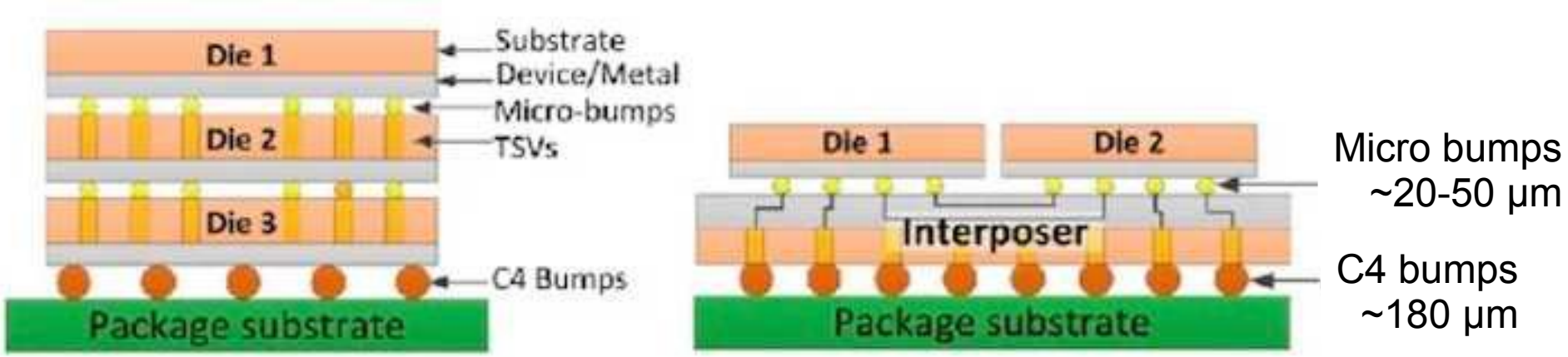
- Slowing transistors scaling
- Reducing cost
  - Yield losses cause fabrication cost to grow super-linearly with die size
- MCM Packaging
  - Assembling large-scale systems using small building blocks known as *chiplets*
  - Multiple chiplets connected via on-package links using a silicon interposer

MCM Package Implementations

- **Large-scale CPUs**
  - Beck, *et al.*, Zeppelin: An SoC for Multichip Architectures. ISSCC'18
  - Erett, *et al.*, A 126mW 56Gb/s NRZ Wireline Transceiver for Synchronous Short-reach Applications in 16nm FinFET. ISSCC'18
  - Kannan, *et al.*, Enabling Interposer-based Disintegration of Multi-core Processors. MICRO'15
  - Wilson, *et al.*, A 1.17pJ/b 25Gb/s/pin Ground-referenced Single-ended Serial Link for Off- and On-package Communication in 16nm CMOS Using a Process- and Temperature-adaptive Voltage Regulator. ISSC'18
- **Large-scale GPUs**
  - Arunkumar, *et al*., MCM-GPU: Multi-Chip-Module GPUs for Continued Performance Scalability. ISCA'17
  - Yin, *et al.*, Modular Routing Design for Chiplet-based Systems. ISCA'18

Die-stacking

- Combination of multiple distinct silicon chips within a single package
  - Vertical or 3D stacking
  - 2.5D stacking



Interposer

- A regular (but larger) silicon chip, with conventional metal layers facing upward
- Two types of interposer
  - Passive
  - Active

Passive Interposer

- Does not provide any transistors
- Only metal routing between chips and TSVs for signals entering/leaving the chip

Active Interposer

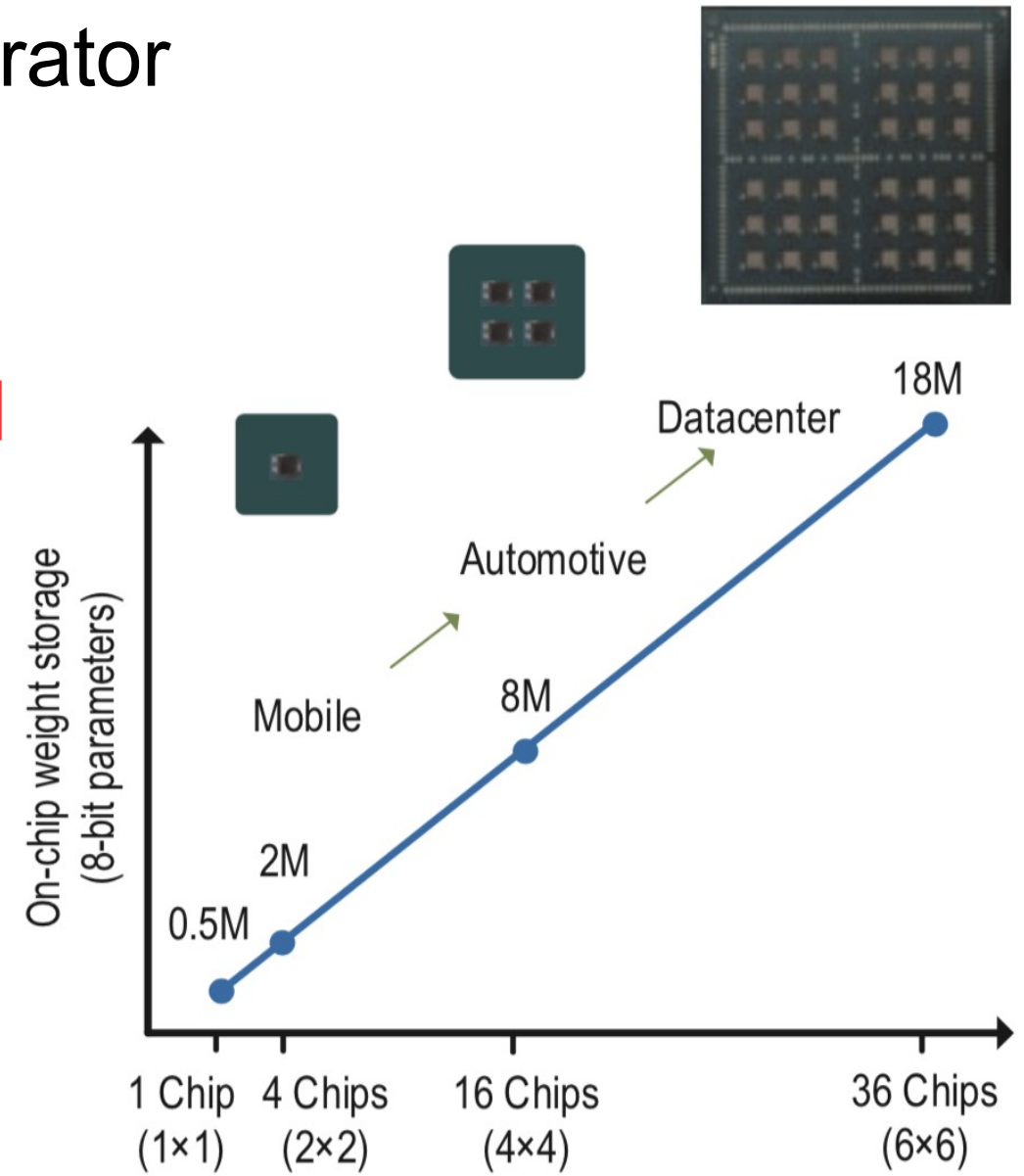
- Integrate some devices (possibly in an older technology)

Agenda

- Simba MCM DNN HW Inference Accelerator
- Improving Energy/Performance with Compression

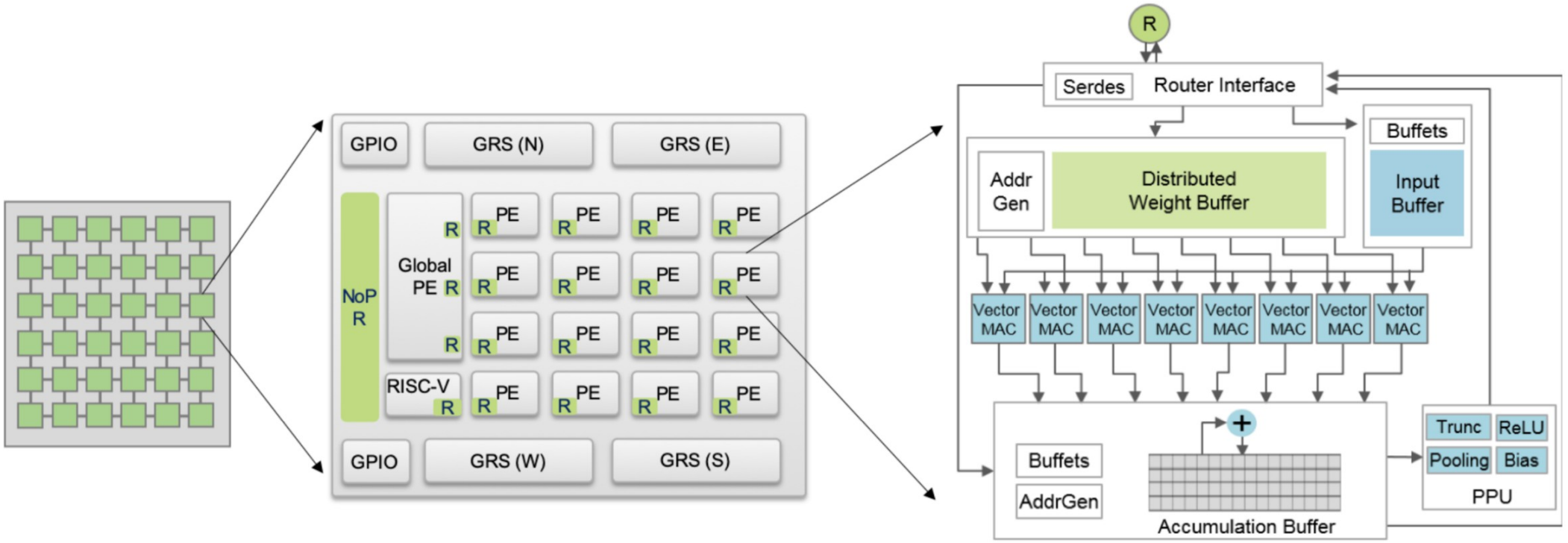
Simba

- DNN inference accelerator
- From edge-scale to data-center-scale
- The first chiplet-based deep-learning system



[Y. S. Shao, *et al.*, Simba: Scaling Deep-Learning Inference with Multi-Chip-Module-Based Architecture. MICRO'19]

3-Level Hierarchy

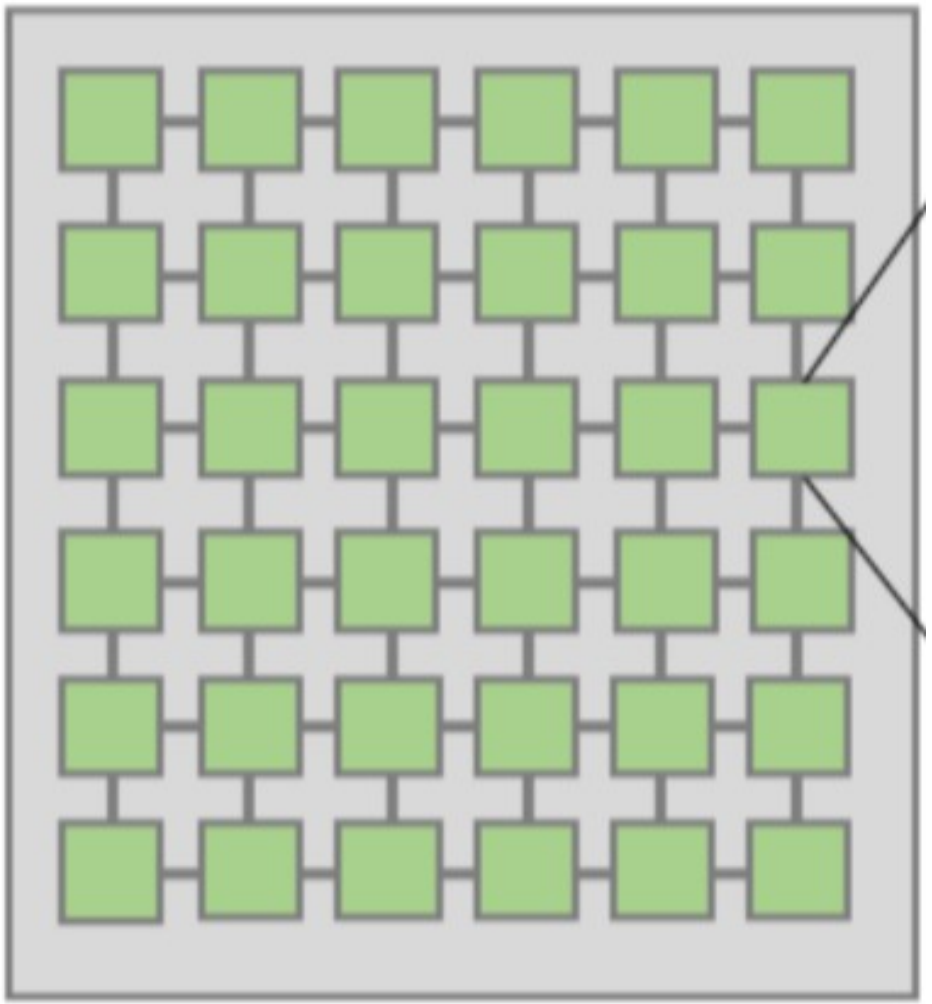


Simba Package

Simba Chiplet

Simba PE

Simba Architecture – Package

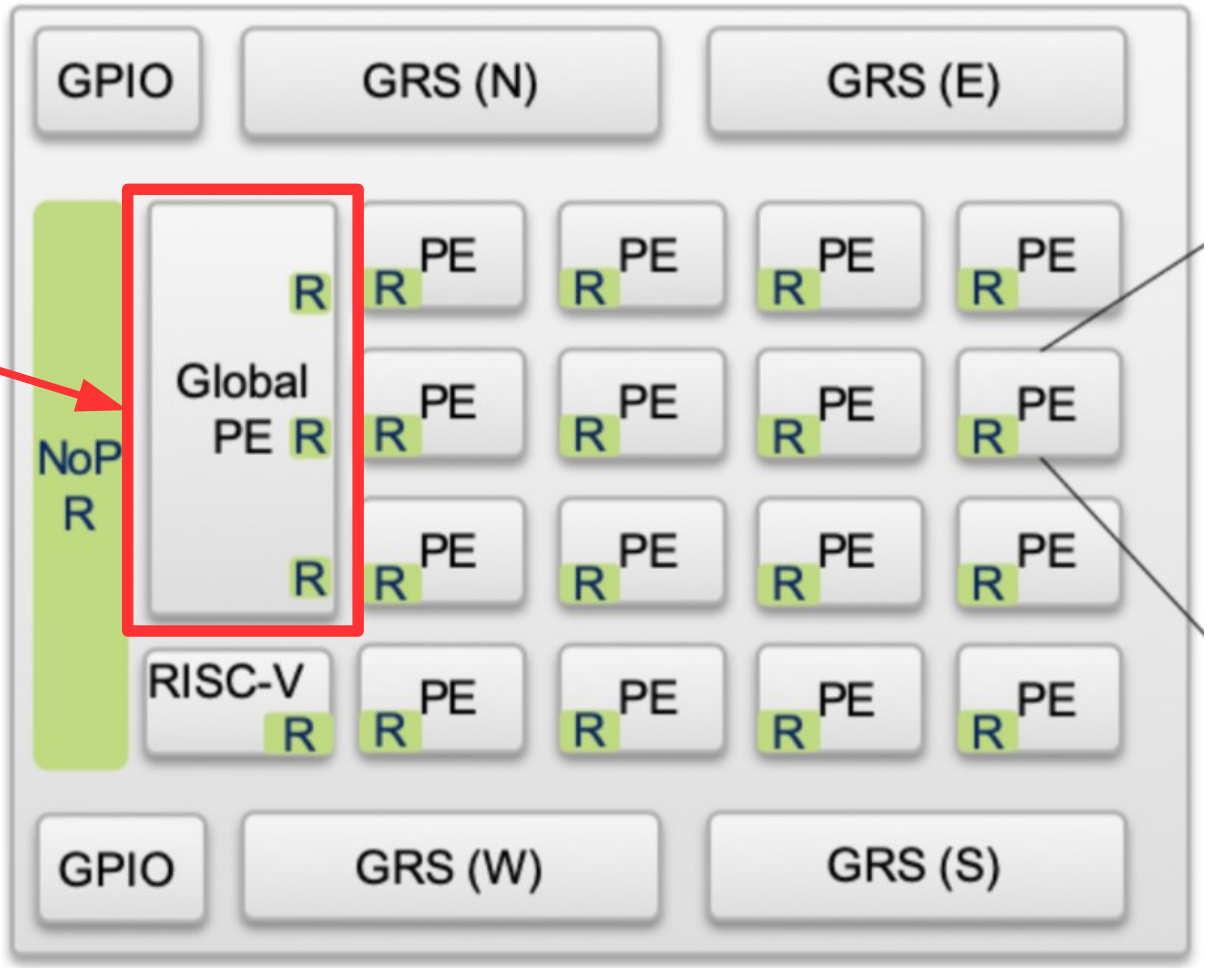


- Package-level MCM integration
- Scalable to data center inference
  - 100 TOPS
  - Google TPU v1 ~ 92 TOPS

Simba Architecture – Chiplet

Second level storage for Input/Output activation data

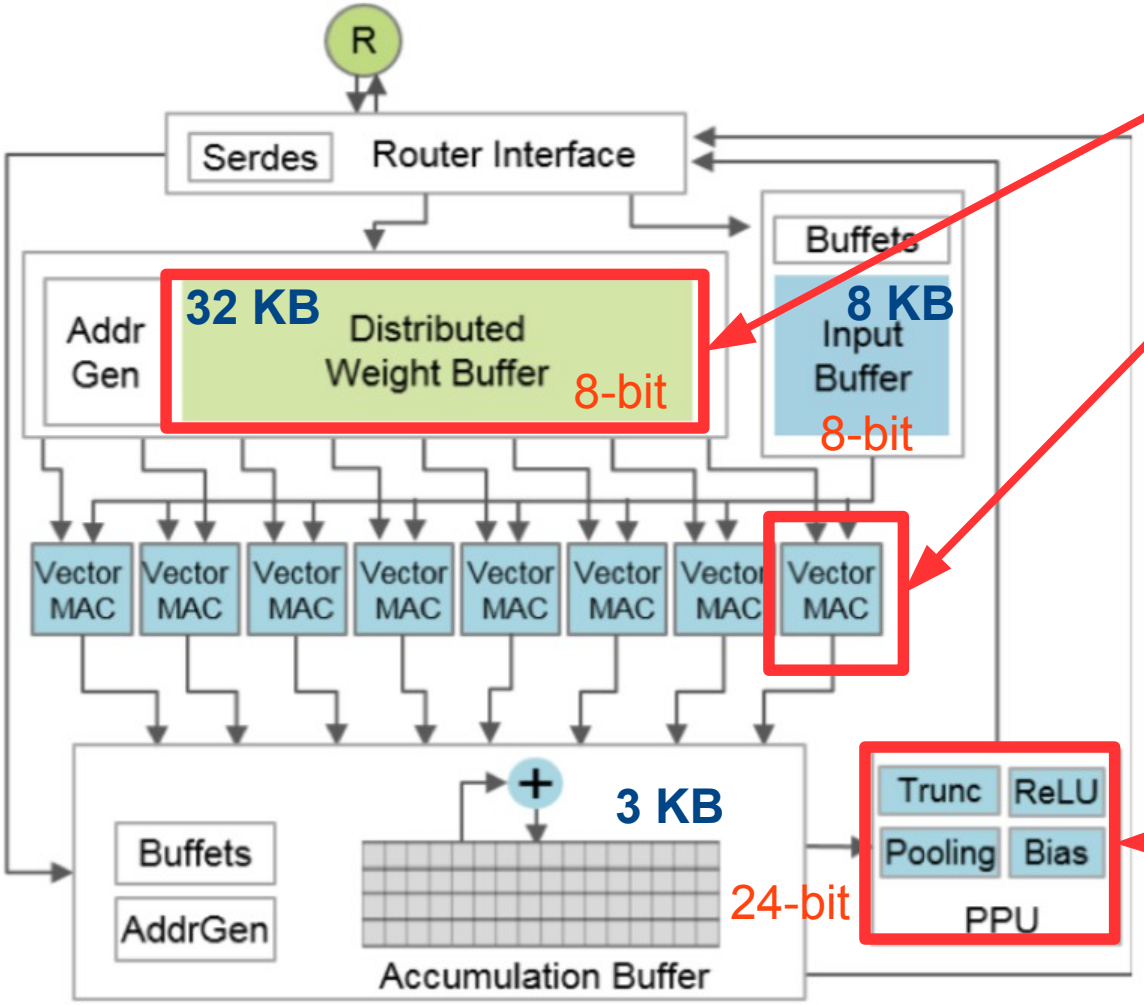
- Supports unicast and multicast (even across chiplets)
- For layers with low data reuse (*e.g.*, depth-wise conv), can perform such computation locally



Simba Architecture – Chiplet

- Responsible for configuring and managing the chiplet's PEs and Global PE
- Triggers execution in the active PEs and Global PEs and waits for done notifications via interrupts

Simba Architecture – PE

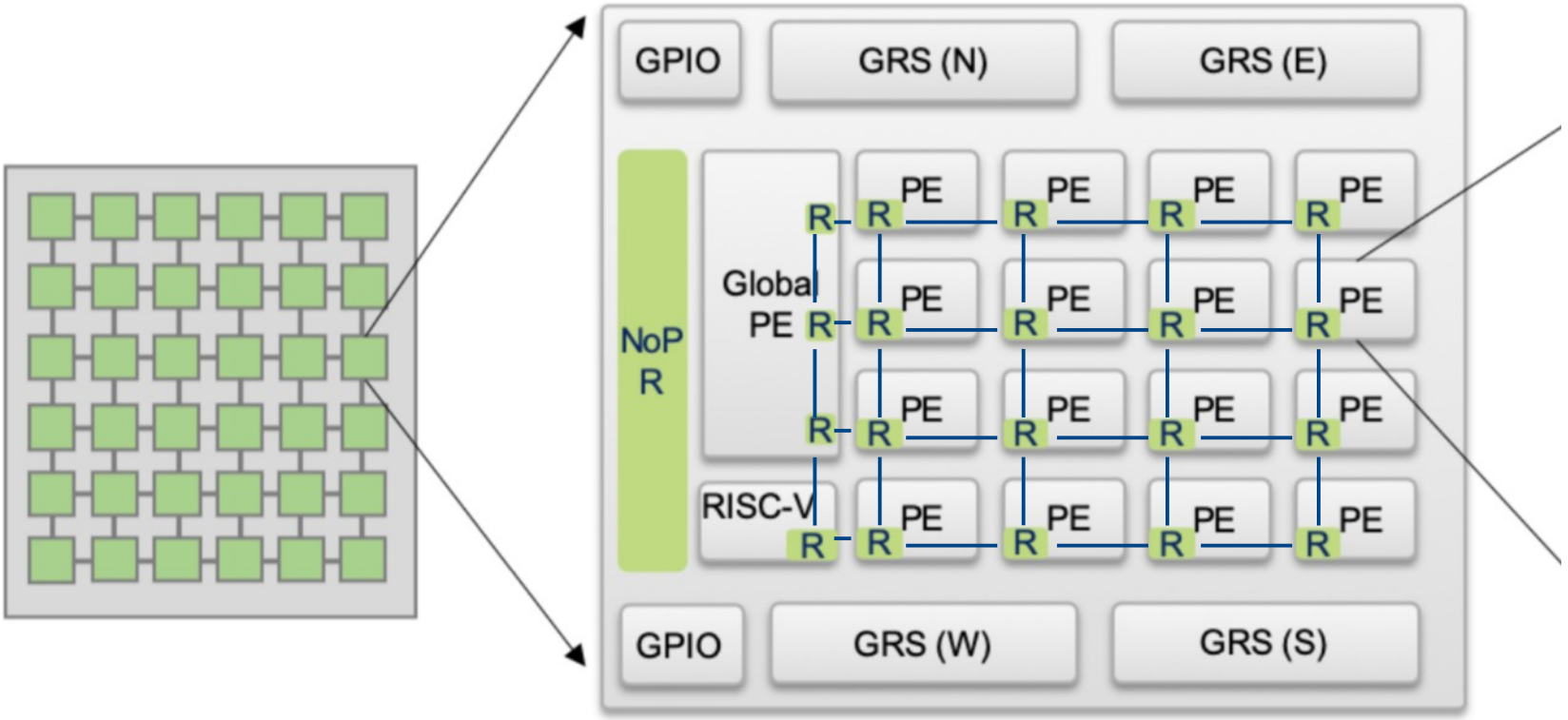


Weight stationary data flow

8:1 dot-product (along channel dim)

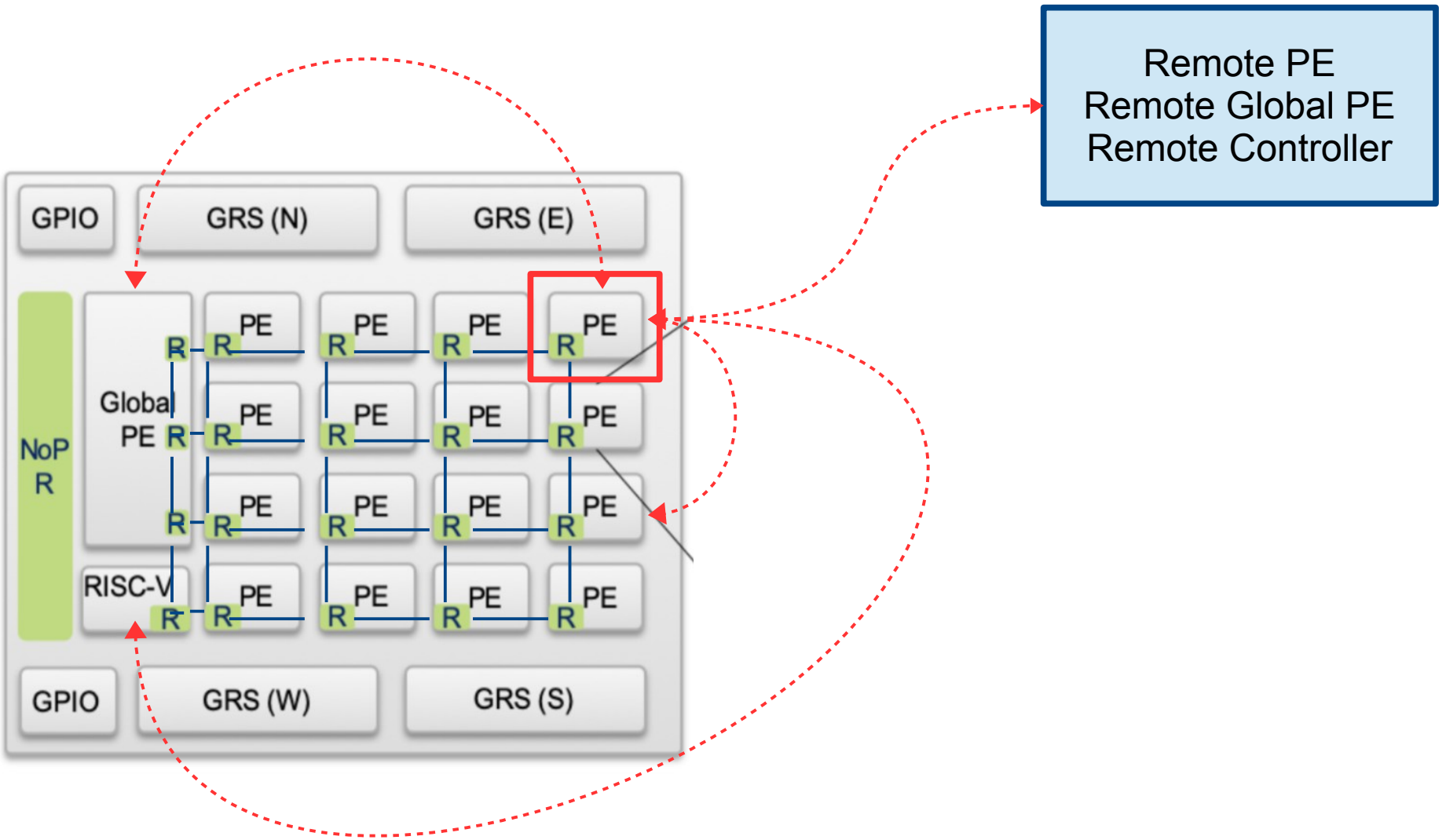
Supports cross-PE reduction for flexible tiling options with configurable producers and consumers.
If the current PE is the last PE on the reduction chain, it first sends partial sums to its local post-processing unit that performs ReLU, truncation and scaling, pooling, and bias addition

Simba Architecture – NoC/NoP

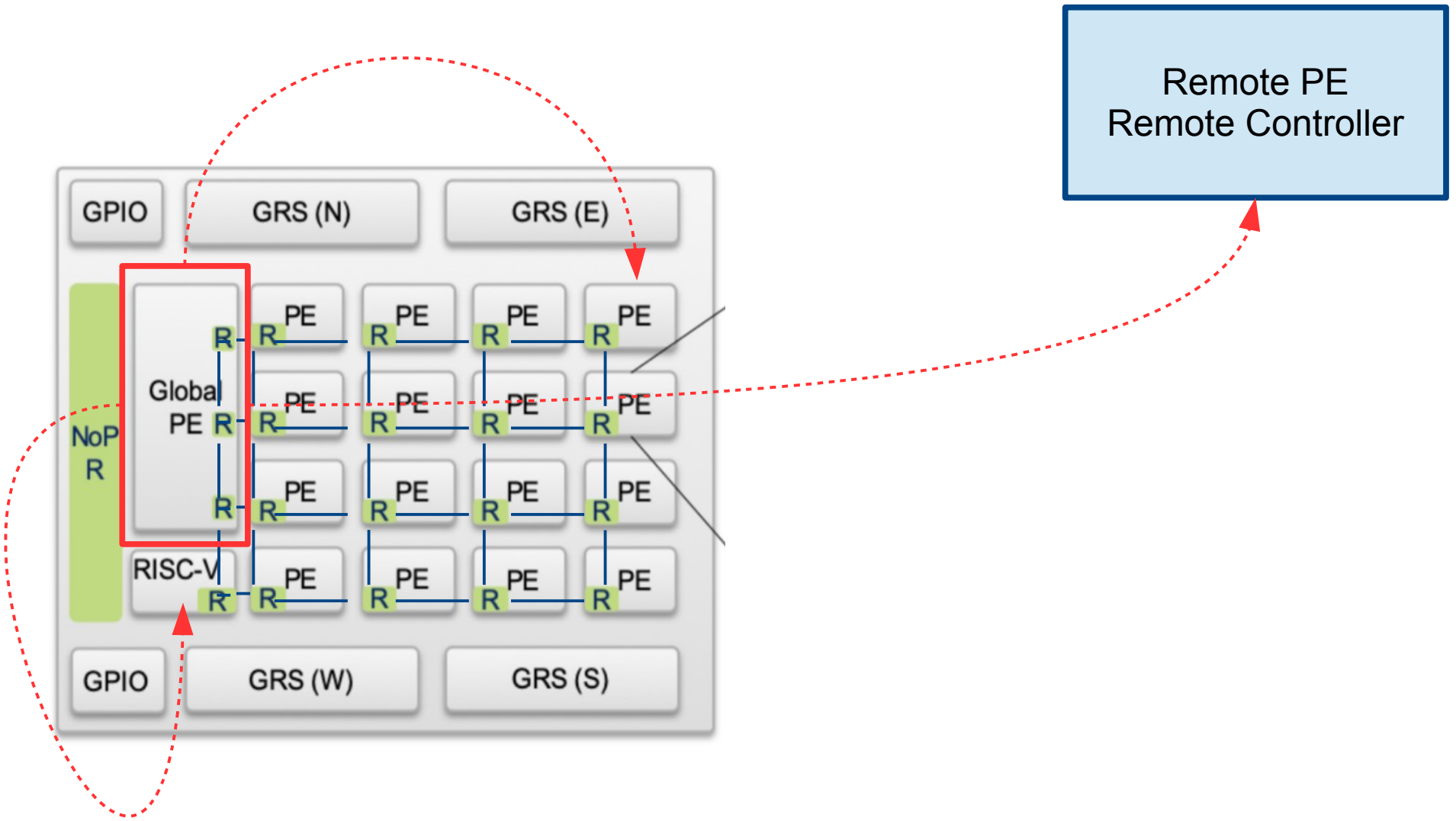


Hybrid NoC/NoP

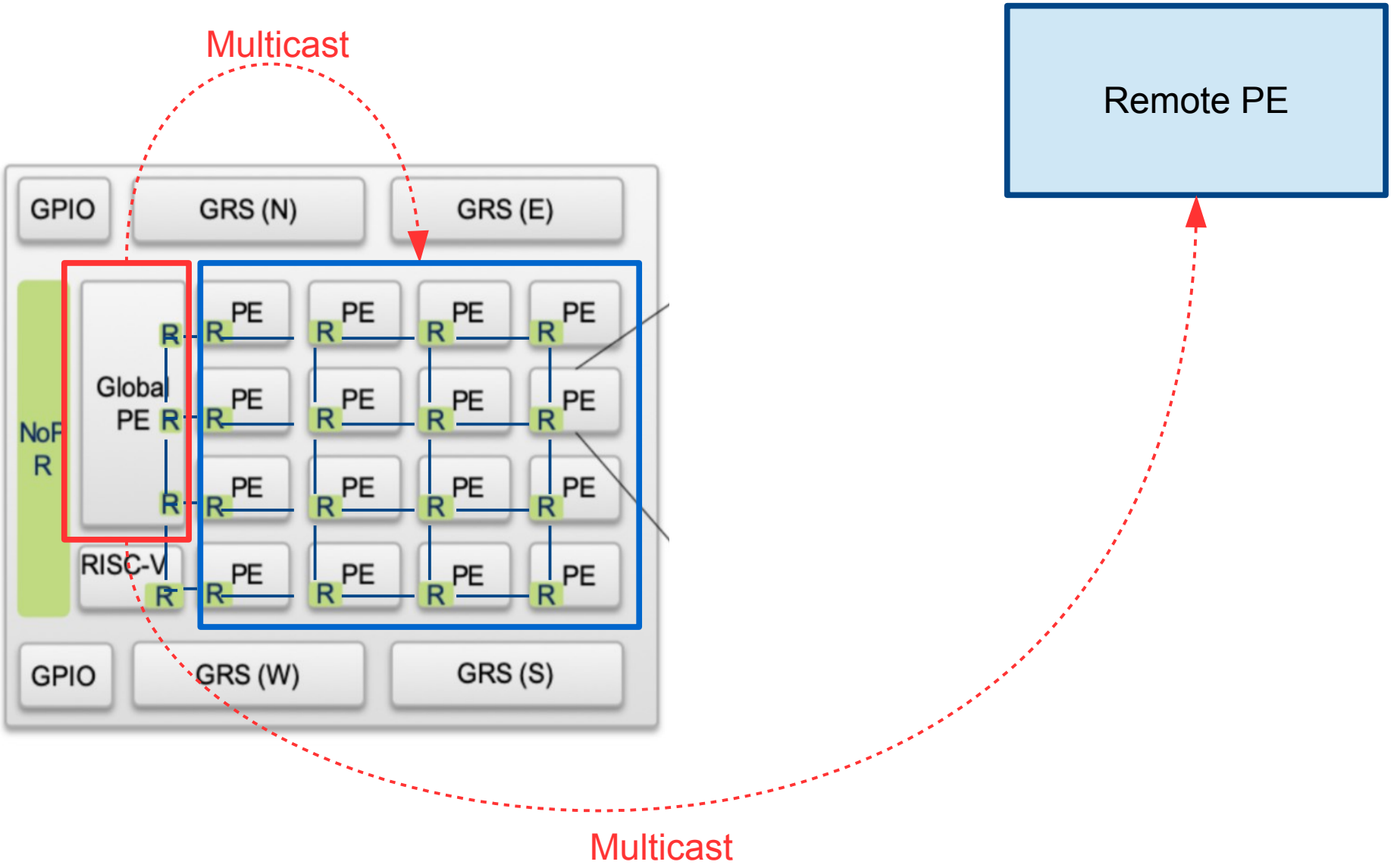
Simba Architecture – NoC/NoP



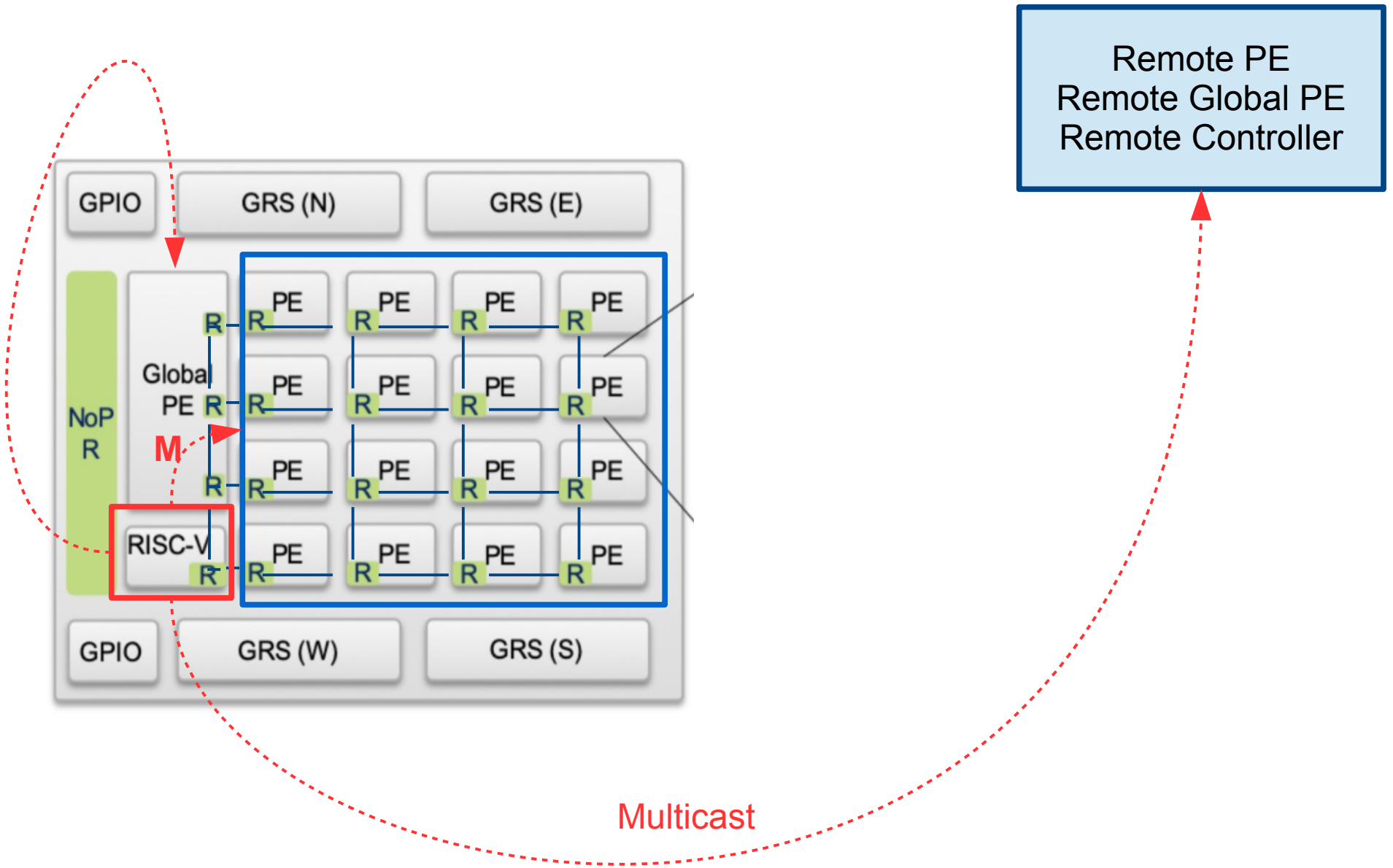
Simba Architecture – NoC/NoP



Simba Architecture – NoC/NoP



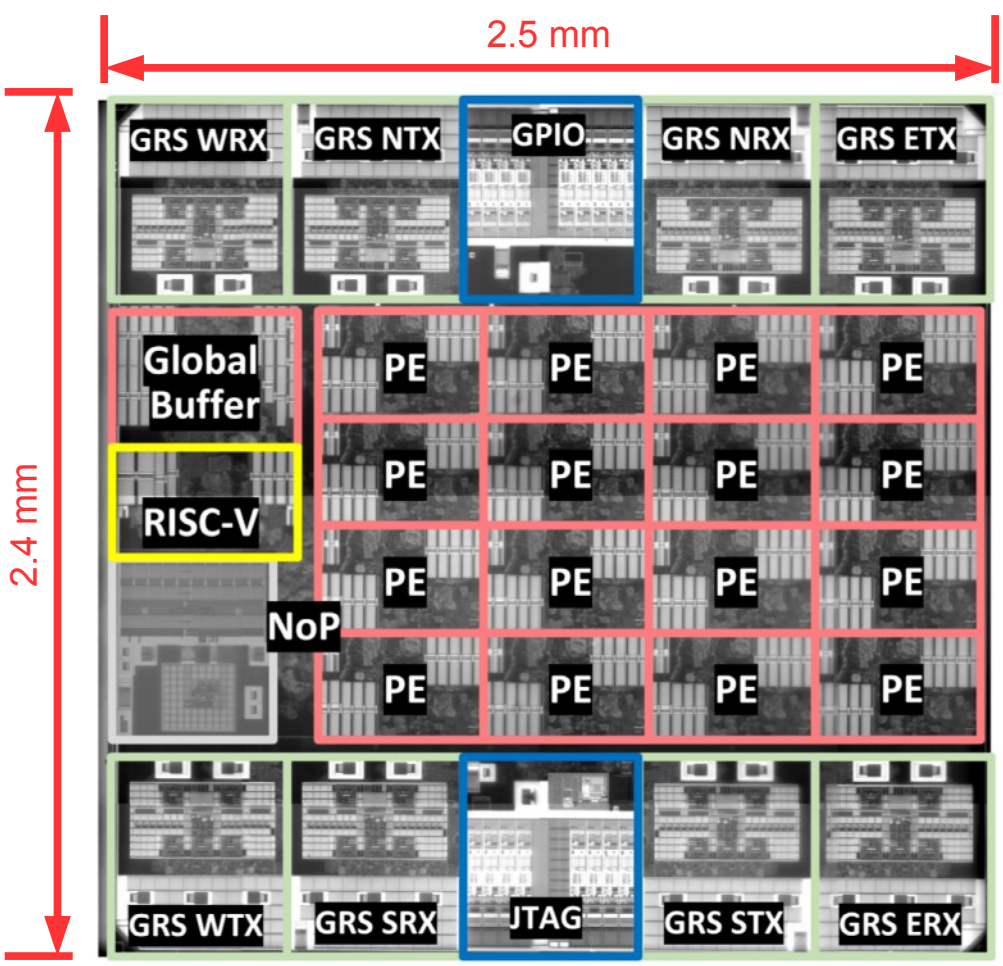
Simba Architecture – NoC/NoP



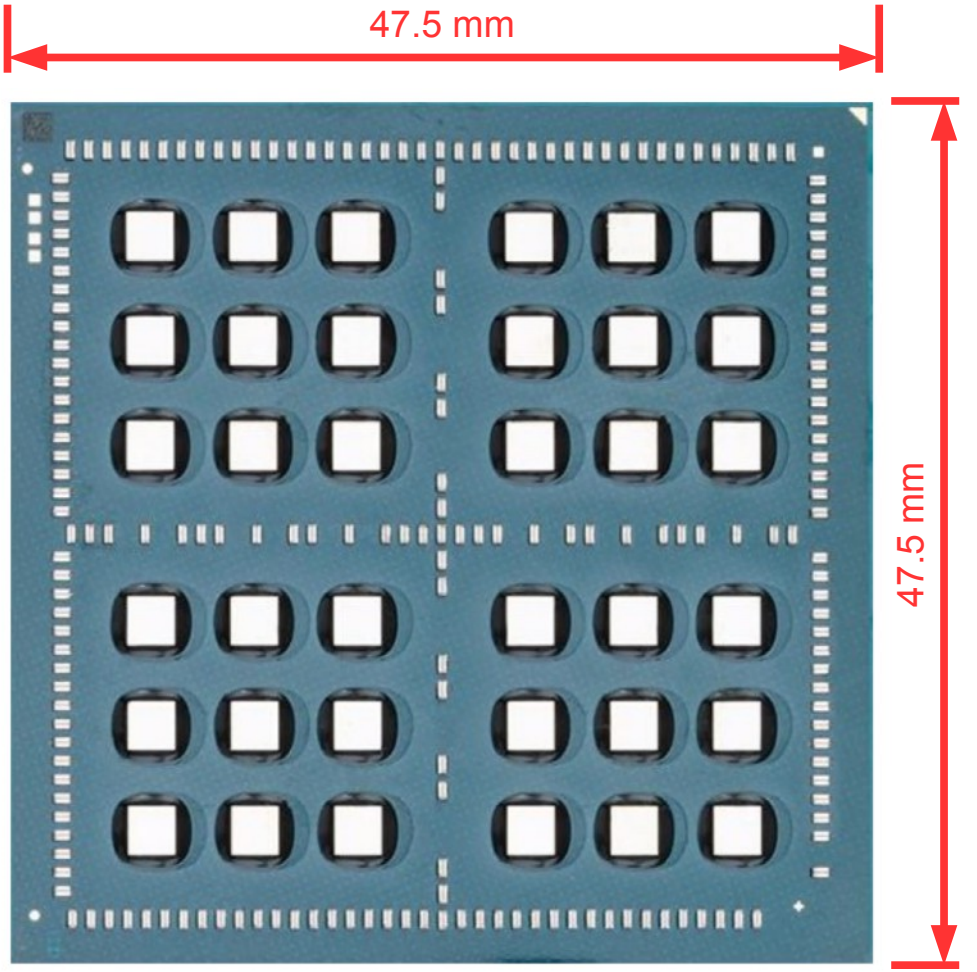
Simba Communication Capability

Packet Source	Unicast Destination	Multicast Destination
PE	Local PEs, Global PE, Controller	-
	Remote PEs, Global PE, Controller	-
Global PE	Local PEs, Controller	Local PEs
	Remote PEs, Controller	Remote PEs
Controller	Local PEs, Global PE	
	Remote PEs, Global PE, Controller	

Simba Silicon Prototype

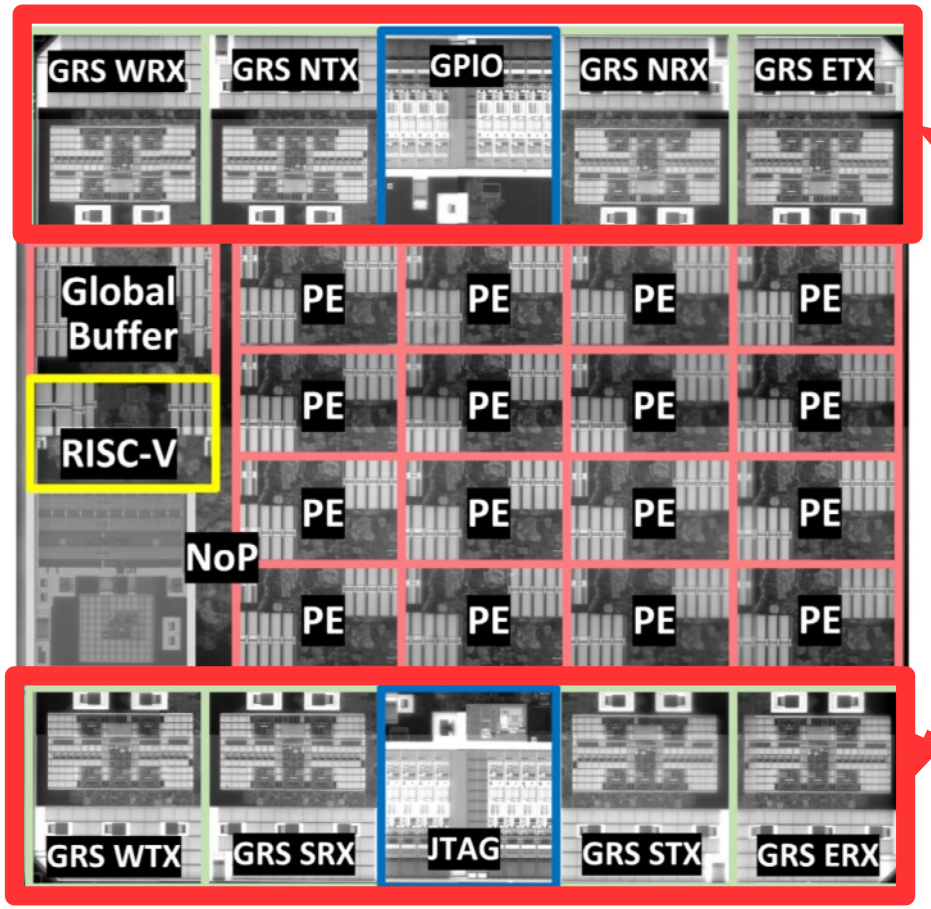


Simba Chiplet
(TSMC 16 nm FinFET)

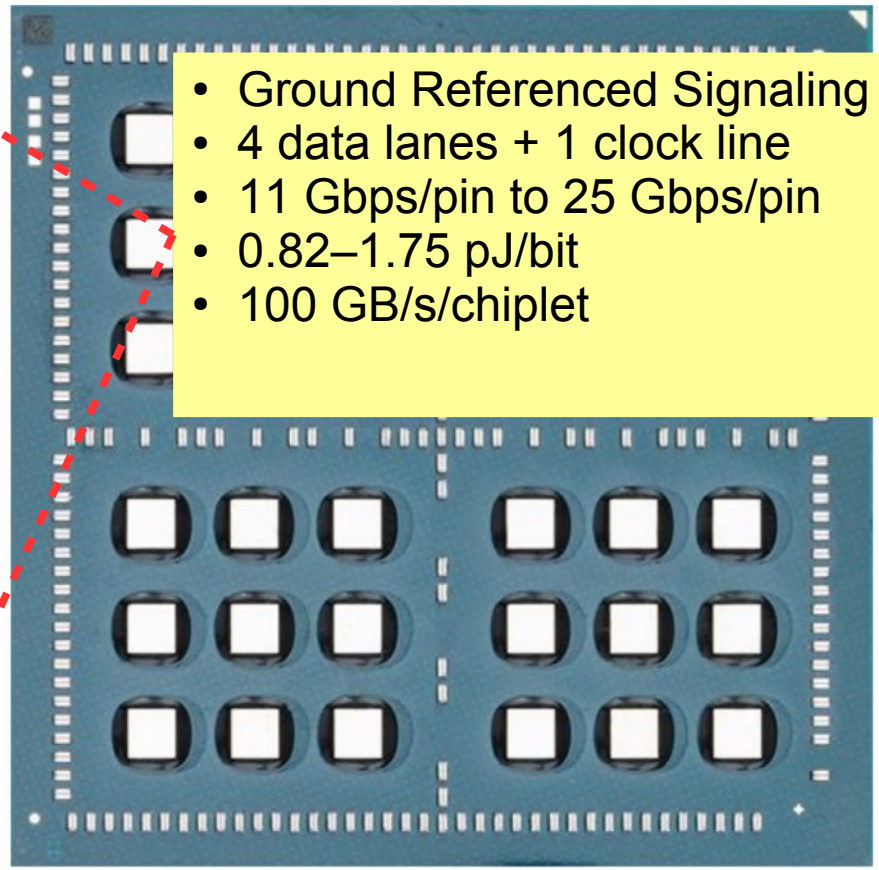


Simba Package

Simba Silicon Prototype



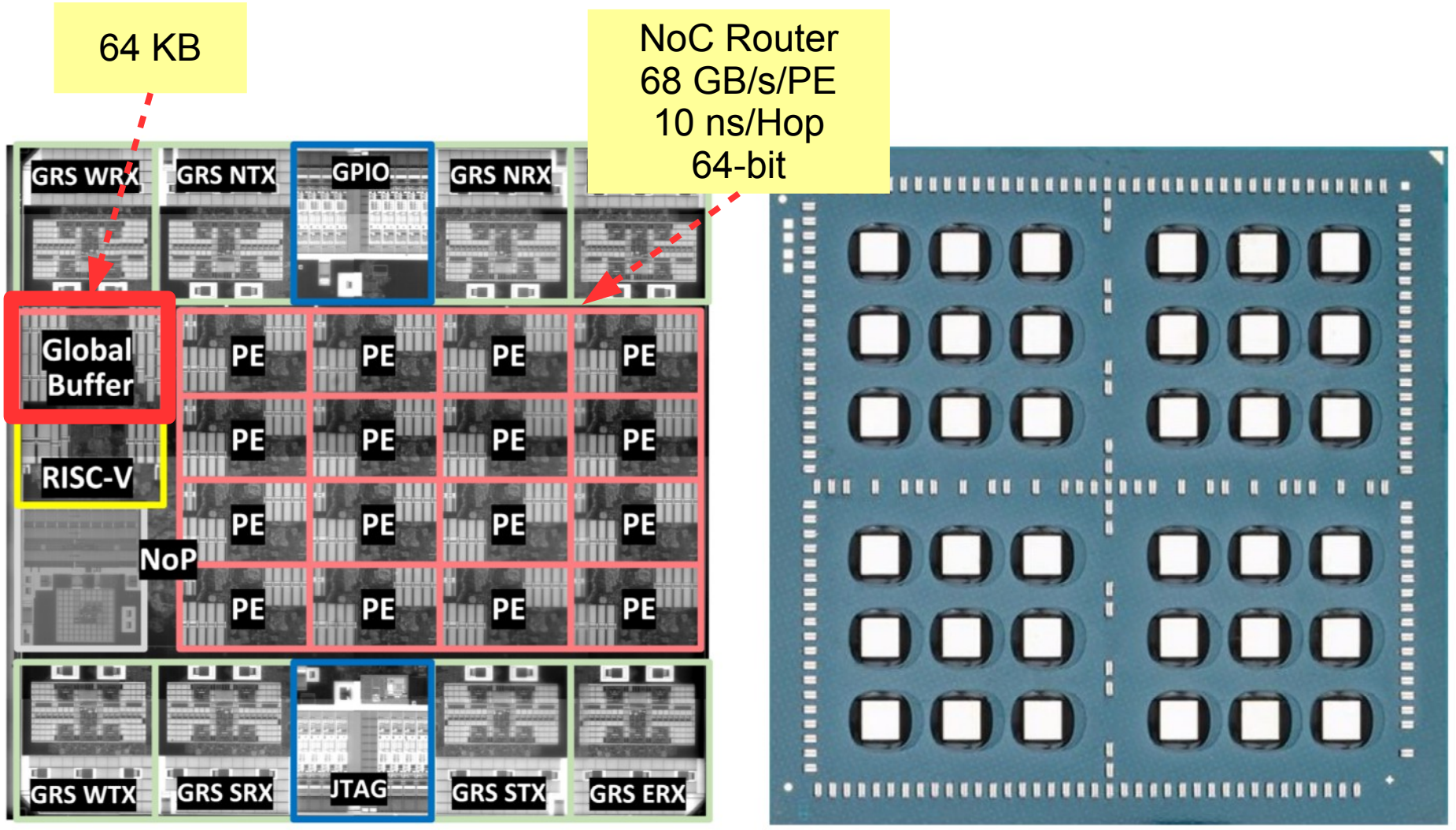
Simba Chiplet
(TSMC 16 nm FinFET)



- Ground Referenced Signaling
- 4 data lanes + 1 clock line
- 11 Gbps/pin to 25 Gbps/pin
- 0.82–1.75 pJ/bit
- 100 GB/s/chiplet

Simba Package

Simba Silicon Prototype



Simba Chiplet
(TSMC 16 nm FinFET)

Simba Package

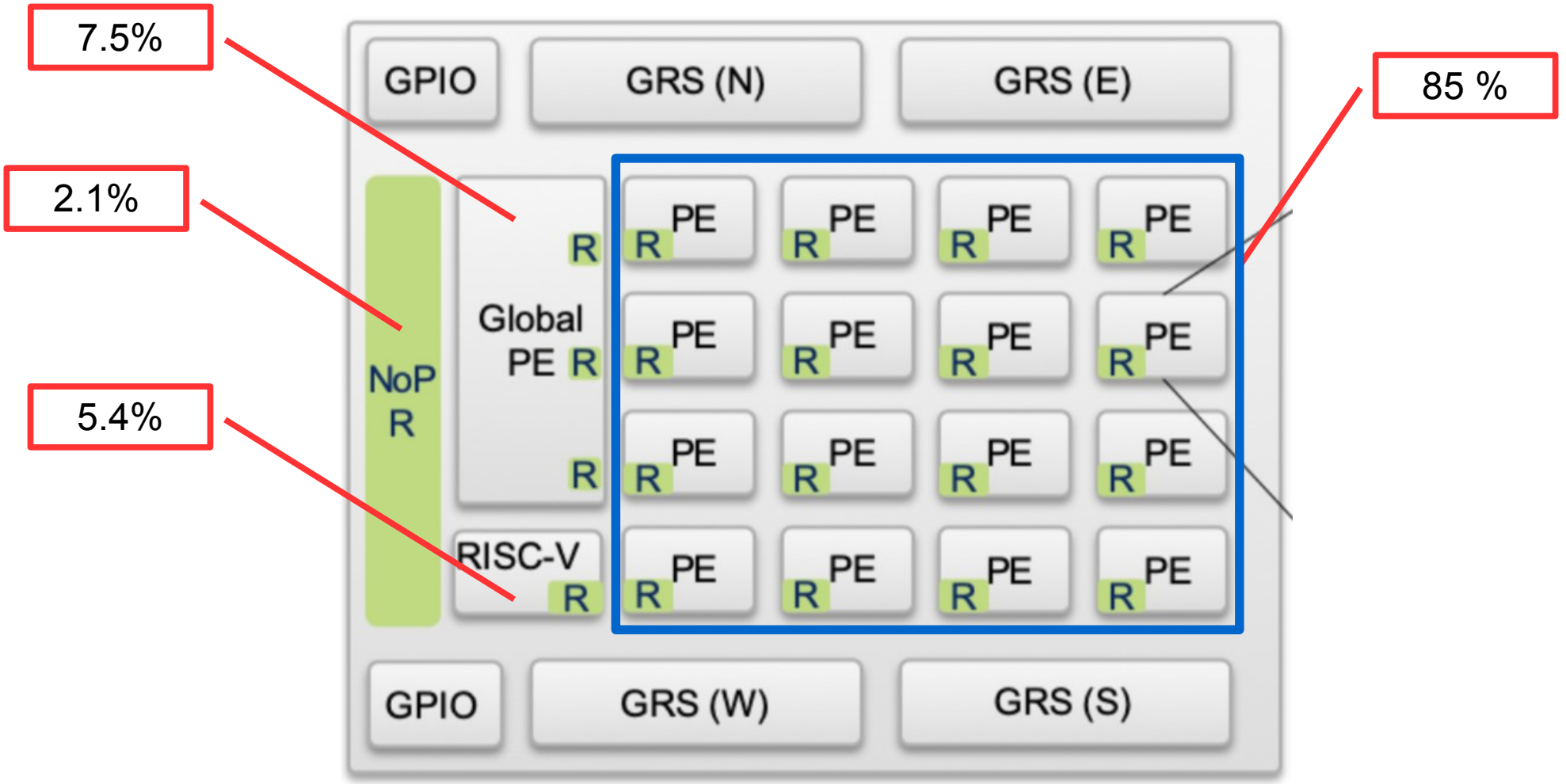
Microarchitecture Parameters

- Chiplet
  - Comparable to efficient edge DNN accelerators (*e.g.*, such as DianNao, Eyeriss)
- Simba package (36 chiplets)
  - Comparable to a data-center-scale system such as TPU

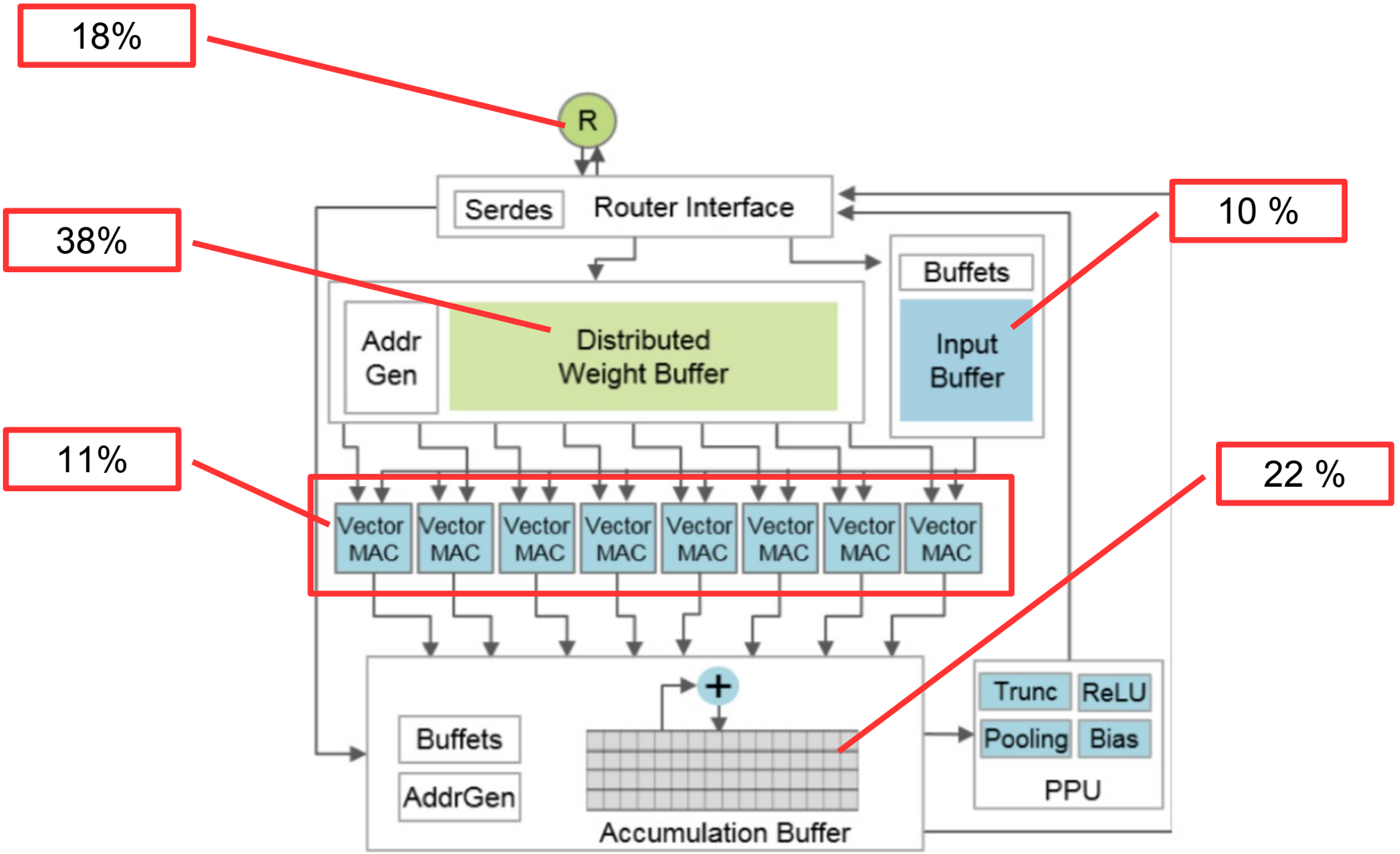
Chiplet Area Breakdown

Partition	Component	Area (μm^2)
PE	Vector MACs	12K
	Weight Buffer	41K
	Input Buffer	11K
	Accumulation Buffer	24K
	NoC Router	19K
Global PE	Distributed Buffer	125K
	NoC Routers	27K
RISC-V	Processor	109K
NoP	NoP Router	42K

Chiplet Area Breakdown



Simba Architecture - PE



Microarchitecture Parameters – Package

Number of Chiplets	36
Size	47.5 mm × 47.5 mm
Core Voltage	0.52–1.1 V
PE Clock Frequency	0.48–1.8 GHz
Chiplet-to-Chiplet Interconnect	Ground-Referenced Signaling
NoP Interconnect Bandwidth	100 GB/s/Chiplet
NoP Interconnect Latency	20 ns/Hop
NoP Interconnect Energy	0.82–1.75 pJ/bit
Number of PEs	16

Microarchitecture Parameters – Chiplet

Number of PEs	16
Area	2.5 mm × 2.4 mm
Technology	16 nm FinFET
Voltage	0.42–1.2 V
PE Clock Frequency	0.16–2.0 GHz
Global PE Buffer Size	64 KiB
Routers Per Global PE	3
NoC Interconnect Bandwidth	68 GB/s/PE
NoC Interconnect Latency	10 ns/Hop
Microcontroller	RISC-V

Microarchitecture Parameters – PE

Weight Buffer Size	32 KiB
Input Buffer Size	8 KiB
Accumulation Buffer Size	3 KiB
Vector MAC Width	8
Number of Vector MACs	8
Dataflow	Weight Stationary
Input/Weight Precision	8 bits
Partial-Sum Precision	24 bits

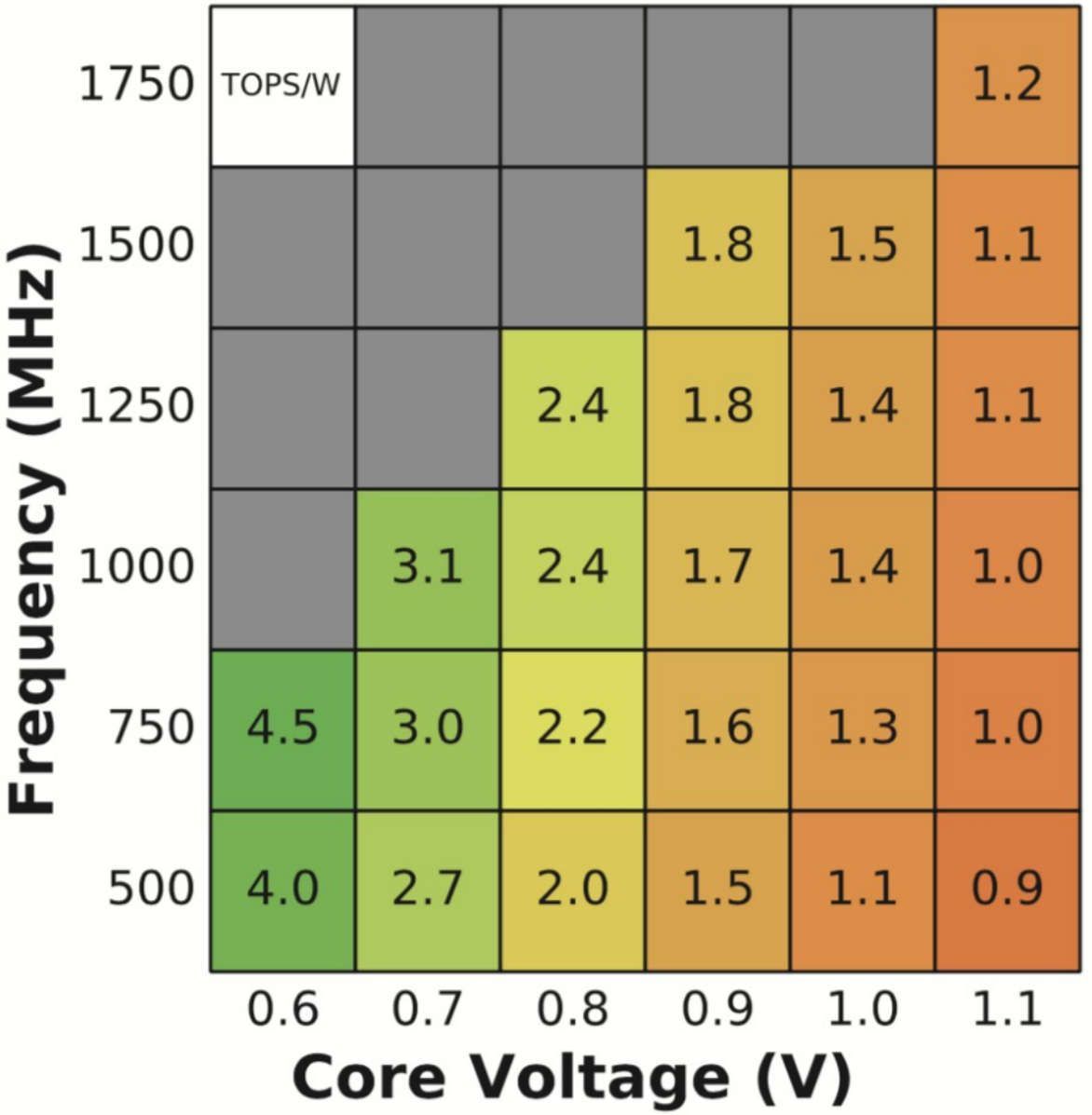
GALS Methodology

- Independent clock rates for
  - PEs,
  - Global PEs
  - RISC-V processors
  - NoP routers

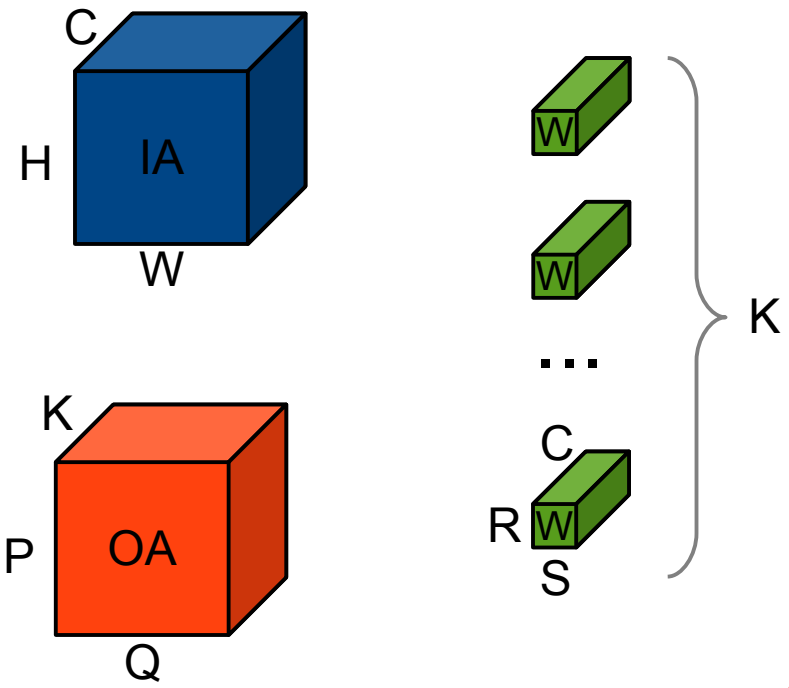
Performance

- Chiplet
  - Low power
    - 0.42 V / 161 MHz PE frequency → 0.11 pJ/Op
  - High performance
    - 1.2 V / 2 Ghz → 4 TOPS
- 36-Chiplet System
  - Low power
    - 0.52 V / 484 MHz PE frequency → 0.16 pJ/Op
  - High performance
    - 1.1 V / 1.8 GHz PE frequenccy → 128 TOPS

System Frequency *vs.* Voltage



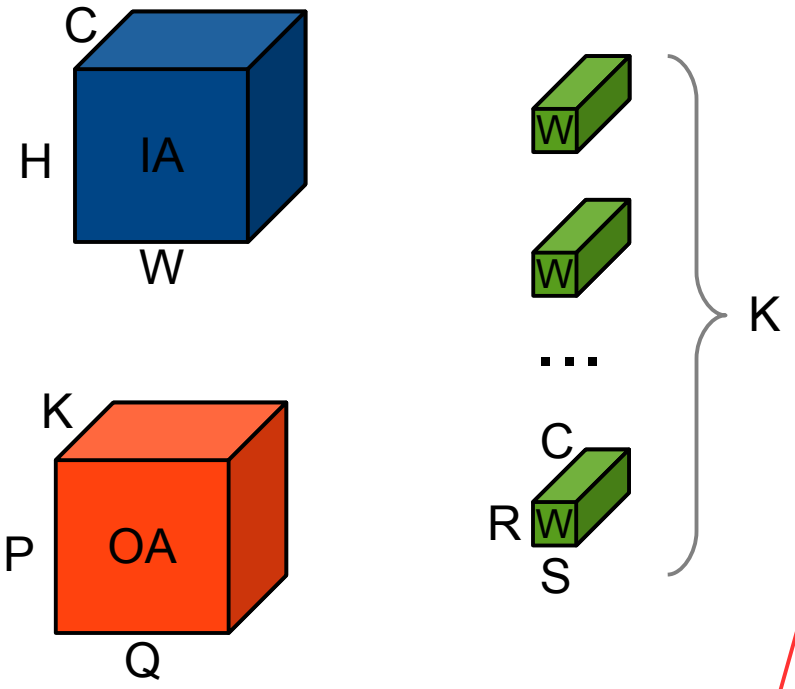
Baseline Tiling



Weights uniformly partitioned along input channels (C) and outputs channels (K) dimensions

```
1   //Package level
2   for p3 = [0 : P3) :
3     for q3 = [0 : Q3) :
4       parallel_for k3 = [0 : K3) :
5         parallel_for c3 = [0 : C3) :
6   // Chiplet level
7   for p2 = [0 : P2) :
8     for q2 = [0 : Q2) :
9       parallel_for k2 = [0 : K2) :
10        parallel_for c2 = [0 : C2) :
11  // PE level
12  for r = [0 : R) :
13    for s = [0 : S) :
14      for k1 = [0 : K1) :
15        for c1 = [0 : C1) :
16          for p1 = [0 : P1) :
17            for q1 = [0 : Q1) :
18  // Vector-MAC level
19  parallel_for k0 = [0 : K0) :
20    parallel_for c0 = [0 : C0) :
21      p = (p3 * P2 + p2) * P1 + p1;
22      q = (q3 * Q2 + q2) * Q1 + q1;
23      k = ((k3 * K2 + k2) * K1 + k1) * K0 + k0;
24      c = ((c3 * C2 + c2) * C1 + c1) * C0 + c0;
25      OA[p,q,k] += IA[p-1+r,q-1+s,c] * W[r,s,c,k];
```

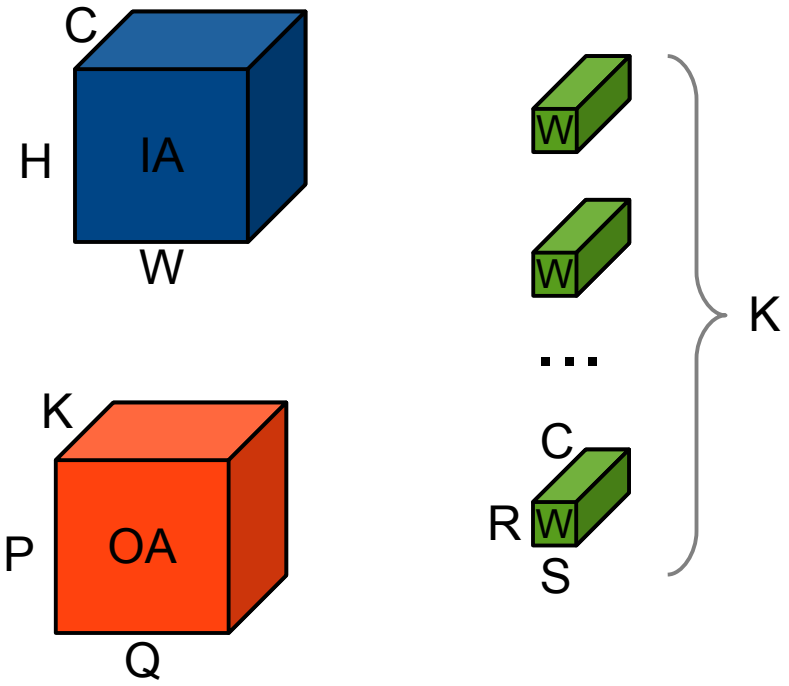
Baseline Tiling



Uniformly partition along the height (P) and width (Q) dimensions of an output activation across chiplets and PEs

```
1   //Package level
2   for p3 = [0 : P3) :
3     for q3 = [0 : Q3) :
4       parallel_for k3 = [0 : K3) :
5         parallel_for c3 = [0 : C3) :
6   // Chiplet level
7   for p2 = [0 : P2) :
8     for q2 = [0 : Q2) :
9       parallel_for k2 = [0 : K2) :
10        parallel_for c2 = [0 : C2) :
11  // PE level
12  for r = [0 : R) :
13    for s = [0 : S) :
14      for k1 = [0 : K1) :
15        for c1 = [0 : C1) :
16          for p1 = [0 : P1) :
17            for q1 = [0 : Q1) :
18  // Vector-MAC level
19  parallel_for k0 = [0 : K0) :
20    parallel_for c0 = [0 : C0) :
21      p = (p3 * P2 + p2) * P1 + p1;
22      q = (q3 * Q2 + q2) * Q1 + q1;
23      k = ((k3 * K2 + k2) * K1 + k1) * K0 + k0;
24      c = ((c3 * C2 + c2) * C1 + c1) * C0 + c0;
25      OA[p,q,k] += IA[p-1+r,q-1+s,c] * W[r,s,c,k];
```

Baseline Tiling

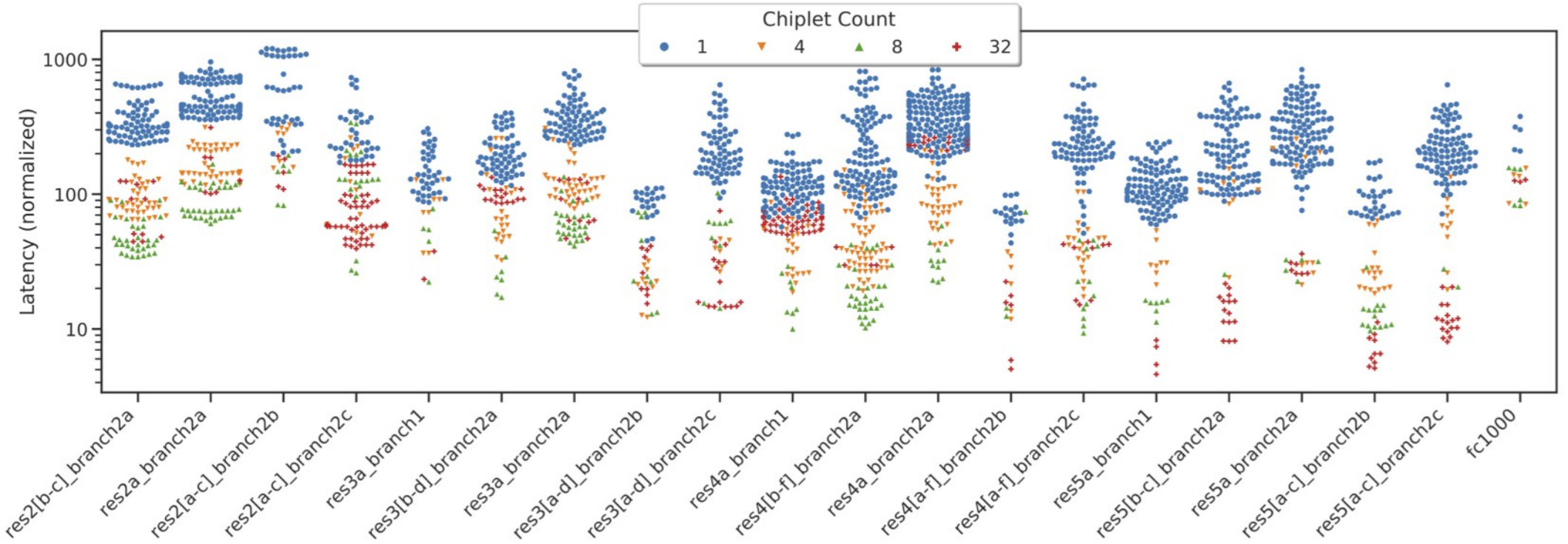


The loop bounds and orderings are configurable

```
1   //Package level
2   for p3 = [0 : P3) :
3     for q3 = [0 : Q3) :
4       parallel_for k3 = [0 : K3) :
5         parallel_for c3 = [0 : C3) :
6   // Chiplet level
7   for p2 = [0 : P2) :
8     for q2 = [0 : Q2) :
9       parallel_for k2 = [0 : K2) :
10        parallel_for c2 = [0 : C2) :
11  // PE level
12  for r = [0 : R) :
13    for s = [0 : S) :
14      for k1 = [0 : K1) :
15        for c1 = [0 : C1) :
16          for p1 = [0 : P1) :
17            for q1 = [0 : Q1) :
18  // Vector-MAC level
19  parallel_for k0 = [0 : K0) :
20    parallel_for c0 = [0 : C0) :
21      p = (p3 * P2 + p2) * P1 + p1;
22      q = (q3 * Q2 + q2) * Q1 + q1;
23      k = ((k3 * K2 + k2) * K1 + k1) * K0 + k0;
24      c = ((c3 * C2 + c2) * C1 + c1) * C0 + c0;
25      OA[p,q,k] += IA[p-1+r,q-1+s,c] * W[r,s,c,k];
```

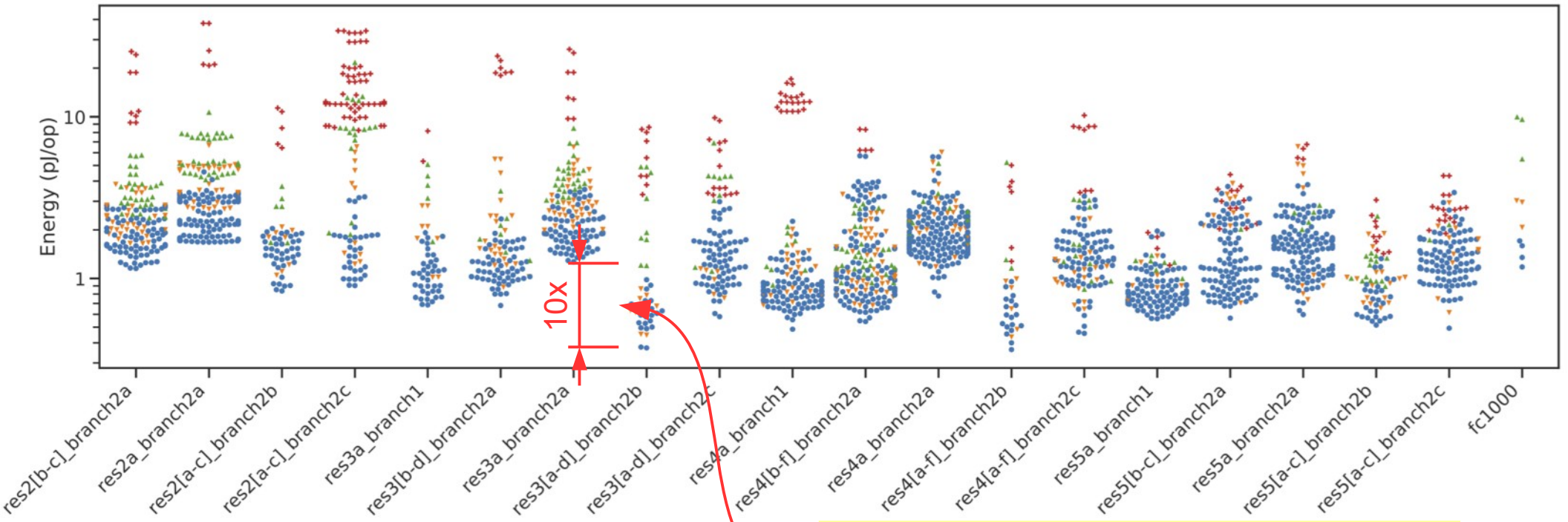
Latency – ResNet-50

Normalized to ideal latency realized if each of the 576 PEs of the system operated with 100% utilization and no communication or synchronization overheads



Energy – ResNet-50

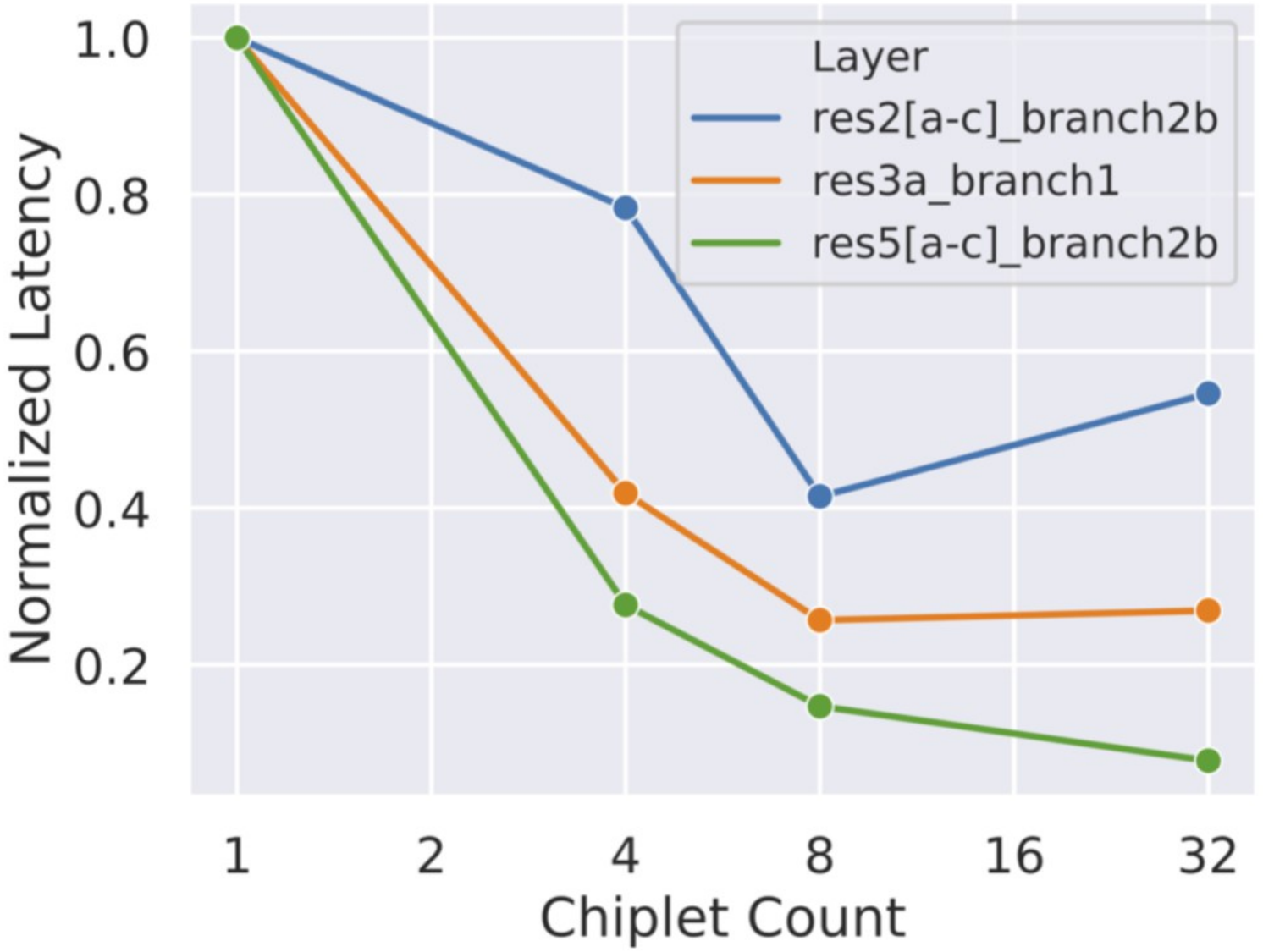
Normalized to ideal energy when the system operated with 100% utilization and no communication or synchronization overheads



32 chiplet confs high energy cost due to chiplet-to-chiplet communication and synchronization

Layers with high reuse factor tend to perform more efficiently than layers that require more data movement.

Performance Scalability

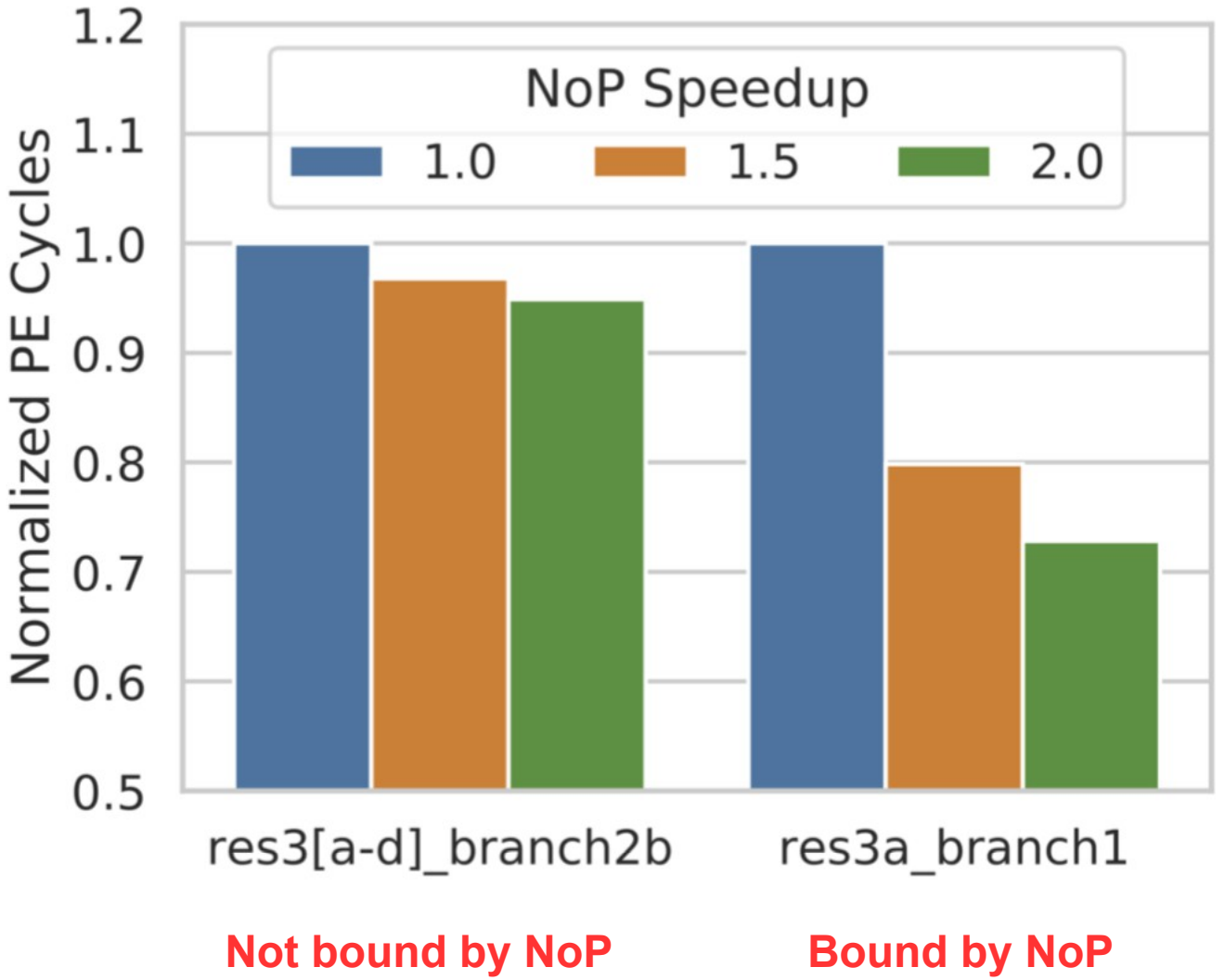


17 layers

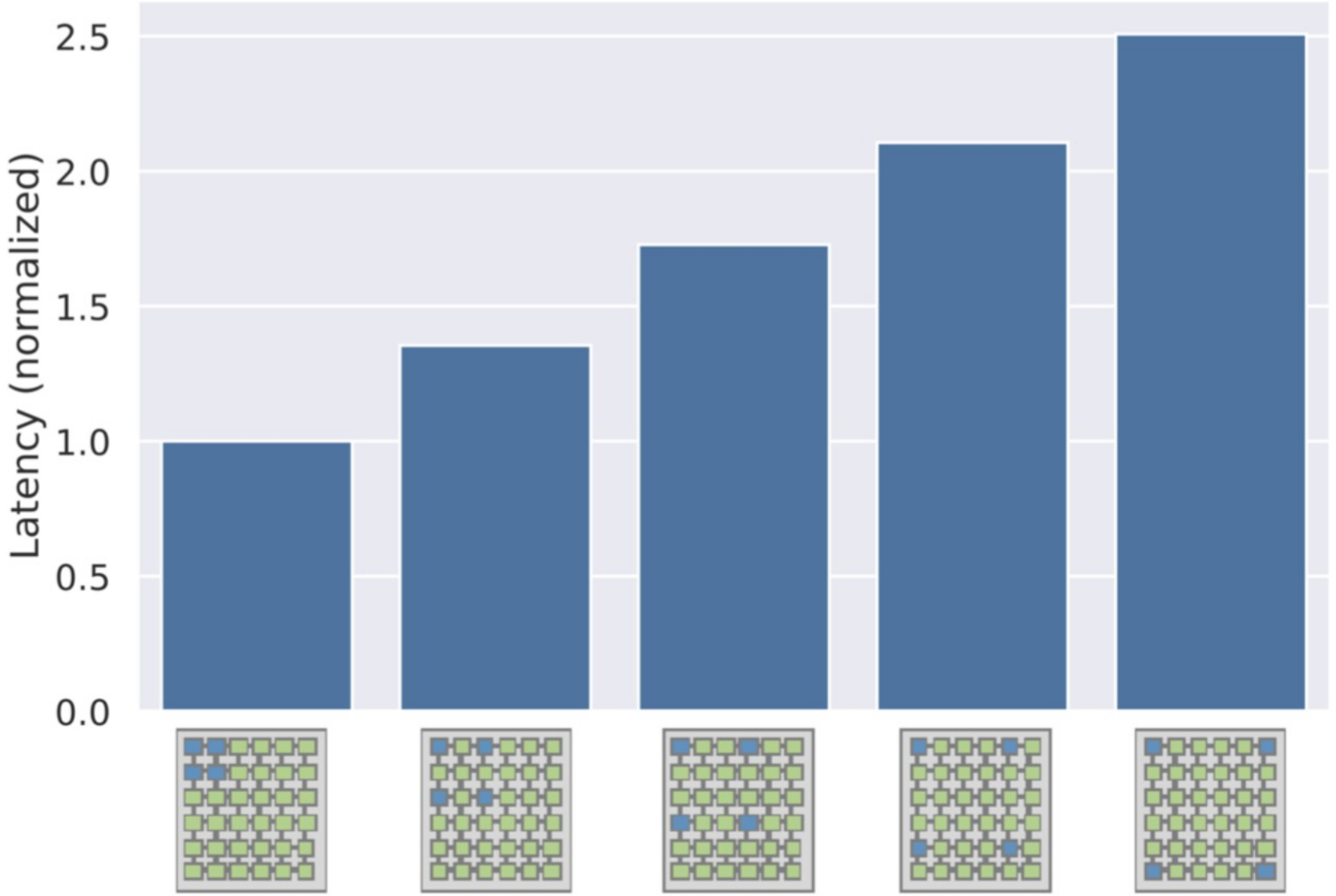
12 layers

24 layers

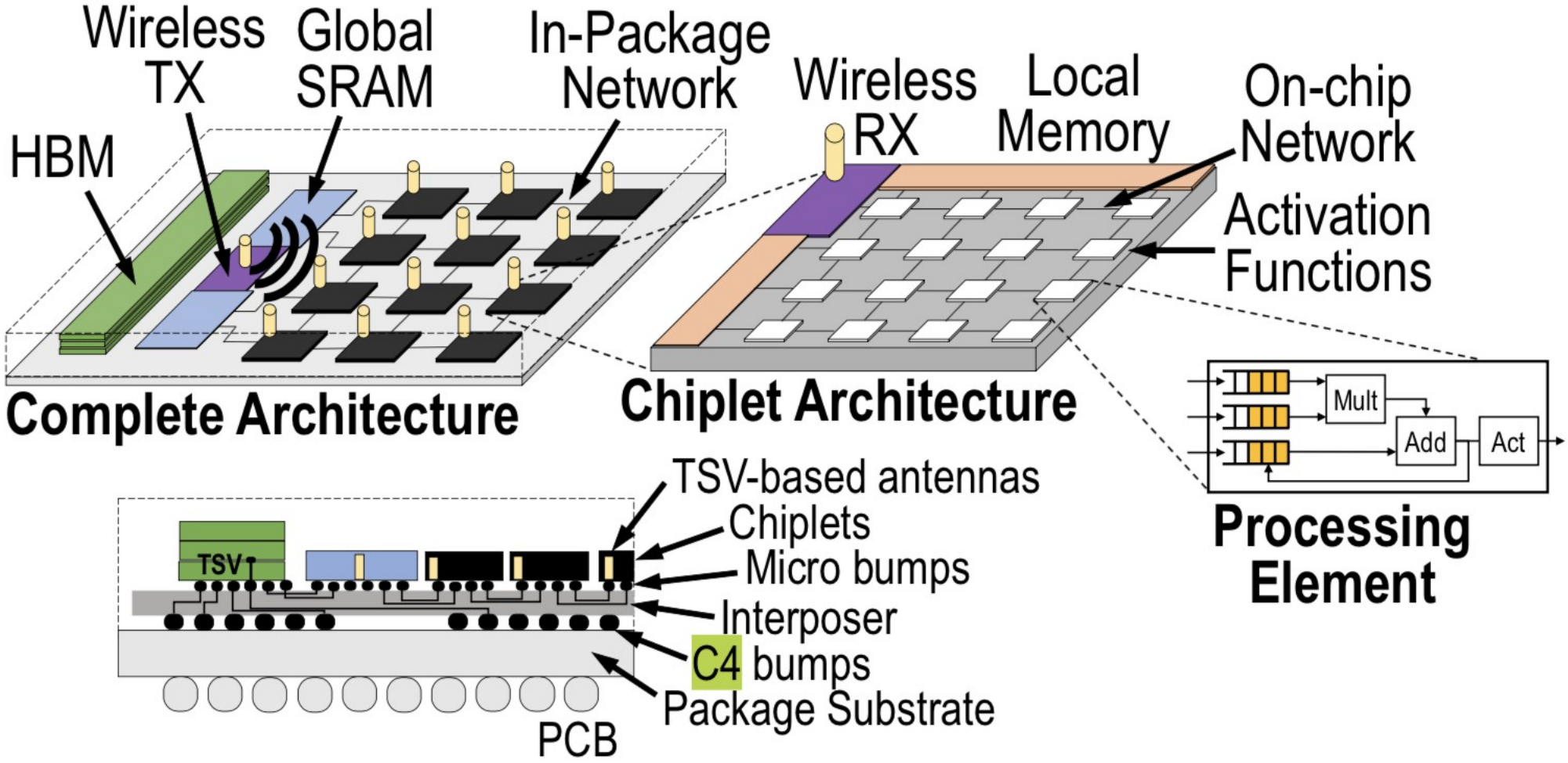
NoP Bandwidth Sensitivity



NoP Latency Sensitivity



Wireless Enabled Inter-Chiplet Communication



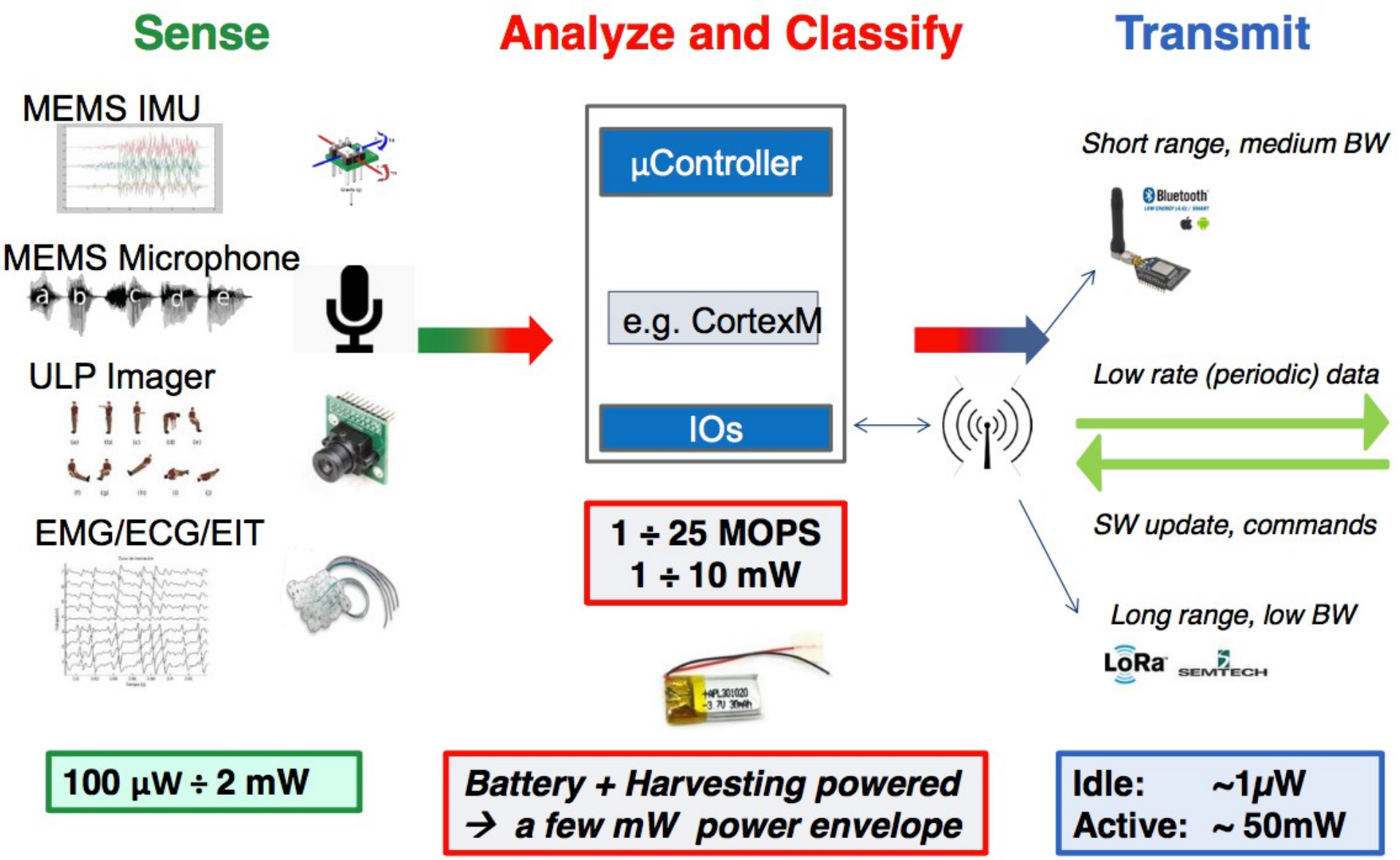
[Guirado, *et al.*, Dataflow-Architecture Co-Design for 2.5D DNN Accelerators using Wireless Network-on-Package. ASPDAC '21]

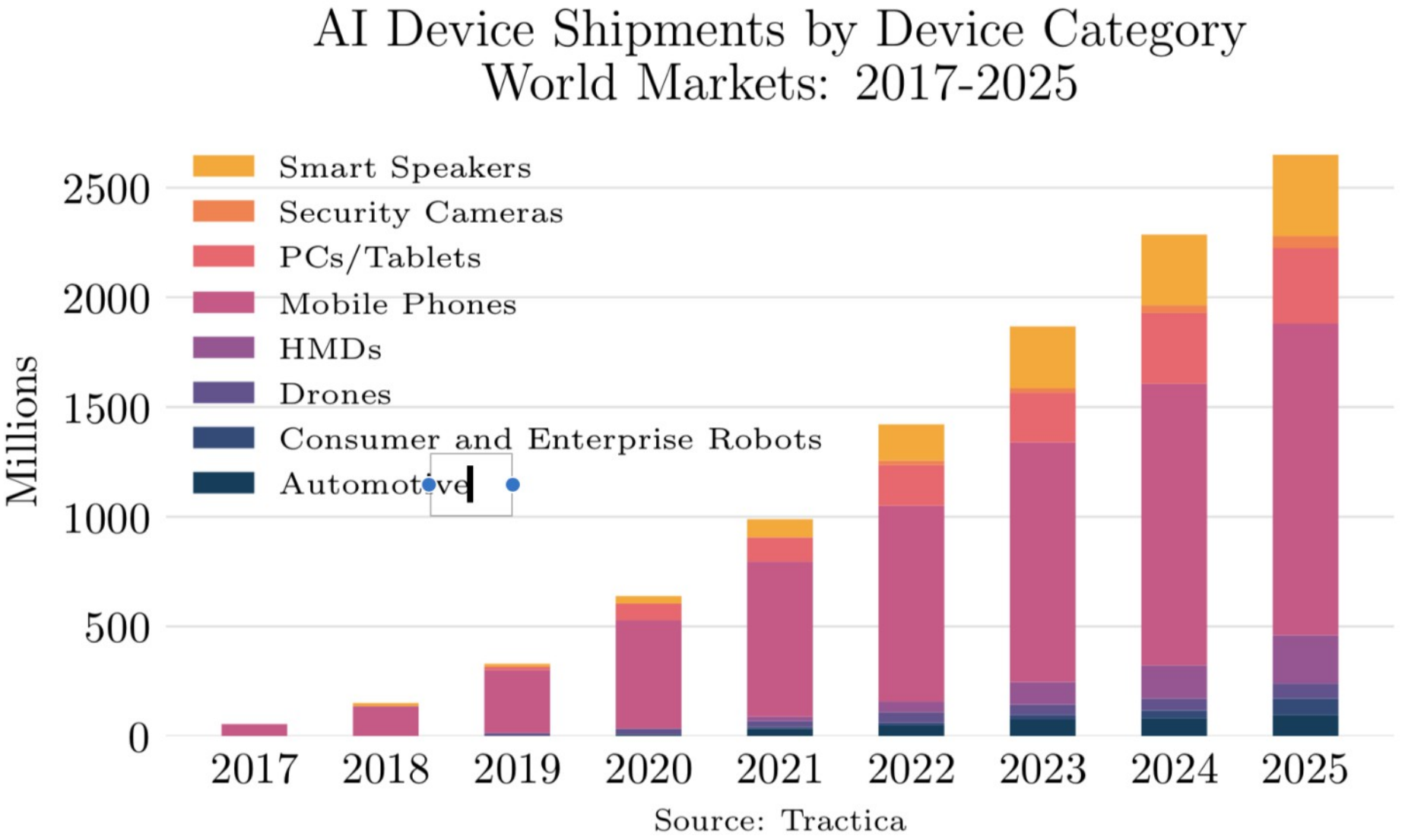
[Ascia, *et al.*, Improving Inference Latency and Energy of DNNs through Wireless Enabled Multi-Chip-Module-based Architectures and Model Parameters Compression. NOCS'2020]

Agenda

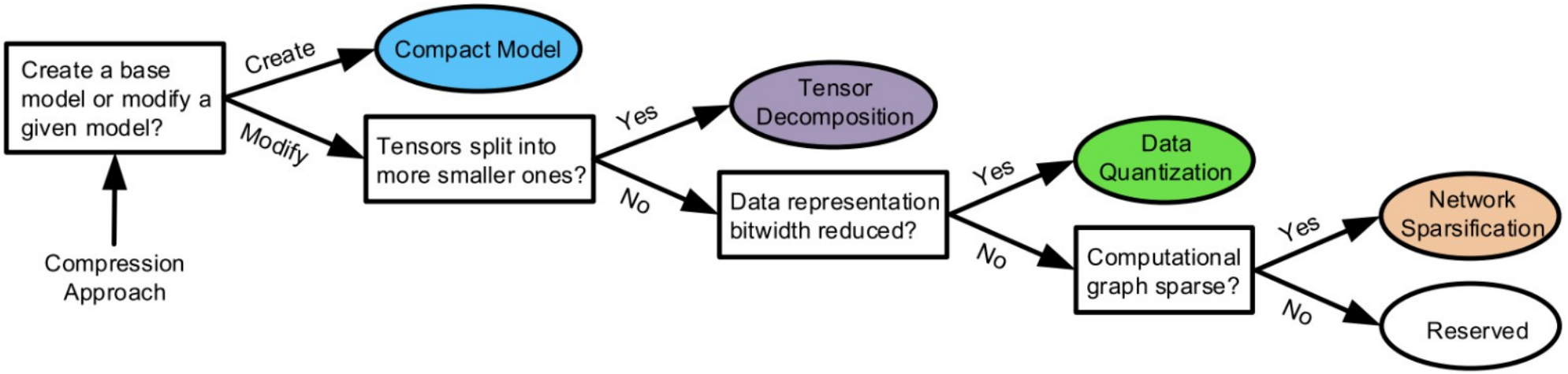
- Simba MCM DNN HW Inference Accelerator
- Improving Energy/Performance with Compression

Computing for the IoT



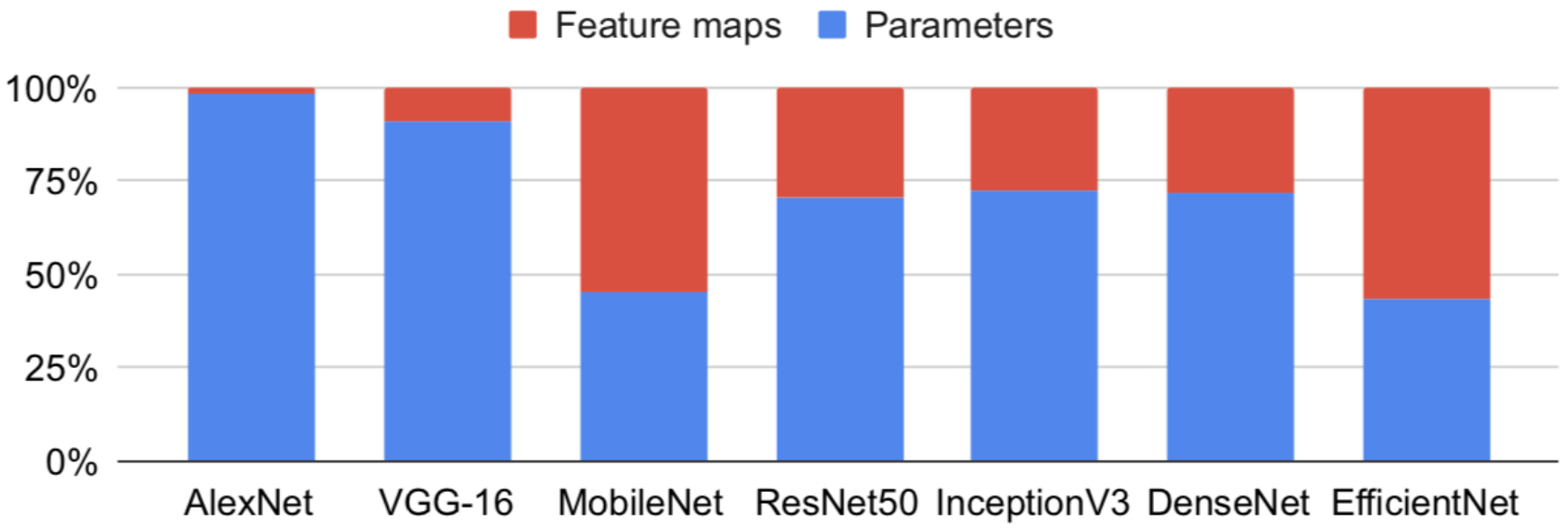

AI Device Shipments by Device Category
World Markets: 2017-2025
Smart Speakers
Security Cameras
PCs/Tablets
Mobile Phones
HMDs
Drones
Consumer and Enterprise Robots
Automotive
Millions
2500
2000
1500
1000
500
0
2017
2018
2019
2020
2021
2022
2023
2024
2025
Source: Tractica

Model Compression Techniques



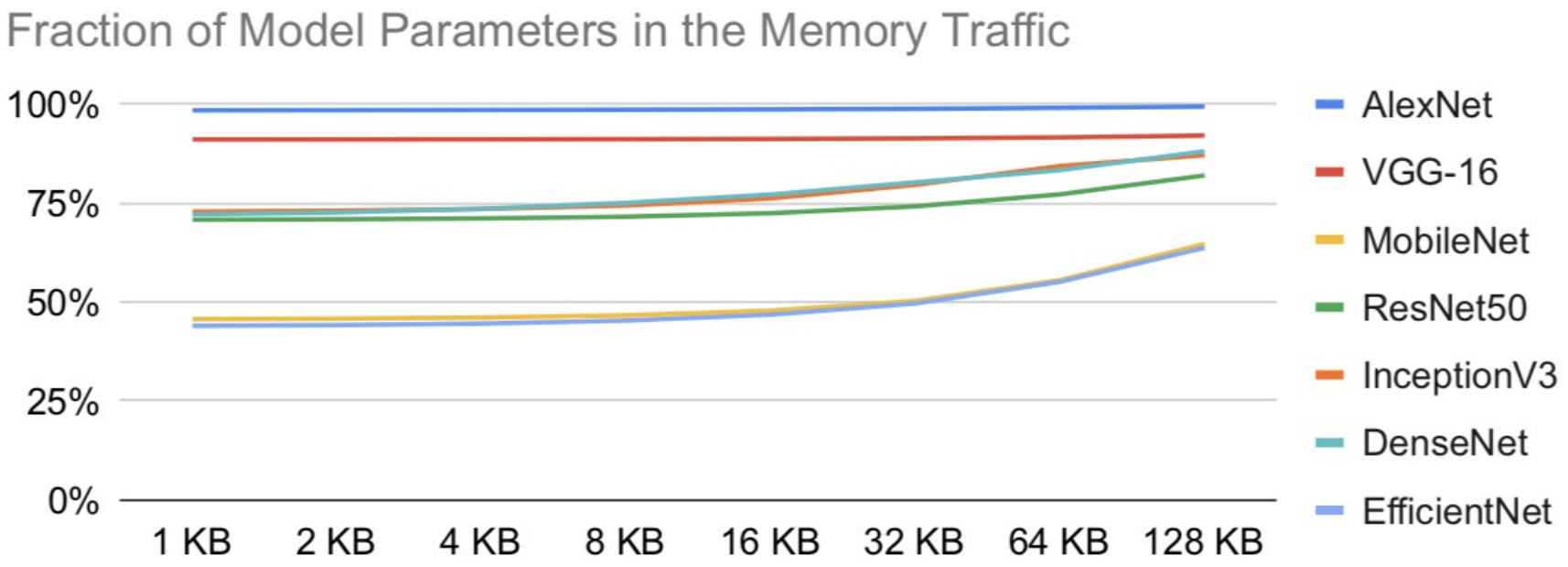
[L. Deng, *et al.*, "Model compression and hardware acceleration for neural networks: A comprehensive survey," Proceedings of the IEEE, 2020]

Memory Traffic Components



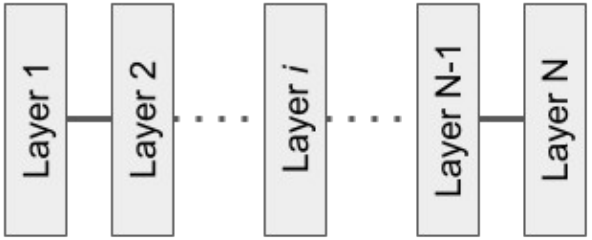
Traffic due to model parameters (filters/weighs) is dominant

Memory Traffic Components



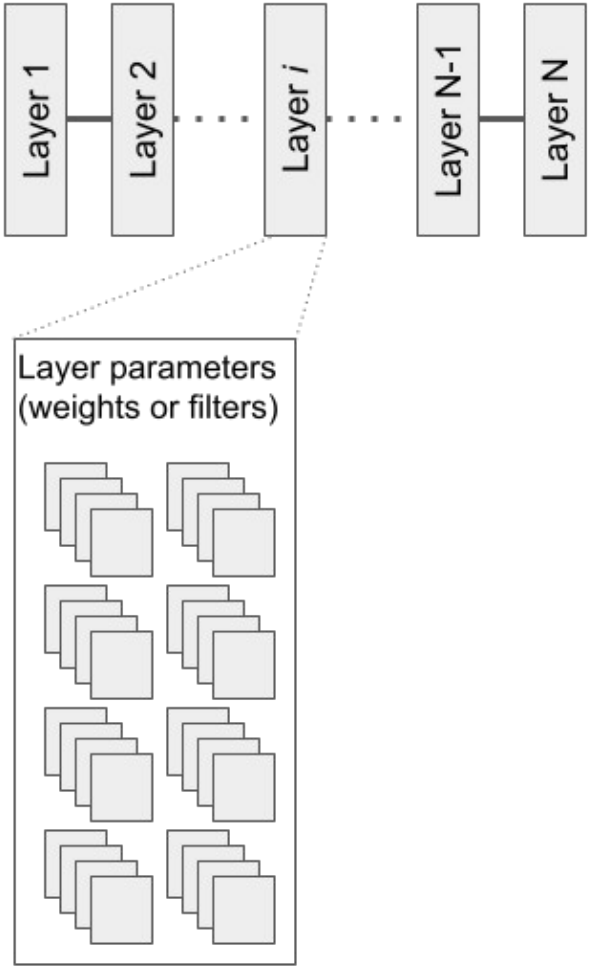
Fraction of traffic due to model paeameters increases with global buffer size

SortCompress – Compression



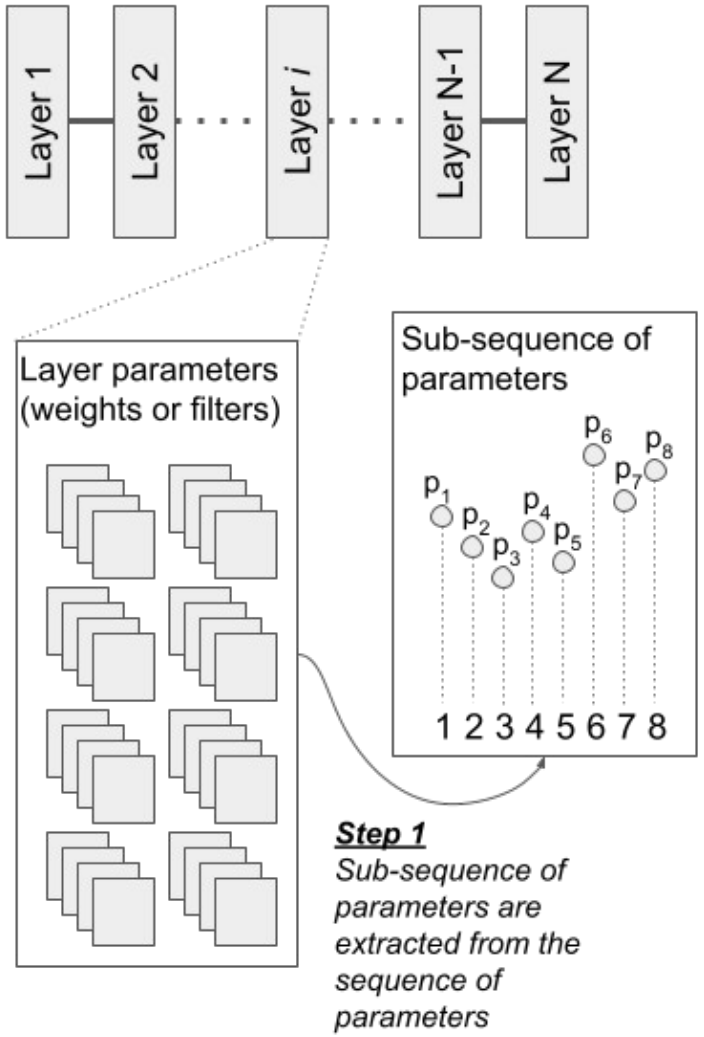
Compression
(*offline*)

SortCompress – Compression



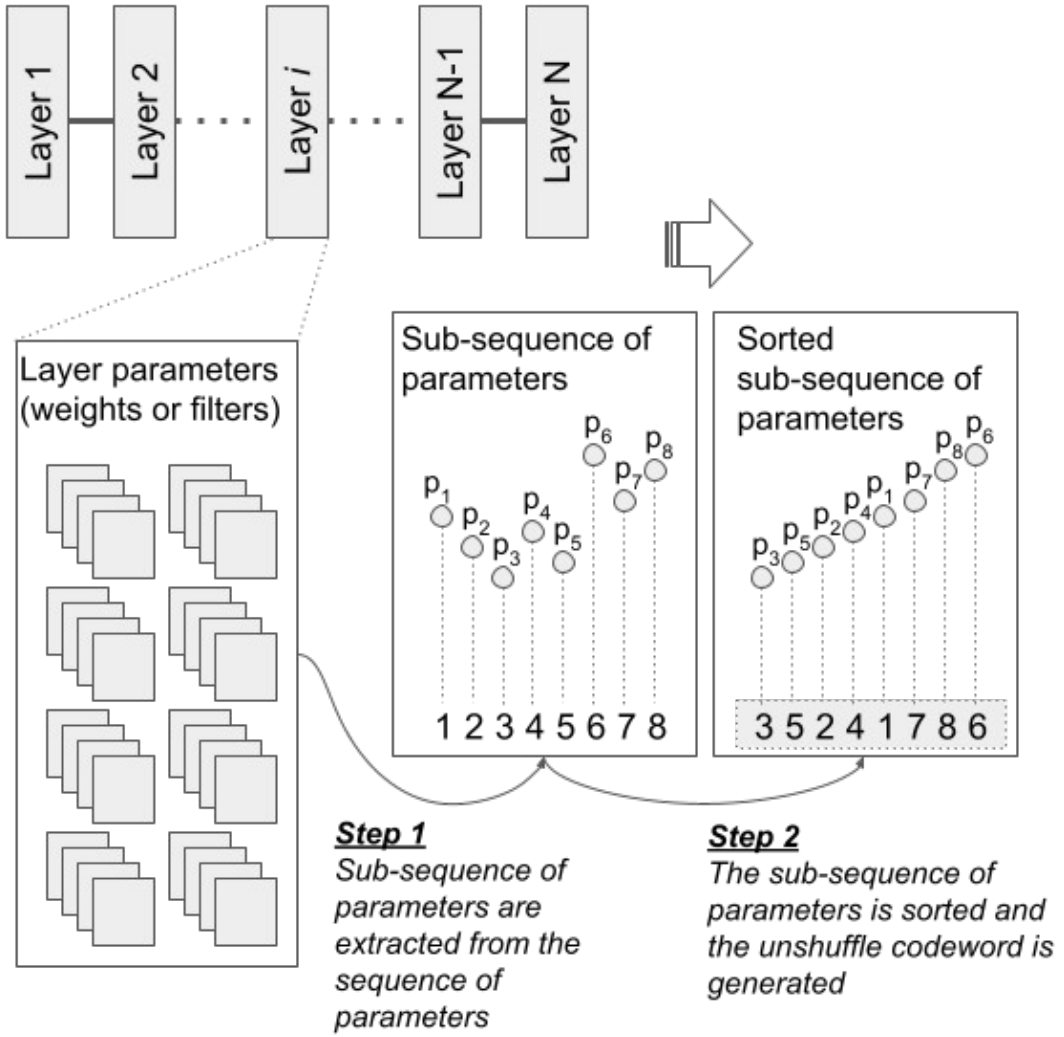
Compression
(*offline*)

SortCompress – Compression



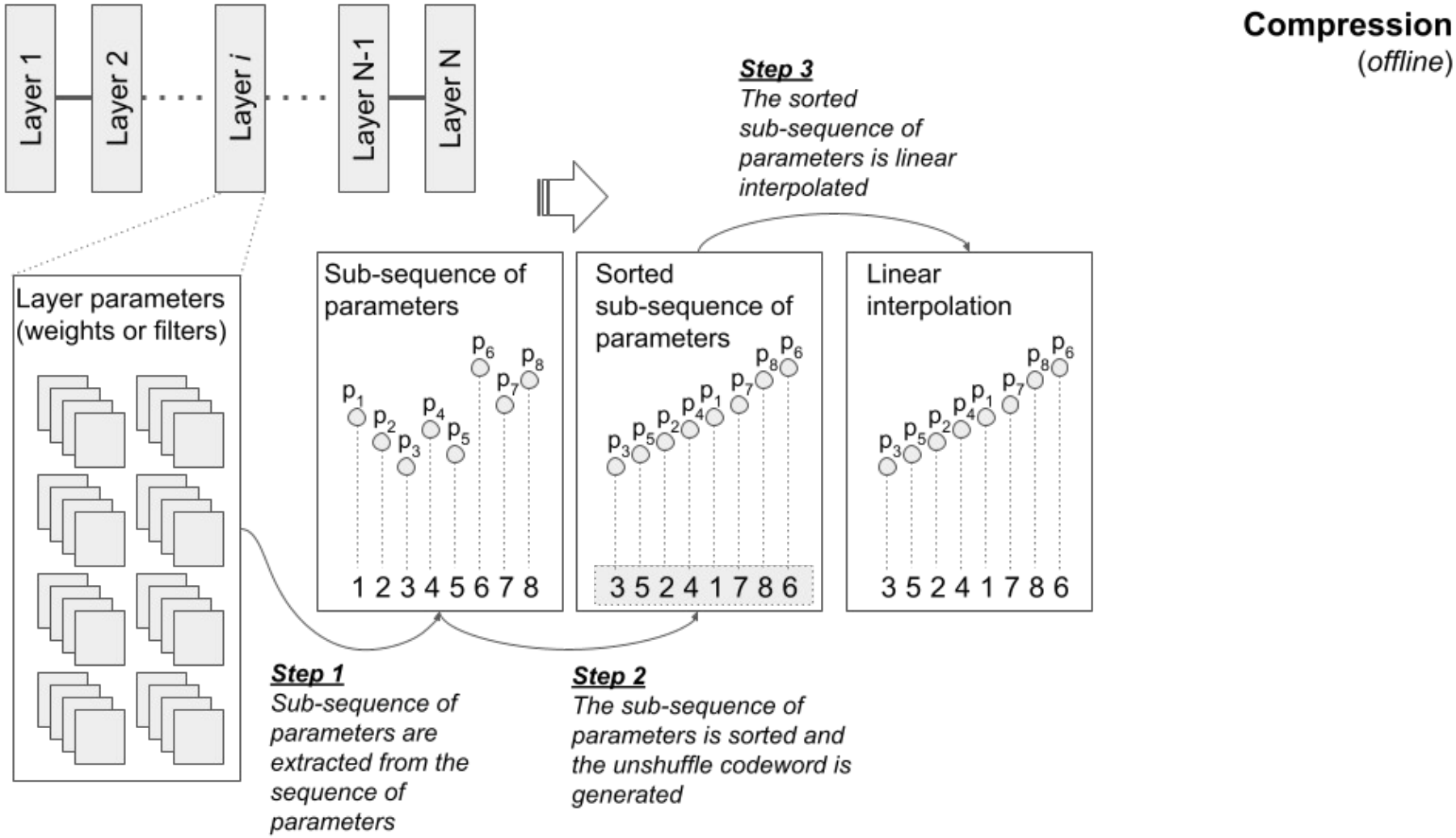
Compression
(*offline*)

SortCompress – Compression

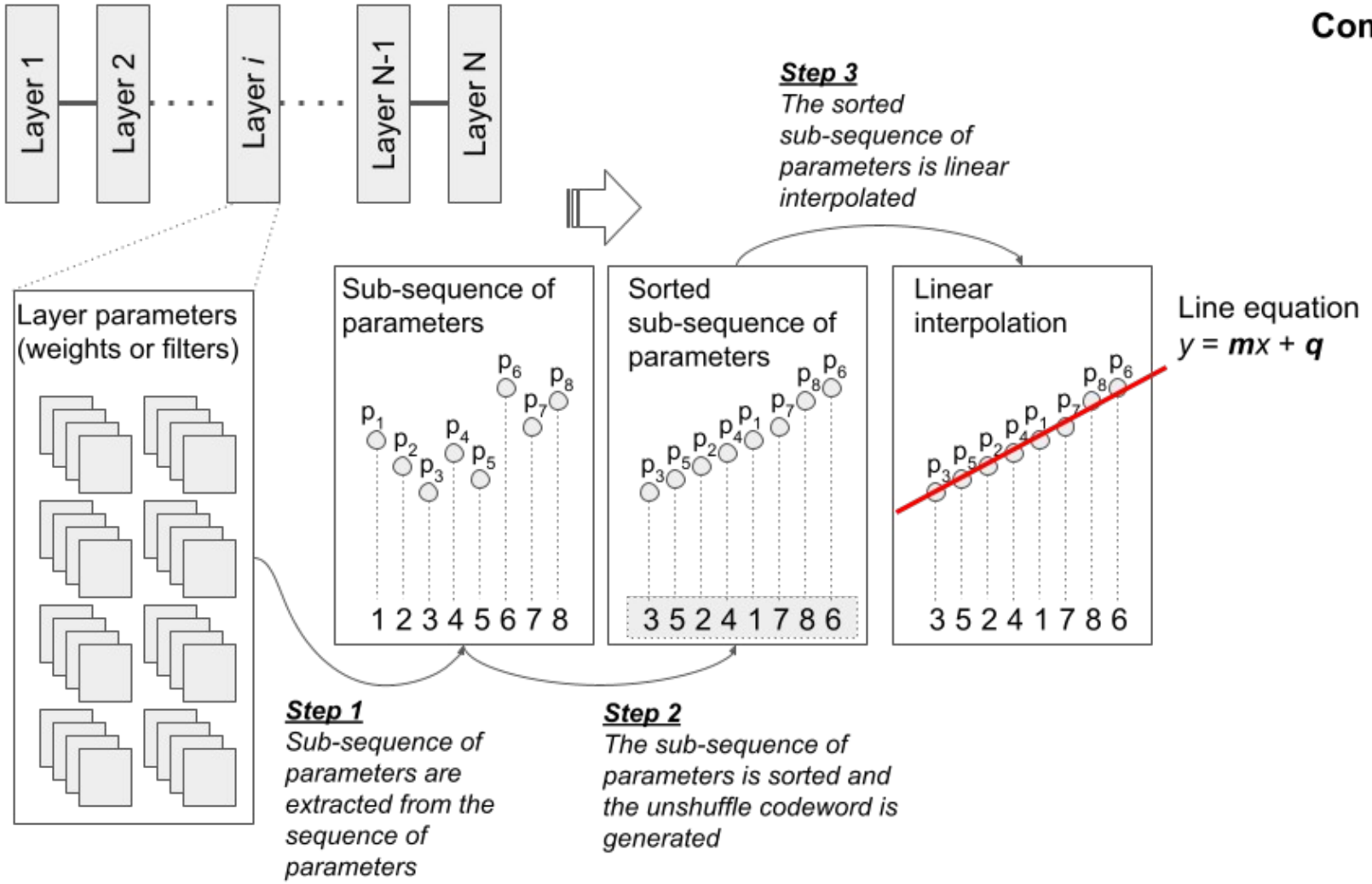


Compression
(*offline*)

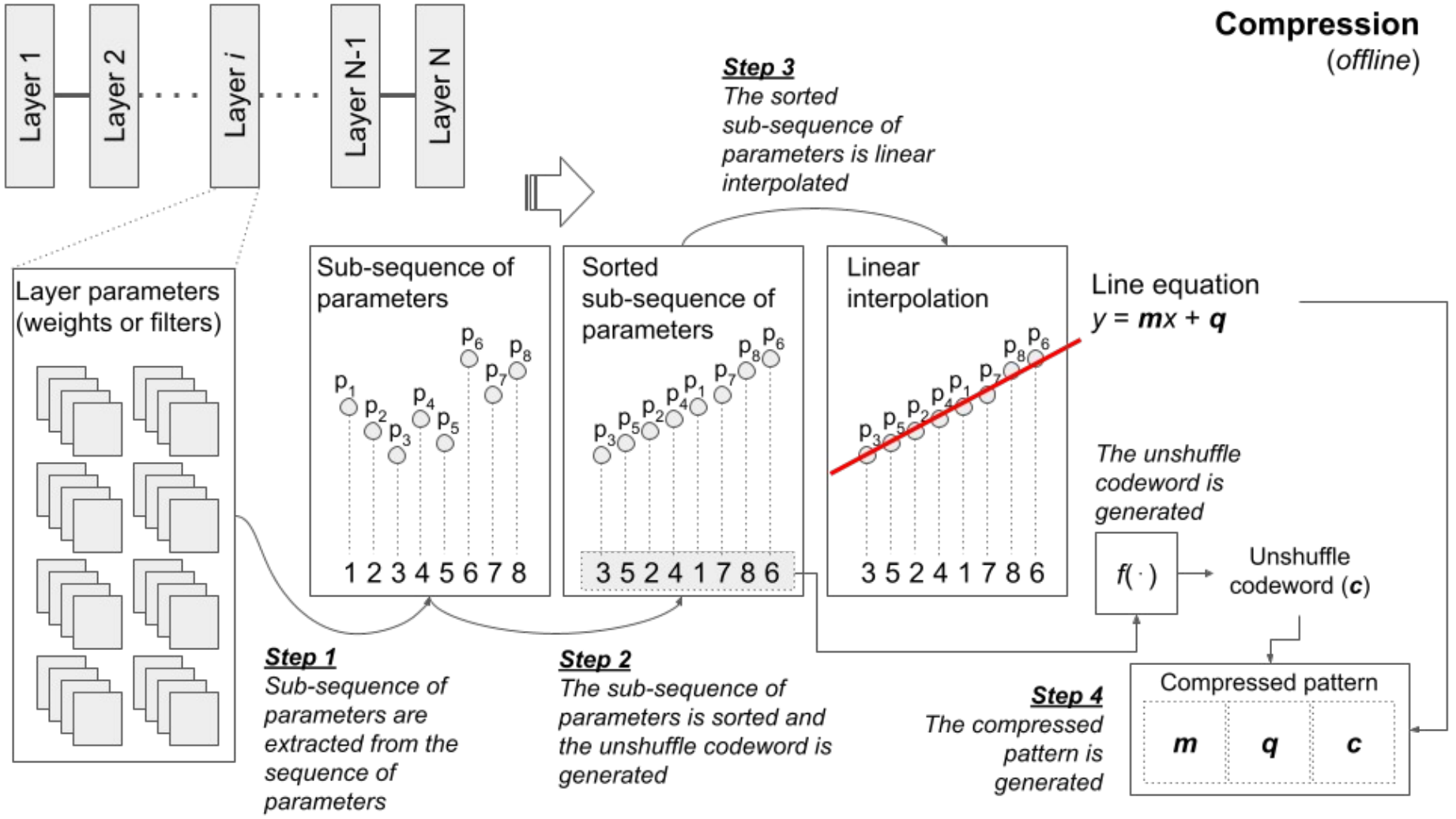
SortCompress – Compression



SortCompress – Compression

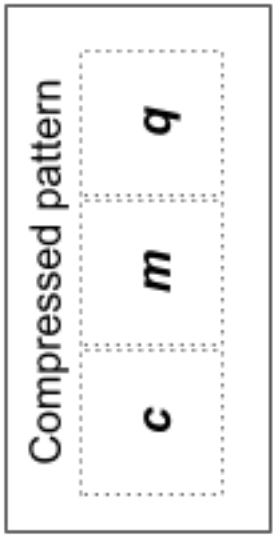


SortCompress – Compression



SortCompress – Decompression

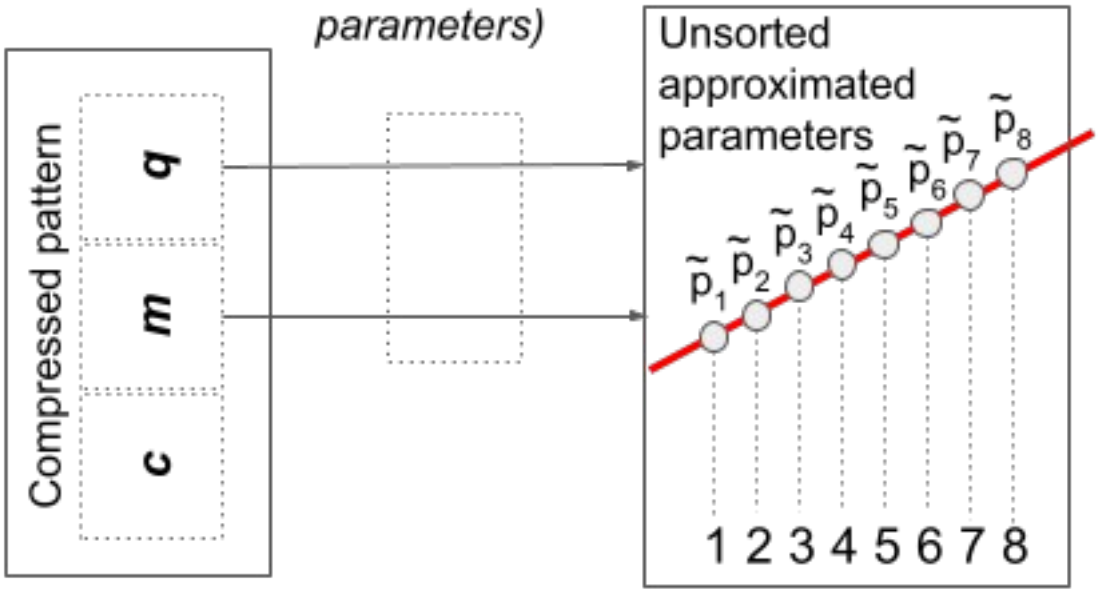
Decompression
(*online*)



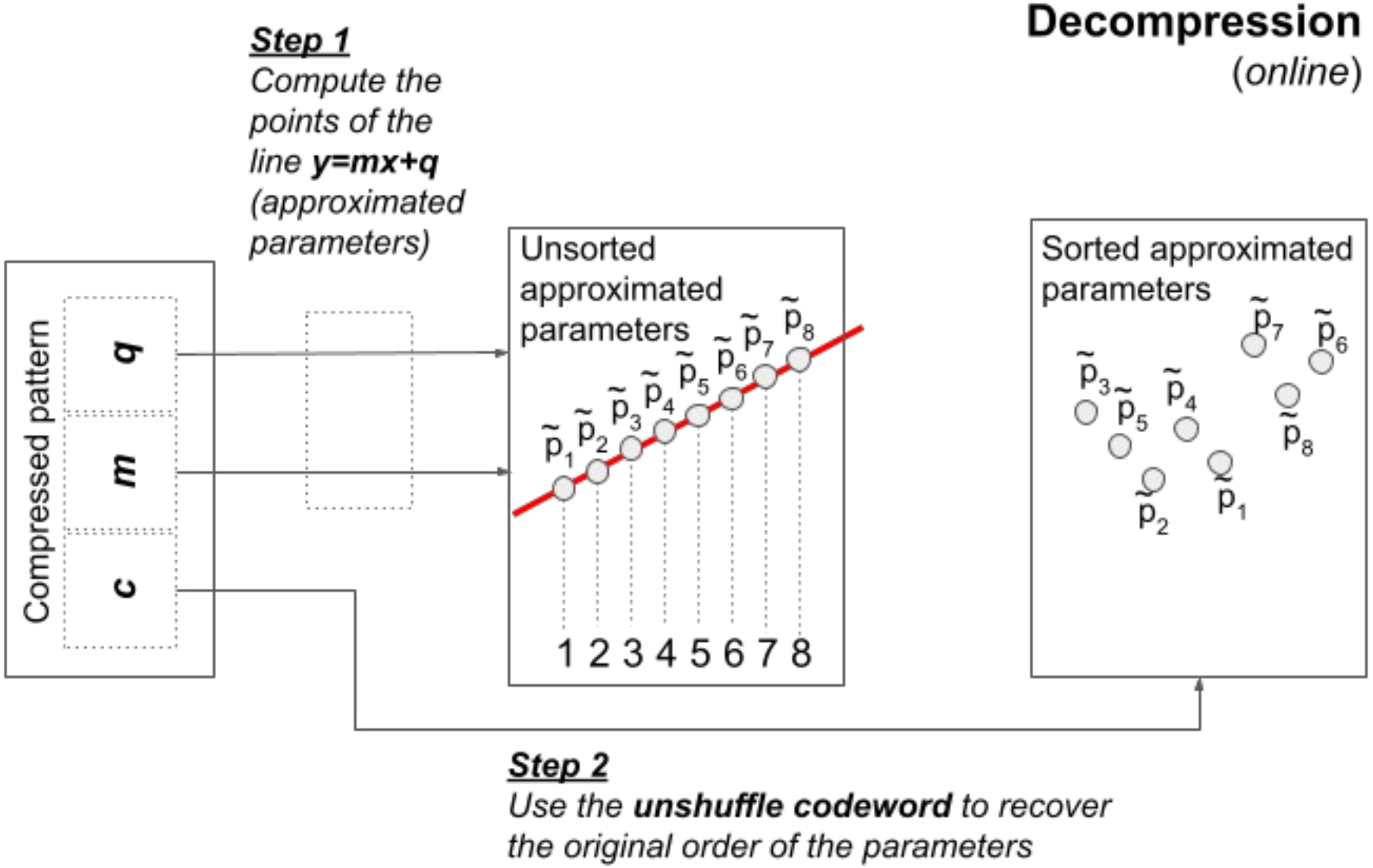
SortCompress – Decompression

Decompression
(*online*)

Step 1
Compute the points of the line ***y=mx+q*** *(approximated parameters)*

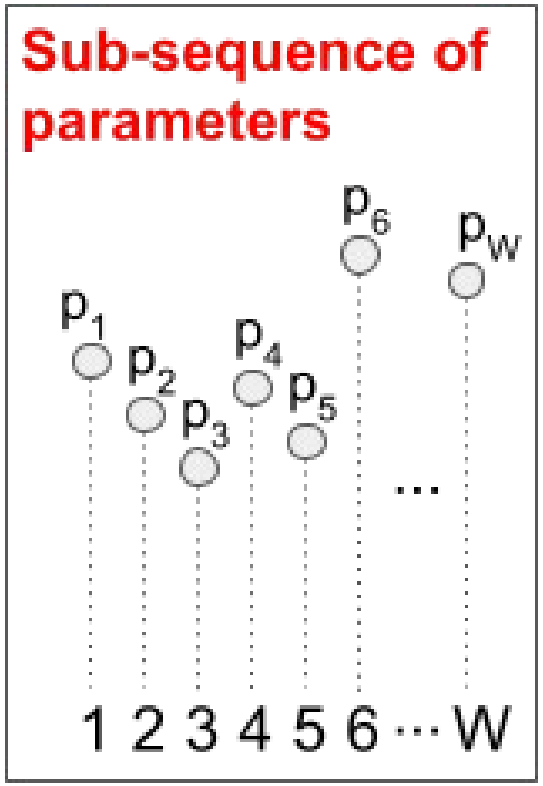


SortCompress – Decompression

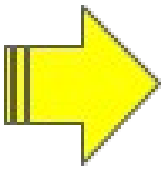


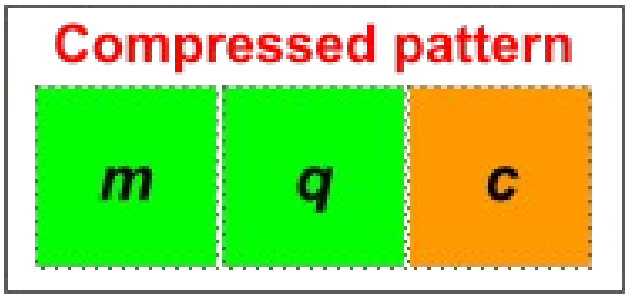
Compression Ratio

$$CR = \frac{MF_{uncomp}}{MF_{comp}}$$



$$MF_{uncomp} = W \times B$$

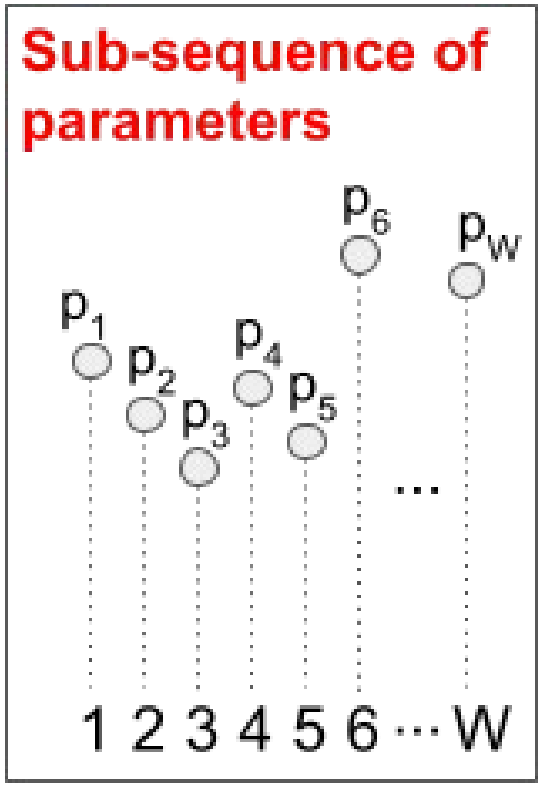




$$MF_{comp} = 2\,\tilde{B} + SizeOf\,(c)$$

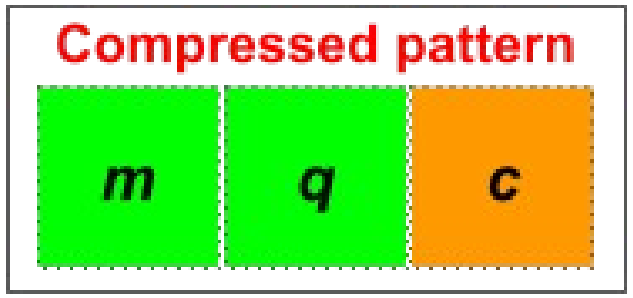
Compression Ratio

By expressing the unshuffle codeword as **factoradic**, and interpreted as a **Lehmer code**



$$MF_{uncomp} = W \times B$$

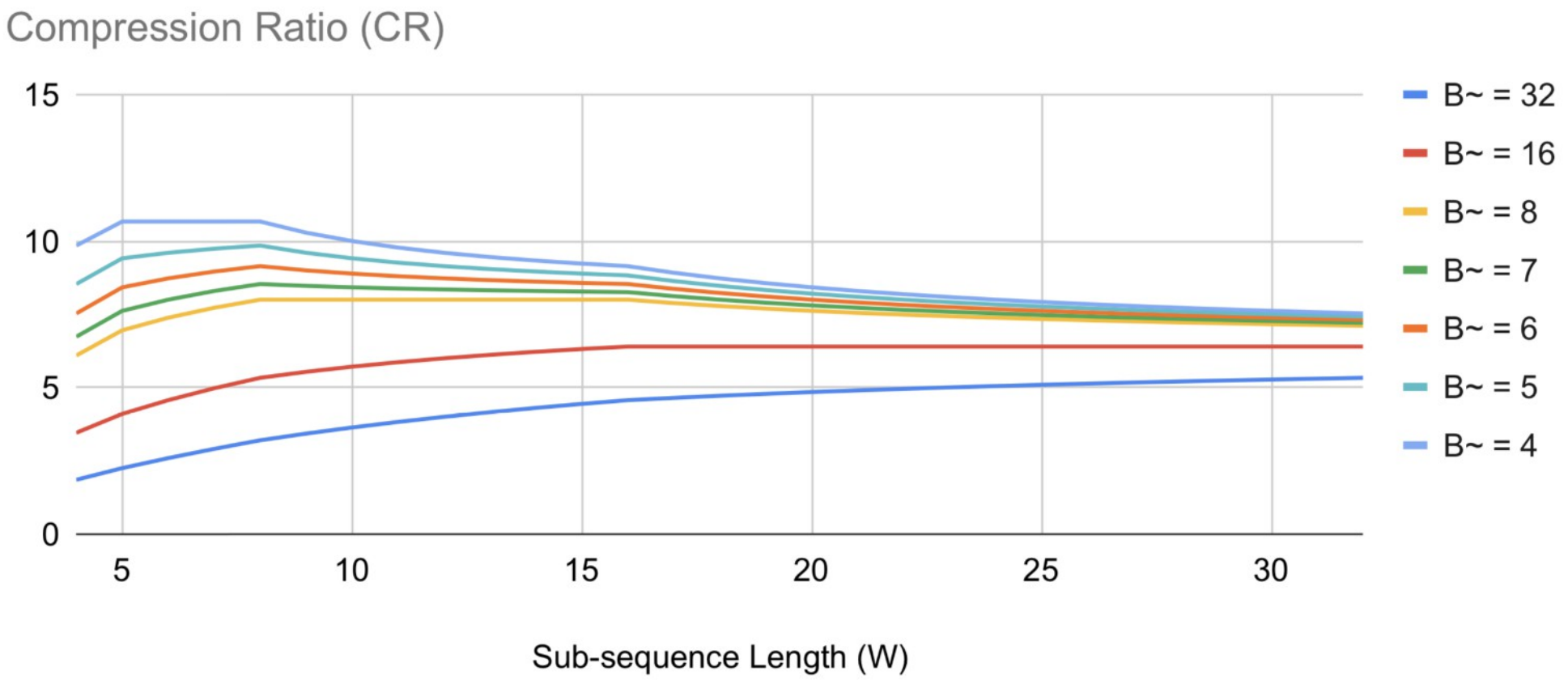
$$SizeOf(c) = \frac{N}{W} \times \left(2\tilde{B} - 1 + \sum_{i=2}^{W} \lceil \log_2 i \rceil \right)$$



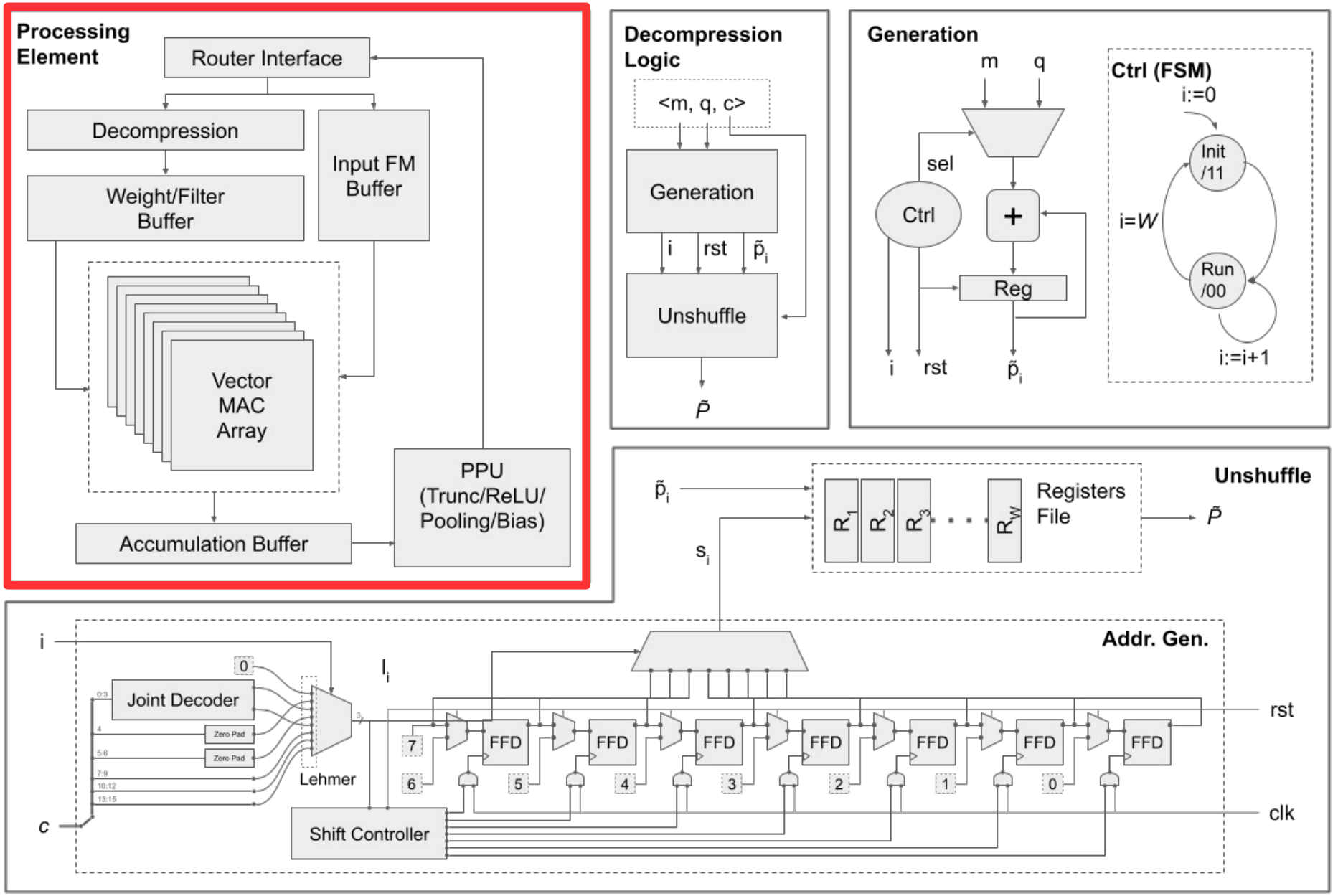
$$MF_{comp} = 2\tilde{B} + SizeOf(c)$$

Compression Ratio

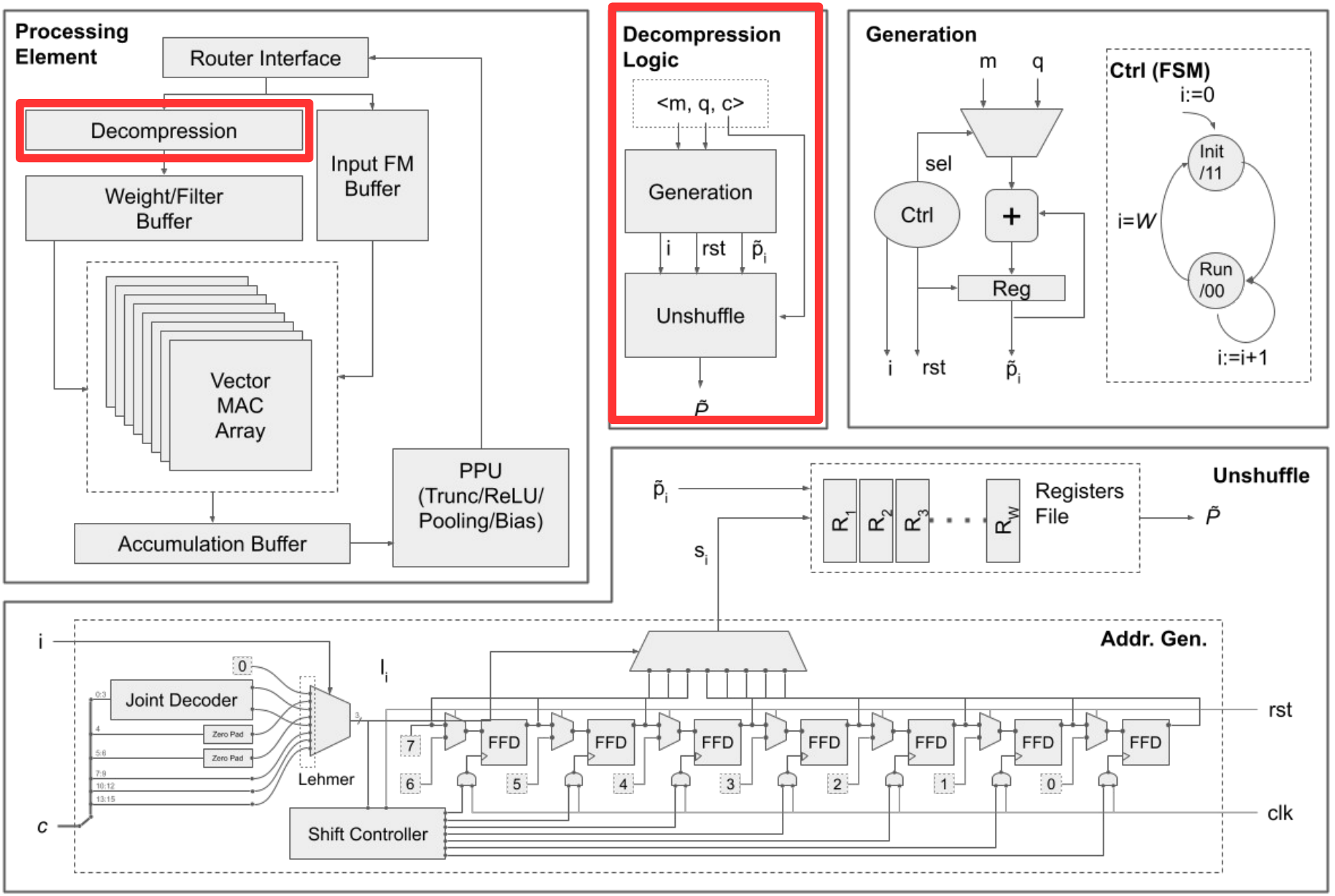
$$CR = \frac{B \times W}{2\tilde{B} - 1 + \sum_{i=2}^{W} \lceil \log_2 i \rceil}$$



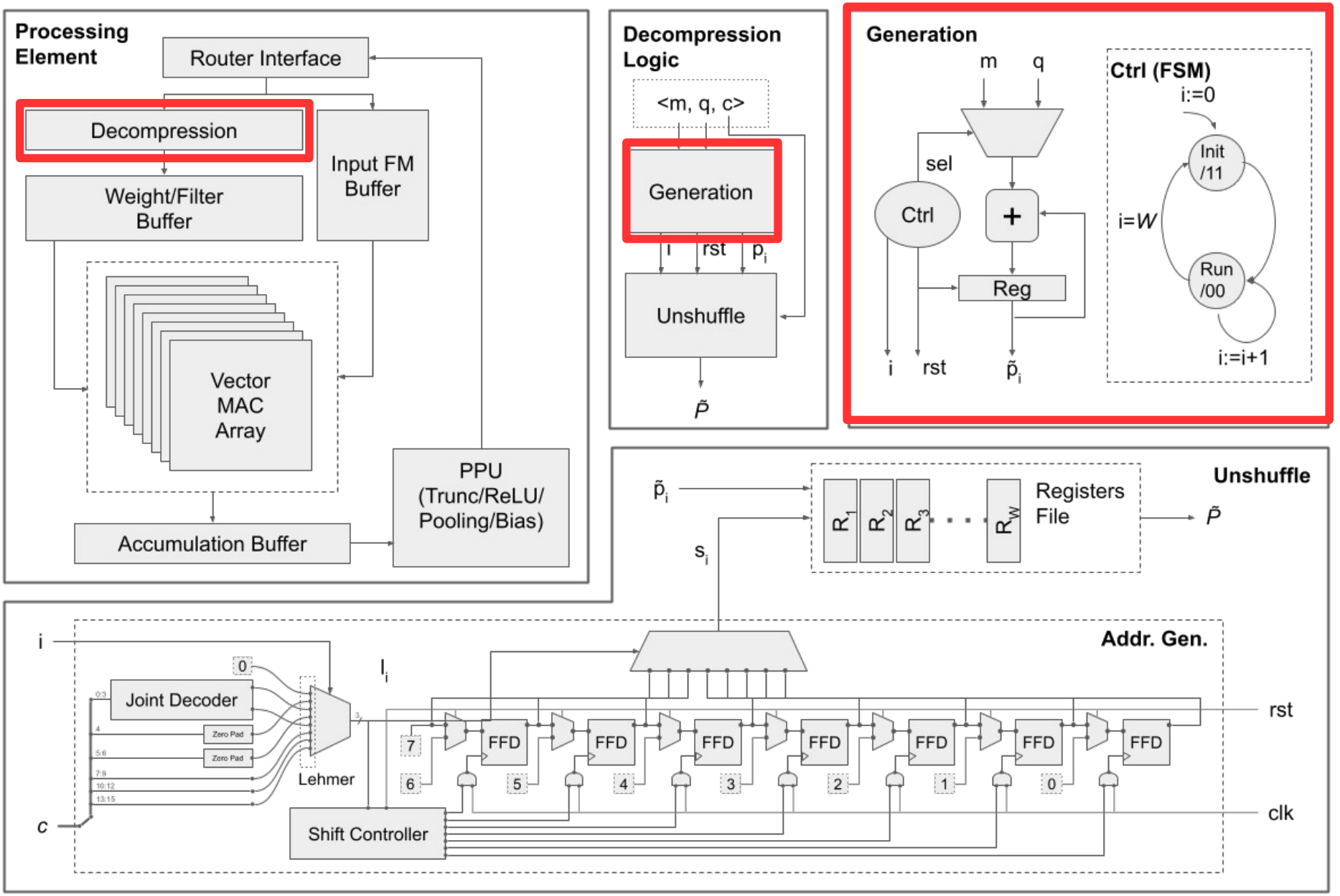
Processing Element



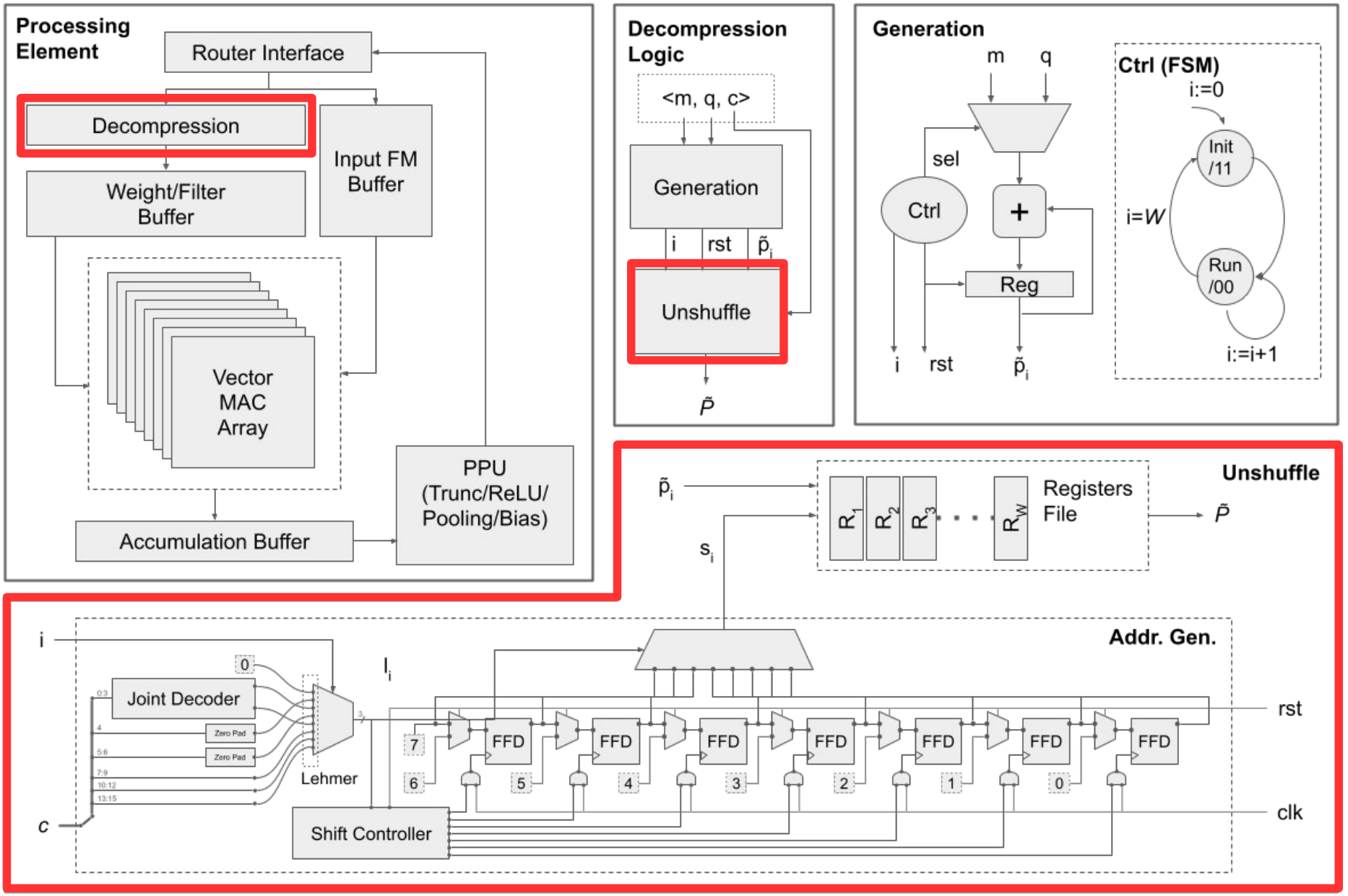
Processing Element



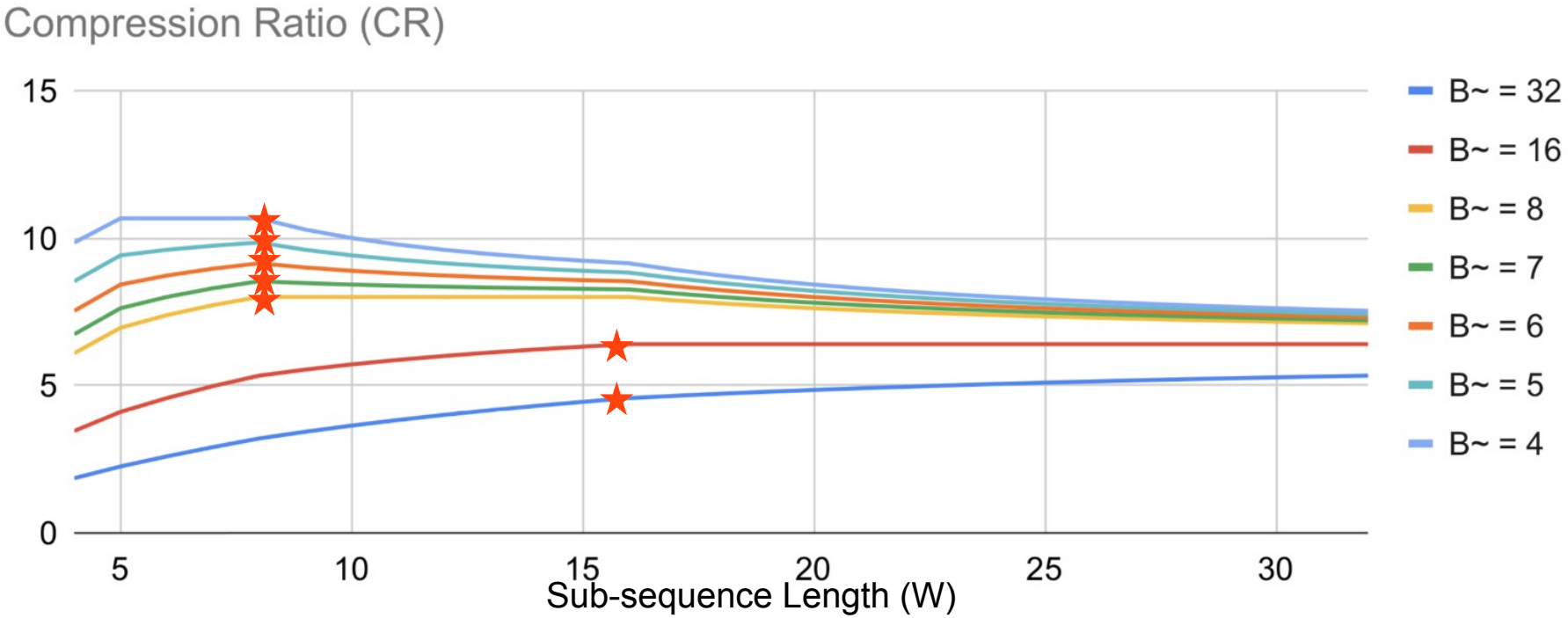
Processing Element



Processing Element



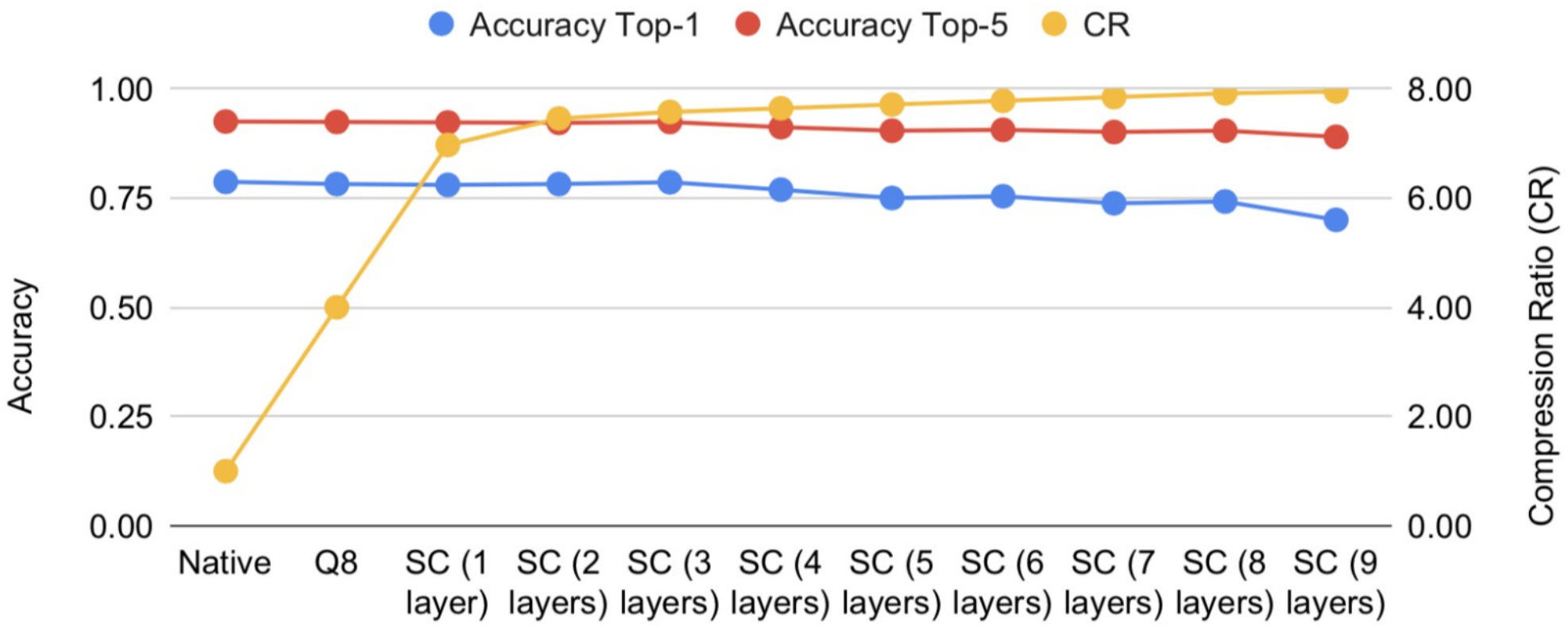
Area/Power Overhead



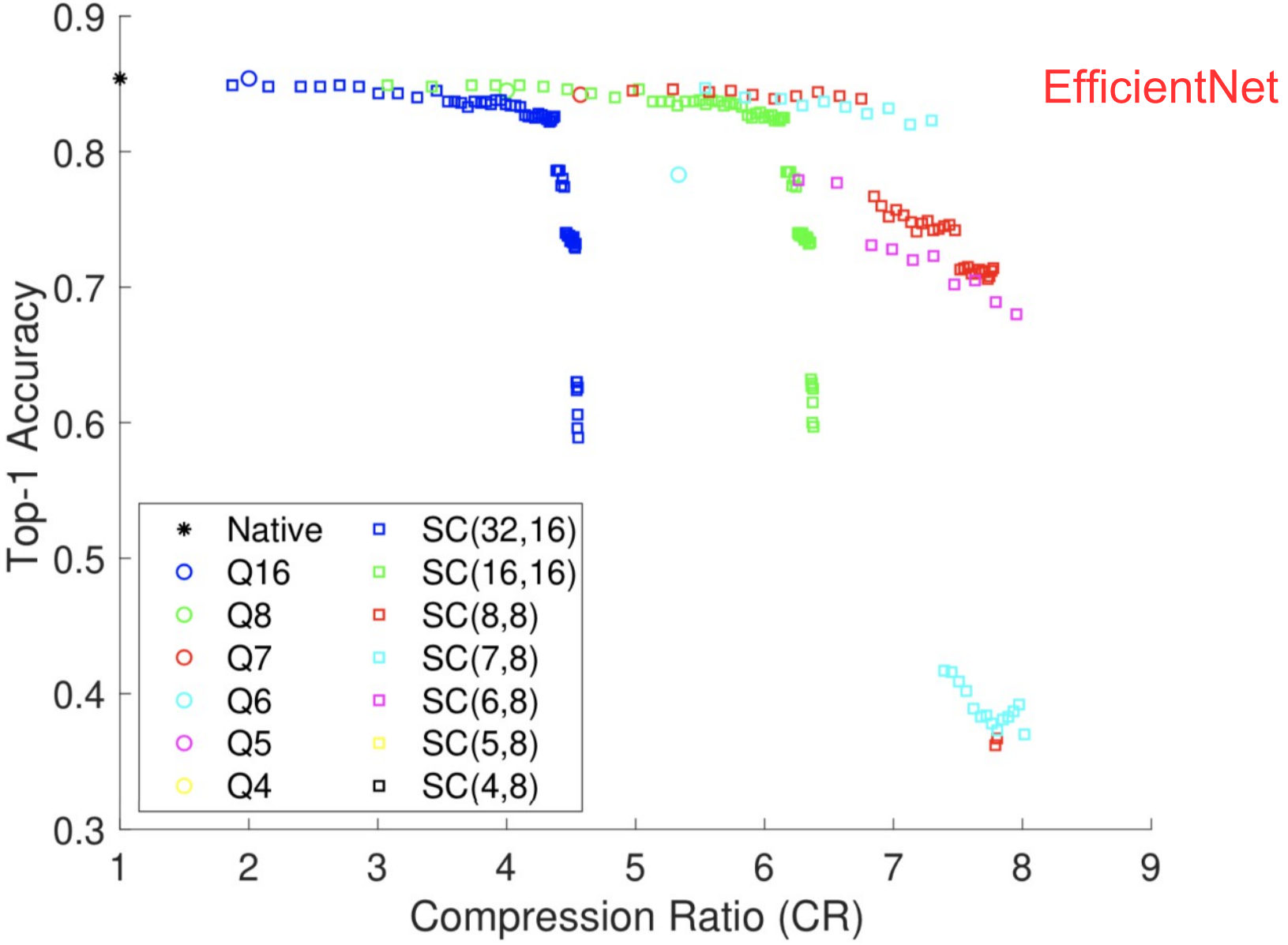
Configuration $(\widetilde{B}, W)$	Area (μm^2)	Power (μW)	Overhead Area	Overhead Power
(4, 8)	60	320	0.15%	0.077%
(5, 8)	385	340	0.17%	0.081%
(6, 8)	415	360	0.18%	0.086%
(7, 8)	442	370	0.19%	0.091%
(8, 8)	470	390	0.20%	0.095%
(16, 16)	1795	720	0.78%	0.175%
(32, 16)	2858	1060	1.24%	0.257%

Accuracy *vs.* Compression Ratio

VGG-16

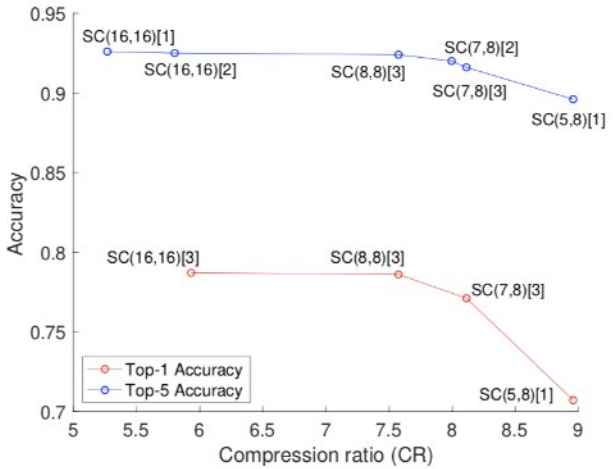


Accuracy *vs.* Compression Ratio

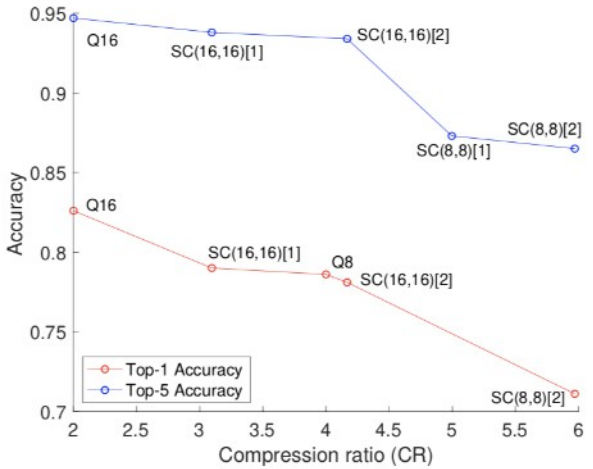


Pareto Fronts

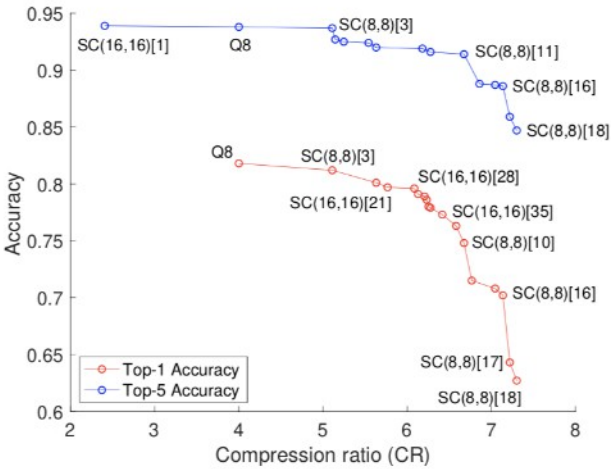
VGG-16



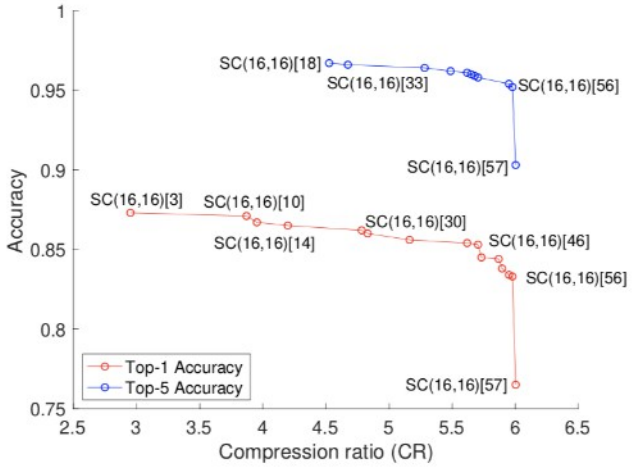
MobileNet



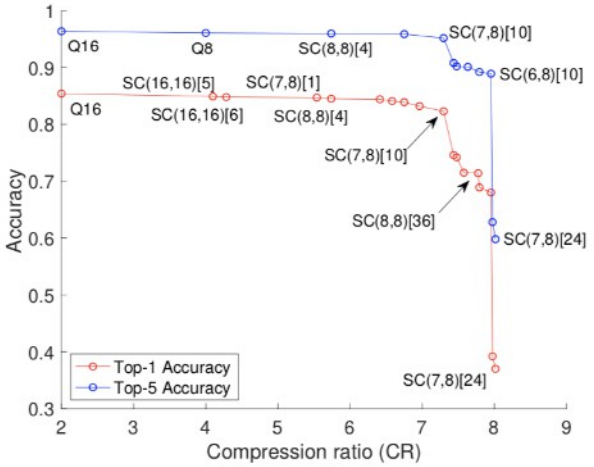
ResNet50



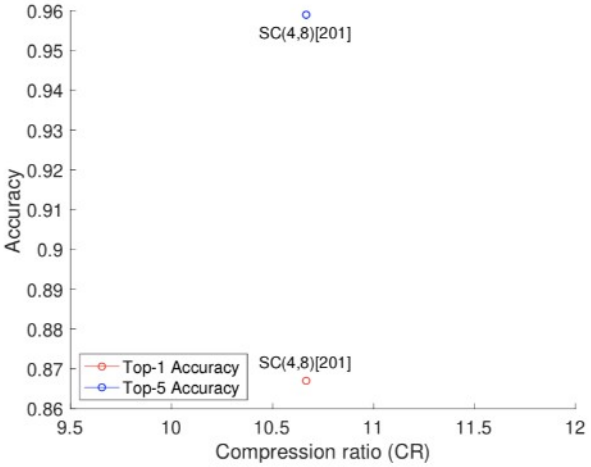
InceptionV3



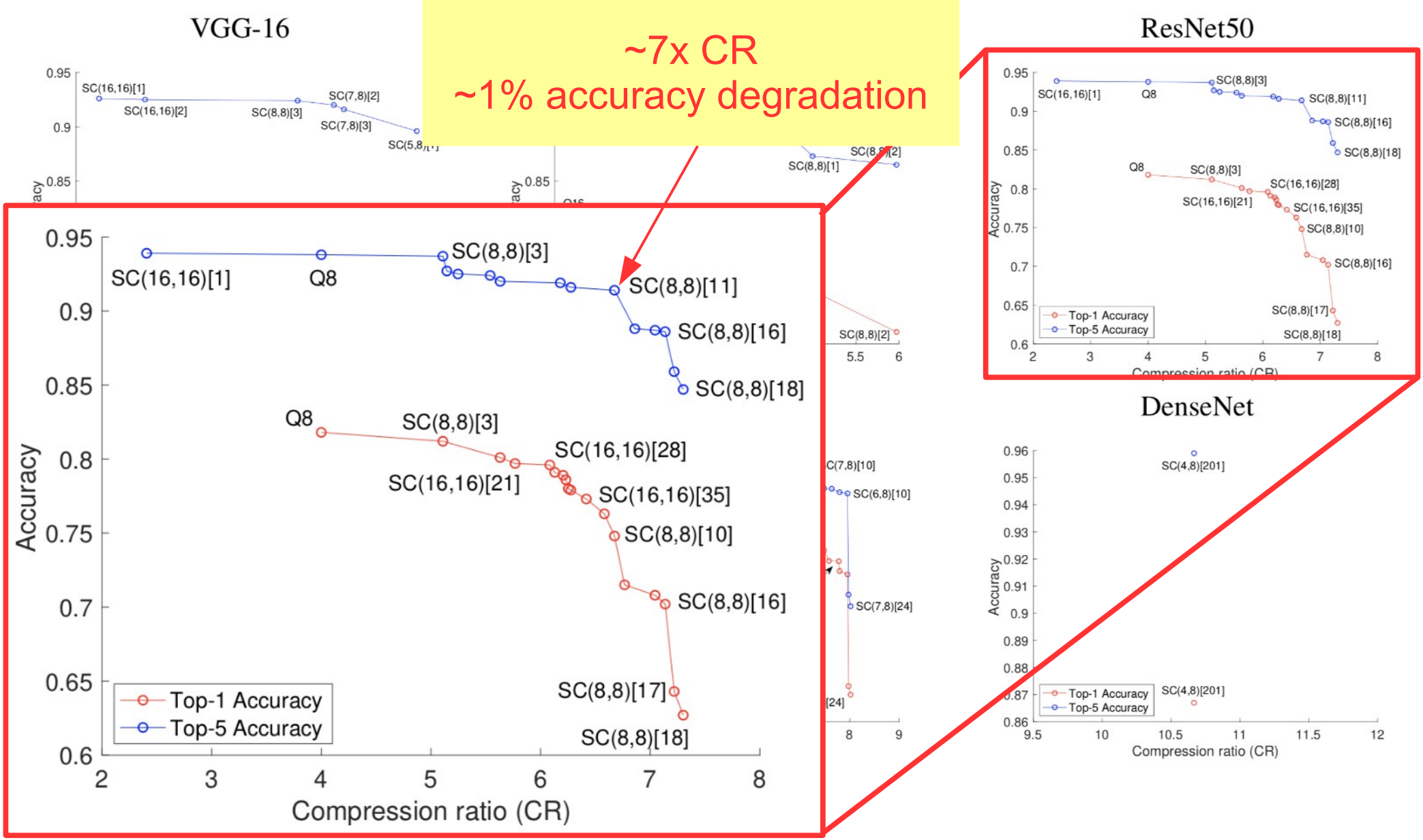
EfficientNet



DenseNet



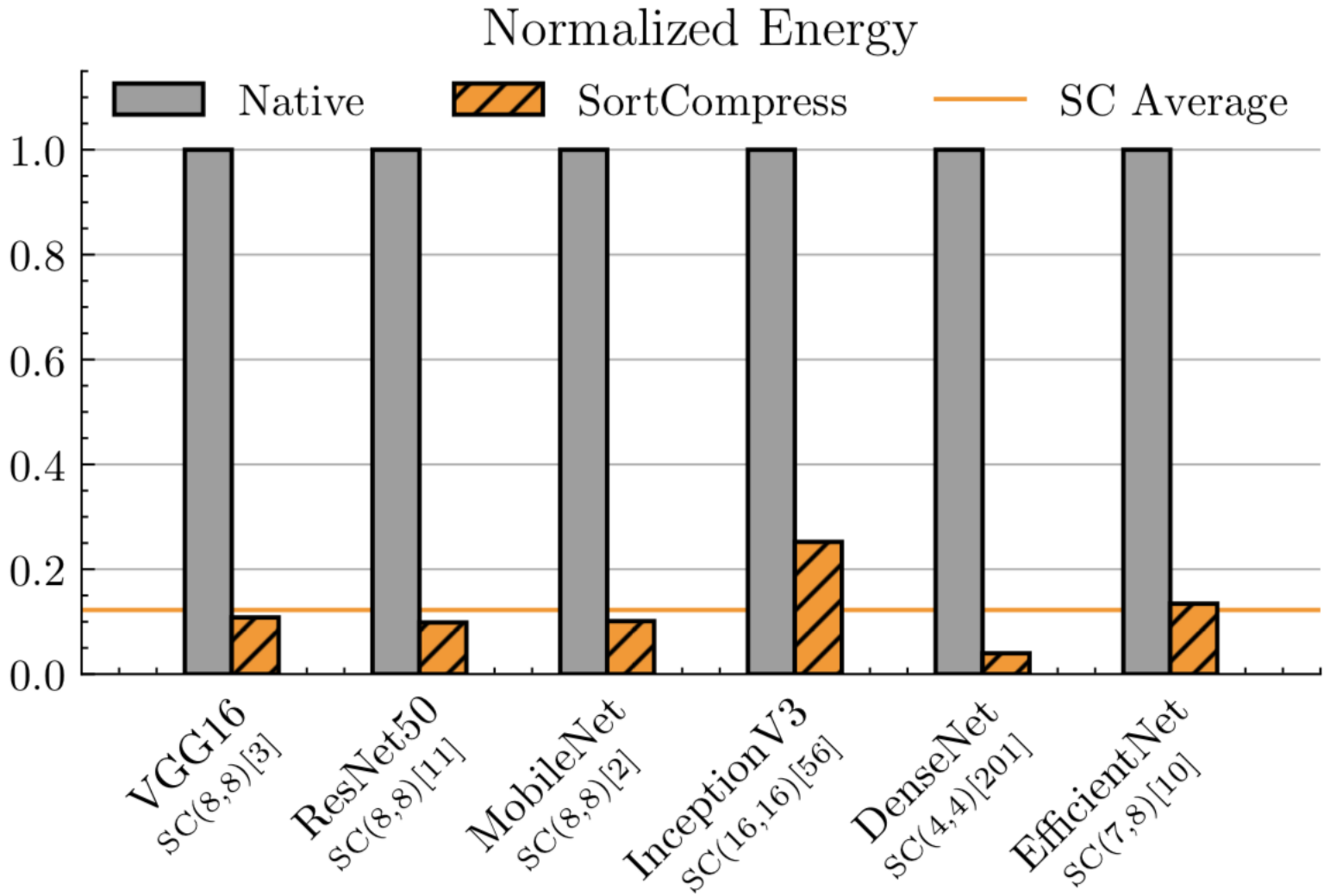
Pareto Fronts



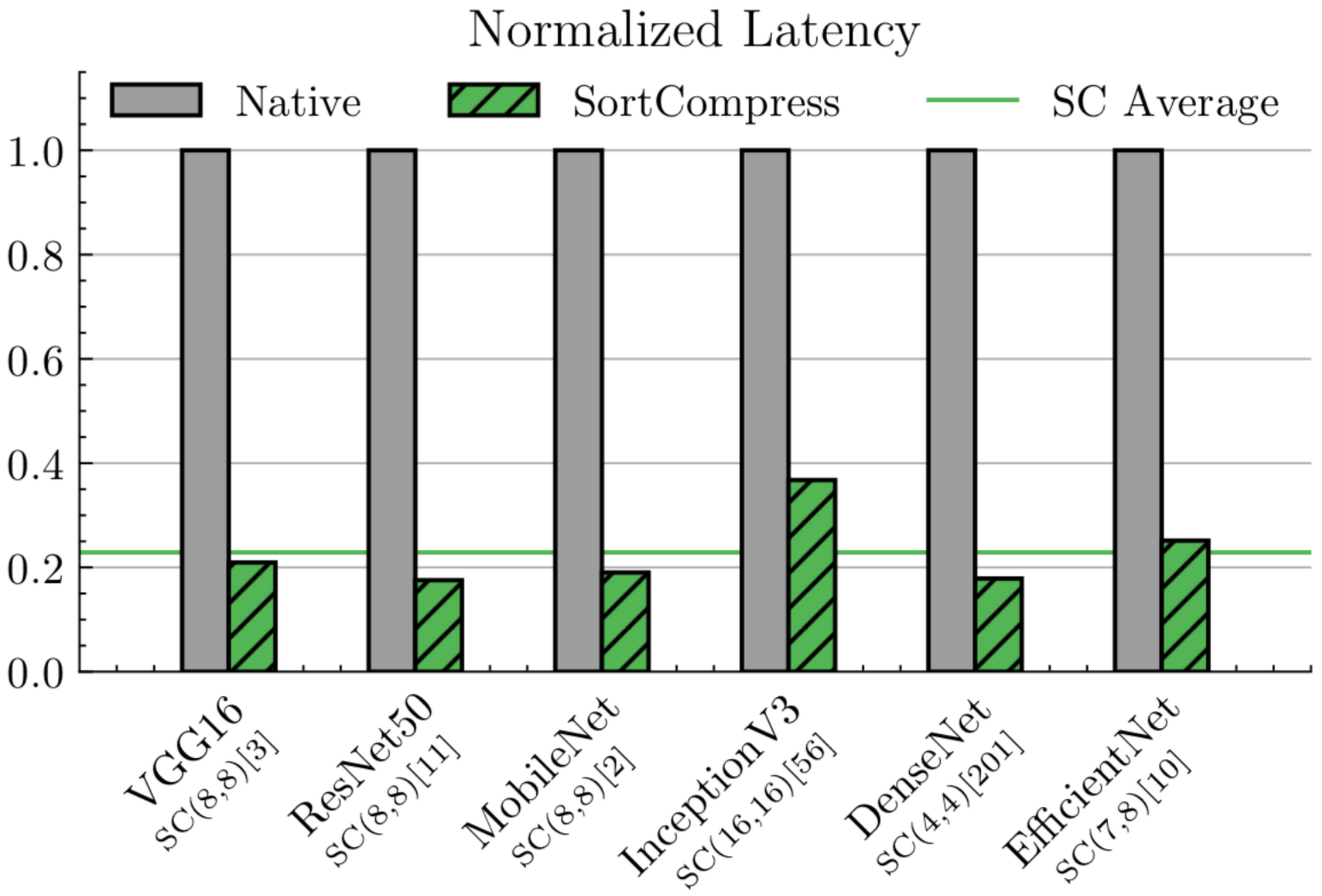
Performance/Energy Analysis

DRAM	
Technology	LPDDR4
Bandwidth	17.9 GB/s
Chiplet	
Technology	45nm
Number of PEs	16
Global Buffer Size	64 KiB
PE	
Clock Frequency	1 GHz
Weight Buffer Size	32 KiB
Input Buffer Size	8 KiB
Accumulation Buffer Size	3 KiB
MACs per PE	64

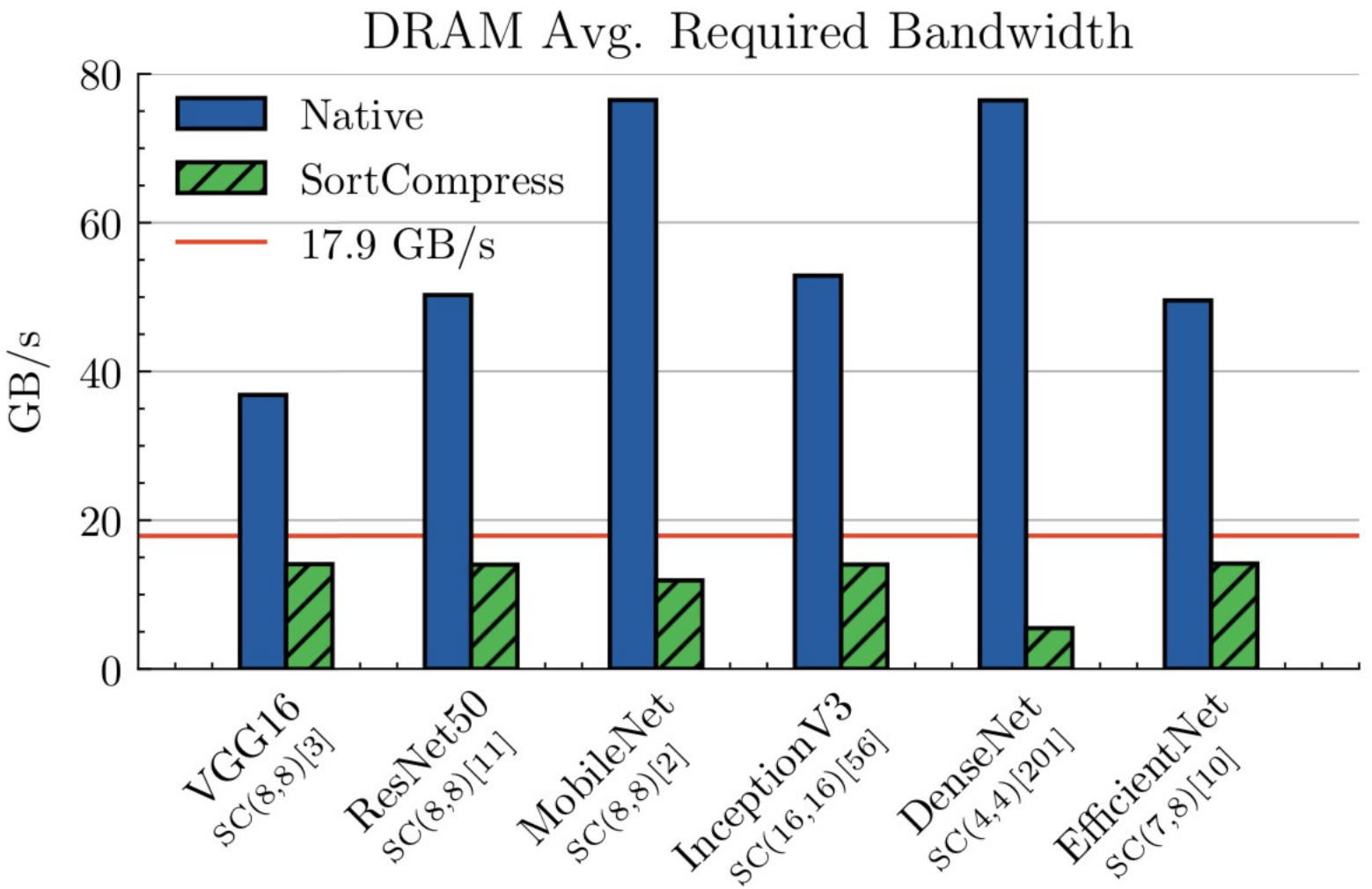
Energy Analysis



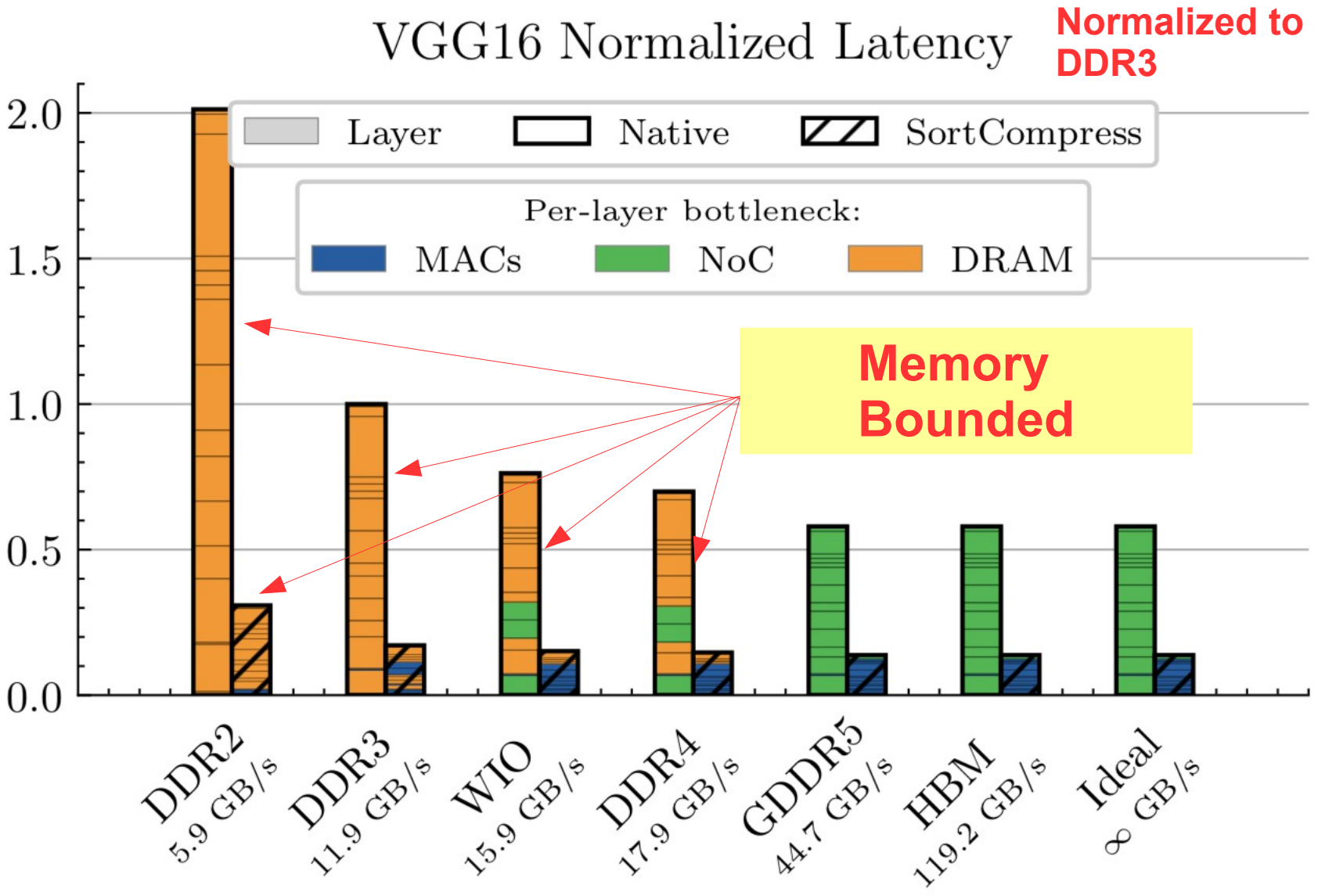
Latency Analysis



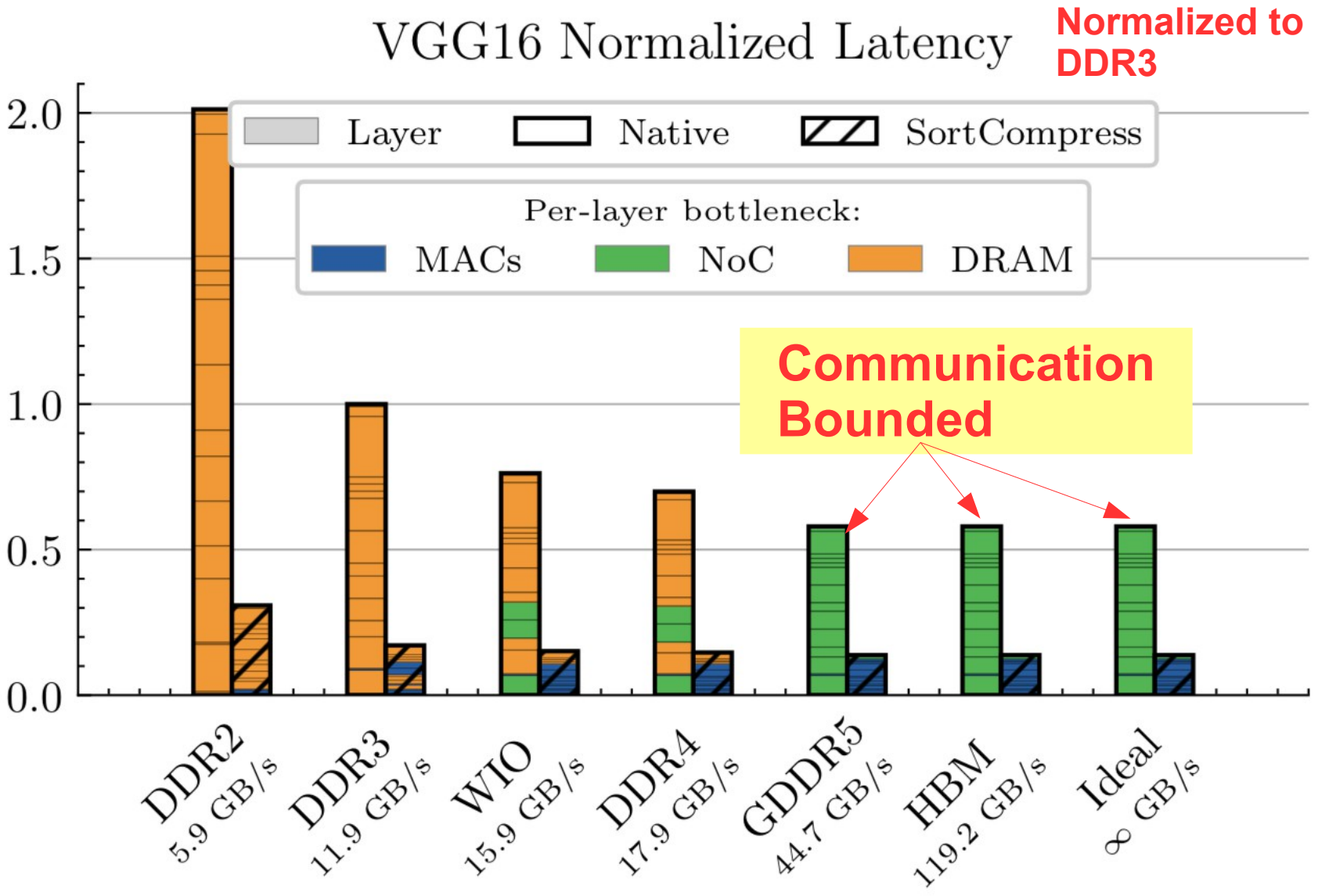
Demanded Bandwidth



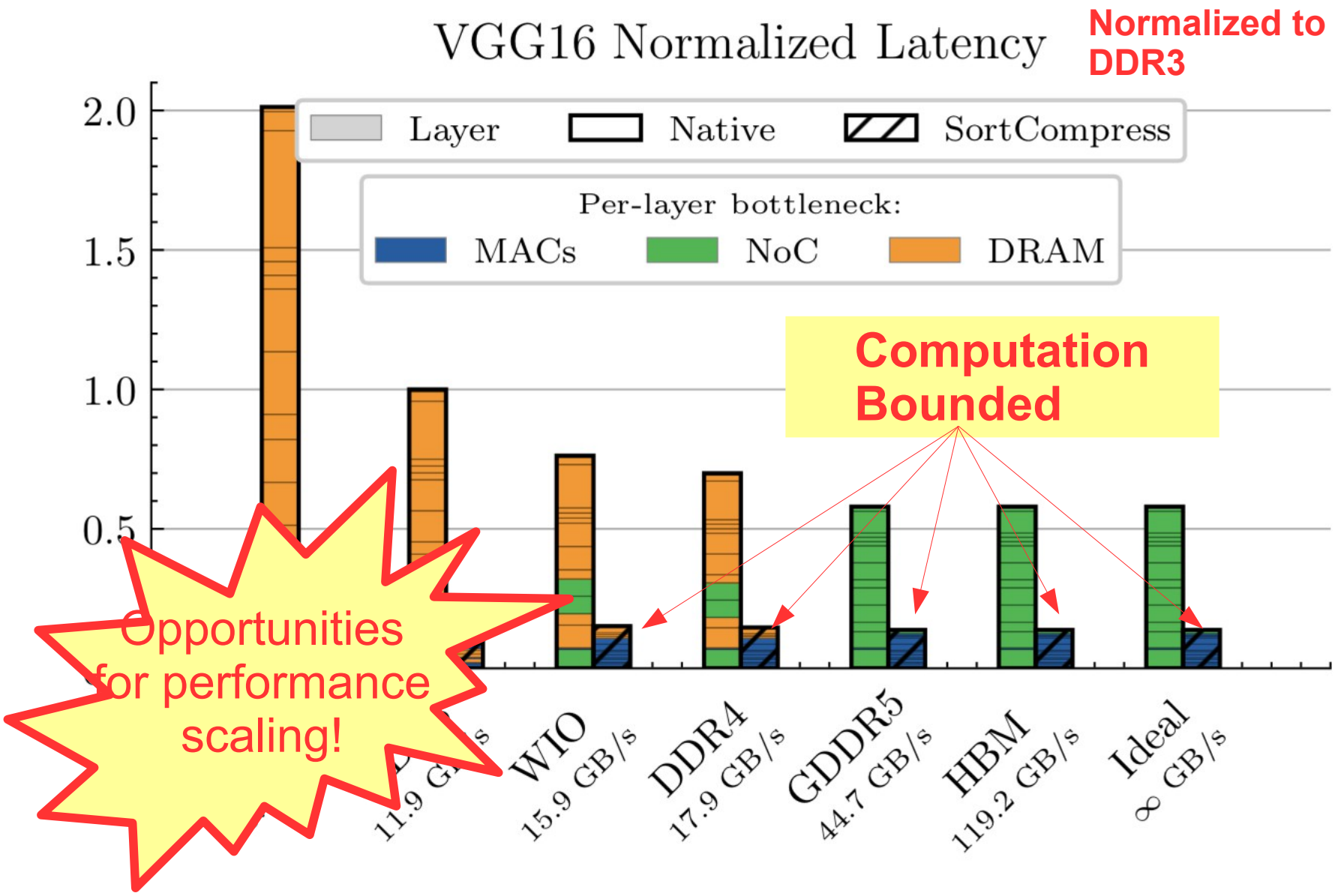
Per-Layer Bottleneck



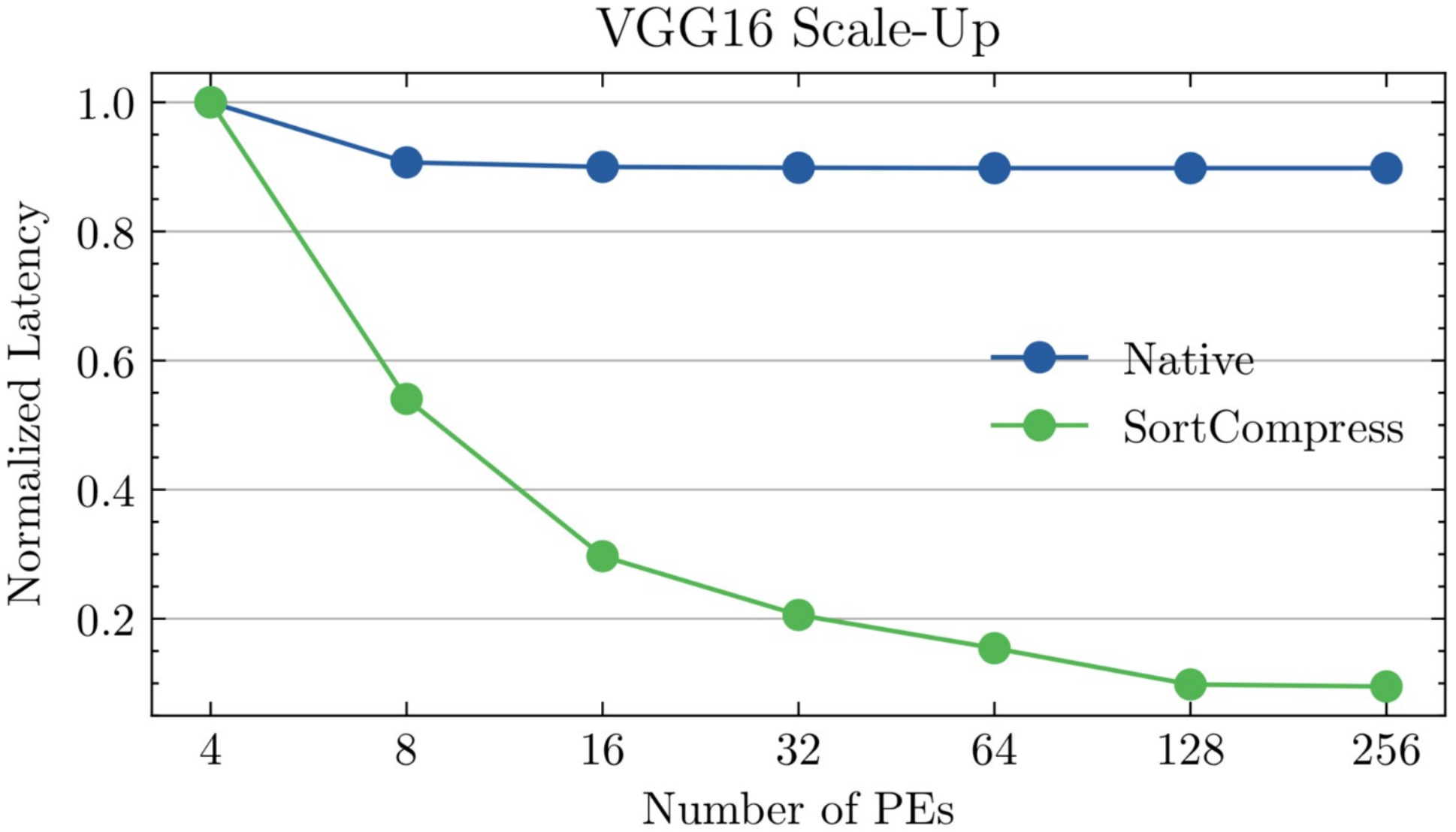
Per-Layer Bottleneck



Per-Layer Bottleneck

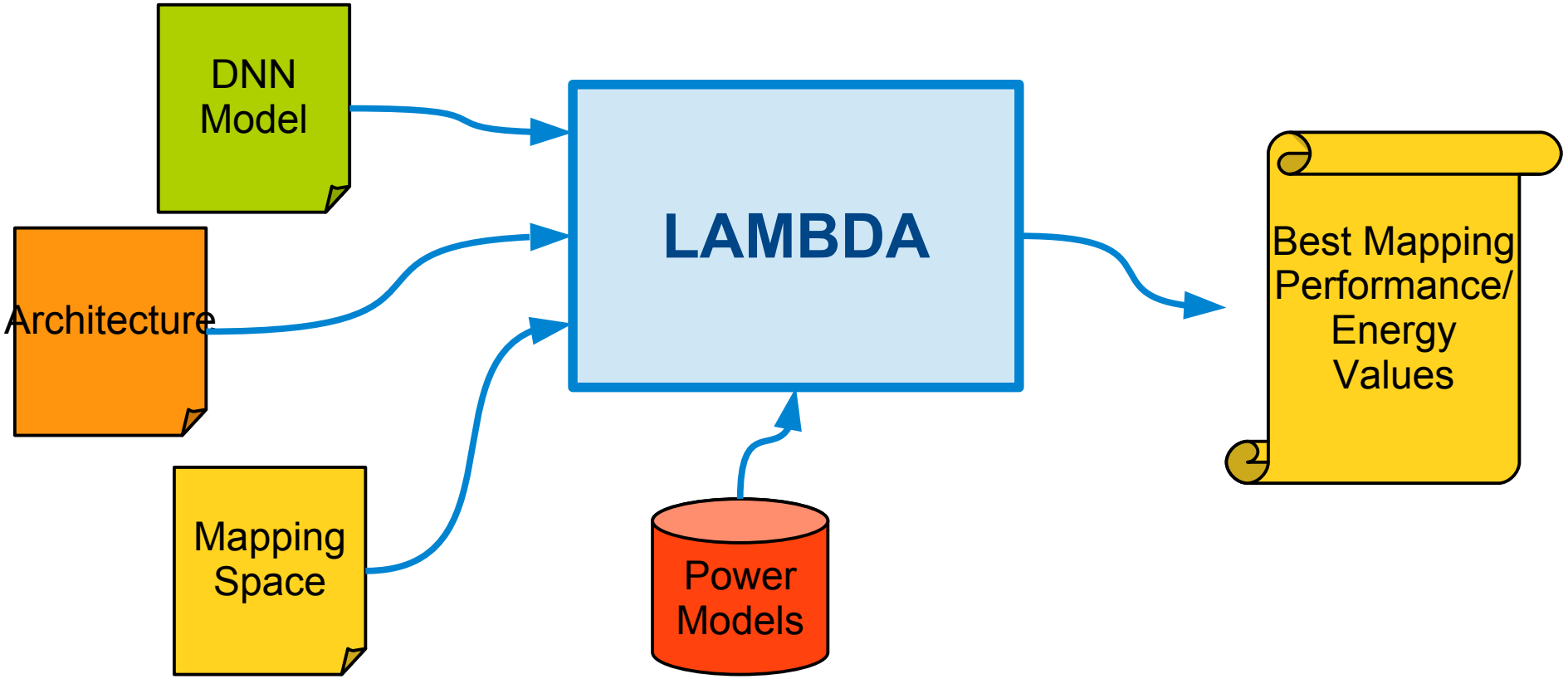


Scaling-Up Performance



Evaluation Platform

- LAMBDA: An Open Framework for Deep Neural Network Accelerators Simulation. PerCom'21
- https://github.com/Haimrich/timeloop



Summary

- DNN accelerators are communication and memory bounded
- Need for invastigating towards techniques to improve memory and communication efficiency
- Hardware-friendly compression techniques
- Emerging communication technologies (*e.g.*, WiNoC)