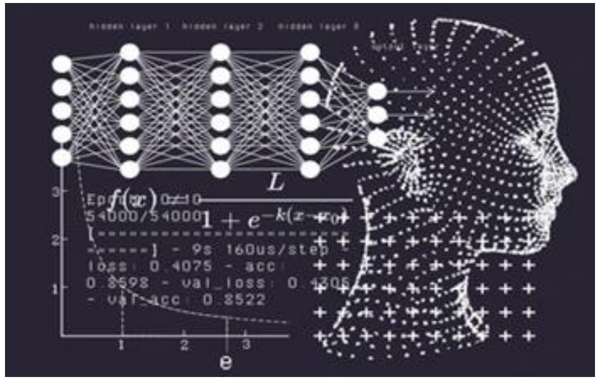
Sparseloop:
An Analytical Approach to Sparse Tensor Accelerator Modeling

Yannan Nellie Wu[1], Po-An Tsai[2], Angshuman Parashar[2], Vivienne Sze[1], Joel Emer[1,2]

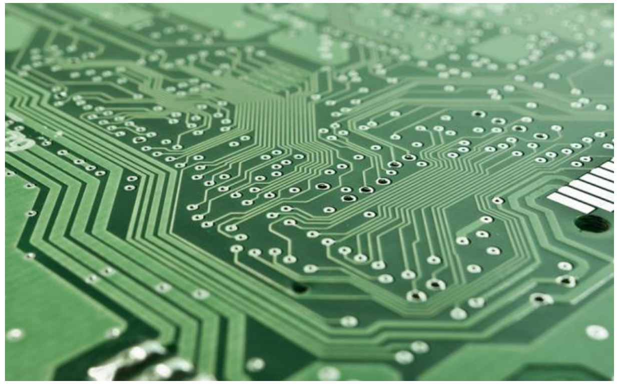
[1]MIT [2]NVIDIA

http://sparseloop.mit.edu/

Many Applications Use Sparse Tensor Algebra



Sparse Neural Networks



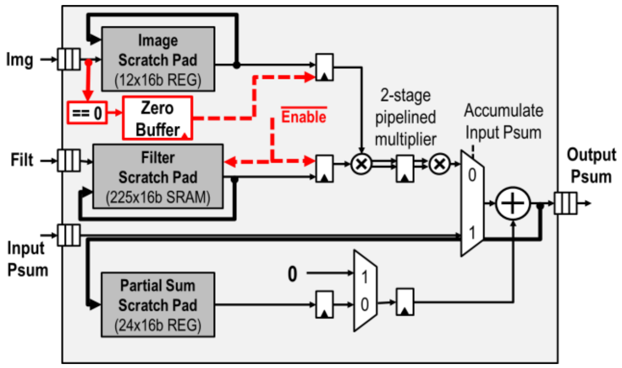
Circuit Simulations

Data Science

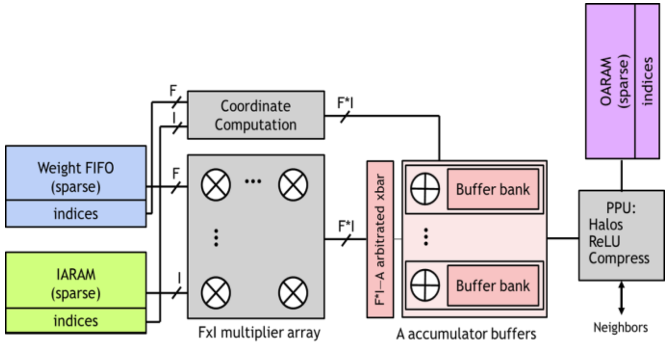
Inefficient Processing on General-Purpose Processors

An Explosion of Sparse Tensor Accelerators

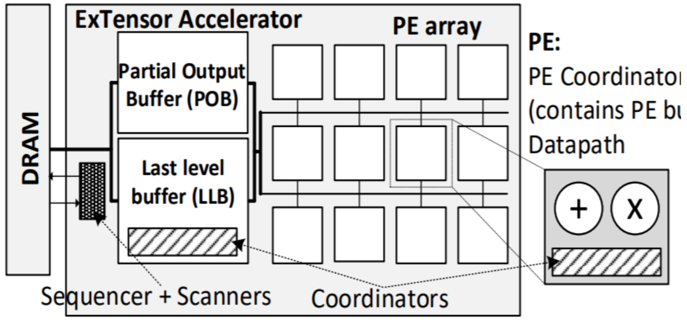
Large Design Space


Img
Image Scratch Pad (12x16b REG)
== 0
Zero Buffer
Enable
Filt
Filter Scratch Pad (225x16b SRAM)
2-stage pipelined multiplier
Accumulate Input Psum
Output Psum
Input Psum
Partial Sum Scratch Pad (24x16b REG)
0
1
0

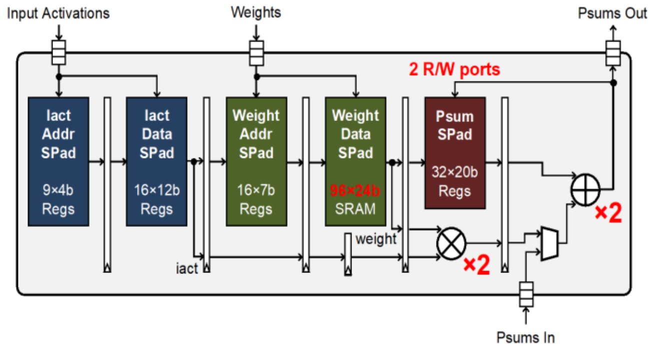
Eyeriss [JSSC2017]



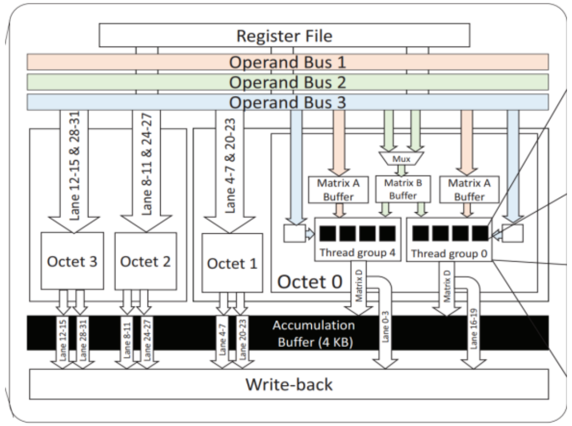
SCNN [ISCA2017]



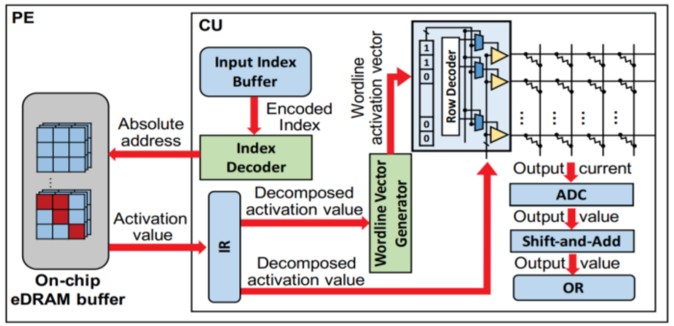
ExTensor [MICRO2019]



Eyeriss V2 [JETCAS2019]



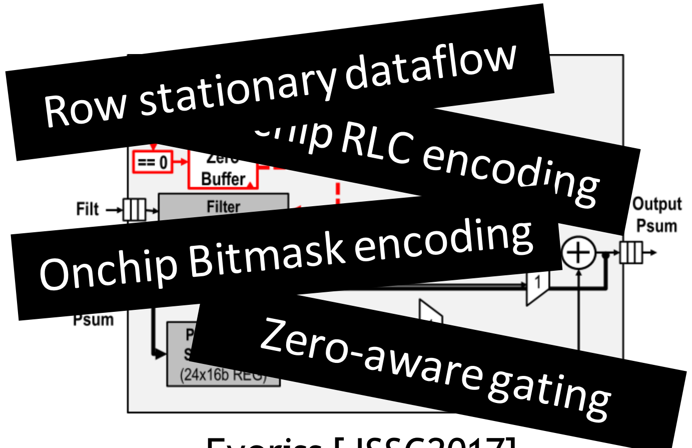
DSTC [ISCA2021]



Sparse-ReRAM [ISCA2019]

Rely on Diverse Design-Specific Terminologies

Large, Unstructured, and Confusing Design Space



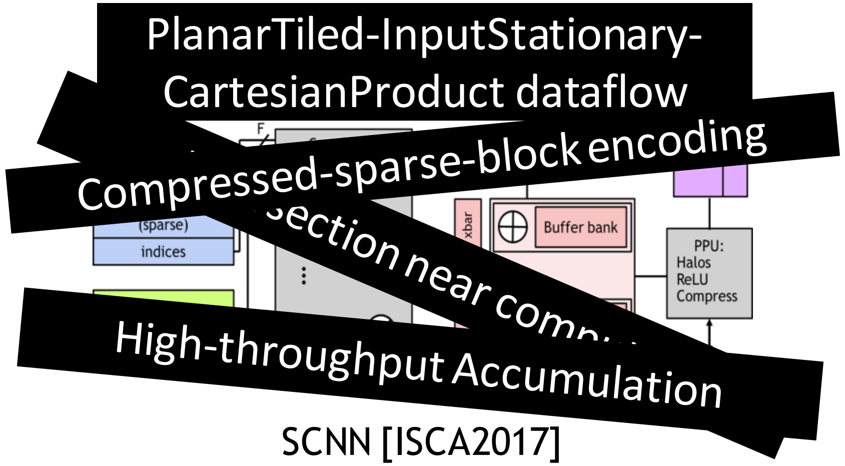
Row stationary dataflow

hip RLC encoding

Onchip Bitmask encoding

Zero-aware gating

Eyeriss [JSSC2017]



PlanarTiled-InputStationary-CartesianProduct dataflow

Compressed-sparse-block encoding

section near comp

High-throughput Accumulation

SCNN [ISCA2017]

DRAM
Output
Last level buffer (LLB
datapath
+
X
Se

Hybrid Cartesian product dataflow

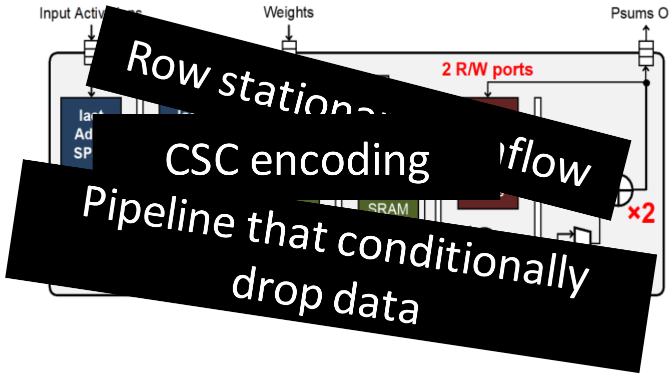
Hierarchical encoding

-elimination

Hie

Fast intersection unit with “skipTo” heuristic

ExTensor [MICRO2019]



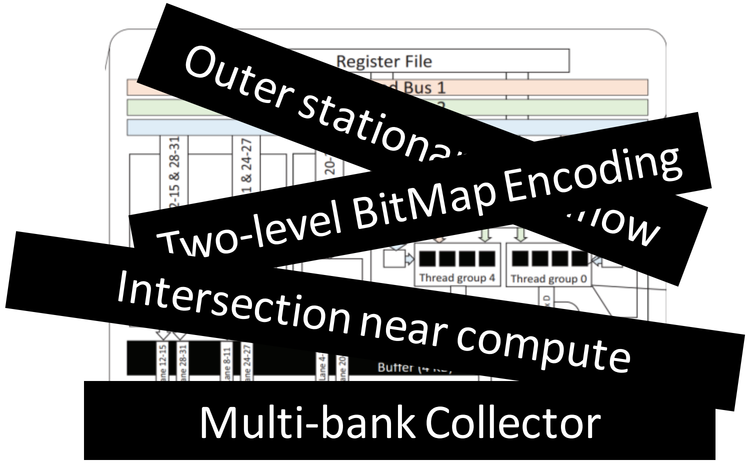
Row station

flow

CSC encoding

Pipeline that conditionally drop data

Eyeriss V2 [JETCAS2019]



Outer stationa

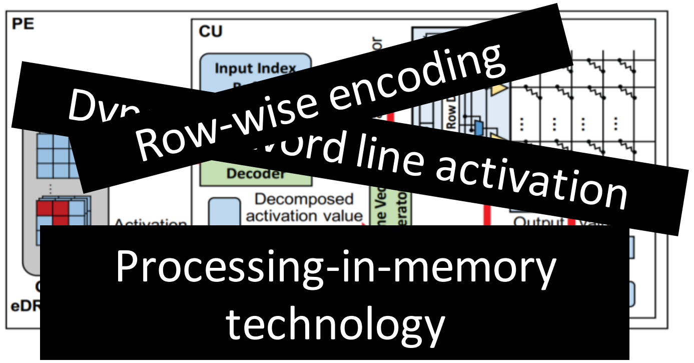
Two-level BitMap Encoding

low

Intersection near compute

Multi-bank Collector

DSTC [ISCA2021]



Dyn

Row-wise encoding

ord line activation

Processing-in-memory technology

Sparse-ReRAM [ISCA2019]

Important to systematically understand and explore the design space

Requirements

A Modeling Framework

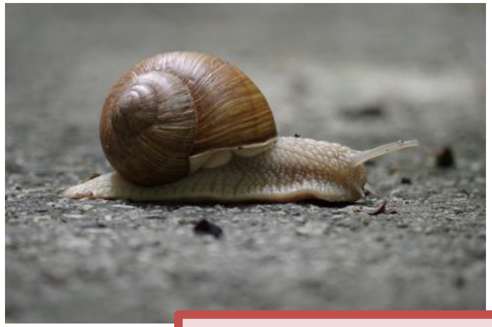
- Flexible
- Fast
- Accurate

Existing Modeling Frameworks are Insufficient

(Design-Specific) Cycle-Level Simulators

SCNN[ISCA16], STONNE[CAL21], MAGNET[ICCAD19], DNNBuilder[ICCAD18], etc.

Slow

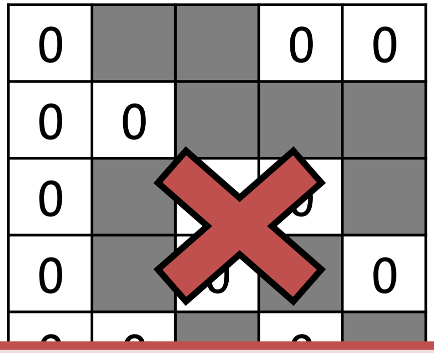


Inflexible

General Analytical Modeling Frameworks

Timeloop[ISPASS19], MAESTRO[MICRO19], Scale-Sim[ISPASS20], CoSA[ISCA21], etc.

No Sparsity Support

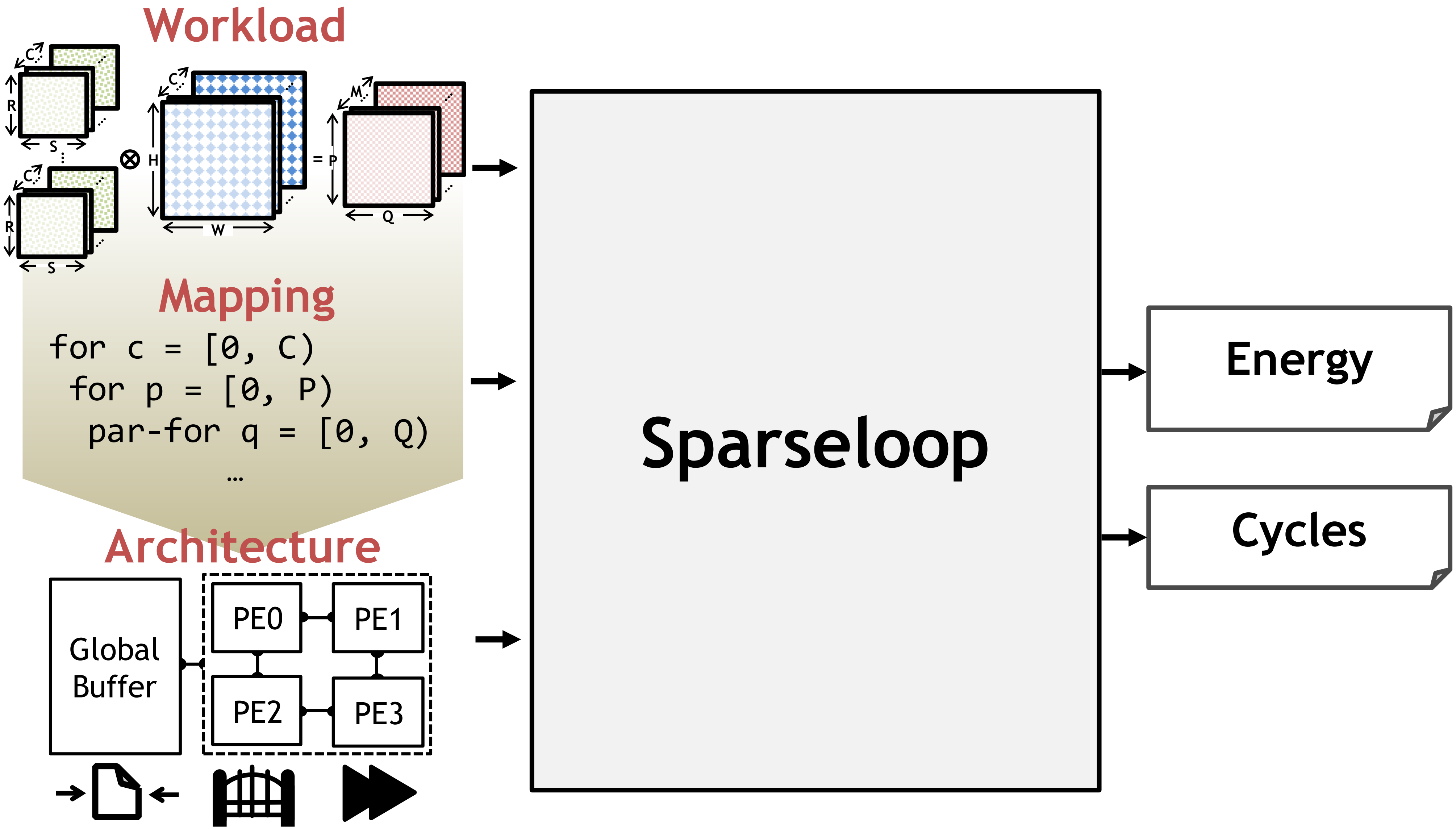


Solution

Sparseloop: The First Analytical Modeling Framework for Sparse Tensor Accelerators

http://sparseloop.mit.edu/

Sparseloop High-Level Framework



Workload

Mapping

```
for c = [0, C)
 for p = [0, P)
  par-for q = [0, Q)
     …
```

Architecture

Sparseloop

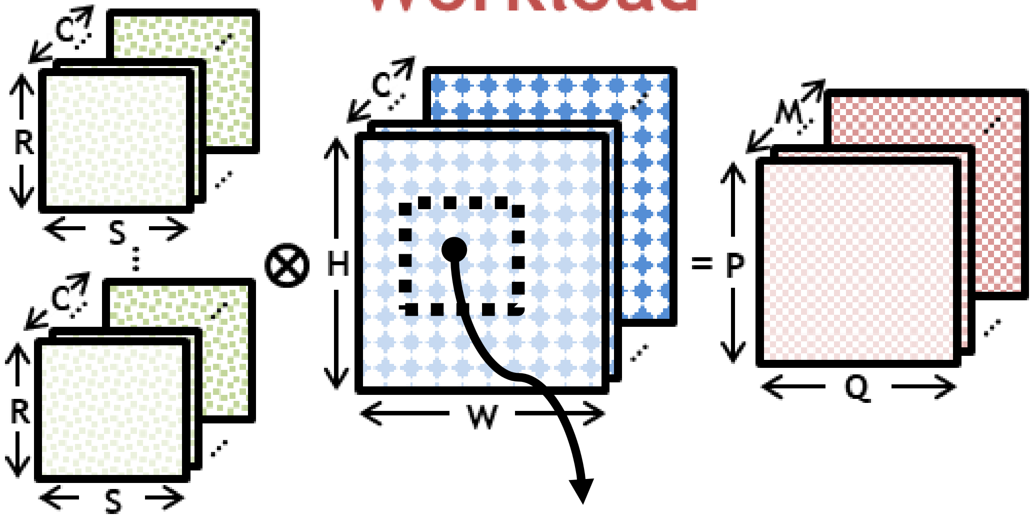
Energy

Cycles

Challenge: Slow Workload Characterization

Accelerator Performance is Data-dependent

Workload

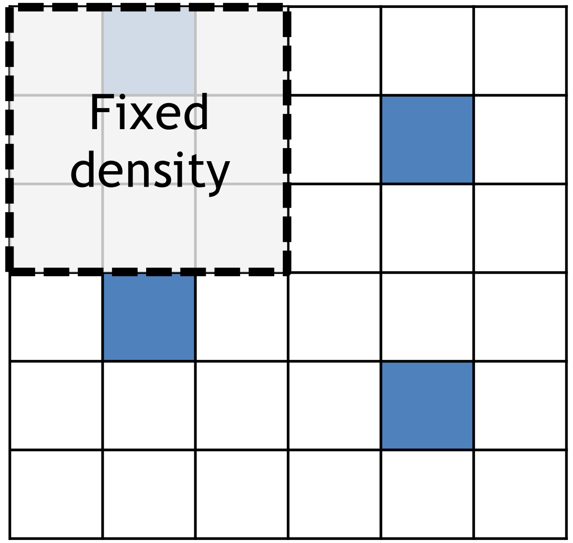


Nonzero values locations
in various subtensors

Traversing the exact values can be very slow

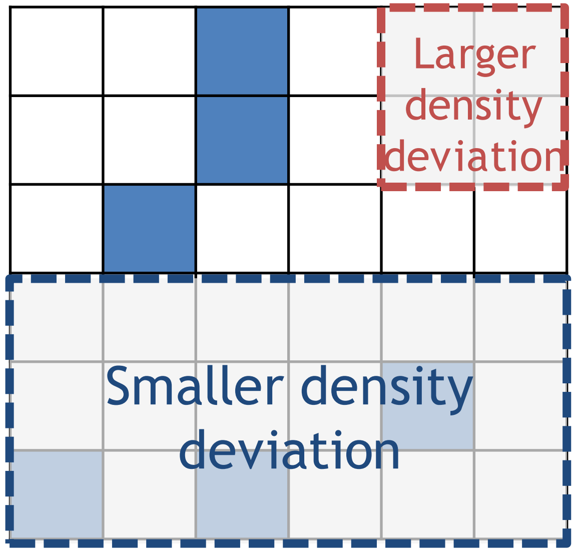
Sparseloop Solution: Statistical Characterization

Fixed-Structured



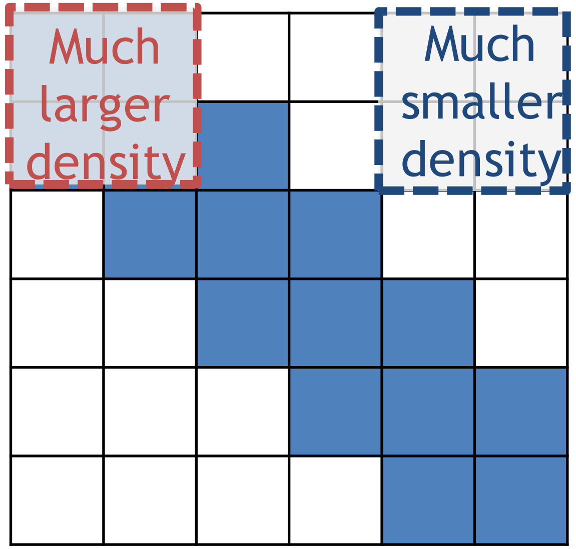
Structured Pruned DNNs

Uniform Random



Unstructured Pruned DNNs

Banded Distribution

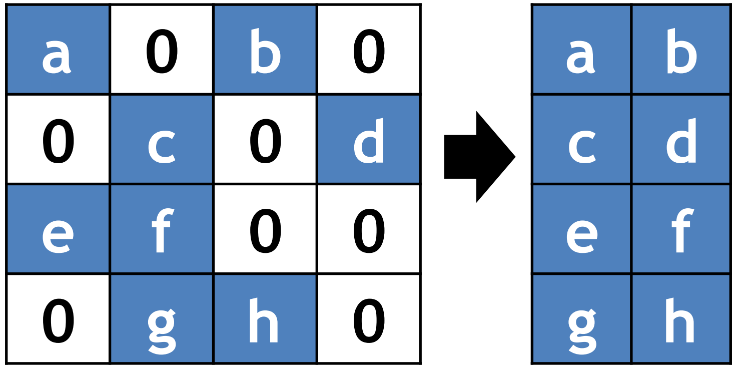


Scientific Simulations

Statistical Modeling Ensures Both Speed and Accuracy

Challenge: Unstructured Architecture Description

High-Level Opportunities



a	0	b	0
0	c	0	d
e	f	0	0
0	g	h	0

a	b
c	d
e	f
g	h

Zero Values

Can be Compressed Away

$$x \times 0 = 0$$

$$x + 0 = x$$

Ineffectual Operations

Can be Eliminated

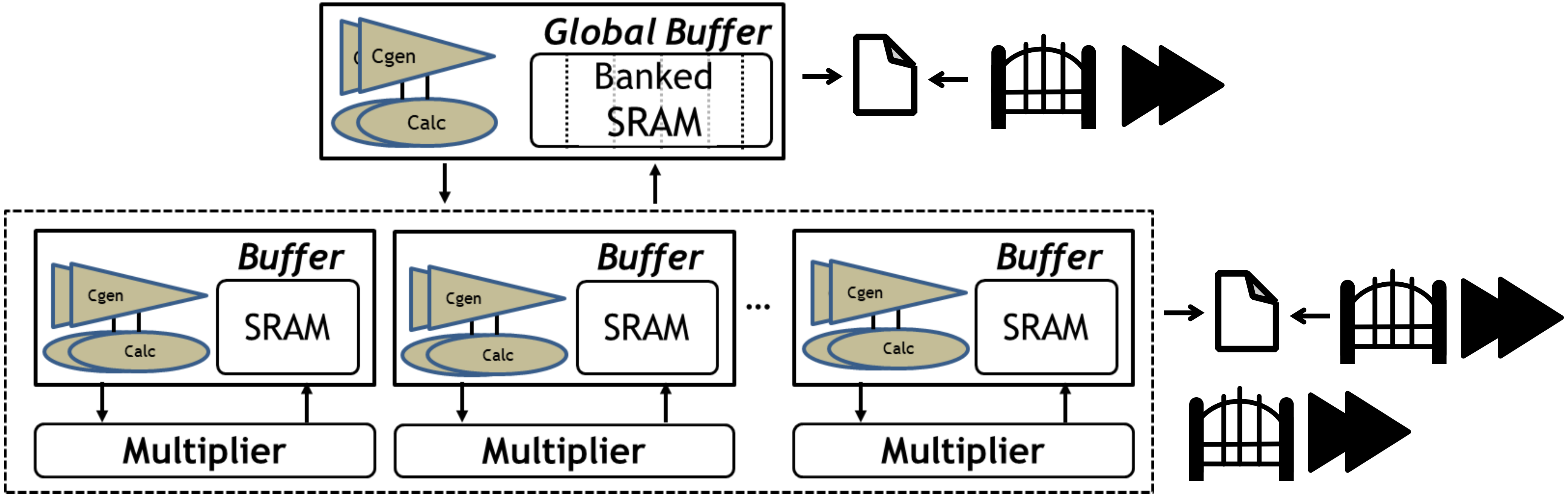
Can be Exploited Differently at Different Architecture Levels

Sparseloop Solution: Sparse Acceleration Features

Format
Choice of tensor representations

Gating
Explicitly let the hardware staying idle

Skipping
Explicitly fast forward to next effectual operation



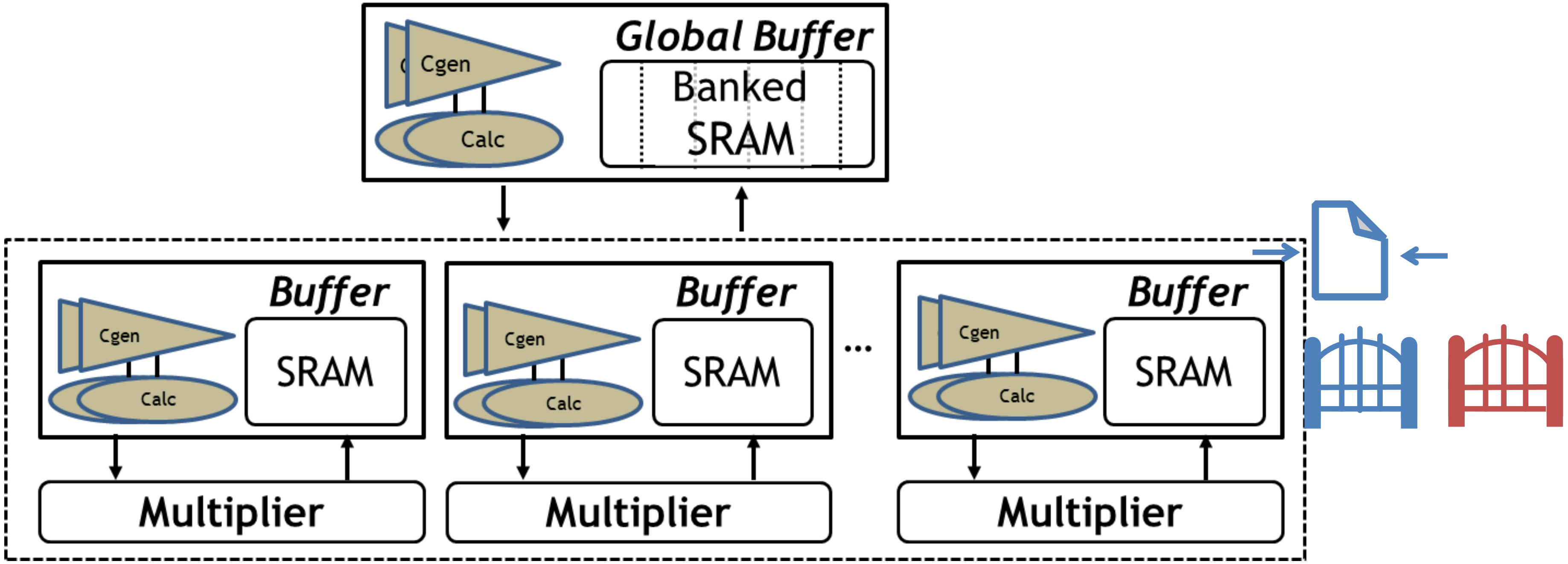
Example Accelerator Architecture Organization

Systematic Descriptions with Various SAF Combinations

Eyeriss-Style
[ISCA2016]

🟦 🟩 🟥

Different color coding represents features applied to different operand



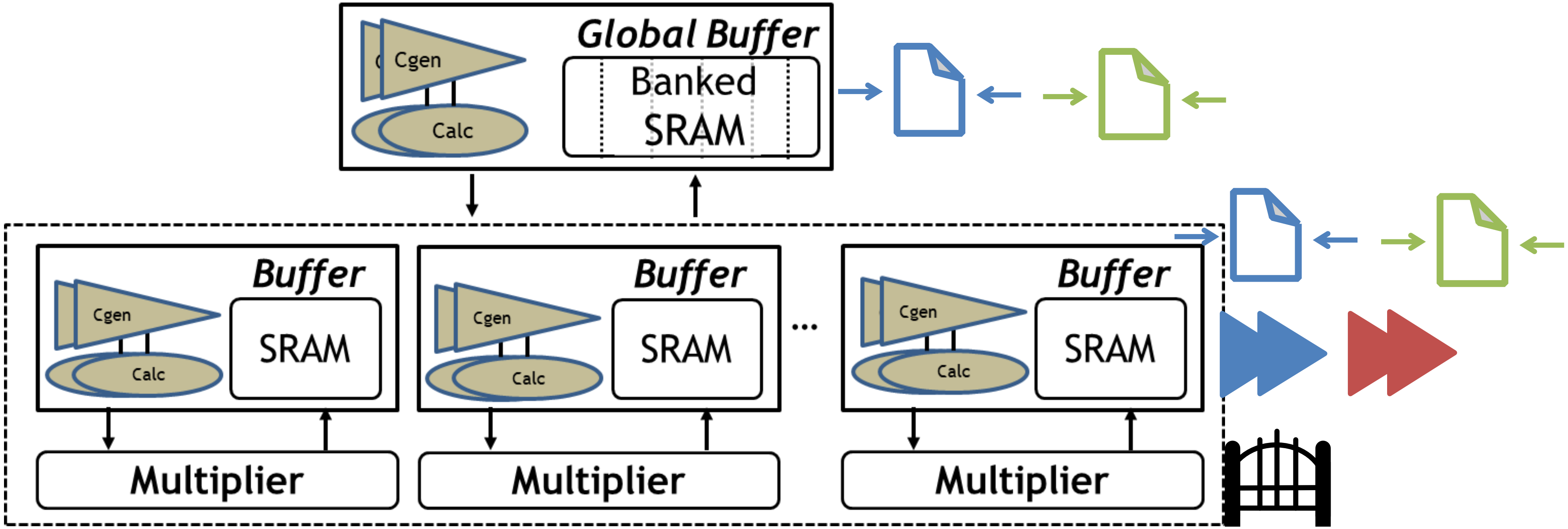
Example Accelerator Architecture Organization

Systematic Descriptions with Various SAF Combinations

Eyeriss V2-Style
[JETCAS2019]

🟦 🟩 🟥

Different color coding represents features applied to different operand

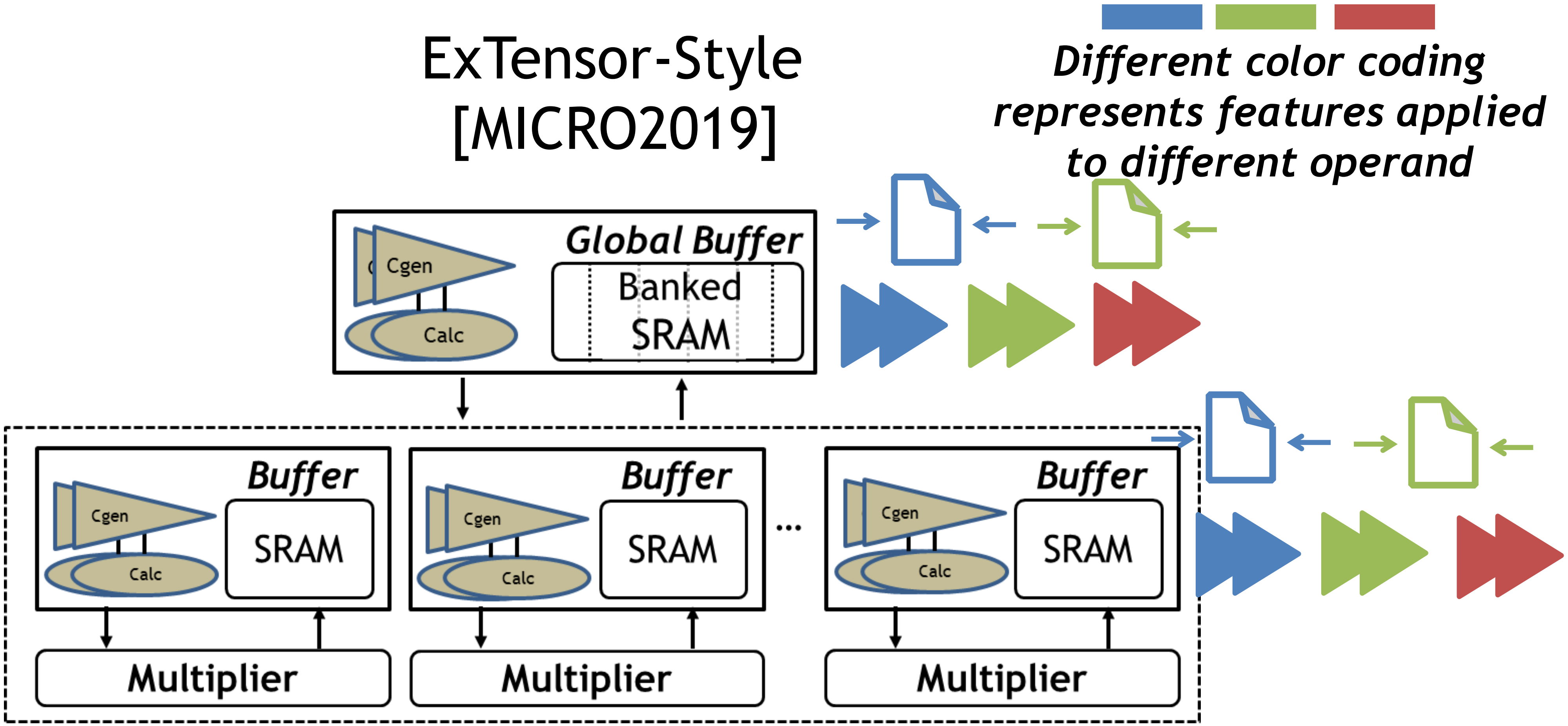


Example Accelerator Architecture Organization

Systematic Descriptions with Various SAF Combinations

ExTensor-Style [MICRO2019]

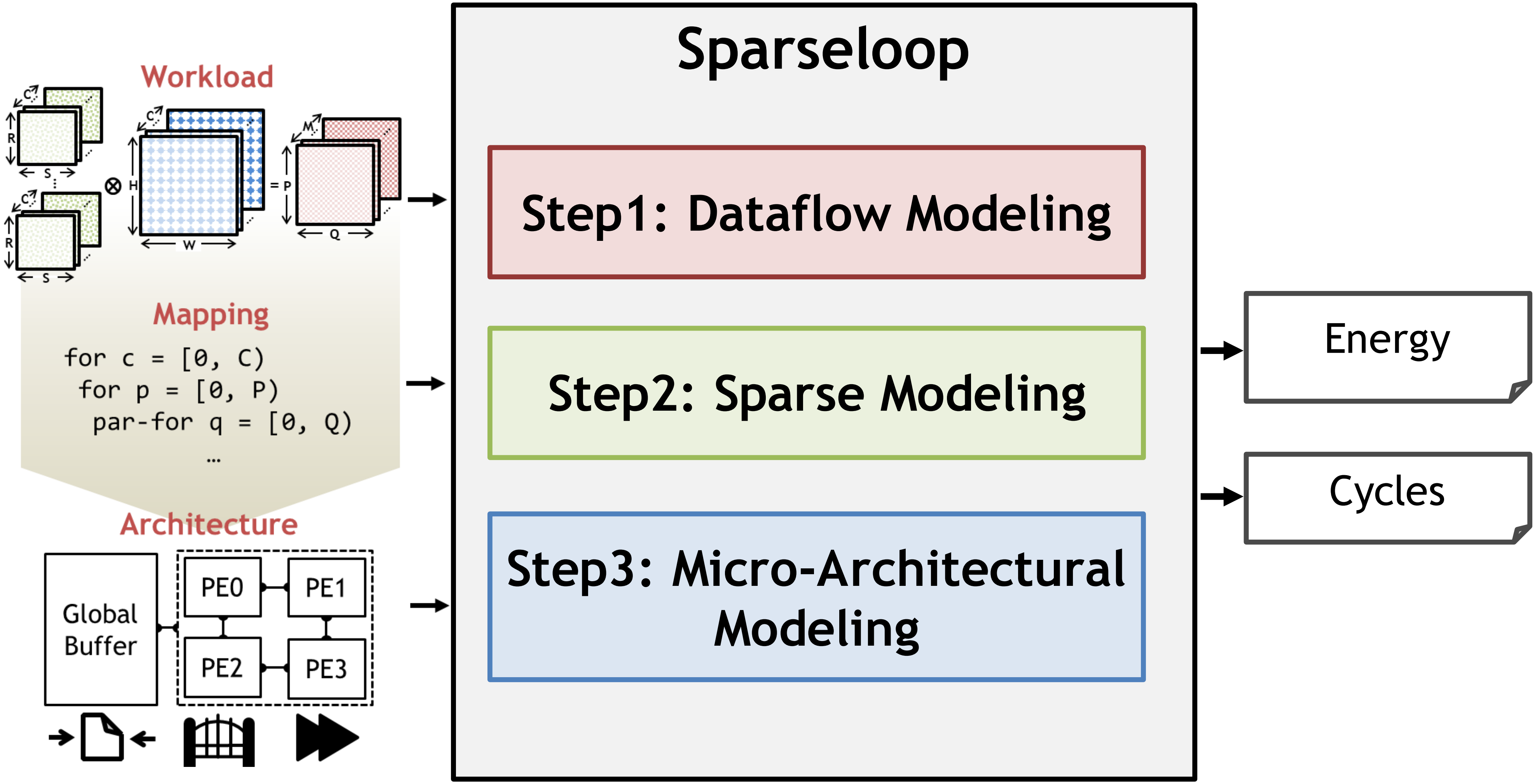
Different color coding represents features applied to different operand



Example Accelerator Architecture Organization

Challenge: Complex Interactions Lead Slow Modeling Speed

Sparseloop Solution: Decoupled Modeling



Workload

Mapping

```
for c = [0, C)
 for p = [0, P)
  par-for q = [0, Q)
   …
```

Architecture

Global Buffer

PE0

PE1

PE2

PE3

Sparseloop

Step1: Dataflow Modeling

Step2: Sparse Modeling

Step3: Micro-Architectural Modeling

Energy

Cycles

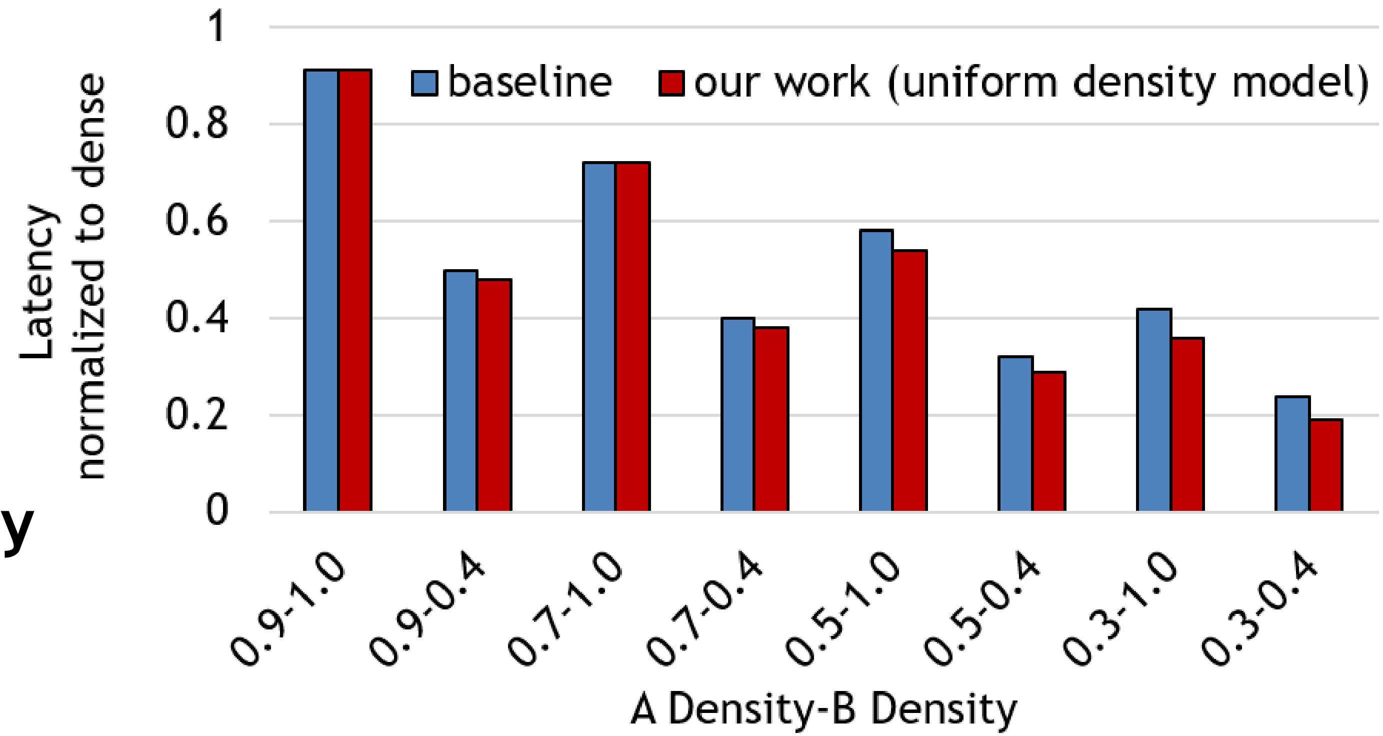
Keep Modeling Complexity Tractable

Modeling Speed and Accuracy

- Speed
  - >2000x faster compared to cycle-level simulations
    - Months -> Hours

- Accuracy
  - Validated on well-known DNN accelerators
    - Maintains relative trends
    - Achieves 0.1% - 8% error in cycle counts and energy consumption

Example DSTC [ISCA21] Validation



More Details in Paper!

- How to build the next-generation sparse tensor core accelerator?
  - *short answer: explore support for different sparsity ratios*

- What happens when we use a sparse DNN accelerator to run much sparser HPC workloads? Or vice versa?
  - *short answer: sparse acceleration features become ineffective for inappropriate workloads*

- …

Summary

- **Sparseloop is a fast, accurate, and flexible analytical modeling framework that enables tensor accelerator design space exploration**
  - **Fast**: achieves >2000x speedup compared to cycle-level simulations
  - **Accurate**: maintains relative trend and achieves 0.1% - 8% error on cycles counts and energy consumption
  - **Flexible**: helps designers understand the critical design trade-offs

- **Resources**
  - Artifact
  - Tutorial at: http://sparseloop.mit.edu/