6.5930/1

Hardware Architectures for Deep Learning

Advanced Technologies (cont'd)

April 24, 2024

Joel Emer and Vivienne Sze

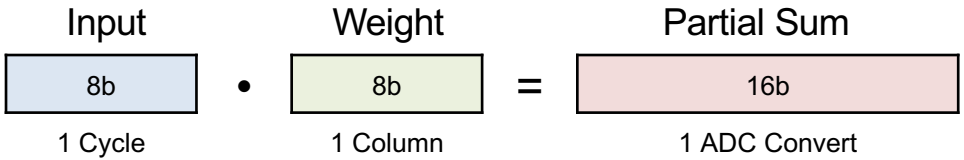
Massachusetts Institute of Technology
Electrical Engineering & Computer Science

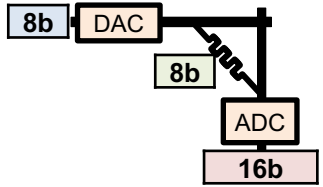
The Titanium Law

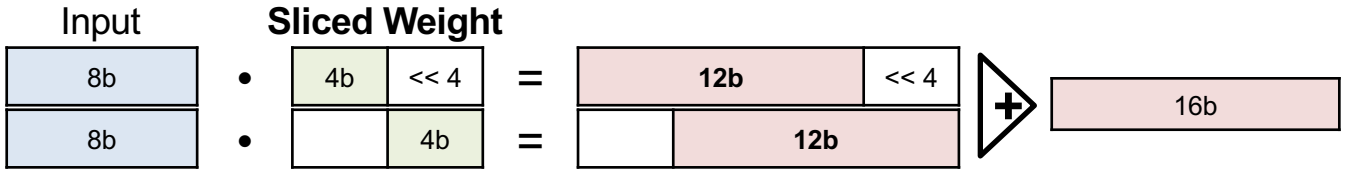
ADC energy is a product of **four** terms

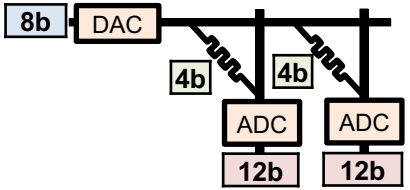
$$\frac{\textbf{ADC Energy}}{\textbf{DNN}} = \overbrace{\frac{\textbf{Energy}}{\textbf{Convert}}}^{\uparrow \text{ exponentially with higher ADC resolution}} \times \underbrace{\frac{\textbf{Converts}}{\textbf{MAC}}}_{\downarrow \text{ with more rows; } \uparrow \text{ with more input/weight slices}} \times \overbrace{\frac{\textbf{MACs}}{\textbf{DNN}}}^{\text{set by the DNN Workload}} \times \underbrace{\frac{1}{\textbf{Utilization}}}_{\geq 1 \text{ based on row utilization}}$$

Use Bit Slicing to Reduce ADC Resolution

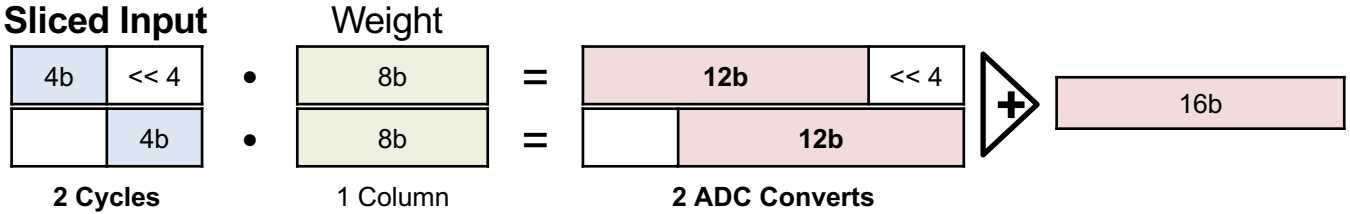


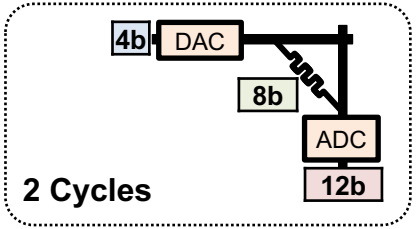






Weight slicing increases **area** and number of **ADC converts**

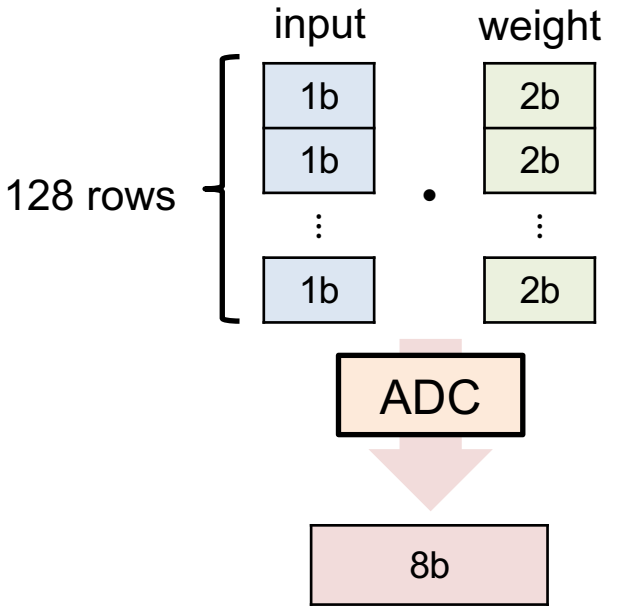




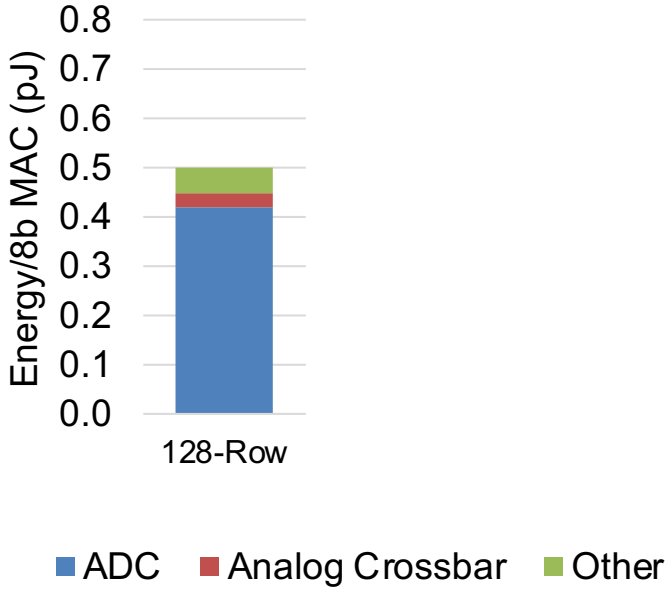
Input slicing increases **time** and number of **ADC converts**

The Titanium Law: Revisit ISAAC

$$\frac{\textbf{\textit{ADC Energy}}}{\textbf{\textit{DNN}}} = \frac{\textbf{\textit{Energy}}}{\textbf{\textit{Convert}}} \times \frac{\textbf{\textit{Converts}}}{\textbf{\textit{MAC}}} \times \frac{\textbf{\textit{MACs}}}{\textbf{\textit{DNN}}} \times \frac{\textbf{1}}{\textbf{\textit{Utilization}}}$$

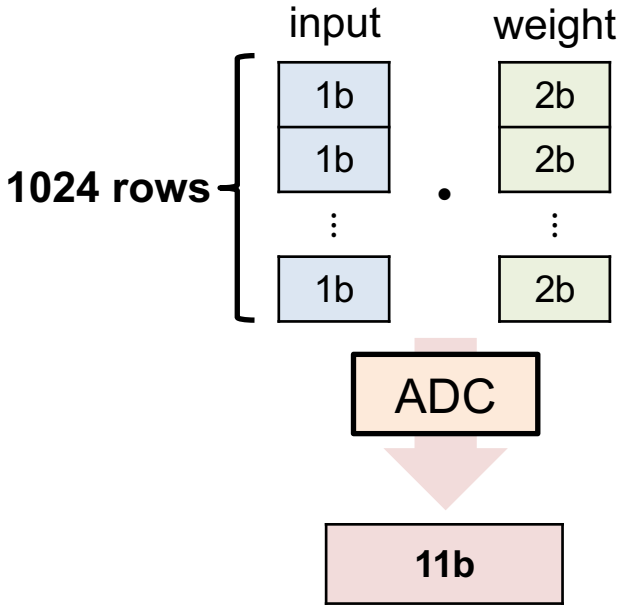


Can we **reduce** ADC energy?

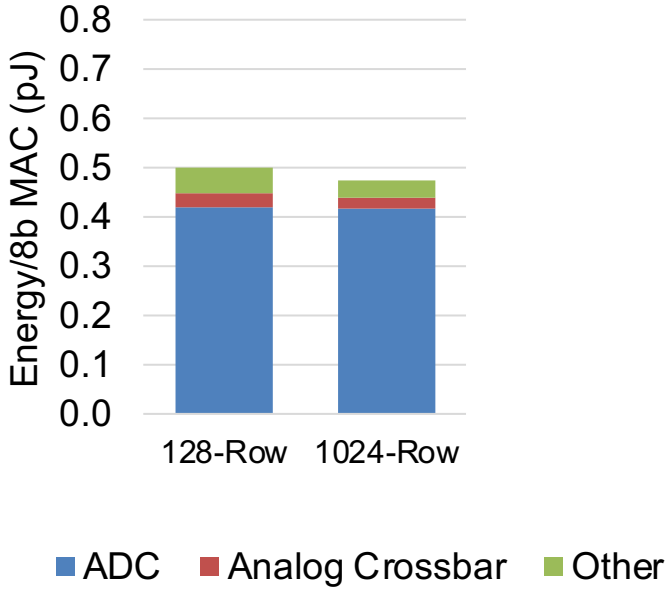


The Titanium Law: Revisit ISAAC

$$\frac{\textbf{\textit{ADC Energy}}}{\textbf{\textit{DNN}}} = \uparrow \frac{\textbf{\textit{Energy}}}{\textbf{\textit{Convert}}} \times \downarrow \frac{\textbf{\textit{Converts}}}{\textbf{\textit{MAC}}} \times \frac{\textbf{\textit{MACs}}}{\textbf{\textit{DNN}}} \times \frac{\textbf{1}}{\textbf{\textit{Utilization}}}$$

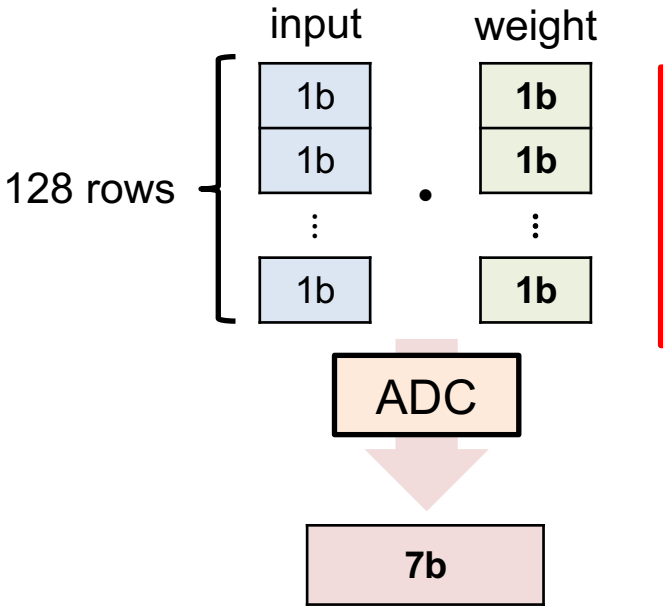


Increase rows
↓
Increase bits

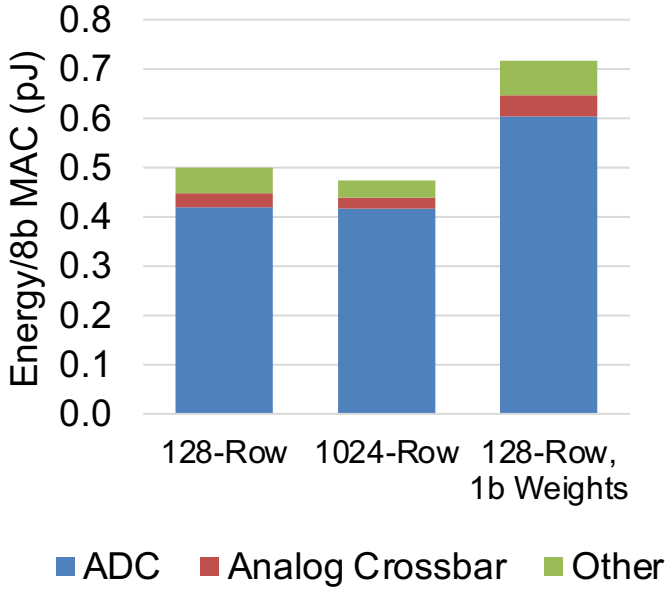


The Titanium Law: Revisit ISAAC

$$\frac{\textbf{\textit{ADC Energy}}}{\textbf{\textit{DNN}}} = \downarrow \frac{\textbf{\textit{Energy}}}{\textbf{\textit{Convert}}} \times \uparrow \frac{\textbf{\textit{Converts}}}{\textbf{\textit{MAC}}} \times \frac{\textbf{\textit{MACs}}}{\textbf{\textit{DNN}}} \times \frac{\textbf{1}}{\textbf{\textit{Utilization}}}$$



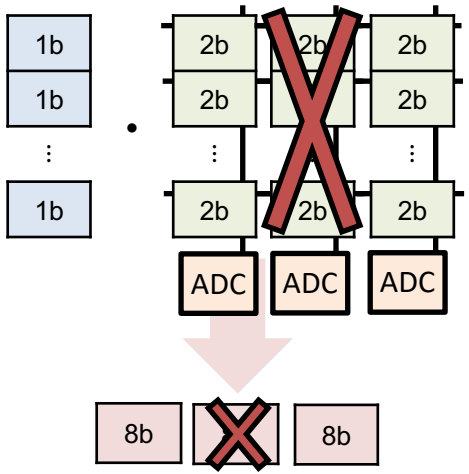
Decrease bits/weight slice
↓
Increase weight slices
↓
Increase ADC converts



How Have Prior Works Escaped These Tradeoffs?

$$\frac{\textbf{\textit{ADC Energy}}}{\textbf{\textit{DNN}}} = \frac{\textbf{\textit{Energy}}}{\textbf{\textit{Convert}}} \times \frac{\textbf{\textit{Converts}}}{\textbf{\textit{MAC}}} \times \downarrow \frac{\textbf{\textit{MACs}}}{\textbf{\textit{DNN}}} \times \frac{\textbf{1}}{\textbf{\textit{Utilization}}}$$

Weight-Count-Limited



Prune weights

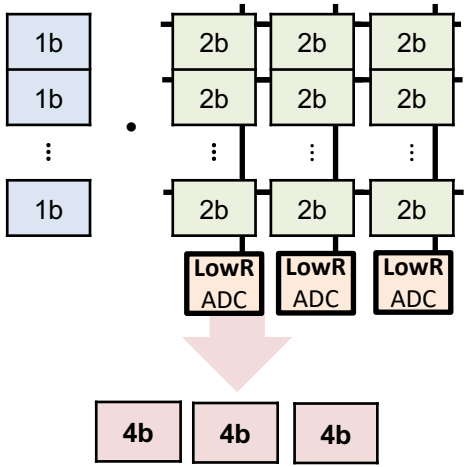
↓

Reduce MACs/DNN

How Have Prior Works Escaped These Tradeoffs?

$$\frac{\textbf{\textit{ADC Energy}}}{\textbf{\textit{DNN}}} = \downarrow \frac{\textbf{\textit{Energy}}}{\textbf{\textit{Convert}}} \times \frac{\textbf{\textit{Converts}}}{\textbf{\textit{MAC}}} \times \frac{\textbf{\textit{MACs}}}{\textbf{\textit{DNN}}} \times \frac{\mathbf{1}}{\textbf{\textit{Utilization}}}$$

Sum-Fidelity-Limited

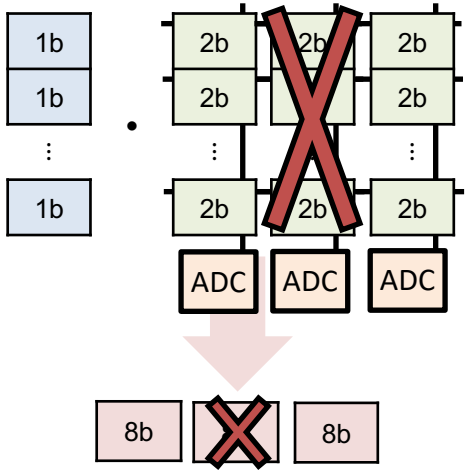


Low-Res ADC
↓
Reduce energy per convert

How Have Prior Works Escaped These Tradeoffs?

$$\frac{\textbf{\textit{ADC Energy}}}{\textbf{\textit{DNN}}} = \frac{\textbf{\textit{Energy}}}{\textbf{\textit{Convert}}} \times \frac{\textbf{\textit{Converts}}}{\textbf{\textit{MAC}}} \times \frac{\textbf{\textit{MACs}}}{\textbf{\textit{DNN}}} \times \frac{\textbf{1}}{\textbf{\textit{Utilization}}}$$

Weight-Count-Limited

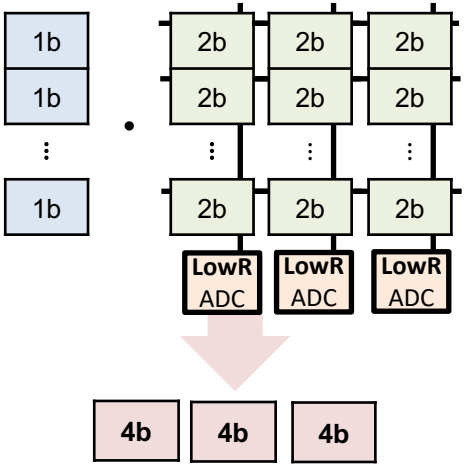


Prune weights

↓

Reduce MACs/DNN

Sum-Fidelity-Limited



Low-Res ADC

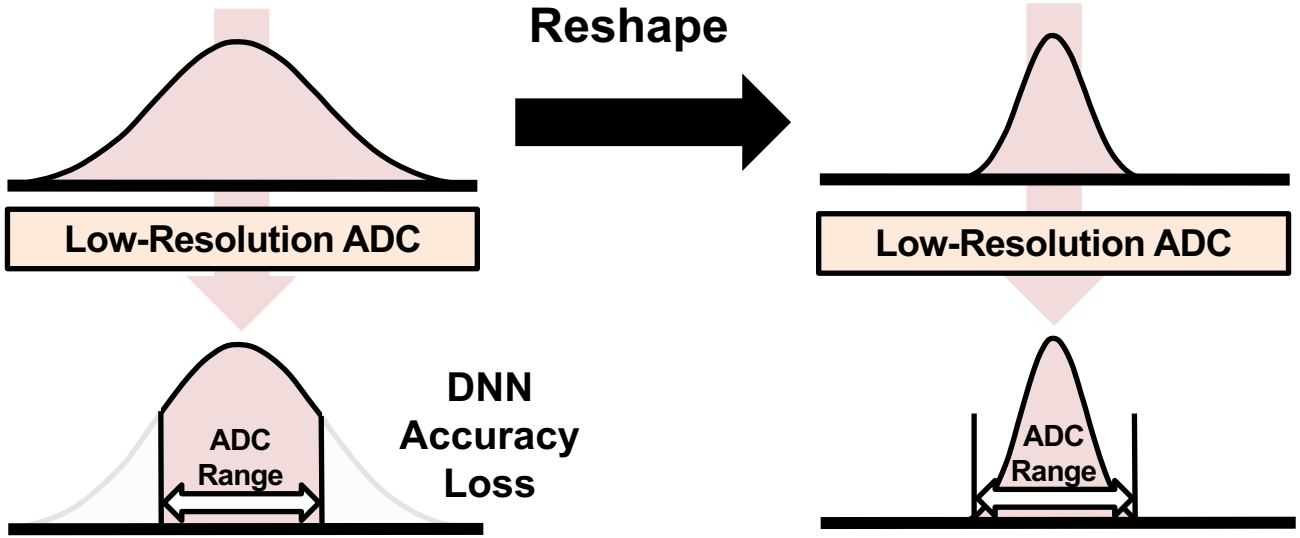
↓

Reduce energy per convert

Both approaches may require **retraining DNN** to preserve accuracy

Reshape Input to ADC to Preserve Accuracy

Distribution of input to ADC (analog column sum)



To preserve accuracy, distribution must fit the ADC range

Reshaping can either be done by changing or retraining DNN or with adaptive hardware that changes analog compute (**RAELLA**)

[**Andrulis**, *ISCA* 2023]

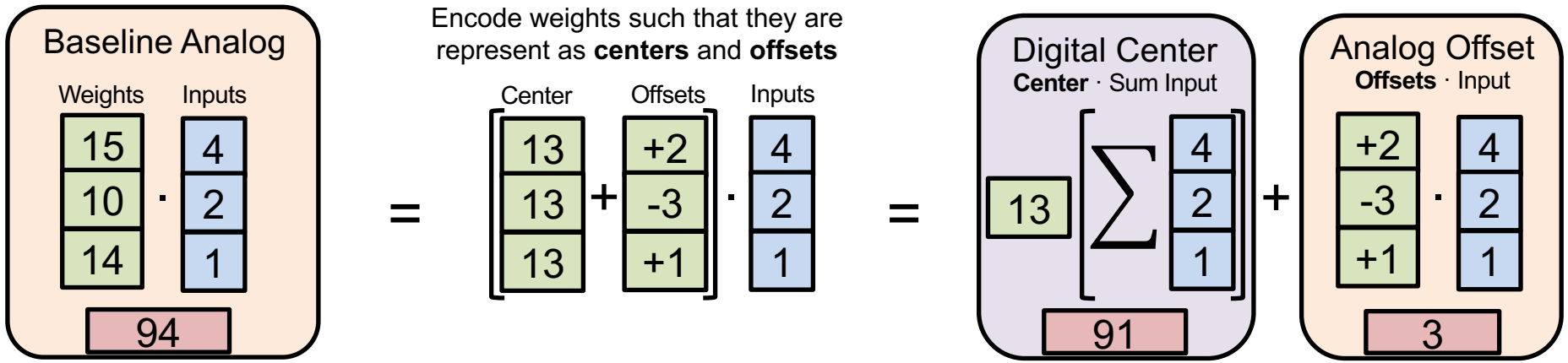
RAELLA's Strategies to Reduce Input to ADC

- **Center+Offset Weight Encoding**
  - Partition compute such that input to ADC smaller and closer to zero
- **Adaptive Weight Slicing**
  - Adapt slicing for each DNN layer to reduce number of ADC converts
- **Dynamic Input Slicing**
  - Dynamically change slicing to reduce number of ADC converts

- Enables ~1000x reduction in range of input to ADC, or 10-bit reduction in ADC resolution

Center + Offset Weight Encoding

Partition computation
Digital calculates high-resolution **center** operations
Analog calculates parallel **offset** operations



Encoding allows input to ADC (output of column sum) to be **smaller and closer to zero**

[**Andrulis**, *ISCA* 2023]

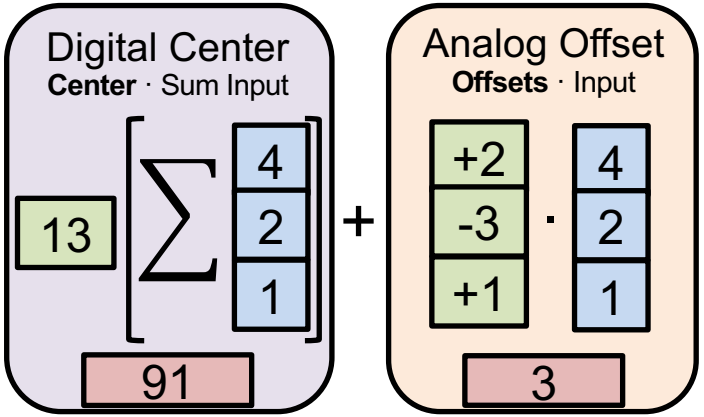
Center + Offset Weight Encoding

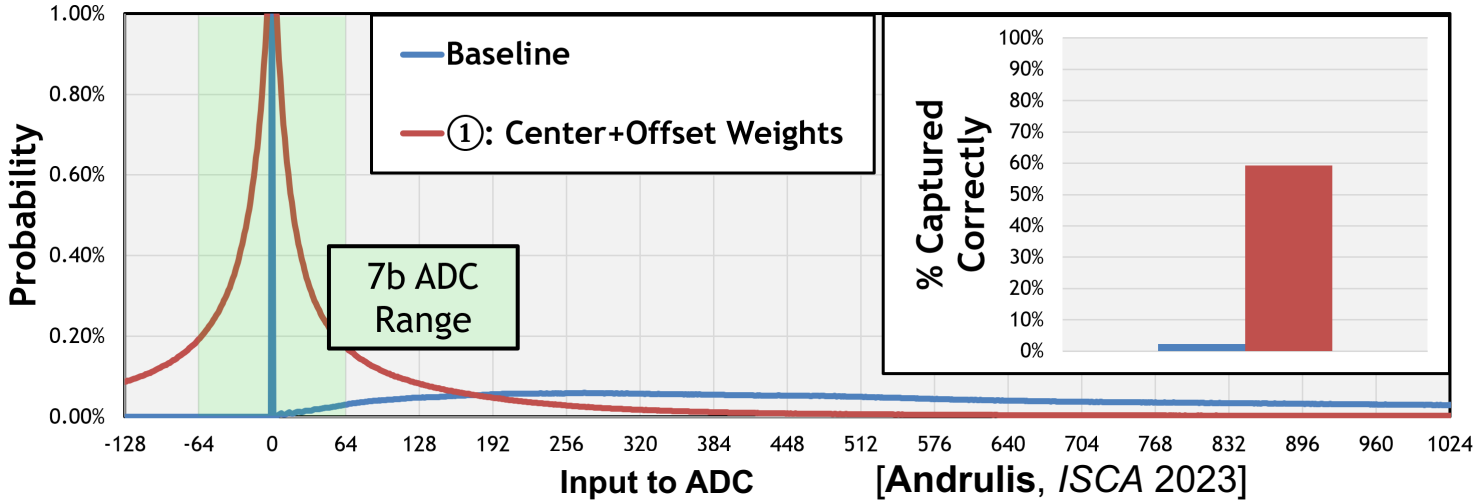
✓ Low-Resolution Analog

High-resolution operations in digital domain

✓ Efficient

Vector-vector operations in analog domain

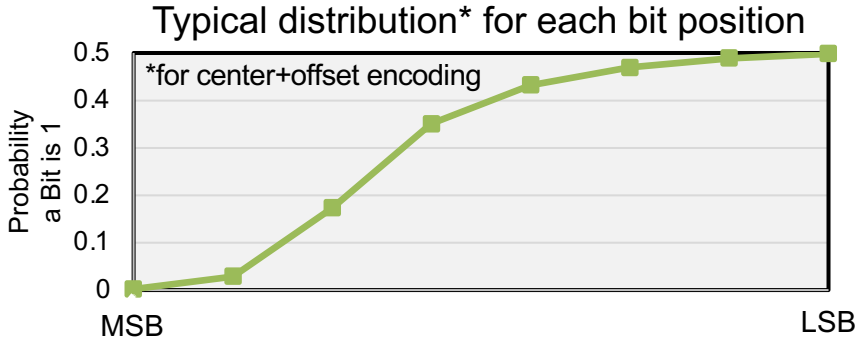




[**Andrulis**, *ISCA* 2023]

Adaptive Weight Slicing

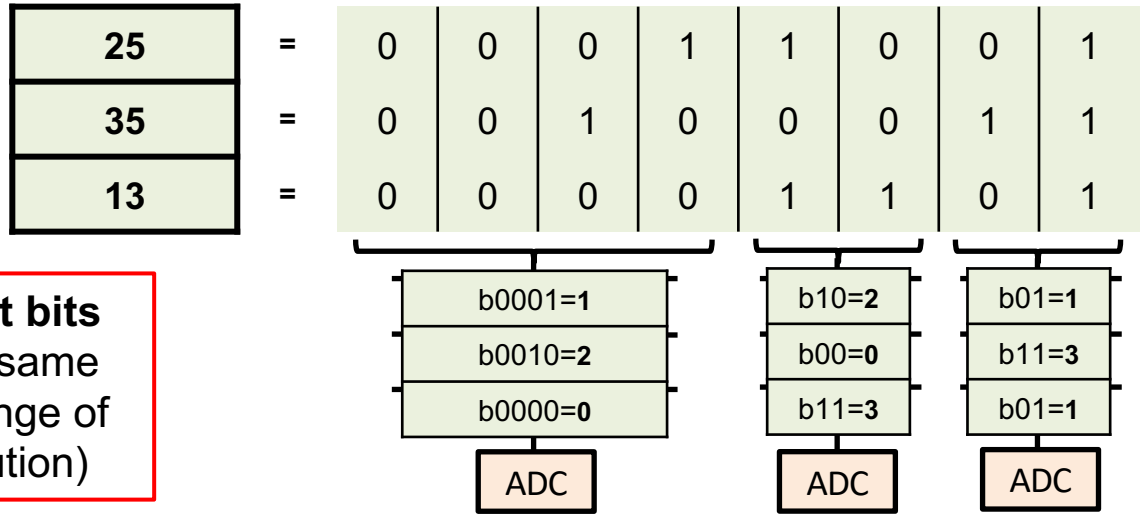
Distribution of input to ADC (output of column sum) depends on distribution of bits at **each bit position in weight**



Reduce slices per weight (**increase** bits per slice)

↓

Reduce ADC converts
Reduce area



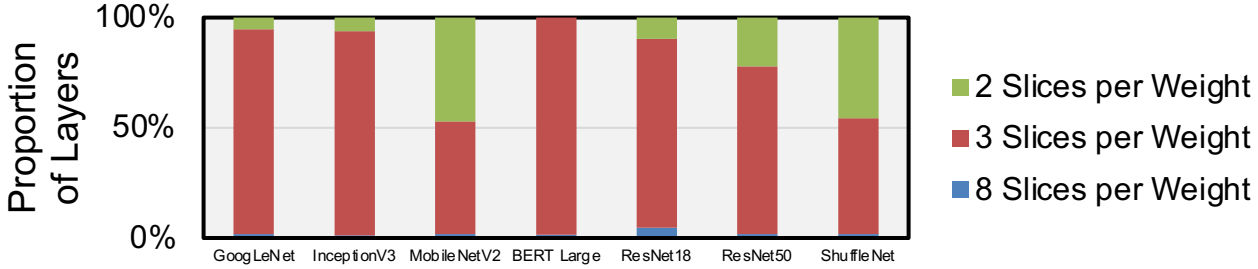
Adaptively merge weight bits with low probabilities into same slice without increasing range of input to ADC (ADC resolution)

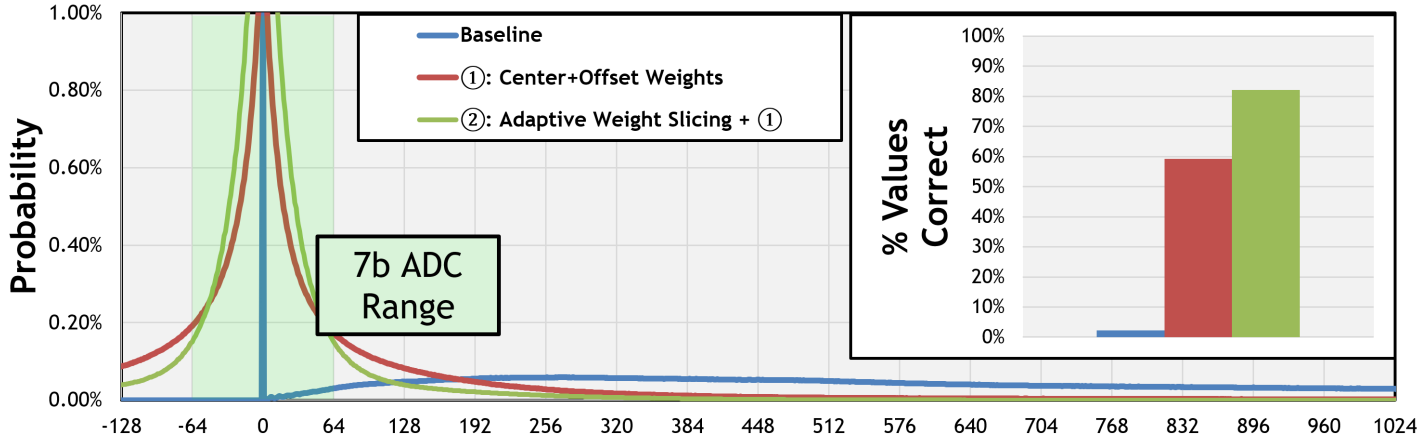
[**Andrulis**, *ISCA* 2023]

Adaptive Weight Slicing

Adapt weight slicing for each layer while preserving correctness

DNN weights are known ahead of time→ Use lightweight preprocessing



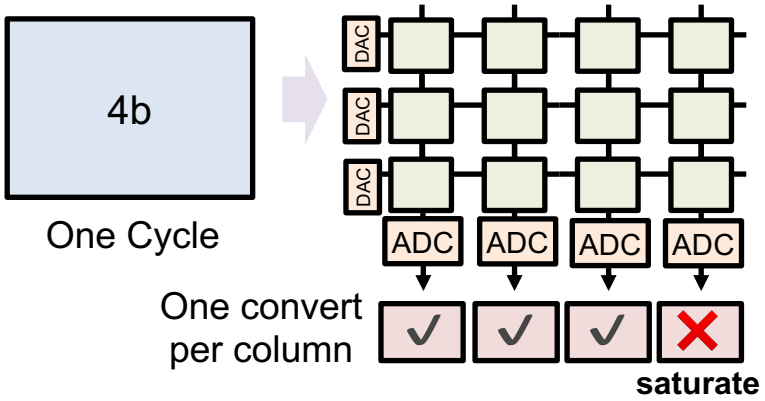


[**Andrulis**, *ISCA* 2023]

Dynamic Input Slicing

- Allocating bits per input slice needs to happen dynamically (at runtime)
- Use **speculation** to allocate many bits and **recovery** when saturate

Step 1: Speculation

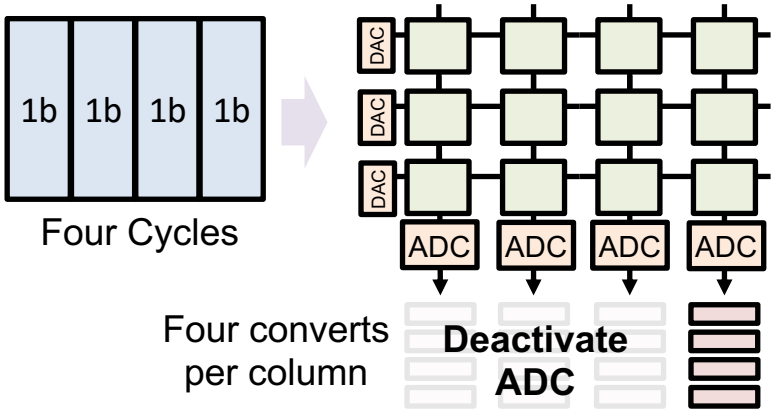


Reduce slices per input
(**increase** bits per slice)

↓

Reduce ADC converts
Reduce cycles

Step 2: Recovery

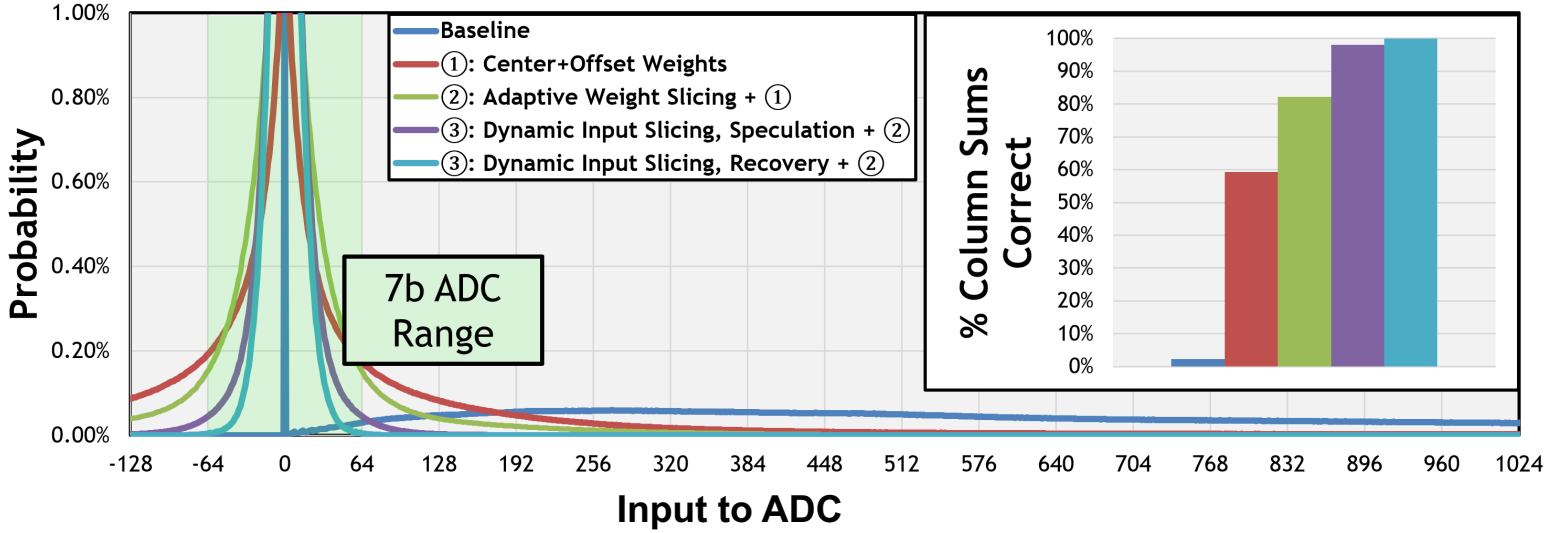


Speculation and recovery **takes more cycles** than no speculation (recovery only)

[**Andrulis**, *ISCA* 2023]

Dynamic Input Slicing

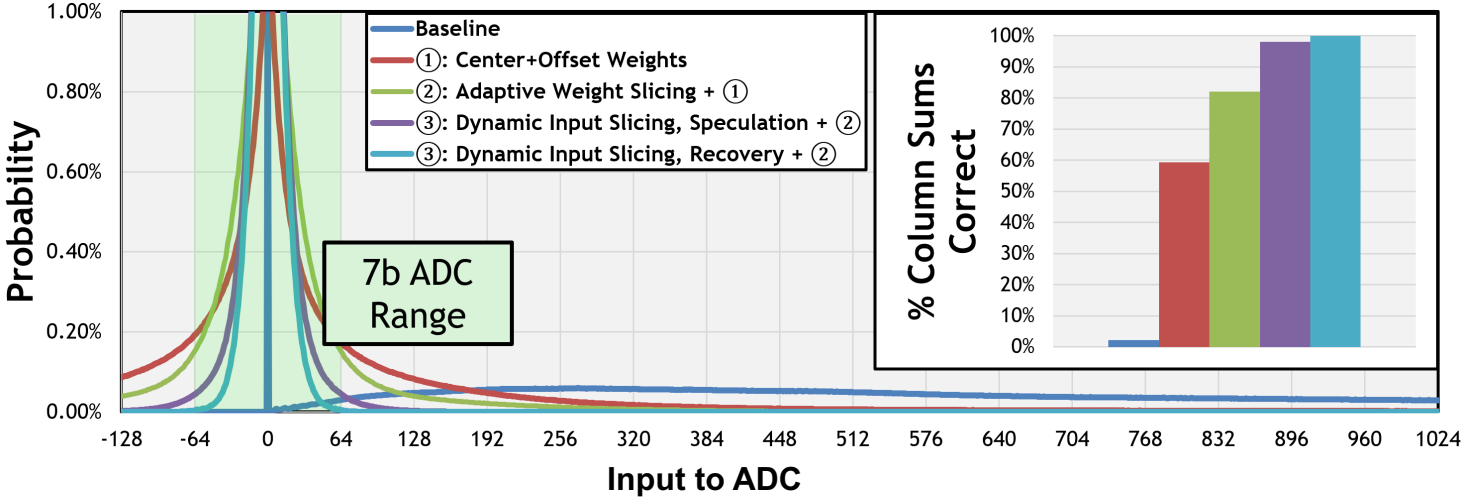
- Comparing **speculation** for 8-bit input (one 4-bit slice + two 2-b slices) and **recovery** (eight 1-b slices) versus **only recovery** slices (eight 1-b slices)
  - Reduces ADC converts by 60%
  - Adds **three extra cycles**
  - **In summary, speculation improves ADC energy efficiency at cost of speed**



[**Andrulis**, *ISCA* 2023]

RAELLA: Reshape Distributions of Input to ADC

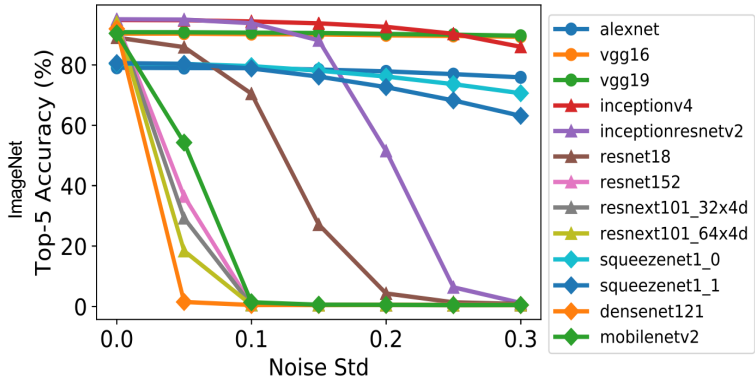
- Makes analog operations produce low-resolution results
  - Center+Offset Weight Encoding, Adaptive Weight Slicing, Dynamic Input Slicing
- Enables more compute per ADC convert while using lower-resolution ADCs
  - Improves energy efficiency by 3.9x and throughput by 1.8x compared to iso-area ISAAC
- Maintains DNN accuracy without changing DNN or retraining

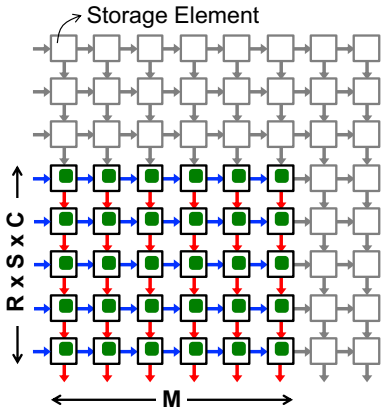


[**Andrulis**, *ISCA* 2023]

Designing DNN Models for CiM

- Designing DNNs for CiM may differ from DNNs for digital processors
- Highest accuracy DNN on digital processor may be different on CiM
  - Accuracy drops based on robustness to non-idealities
- Reducing number of weights is less desirable
  - Since CiM is weight stationary, may be better to reduce number of activations
  - PIM tend to have larger arrays → fewer weights may lead to low utilization on CiM
- Current trend is deeper and smaller filters
  - CiM may prefer to have shallower and larger filters

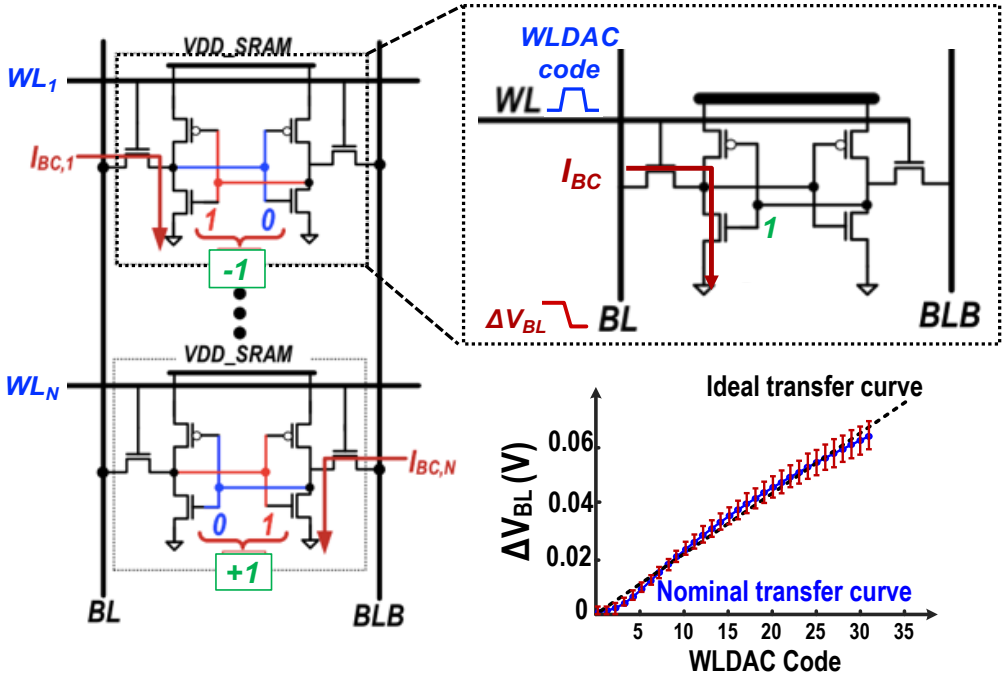




[**Yang**, *IEDM* 2019]

CiM Using SRAM Bit Cell

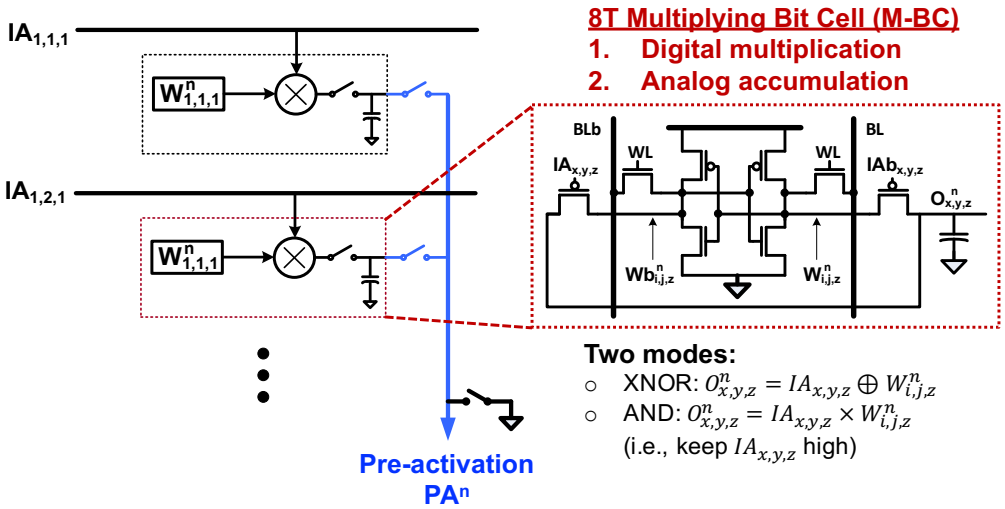
- Multiplication uses I-V relationship of access transistor (WL) and stored value in bit-cell
  - Assumes binary weights and multi-bit input activation
- Addition using **current addition** on bit line (BL)
  - Limited by nonlinearity and sensitivity to variations



[**Verma**, *SSCS* 2019]

CiM Using SRAM Bit Cell

- Binary multiplication (AND or XNOR) using access transistor (WL) and stored value in bit-cell
  - Explicit capacitor to store charge
- Addition using **charge sharing** on bit line (BL)
  - Better linearity and matching



[**Verma**, *SSCS* 2019]

Using Charge Sharing for Addition

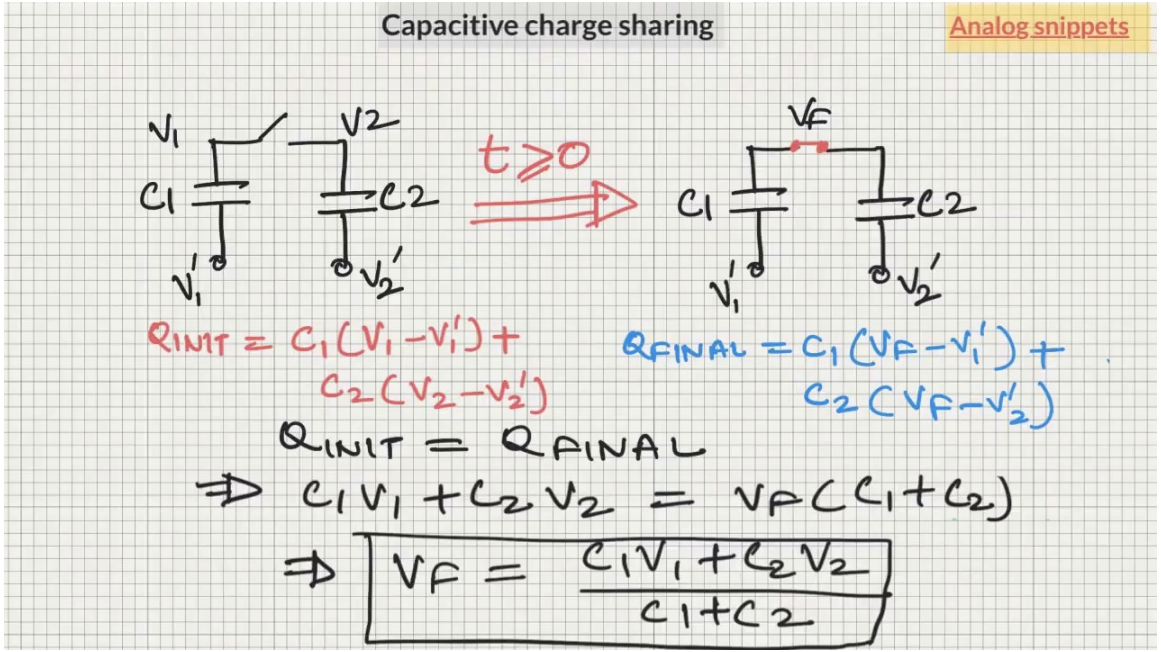


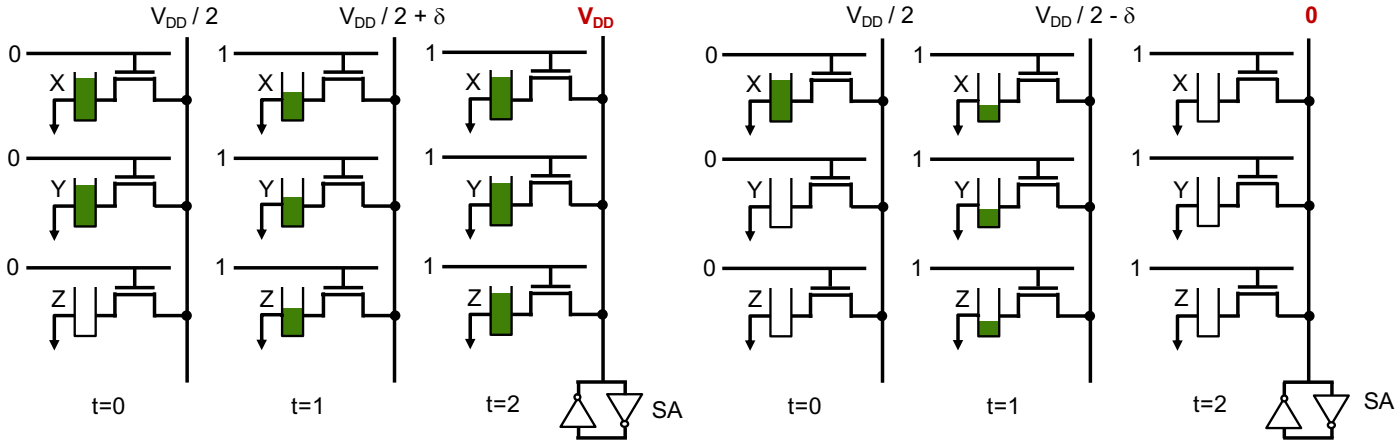
Image Source: https://www.youtube.com/watch?v=XRQ_Xldr2nk

If $C_1=C_2$, $V_f = ½ (V_1 + V_2)$, which is a scaled value of the sum (addition)

CiM Using DRAM

Performs bit-wise operations using charge sharing

If Z=0, perform AND(X, Y)



AND (X=1, Y=1) = 1

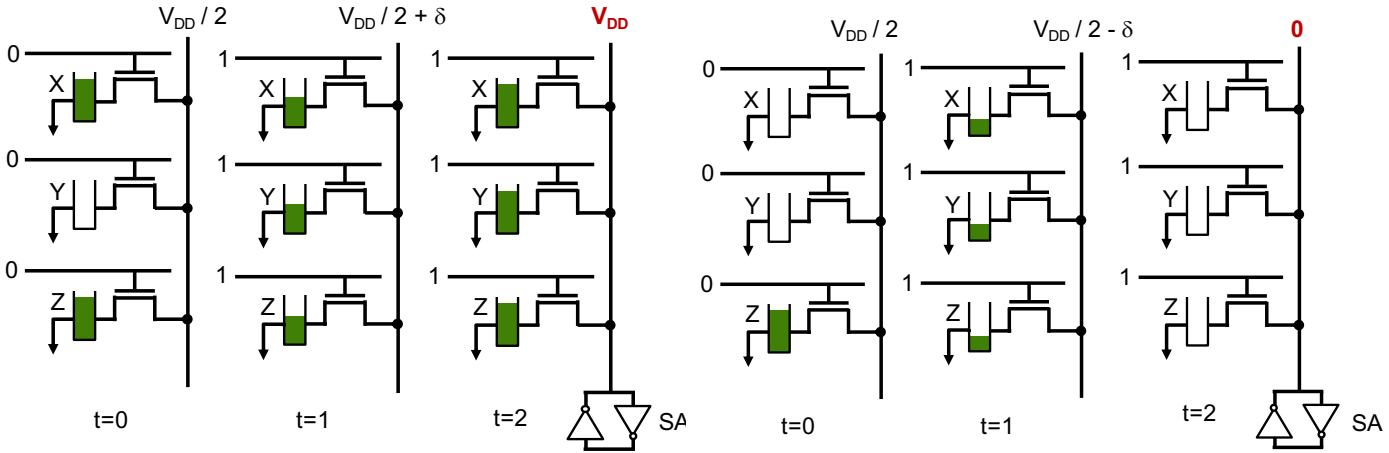
AND (X=1, Y=0) = 0

[Ambit, *MICRO* 2017]

CiM Using DRAM

Performs bit-wise operations using charge sharing

If Z=1, perform OR(X, Y)



OR (X=1, Y=1) = 1

OR (X=0, Y=0) = 0

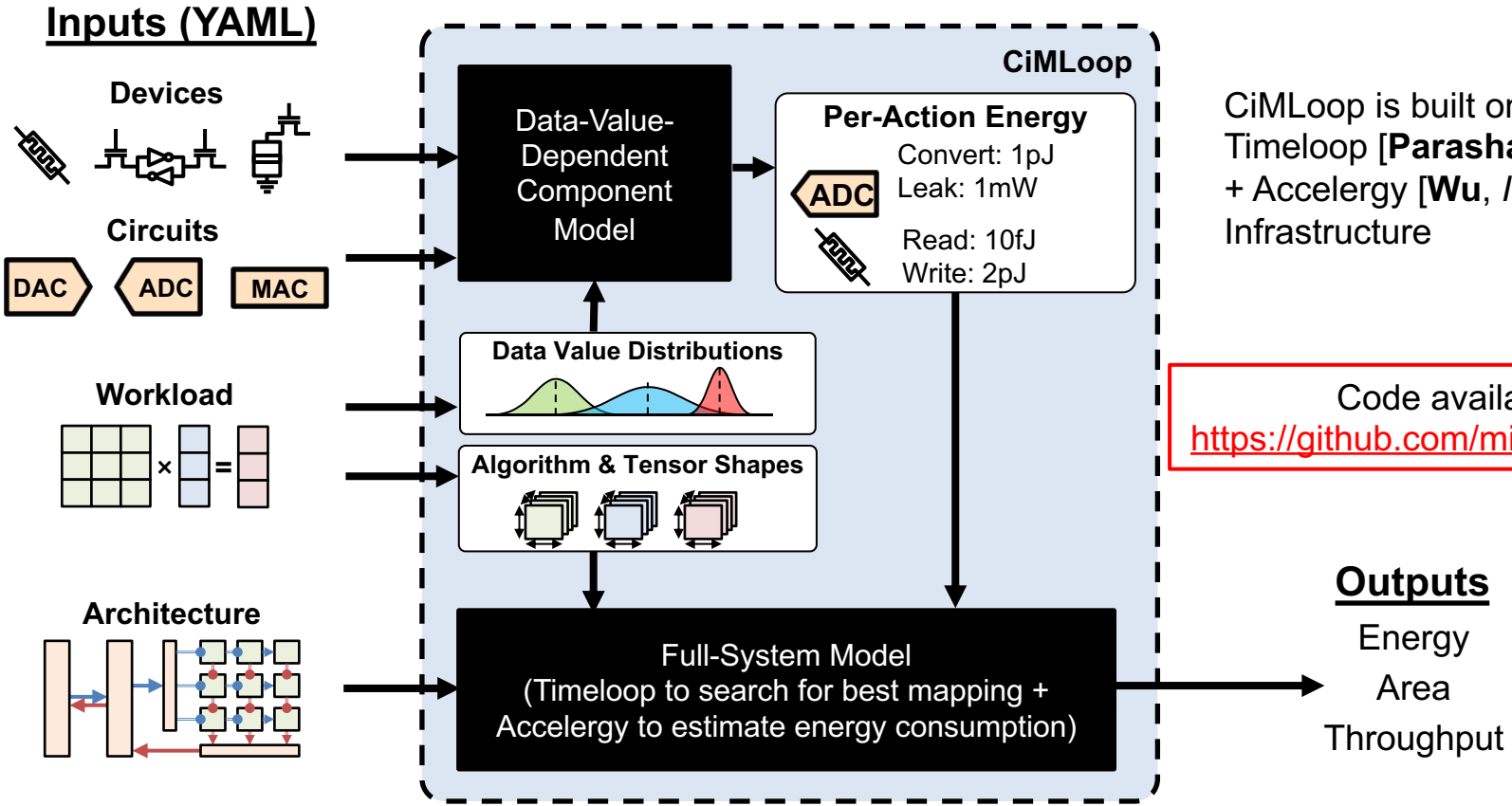
Takes multiple cycles to built up to a multiplication.
However, can perform many operations in parallel (bus width of DRAM)

CiM Research Spans Full Stack

- **Devices:** The components forming each memory cell (e.g., SRAM, DRAM, ReRAM, STT-RAM)
- **Circuits:** The components performing computation, analog/digital conversion, storage, data movement, and other actions
- **Architecture:** The organization of components into a larger system (e.g., the number of each component and how components are connected)
- **Workload:** The DNN to be run, which we model as a series of extended-Einsum operations with tensors of varying shapes and values
- **Mapping:** The temporal and spatial scheduling of the workload onto the system

Need for modeling tool to enable apple-to-apple comparison and design space exploration → **CiMLoop** (used in Lab 5)

CiMLoop



CiMLoop is built on the Timeloop [**Parashar**, *ISPASS* 2019] + Accelergy [**Wu**, *ICCAD* 2019] Infrastructure

Code available at https://github.com/mit-emze/cimloop

Outputs

Energy

Area

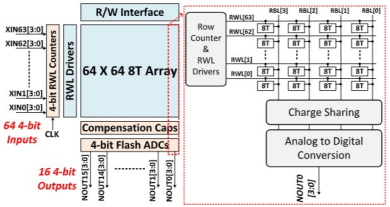
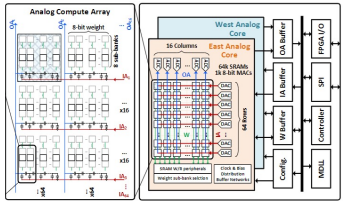
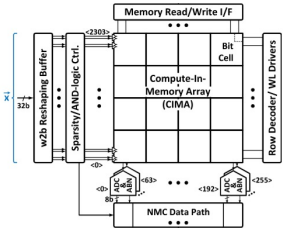
Throughput

[**Andrulis**, *ISPASS* 2024]

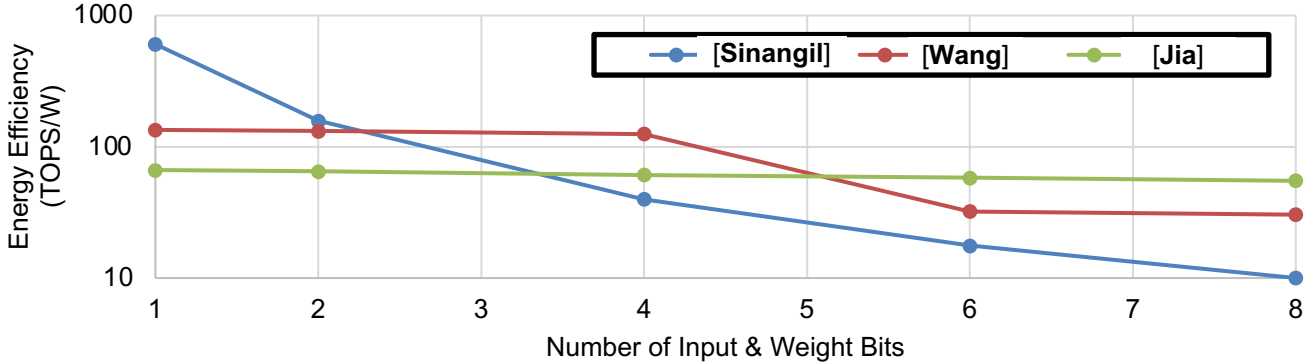
CiMLoop

- **Flexibility**
  - A flexible specification that lets users describe, model, and map workloads to both circuits and architecture
- **Accuracy**
  - A data-value-dependent energy model that captures the interaction between DNN operand values, data representations, and analog/digital values
  - Estimated values from model are within 8% of values reported for measured designs
- **Speed**
  - A fast statistical model that uses the average energy per component action for constant runtime w.r.t. number of components and amortizes overhead across mappings
  - Enables orders-of-magnitude speed up relative to other high-accuracy models

Example: Apples-to-Apples Comparison

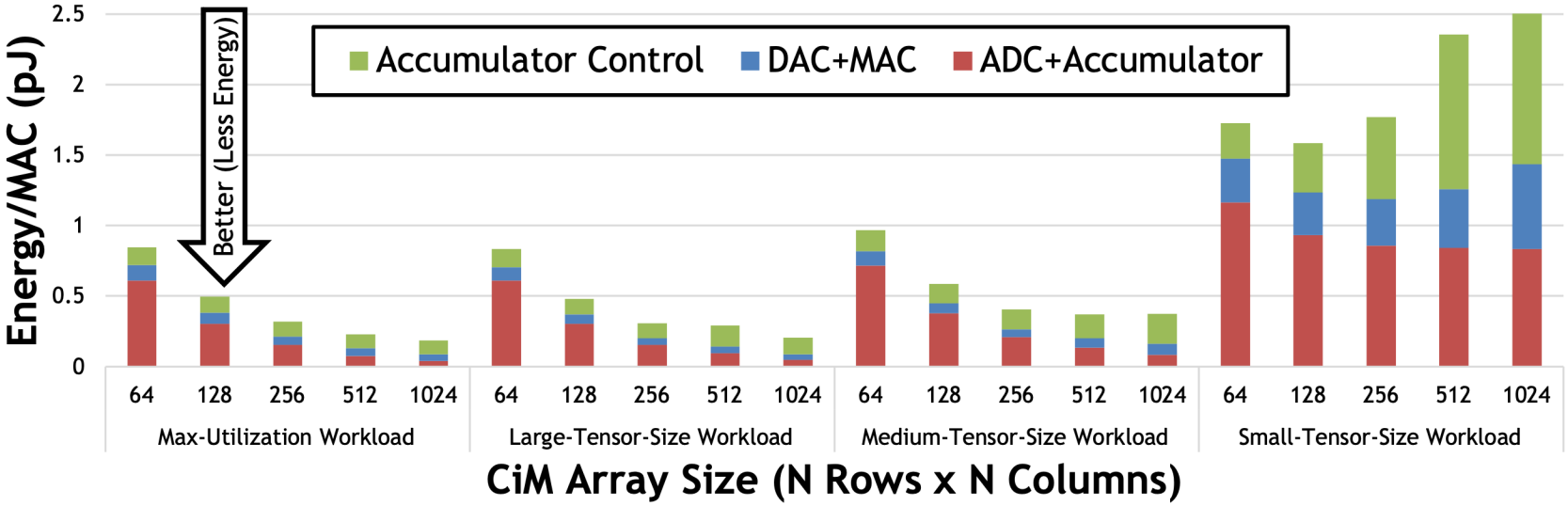
	[**Sinangil**, *JSSC* 2021]	[**Wang**, *VLSI* 2022]	[**Jia**, *JSSC* 2020]
Macro			
Technology Node	7nm	22nm	65nm
ADC Type	4b Flash	8b SAR	8b SAR
Memory Device	6T SRAM	8T SRAM + Capacitor	6T SRAM

Compare Designs:
Same technology, ADC, device for all macros



[**Andrulis**, *ISPASS* 2024]

Example: Design Space Exploration



Explore array size (architecture) and DNN shapes (workload)

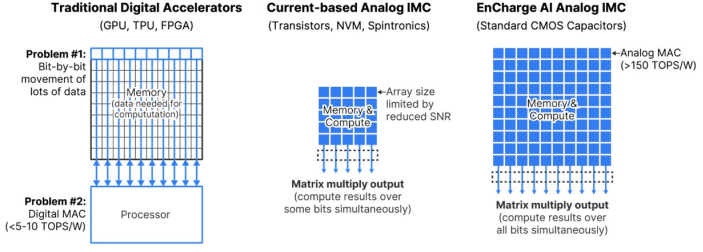
[**Andrulis**, *ISPASS* 2024]

Companies doing Analog CiM

EnCharge AI | Technology | About Us | News & Publications | Events | Careers | Get Started

In-Memory Computing (IMC)

In-memory computing greatly enhances compute efficiency and reduces data movement.

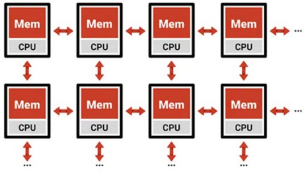


MYTHIC | PRODUCTS | TECHNOLOGY | MARKETS | BLOG | COMPANY

Technology » Compute-in-Memory

Compute-in-Memory

Boosting memory capacity and processing speed



Compute-in-Memory IC

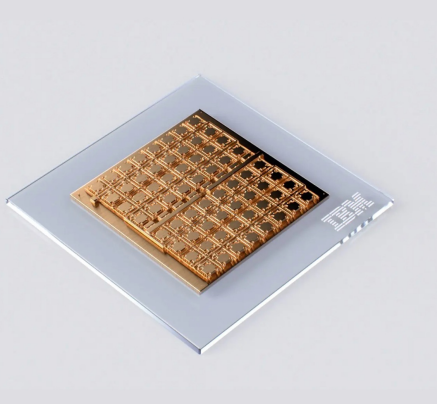
Today's most common computing architectures are built on assumptions about how memory is accessed and used. These systems assume that the full memory space is too large to fit on-chip near the processor, and that we do not know what memory will be needed at what time. To address

IBM | Research | Focus areas | Blog | Publications | Careers | About

Home
↳ Projects

Analog AI

Making Deep Neural Network systems more capable and energy-efficient.



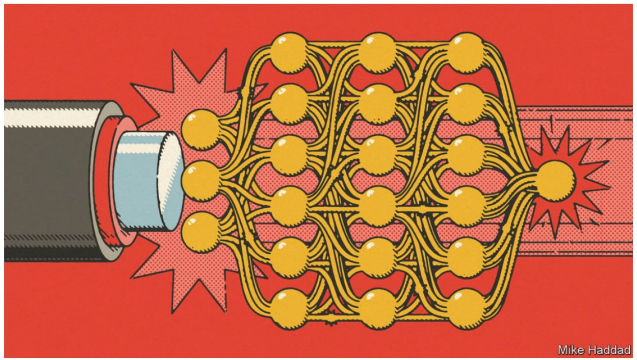
Compute with Light

The Economist ≡ Menu | Weekly edition | Search

Science & technology | Information technology

Artificial intelligence and the rise of optical computing

Photonic data-processing is well-suited to the age of deep learning



Dec 20th 2022 Save Share Give

MODERN INFORMATION technology (IT) relies on division of labour. Photons carry data around the world and electrons process them. But, before optical fibres, electrons did both—and some people hope to complete the transition by having photons process data as well as carrying them.

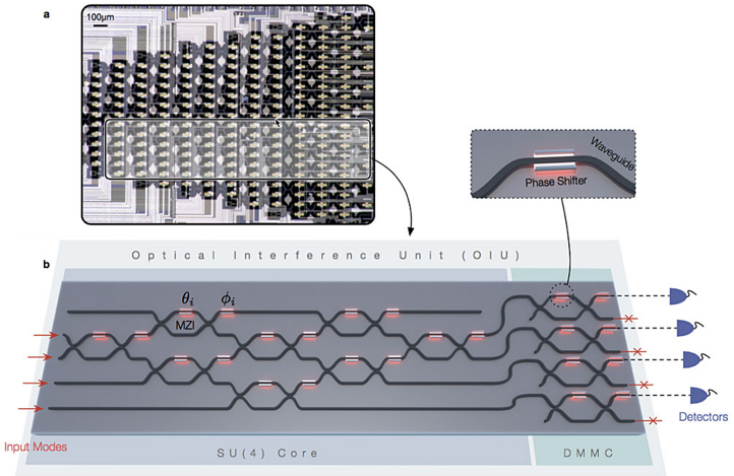
"Unlike electrons, photons (which are electrically neutral) can cross each others' paths without interacting, so glass fibres can handle many simultaneous signals in a way that copper wires cannot. **An optical computer could likewise do lots of calculations at the same time.** Using photons reduces power consumption, too. **Electrical resistance generates heat, which wastes energy. The passage of photons through transparent media is resistance-free.**"

https://www.economist.com/science-and-technology/2022/12/20/artificial-intelligence-and-the-rise-of-optical-computing

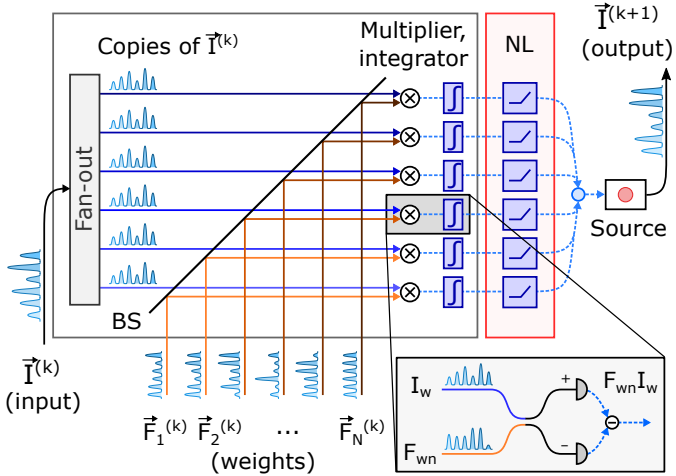
Compute with Light

Matrix Multiplication in the Optical Domain

- Cost of moving a photon can be **independent** of distance
- Multiplication can be performed **passively**



[**Shen**, *Nature Photonics* 2017]



[**Bernstein**, *CLEO* 2020]

Compute with Light

WILL KNIGHT BUSINESS 03.10.2021 07:00 AM

This Chip for AI Works Using Light, Not Electrons

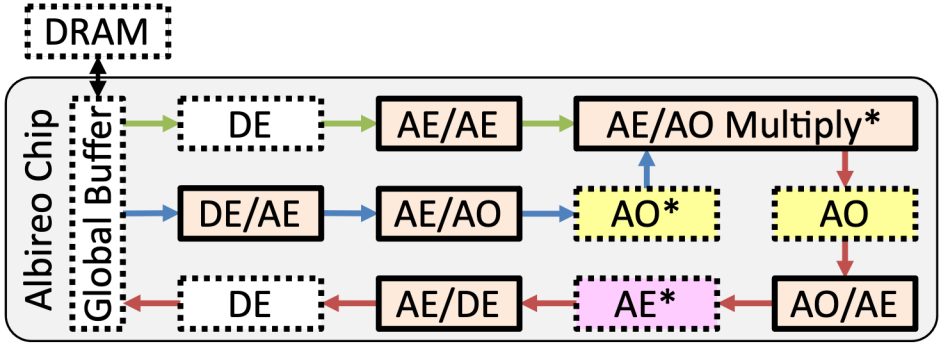
Lightmatter says the computing and power demands of complex neural networks need new technologies like these to keep up.

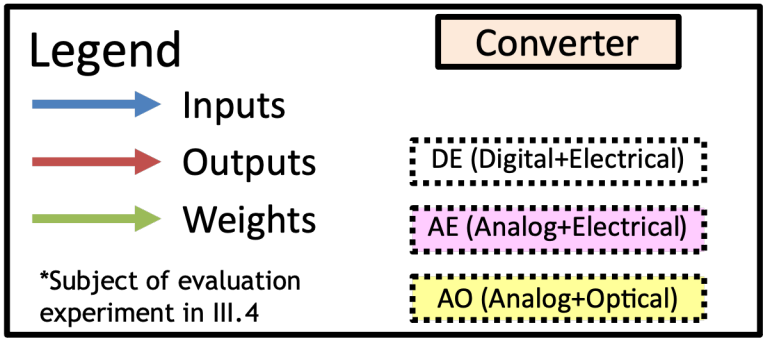
"…chip runs **1.5 to 10 times** faster than a top-of-the-line Nvidia A100 AI chip,

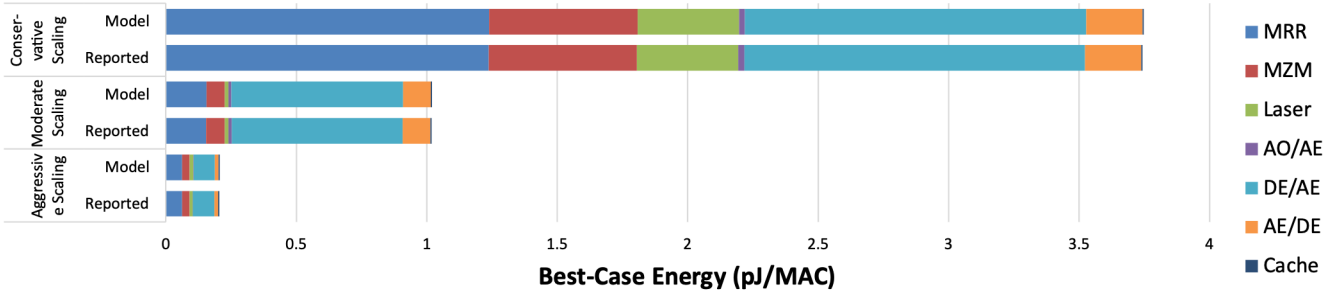
Running a natural language model called BERT, for example, Lightmatter says Envise is **five times faster** than the Nvidia chip; it also consumes **one-sixth of the power**"

https://www.wired.com/story/chip-ai-works-using-light-not-electrons/

CiMLoop for Photonics Modeling







Last year's final project in 6.5930 involved modeling and validation

[**Andrulis**, *ISPASS* 2024]

Summary

- Cross-layer design critical for providing additional efficiency improvements
- For DNN processing using Advanced Technologies, it is important to factor in device and circuit limitations into the architecture
- Textbook Chapter 10
  - https://doi.org/10.1007/978-3-031-01766-7
- Other References
  - Y. N. Wu, V. Sze, J. S. Emer, "An Architecture-Level Energy and Area Estimator for Processing-In-Memory Accelerator Designs," *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS),* April 2020 [paper PDF | code github]
  - T. Andrulis, J. Emer, V. Sze, "RAELLA: Reforming the Arithmetic for Efficient, Low-Resolution, and Low-Loss Analog PIM: No Retraining Required!," *International Symposium on Computer Architecture (ISCA)*, June 2023 [PDF]
  - T. Andrulis, J. Emer, V. Sze, "CiMLoop: A Flexible, Accurate, and Fast Compute-In-Memory Modeling Tool," *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, May 2024