6.5930/1

Hardware Architectures for Deep Learning

Advanced Technologies

March 9, 2026

Joel Emer and Vivienne Sze

Massachusetts Institute of Technology
Electrical Engineering & Computer Science

Advanced Storage Technology

- **Bring compute and memory closer together** to reduced the cost of data movement
- **Processing/Compute Near Memory** *(a.k.a. near-data processing)*
  - Embedded DRAM (eDRAM)
    - Increase on-chip storage capacity
  - 3D Stacked DRAM (e.g., Hybrid Memory Cube (HMC), High Bandwidth Memory (HBM))
    - Increase memory bandwidth
- **Processing/Compute In Memory** *(a.k.a. in-memory computing)*
  - Static Random Access Memories (SRAM), Dynamic Random Access Memories (DRAM), and Non-Volatile Memories (NVM)
  - Processing ***integrated into*** memory (more aggressive form of processing near memory) versus processing ***using*** memory (using memory device to perform compute)

SRAM (kB – MB)

- SRAM accounts for majority of on-chip storage
  - SRAM is significant portion of chip area
- Usually, 6 transistors per bit-cell

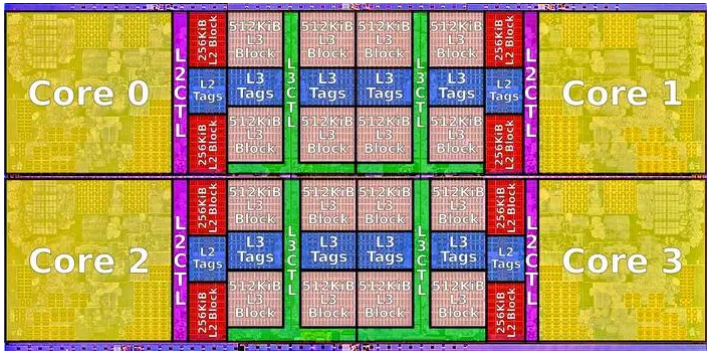
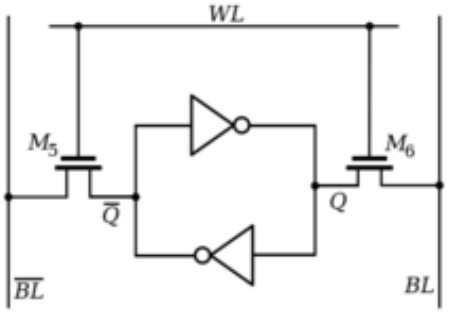
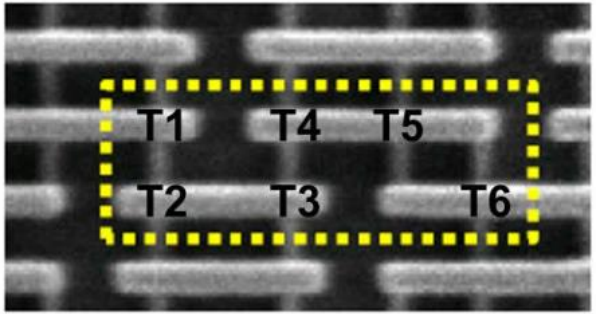


Image Source: https://en.wikichip.org/wiki/amd/microarchitectures/zen

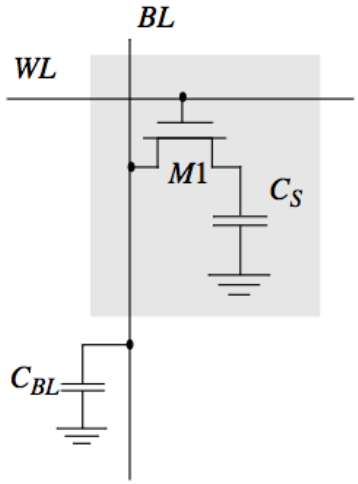


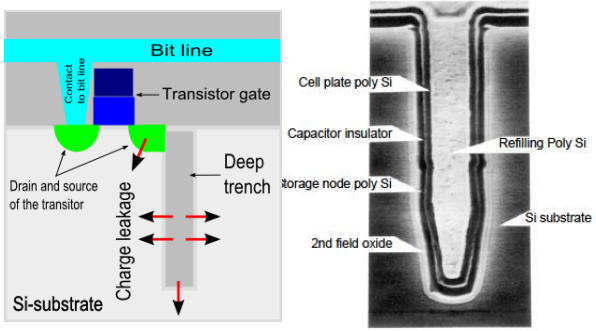


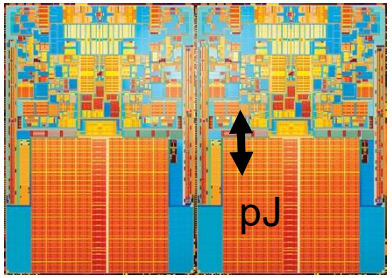
Bit cell size
0.75um^2 in 14nm

DRAM (GB)

- Higher density than SRAM
  - 1 transistor per bit-cell
  - Needs periodic refresh
- Special device process
  - Usually off-chip (except eDRAM – which is pricey!)
  - Off-chip interconnect has much higher capacitance





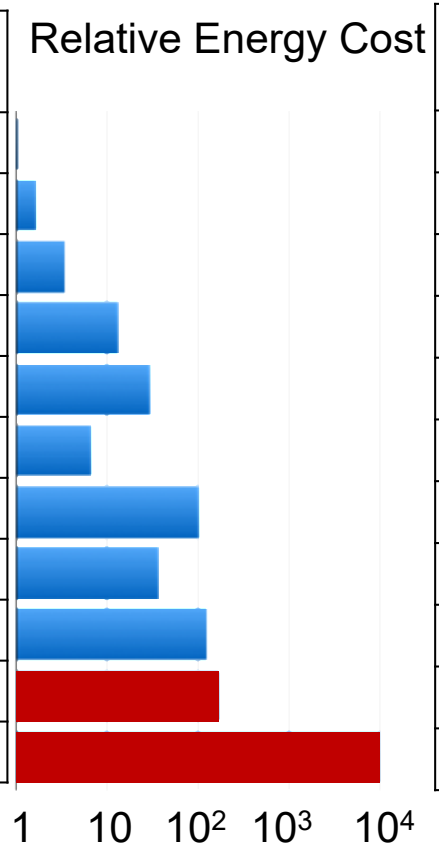


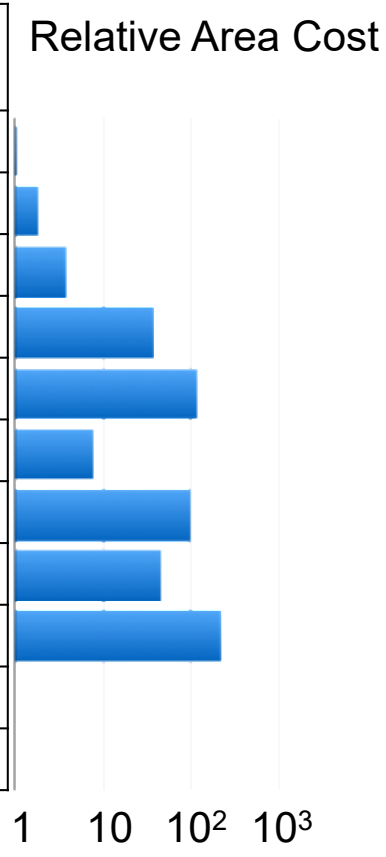
nJ

Sze and Emer

Memory Access Cost for DRAM is Expensive

Operation:	Energy (pJ)	Area (∝m²)
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FP Add	0.4	1360
32b FP Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FP Mult	1.1	1640
32b FP Mult	3.7	7700
32b SRAM Read (8KB)	5	N/A
32b DRAM Read	640	N/A



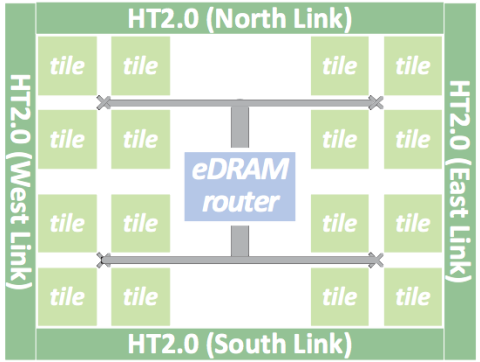


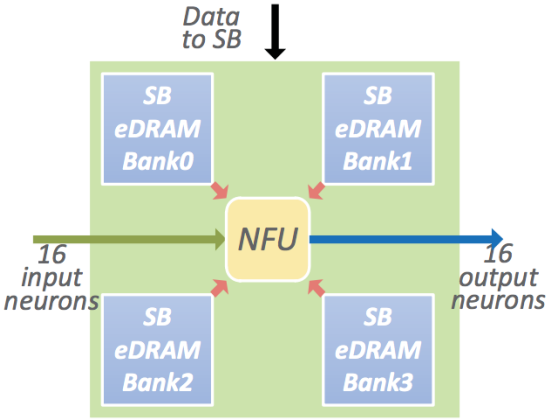
[**Horowitz**, *ISSCC* 2014]

eDRAM (DaDianNao)

- Advantages of eDRAM
  - 2.85x higher density than SRAM
  - 321x more energy-efficient than DRAM (DDR3)
- Store weights in eDRAM (36MB)
  - Target fully connected layers since dominated by weights

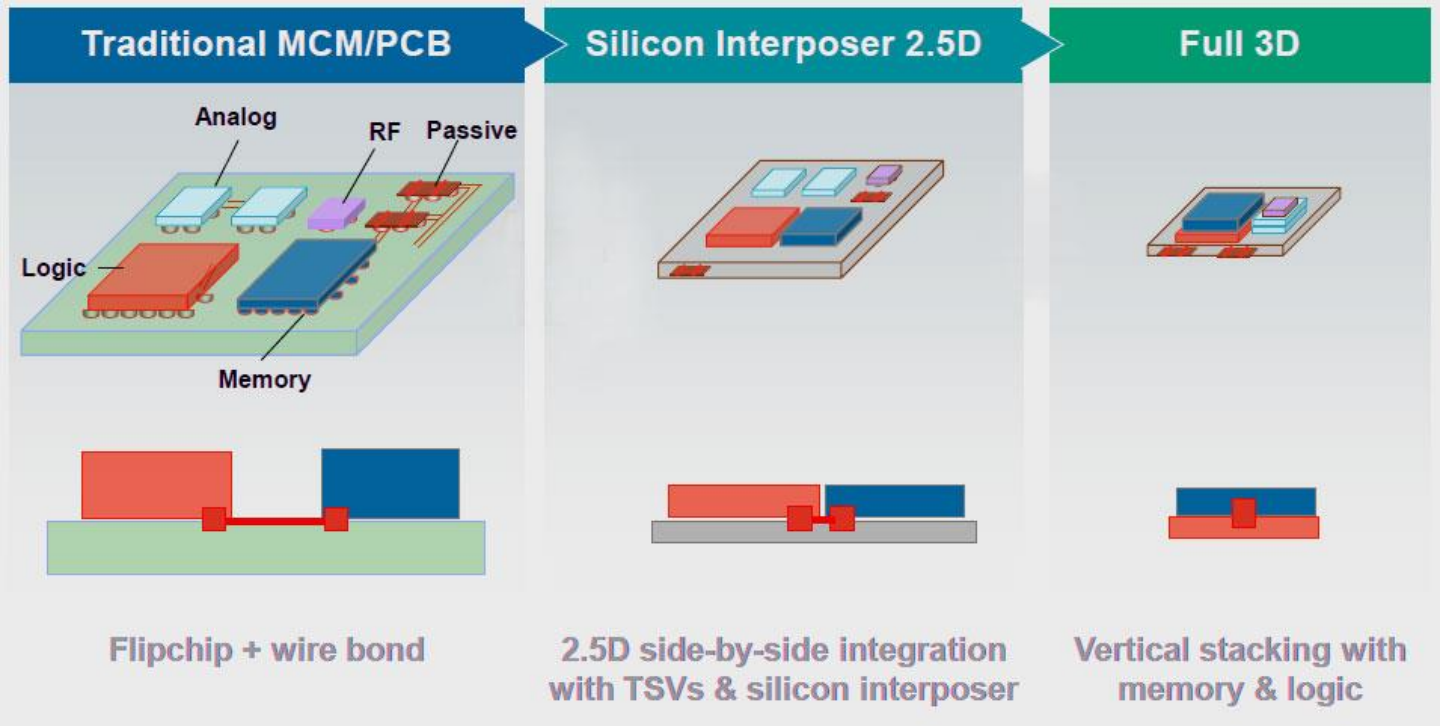
16 Parallel Tiles





[**Chen**, *MICRO* 2014]

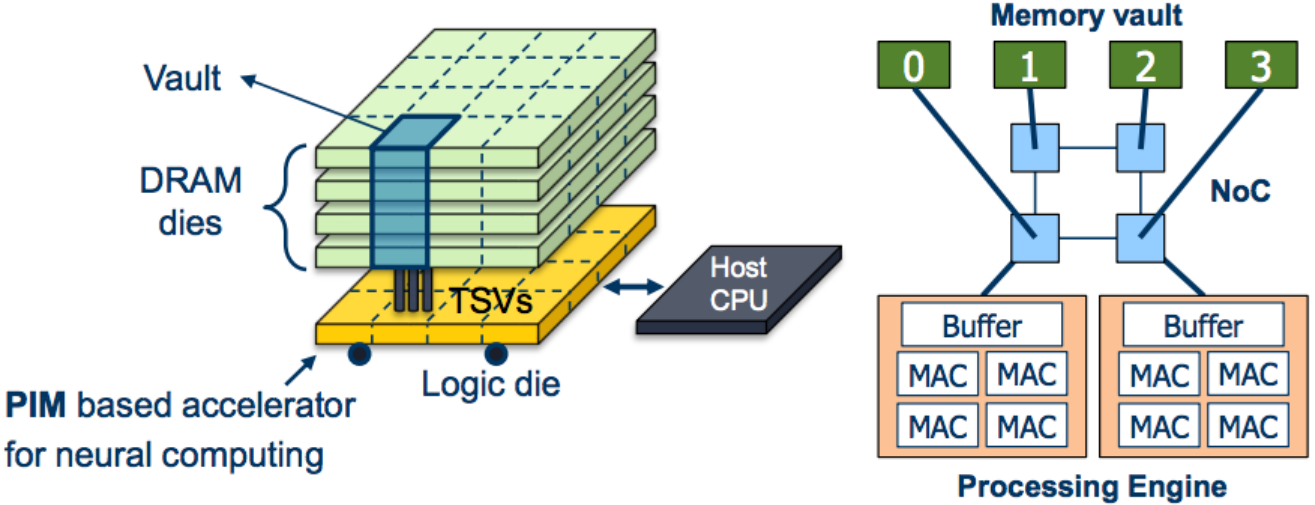
Going 3D!



Source: https://www.einfochips.com/blog/2-5d-3d-ics-new-paradigms-in-asic/

Stacked DRAM (NeuroCube)

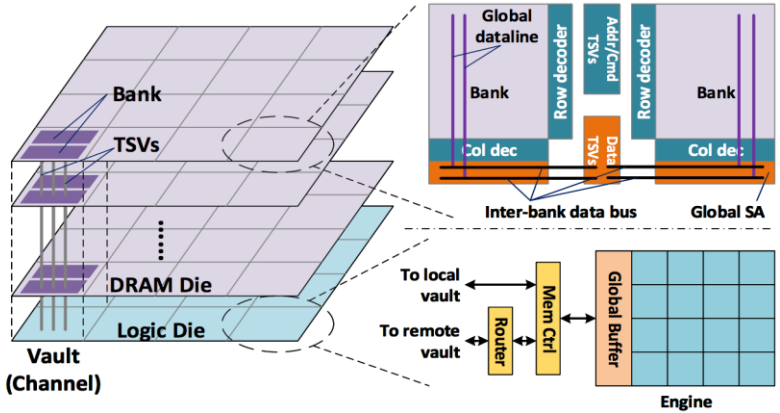
- NeuroCube on Hybrid Memory Cube Logic Die
  - 6.25x higher BW than DDR3
    - HMC (16 ch x 10GB/s) > DDR3 BW (2 ch x 12.8GB/s)
  - Computation closer to memory (reduce energy)



[**Kim**, *ISCA* 2016]

Stacked DRAM (Tetris)

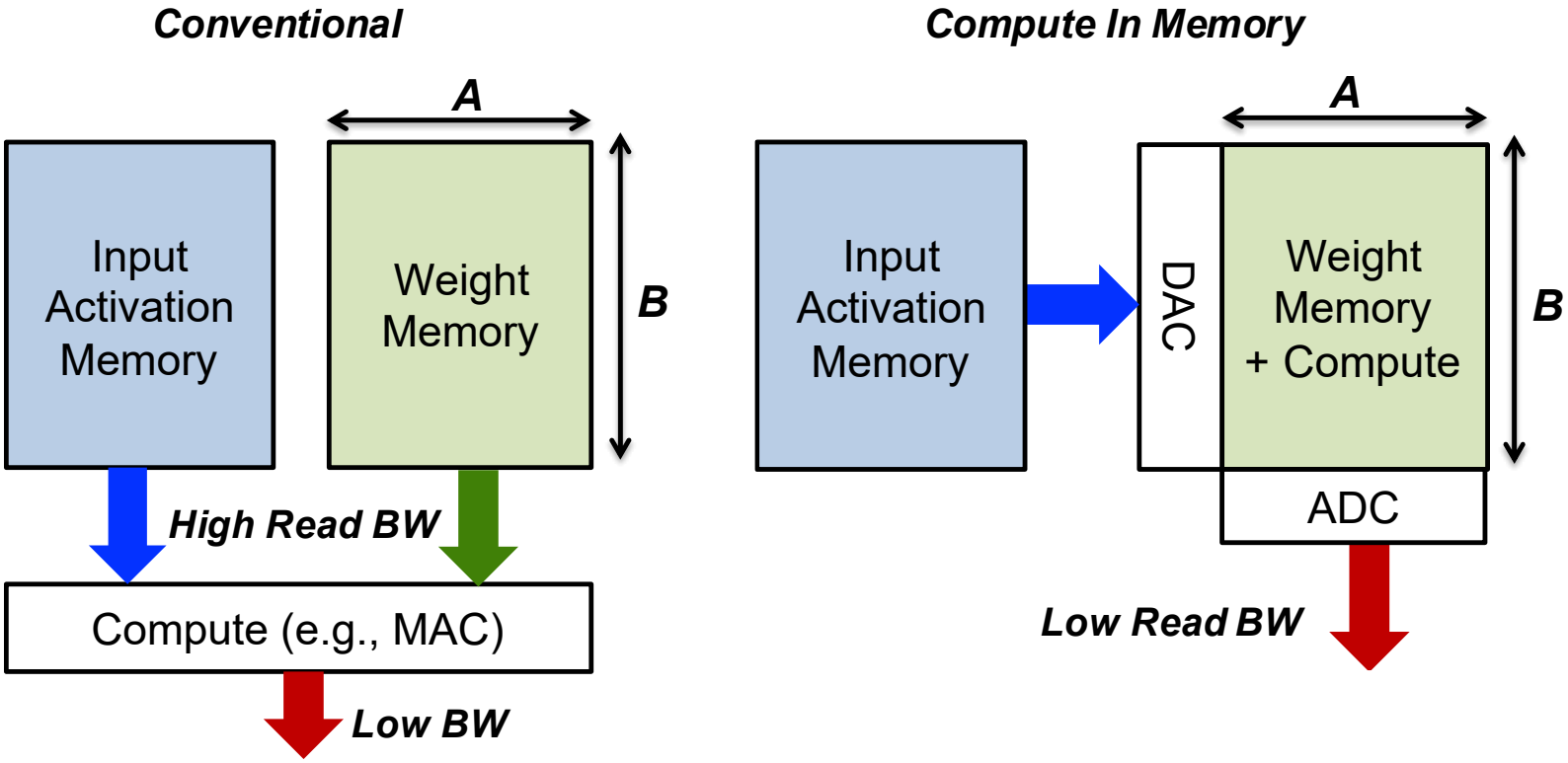
- Explores the use of HMC with the Eyeriss **spatial architecture** and **row stationary dataflow**
- Allocates **more area to the computation (PE array) than on-chip memory (global buffer)** to exploit the low energy and high throughput properties of the **HMC**
  - **1.5x energy reduction, 4.1x higher throughput** vs. 2-D DRAM



[**Gao**, *ASPLOS* 2017]

Eyeriss design

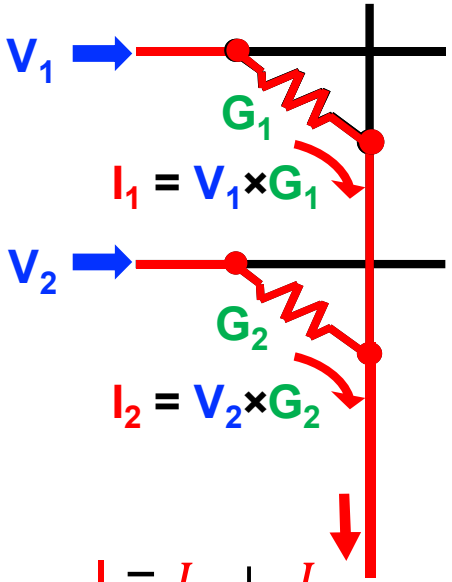
Conventional Processing vs. CiM



Analog Computing

Analog computing is typically required to bring the computation into the array of storage elements or into its peripheral circuits

Activation is input voltage (V_i)



Weight is resistor conductance (G_i) [= 1/resistance]

Many options for "resistance"

Partial sum is output current (I_i)

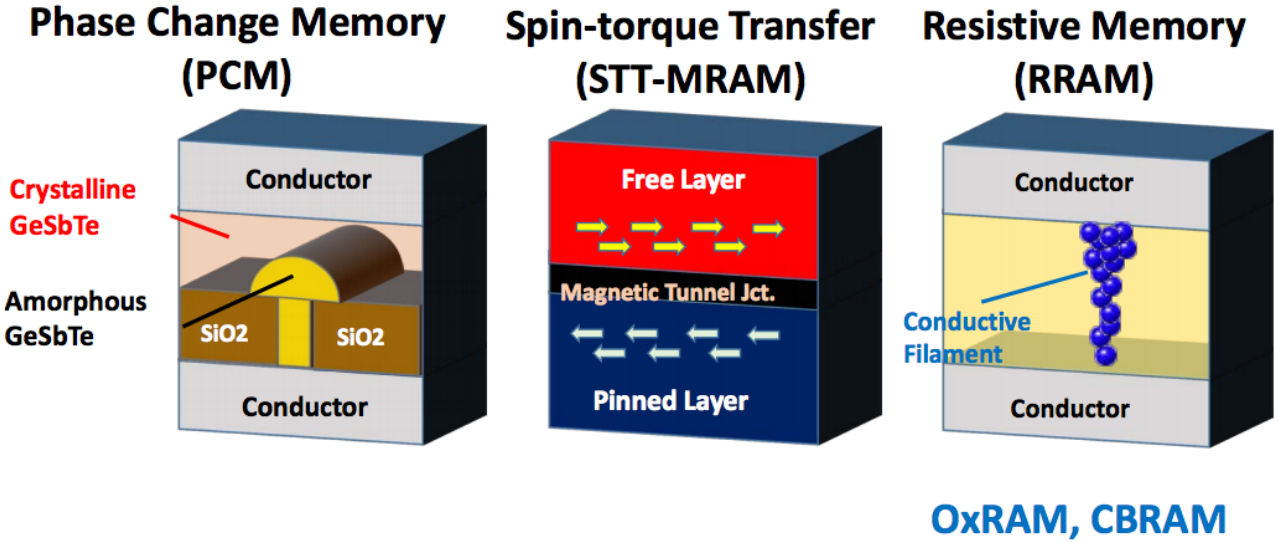
$$I = I_1 + I_2$$

$$I = V_1 \times G_1 + V_2 \times G_2$$

CiM Using Memristors in Non-Volatile Memory

Use memristors as programmable weights (resistance)

Candidate memristor devices



Source: Darsen Lu, MICRO-49 Tutorial on Emerging Memory

Candidate Devices for Memristor

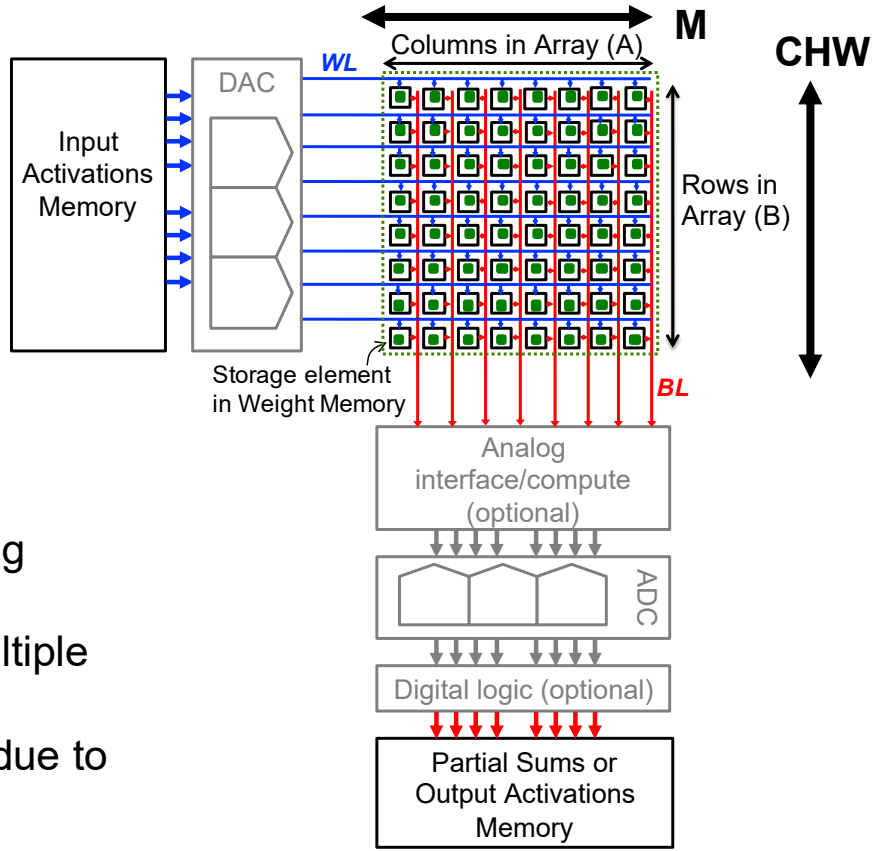
		PCM	STT-MRAM	RRAM	NAND Flash	DRAM
Power	Ewrite/ Bit Iwrite	18pJ [15][1] 100µA [15]	1.0pJ [20] 50µA [33]	1.0pJ [20] 1.0µA [18]	100pJ [31]	<1.0 pJ [9]
Perform-ance[2]	Write Lat.	150ns [15]	5ns [6]	50ns [20]	>100µs	5ns [27]
	Read Lat.	80ns [28]	10ns [5]	<10ns [30]	15-50µs [35]	20–80ns [35]
Reliabi-lity	Program Window	3 bit/cell [12]	Good [3]	Variable [23]	4 bit/cell [32]	Good
	Endurance	10^8-10^9 [4,9]	Unlimited [1]	10^5-10^{10} [1]	10^5-10^6 [8]	Unlimited
	Retention	R-drift[12]	Good [6]	RTN [24]	Good	64ms
Density	Cell size	4 F^2 [15]	12 F^2 [33]	4-6F^2 [21]	<4 F^2 [15]	7 F^2 [15]

Emerging Memory Technologies
(Speed of DRAM; Non-volatility of NAND)

1. Estimated using I_reset * Vdd * t_write
2. Required programming pulse duration

Weight Stationary Dataflow for CiM

Transform into **matrix-vector multiply** with weights in matrix, and input activations in vector



Benefits

- Reduces data movement of weights
- Higher memory bandwidth by reading multiple weights in parallel
- Higher throughput by performing multiple computations in parallel
- Lower input activation delivery cost due to increased density of compute

Loop Nest for CiM

For Fully Connected layer

```
i = Array(CHW)          # Input activations
f = Array(M, CHW)       # Filter weights
o = Array(M)            # Output partial sums

parallel–for m in [0, M):
    parallel–for chw in [0, CHW):
        o[m] += i[chw] * f[m, chw]
```

For Convolutional layer (1-D toy example)

```
i = Array(C, W)         # Input activations
f = Array(M, C, S)      # Filter weights
o = Array(M, Q)         # Output partial sums

parallel–for m in [0, M):
    parallel–for s in [0, S):
        parallel–for c in [0, C]:
            for q in [0, Q):
                w = q + s
                o[m, q] += i[c, w] * f[m, c, s]
```

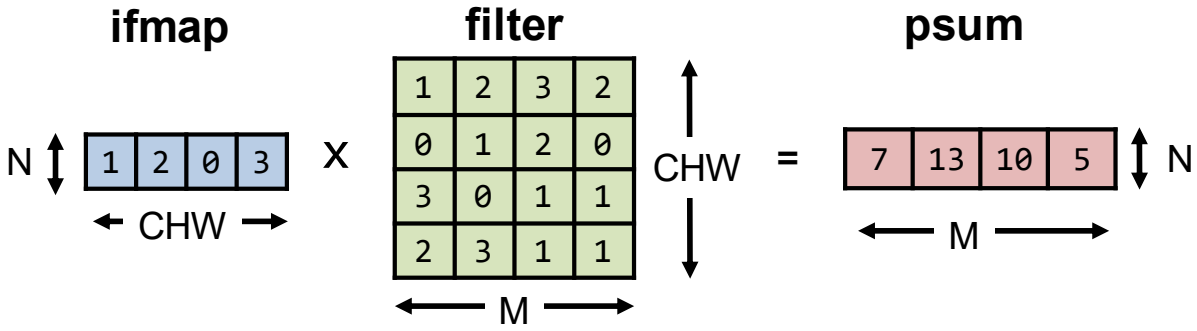
Design Considerations for CiM

- Analog computing has **increased sensitivity** to circuit and device non-idealities (i.e., non-linearity and process, voltage and temperature variations)

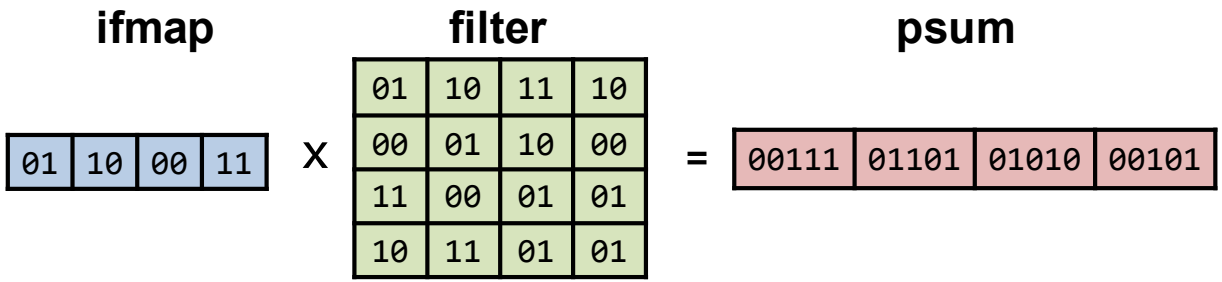
- Requires **trade offs between energy efficiency, throughput, area density, and accuracy**, which reduce the achievable gains over conventional architectures

Toy Example for CiM

Decimal Representation

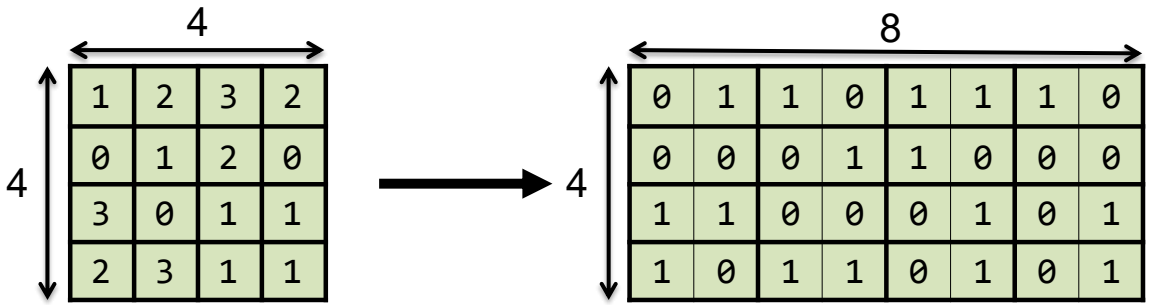


Binary Representation



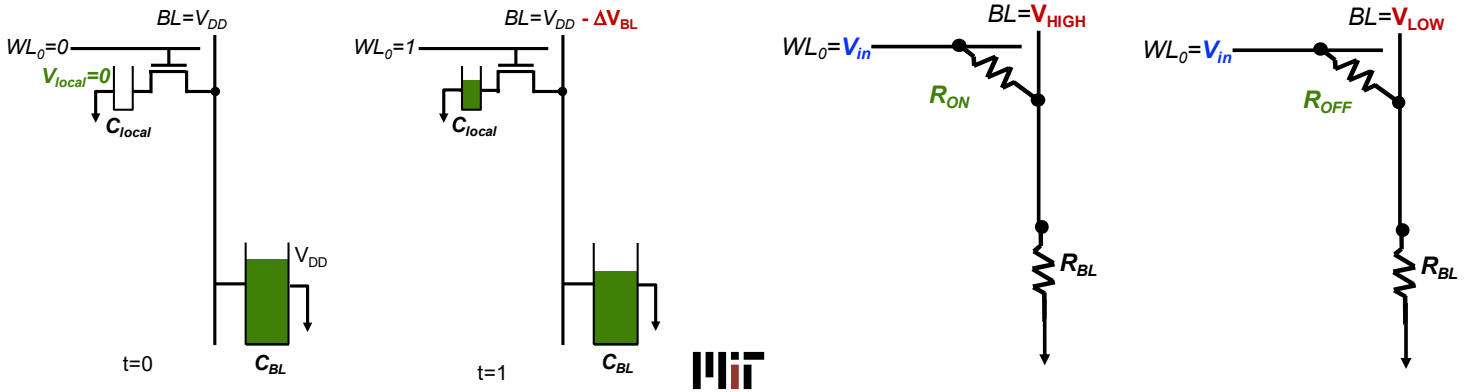
Number of Storage Elements per Weight

- Ideally, it would be desirable to be able to use one storage element (i.e., one device or bit cell) per weight to maximize density.
- In practice, **multiple storage elements are required per weight** due to the limited precision of each device or bit cell (typically on the order of 1 to 4 bits)
  - This is also referred to as **bit slicing** of the weight **(weight slicing)**



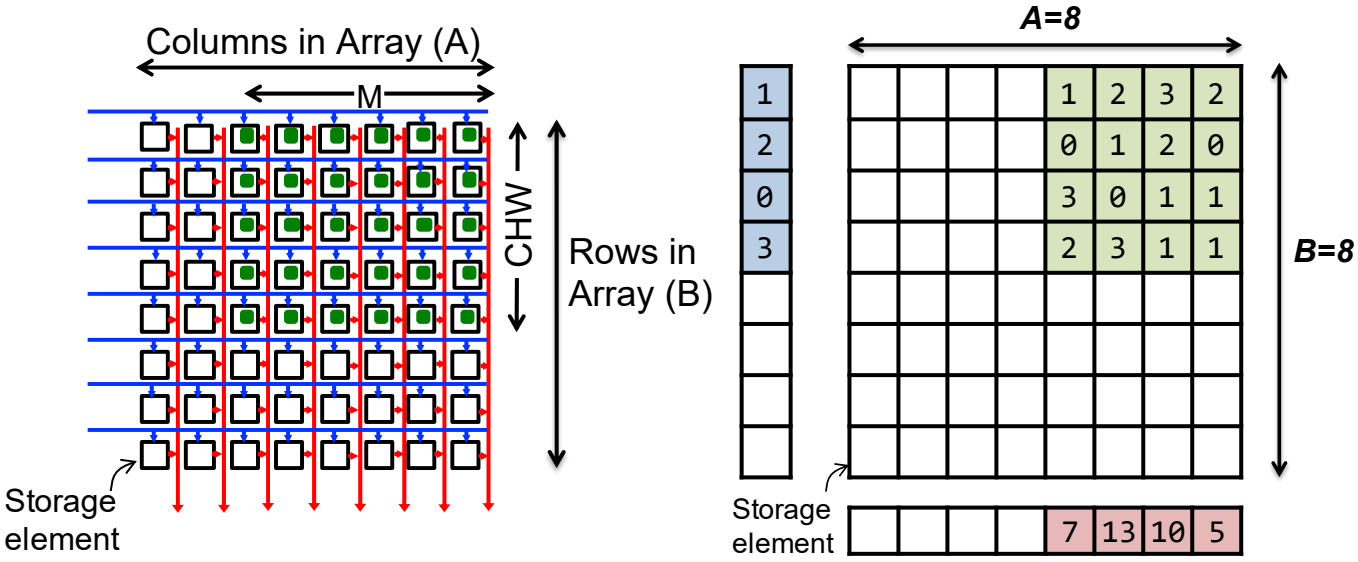
Array Size

- Ideally, it would be **desirable to have a large array size**
  - Increases weight read bandwidth and throughput
  - Increases area density → amortize the cost of the peripheral circuits (e.g., ADC, DAC) which can account for over 50% of Non-Volatile Memory (NVM) designs
- Array size limited by the resistance and capacitance of word line and bit line wires, which impacts robustness, speed and energy consumption
  - If capacitance or resistance of bit line much larger than storage element, it is difficult to sense value in storage element (i.e., change in bit-line voltage/current due to value in storage element is reduced)



Array Size

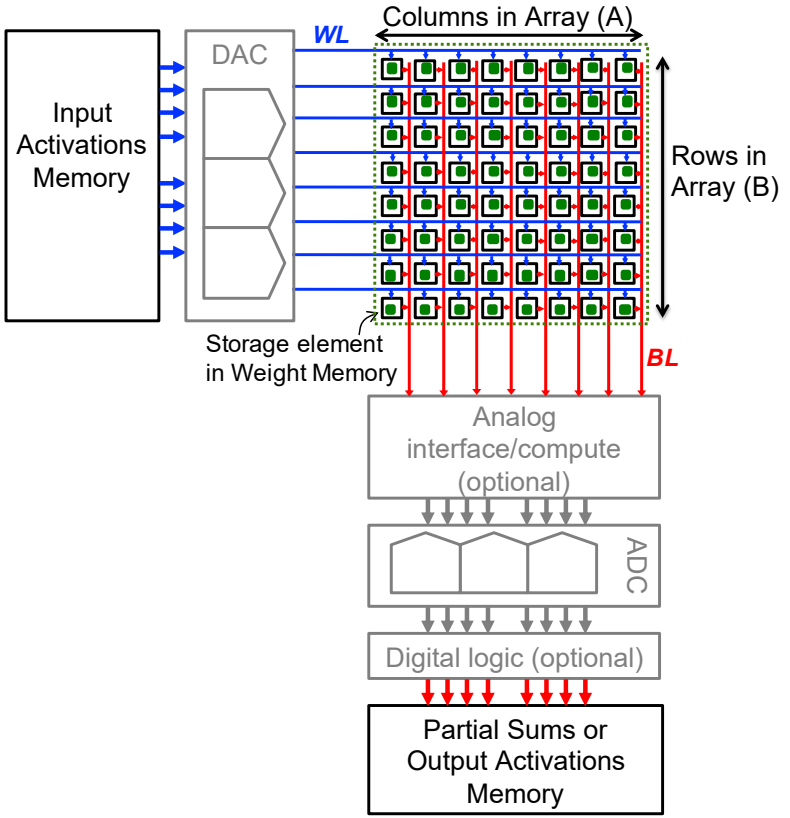
Utilization of the array will drop if the workload cannot fill entire column or entire row



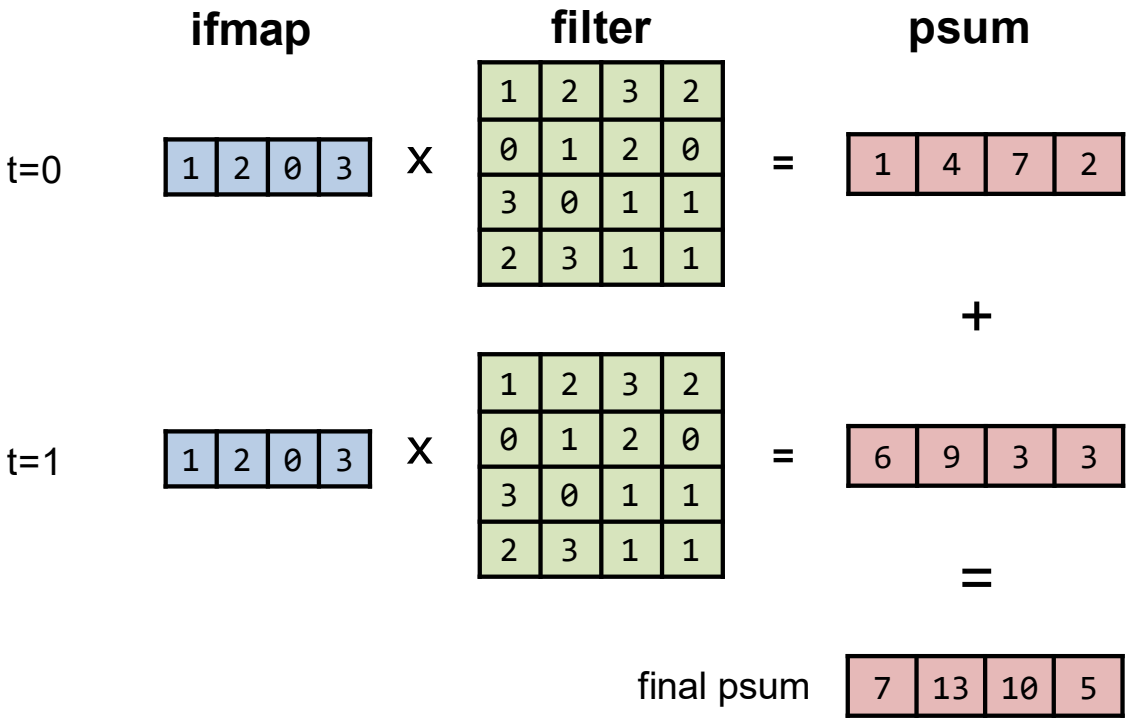
Number of Rows Activated in Parallel

- Ideally, it would be desirable to use all rows at once to maximize parallelism for high bandwidth and high throughput

- In practice, the number of rows that can be used at once is limited due to
  - ADC resolution (number of bits it can resolve)
  - Cumulative effect of the device variations
  - Maximum voltage drop or accumulated current on bit line



Number of Rows Activated in Parallel

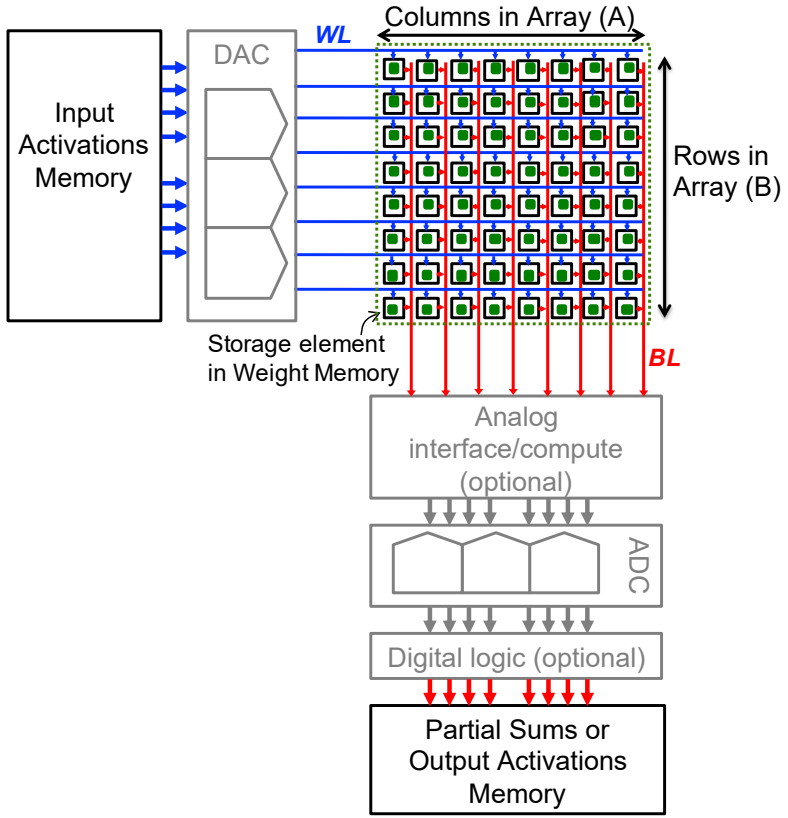


Multiple cycles to complete accumulation across rows → **Reduces Throughput**

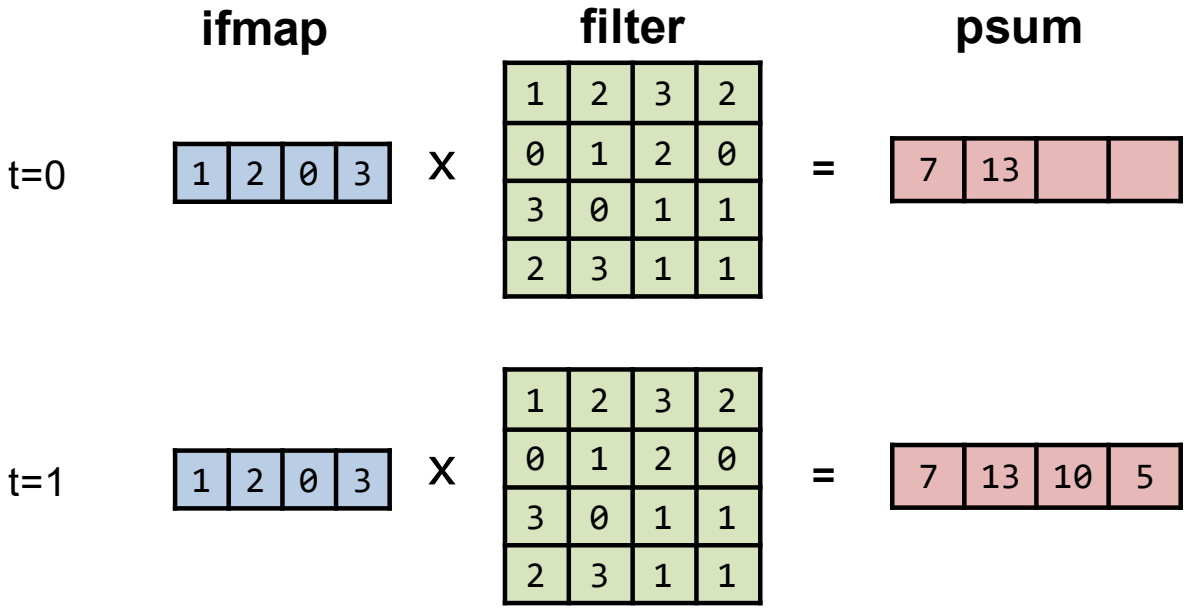
Number of Columns Activated in Parallel

- Ideally, it would be desirable to use all columns at once to maximize parallelism for high bandwidth and high throughput

- In practice, the number of columns that can be used is limited by ADC area



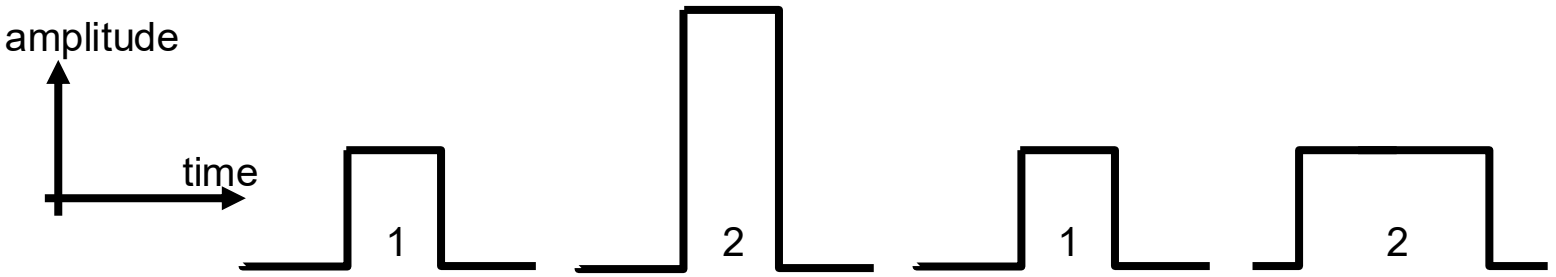
Number of Columns Activated in Parallel



Multiple cycles to complete accumulation for all columns → **Reduces Throughput**

Time to Deliver Input

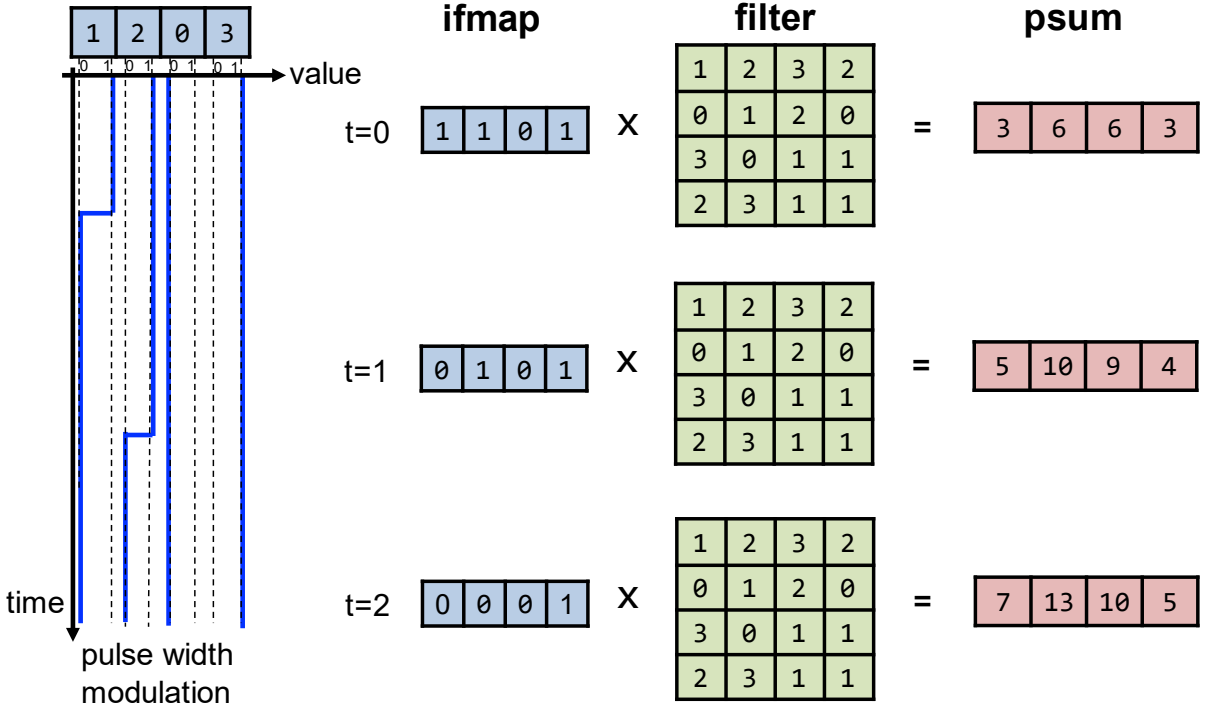
- Ideally, encode input activations onto the word line in the minimum amount of time for maximum throughput (e.g., voltage modulation)
- In practice, challenging due to non-linearity of devices and complexity of DAC → need to encode in time → **Reduces Throughput**



Voltage (amplitude) modulation

Pulse width (time) modulation

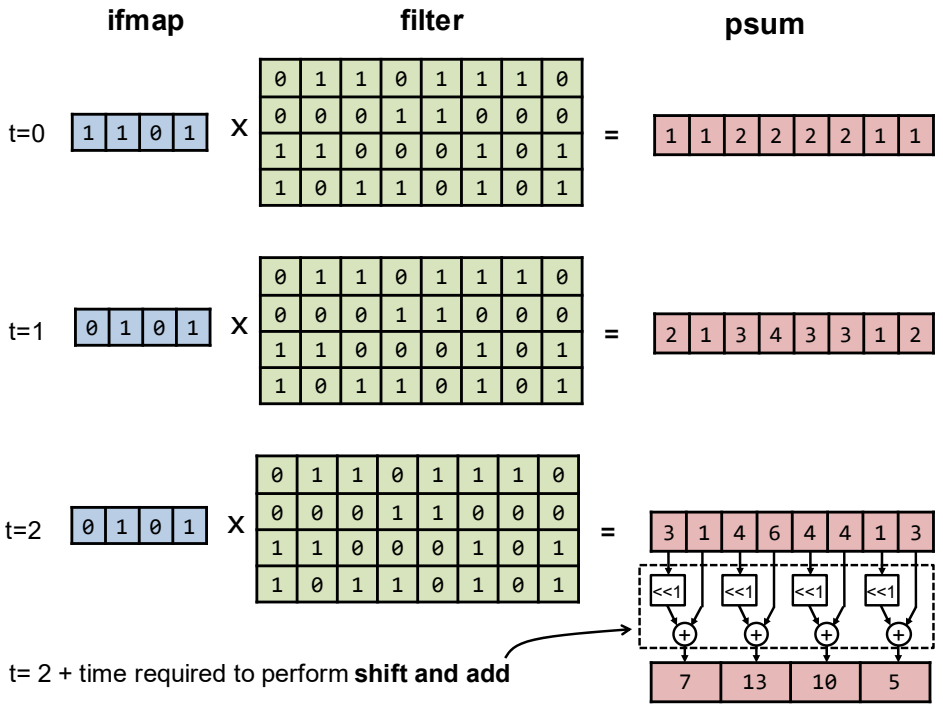
Time to Deliver Input



Time to Compute a MAC

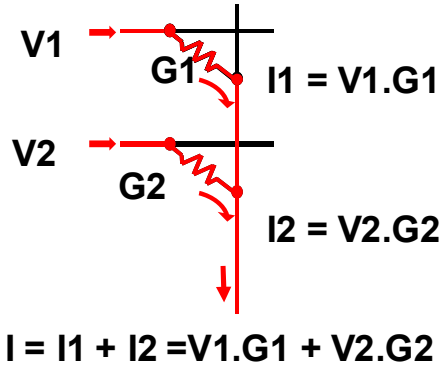
- Ideally, compute MAC in a single cycle for high throughput

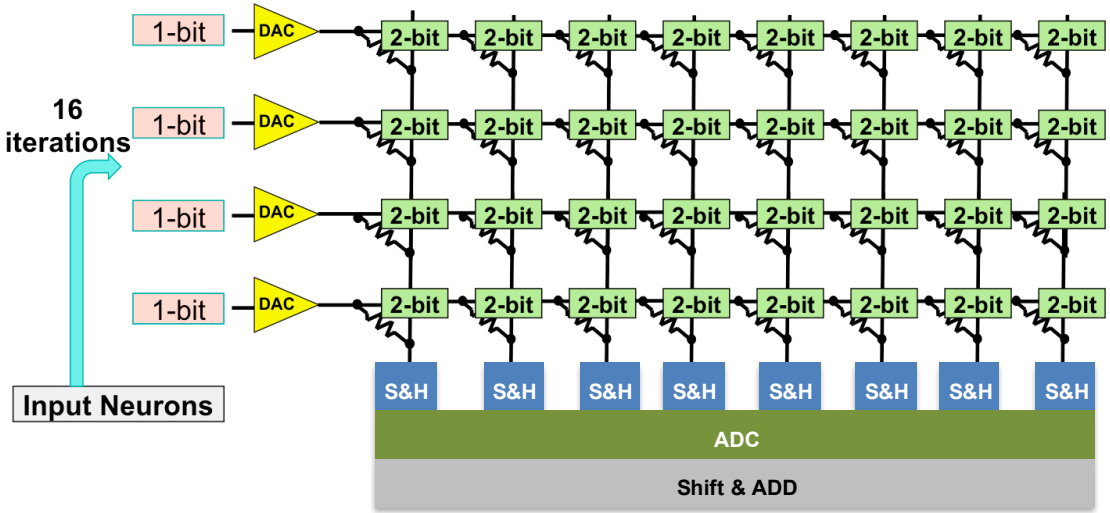
- In practice, the storage element typically can only perform one-bit or two-bit operations → need multiple cycles to build up to a multi-bit operation (temporal accumulation) → **Reduces throughput**



ISAAC

- Replace eDRAM in DaDianNao with memristors
- 16-bit dot-product operation
  - 8 x 2-bits per memristors (weight slicing)
  - 1-bit per cycle computation (input slicing)

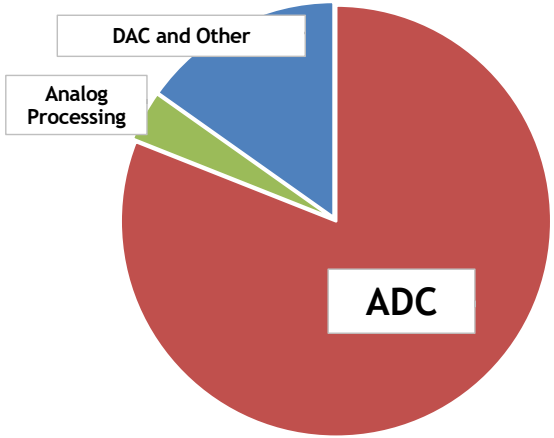




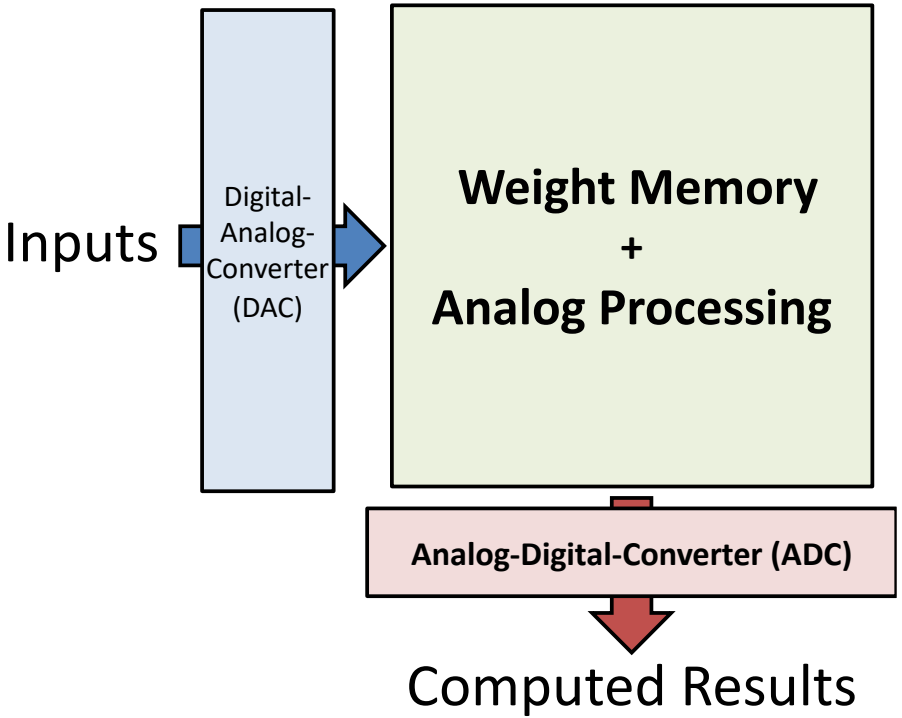
[**Shafiee**, *ISCA* 2016]

Compute In Memory (CiM) Accelerators

Energy Breakdown



ADC consumes significant energy



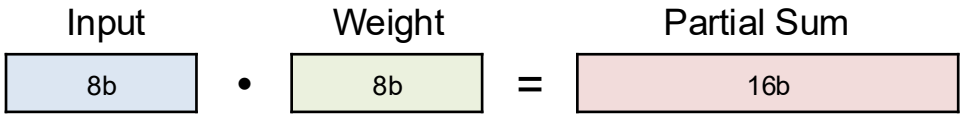
The Titanium Law

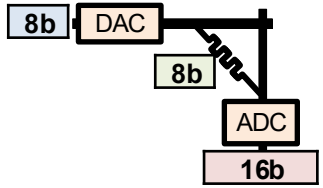
ADC energy is a product of **four** terms

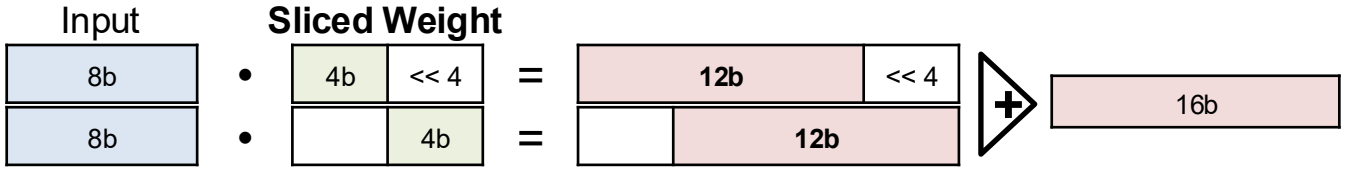
$$\frac{\textbf{ADC Energy}}{\textbf{DNN}} = \overbrace{\frac{\textbf{Energy}}{\textbf{Convert}}}^{\uparrow \text{ exponentially with higher ADC resolution}} \times \underbrace{\frac{\textbf{Converts}}{\textbf{MAC}}}_{\downarrow \text{ with more rows; } \uparrow \text{ with more input/weight slices}}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

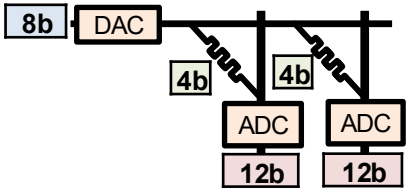
[**Andrulis**, *ISCA* 2023]

Use Bit Slicing to Reduce ADC Resolution

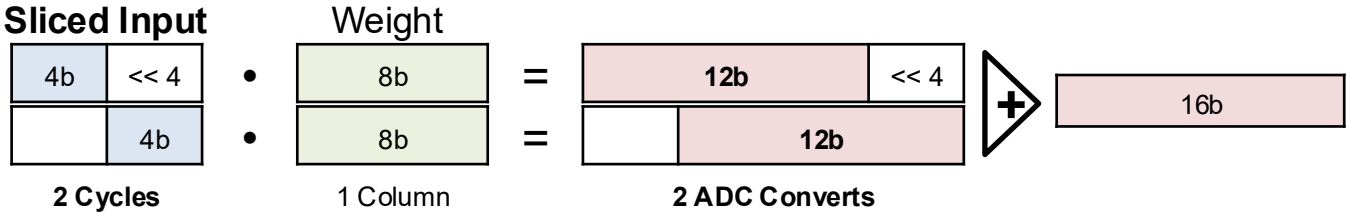


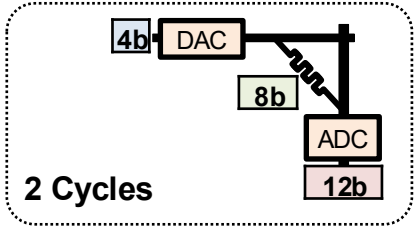






Weight slicing increases **area** and number of **ADC converts**

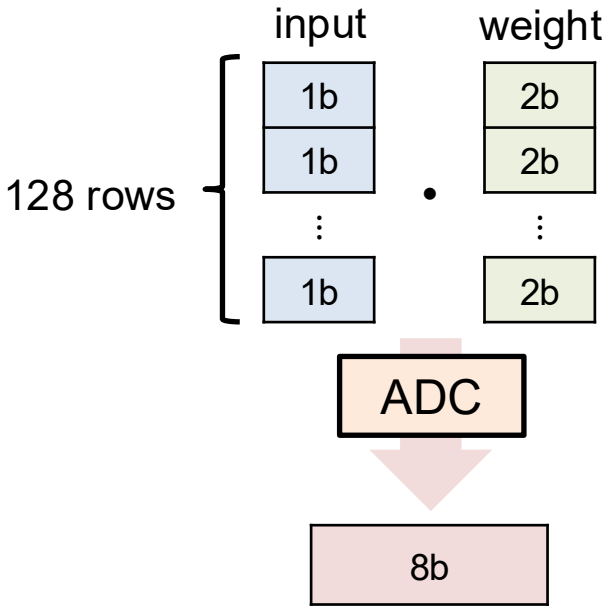




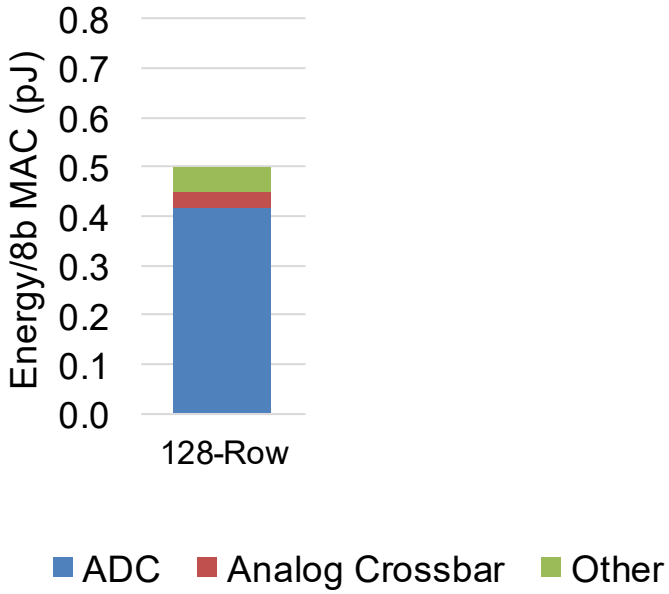
Input slicing increases **time** and number of **ADC converts**

The Titanium Law: Revisit ISAAC

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

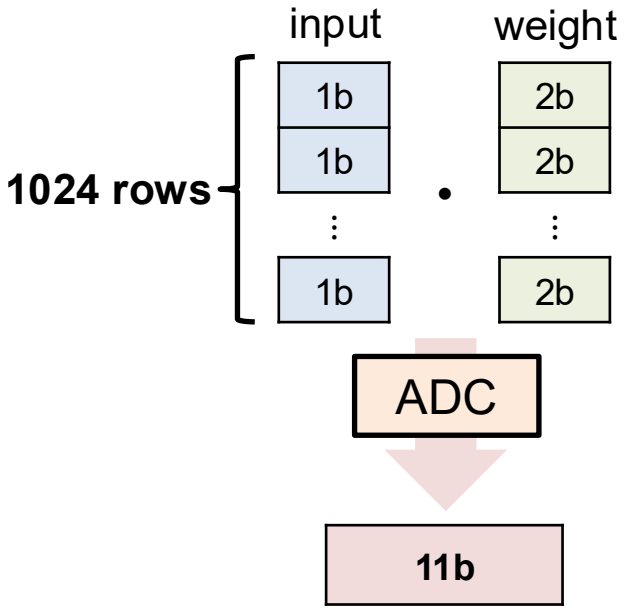


Can we **reduce** ADC energy?

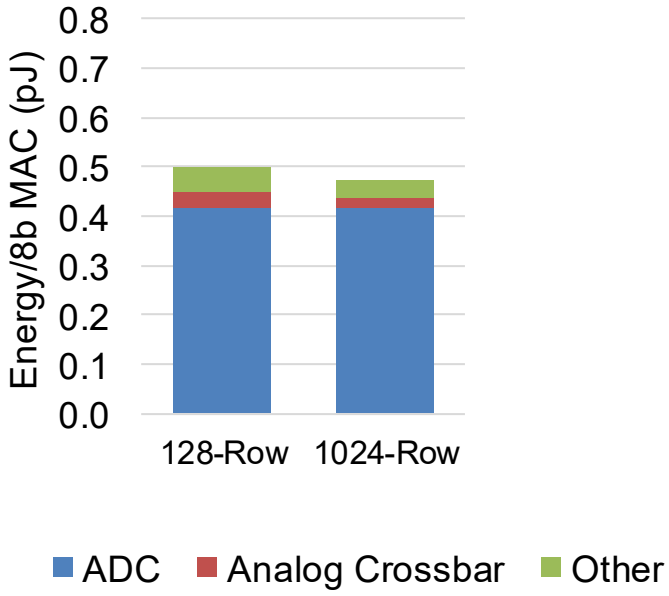


The Titanium Law: Revisit ISAAC

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

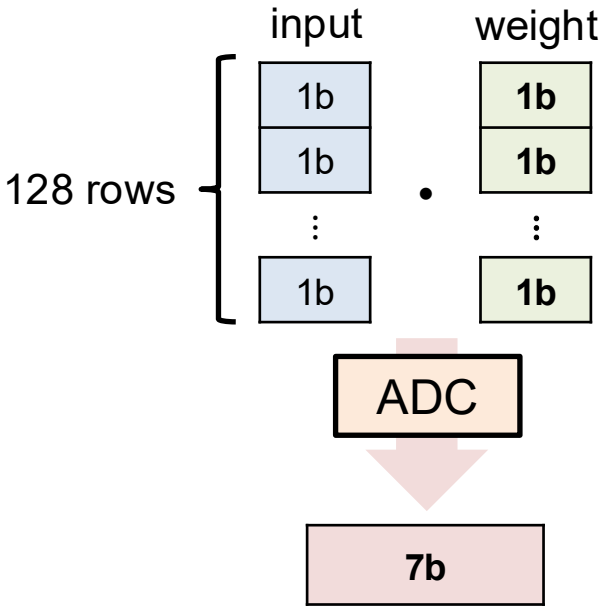


Increase rows
↓
Increase bits

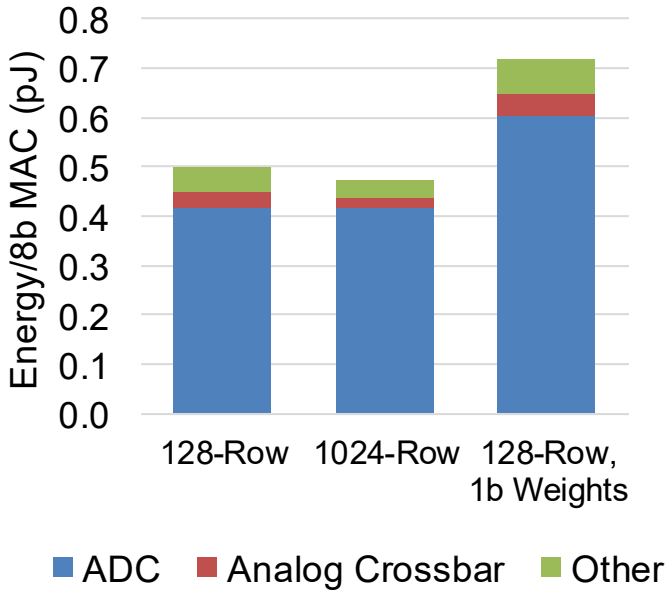


The Titanium Law: Revisit ISAAC

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



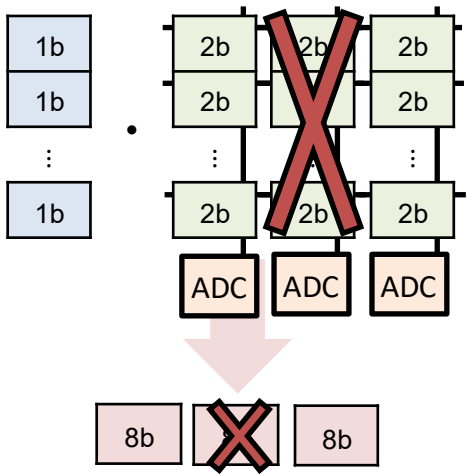
Decrease bits/weight slice
↓
Increase weight slices
↓
Increase ADC converts



How Have Prior Works Escaped These Tradeoffs?

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

Weight-Count-Limited



Prune weights

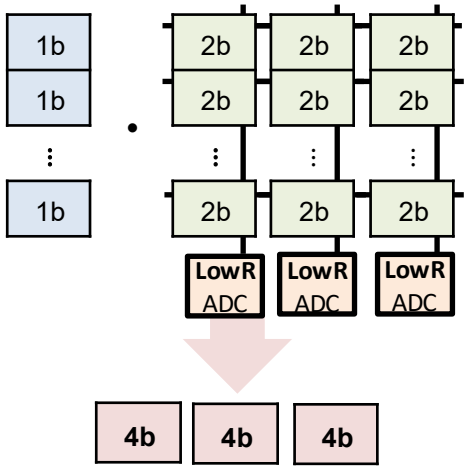
↓

Reduce MACs/DNN

How Have Prior Works Escaped These Tradeoffs?

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

Sum-Fidelity-Limited

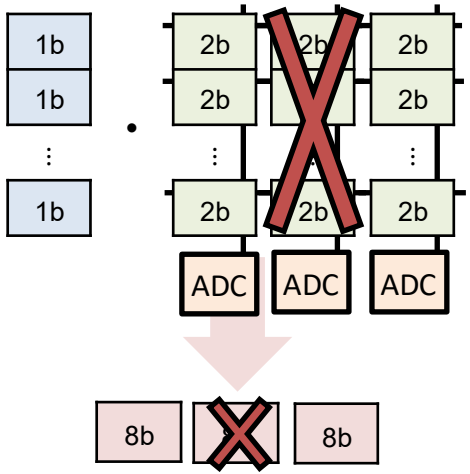


Low-Res ADC
↓
Reduce energy per convert

How Have Prior Works Escaped These Tradeoffs?

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

Weight-Count-Limited

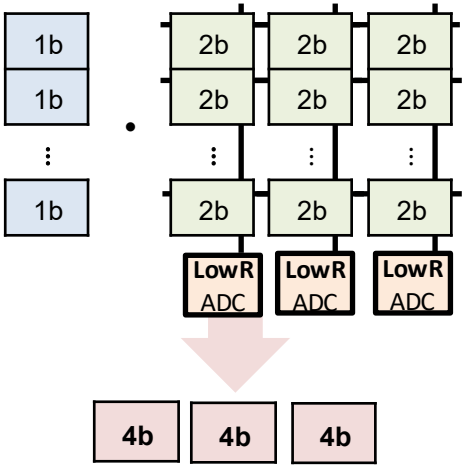


Prune weights

↓

Reduce MACs/DNN

Sum-Fidelity-Limited



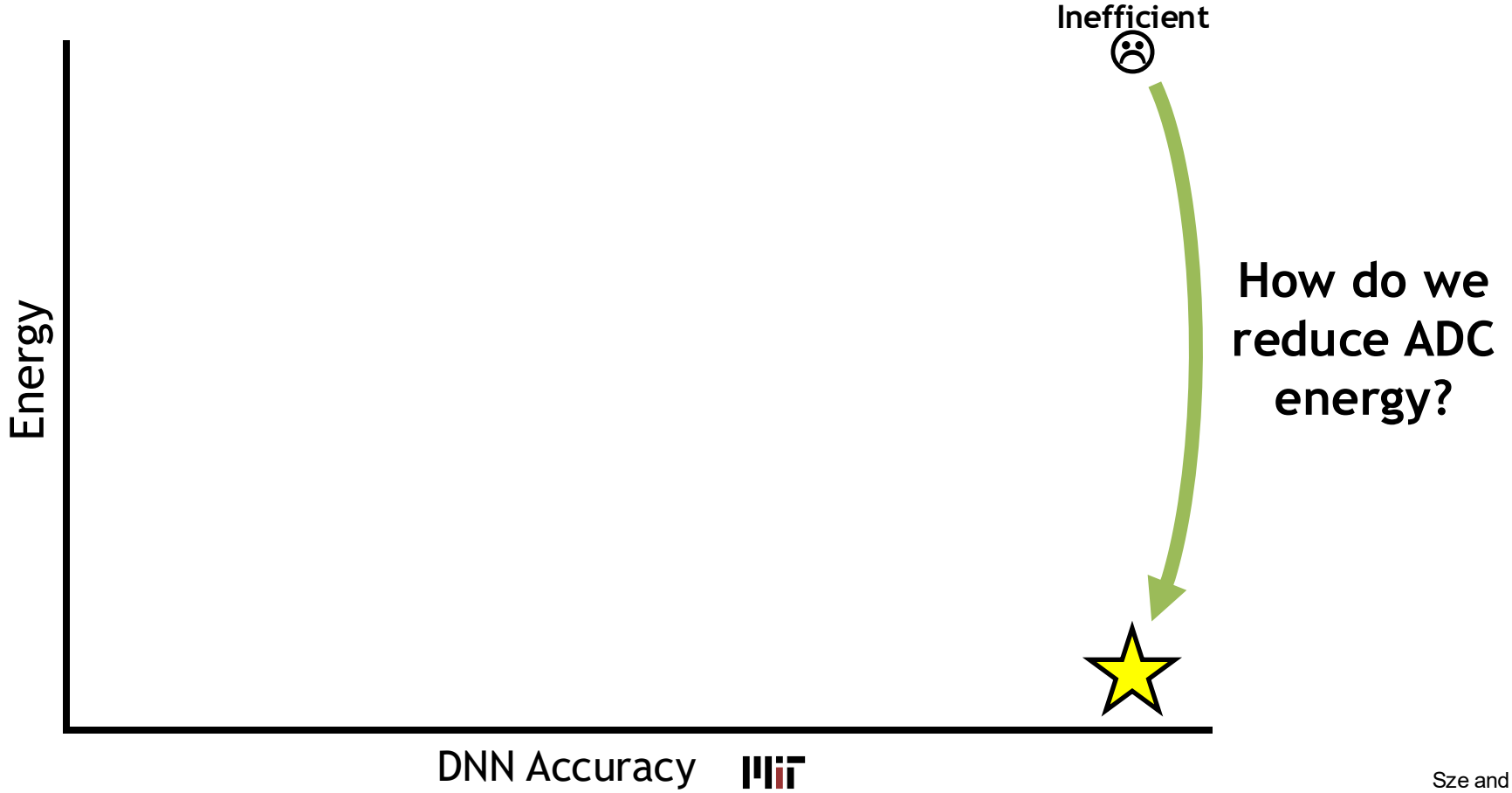
Low-Res ADC

↓

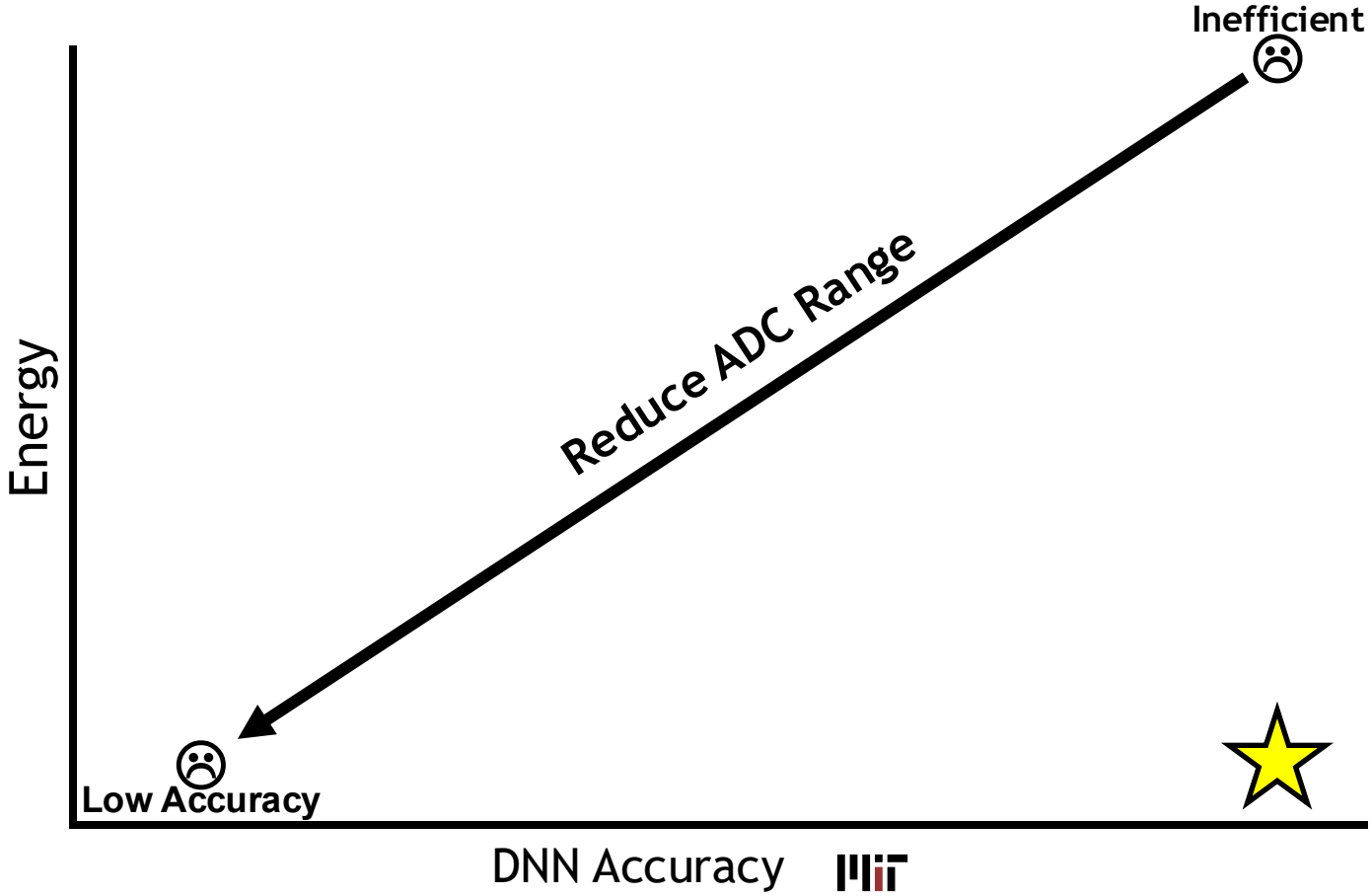
Reduce energy per convert

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

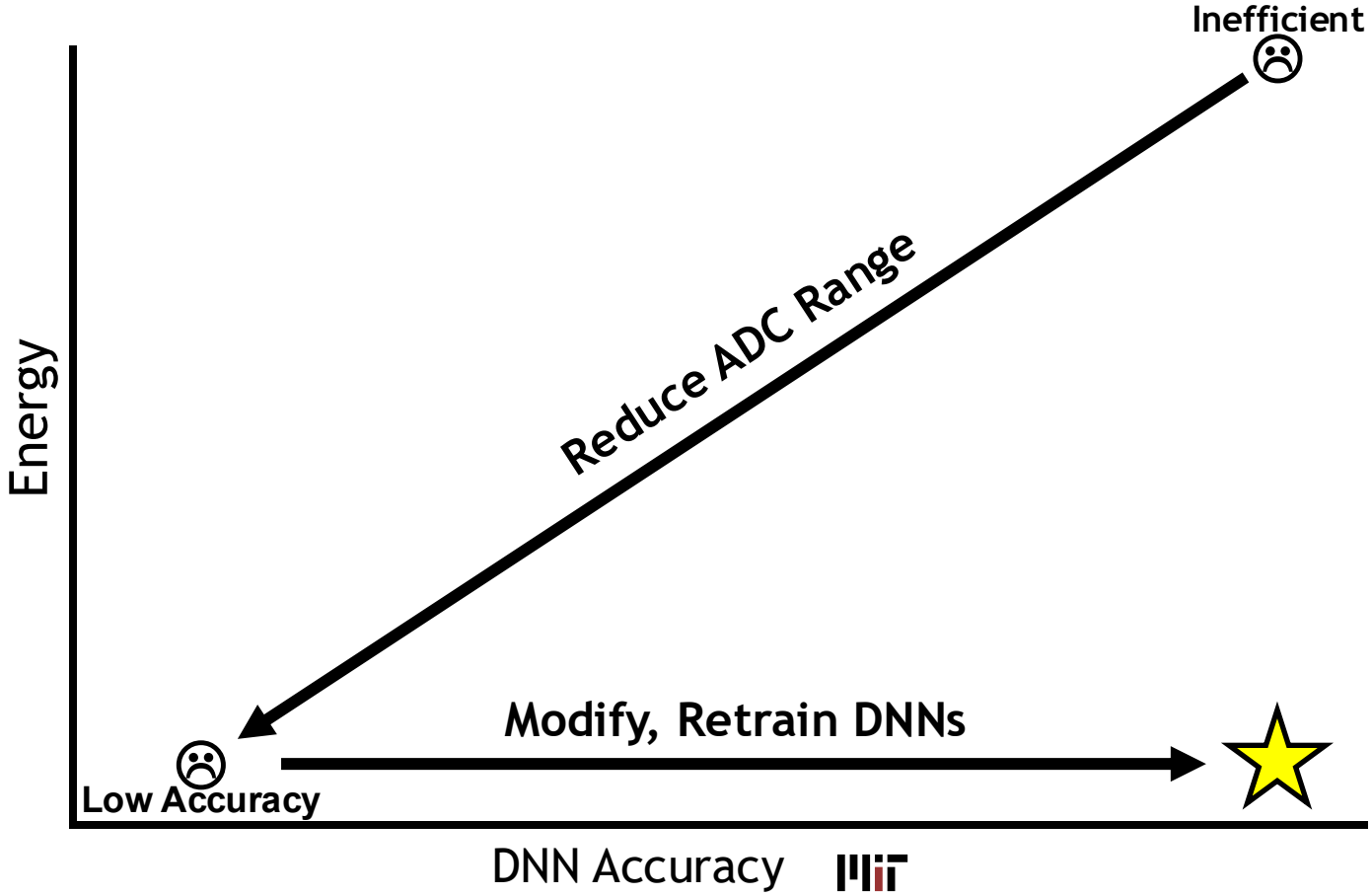
Reducing ADC Energy



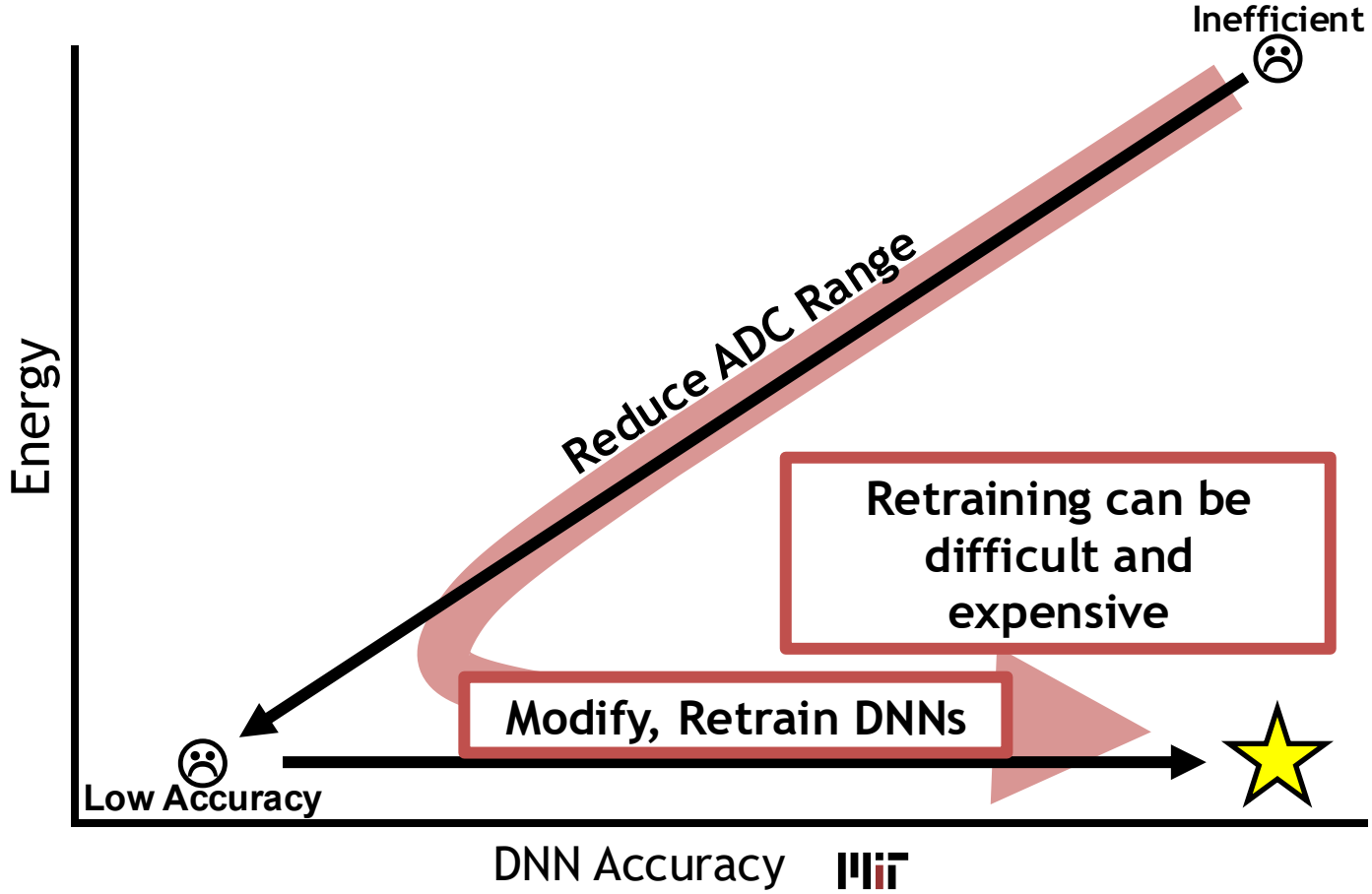
Reducing ADC Energy



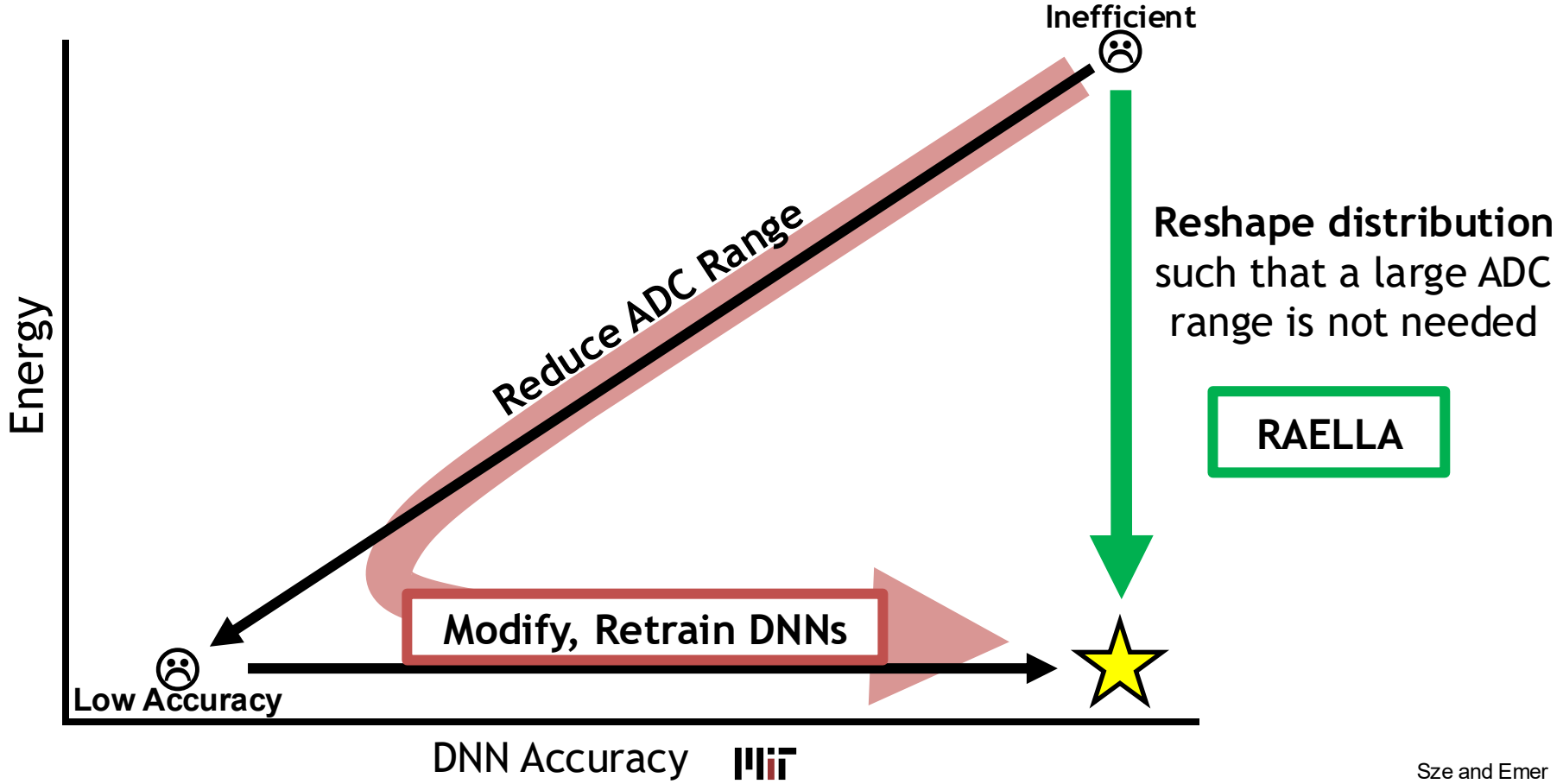
Reducing ADC Energy



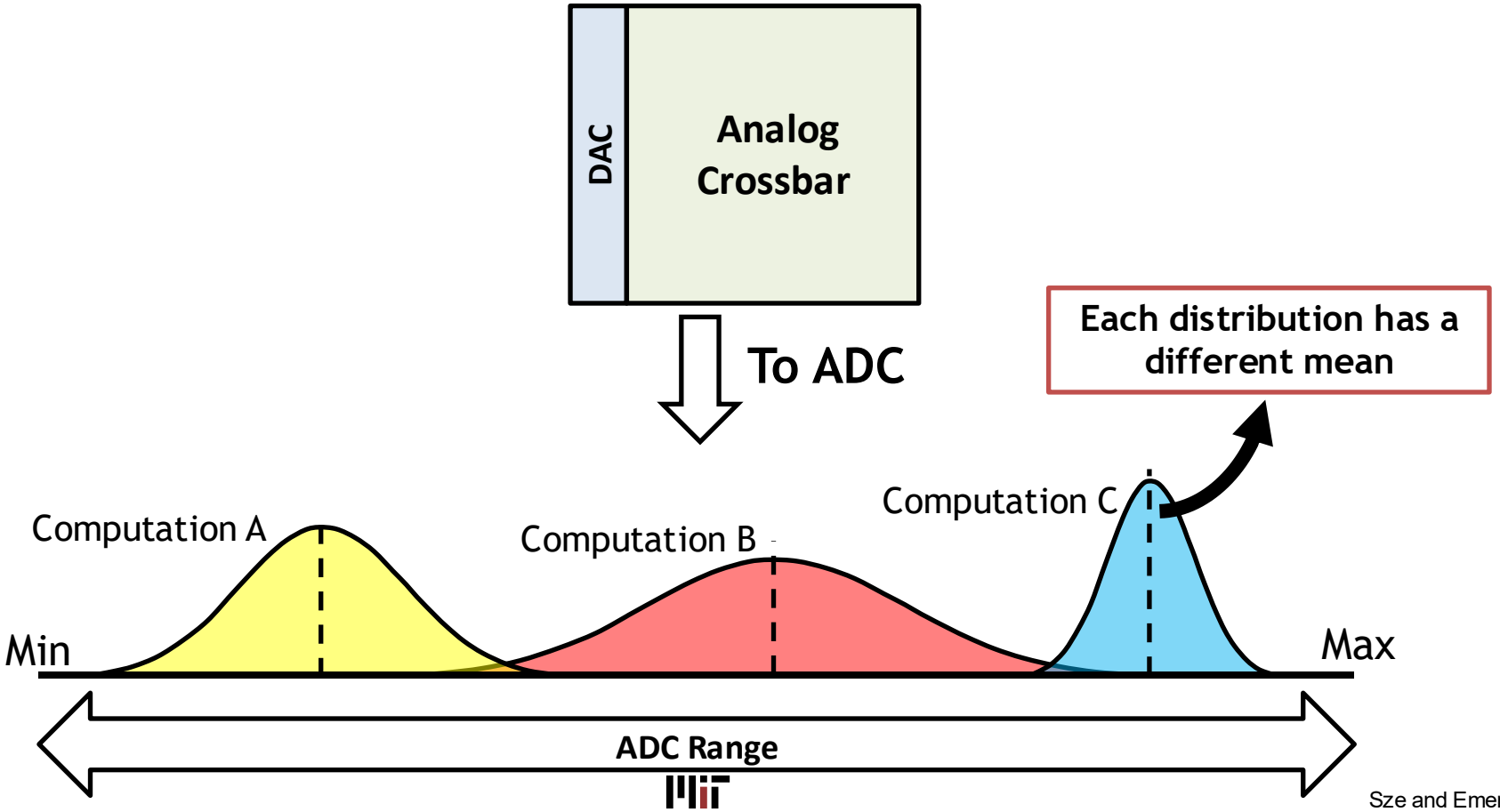
Reducing ADC Energy



Reducing ADC Energy

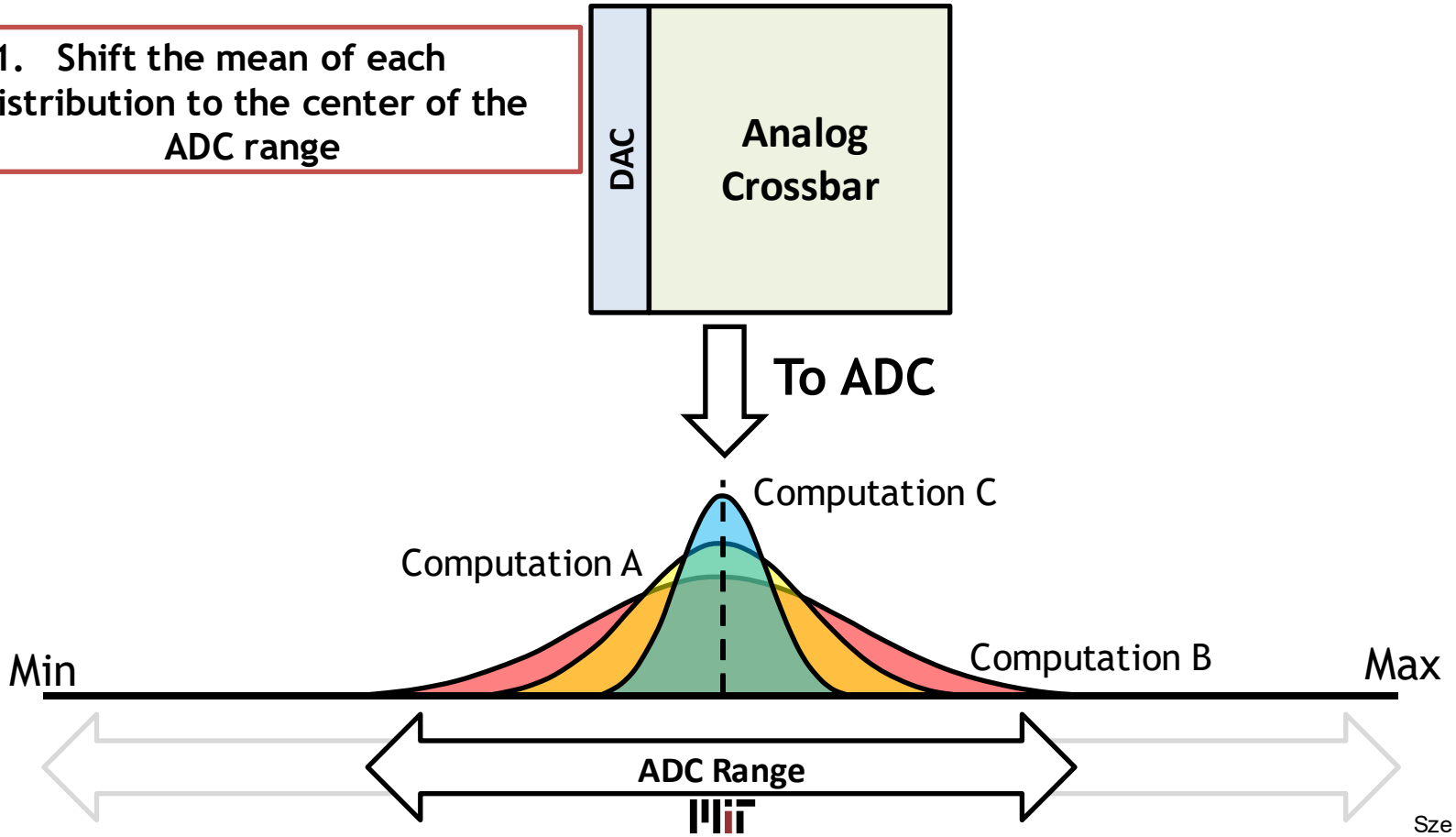


RAELLA: Reshape Distributions to Reduce Input to ADC



RAELLA: Reshape Distributions

1. Shift the mean of each distribution to the center of the ADC range

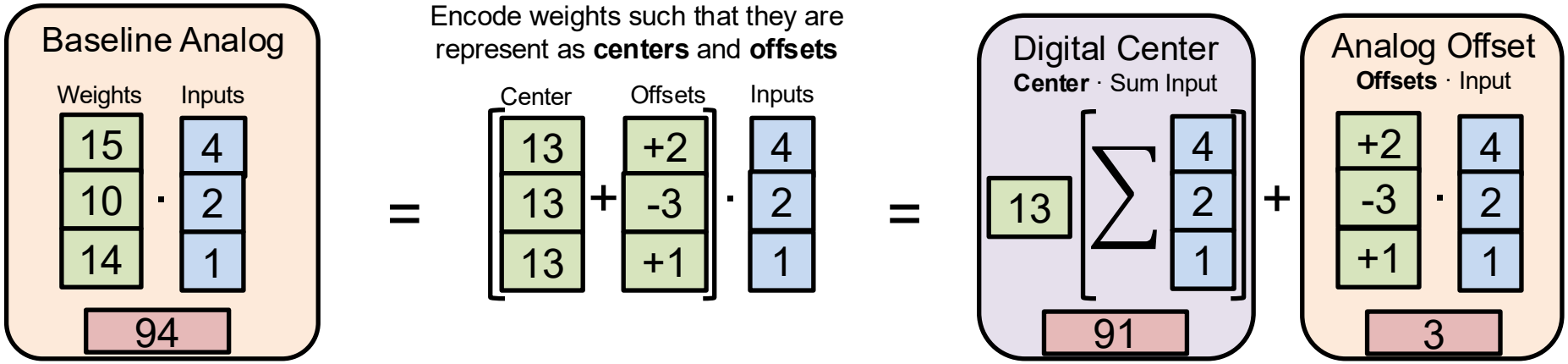


Center + Offset Weight Encoding Zero-Average Analog Results

Partition computation

Digital calculates high-resolution **center** operations

Analog calculates parallel **offset** operations

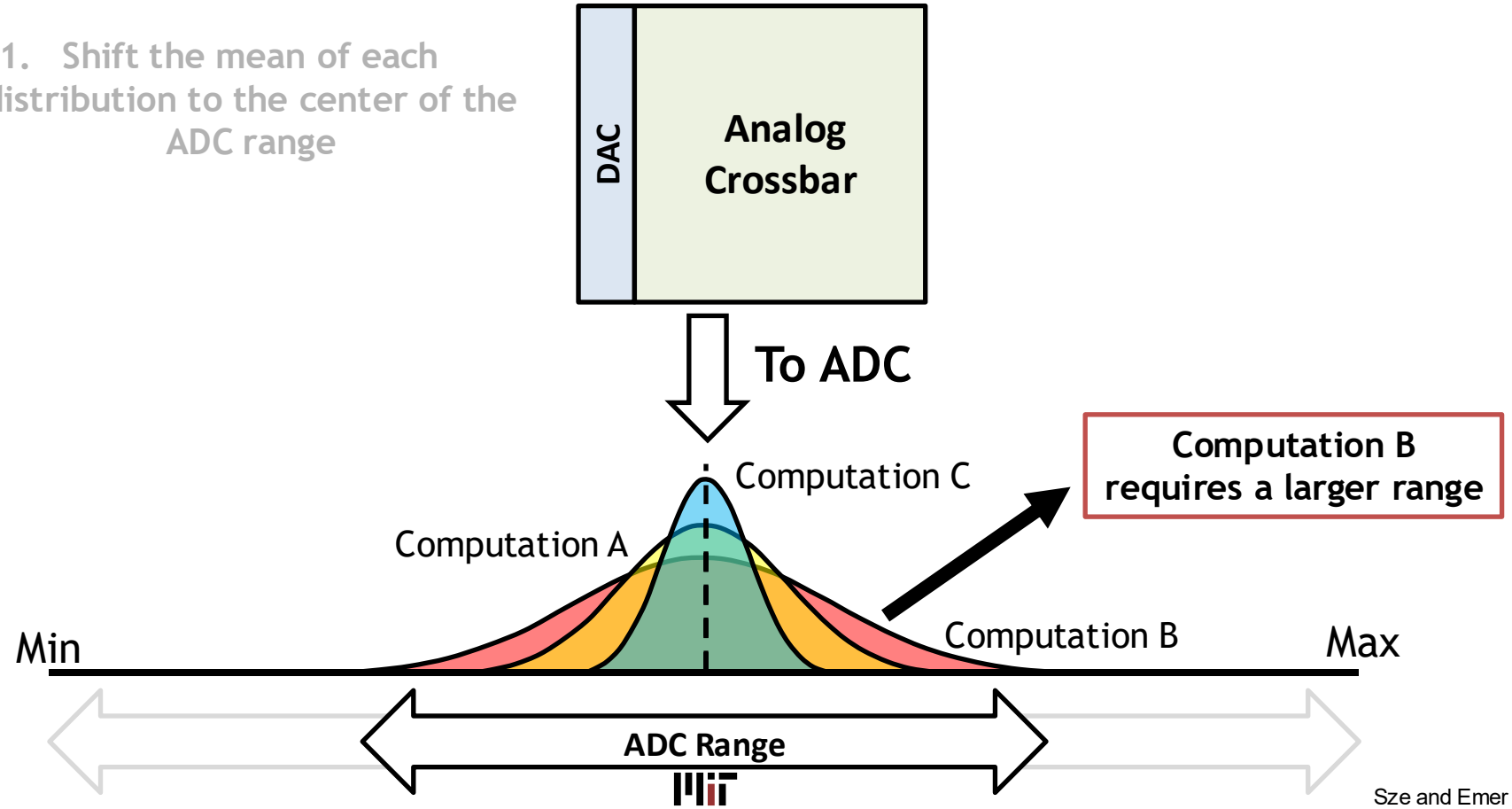


Encoding allows analog input to ADC to be **smaller and closer to zero**

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

RAELLA: Reshape Distributions

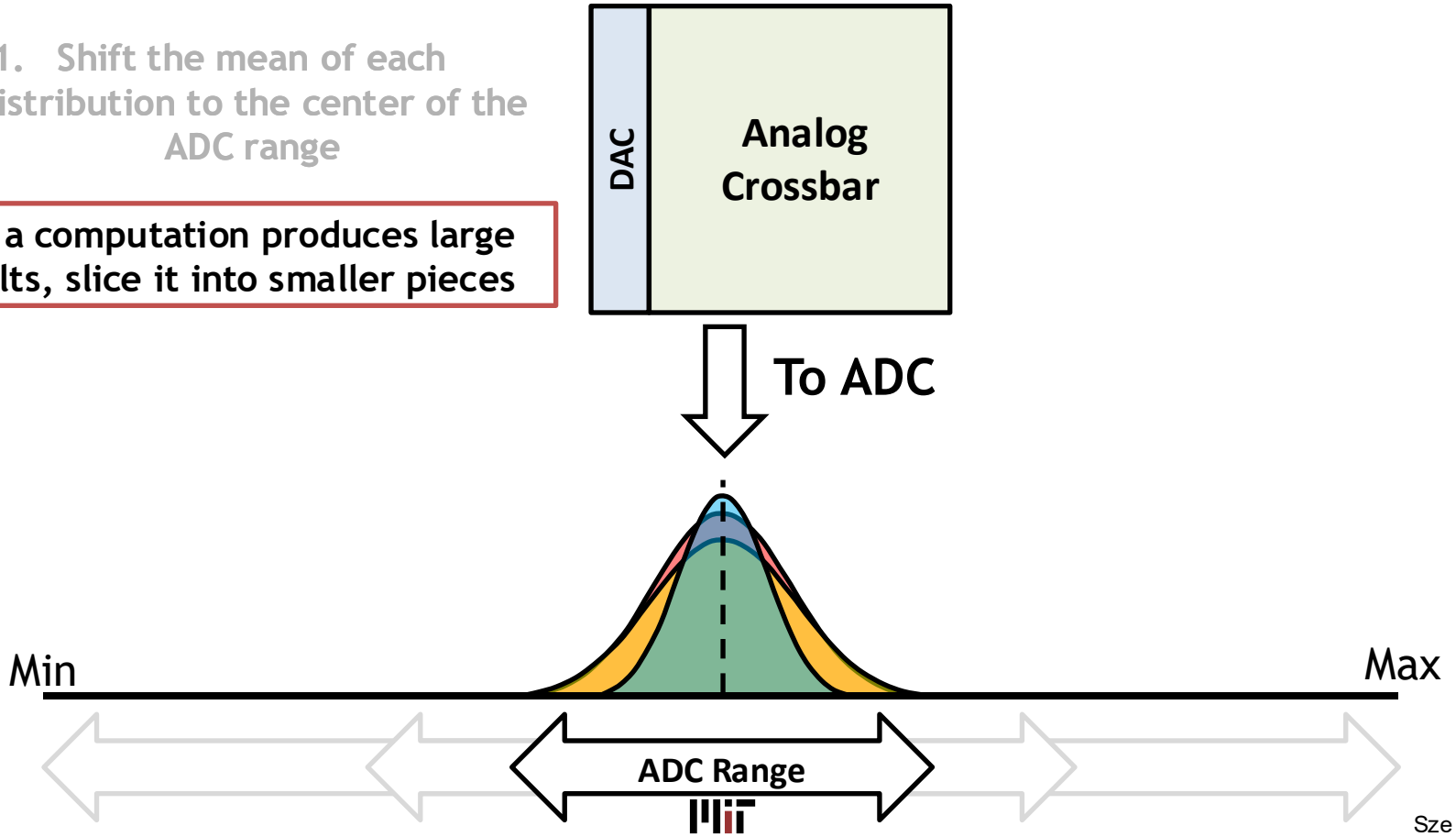
1. Shift the mean of each distribution to the center of the ADC range



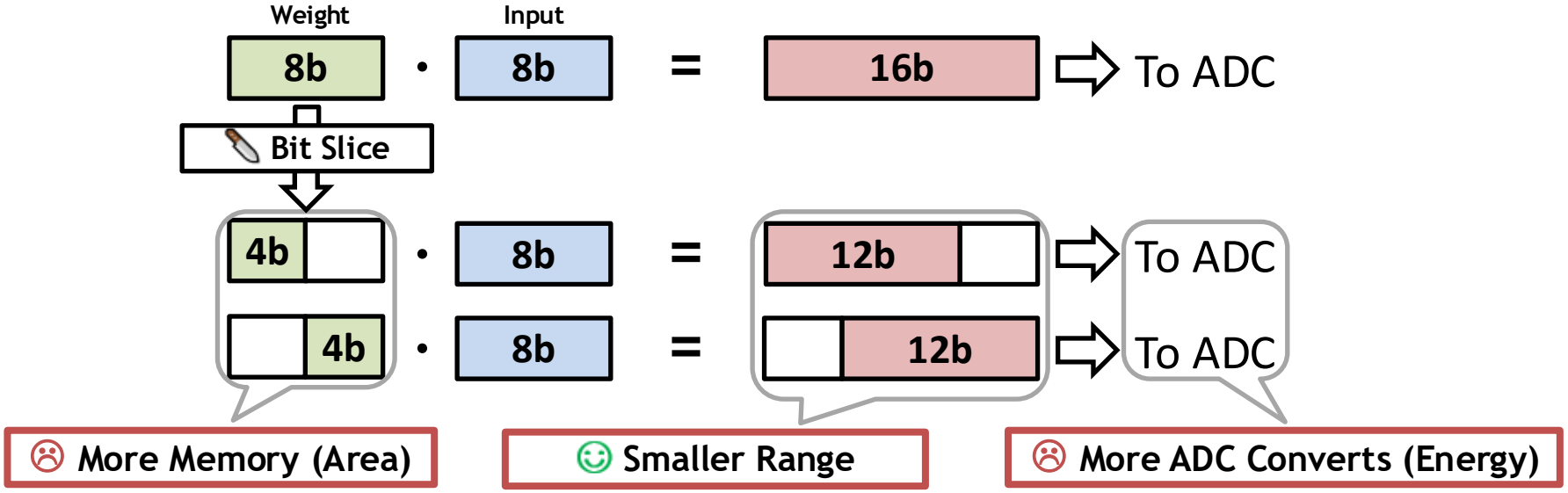
RAELLA: Reshape Distributions

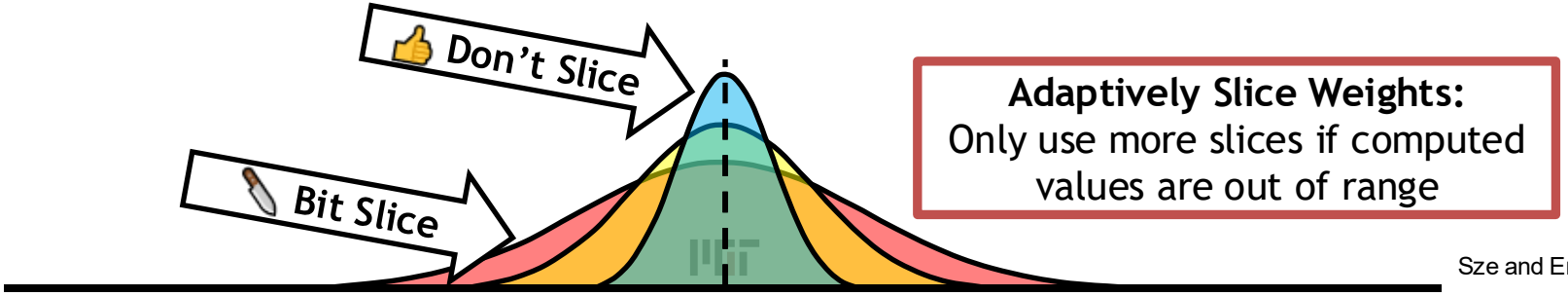
1. Shift the mean of each distribution to the center of the ADC range

2. If a computation produces large results, slice it into smaller pieces



Adaptive Weight Slicing: Slice Large-Result Computations



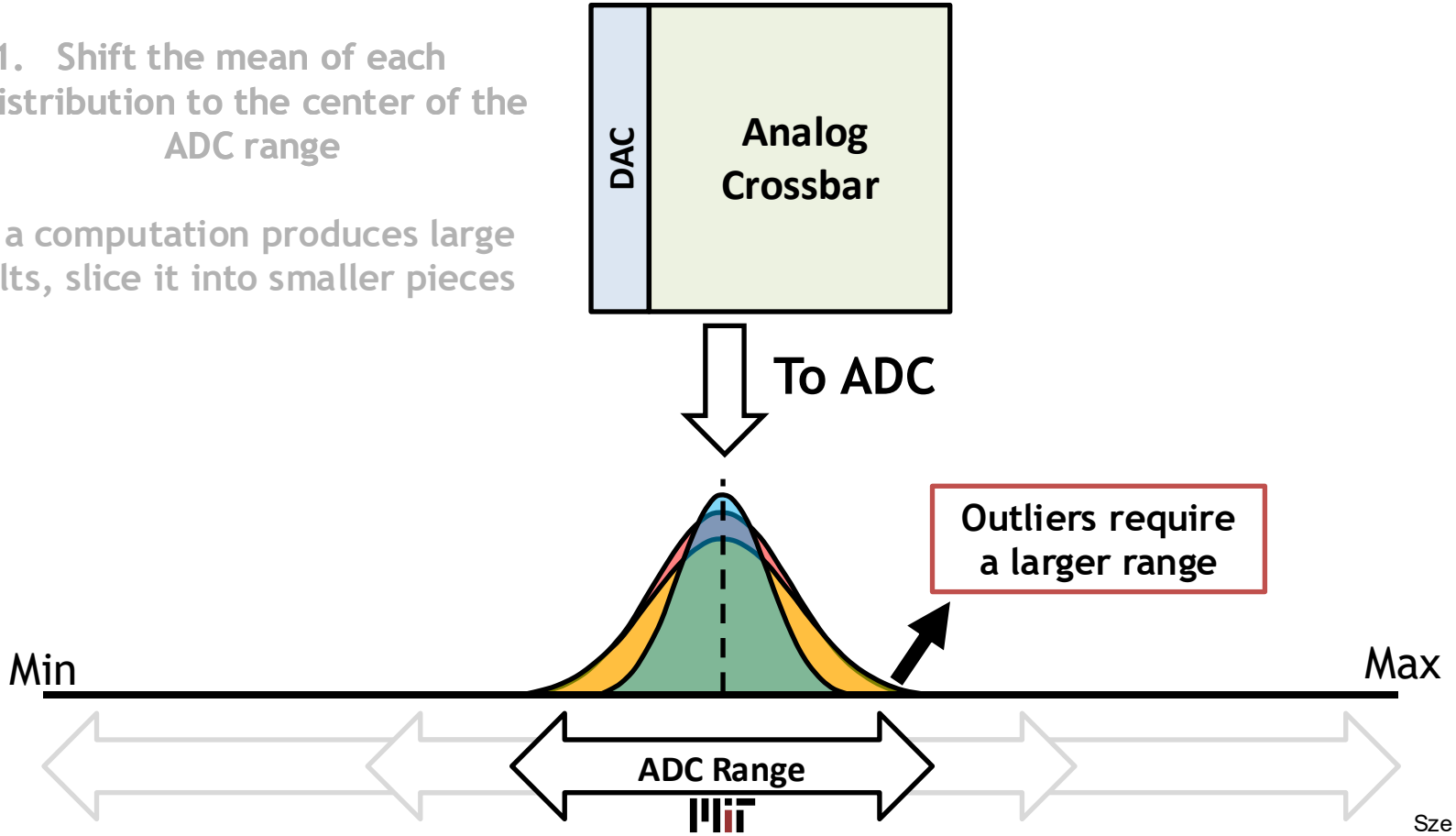


Adaptively Slice Weights:
Only use more slices if computed values are out of range

RAELLA: Reshape Distributions

1. Shift the mean of each distribution to the center of the ADC range

2. If a computation produces large results, slice it into smaller pieces

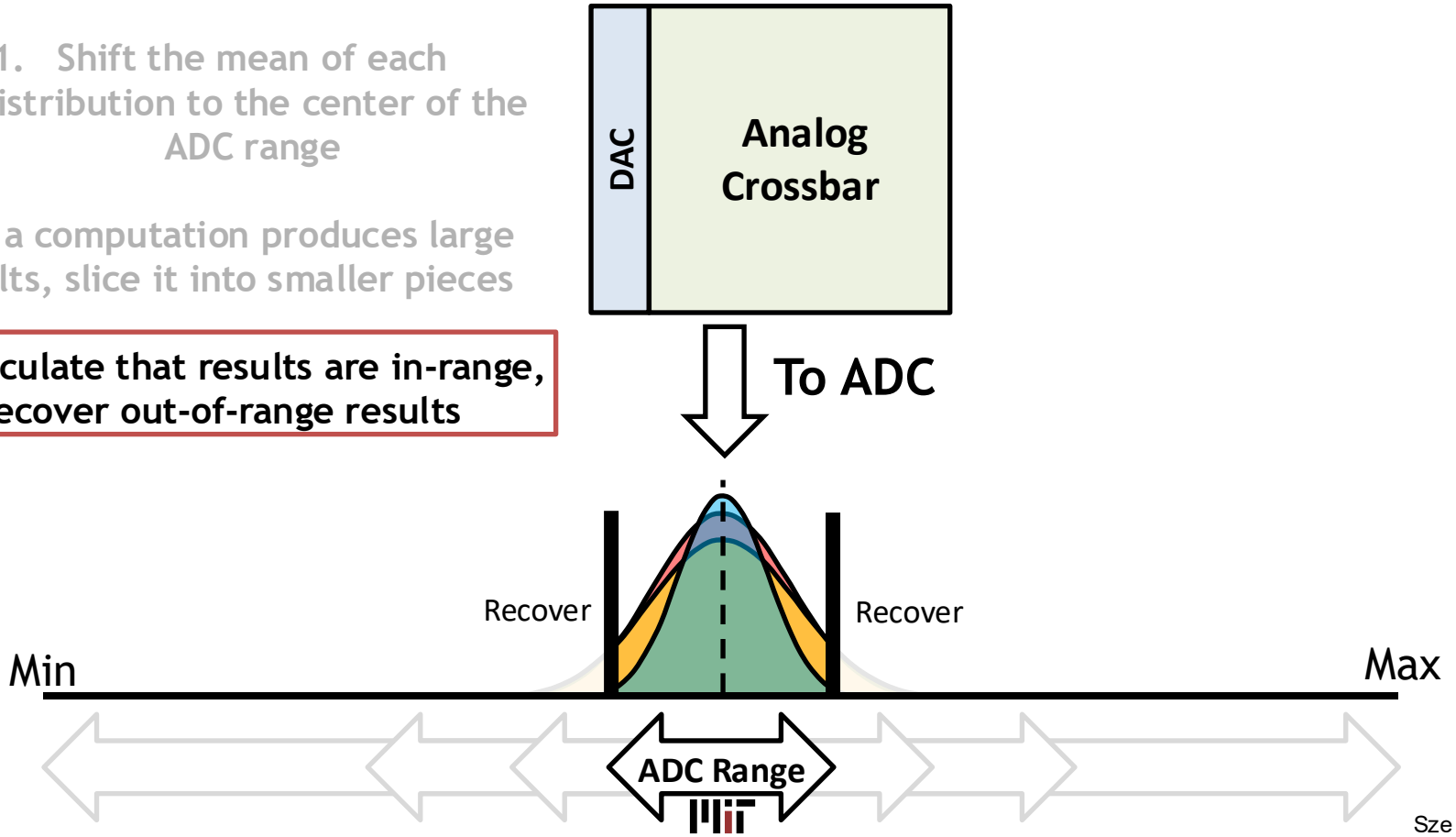


RAELLA: Reshape Distributions

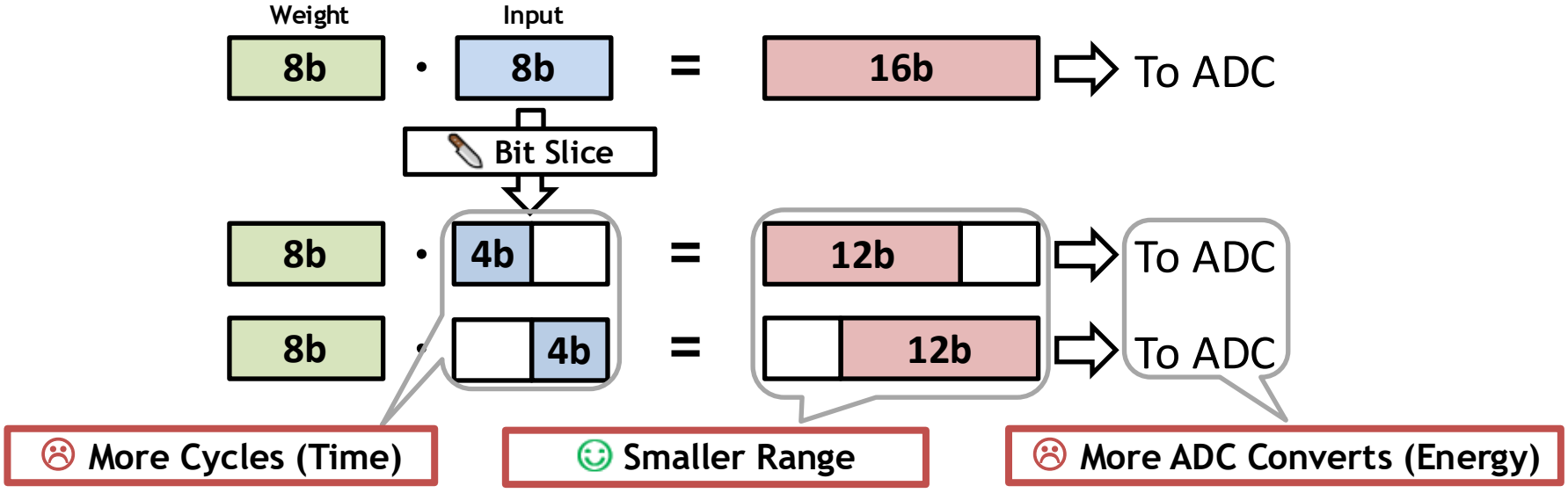
1. Shift the mean of each distribution to the center of the ADC range

2. If a computation produces large results, slice it into smaller pieces

3. Speculate that results are in-range, recover out-of-range results

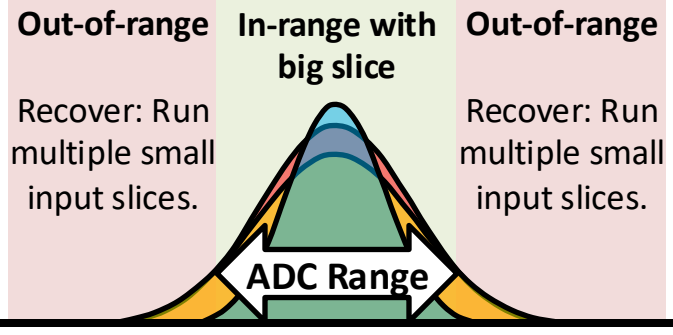


Dynamic Input Slicing: Try Again with Smaller Slices



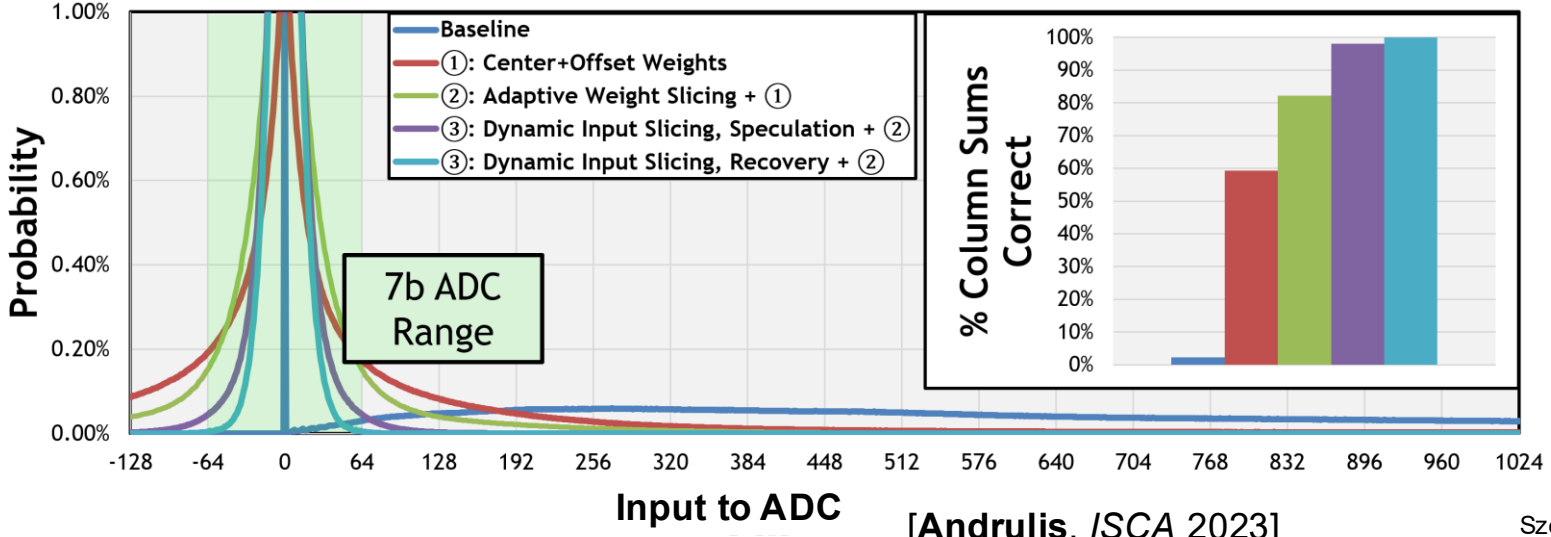
1. Speculate with big input slice

2. Recover out-of-range results with multiple smaller input slices



RAELLA: Reshape Distributions of Input to ADC

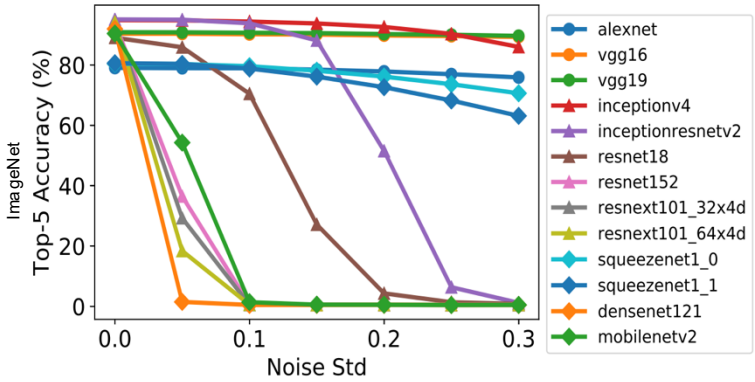
- Makes analog operations produce low-resolution results
  - 1024x reduction of input to ADC
- Enables more compute per ADC convert while using lower-resolution ADCs
  - Improves energy efficiency by 3.9x and throughput by 1.8x compared to iso-area ISAAC
- Maintains DNN accuracy without changing DNN or retraining

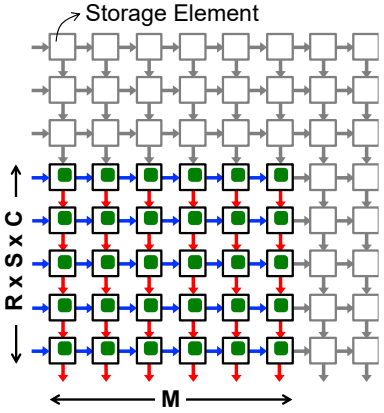


[**Andrulis**, *ISCA* 2023]

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
  - CiM 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

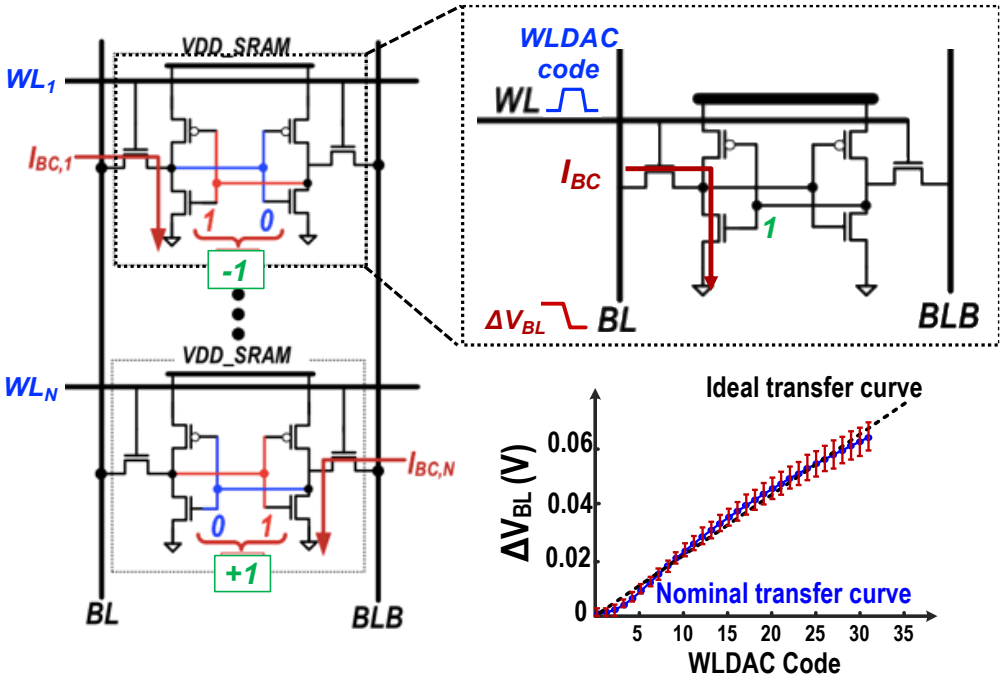




[**Yang**, *IEDM* 2019]

CiM Using SRAM Bit Cell

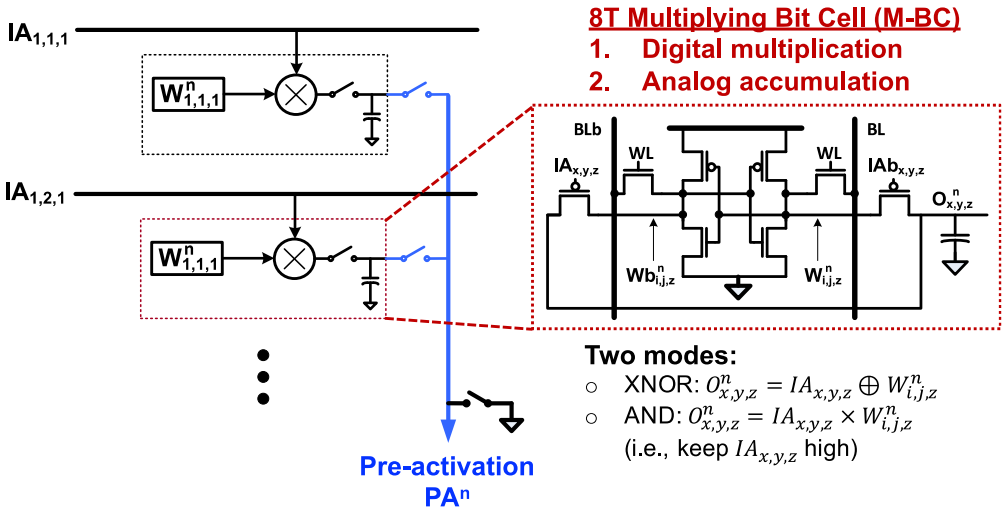
- Multiplication uses I-V relationship of access transistor (WL) and stored value in bit-cell
  - Assumes binary weights and multi-bit input activation
- Addition using **current addition** on bit line (BL)
  - Limited by nonlinearity and sensitivity to variations



[**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

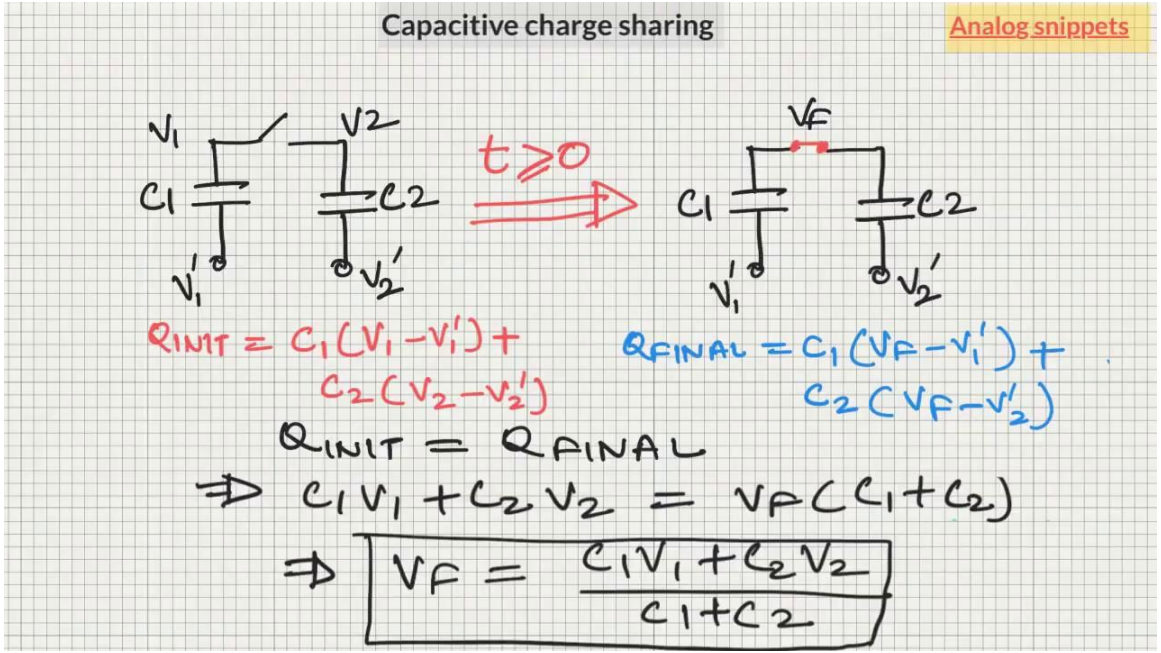


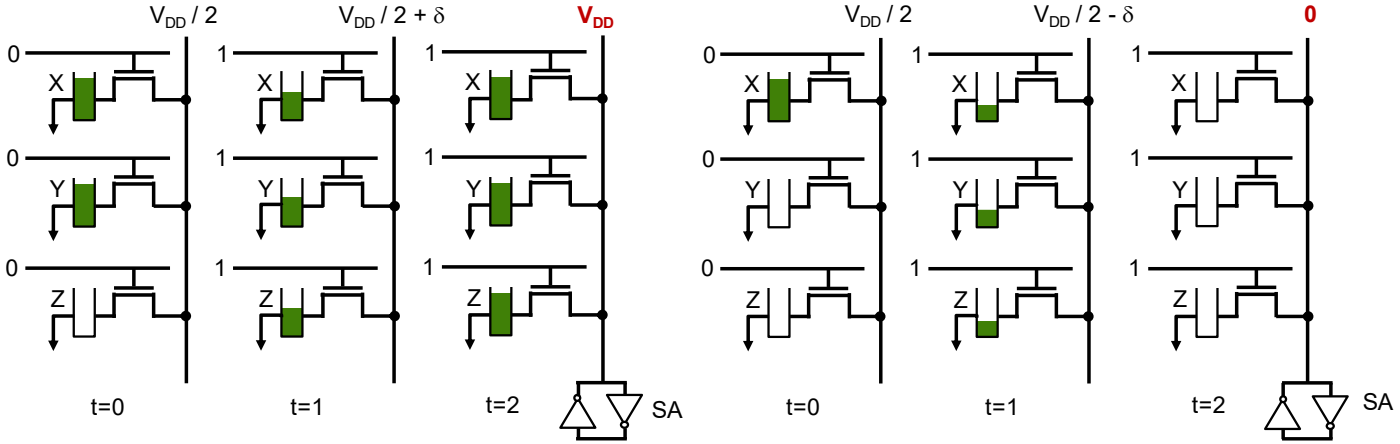
Image Source: https://www.youtube.com/watch?v=XRQ_Xldr2nk

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

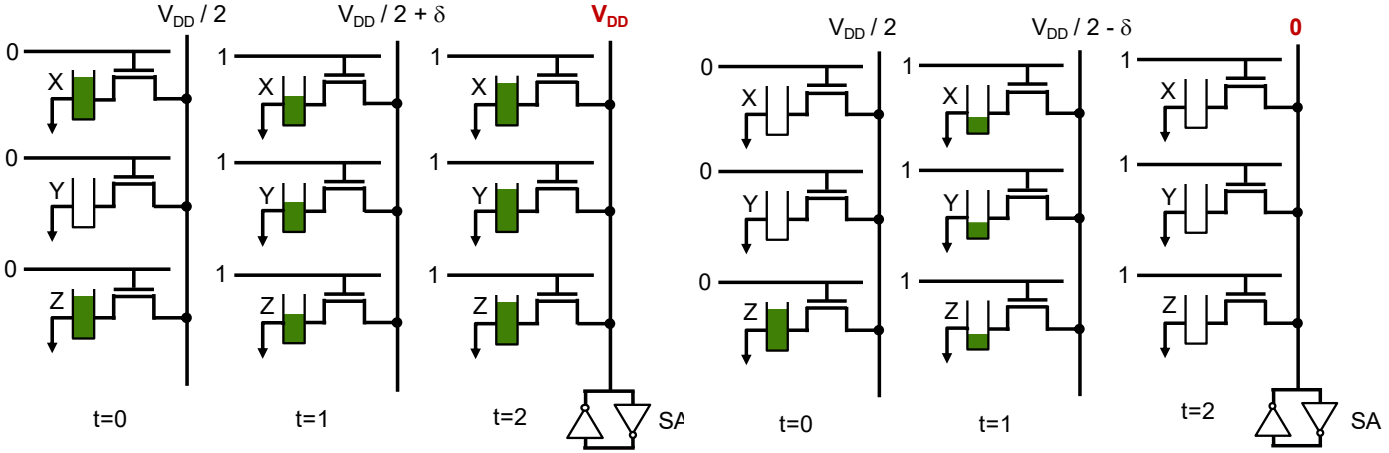
AND (X=1, Y=0) = 0

[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)

Lots of Compute In Memory Research Across the Stack

SESSION 34 Wednesday, February 21st, 1:30 PM

Compute-In-Memory

Session Chair: Ankur Agrawal, *IBM T. J. Watson Research Center, Yorktown Heights, NY*

Session Co-Chair: Eric Wang, *TSMC, Hsinchu City, Taiwan*

1:30 PM

34.1 A 28nm 83.23TFLOPS/W POSIT-Based Compute-in-Memory Macro for High-Accuracy AI Applications

Y. Wang, X. Yang, Y. Qin, Z. Zhao, R. Guo, Z. Yue, H. Han, S. Wei, Y. Hu, S. Yin

Tsinghua University, Beijing, China

1:55 PM

34.2 A 16nm 96Kb Integer-Floating-Point Dual-Mode Gain-Cell-Computing-in-Memory Macro Achieving 73.3–163.3TOPS/W and 33.2-91.2TFLOPS/W for AI-Edge Devices

W-S. Khwa*1, P-C. Wu*2, J-J. Wu2, J-W. Su2,3, H-Y. Chen2, Z-E. Ke2, Y-C. Chou2, J-M. Hou2, C-Y. Cheng2, Y-C. Chen2, C-C. Lo2, R-S. Liu2, C-C. Hsieh4, K-T. Tang4, M-F. Chang1,2

1TSMC Corporate Research, Hsinchu, Taiwan; 2National Tsing Hua University, Hsinchu, Taiwan; 3Industrial Technology Research Institute, Hsinchu, Taiwan; *Equally Credited Authors (ECAs)

2:20 PM

34.3 A 22nm 64kb Lightning-Like Hybrid Computing-in-Memory Macro with Compressed Adder Tree and Analog-Storage Quantizers for Transformer and CNNs

A. Guo1, X. Chen1, F. Dong1, J. Chen1, Z. Yuan2,3, X. Hu2, Y. Zhang2, J. Zhang2, Y. Tang2, Z. Zhang2, G. Chen2, D. Yang2, Z. Zhang2, L. Ren2, T. Xiong2, B. Wang2, B. Liu1, W. Shan1, X. Liu1, H. Cai1, G. Sun2, J. Yang2, X. Si1

1Southeast University, Nanjing, China; 2Peking University, Beijing, China; 3HOUMO, Beijing, China

2:45 PM

34.4 A 3nm 32.5TOPS/W, 55.0TOPS/mm² and 3.78Mb/mm² Fully-Digital Computing-in-Memory Macro Supporting INT12 x INT12 with a Parallel-MAC Architecture and Foundry 6T SRAM Bit Cell

H. Fujiwara1, H. Mori1, W-C. Zhao1, K. Khare1, C-E. Lee1, X. Peng2, V. Joshi2, C-K. Chuang3, S-H. Hsu3, T. Hashizume4, T. Nagayama4, C-H. Tien4, Y-Y. Liu4, Y-C. Lu4, C-E. Lee4, T-L. Chou4, K. Akarvardar4, S. Adham4, Y. Wang5, Y-D. Chih1, Y-H. Chen1, H-J. Liao1, T-Y. J. Chang1

1TSMC, Hsinchu, Taiwan; 2TSMC, San Jose, CA; 3TSMC, Ottawa, Canada; 4TSMC, Yokohama, Japan

3:00 PM

34.5 A 818-4094TOPS/W Capacitor-Reconfigured CIM Macro for Unified Acceleration of CNNs and Transformers

K. Yoshioka, Keio University, Yokohama, Japan

Break 3:15 PM

3:35 PM

34.6 A 28nm 72.12TFLOPS/W Hybrid-Domain Outer-Product Based Floating-Point SRAM Computing-in-Memory Macro with Logarithm Bit-Width Residual ADC

Y. Yuan1,2, Y. Yang2, X. Wang2, X. Li2, C. Ma1,2, Q. Chen2, M. Tang2, X. Wei2, Z. Hou2, J. Zhu1,2, H. Wu1,2, Q. Ren1,2, G. Xing2, P-I. Mak2, F. Zhang2

1Institute of Microelectronics of the Chinese Academy of Sciences, Beijing, China; 2University of Chinese Academy of Sciences, Beijing, China; 3Beijing Institute of Technology, Beijing, China; 4University of Macau, Macau, China

4:00 PM

34.7 A 28nm 2.4Mb/mm² 6.9-16.3TOPS/mm² eDRAM-LUT-Based Digital-Computing-in-Memory Macro with In-Memory Encoding and Refreshing

Y. He1, S. Fan1, X. Li1, L. Lu2, W. Jia1, C. Tang1, Y. Li1, Z. Huang1, Z. Du1, J. Yue2, X. Li1, H. Yang1, H. Jia1, Y. Liu1

ISSCC 2024

Session C3: AI/ML Accelerators and CiM

10:15 AM, Honolulu 3

Co-Chairs: Jaydeep Kulkarni, University of Texas at Austin
Masanao Yamaoka, Hitachi, Ltd.

10:15 AM

C3.1 122.7 TOPS/W Stdcell-Based DNN Accelerator Based on Transition Density Data Representation, Clock-Less MAC Operation, Pseudo-Sparsity Exploitation in 40 nm, Animesh Gupta[1], Japesh Vohra[1], Viveka Konandur Rajanna[1], Massimo Alioto[1] [1]National University of Singapore

A DNN whose activation magnitude is represented by digital transition density is introduced for low energy, under the proposed Dyadic Digital Transition Modulation (DDTM). MAC operations are simplified into transition counting, enabling 1) activation pseudo-sparsity for lower energy, 2) clock-less neuron operation via simple up-down asynchronous counters. >100 TOPS/W in 40 nm stdcell design is on par with

VLSI 2024

Session 23: Neuromorphic Computing (NC) - Compute-in-Memory for Deep Learning

2:15 PM, Continental 6

Co-Chairs: Martin Frank, IBM and Duygu Kuzum, University of California San Diego

This session describes advances in compute-in-memory (CiM) technologies and 3D integration for deep learning. Such circuits hold promise for deep learning inference and training by enabling faster and more energy-efficient neural network operations than digital CMOS. The session will be opened with a report on an in-memory compute chip fabricated in 14 nm CMOS technology that employs a carbon-based liner underneath the phase-change material to improve temporal stability and inference accuracy. The second paper reports on low-temperature monolithic 3D integration of carbon nanotube FETs and resistive RAM (ReRAM) arrays to

IEDM 2024

3:45 PM – 5:00 PM

Session 4A: PIM ACCELERATORS

Location: Pacífico B (floor map)

Session Chair: Jongse Park

3:45 PM – 4:00 PM

PreSto: An In-Storage Data Preprocessing System for Training Recommendation Models

Y. Lee, H. Kim, M. Rhu

4:00 PM – 4:15 PM

pSyncPIM: Partially Synchronous Execution of Sparse Matrix Operations for All-bank PIM Architectures

D. Baek, S. Hwang, J. Huh

4:15 PM – 4:30 PM

NDSearch: Accelerating Graph-Traversal-Based Approximate Nearest Neighbor Search through Near Data Processing

Y. Wang, S. Li, Q. Zheng, L. Song, Z. Li, A. Chang, H. Li, Y. Chen

4:30 PM – 4:45 PM

Enabling Efficient Large Recommendation Model Training with Near CXL Memory Processing

H. Liu, L. Zheng, Y. Huang, J. Zhou, C. Liu, R. Wang, X. Liao, H. Jin, J. Xue

4:45 PM – 5:00 PM

Exploiting Similarity Opportunity of Emerging AI Models on 3D Hybrid Bonding Architecture

ISCA 2024

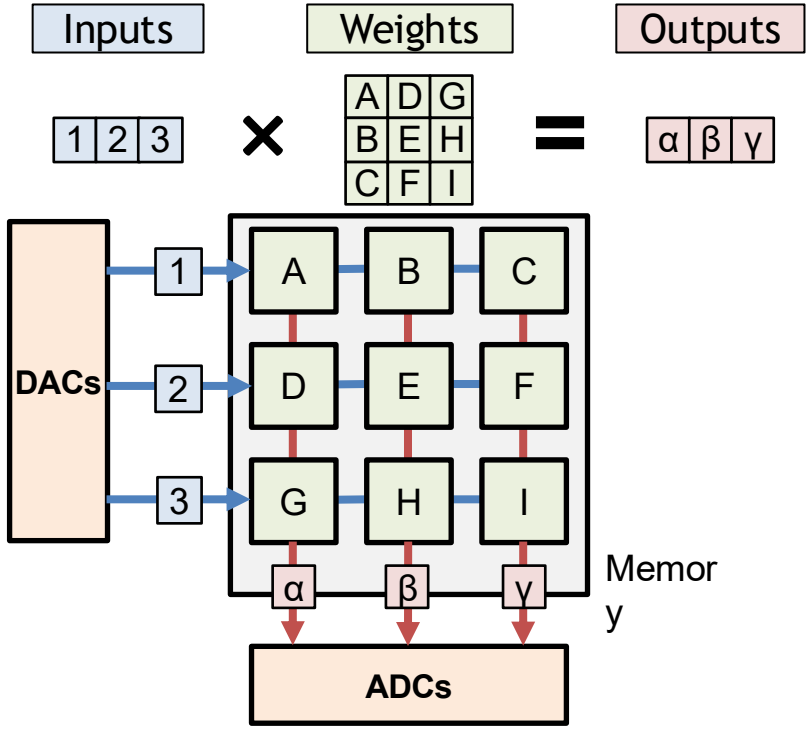
Many dedicated sessions on CiM at architecture, circuits, and devices conferences

CiM Research Spans Full Stack

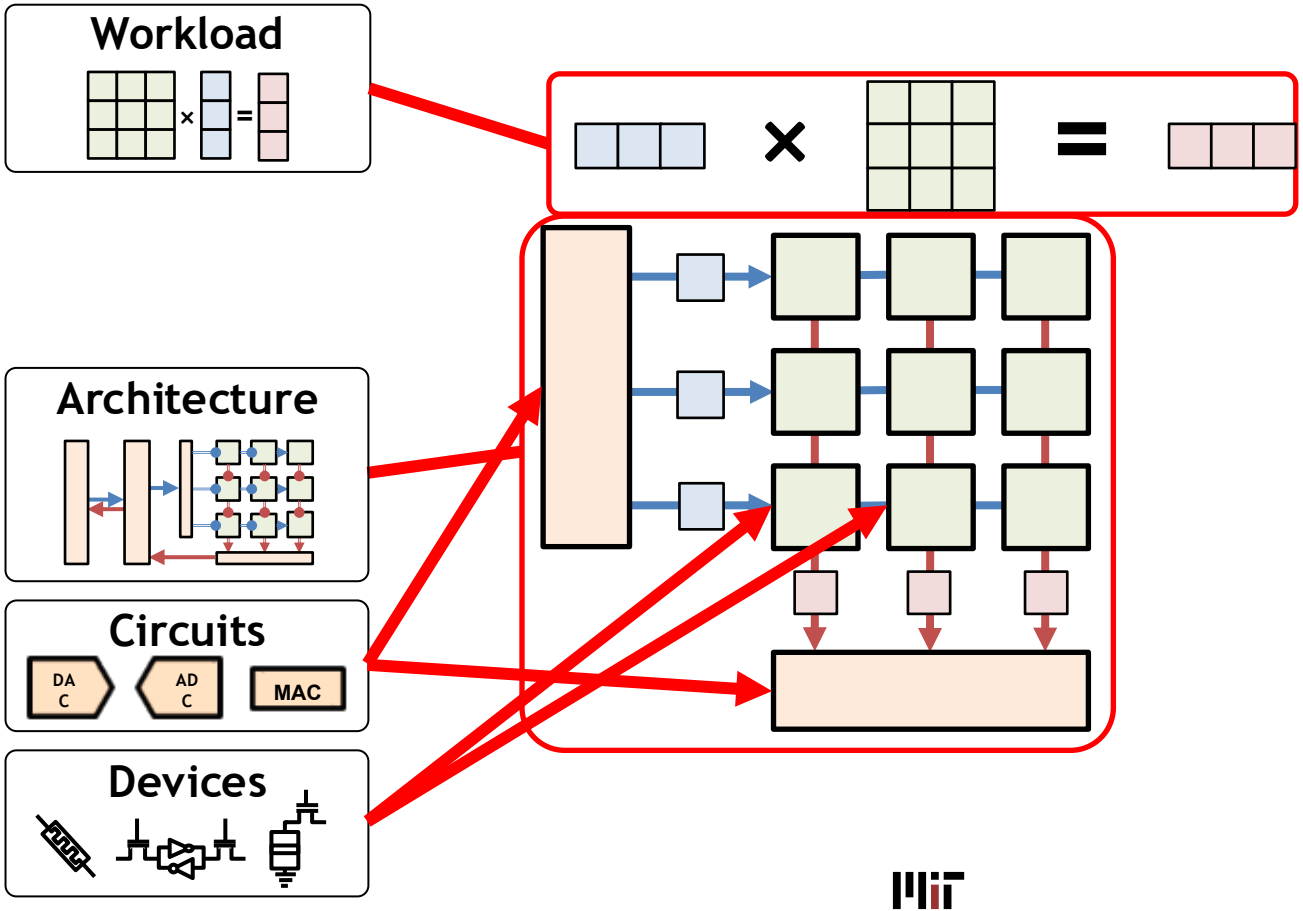
- **Devices:** The components forming each memory cell (e.g., SRAM, DRAM, ReRAM, STT-RAM)
- **Circuits:** The components performing computation, analog/digital conversion, storage, data movement, and other actions
- **Architecture:** The organization of components into a larger system (e.g., the number 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)

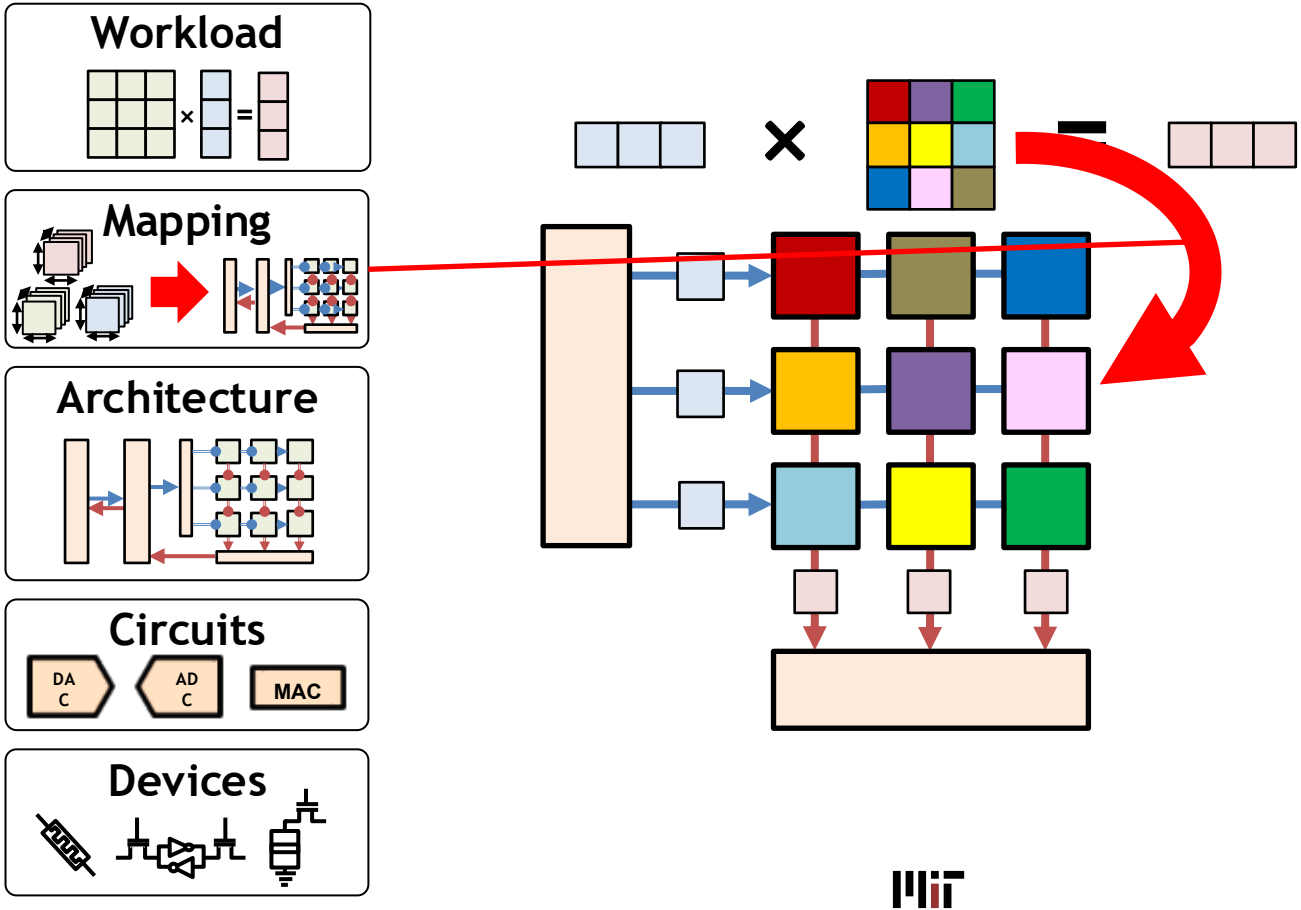
Compute In Memory (CiM) Accelerators



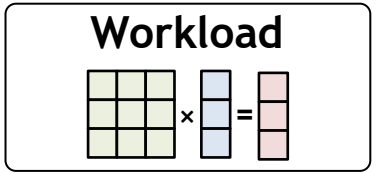
The CiM Stack: Large Design Space

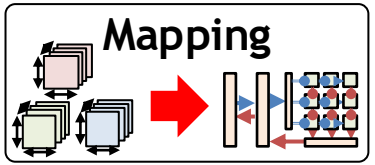


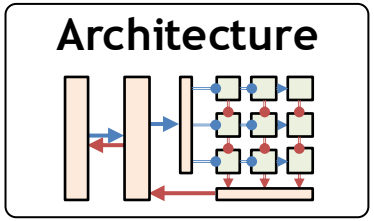
The CiM Stack: Large Design Space

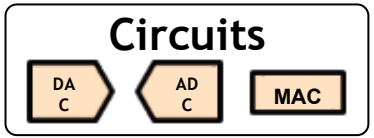


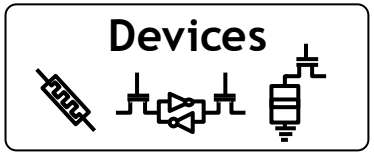
The CiM Stack: Large Design Space

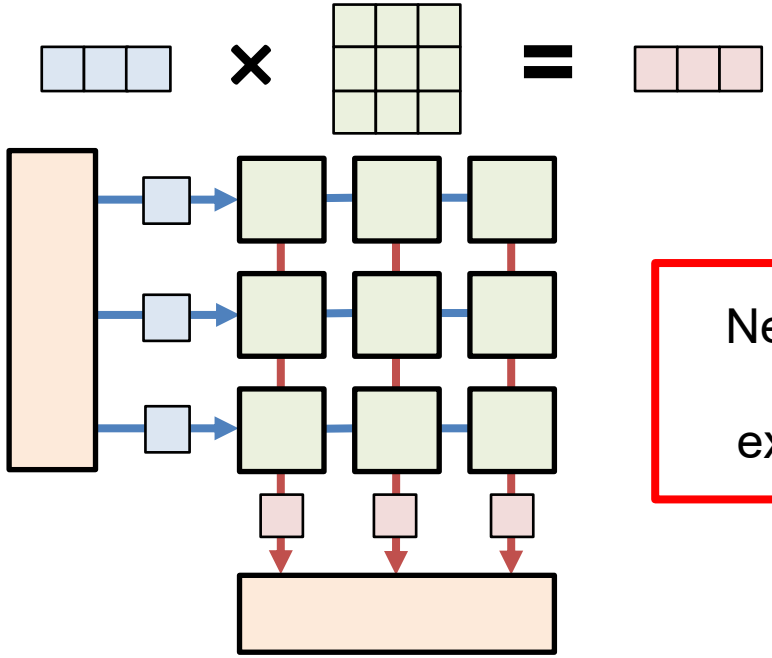






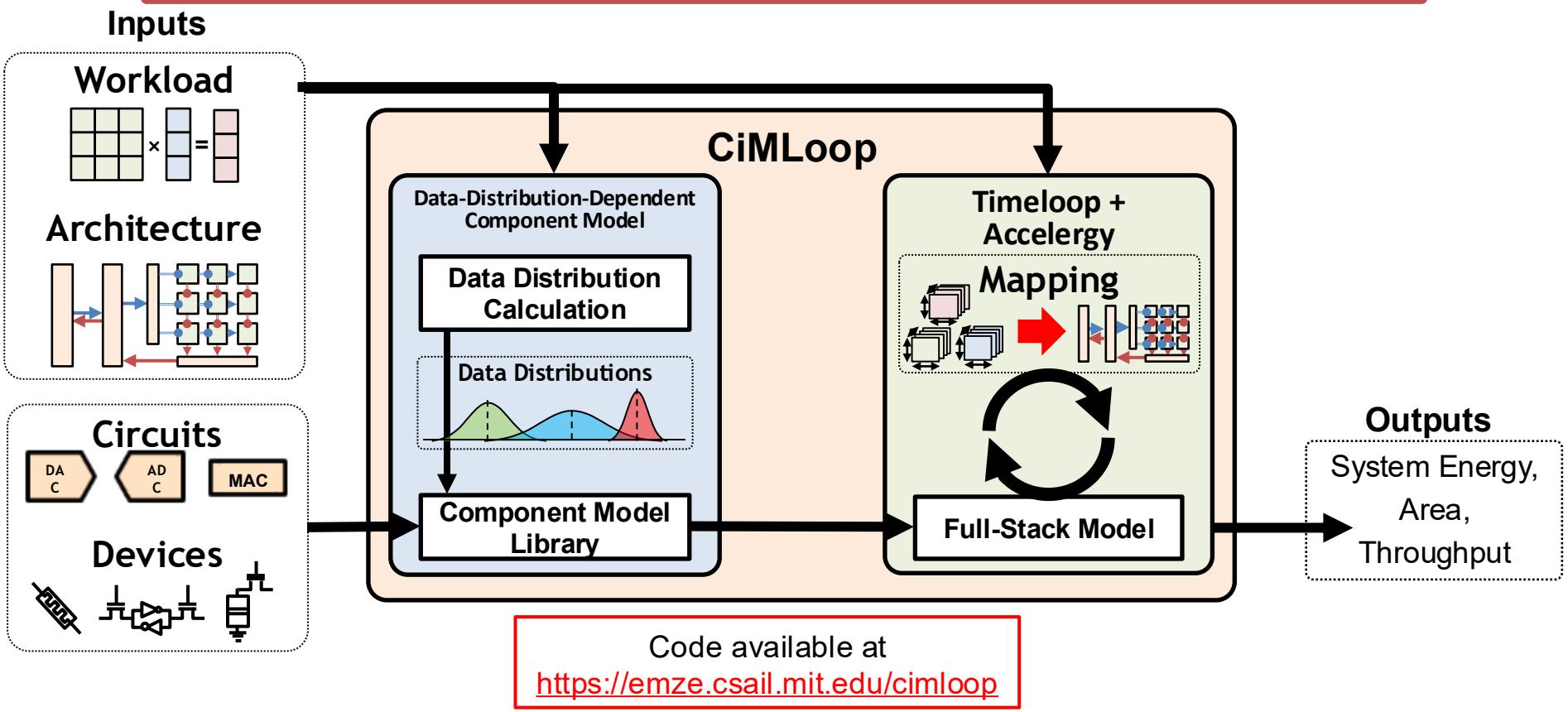






Need for modeling tool to enable design space exploration → **CiMLoop**

CiMLoop: A Flexible, Accurate, and Fast CiM Modeling Tool



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

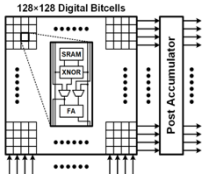
ISPASS Best Paper Award

CiMLoop: A Flexible, Accurate, and Fast CiM Modeling Tool

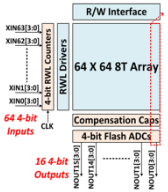
- Flexibility to represent co-design space
  - Challenge: There are diverse choices at each level
  - Solution: Flexible user-defined specifications

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

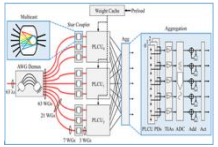
The Co-Design Space: Components



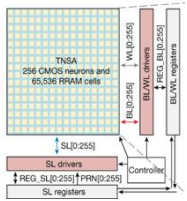
[**Kim**, JSSC 2021]



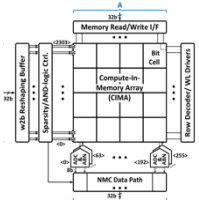
[**Sinangil**, JSSC 2021]



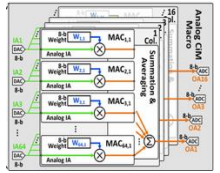
[**Shiflett**, ISCA 2021]



[**Wan**, Nature 2022]



[**Jia**, JSSC 2020]



[**Wang**, VLSI 2022]

Eight-way multiplexed NMC data path

Weighting Capacitors

Arrayed Waveguide Grating

Pulse Train DAC

9T SRAM

Post Accumulator

SRAM + Digital MAC

C-2C Multiplier

8T SRAM

Photodiodes

4b Flash ADC

Asynchronous SAR ADC

Column Registers

R-2R DAC

6T SRAM

Library of circuit and device models

+

Plug-in interface for users to create more models

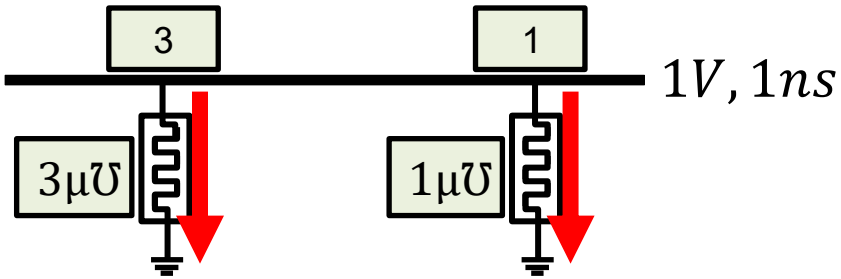
CiMLoop: A Flexible, Accurate, and Fast CiM Modeling Tool

- Flexibility to represent co-design space
  - Challenge: There are diverse choices at each level
  - Solution: Flexible user-defined specifications
- Accurately model energy
  - Challenge: Workload values and representation affect component energy
  - Solution: Models capture these cross-stack interactions (error within 8%)

Accurately Modeling Energy: Data-Value-Dependence

$$Voltage^2 \times Conductance \times Time$$

$$Conductance\ \mho = \frac{1}{Resistance\ \Omega}$$



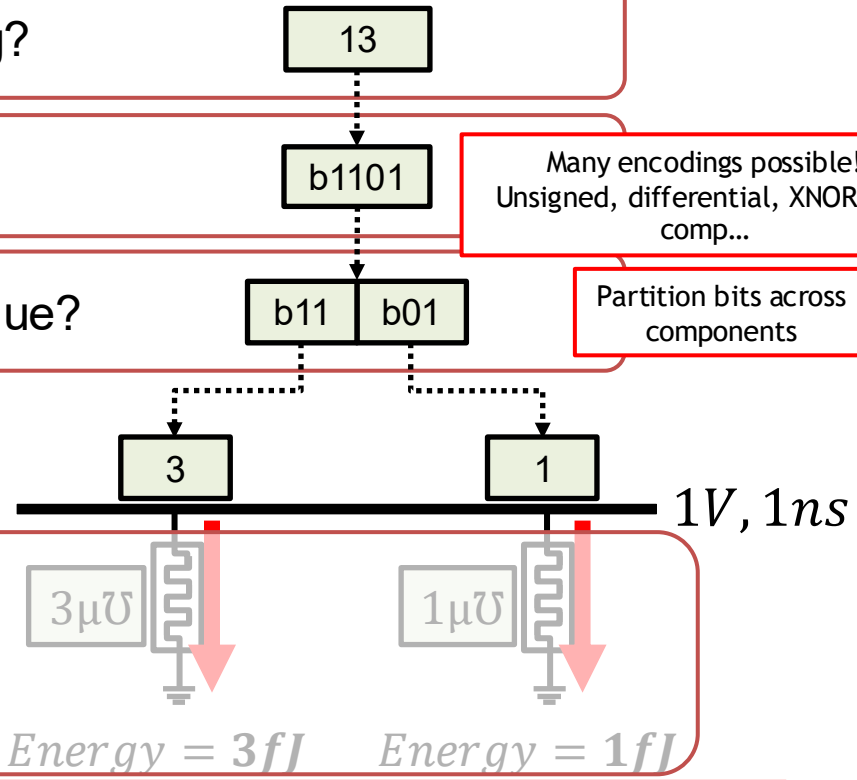
Data-value-dependence significantly impacts device and circuit energy
Prior works assume fixed energy → significant error

Accurately Modeling Energy: Data-Value-Dependence

What value are we processing?

How do we represent it?

Where do we map bits of this value?



Many encodings possible! Unsigned, differential, XNOR, 2's comp…

Partition bits across components

Calculate Energy

$$Voltage^2 \times Conductance \times Time$$

$$Conductance\ \mho = \frac{1}{Resistance\ \Omega}$$

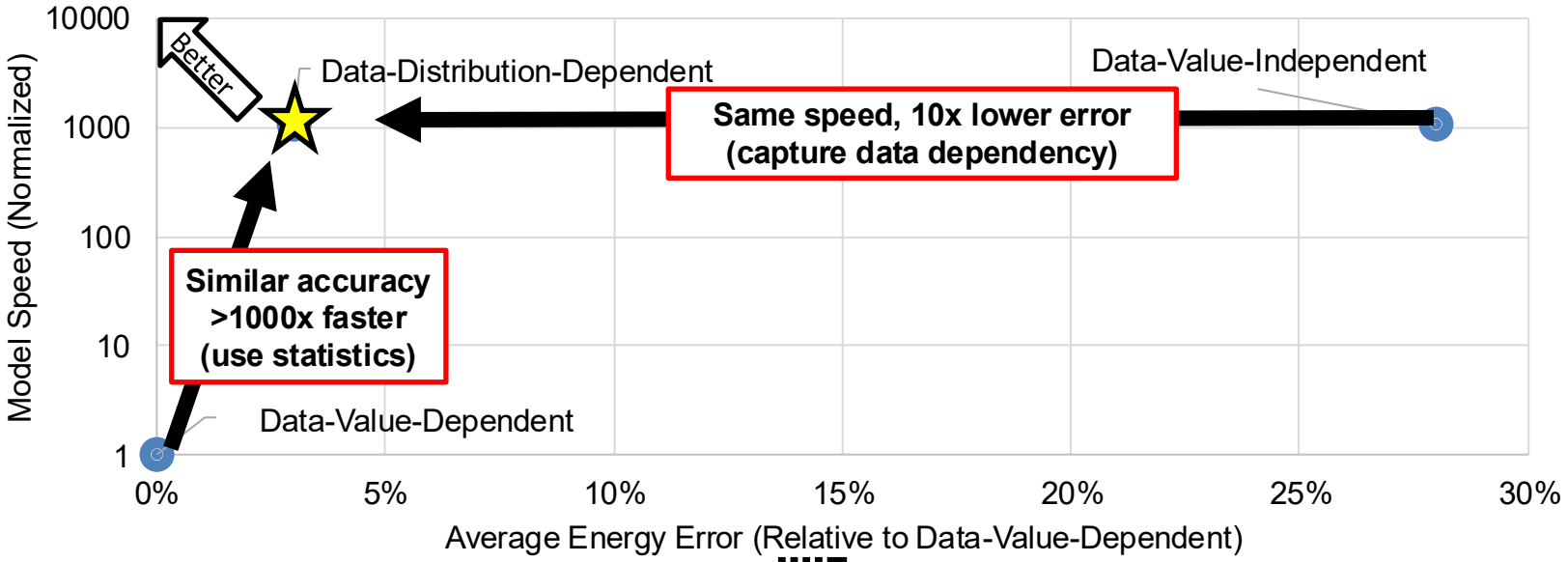
Capture data-value-dependence:
What values are there? ***How*** do we represent them? ***Where*** do we map their bits?

CiMLoop: A Flexible, Accurate, and Fast CiM Modeling Tool

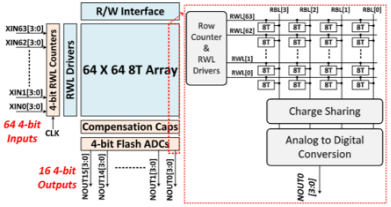
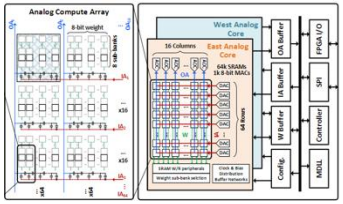
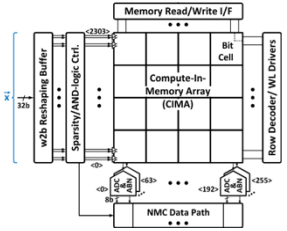
- Flexibility to represent co-design space
  - Challenge: There are diverse choices at each level
  - Solution: Flexible user-defined specifications
- Accurately model energy
  - Challenge: Workload values and representation affect energy
  - Solution: Models capture these cross-stack interactions (error within 8%)
- Fast exploration of co-design space
  - Challenge: Accurate energy models may simulate many (>10^{12}) values
  - Solution: Statistical models that are 1000x faster than prior accurate models

Fast and Accurate Statistical Energy Modeling

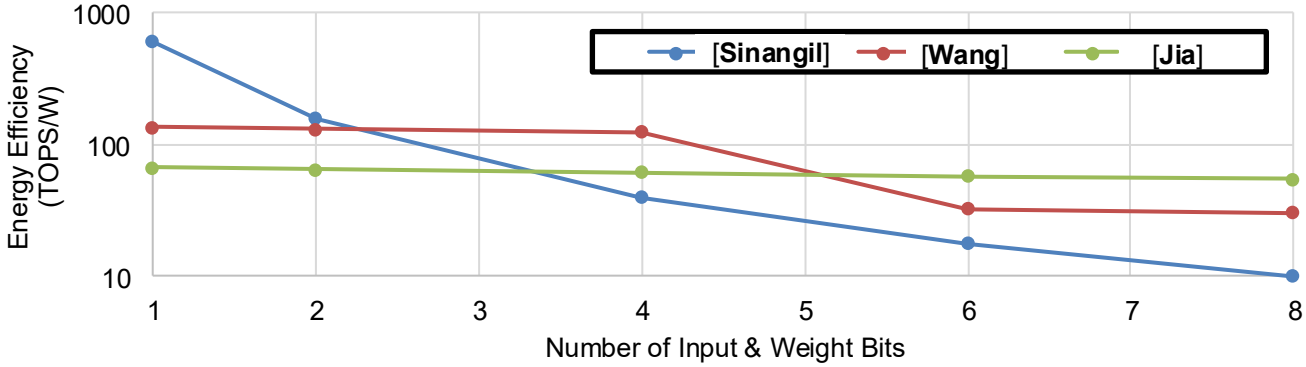
	Data-Value-Independent Timeloop [**Parashar**, *ISPASS* 2019]	**Data-Value-Dependent** NeuroSim [**Peng**, *TCAD* 2021]	**Data-Distribution-Dependent** CiMLoop [**Andrulis**, *ISPASS* 2024]
Model Accuracy	Low	High	High
Model Speed	High	Low	High



Example: Apples-to-Apples Comparison

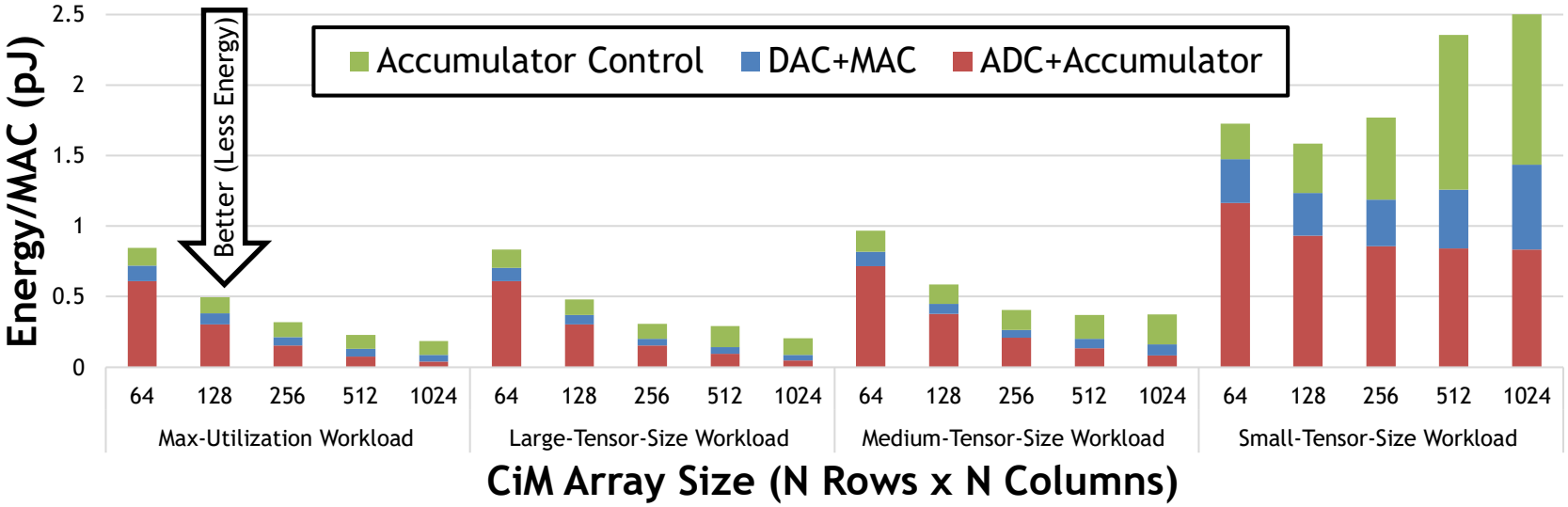
	[**Sinangil**, *JSSC* 2021]	[**Wang**, *VLSI* 2022]	[**Jia**, *JSSC* 2020]
Macro			
Technology Node	7nm	22nm	65nm
ADC Type	4b Flash	8b SAR	8b SAR
Memory Device	6T SRAM	8T SRAM + Capacitor	6T SRAM

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



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

Example: Design Space Exploration



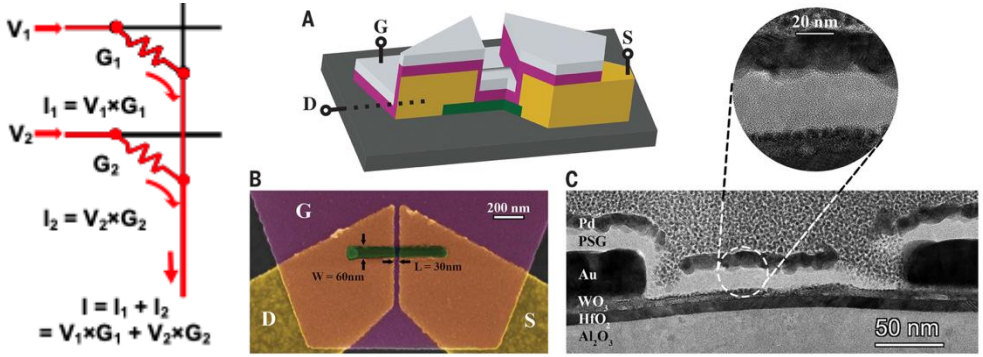
Explore array size (architecture) and DNN shapes (workload)

Explore in Final Project!

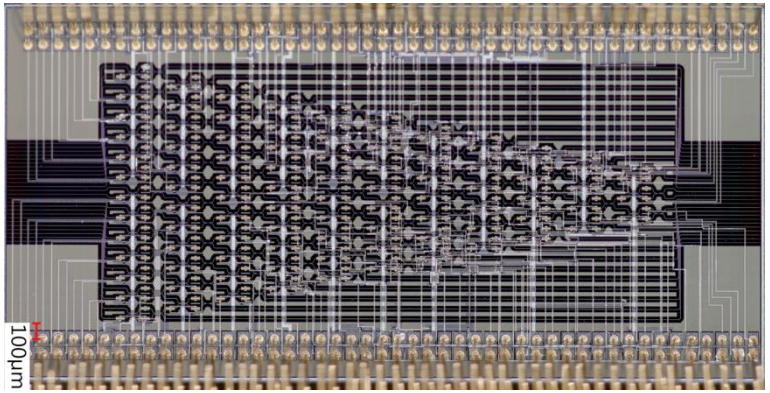
[**Andrulis**, *ISPASS* 2024]

CiMLoop Enabling Collaborations Across Stack at MIT

Computing in memory with programmable resistive devices [w/ Jesus del Alamo]



Computing with light [w/ Dirk Englund]



Ultra-low Power Superconducting Electronics for Deep Learning Accelerator Architectures: Evaluating Energy Efficiency and Scalability 9.22

L. Camron Blackburn, Evan Golden, Tanner Andrulis, Vivienne Sze, Joel Emer, Neil Gershenfeld, Karl K. Berggren

Sponsorship: MIT Lincoln Laboratory, the MIT AI Hardware Program

Computing with superconducting electronics [w/ Karl Berggren and Neil Gershenfeld]

(Started from 6.5930/1 final project!)

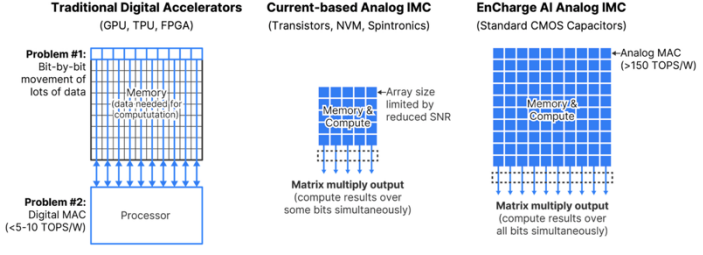
Modeling helps identify level of abstraction and facilitates communication between teams

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.

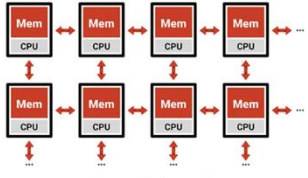


MYTHIC PRODUCTS TECHNOLOGY MARKETS BLOG COMPANY

Technology » Compute-in-Memory

Compute-in-Memory

Boosting memory capacity and processing speed



Compute-in-Memory IC

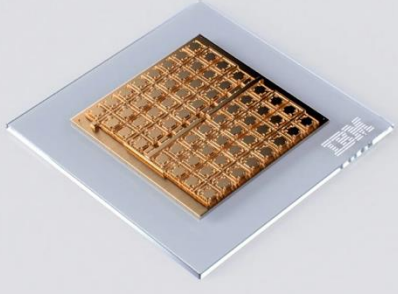
Today's most common computing architectures are built on assumptions about how memory is accessed and used. These systems assume that the full memory space is too large to fit on-chip near the processor, and that we do not know what memory will be needed at what time. To address

IBM Research Focus areas Blog Publications Careers About

Home
↳ Projects

Analog AI

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



Sze and Emer

Compute with Light

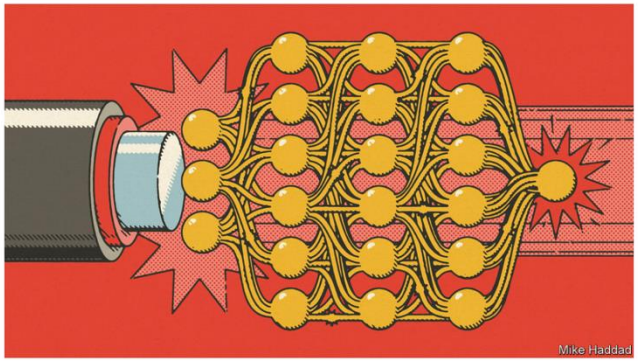
The Economist

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.

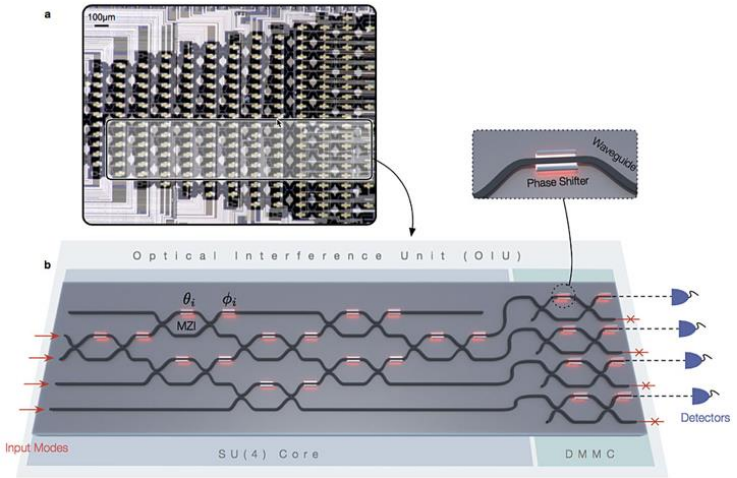
"Unlike electrons, photons (which are electrically neutral) can cross each others' paths without interacting, so glass fibres can handle many simultaneous signals in a way that copper wires cannot. **An optical computer could likewise do lots of calculations at the same time.** Using photons reduces 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

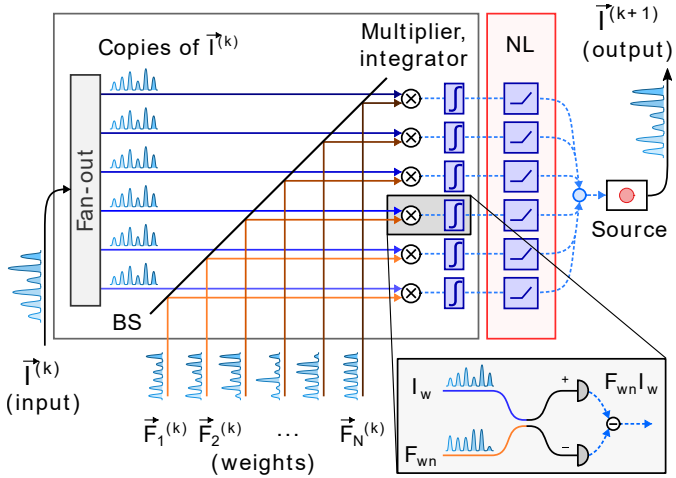
Compute with Light

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

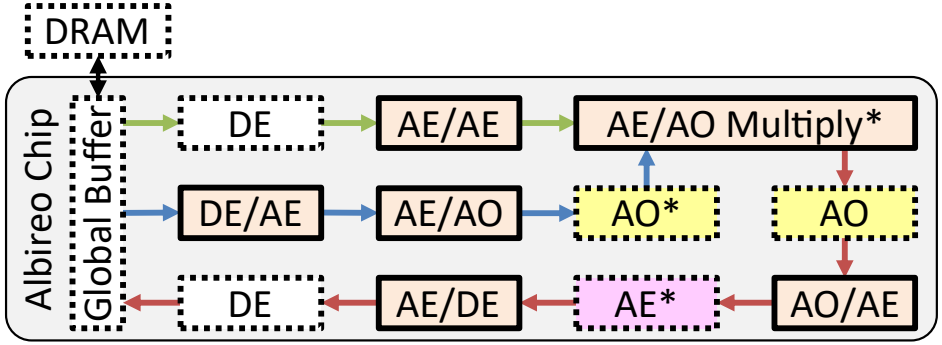
Lightmatter says the computing and power demands of complex neural networks need new technologies like these to keep up.

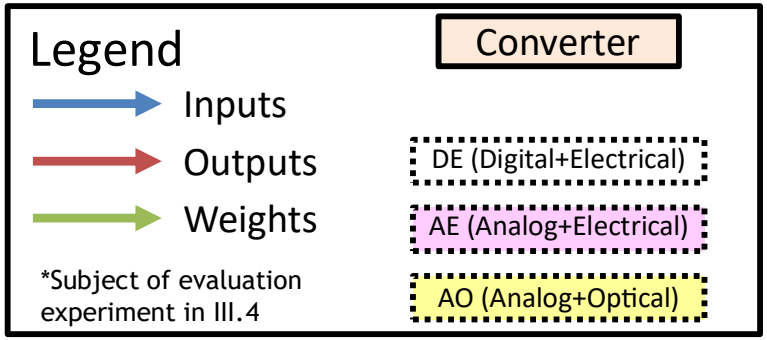
"…chip runs **1.5 to 10 times** faster than a top-of-the-line Nvidia A100 AI chip,

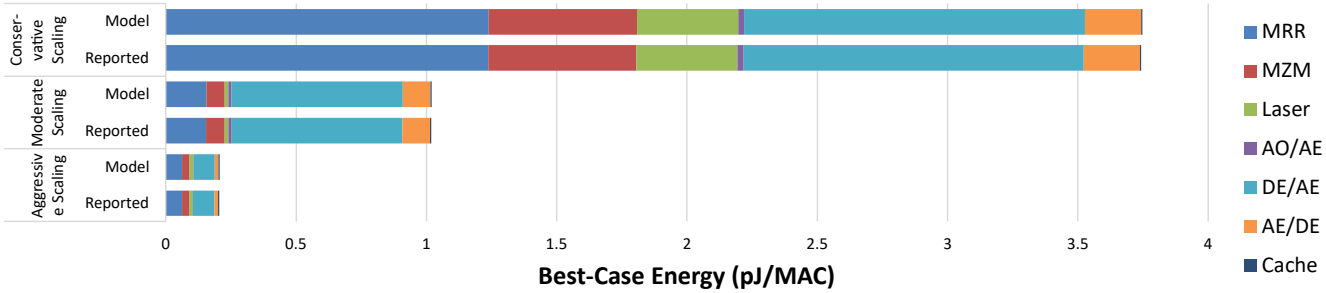
Running a natural language model called BERT, for example, Lightmatter says Envise is **five times faster** than the Nvidia chip; it also consumes **one-sixth of the power**"

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

CiMLoop for Photonics Modeling







Previous final projects in 6.5930 involved modeling and validation

Explore in Final Project!

[**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