Compute In Memory

(In-Memory Computing)

Vivienne Sze and Joel Emer

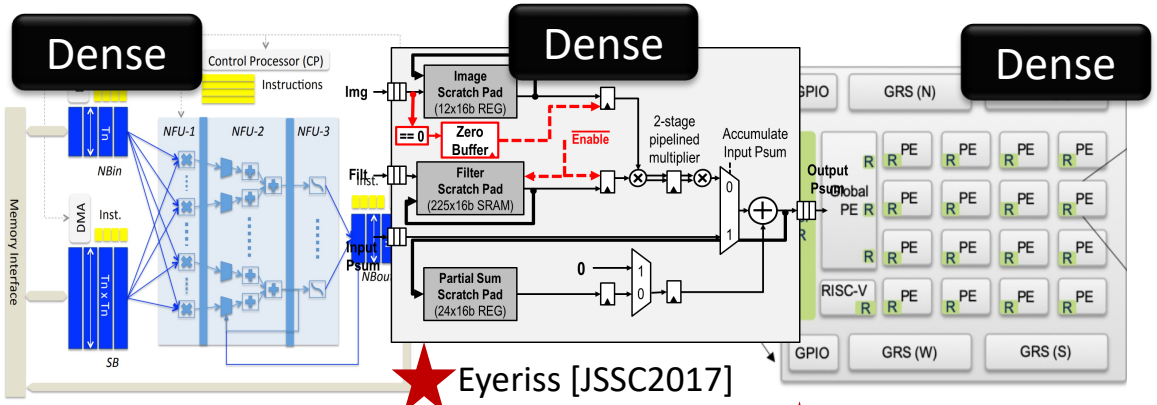
Group Website: http://emze.mit.edu

Research Objectives

- Provide a systematic way to:
  - Describe a wide range of AI hardware accelerators, e.g., deep neural networks (DNN), analog neural networks and (sparse) tensor, that vary in architecture, flexibility and technology
  - Evaluate different designs across a variety of metrics, including latency, bandwidth, area and energy
  - Provide capability to make fair comparisons between different designs across a variety of workloads, with a holistic view of the **entire** system
  - Rapidly explore the design space

Design Space for Deep Neural Networks and Tensor Accelerators

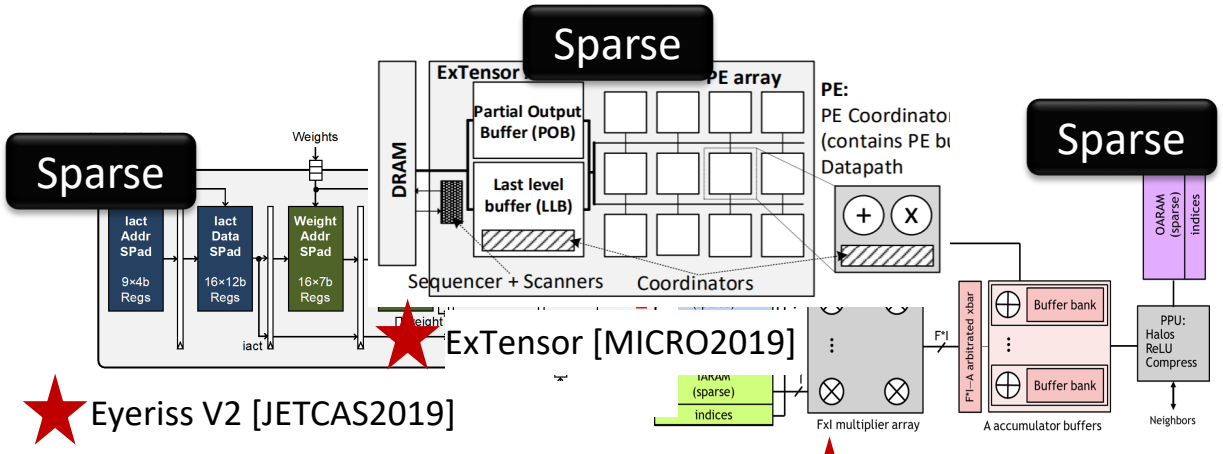
Digital-Compute Accelerator Designs



DianNao [ASPLOS2014]

★ Eyeriss [JSSC2017]

★ Simba [MICRO2019]

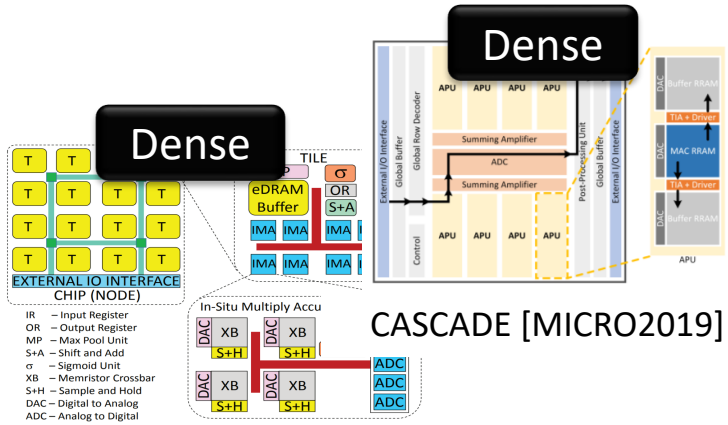


★ Eyeriss V2 [JETCAS2019]

★ ExTensor [MICRO2019]

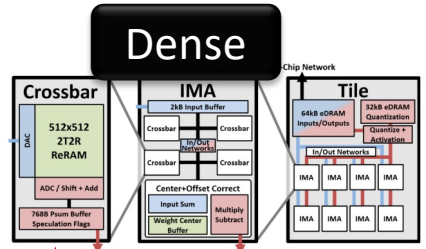
★ SCNN [ISCA2017]

Analog-Compute (CiM) Accelerator Designs

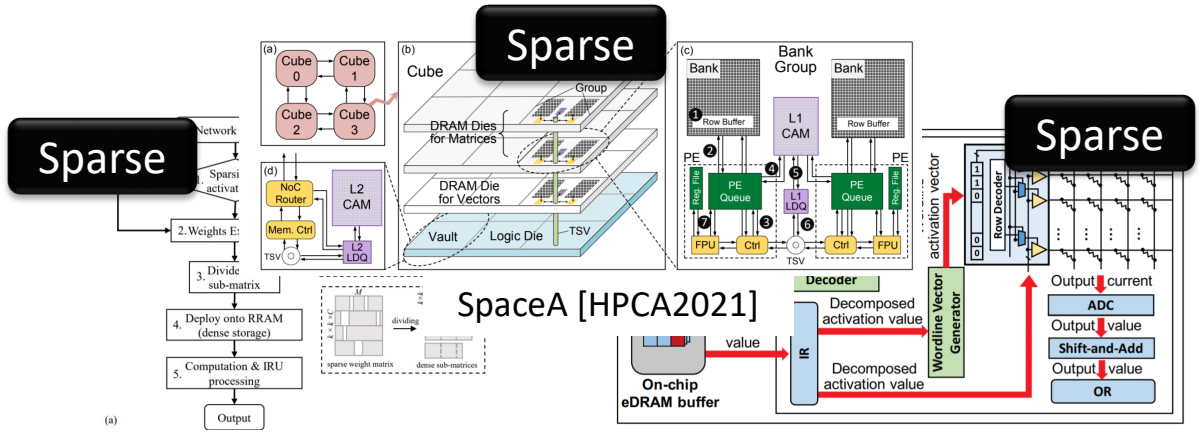


ISAAC [ISCA2016]

★ CASCADE [MICRO2019]



★ RAELLA [ISCA2023]



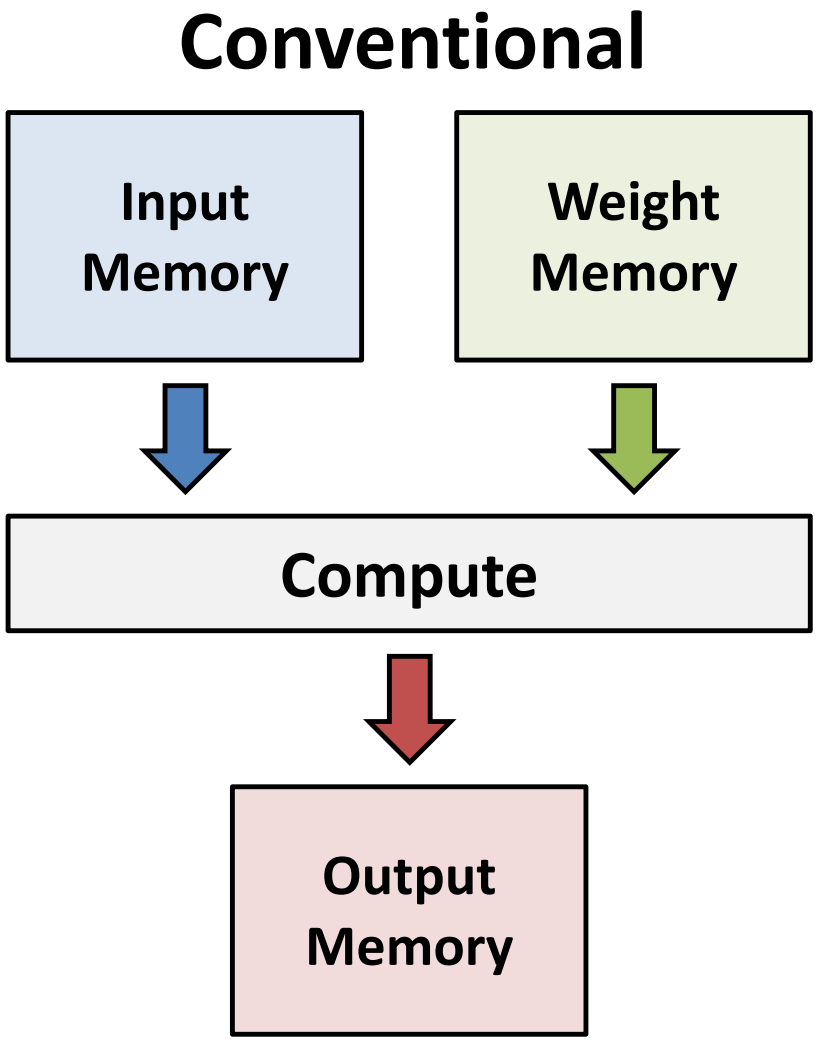
★ SpaceA [HPCA2021]

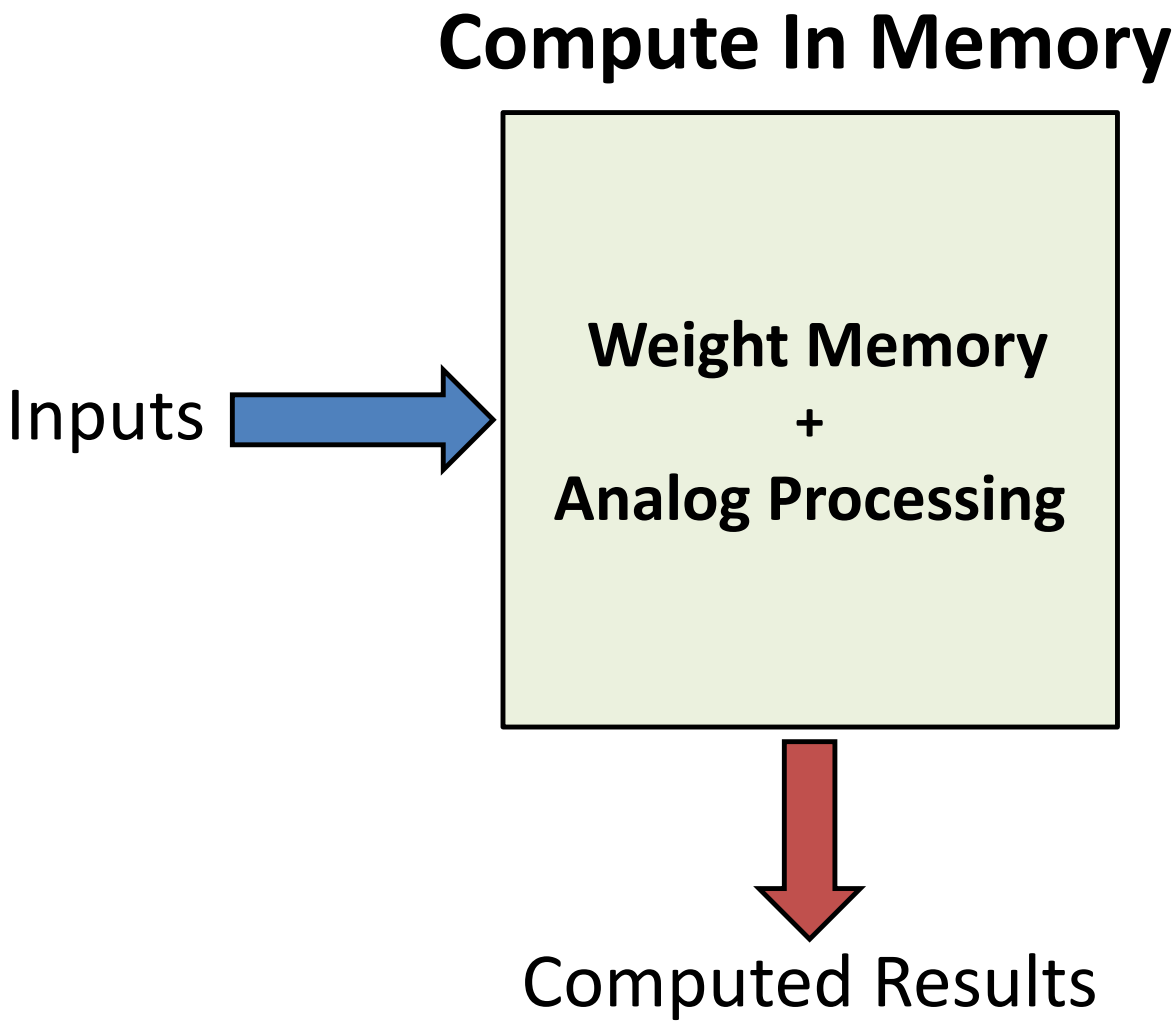
SNrram [DAC2018]

Sparse-ReRAM [ISCA2019]

★ *Works we contributed to*

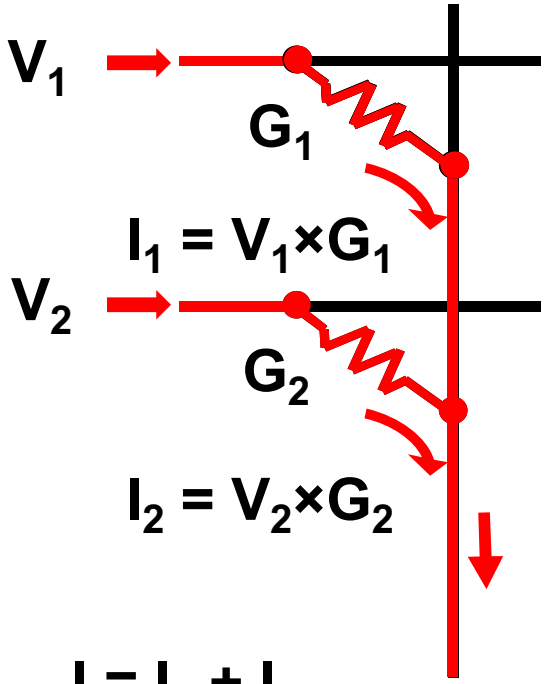
Compute In Memory (CiM) Accelerators





Compute In Memory

Activation is input voltage (V_i)
Weight is resistor conductance (G_i)



Psum is output current

$$I = I_1 + I_2 = V_1 \times G_1 + V_2 \times G_2$$

Image Source: [**Shafiee**, *ISCA* 2016]

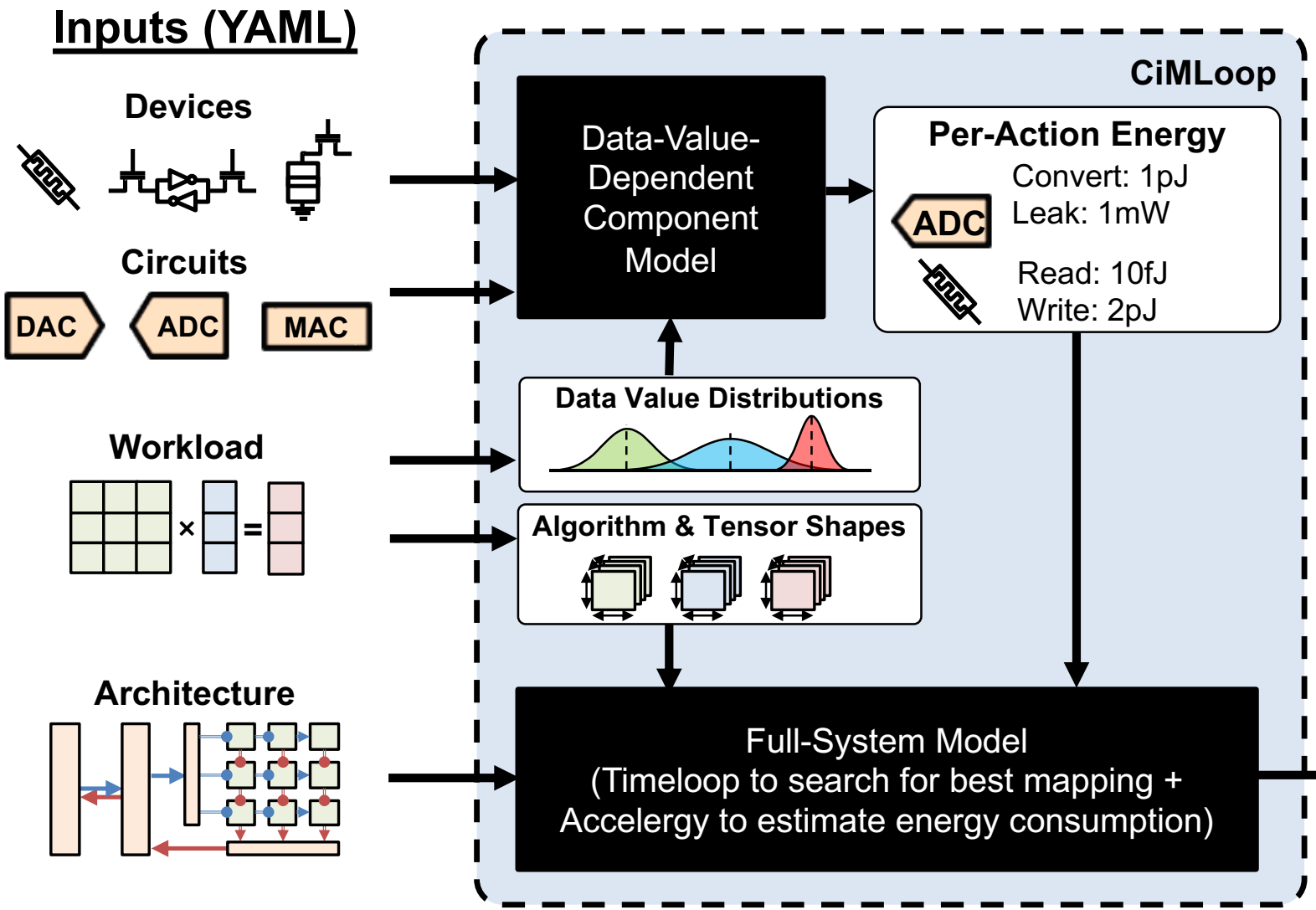
- Reduce data movement by **moving compute into memory**
- Compute MAC with memory storage element
- **Analog Compute**
  - Activations, weights and/or partial sums are encoded with analog voltage, current, or resistance
  - Increased sensitivity to circuit non-idealities
  - A/D and D/A circuits to interface with digital domain
- Leverage **emerging memory device technology**

CiM Research Spans Full Stack

- **Devices:** The components forming each memory cell (e.g., SRAM, DRAM, eNVM)
- **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 processed, 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**

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



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

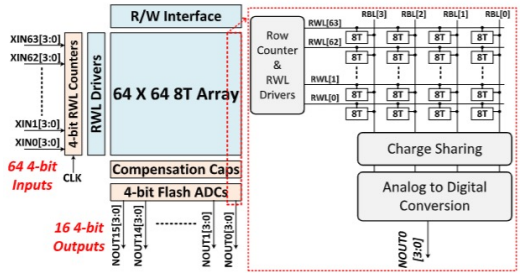
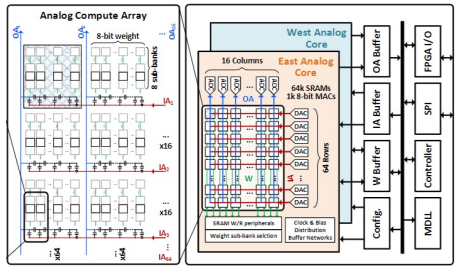
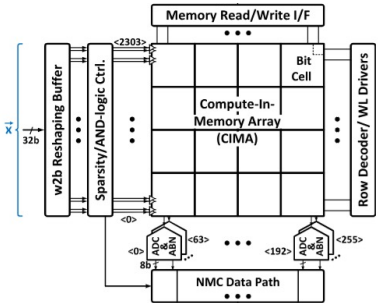
Throughput

[**Andrulis**, *ISPASS* 2024] ***Best Paper Award***

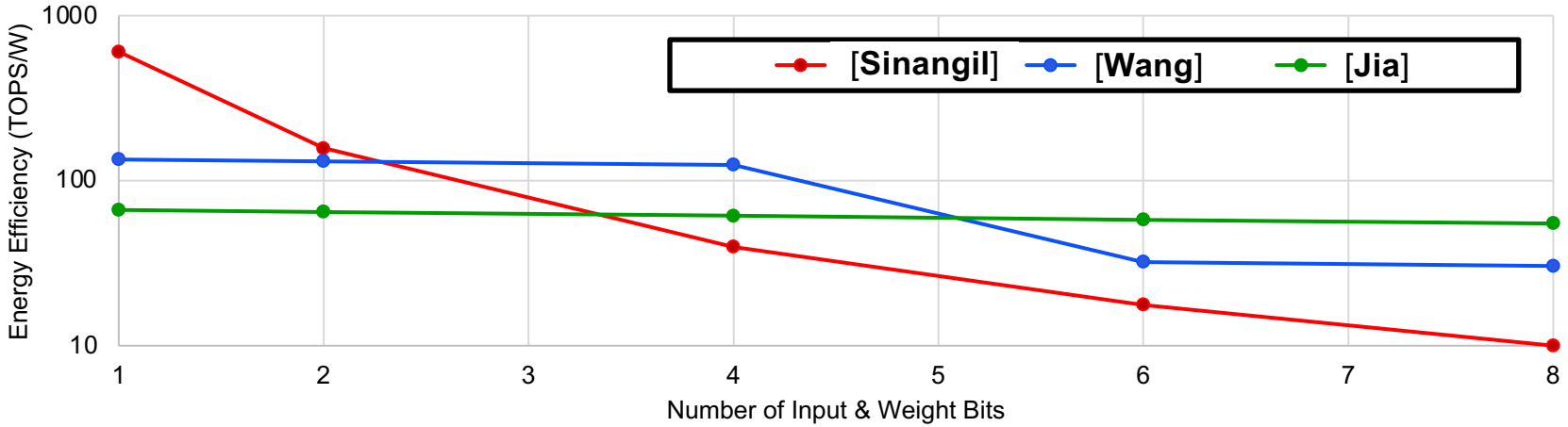
CiMLoop: A Flexible, Accurate, and Fast CiM Modeling Tool

- **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 are within 8% of values reported for measured designs***
- **Speed**
  - A fast statistical model to enable 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

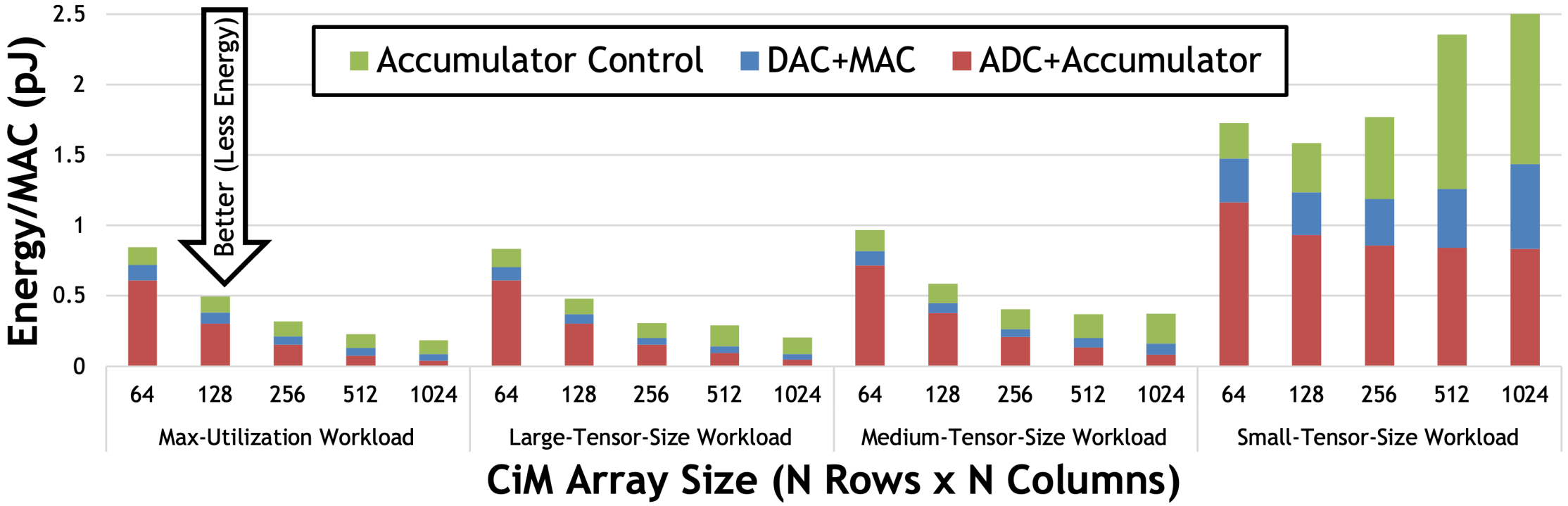
Macro			
	[**Sinangil**, *JSSC* 2021]	[**Wang**, *VLSI* 2022]	[**Jia**, *JSSC* 2020]
Technology Node	7nm	22nm	65nm
ADC Type	4b Flash	8b SAR	8b SAR
Memory Device	6T SRAM	8T SRAM + Capacitor	6T SRAM

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



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

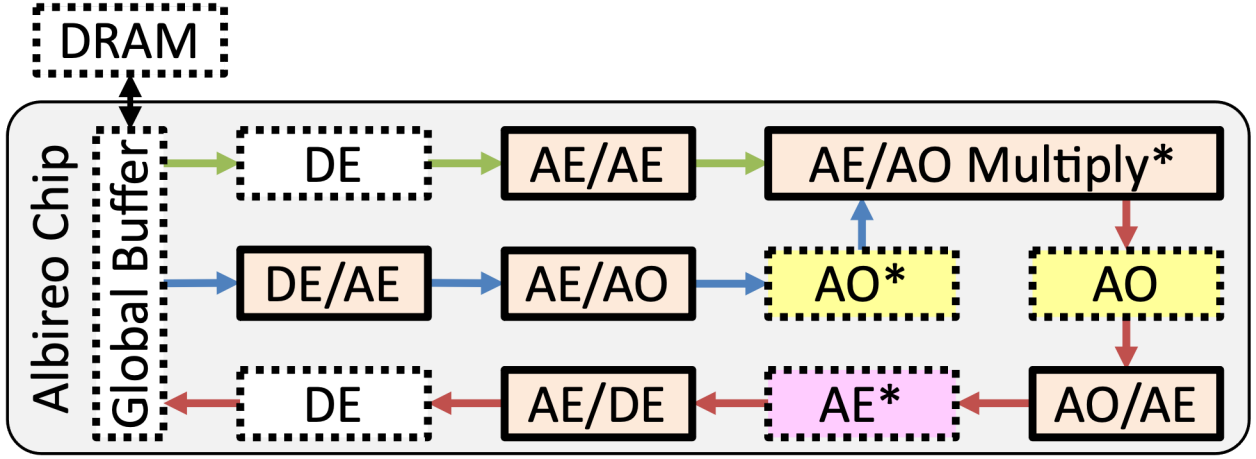
Example: Design Space Exploration

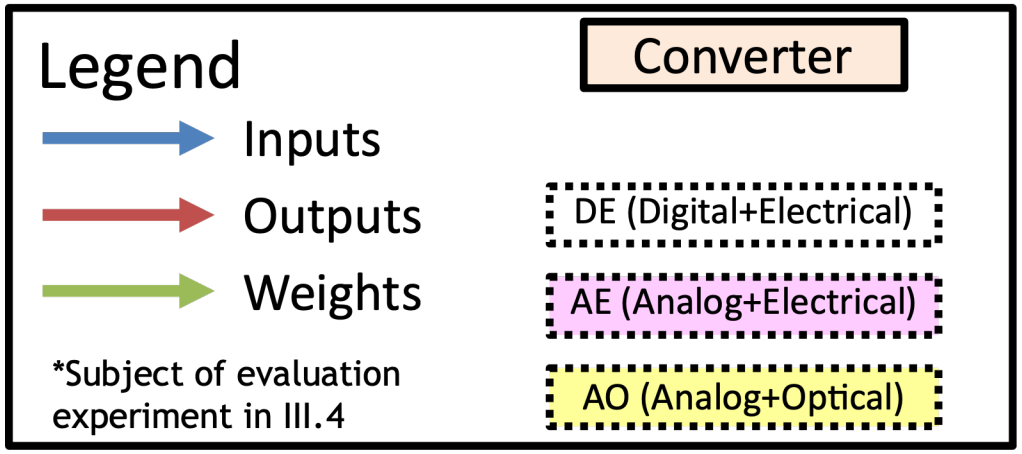


Explore array size (architecture) and DNN shapes (workload)

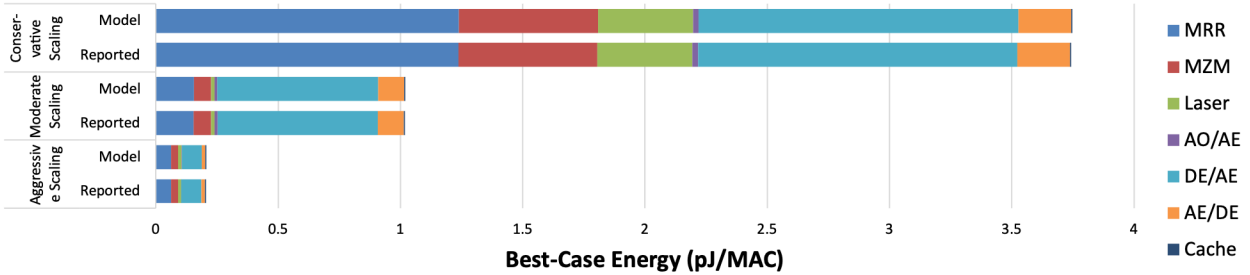
[**Andrulis**, *ISPASS* 2024]

CiMLoop for Photonic Accelerator Modeling

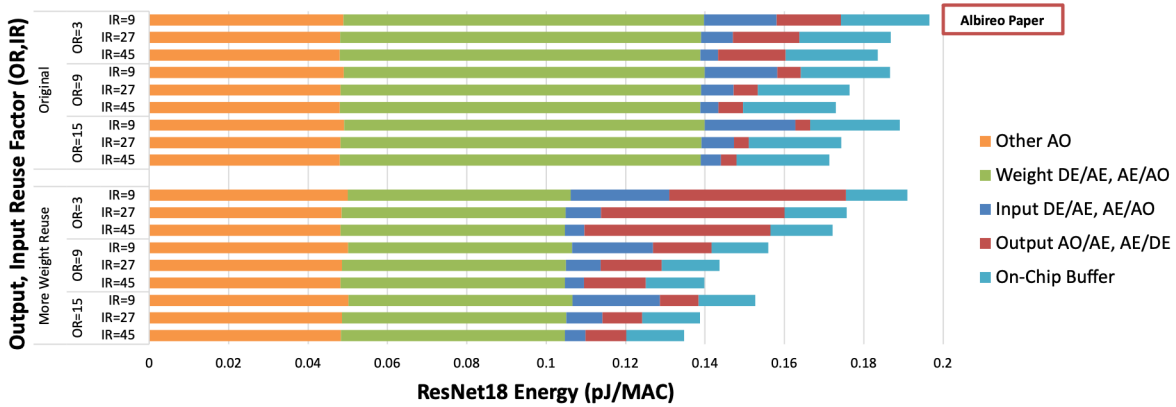




Validation



Design Space Exploration

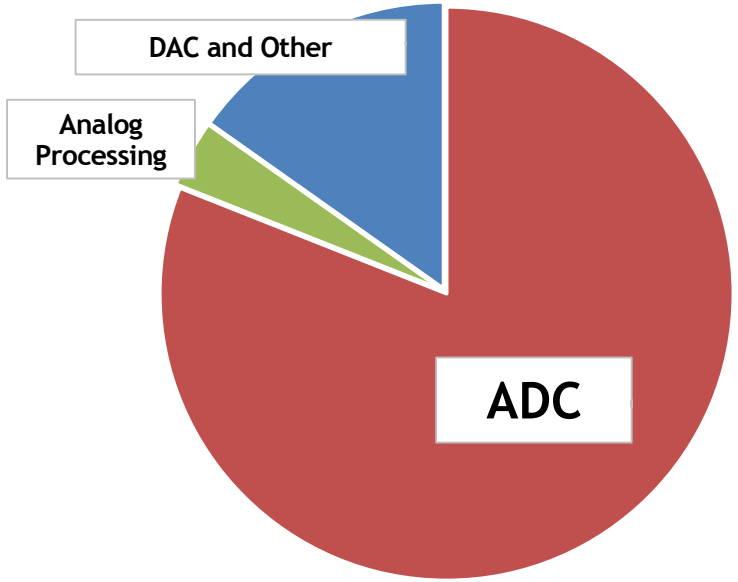


Many similarities in design of CiM and Photonic accelerators → Can model with **CiMLoop!**

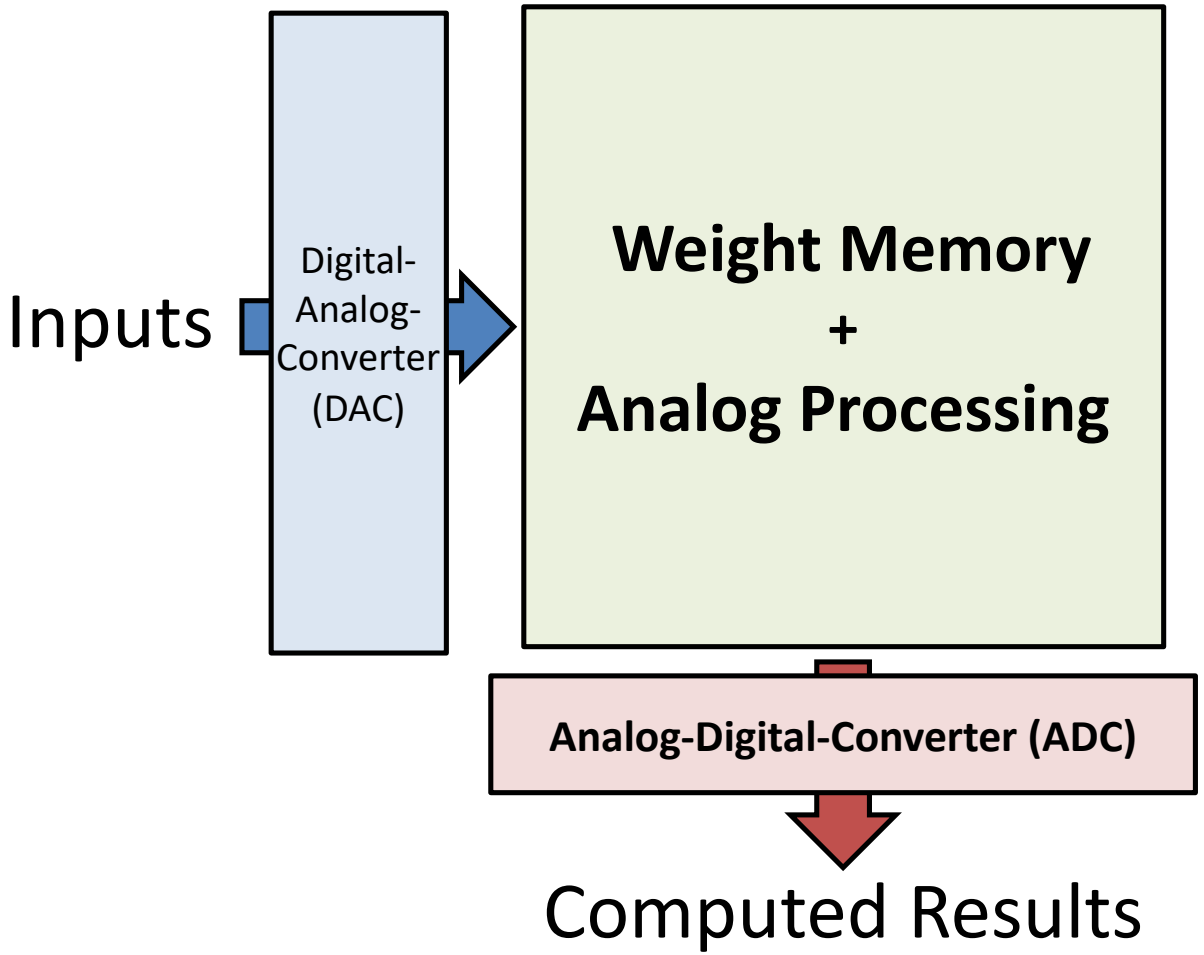
[**Andrulis**, *ISPASS* 2024]

Compute In Memory (CIM) Accelerators

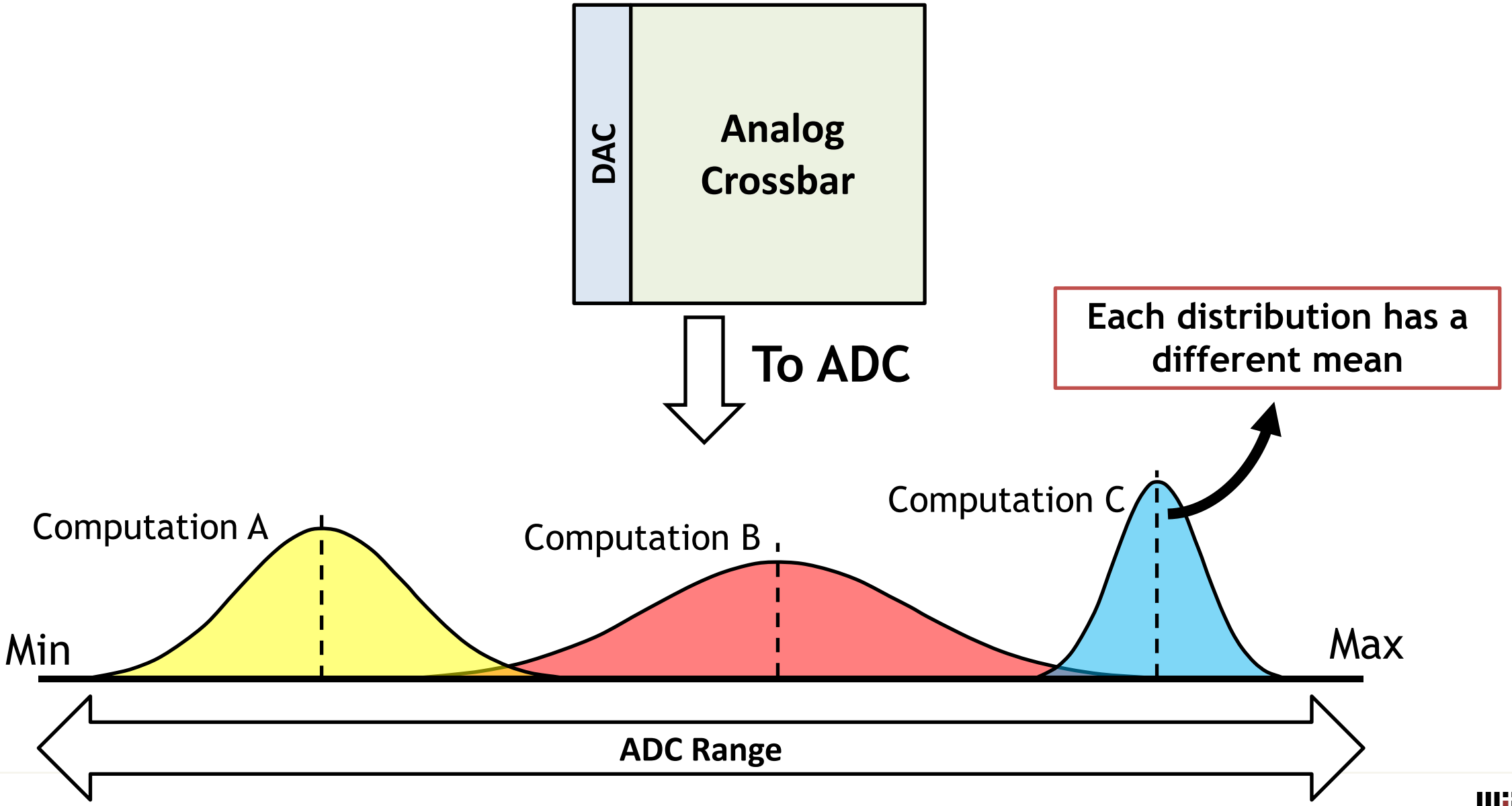
Energy Breakdown



ADC consumes significant energy

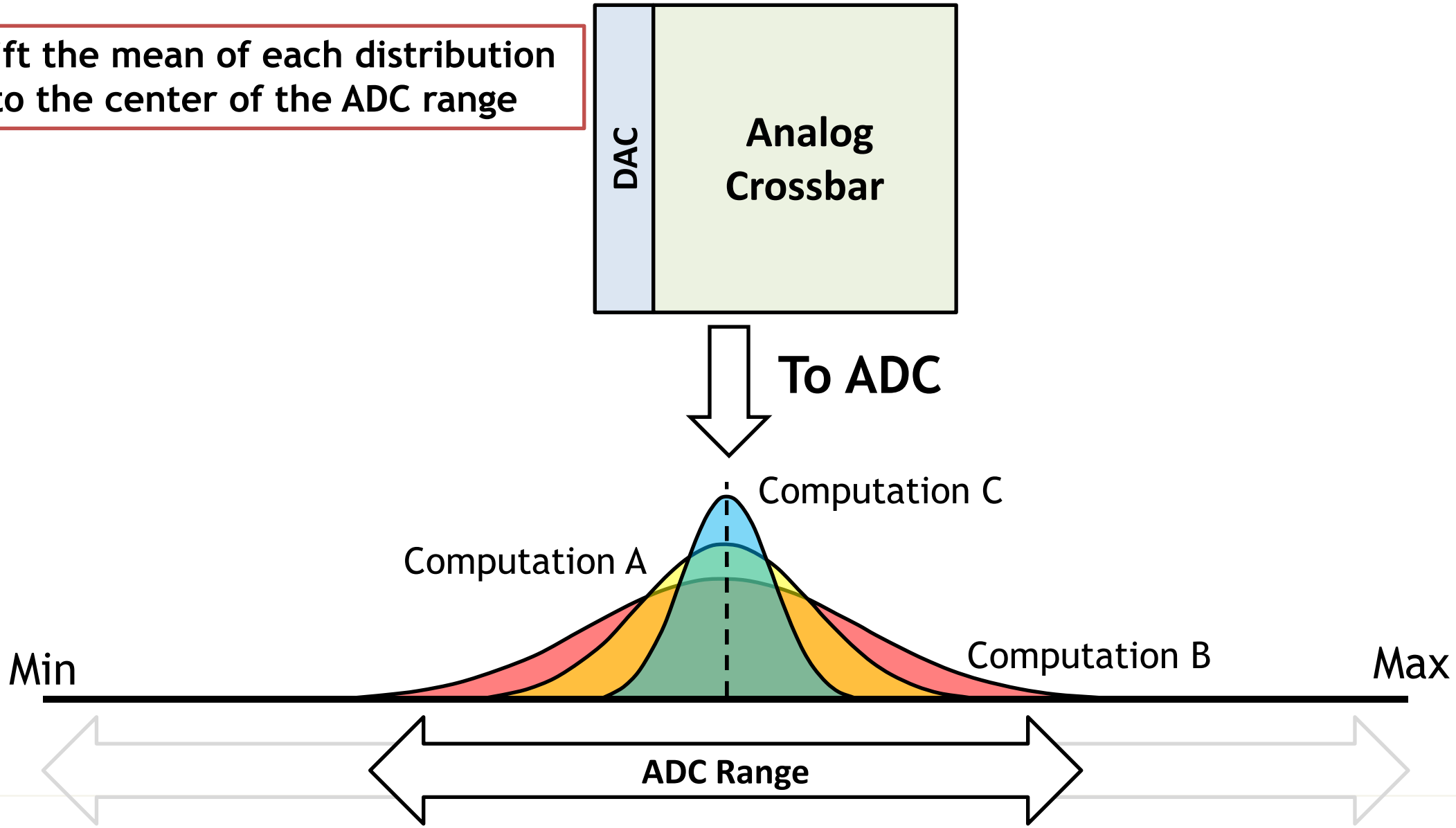


RAELLA: Shifting Distributions to Reduce Input to ADC



RAELLA: Shifting Distributions

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

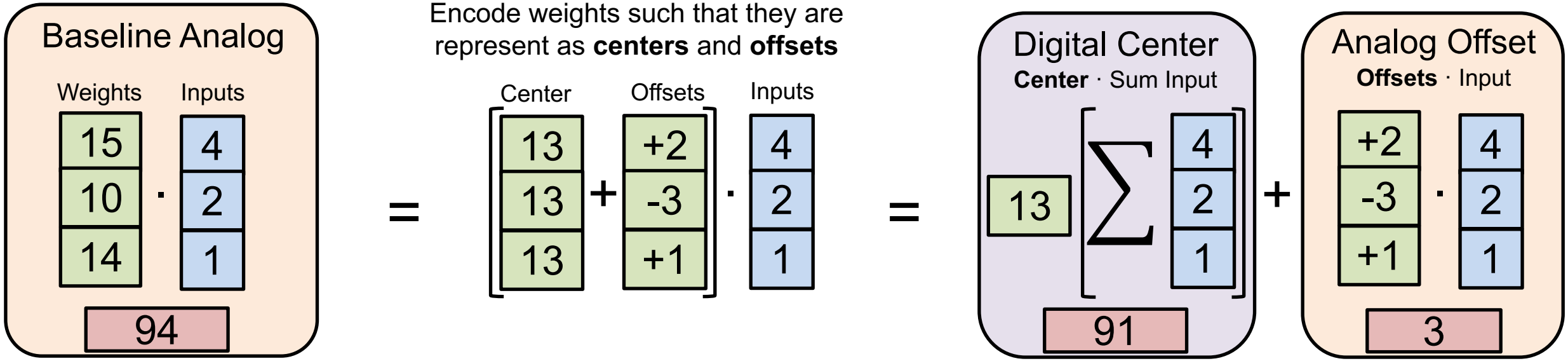


Center + Offset Weight Encoding Zero-Average Analog Results

Partition computation

Digital calculates high-resolution **center** operations

Analog calculates parallel **offset** operations

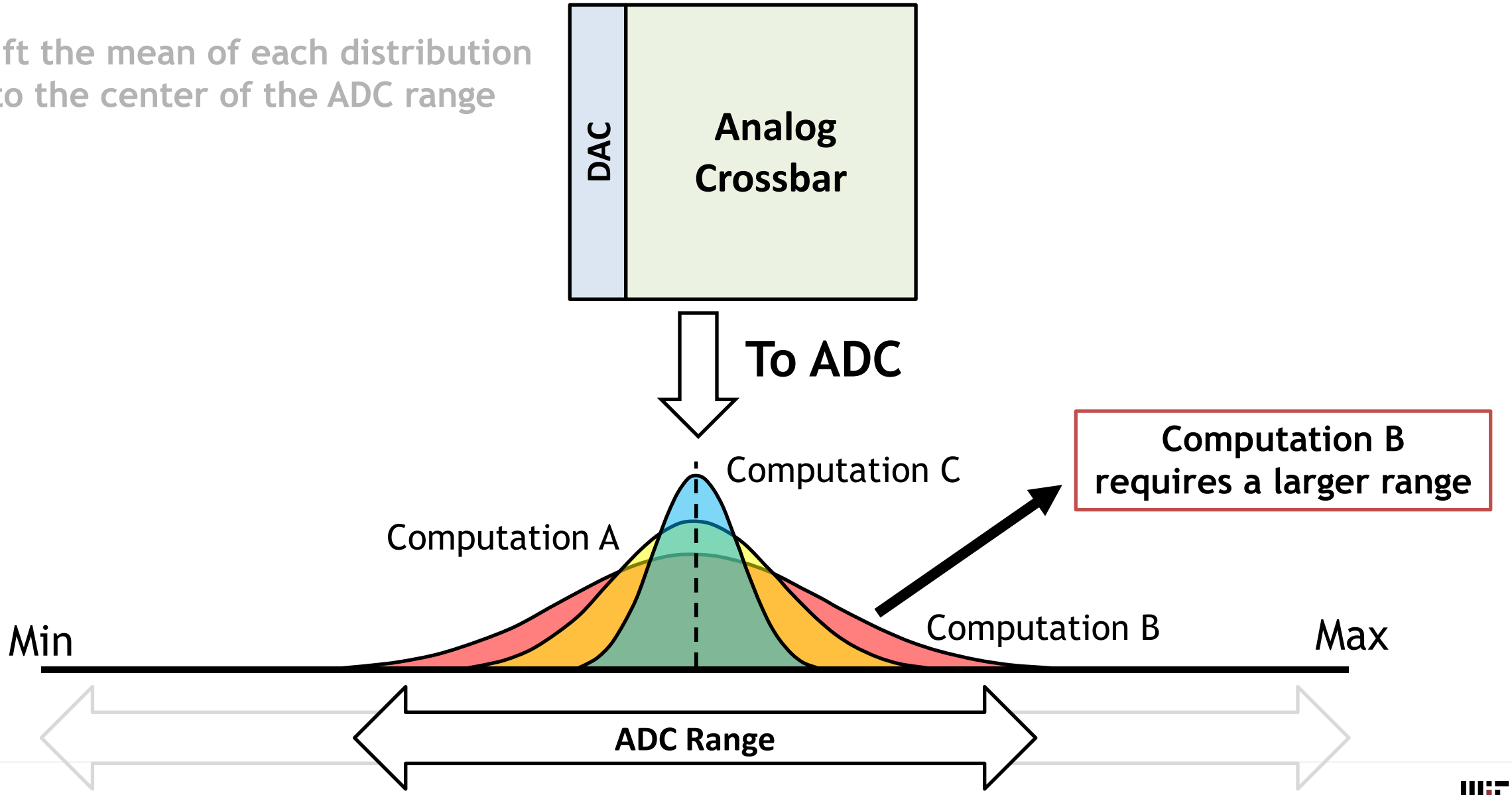


Encoding allows analog input to ADC to be **smaller and closer to zero**

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

RAELLA: Shifting Distributions

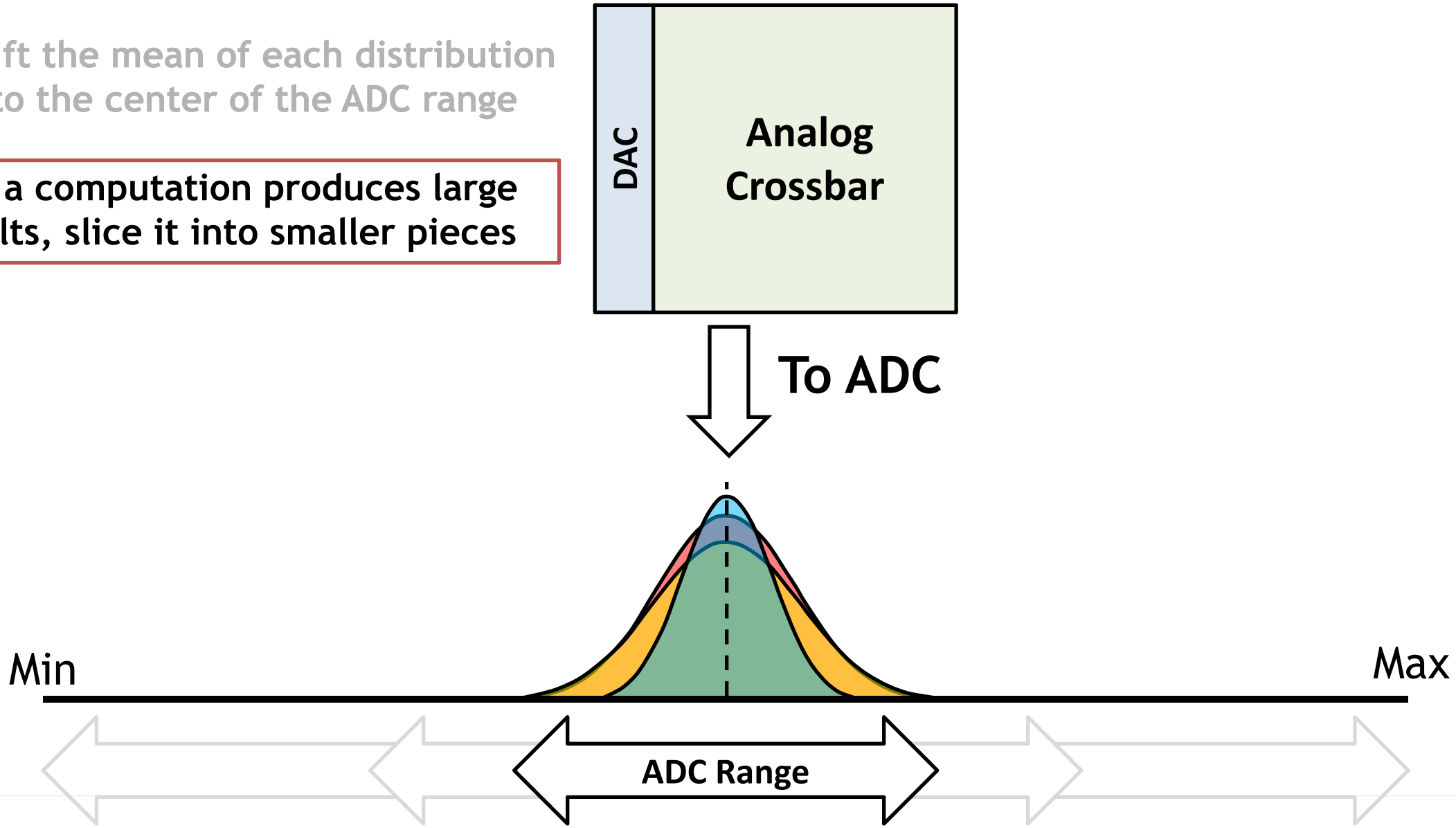
1. Shift the mean of each distribution to the center of the ADC range



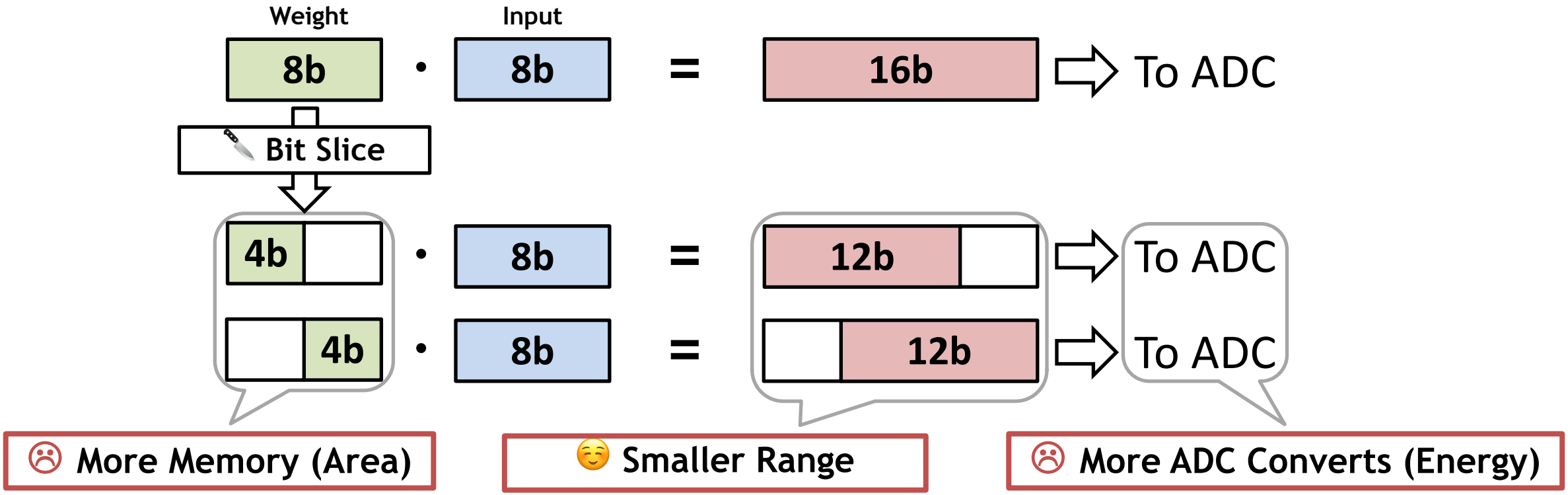
RAELLA: Shifting Distributions

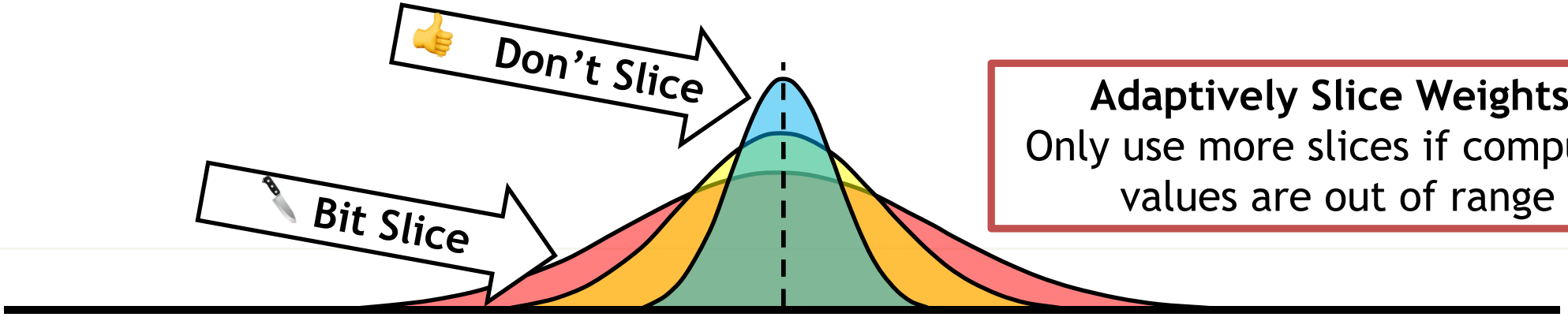
1. Shift the mean of each distribution to the center of the ADC range

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



Adaptive Weight Slicing: Slice Large-Result Computations



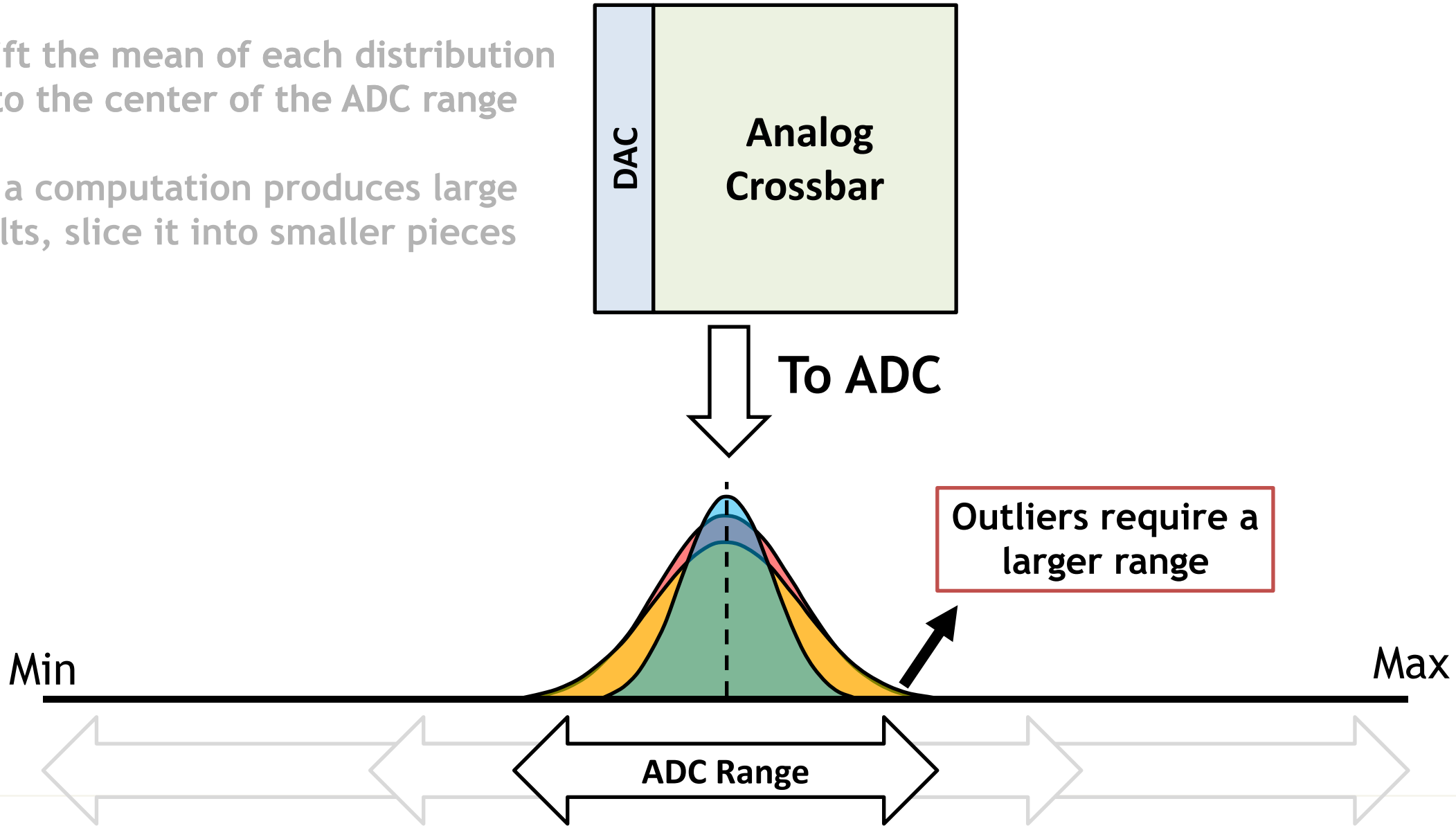


Adaptively Slice Weights:
Only use more slices if computed values are out of range

RAELLA: Shifting 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

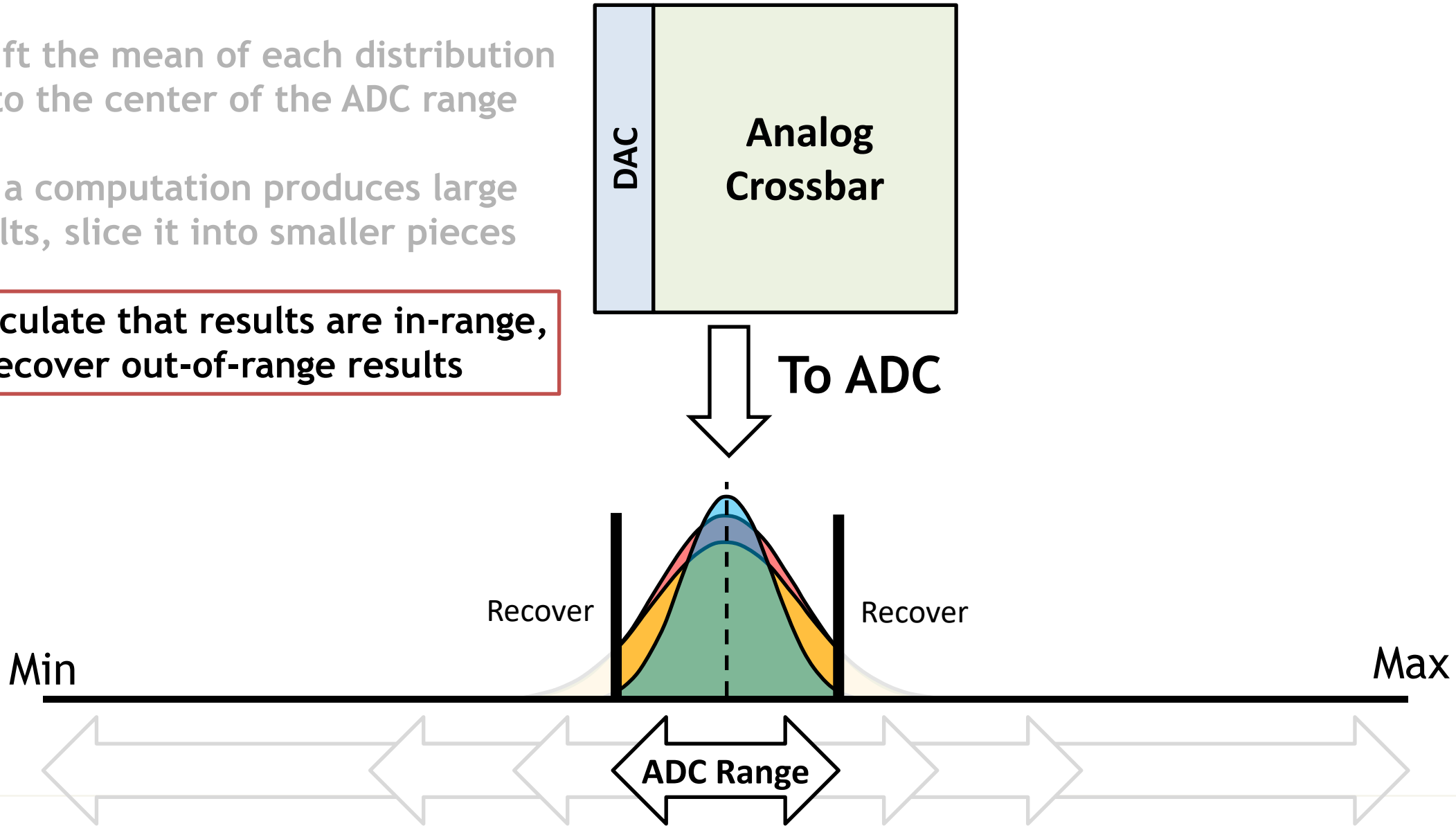


RAELLA: Shifting Distributions

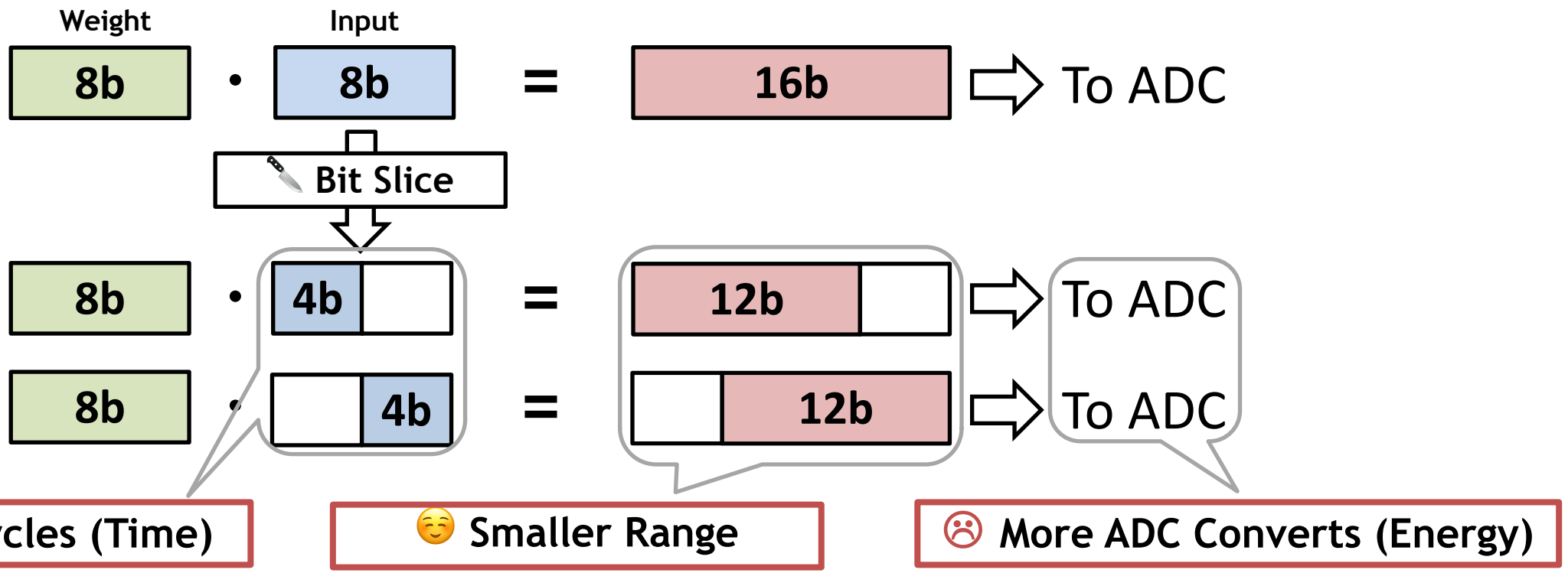
1. Shift the mean of each distribution to the center of the ADC range

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

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

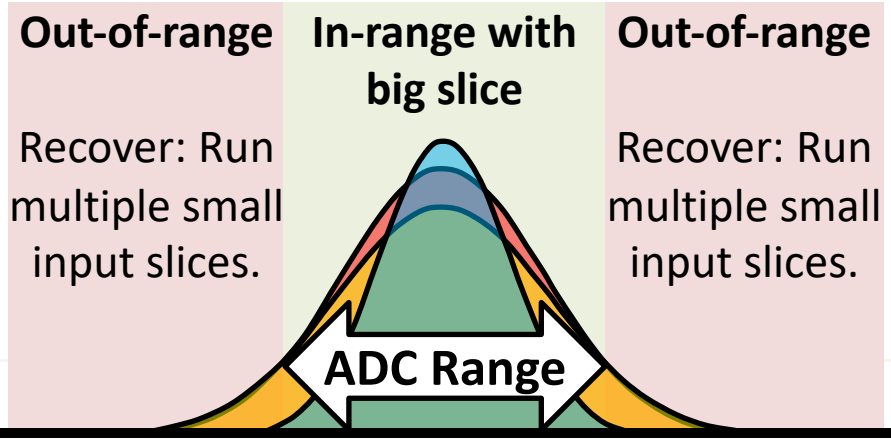


Dynamic Input Slicing: Try Again with Smaller Slices



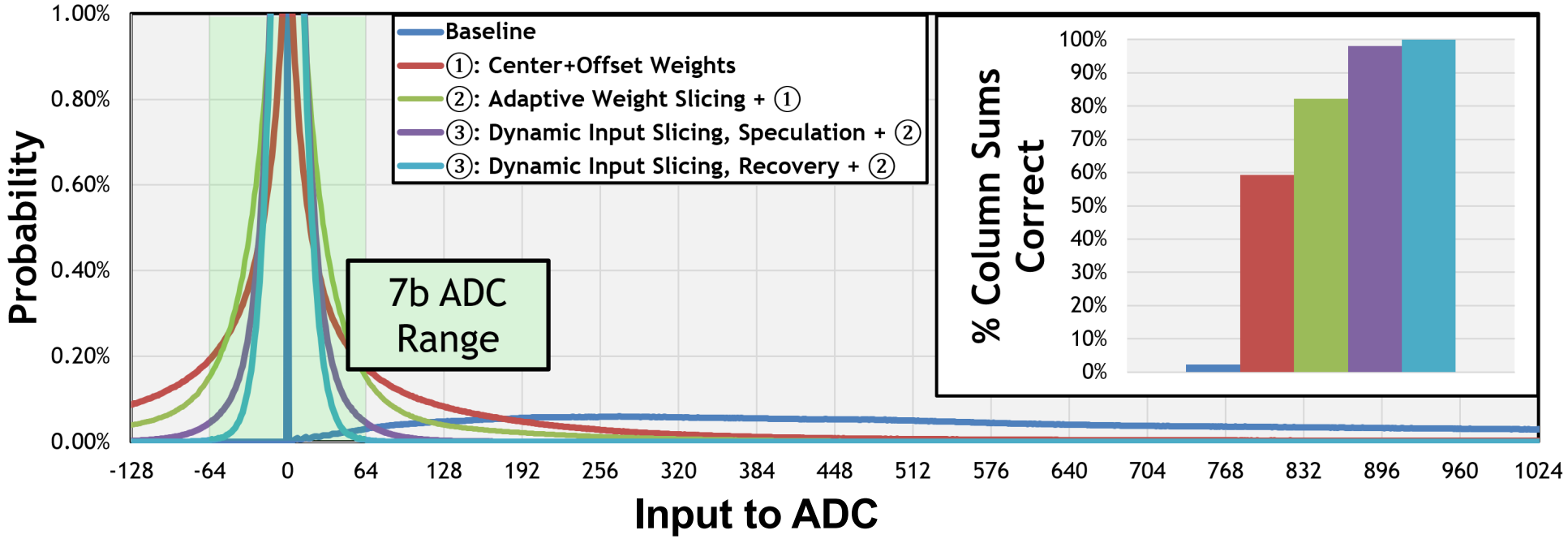
1. Speculate with big input slice

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



RAELLA: Reshape Distributions of Input to ADC

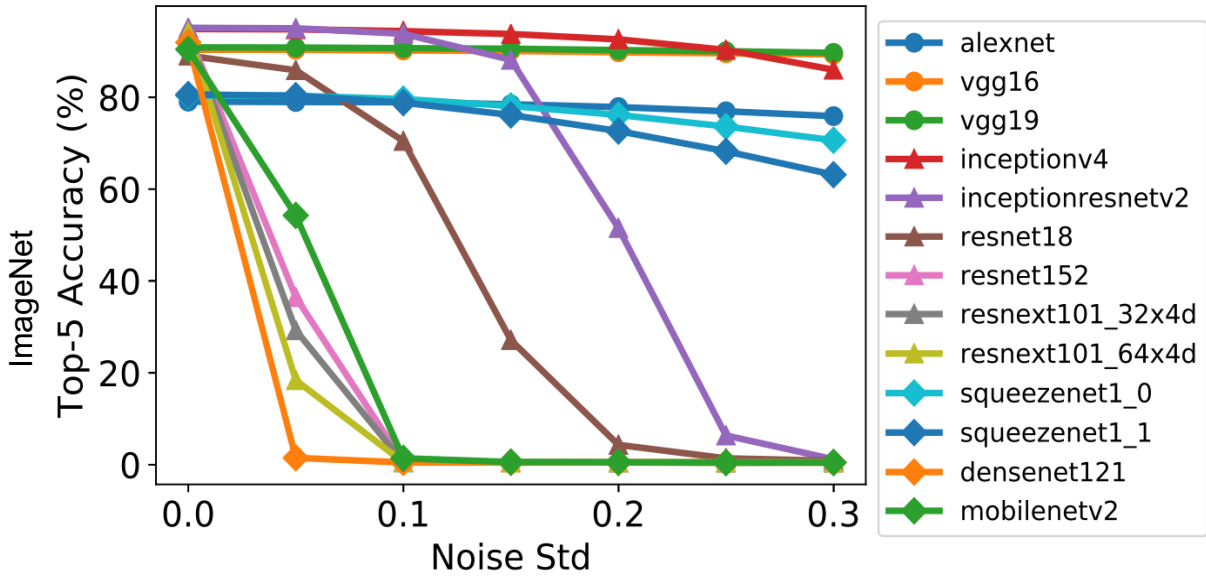
- Makes analog operations produce low-resolution results
  - 1024x reduction of input to ADC
- Enables more compute per ADC convert while using lower-resolution ADCs
  - Improves energy efficiency by 3.9x and throughput by 1.8x compared to iso-area ISAAC
- Maintains DNN accuracy without changing DNN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

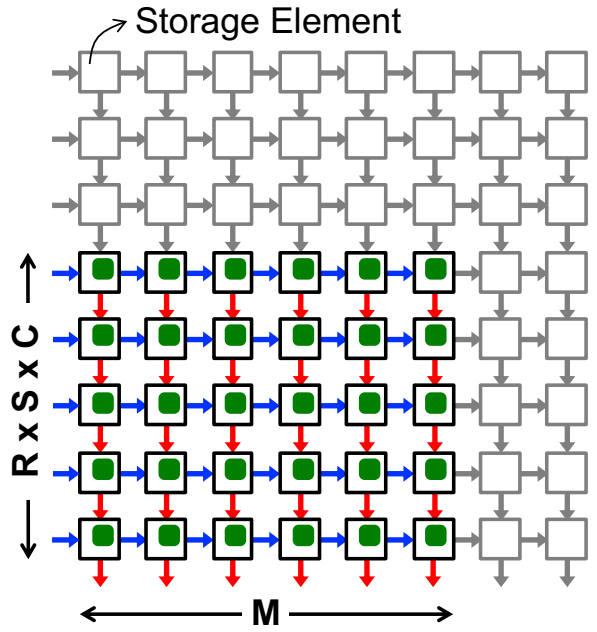


[**Andrulis**, *ISCA* 2023]

Designing DNNs 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





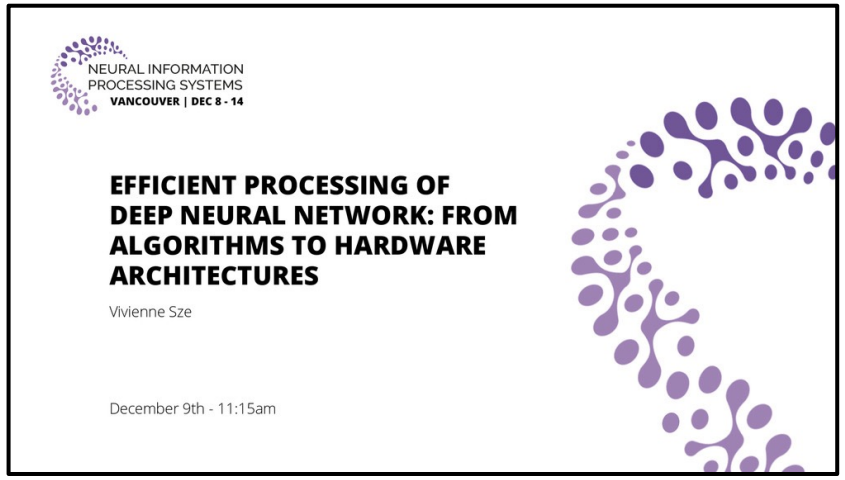
[**Yang**, *IEDM* 2019]

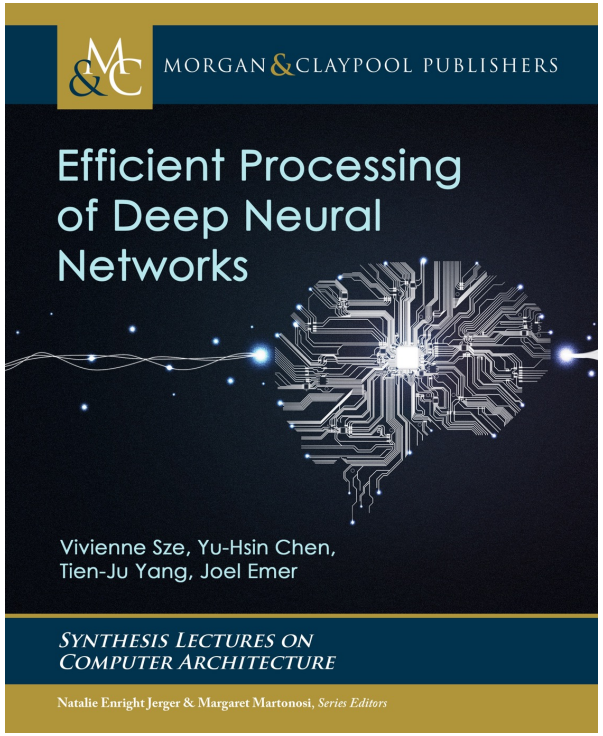
Next Steps

- Enhance our prior work that allows for the **whole system analysis at the architectural level** combined with **characterizations of each technological component of the system**, to project the energy, speed, and cost of a proposed design with a new technology (including **emerging devices and improved packaging**).

- Analysis can comprehend not only custom devices, but also a range of tensor calculations for a range of applications from different domains, including deep learning, graph analytics and databases.

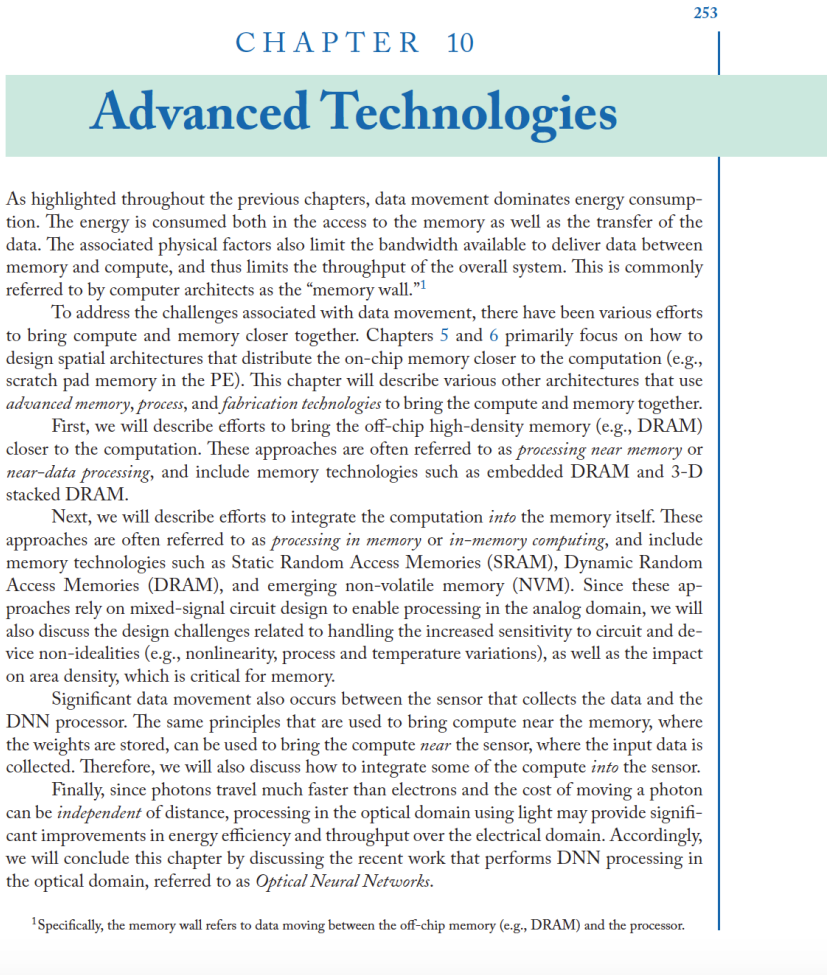
Resources on Efficient Processing of DNNs





http://eyeriss.mit.edu/tutorial.html

Book Chapter on In-Memory Computing



253

CHAPTER 10

Advanced Technologies

As highlighted throughout the previous chapters, data movement dominates energy consumption. The energy is consumed both in the access to the memory as well as the transfer of the data. The associated physical factors also limit the bandwidth available to deliver data between memory and compute, and thus limits the throughput of the overall system. This is commonly referred to by computer architects as the "memory wall."[1]

To address the challenges associated with data movement, there have been various efforts to bring compute and memory closer together. Chapters 5 and 6 primarily focus on how to design spatial architectures that distribute the on-chip memory closer to the computation (e.g., scratch pad memory in the PE). This chapter will describe various other architectures that use *advanced memory*, *process*, and *fabrication technologies* to bring the compute and memory together.

First, we will describe efforts to bring the off-chip high-density memory (e.g., DRAM) closer to the computation. These approaches are often referred to as *processing near memory* or *near-data processing*, and include memory technologies such as embedded DRAM and 3-D stacked DRAM.

Next, we will describe efforts to integrate the computation *into* the memory itself. These approaches are often referred to as *processing in memory* or *in-memory computing*, and include memory technologies such as Static Random Access Memories (SRAM), Dynamic Random Access Memories (DRAM), and emerging non-volatile memory (NVM). Since these approaches rely on mixed-signal circuit design to enable processing in the analog domain, we will also discuss the design challenges related to handling the increased sensitivity to circuit and device non-idealities (e.g., nonlinearity, process and temperature variations), as well as the impact on area density, which is critical for memory.

Significant data movement also occurs between the sensor that collects the data and the DNN processor. The same principles that are used to bring compute near the memory, where the weights are stored, can be used to bring the compute *near* the sensor, where the input data is collected. Therefore, we will also discuss how to integrate some of the compute *into* the sensor.

Finally, since photons travel much faster than electrons and the cost of moving a photon can be *independent* of distance, processing in the optical domain using light may provide significant improvements in energy efficiency and throughput over the electrical domain. Accordingly, we will conclude this chapter by discussing the recent work that performs DNN processing in the optical domain, referred to as *Optical Neural Networks*.

[1] Specifically, the memory wall refers to data moving between the off-chip memory (e.g., DRAM) and the processor.

Many Design Considerations for In-Memory Computing

- Number of Storage Elements per Weight
- Array Size
- Number of Rows Activated in Parallel
- Number of Columns Activated in Parallel
- Time to Deliver Input
- Time to Compute MAC

Tradeoffs between energy efficiency, throughput, area density, and accuracy, which *reduce the achievable gains over conventional architectures*

Available on DNN tutorial website
http://eyeriss.mit.edu/tutorial.html

Related Work by PIs (http://emze.mit.edu):

- **Modeling**
  - 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 ***Best Paper Award***
  - Y. N. Wu, P. Tsai, A. Parashar, V. Sze, J. Emer, "Sparseloop: An Analytical Approach to Sparse Tensor Accelerator Modeling," ACM/IEEE International Symposium on Microarchitecture (MICRO), October 2022. ***Distinguished Artifact Award***
  - 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
  - Y. N. Wu, J. S. Emer, V. Sze, "Accelergy: An Architecture-Level Energy Estimation Methodology for Accelerator Designs," International Conference on Computer Aided Design (ICCAD), November 2019.
- **Spatial accelerator design**
  - 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
  - Y.-H. Chen, T. Krishna, J. Emer, V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," IEEE Journal of Solid-State Circuits (JSSC), ISSCC Special Issue, Vol. 52, No. 1, pp. 127-138, January 2017. ***Top 5 most cited JSSC paper of all time***
  - Y.-H. Chen, J. Emer, V. Sze, "Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks," International Symposium on Computer Architecture (ISCA), pp. 367-379, June 2016. ***Selected for IEEE Micro’s Top Picks special issue on "most significant papers in computer architecture based on novelty and long-term impact" from 2016***