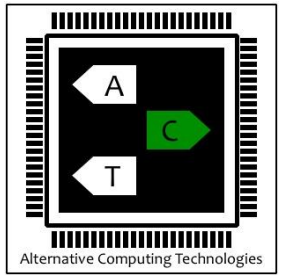
BiHiwe: Mixed-Signal Charge-Domain Acceleration

Soroush Ghodrati

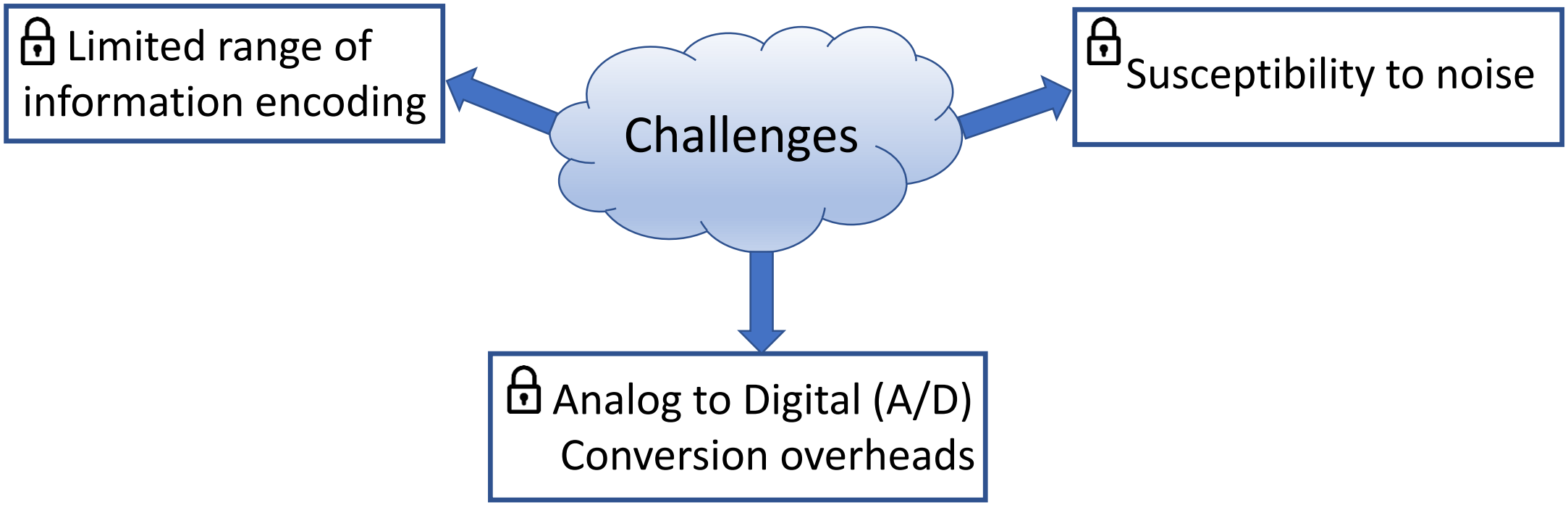
PhD Student at Computer Science and Engineering

Alternative Computing Technologies (ACT) Lab

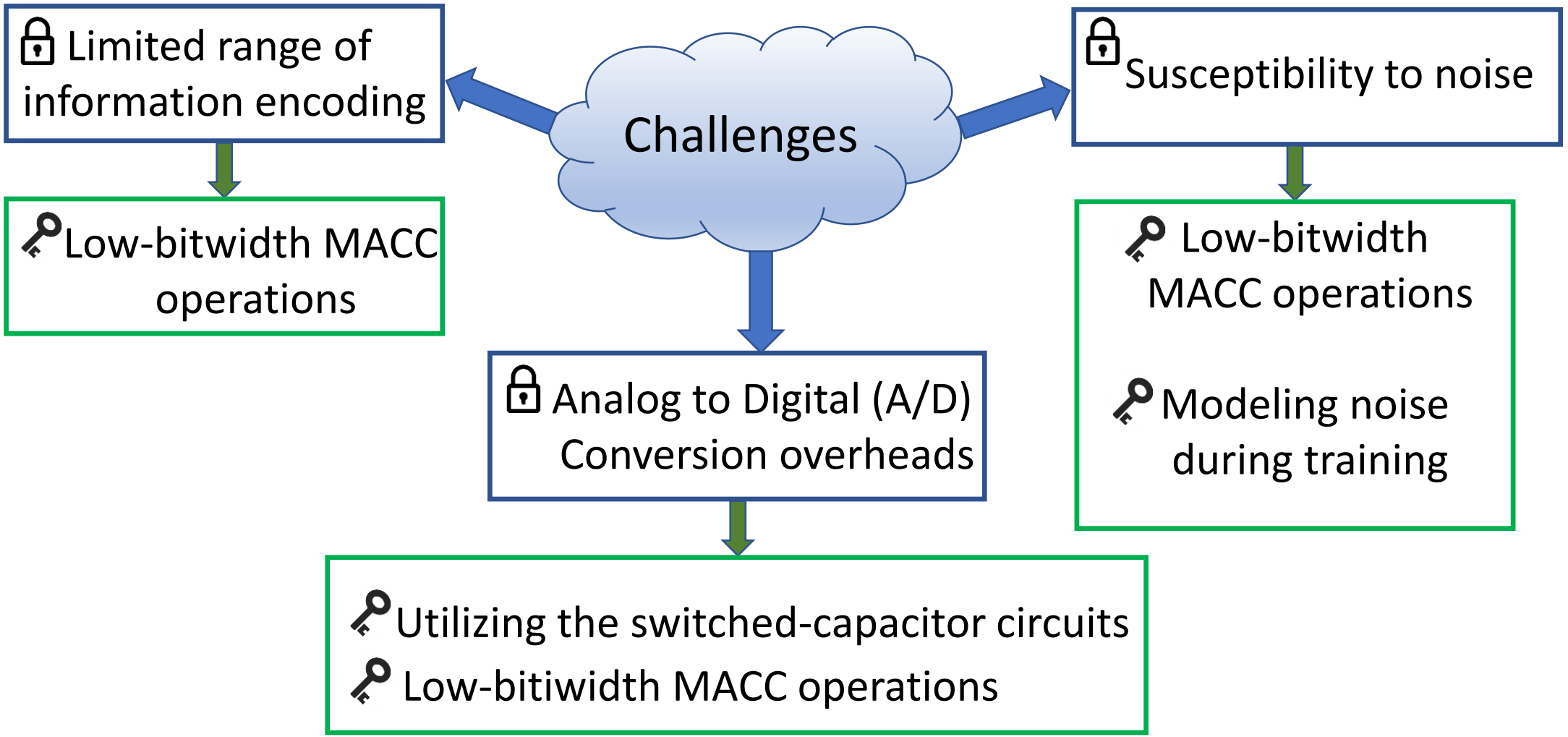
University of California, San Diego



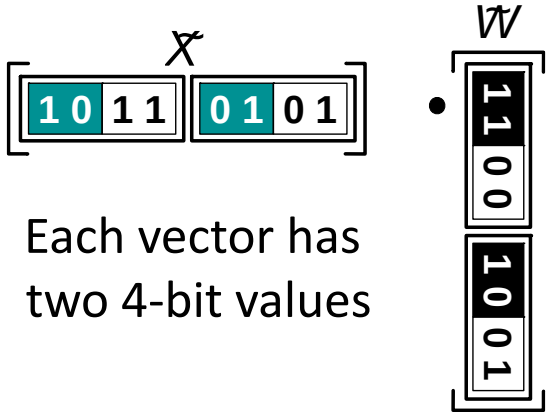
Challenges in Analog Computing



Challenges in Analog Computing



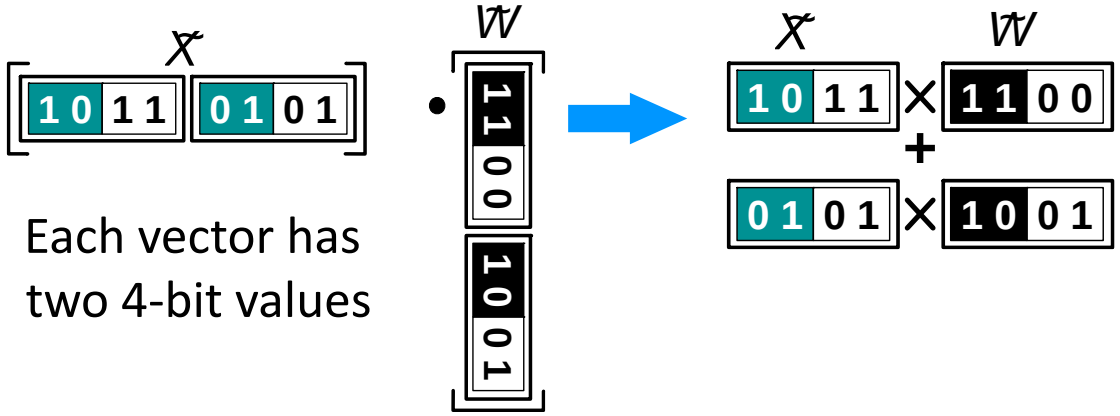
Our Approach:
Wide, Interleaved, and Bit-Partitioned Arithmetic



Each vector has two 4-bit values

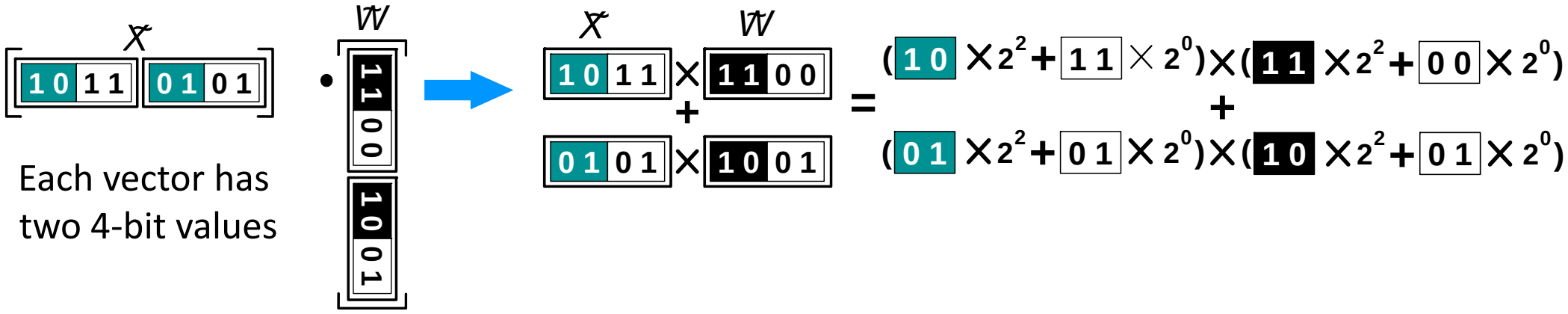
Vector dot-products can be bit-partitioned into groups of low bitwidth operations

Our Approach:
Wide, Interleaved, and Bit-Partitioned Arithmetic



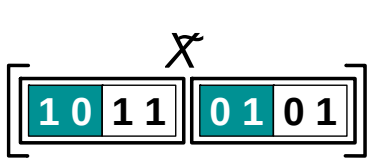
Vector dot-products can be bit-partitioned into groups of low bitwidth operations

Our Approach: Wide, Interleaved, and Bit-Partitioned Arithmetic

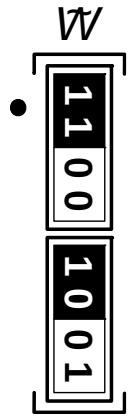


Vector dot-products can be bit-partitioned into groups of low bitwidth operations

Our Approach: Wide, Interleaved, and Bit-Partitioned Arithmetic



Each vector has two 4-bit values



$$\begin{bmatrix}1\,0 & 1\,1\end{bmatrix} \times \begin{bmatrix}1\,1 & 0\,0\end{bmatrix} + \begin{bmatrix}0\,1 & 0\,1\end{bmatrix} \times \begin{bmatrix}1\,0 & 0\,1\end{bmatrix} = (1\,0 \times 2^2 + 1\,1 \times 2^0) \times (1\,1 \times 2^2 + 0\,0 \times 2^0) + (0\,1 \times 2^2 + 0\,1 \times 2^0) \times (1\,0 \times 2^2 + 0\,1 \times 2^0)$$

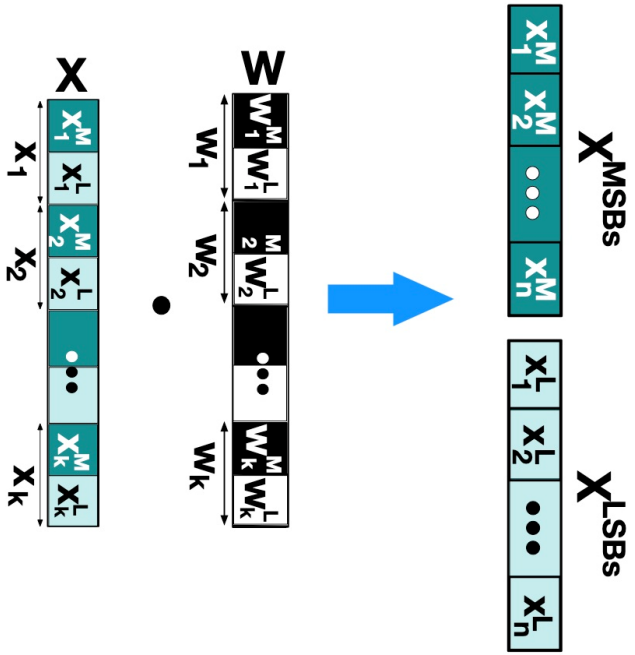
$$= (+ \begin{matrix}1\,0 \times 1\,1 \\ 0\,1 \times 1\,0\end{matrix}) << 2{+}2 \quad (+ \begin{matrix}1\,0 \times 0\,0 \\ 0\,1 \times 0\,1\end{matrix}) << 2{+}0 + (+ \begin{matrix}1\,1 \times 1\,1 \\ 0\,1 \times 1\,0\end{matrix}) << 0{+}2 \quad (+ \begin{matrix}1\,1 \times 0\,0 \\ 0\,1 \times 0\,1\end{matrix}) << 0{+}0$$

$$1\,0\,0\,1\,1\,0\,0\,0\,1$$

Vector dot-products can be bit-partitioned into groups of low bitwidth operations

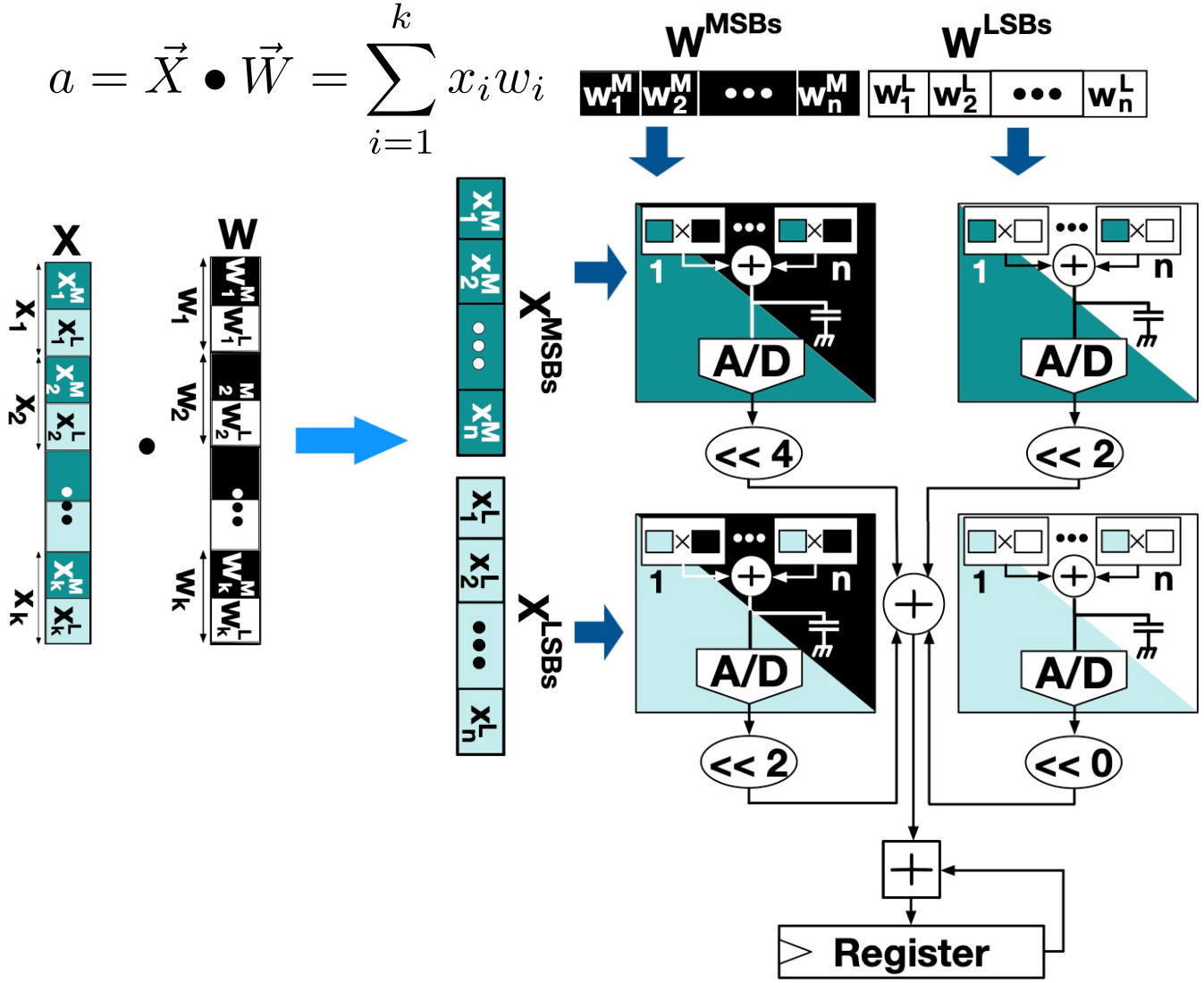
Wide, Interleaved, and Bit-Partitioned Vector Dot-Product

$$a = \vec{X} \bullet \vec{W} = \sum_{i=1}^{k} x_i w_i$$



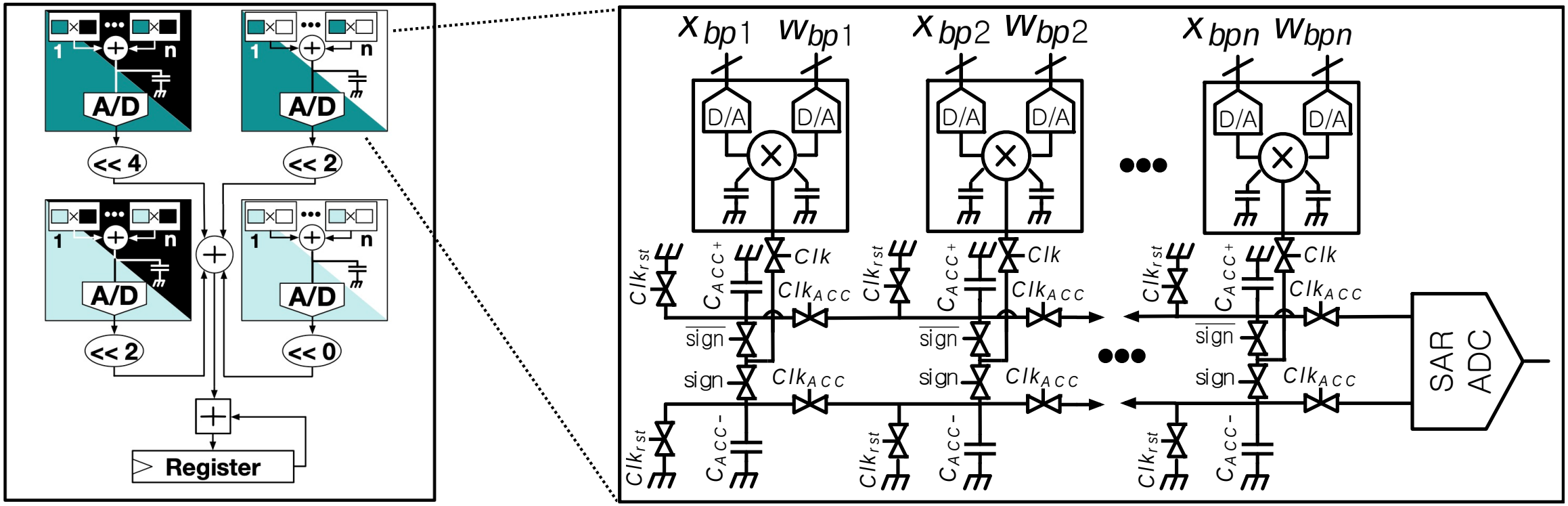
Wide, Interleaved, and Bit-Partitioned Vector Dot-Product

$$a = \vec{X} \bullet \vec{W} = \sum_{i=1}^{k} x_i w_i$$



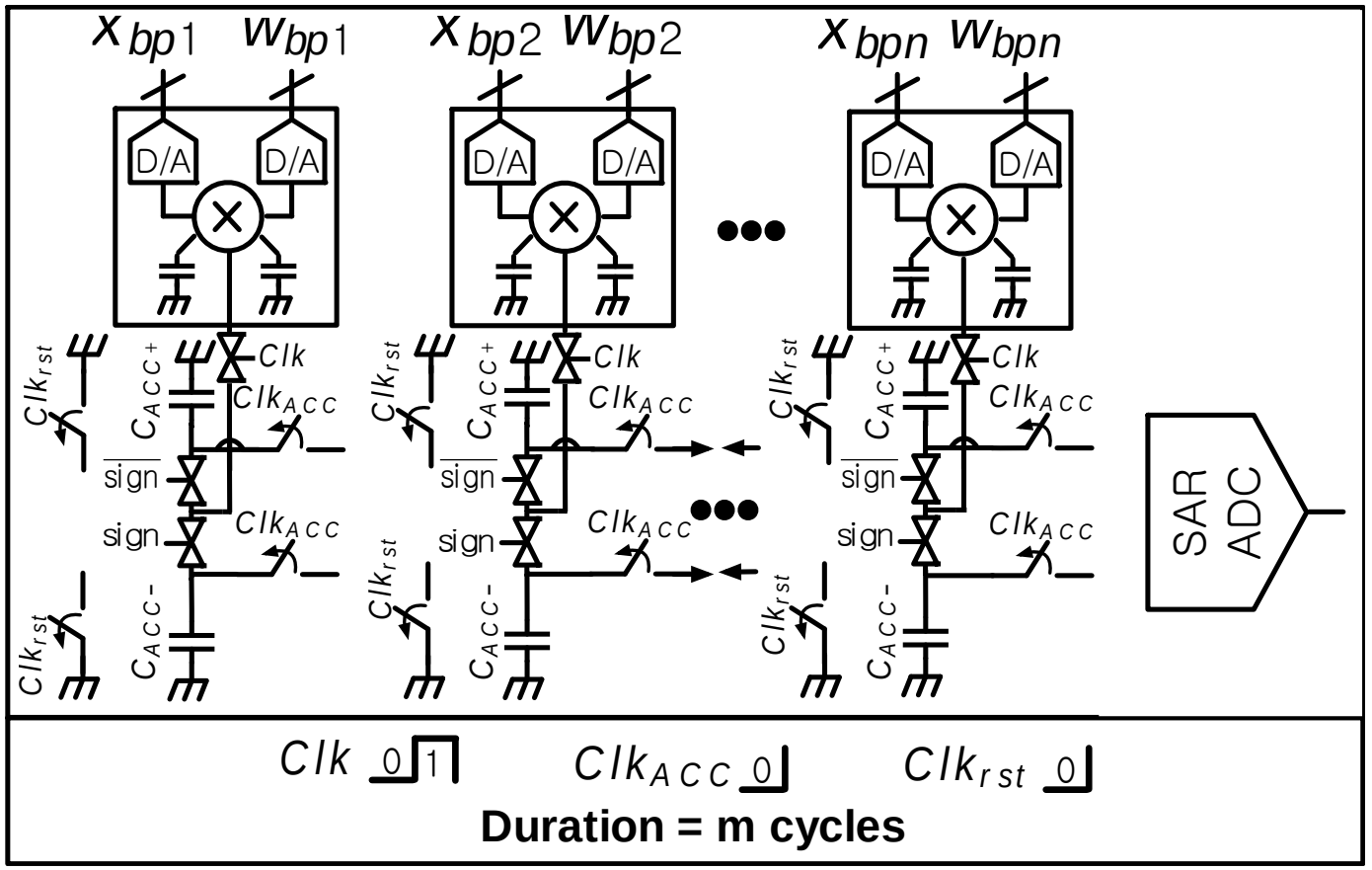
Using low-bitwidth operands provides larger headroom between value encoding in analog domain and reduces the energy/area overhead of A/D and D/A converters

BIHIWE Microarchitecture



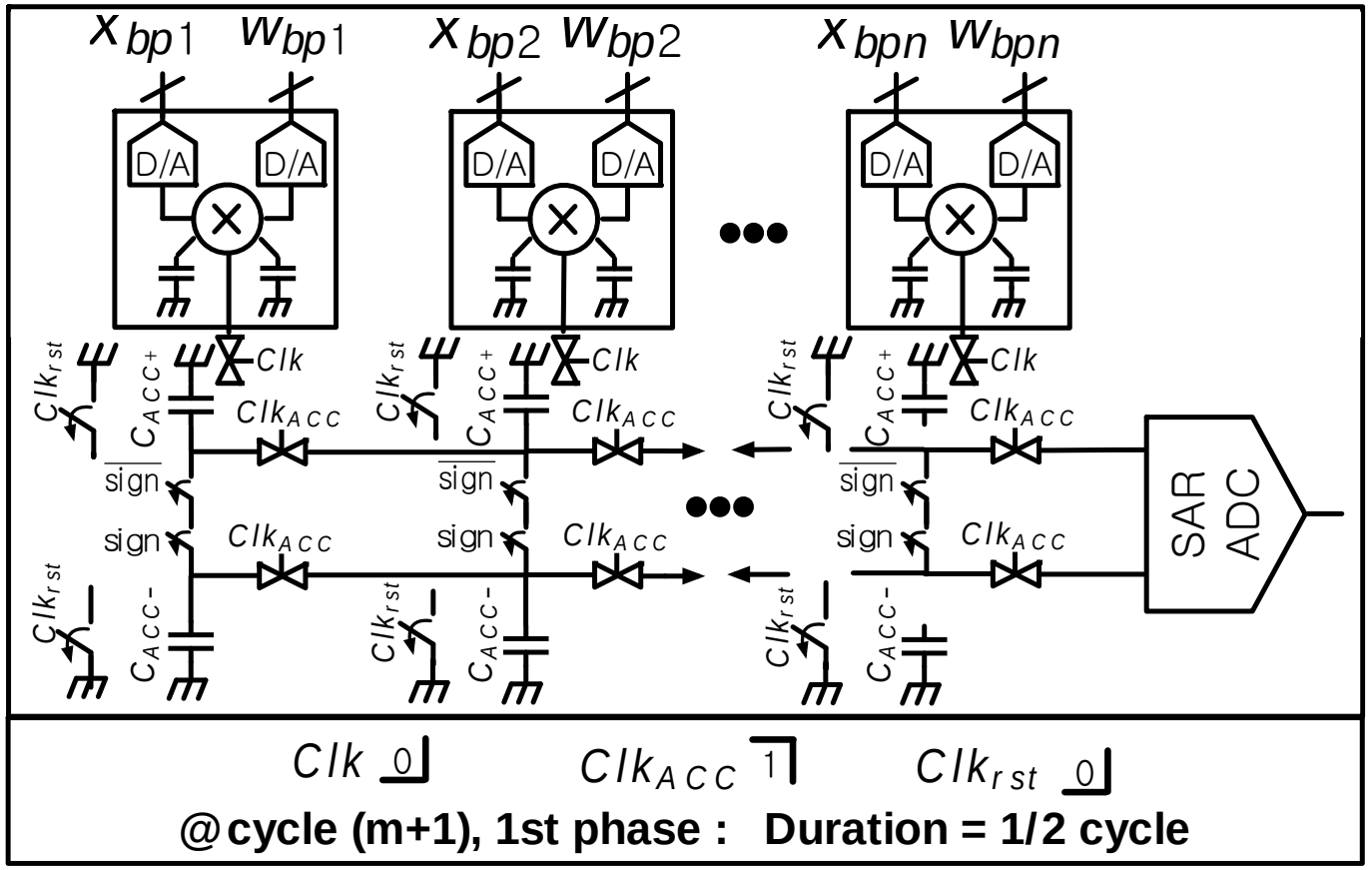
Mixed-Signal Bit-Partitioned MACC Array:
A wide array of low-bitwidth MACC units share single A/D converter

BIHIWE Microarchitecture



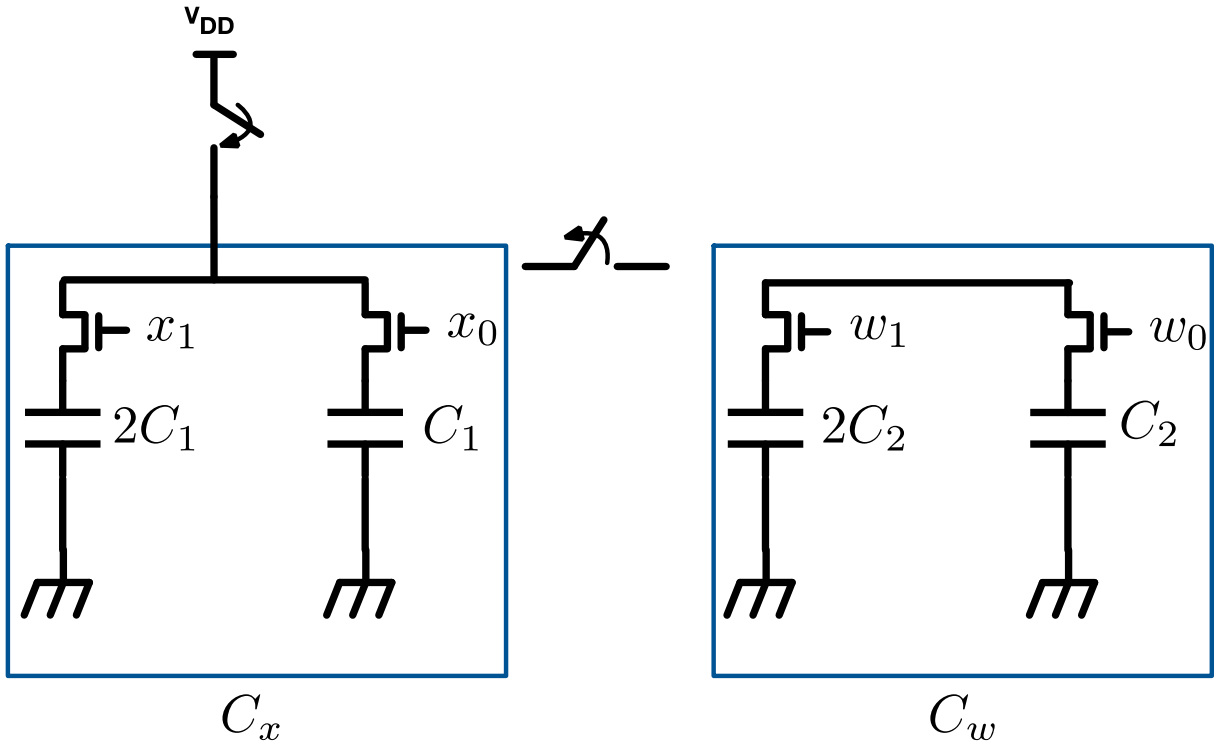
Mixed-Signal Bit-Partitioned MACC Array: MACC Operations and Private Accumulation

BIHIWE Microarchitecture



Mixed-Signal Bit-Partitioned MACC Array:
Accumulating across MACCs and starting A/D conversion

Low-Bitwidth Switched-Capacitor MACC

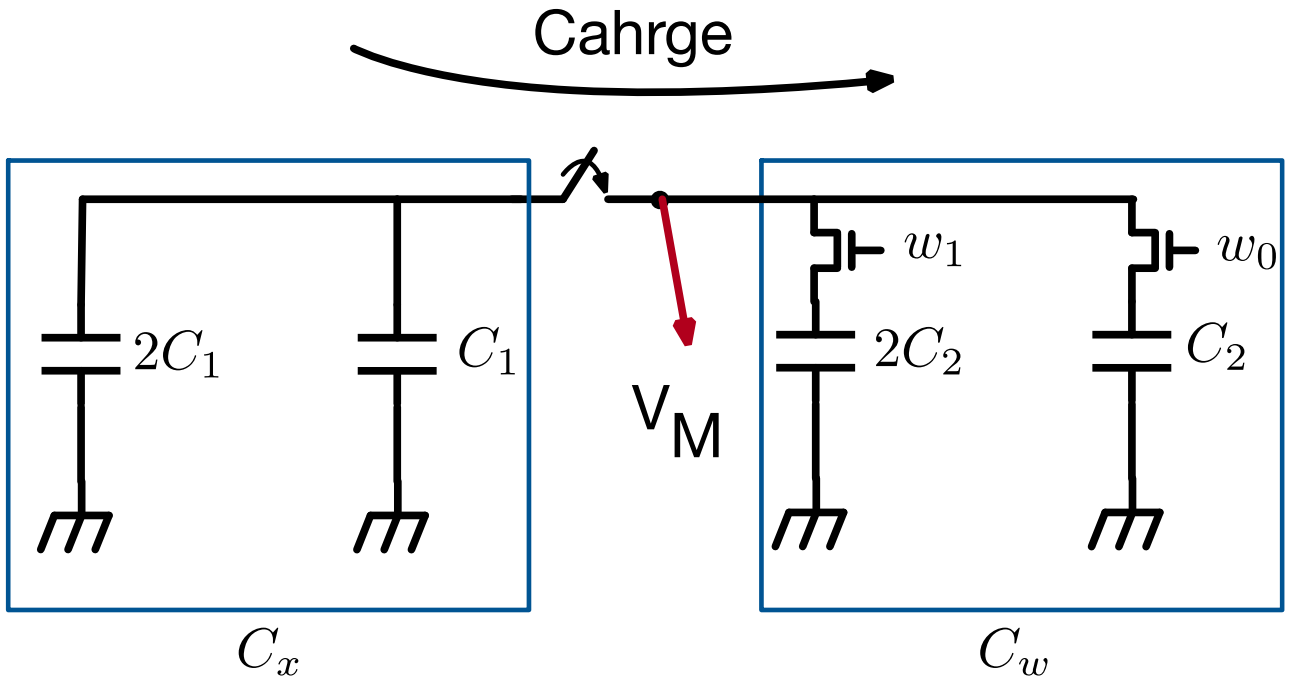


$$Q_x = C_x V_{DD}$$

$$Q_x \propto |X| C_1$$

A charge proportional to the magnitude of X is stored on C_x

Low-Bitwidth Switched-Capacitor MACC



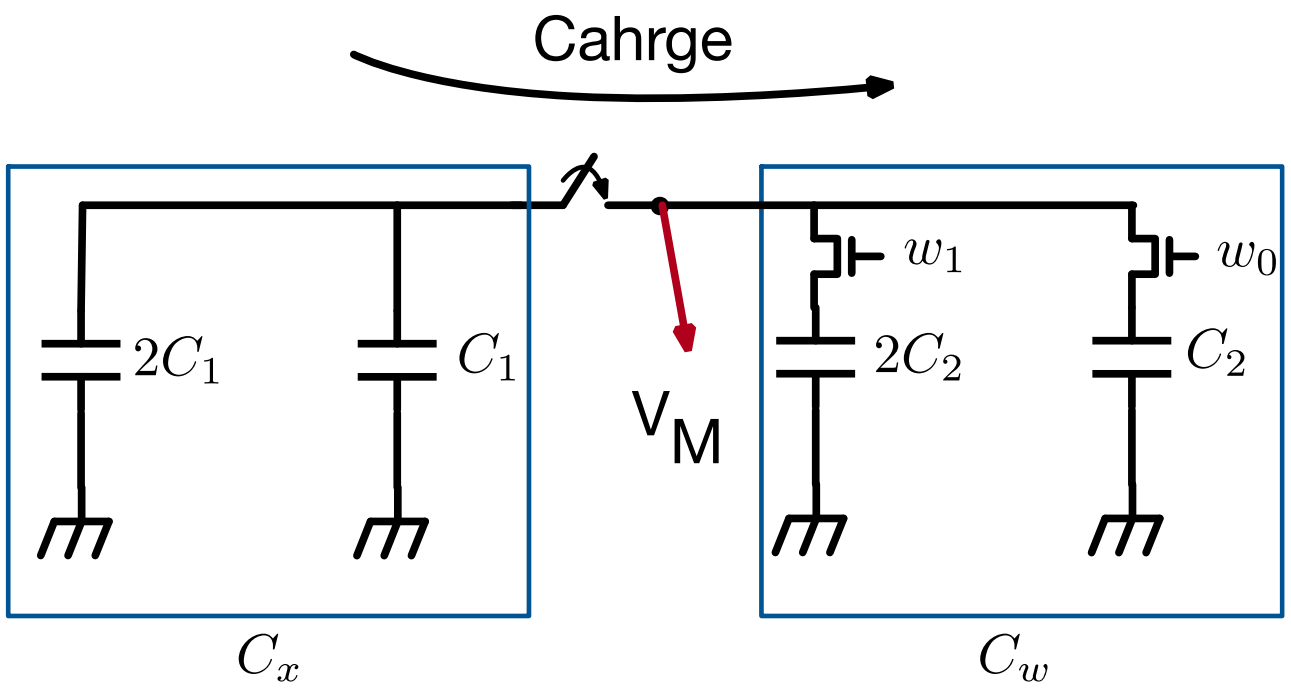
$$V_M = \frac{Q_x}{C_{eq}} = \frac{|X|C_1V_{DD}}{3C_1 + |W|C_2}$$

$$Q_w = C_wV_M = |W|C_2V_M$$

$$Q_w = |X||W|\frac{C_1C_2V_{DD}}{3C_1 + |W|C_2}$$

The sampled charge by C_x is shared by C_w .

Low-Bitwidth Switched-Capacitor MACC



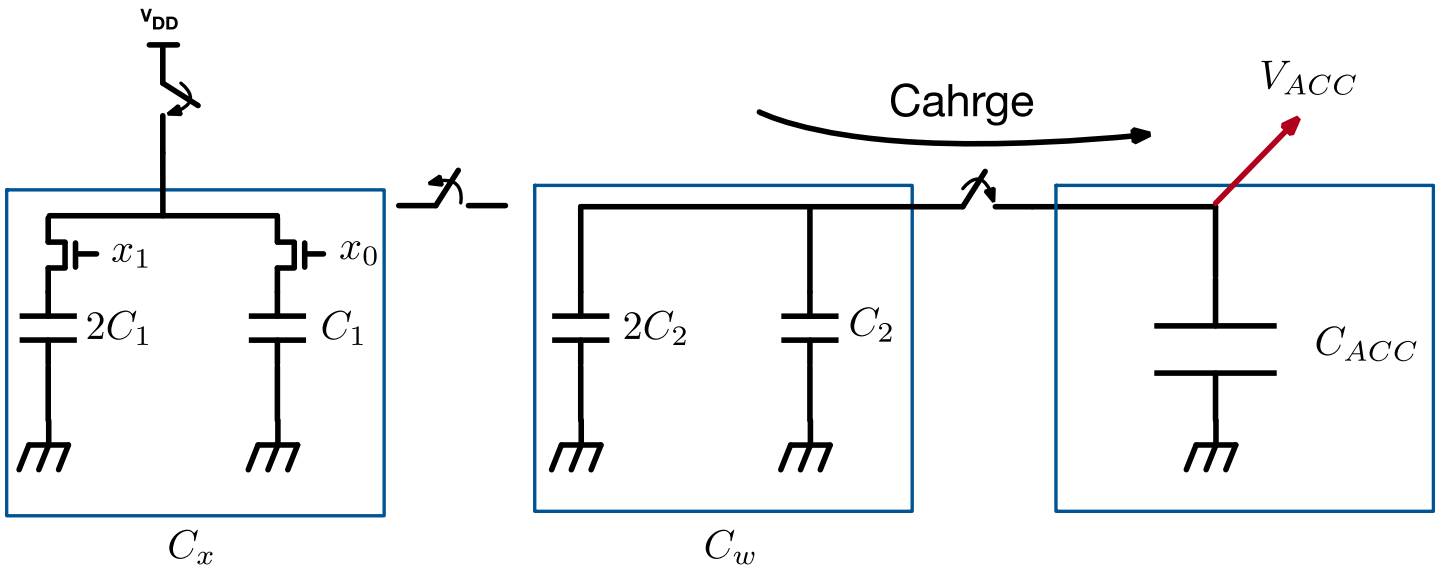
$$C_1 >> C_2 \quad \frac{C_1}{C_2} \to \infty$$

$$Q_w = |X||W|\frac{C_2 V_{DD}}{3}$$

$$Q_w \propto |X||W|$$

A charge proportional to |X||W| is stored on C_w and a multiplication happens

Low-Bitwidth Switched-Capacitor MACC

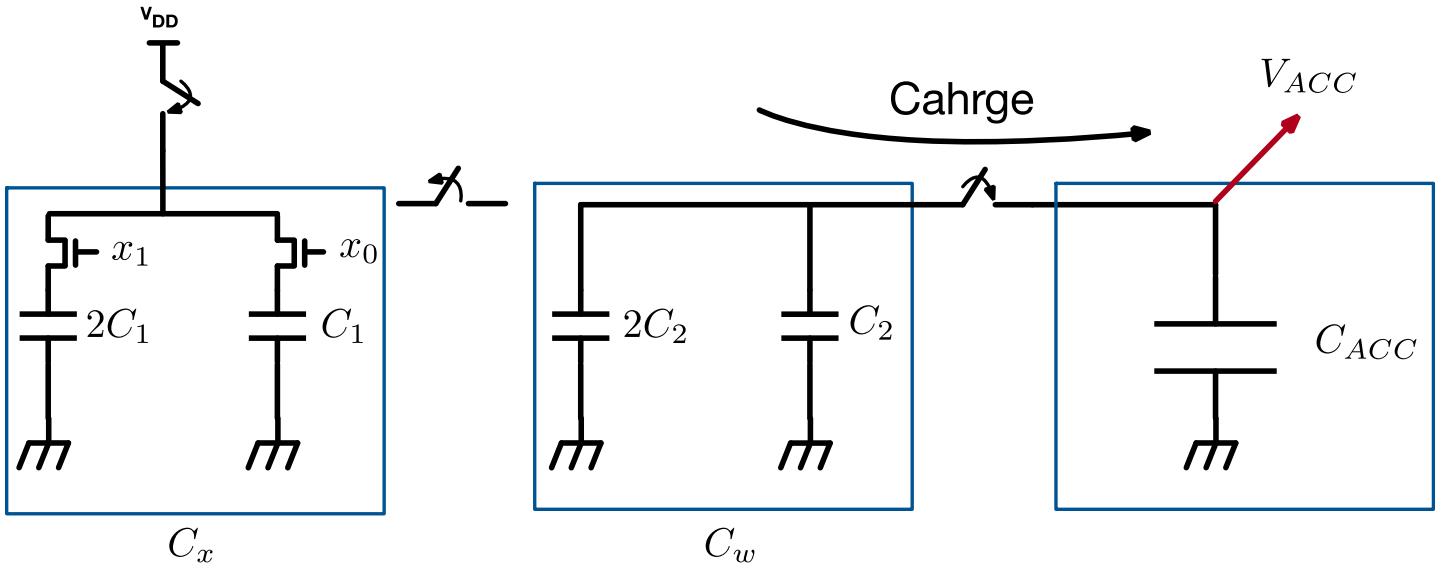


$$C_{ACC} >> C_2$$

$$Q_{ACC} = \frac{C_{ACC}}{C_{ACC} + 3C_2} Q_w \approx 1 \cdot Q_w$$

The sampled charge by C_w is transferred to C_{ACC} and accumulated there

Low-Bitwidth Switched-Capacitor MACC

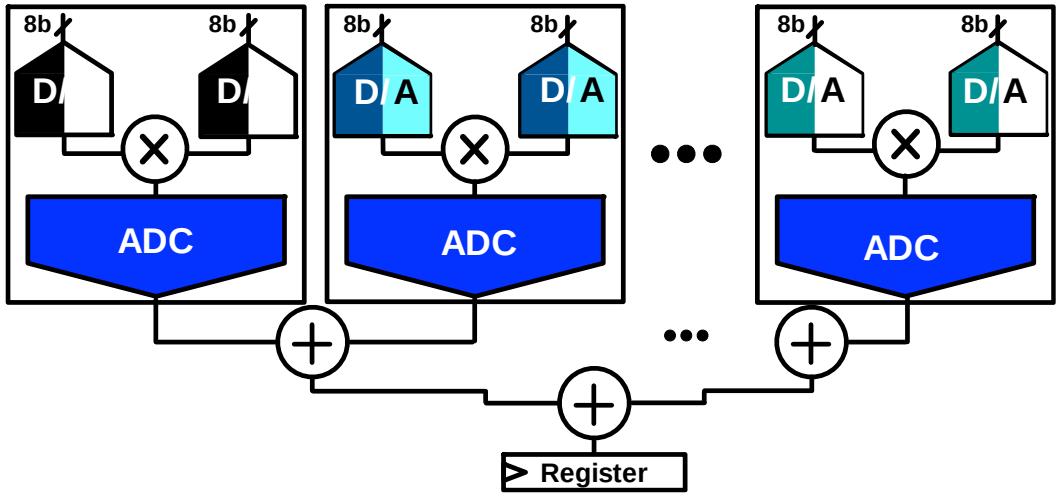


$$V_{ACC} = \frac{Q_{ACC}}{C_{ACC}} \approx \frac{Q_w}{C_{ACC}}$$

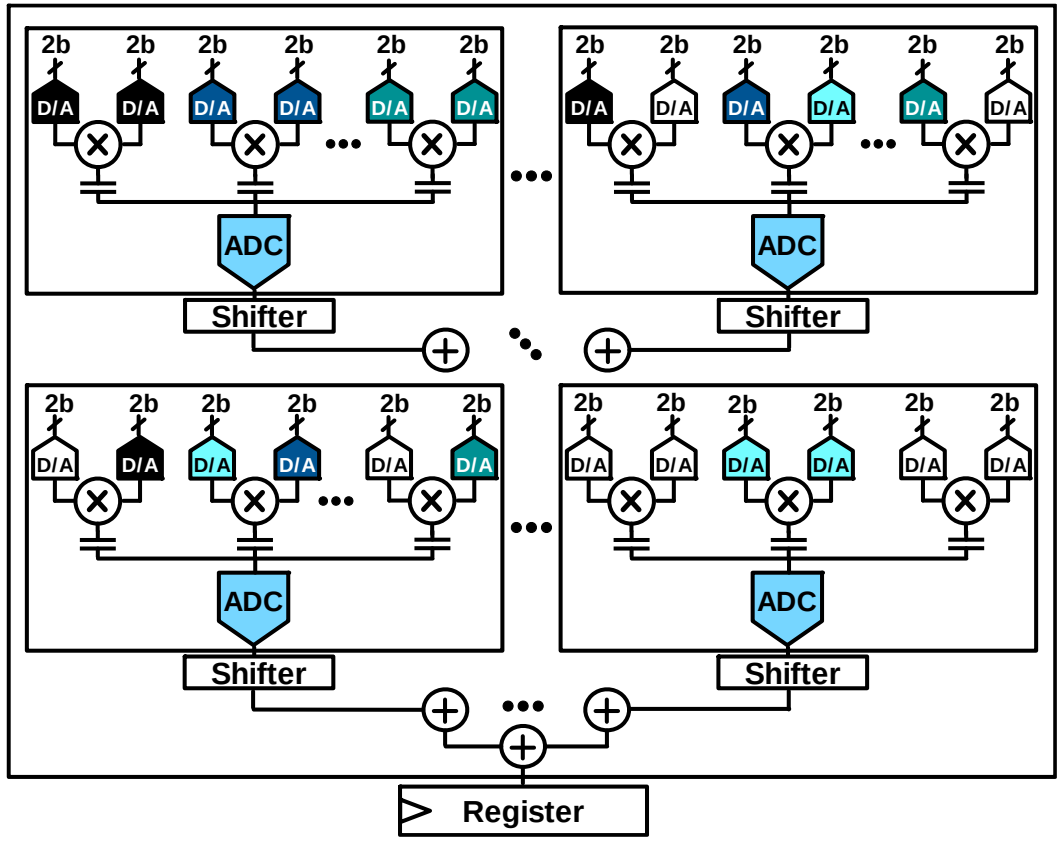
$$V_{ACC} = |X||W| \frac{C_2 V_{DD}}{3C_{ACC}}$$

While the result of the multiplication is being accumulated,
a new input is sampled and a new round begins

BIHIWE Microarchitecture

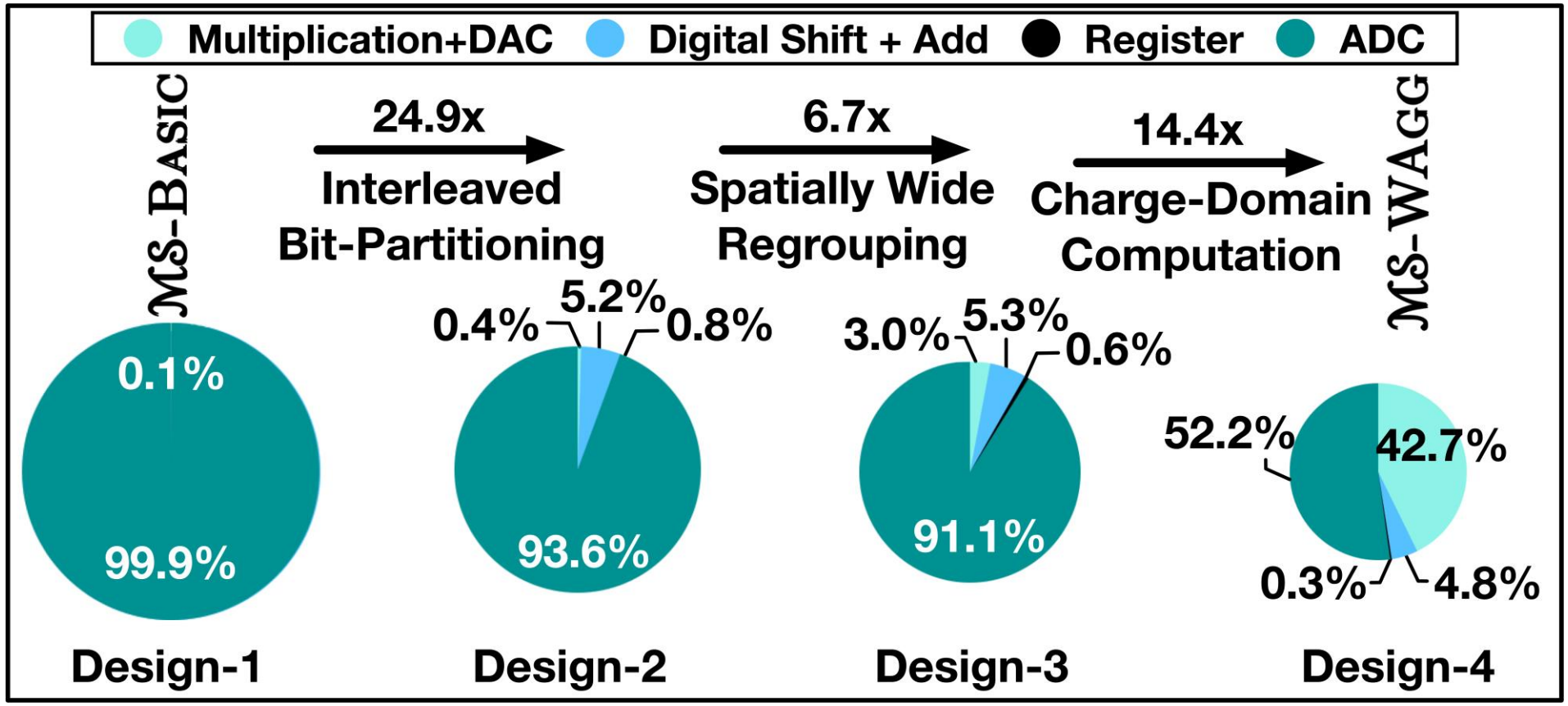


Basic Dot-Product Engine



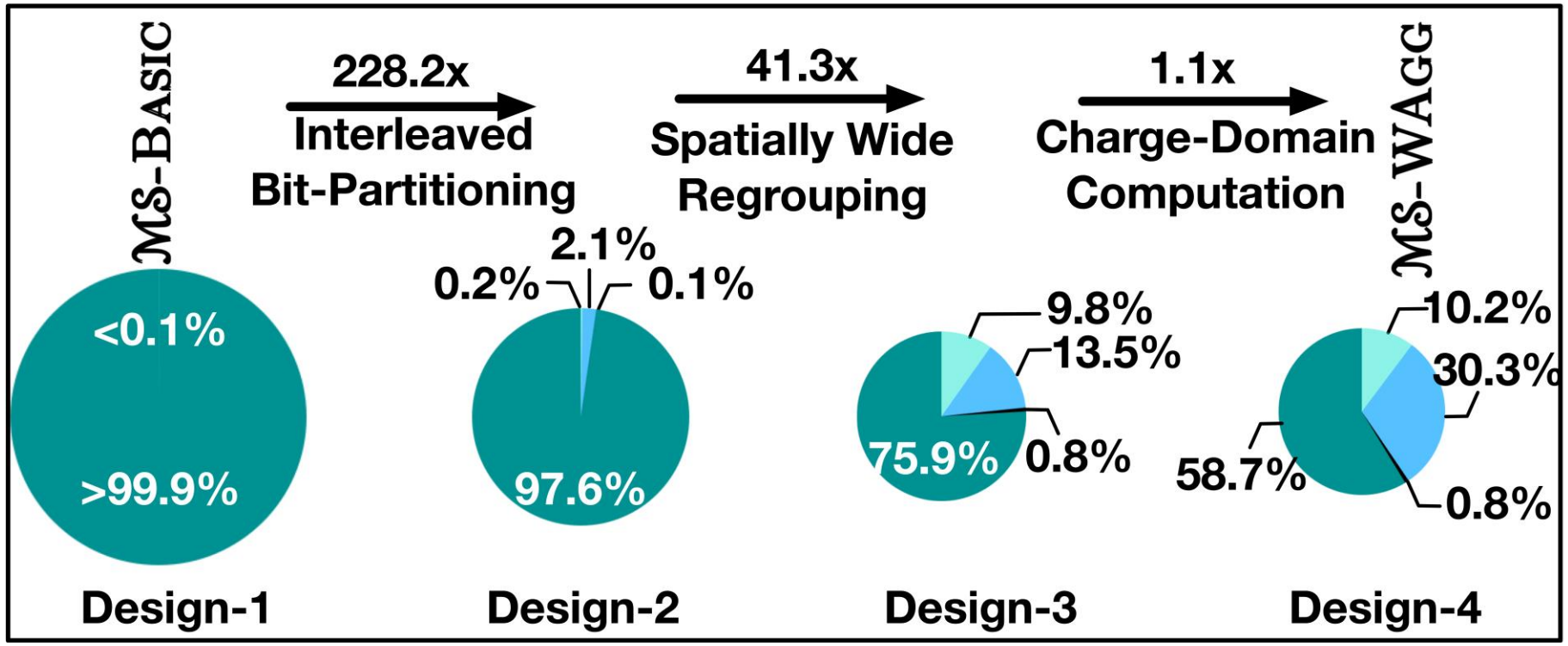
Our Dot-Product Engine:
MS-WAGG

BIHIWE Microarchitecture: Design Decisions & Tradeoffs



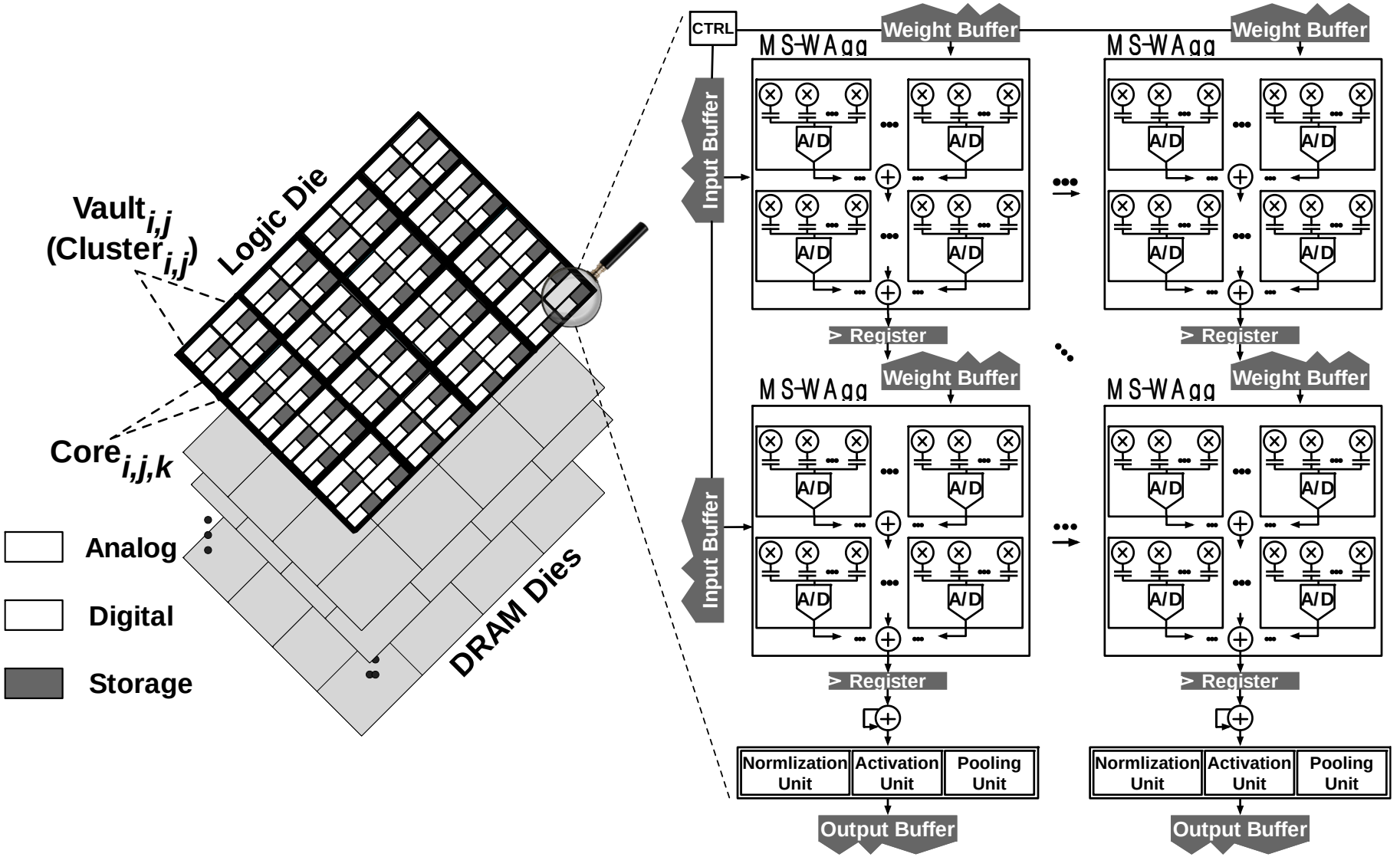
Improvement in Power; Step-by-Step Analysis

BiHiwe Microarchitecture: Design Decisions & Tradeoffs



Improvement in Area; Step-by-Step Analysis

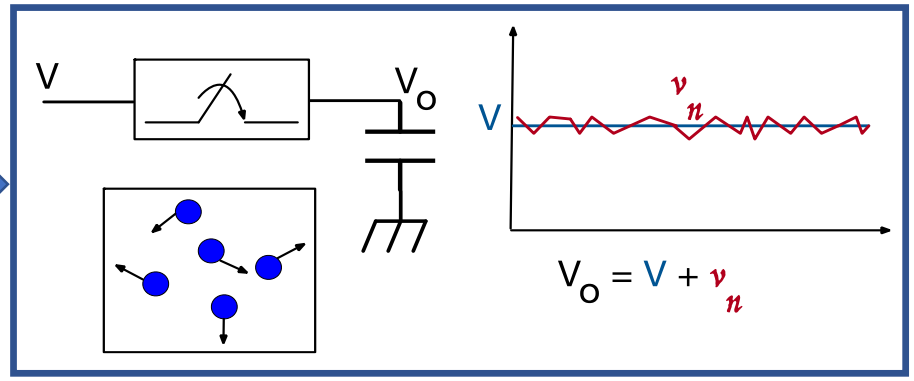
BIHIWE Hierarchical Clustered Architecture



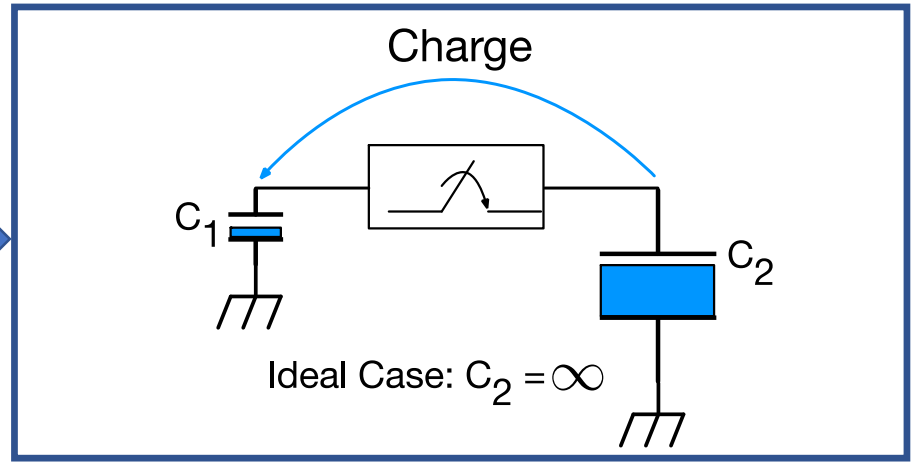
Mixed-Signal Non-Idealities and Their Mitigation

Charge-Domain Computation Non-Idealities

Thermal Noise

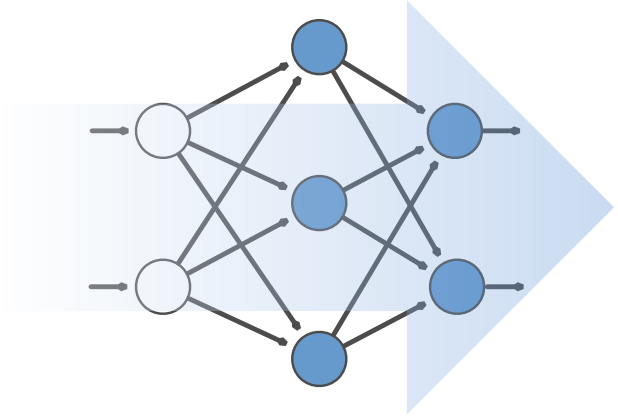


Incomplete Charge Sharing

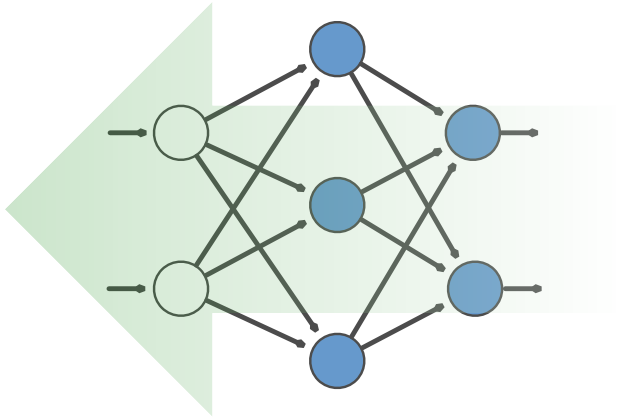


Injecting the non-idealities to the model and fine-tuning the parameters of the model by retraining the network

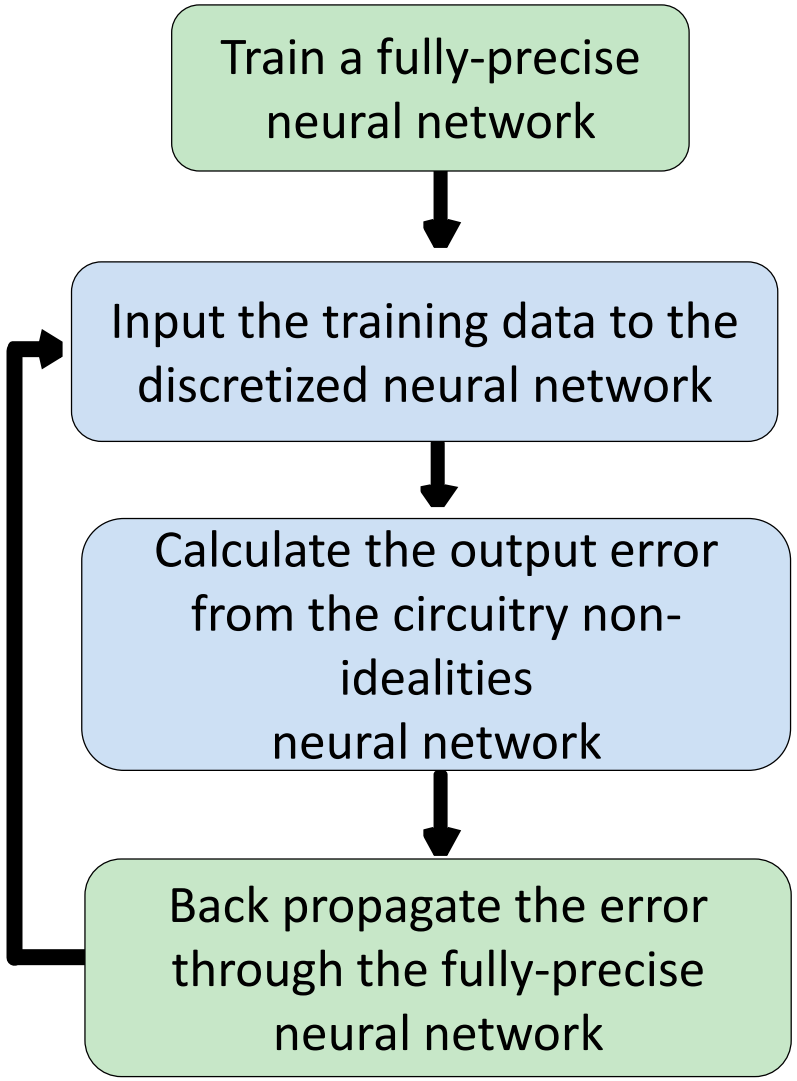
Mixed-Signal Non-Idealities and Their Mitigation



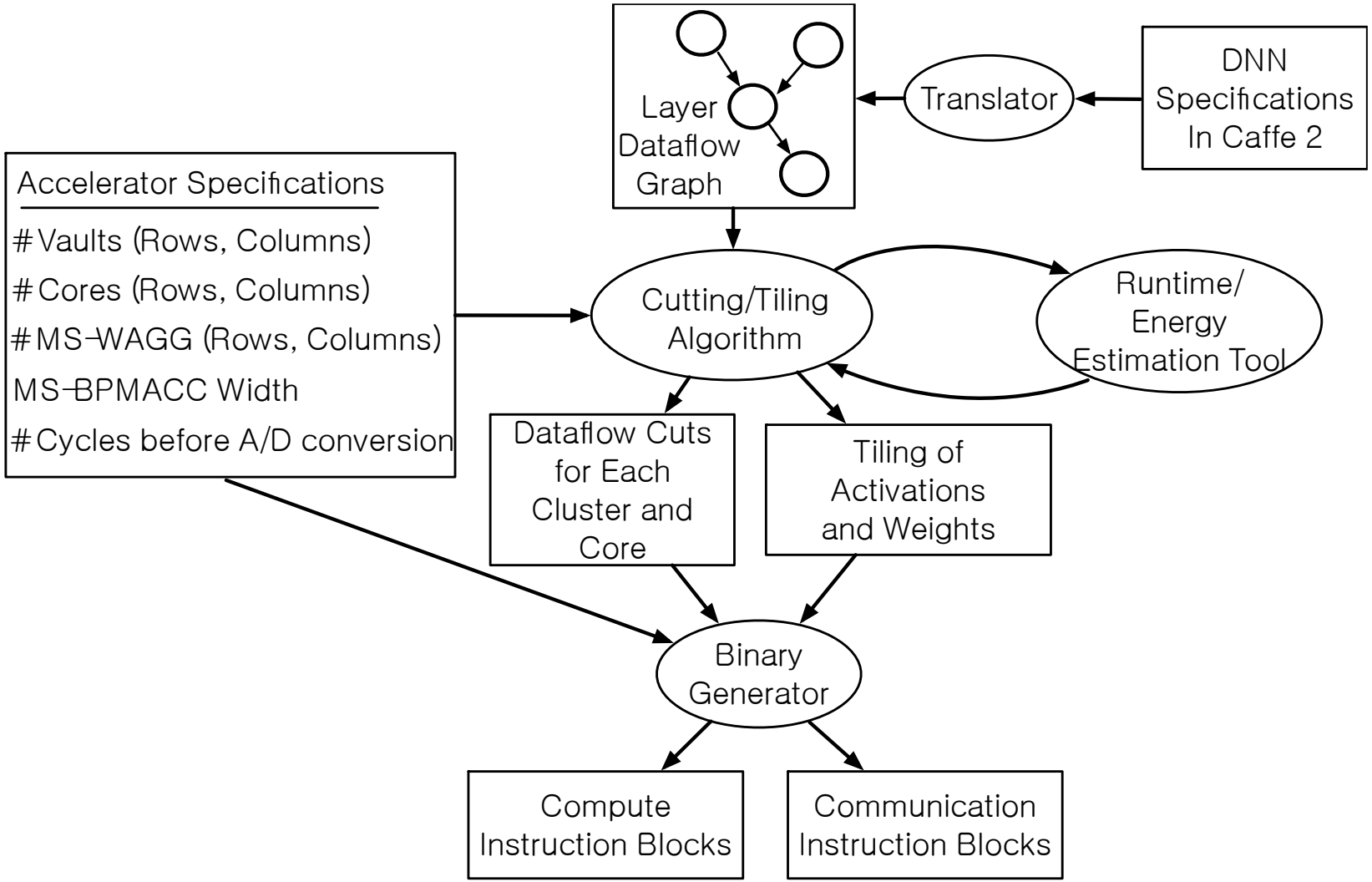
Noisy Network



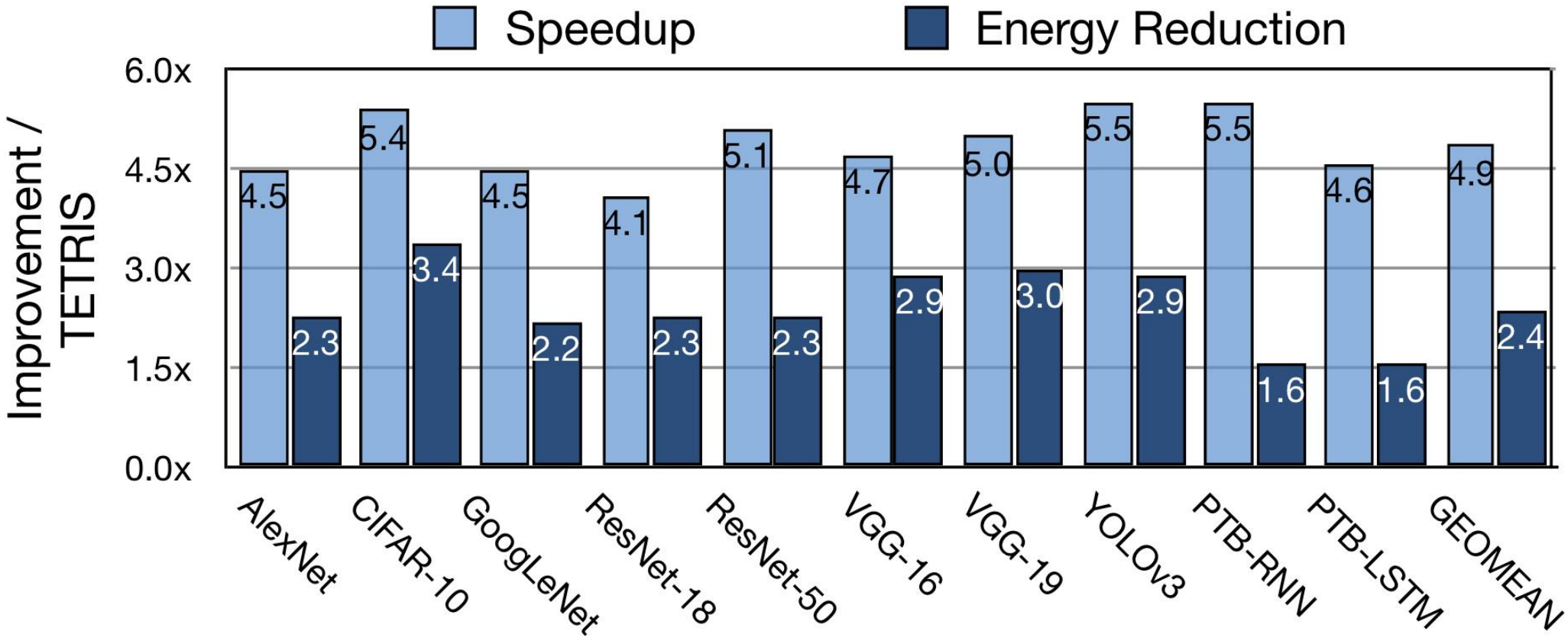
Ideal Network



BIHIWE Compilation Stack

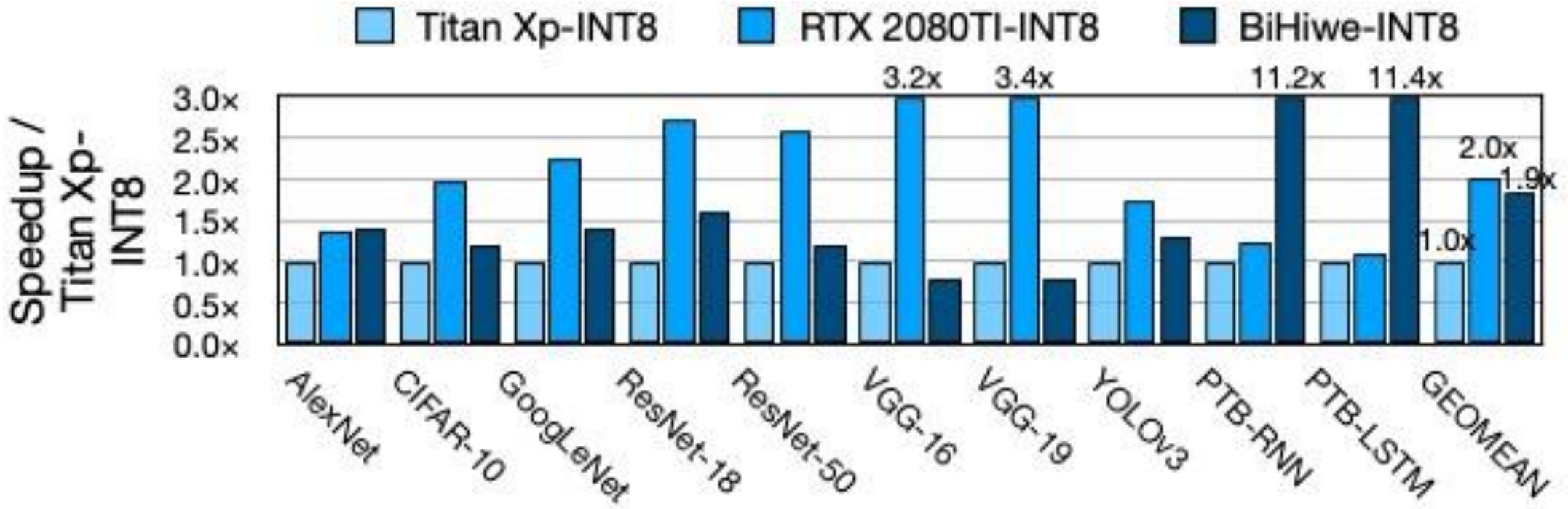


Comparison with TETRIS



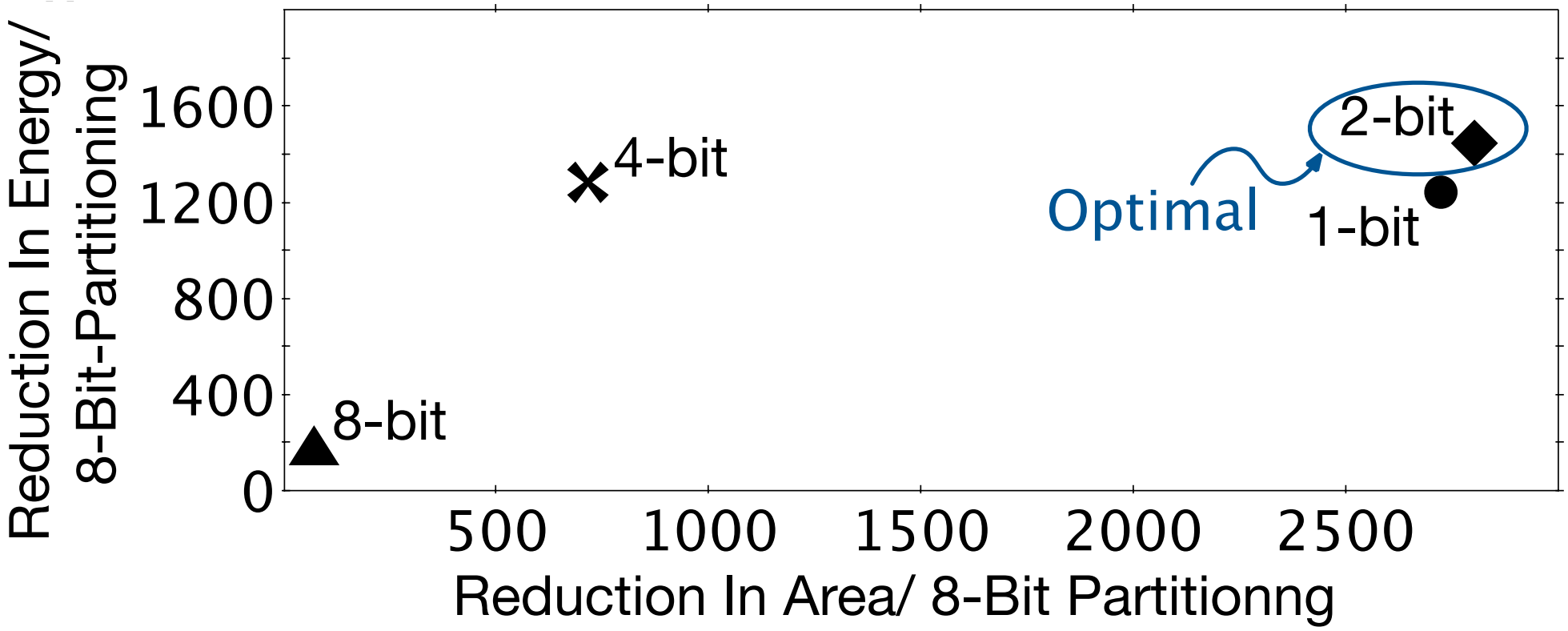
4.9x speedup and 2.4x energy reduction over TETRIS,
an optimized 3D-stacked fully-digital accelerator for DNNs

Comparison with GPUs



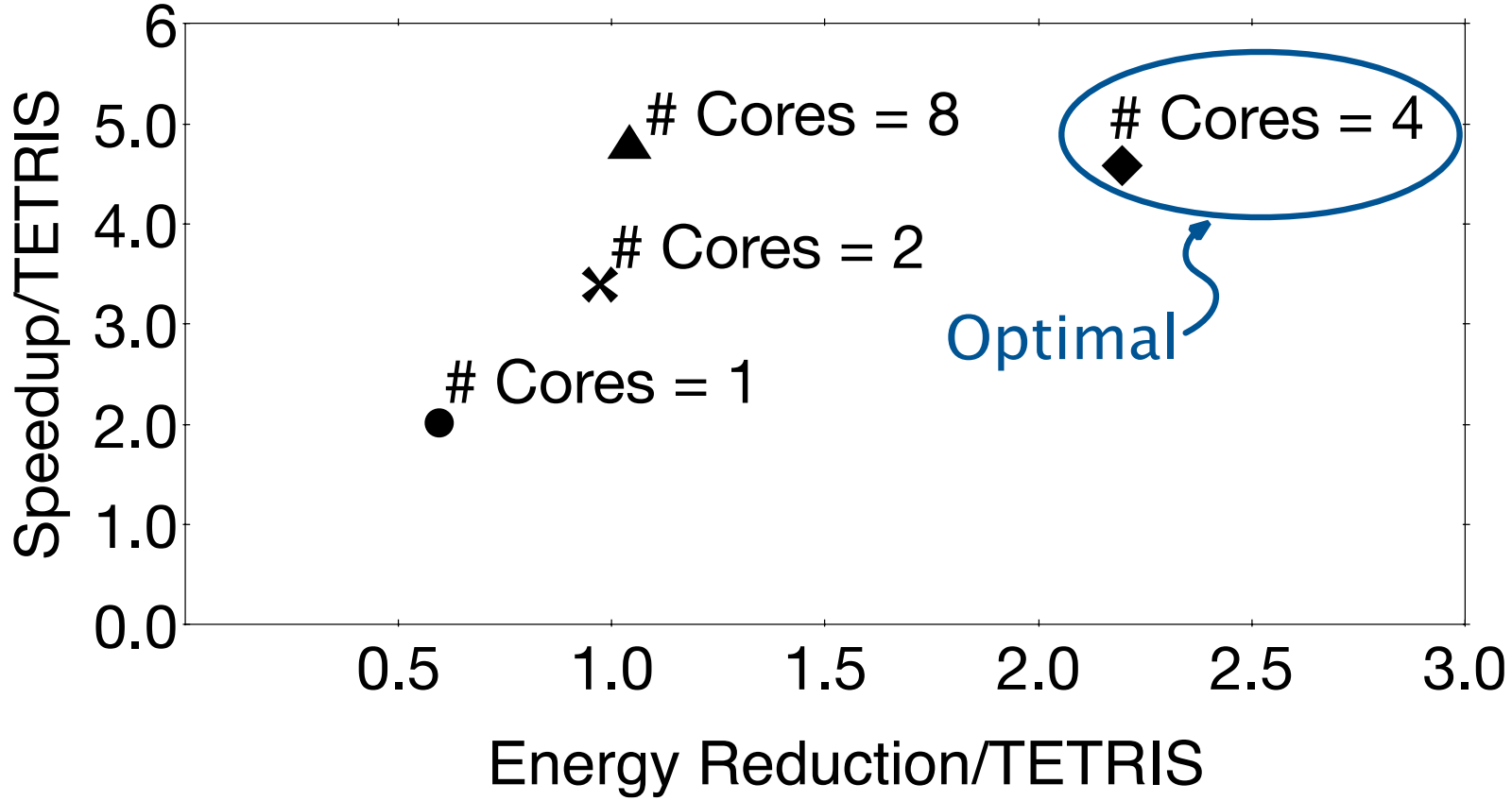
BiHiwe delivers 70.1x and 35.4x higher Performance-per-Watt compared to Nvidia Titan Xp and RTX 2080 TI

Design Space Exploration for Bit-Partitioning



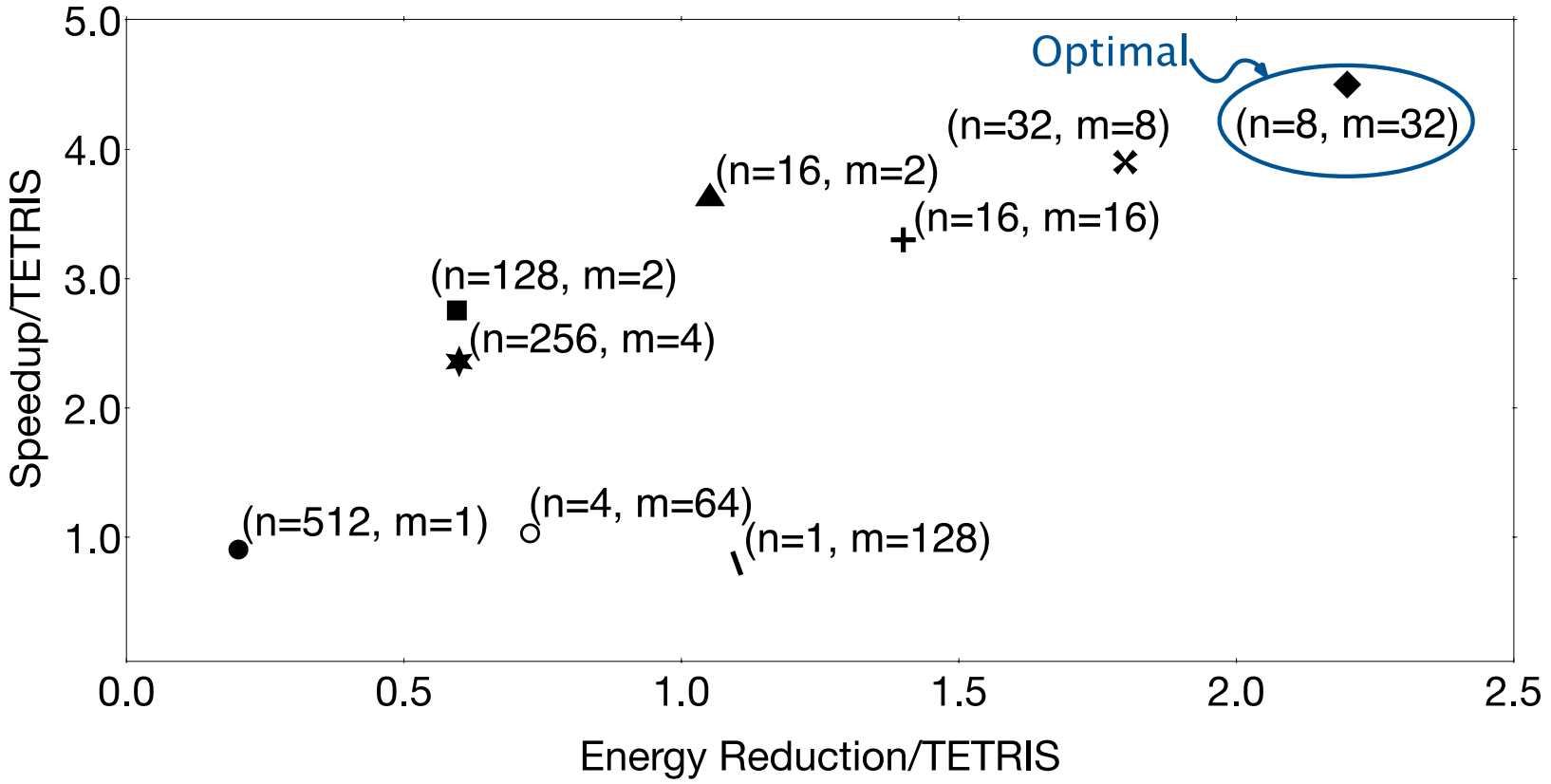
2-bit bit-partitioning is the optimal choice based on this design style and technology node

Design Space Exploration for # of cores



Each cluster (vault) of the BIHIWE consists of four accelerator cores

Design Space Exploration for MS-BPMAcc



Each MS-BPMAcc in BIHIWE has an array of 8 low-bitwidth MACC units which perform operations for 32 cycles before A/D conversion

Evaluating Circuitry Non-Idealities

DNN Model	Dataset	Top-1 Accuracy (With Non-Idealities)	Top-1 Accuracy (After Fine-Tuning)	Top-1 Accuracy (Ideal)	Final Accuracy Loss
AlexNet	Imagenet	53.12%	56.64%	57.11%	0.47 %
CIFAR-10	CIFAR-10	90.82%	91.01%	91.03%	0.02 %
GoogLeNet	Imagenet	67.15%	68.39%	68.72%	0.33 %
ResNet-18	Imagenet	66.91%	68.96%	68.98%	0.02 %
ResNet-50	Imagenet	74.5%	75.21%	75.25%	0.04 %
VGG-16	Imagenet	70.31%	71.28%	71.46%	0.18 %
VGG-19	Imagenet	73.24%	74.20%	74.52%	0.32 %
YOLOv3	Imagenet	75.92%	77.1%	77.22%	0.21 %
PTB-RNN	Penn TreeBank	1.1 BPC	1.6 BPC	1.1 BPC	0.0 BPC
PTB-LSTM	Penn TreeBank	97 PPW	170 PPW	97 PPW	0.0 PPW

BIHIWE has no virtual impact on the classification accuracy of the DNN models