AFE-CIM: A Current-Domain Compute-In-Memory Macro for Analog-to-Feature Extraction

Sudarshan Sharma∗, Wei-Chun Wang∗, Coleman DeLude, Minah Lee, Nael Mizanur Rahman, Narasimha Vasishta Kidambi, Justin Romberg and Saibal Mukhopadhyay

Electrical and Computer Engineering

Georgia Institute of Technology, USA

ssharma497@gatech.edu

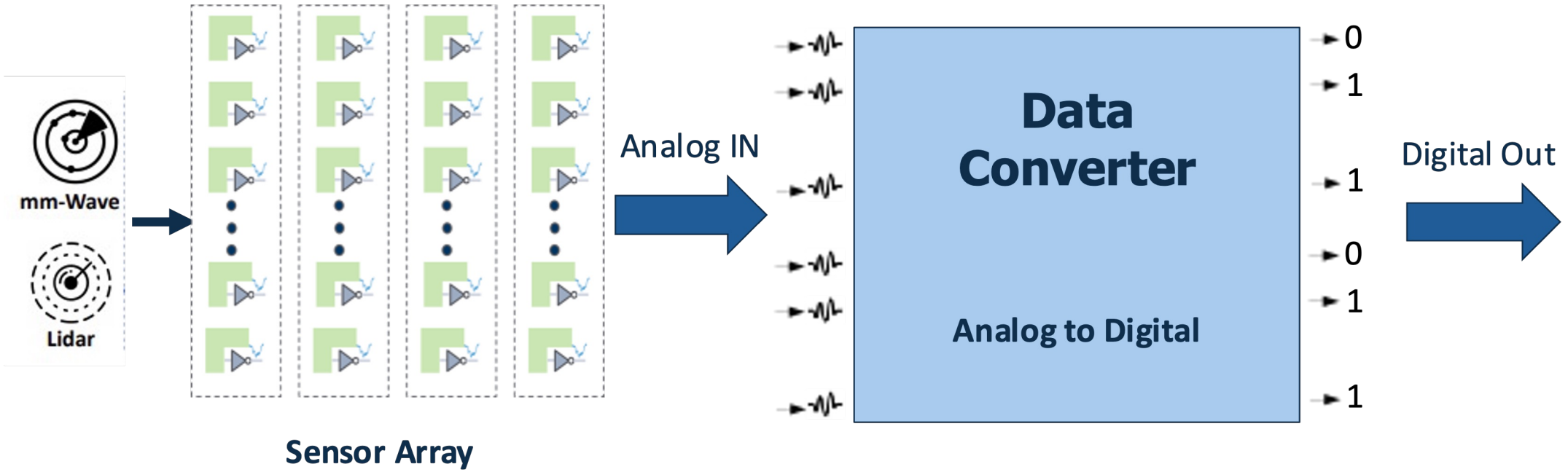
**equal contribution*

Outline

- Motivation
- AFE-CIM
  - Approach
  - Circuit Motivation
  - Architecture
  - Measurement Result
  - Chip Summary
  - Application- Simulation Study from Measurement Data
    - Digital Beamforming
    - Image Classification

AFE-Motivation

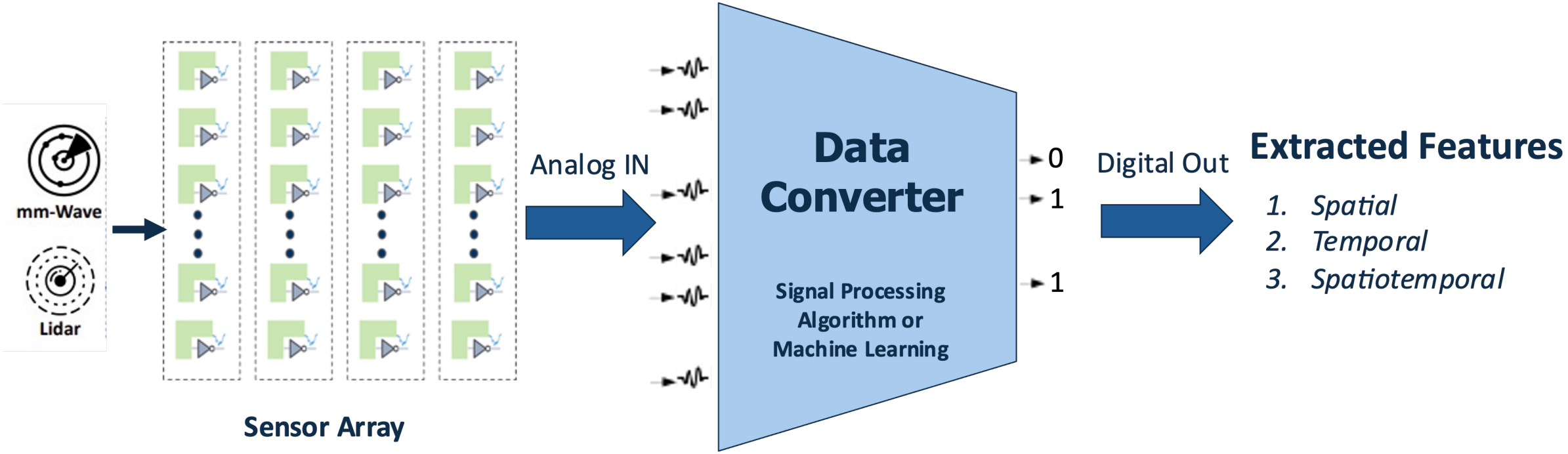
Typical Sensor Front-End



Problems

- Larger ADC power
- Data deluge challenge

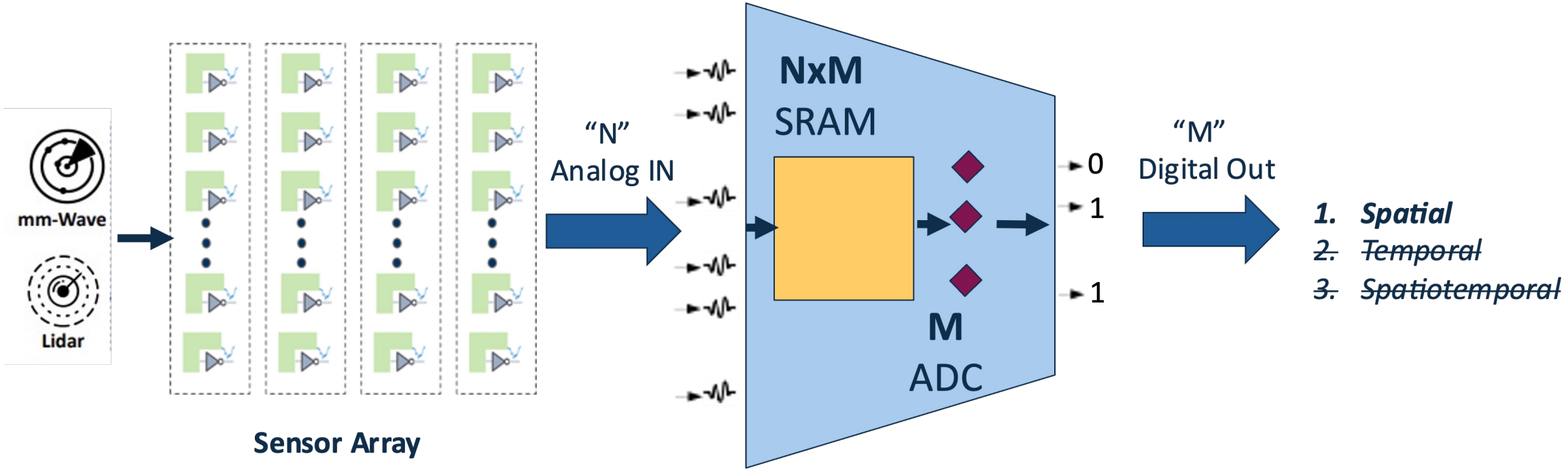
Analog-To-Feature Extraction



Reduces data while retaining quality for downstream tasks

- Reduction in sensor power
- Reduction in output data volume

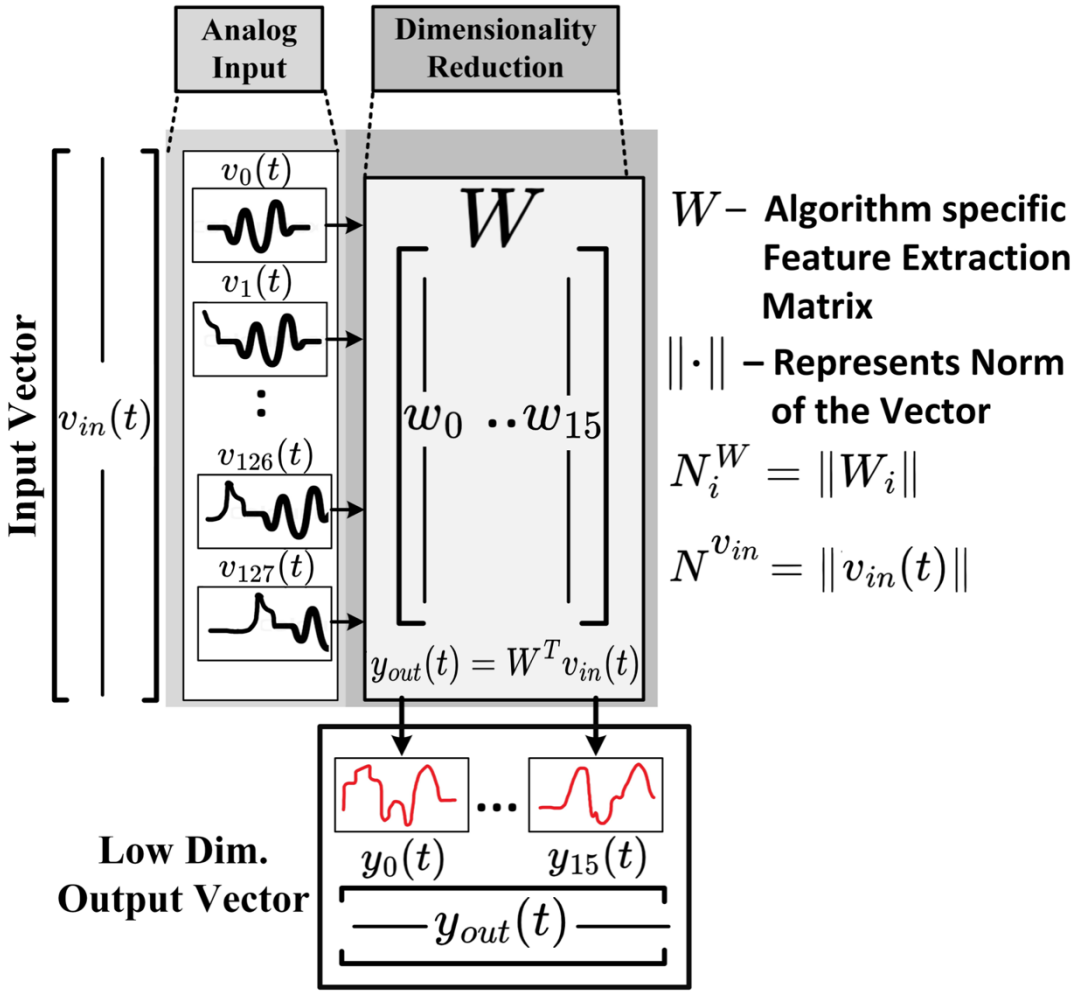
AFE Compute-IN-Memory



Analog-Vector Digital Matrix Multiplication (A/D VMM) using Compute-In-Memory (CIM) technique.

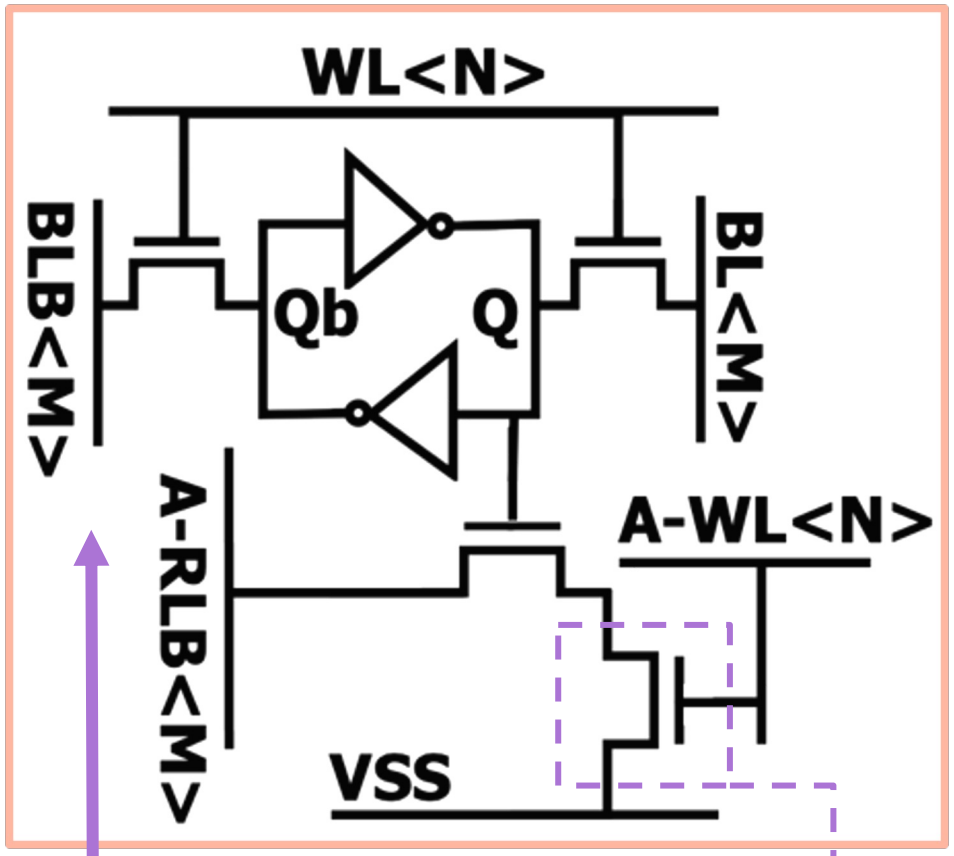
AFE Compute-In-Memory (CIM)

Approach



- The design employs an analog-vector digital-matrix multiplication (A/D-VMM) engine to compute a weighted linear combination of input analog signals to generate lower-dimension digital features.
- The real time update of weight matrix ***W*** supports adaptive feature extraction

Circuit Level Motivation - I



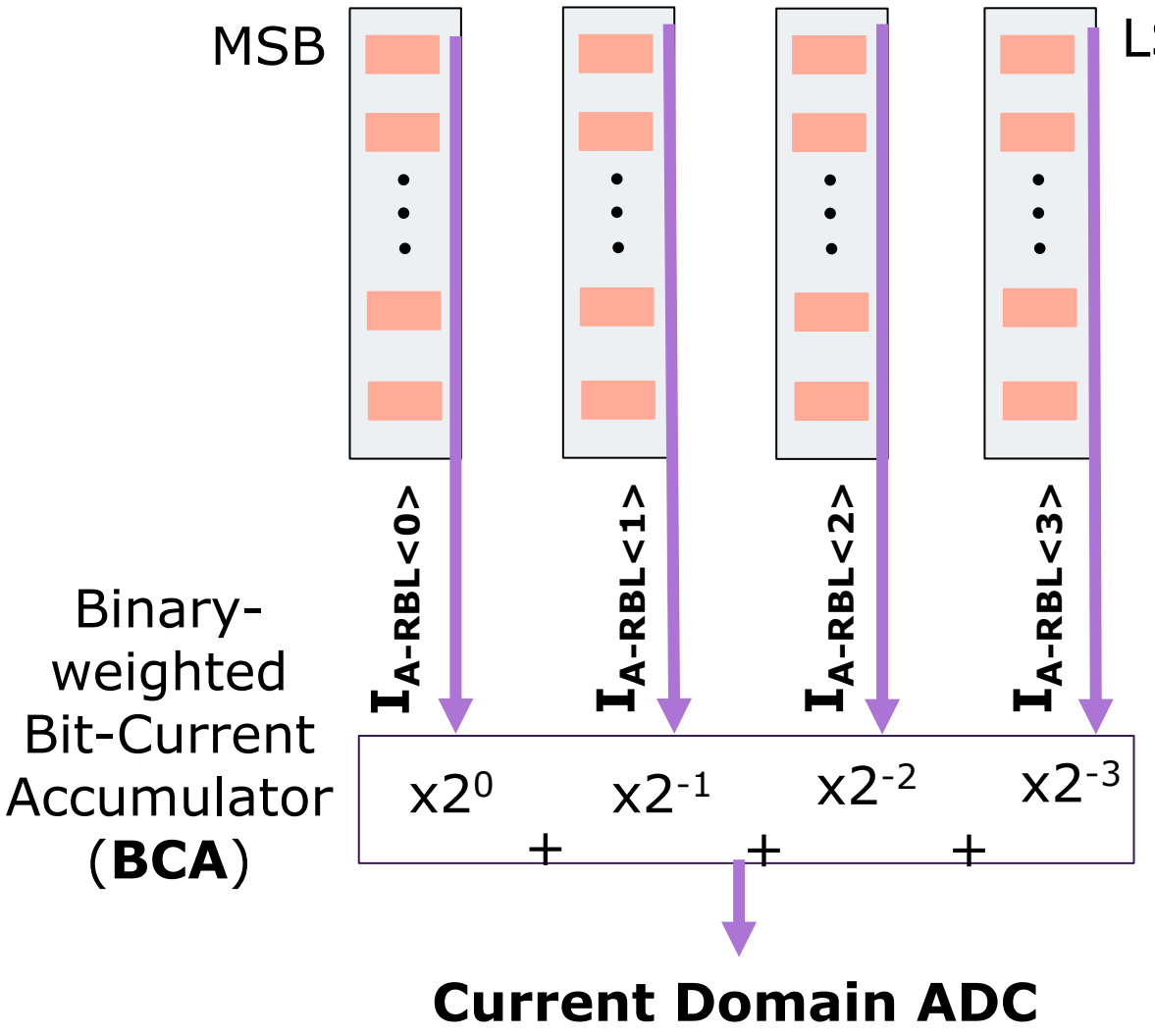
Analog Input (V_{in})

$I_{cell} \propto Q*V_{in}$

Operates in Linear Region

- Upon varying the A-WL from 0.4 to 1V and storing 1 or 0 in the SRAM, we observe a highly linear on-current curve and negligible off-current respectively.

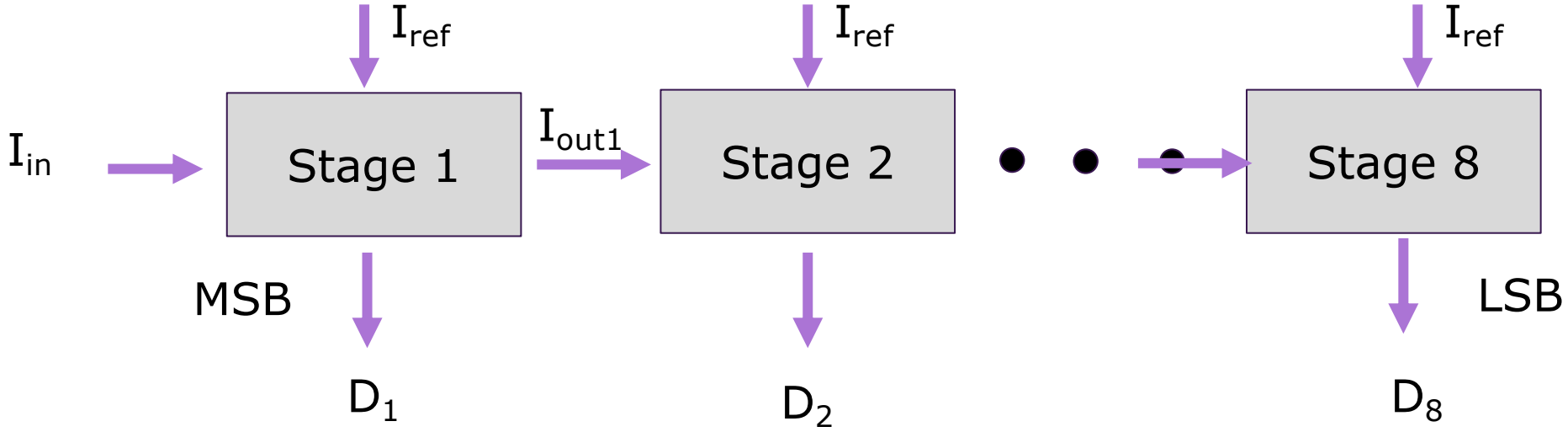
Circuit Level Motivation - II



- Accumulated current through each bitline undergoes shift and add operation in analog domain
- Current mirror ratios for each column designed to reduce the current before combination.

Circuit Level Motivation - III

Current Domain ADC



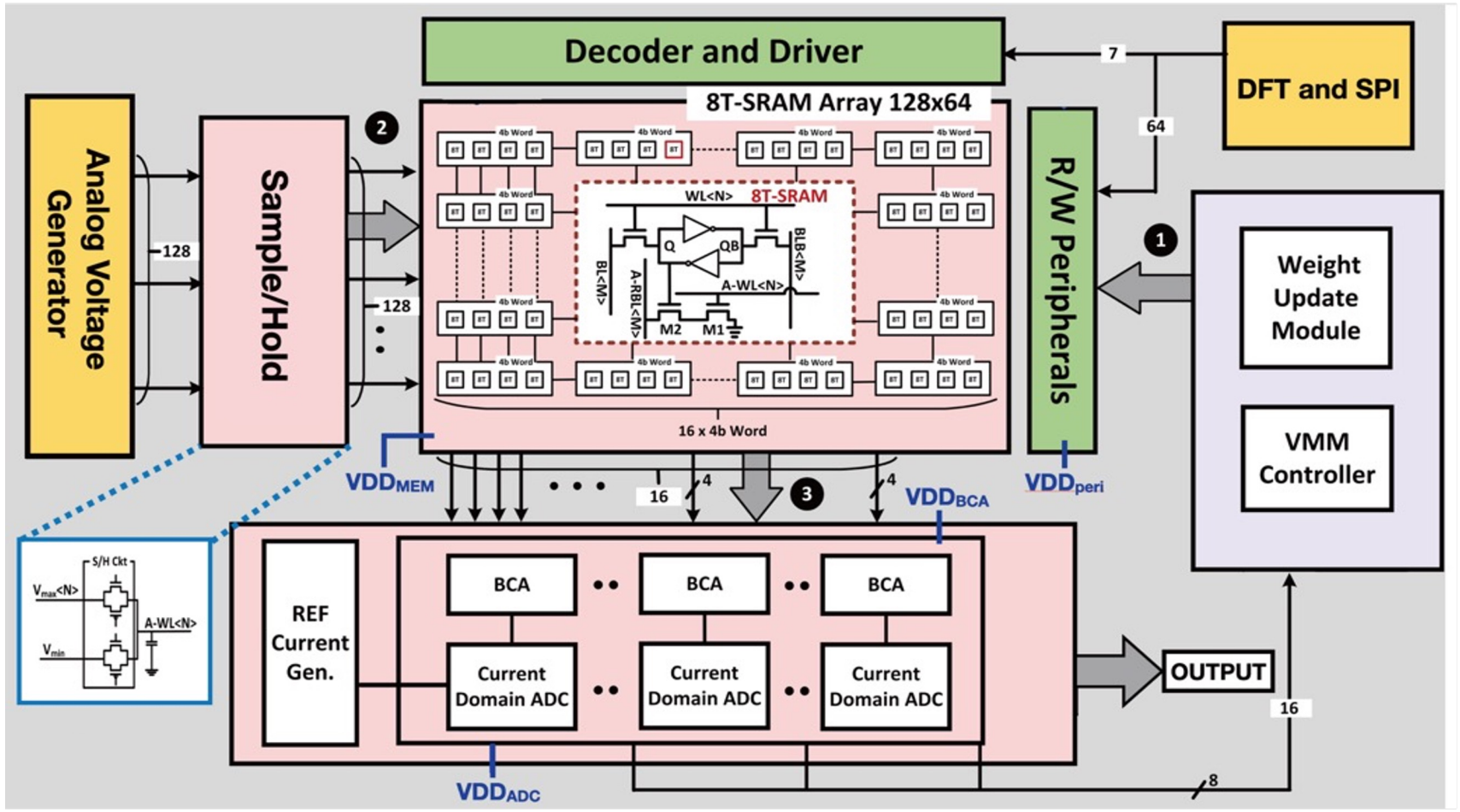
$2*I_{in} < I_{ref} \rightarrow D_1 = 0$ and $I_{out1} = 2*I_{in}$

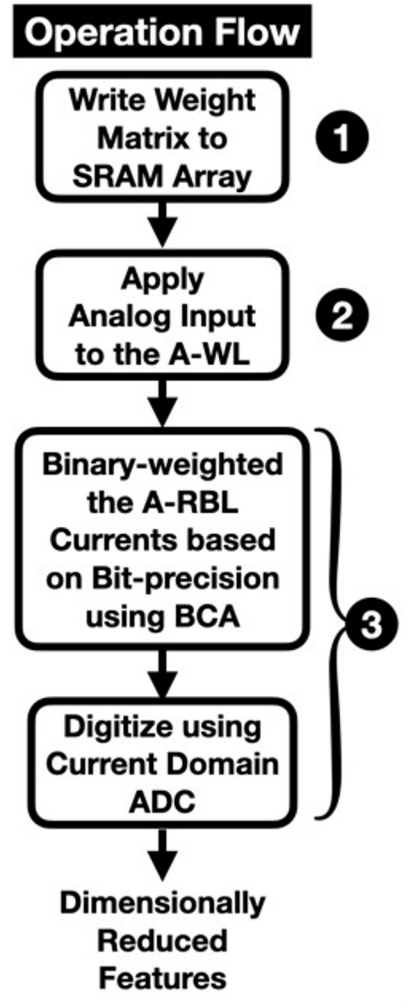
$2*I_{in} > I_{ref} \rightarrow D_1 = 1$ and $I_{out1} = 2*I_{in} - I_{ref}$

$I_{ref} = I_{max}$

[1] B. D. Smith, "An Unusual Electronic Analog-Digital Conversion Method," in *IRE Transactions on Instrumentation*, vol. PGI-5, pp. 155-160, June 1956, doi: 10.1109/IRE-I.1956.5007017.
[2] D. G. Nairn and C. A. T. Salama, "Current-mode algorithmic analog-to-digital converters," in *IEEE Journal of Solid-State Circuits*, vol. 25, no. 4, pp. 997-1004, Aug. 1990, doi: 10.1109/4.58292.

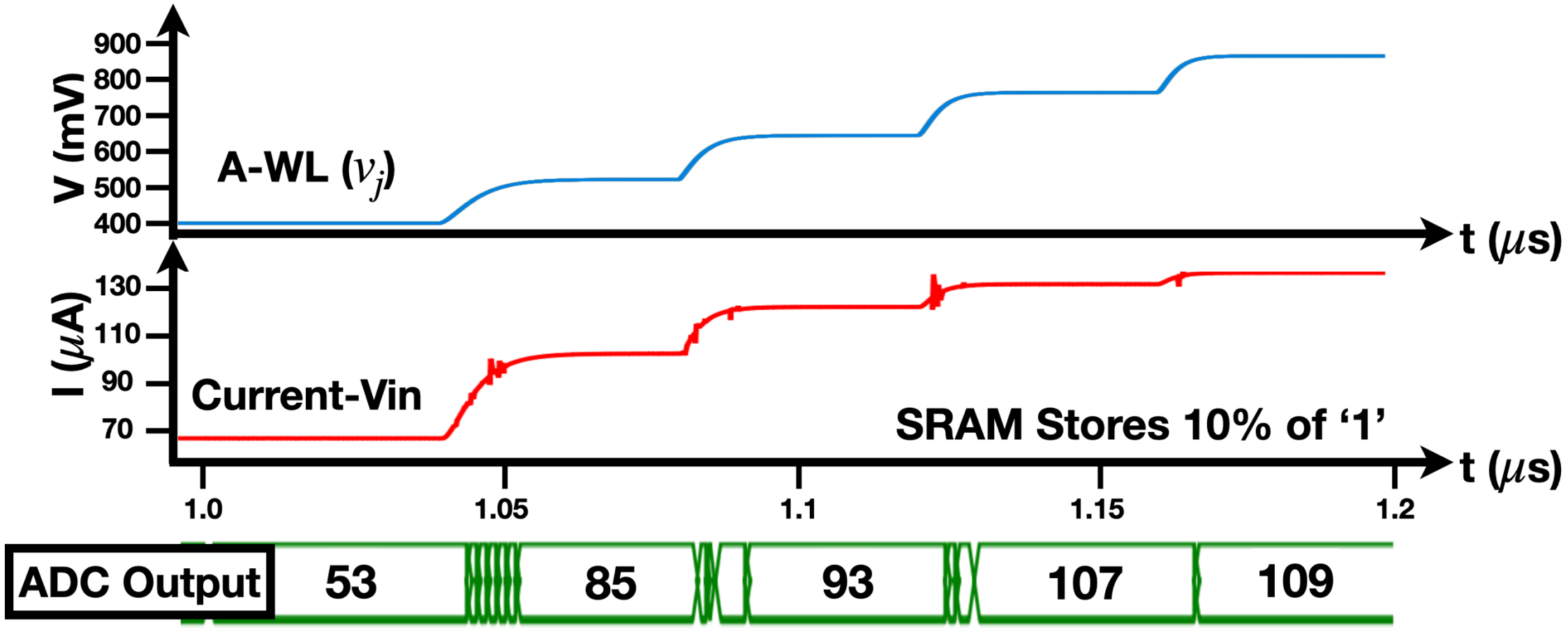
Architecture



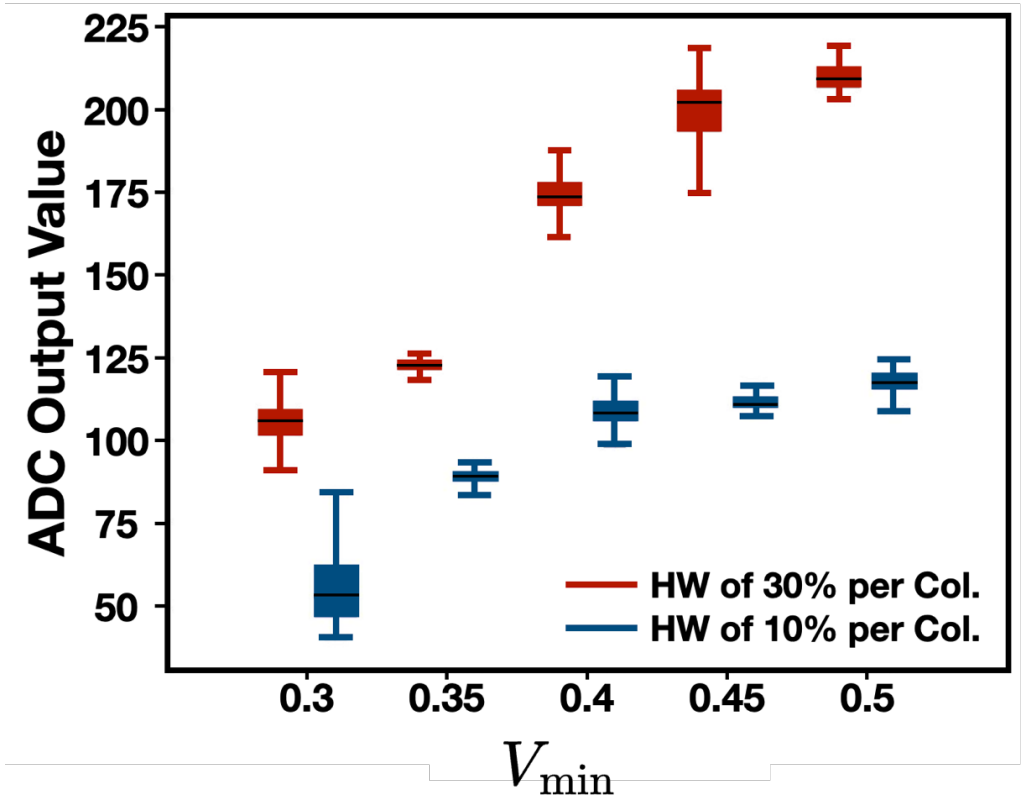


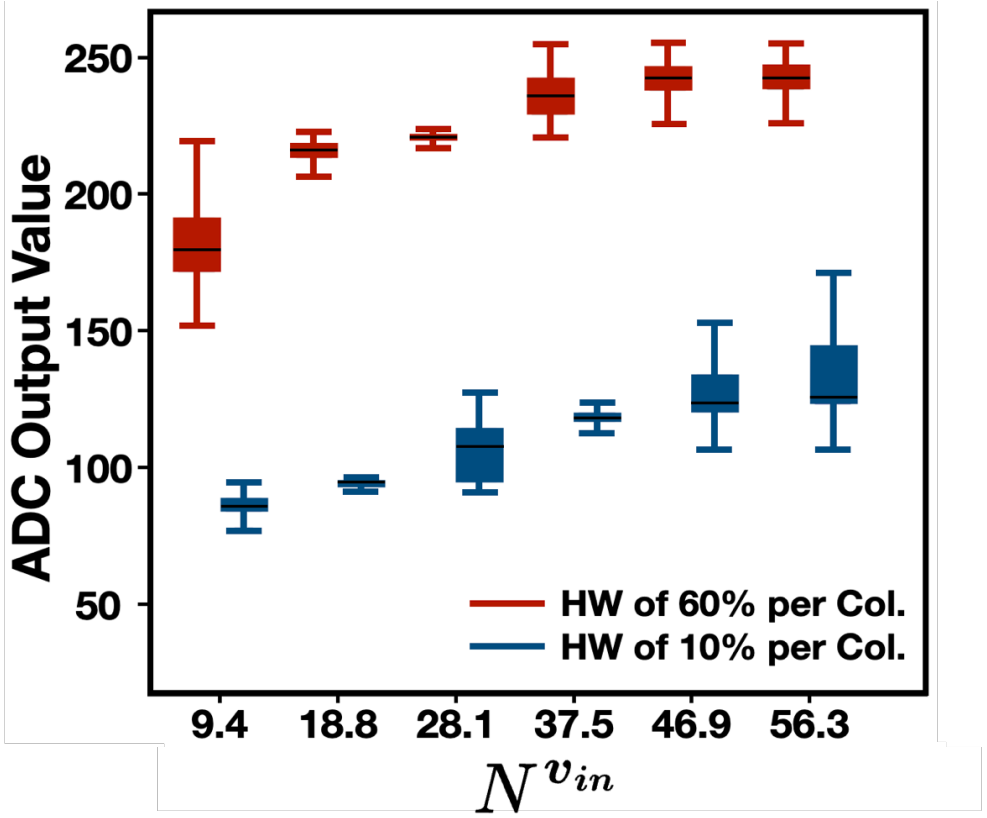
AFE-CIM Simulation and Measurement Results

Simulation


V (mV)
900
800
700
600
500
400
A-WL (v_j)
t (μs)
I (μA)
130
110
90
70
Current-Vin
SRAM Stores 10% of '1'
t (μs)
1.0
1.05
1.1
1.15
1.2
ADC Output
53
85
93
107
109

Measurement Results –I

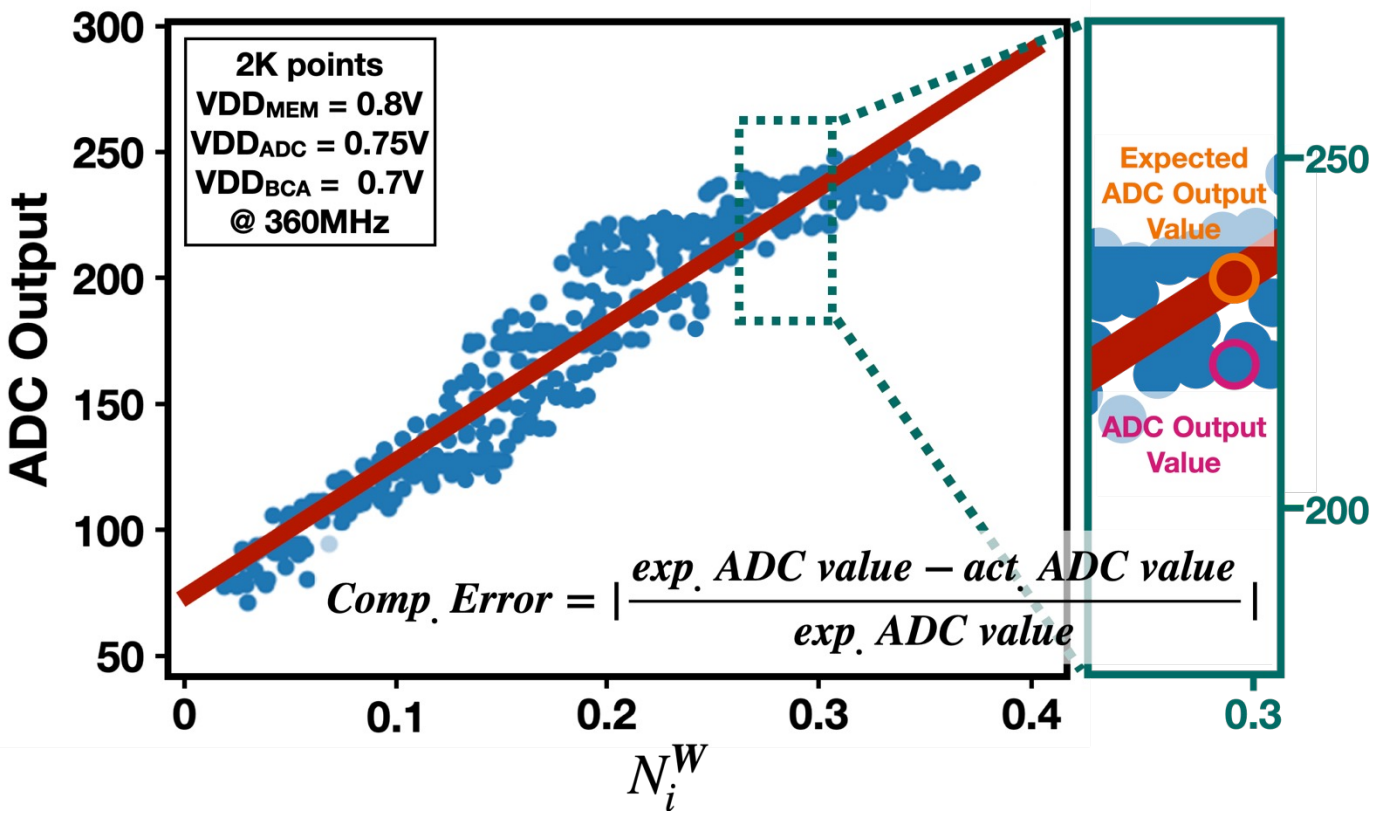


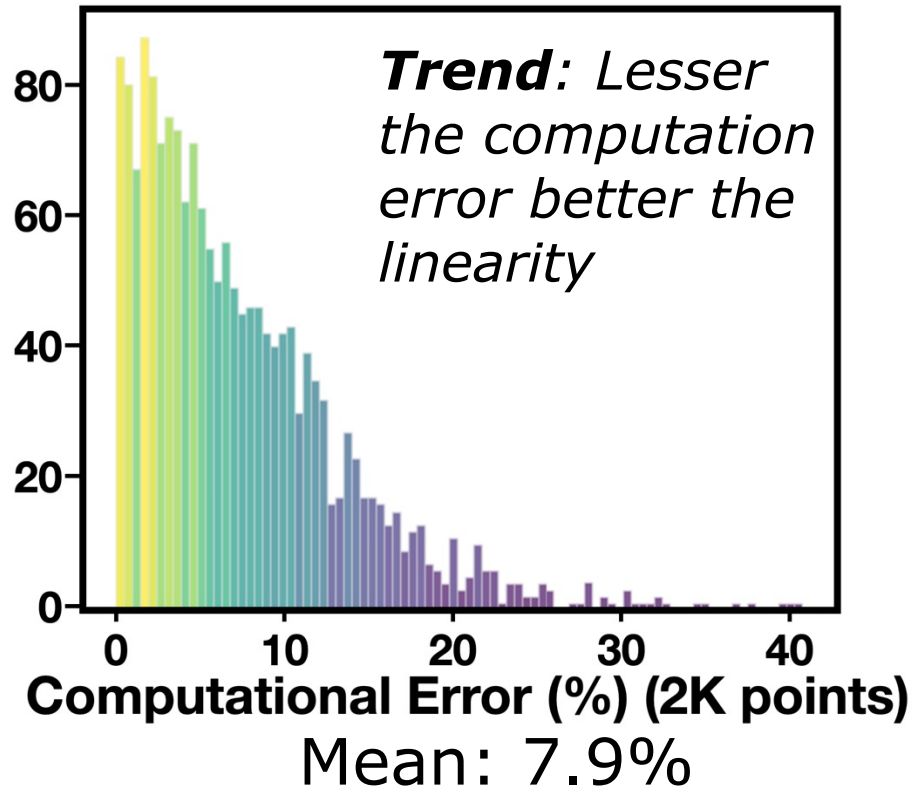


***Trend**: More Linear the better*

- The features $[\boldsymbol{y_{out}(t)}]$ generated by AFE-CIM are measured considering random $\boldsymbol{W}$ and/or varying $\boldsymbol{v_{in}(t)}$. The ADC output increases linearly with V_{min} and with the norm of the input vector $\boldsymbol{v_{in}}$.

Measurement Results –II

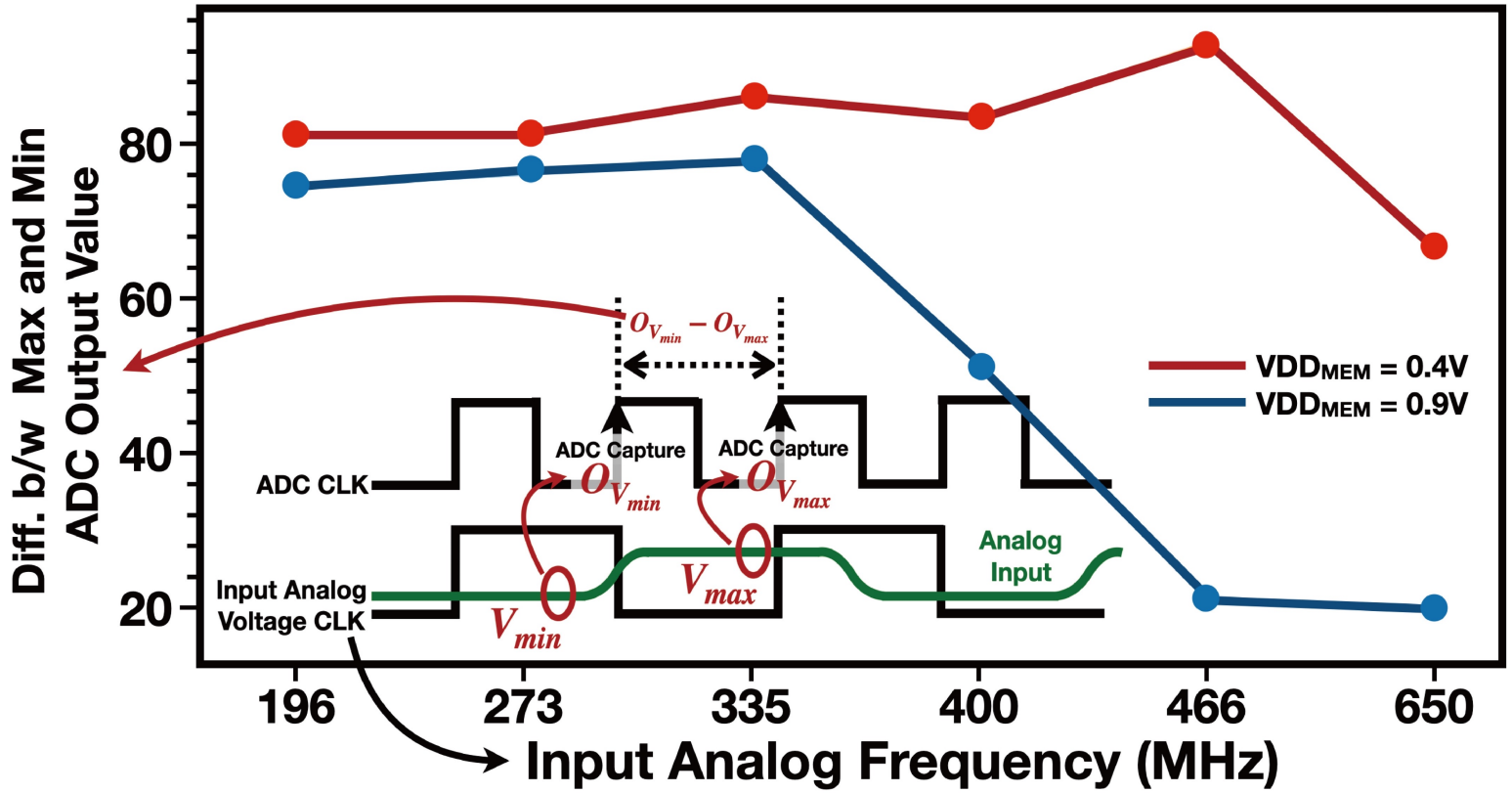




Mean: 7.9%

- Given a fixed v_{in} and random W, the measured ADC output for the feature $y_k(t)$ increases linearly with an increase in $||W_k||$.

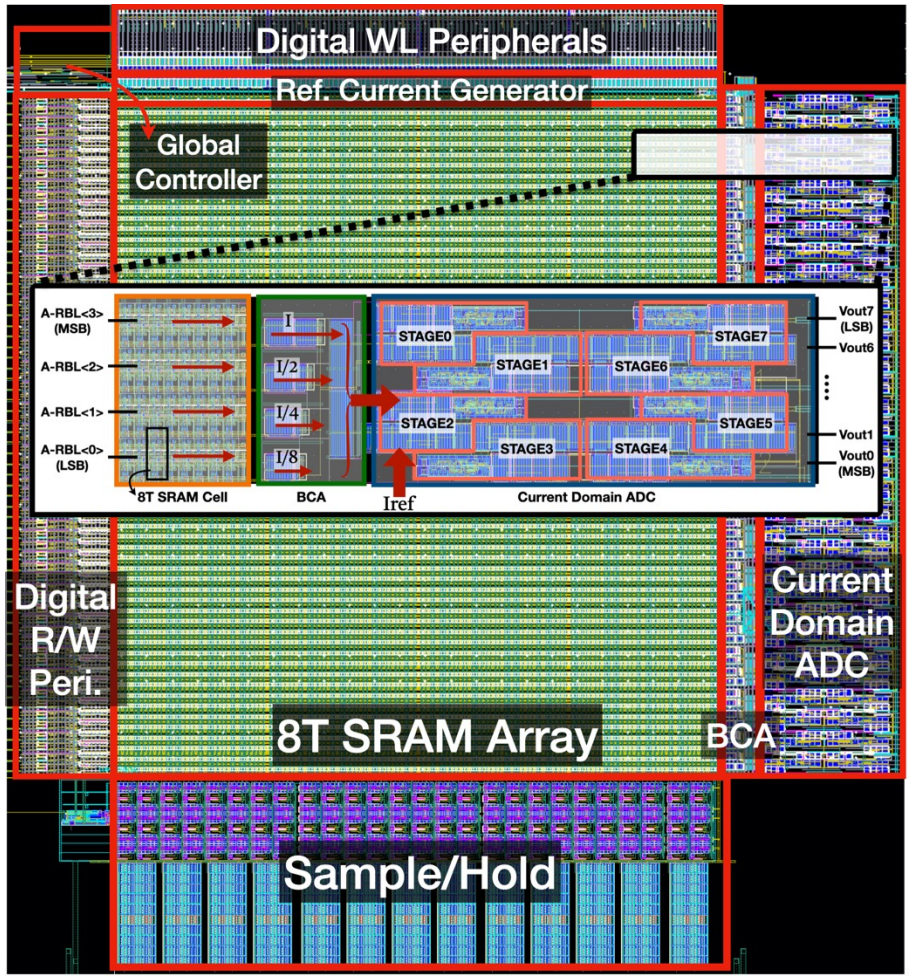
Measurement Results –III



***Trend**: If diff b/w max and min ADC output decreases that frequency determines the throughput.*

- The maximum AFE-CIM throughput is estimated by switching the input between V_{max} & $V_{\min}$ and computing the swing in the ADC output.

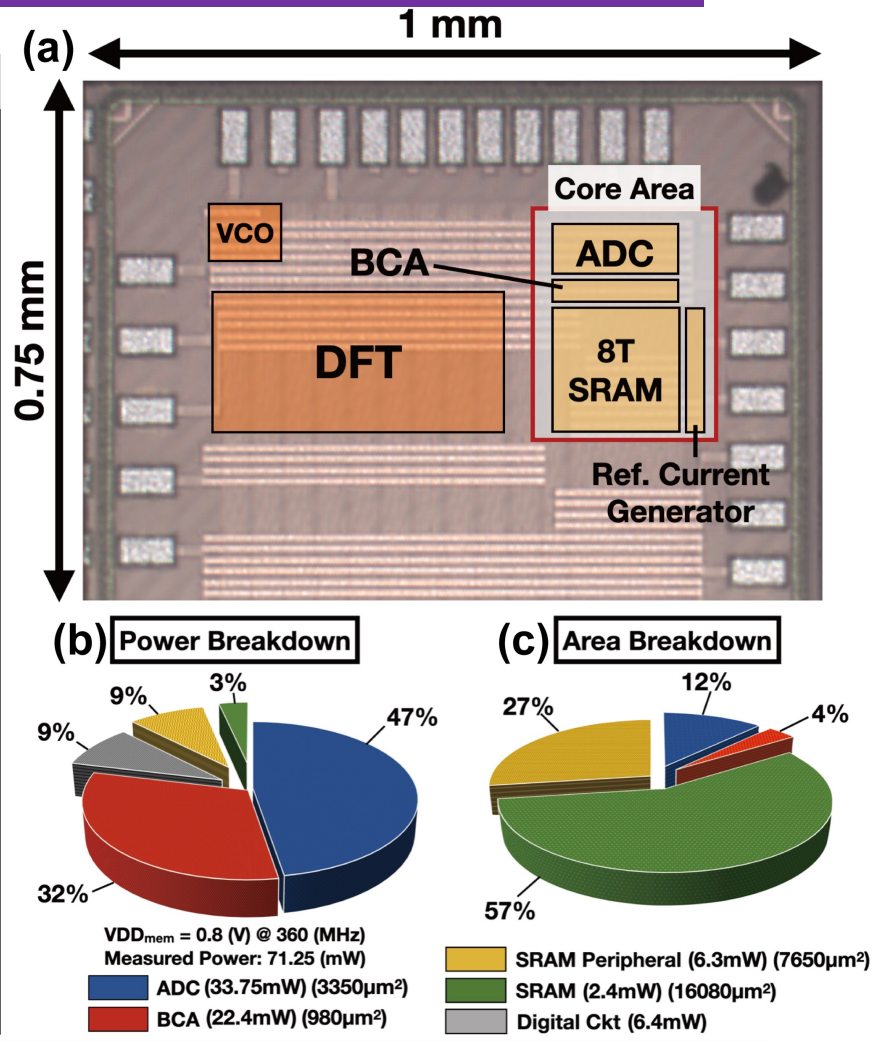
Physical Design & Area Overhead



- The AFE-CIM layout and the pitch-matching of the BCA and ADC across 4 columns of the SRAM.
- Peripheral circuits, including BCA and ADC occupied only 16% of the core area.

Chip Summary

Summary of Chip	
Technology	TSMC 28nm
Chip Area (mm²)	0.028
Core Area	8T-SRAM w/ Current Accum.
Application	Analog Feature Extractor
MAC Operation	Analog
Input Bit-Precision	Analog
Weight Bit-Precision	4
ADC Bit-Precision	8
Total SRAM	8Kb
Supply Voltage (Digital) (V)	0.9
Supply Voltage (Analog) (V)	0.6~0.9
Frequency (MHz)	200~600
Power (mW)	77.18
Performance (Giga sample/sec)	800~2400



- The chip summary, die-photo, area and power breakdown of the AFE-CIM in 28nm CMOS.

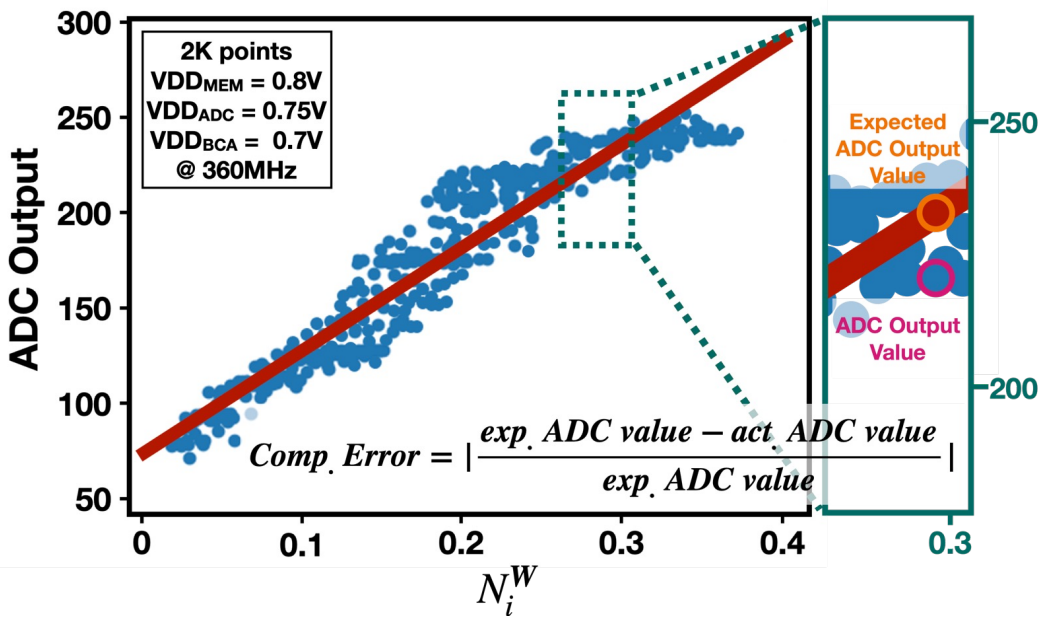
Comparison with Prior Works

	This Work	VLSI' 22 [4]	VLSI' 22 [6]	ISSCC' 21 [7]	TCASI' 21 [8]
Technology	28nm	22nm	12nm	22nm	28nm
Core Area (mm^2)	0.028	0.165 (excludes DAC area)	0.323	0.202	0.05
Cell Type	8T-SRAM w/ Current Accum.	9T-SRAM w/ C-2C Ladder	SRAM	6T-SRAM	Dual-SRAM
On-Chip Memory	8Kb	128Kb	8Kb	64Kb	16Kb
CLK Freq. (MHz)	200 - 600	145 - 240	800	100	214
ADC Precision	8	8	18	16/24	2 5
MAC Opereation	Analog	Analog	Digital	Digital	Analog
Input	Analog	8 bit	4-8 bit	1-8 bit	5 bit
Weight	4 bit	8 bit	4/8 bit	4/8/12/16 bit	2/4/8 bit
Throughput (TOPS)	0.8 - 2.4	0.6 - 1 (8b/8b)	1.343 (4b/4b)	3.3 (4b/4b)	0.125 (5b/8b)
Area Efficiency ($TOPS/mm^2$)	40 - 120	3.64 - 6.1 (8b/8b) (excludes DAC area)	41.58 (4b/4b)	16.33 (4b/4b)	2.455 (5b/8b)
Energy Efficiency (TOPS/W)	14.6 - 43.7	15.5 - 32.3 (8b/8b))	121 (4b/4b)	89 (4b/4b)	147.6 (5b/8b)

*One operation ≈ multiplication of an analog input with a 4-bit weight or an addition.
*All power and throughput are measured for inference operation.

AFE-CIM Application – Simulation Study from Measurement Data

Sim. from Measurement Data

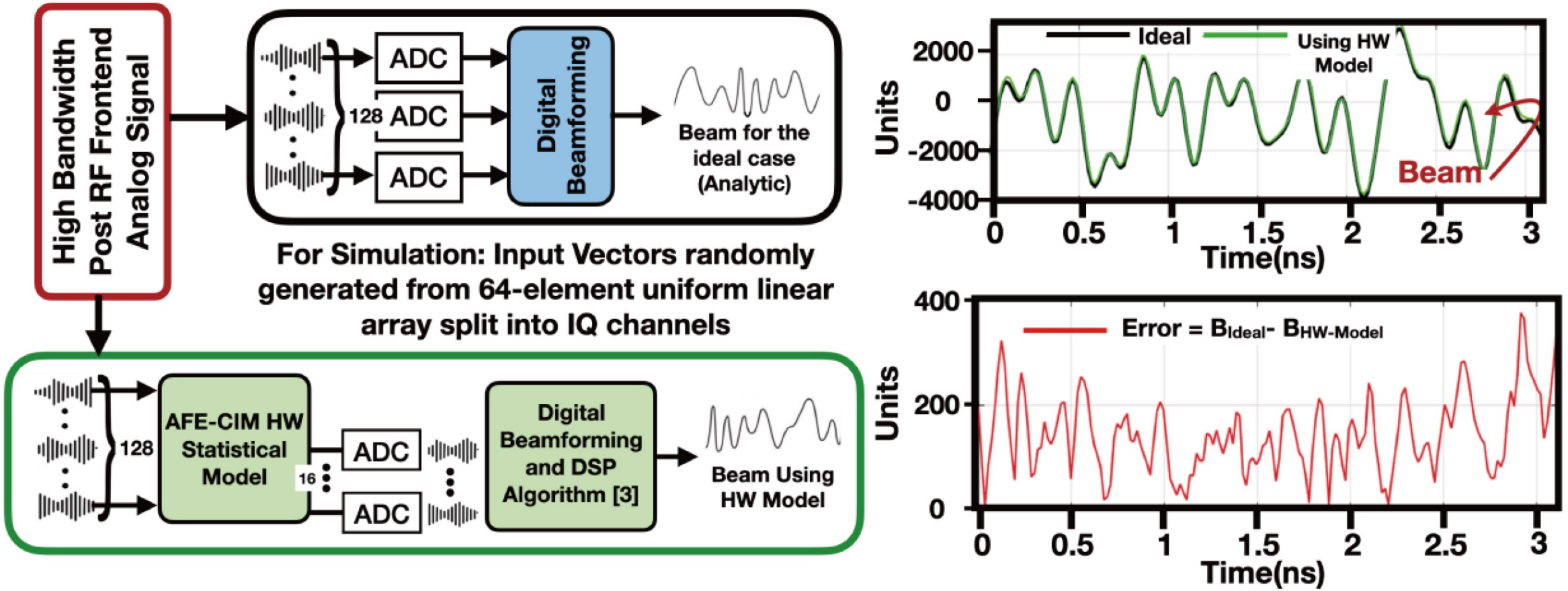


Statistical ADC output referred error model

- 0 – N(μ_0, σ_0)
- 1 – N(μ_1, σ_1)
- ●
- ●
- 255 – N(μ_{255}, σ_{255})

- Use the error model to emulate the non-linearity in the VMM computation in AFE-CIM

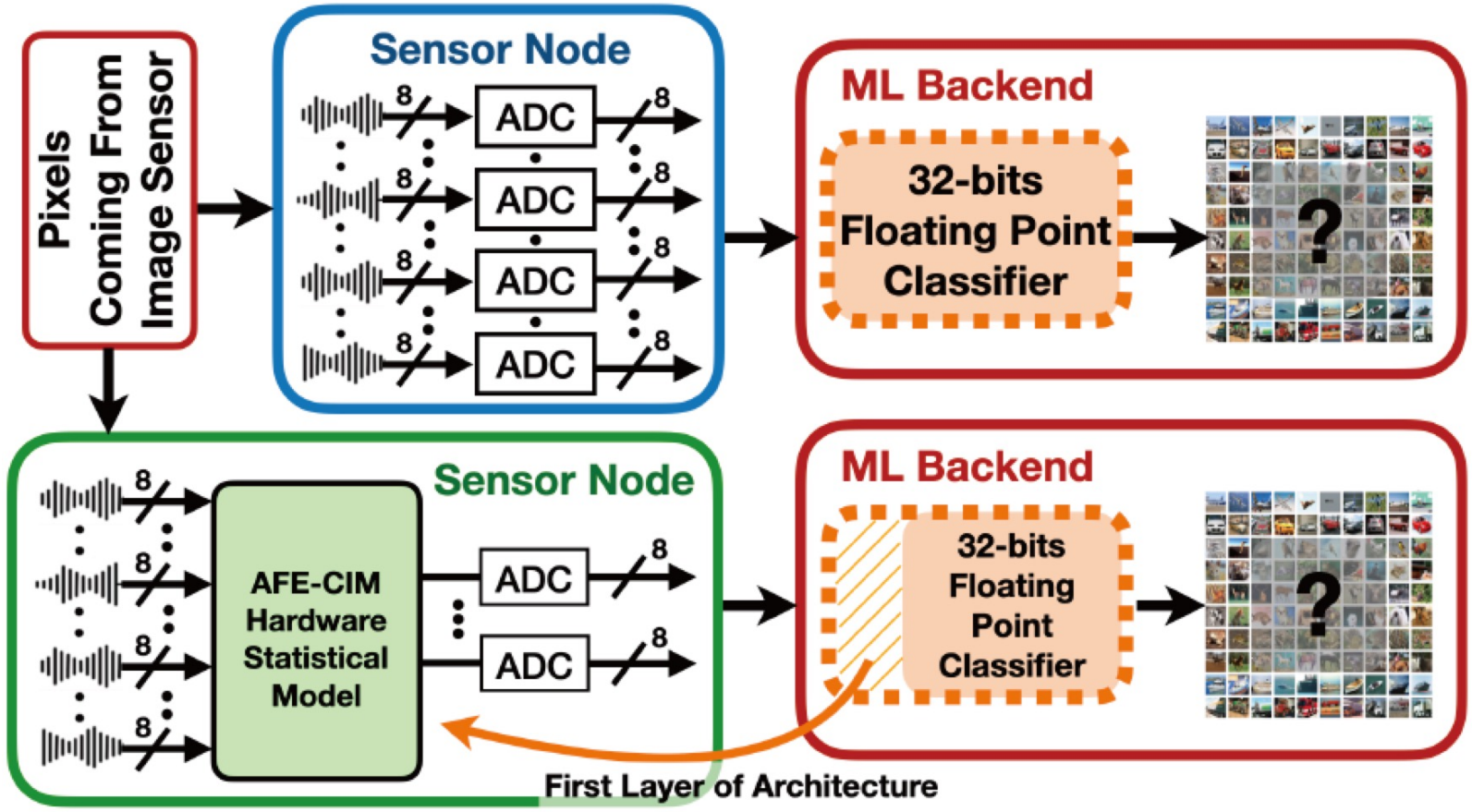
Beamforming using AFE-CIM



***Trend**: Error Lower the better*

[3] C. DeLude et al., ”Broadband Beamforming via Linear Embedding,” arXiv preprint arXiv:2206.07143, 2022.

Image Class. using AFE-CIM



Scenario	Classification Accuracy on Test Data CIFAR-10 (%)
Baseline	88.24
AFE-CIM HW with Noise Aware Training	86.01

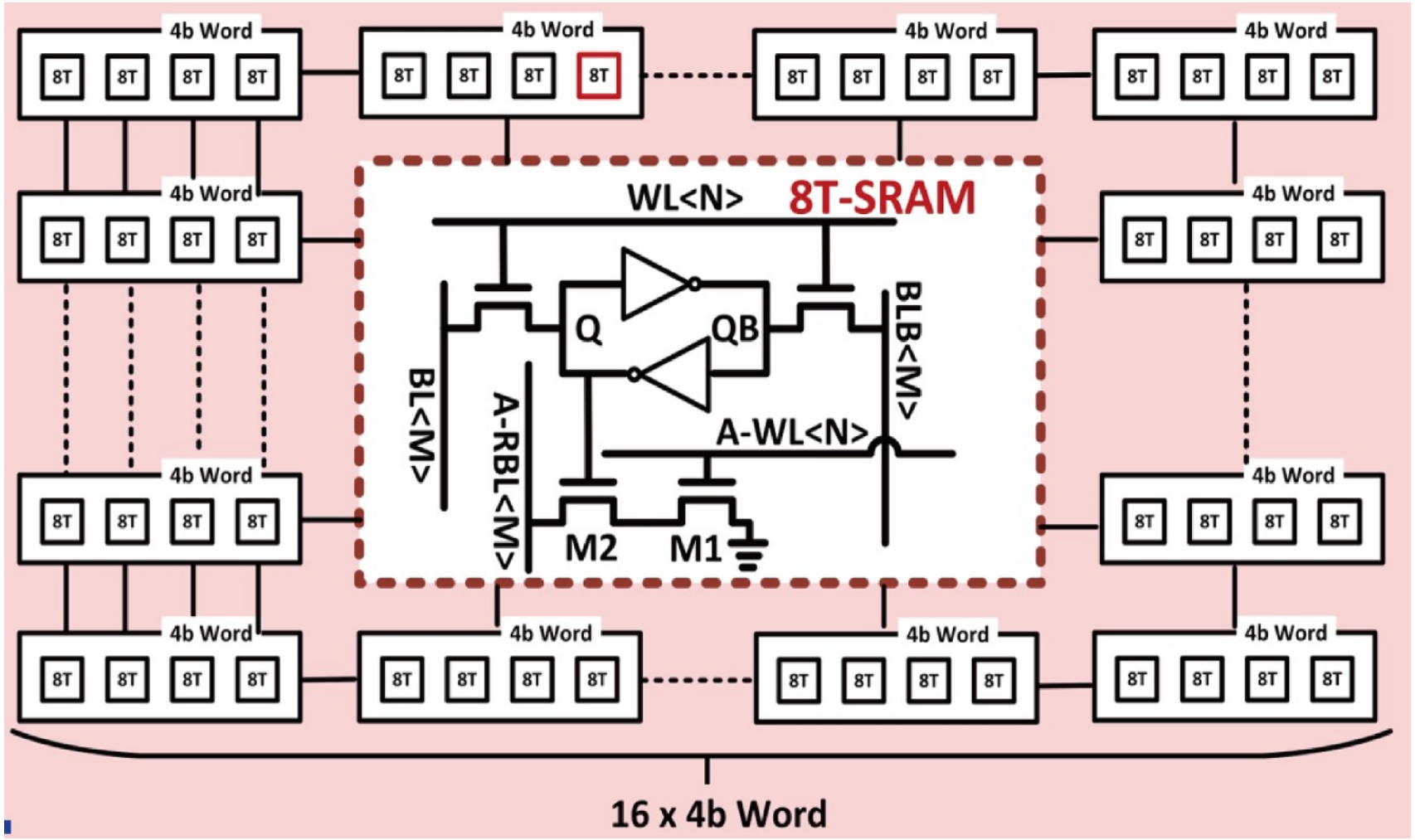
Dataset: CIFAR-10
Architecture: ResNET18

Thank you! Questions?

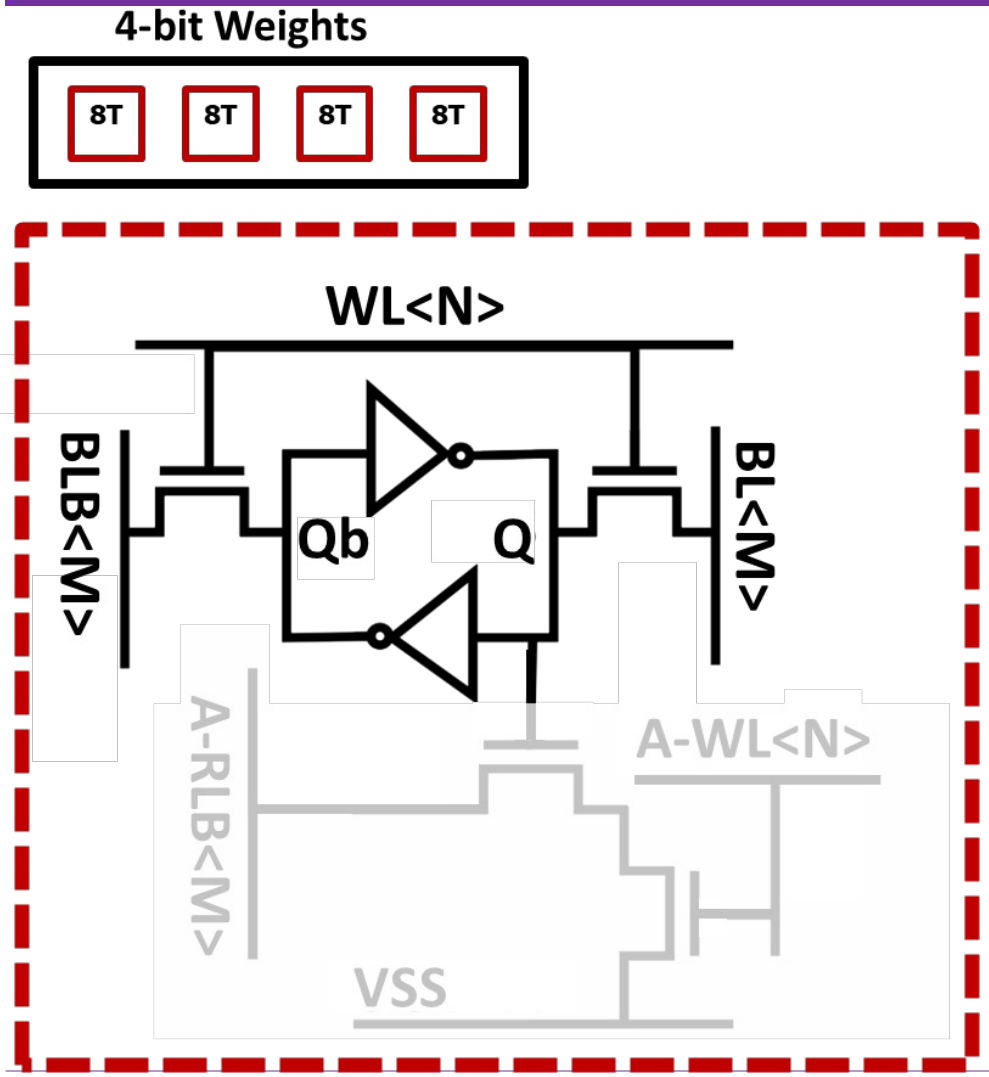
ssharma497@gatech.edu

Backup

Circuit Level Intuition

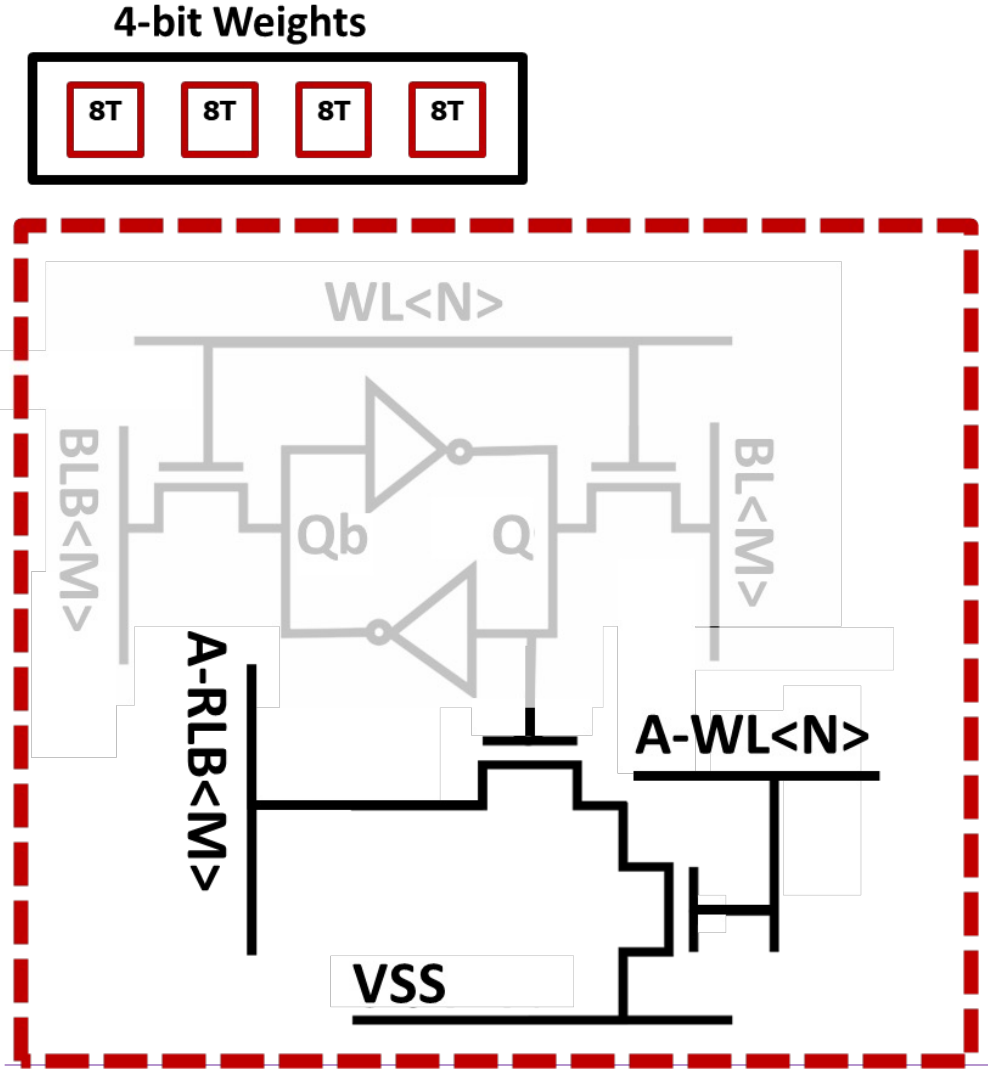


Circuit Level Intuition



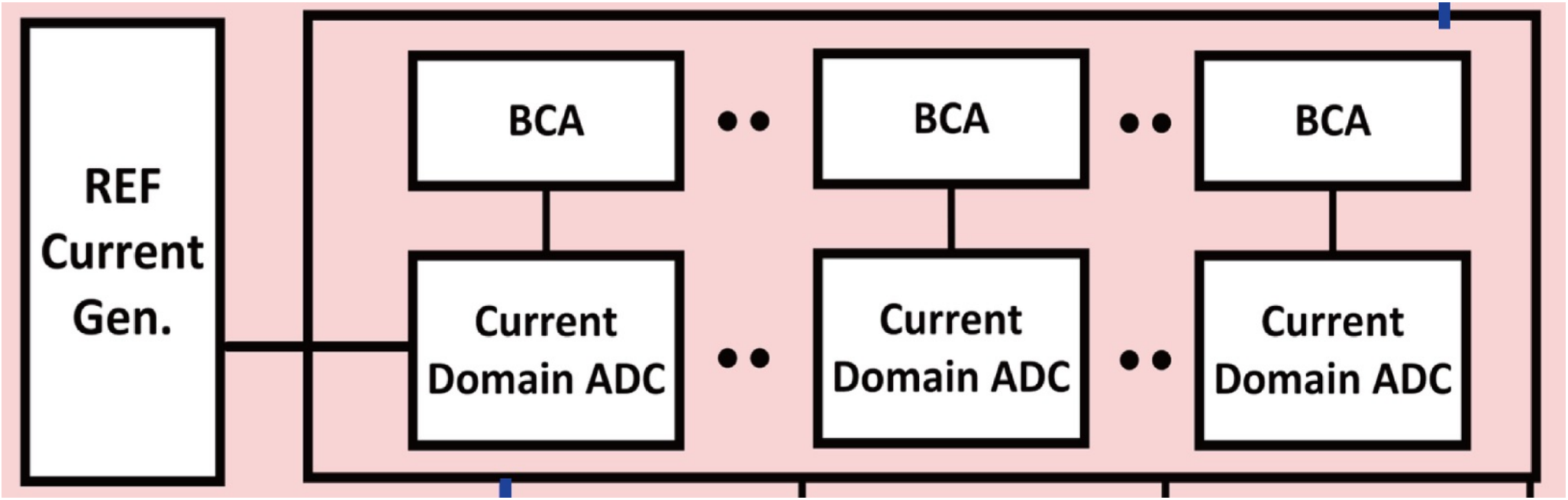
- 108 MHz (measured) operation of the digital bit-line (BL/BLB) and word-line (WL) peripherals enables the real-time update of matrix $\boldsymbol{W}$.

Circuit Level Intuition

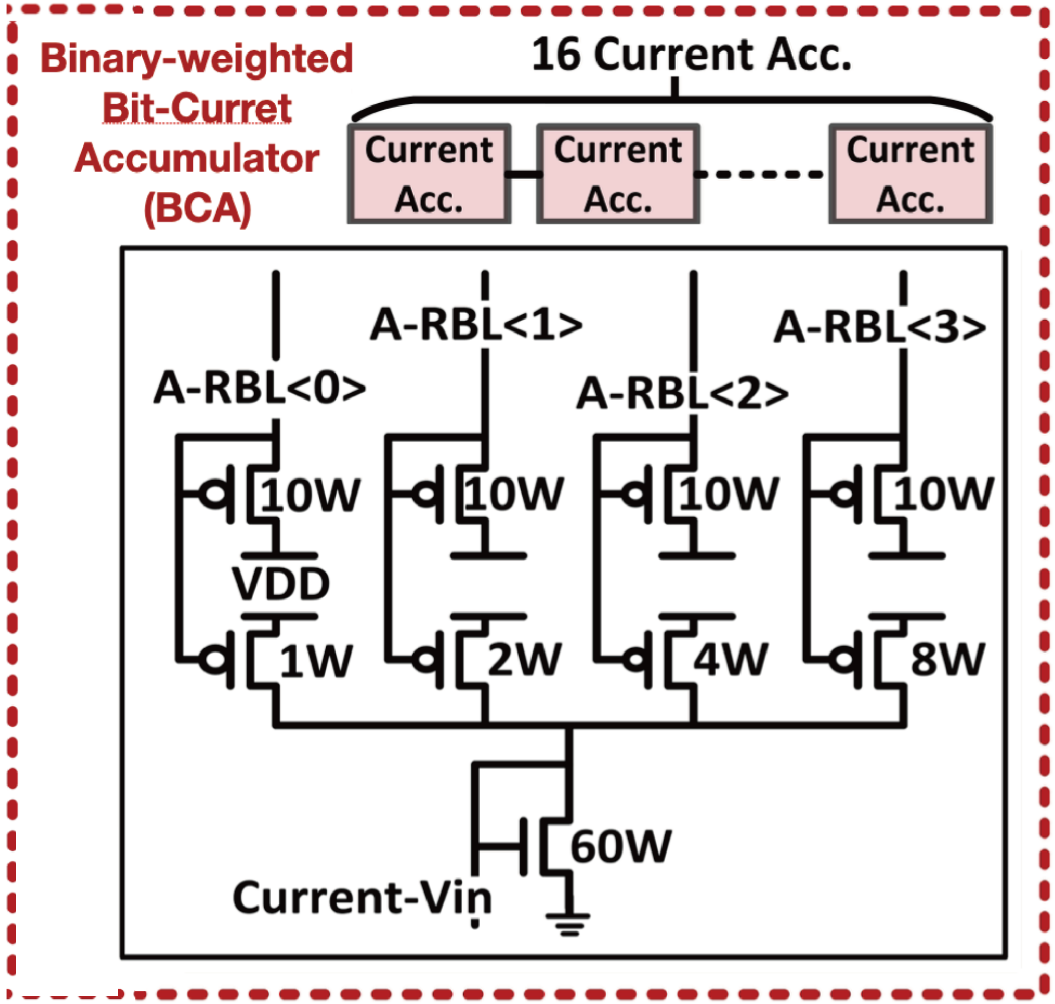


- During A/D-VMM operation, the analog vector v_{in} is applied to the A-WL (analog word-line) of all rows and the 2T-read path of 8T-cell is used for CIM operation.

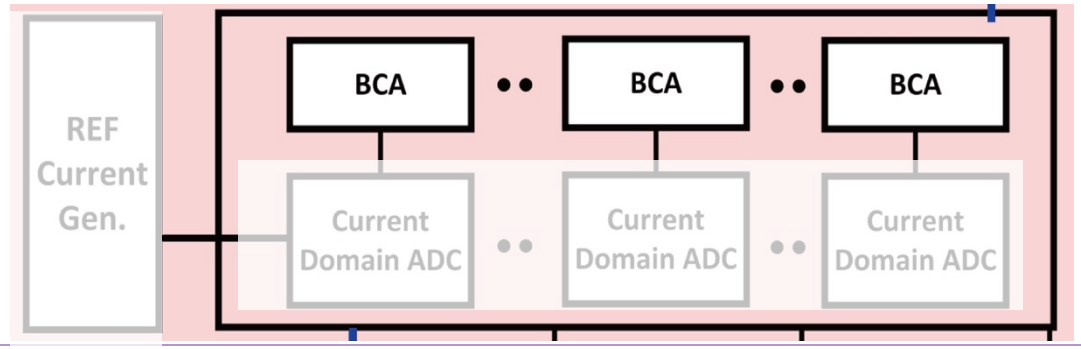
Circuit Level Intuition



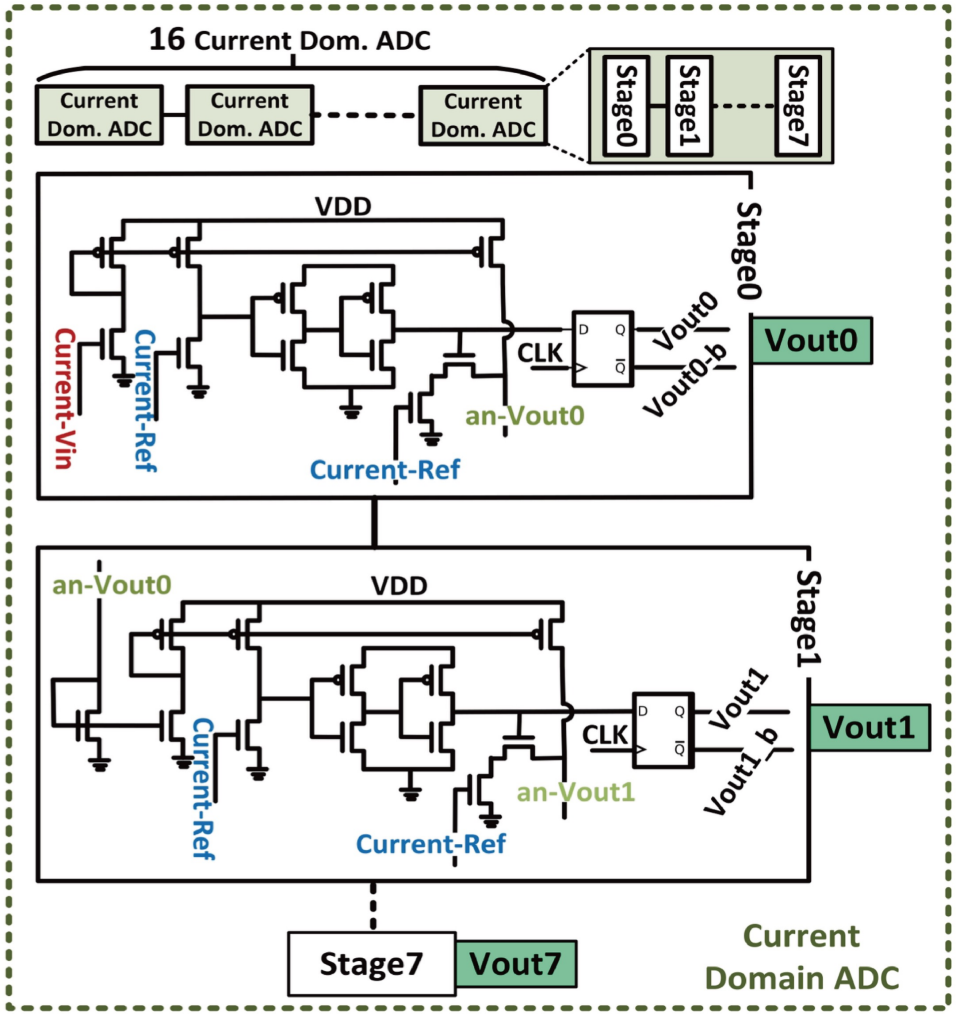
Circuit Level Intuition



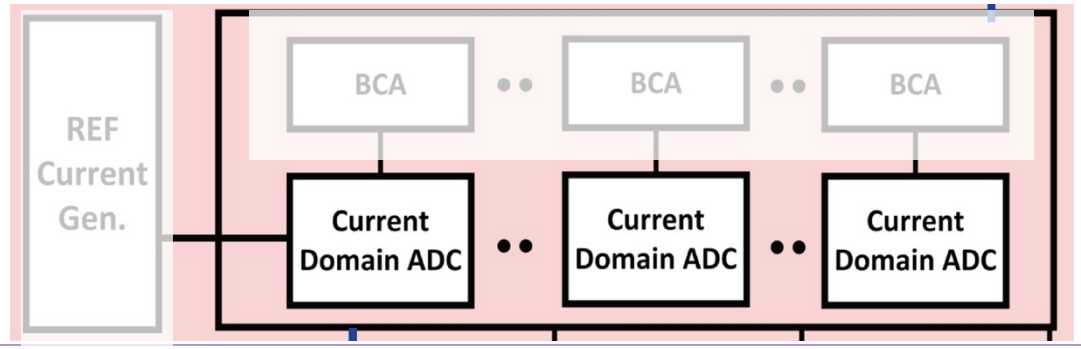
- The read-currents from all cells in an SRAM column are accumulated at the A-RBL. The column currents from 4 A-RBLs are weighted by their bit-significance and accumulated using Binary-weighted Bit-Current Accumulator (BCA).



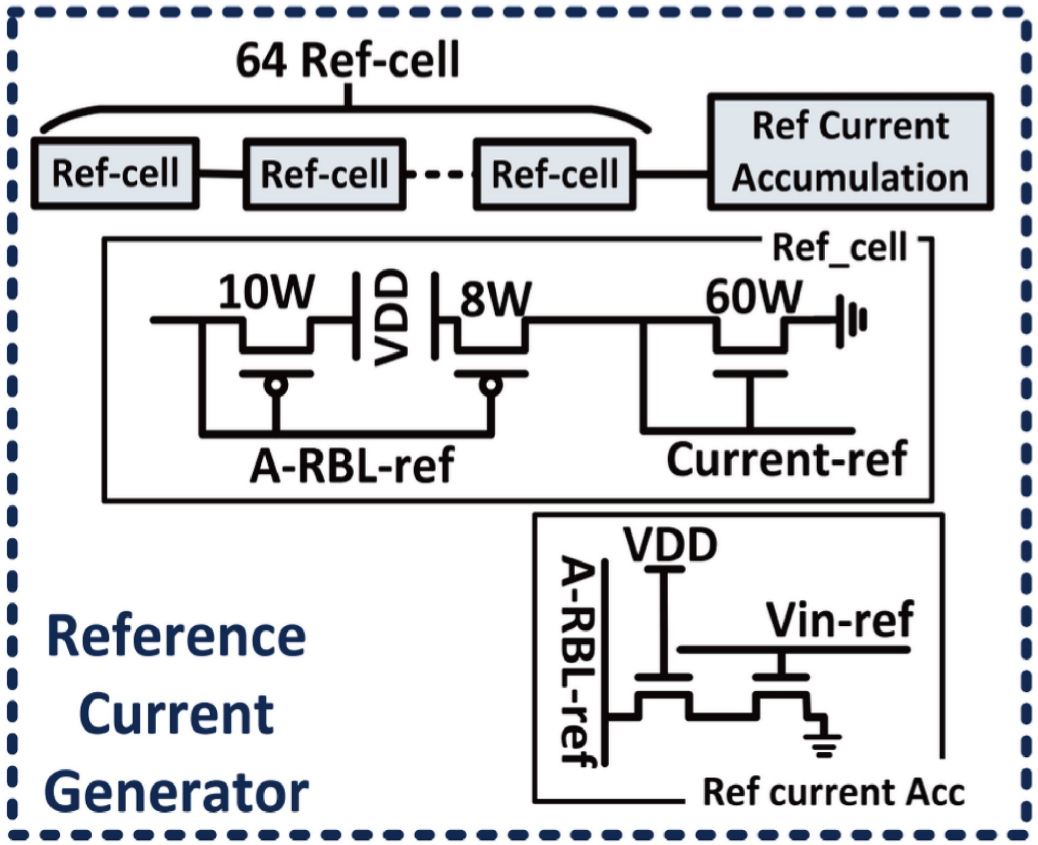
Circuit Level Intuition



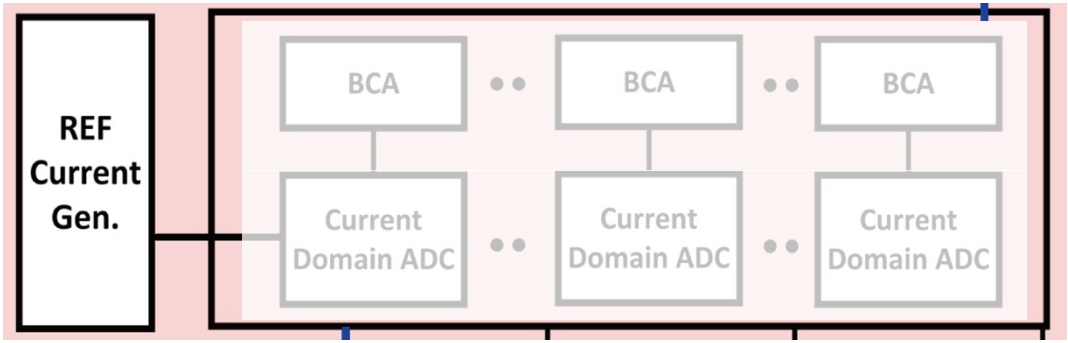
- The output current from the BCA represents one element of the feature vector (i.e., $y_k(t) = \boldsymbol{w_k^T v_{in}(t)}$), which is then digitized using an 8-bit current domain ADC.



Circuit Level Intuition



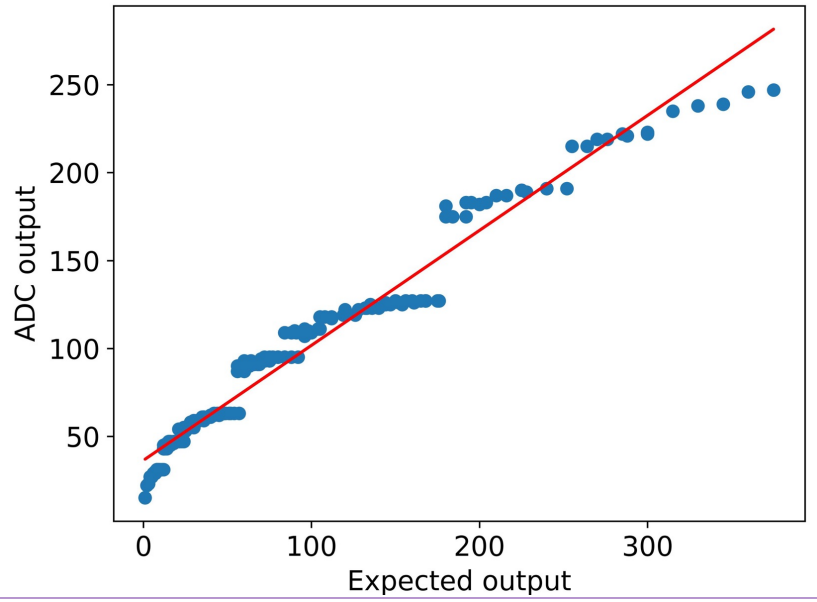
- The reference current is generated using a redundant column of the SRAM array with 64 number of on cells programmed to replicate half of the maximum current.



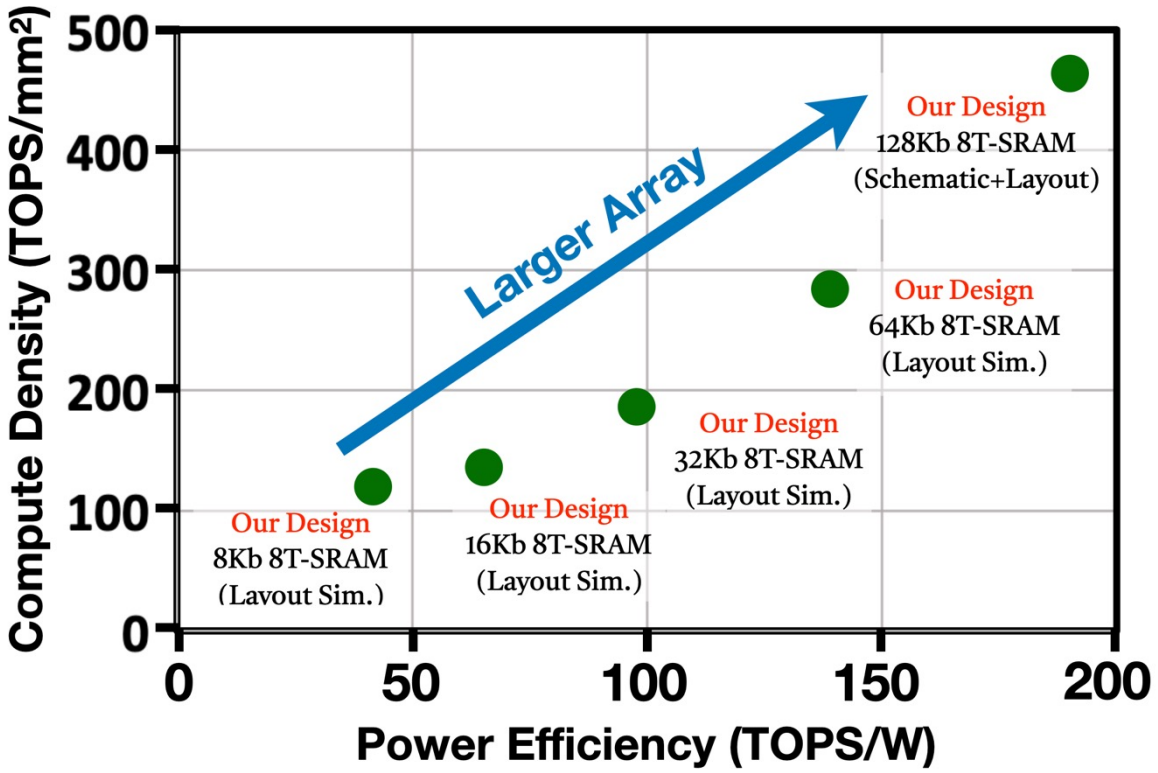
Scalability Simulation

We define linearity error (Δ) as:

$$\Delta = 100 \sum_{n=1}^{n} \frac{|expected\ ADC\ output - actual\ ADC\ output|}{|expected\ ADC\ output|}$$

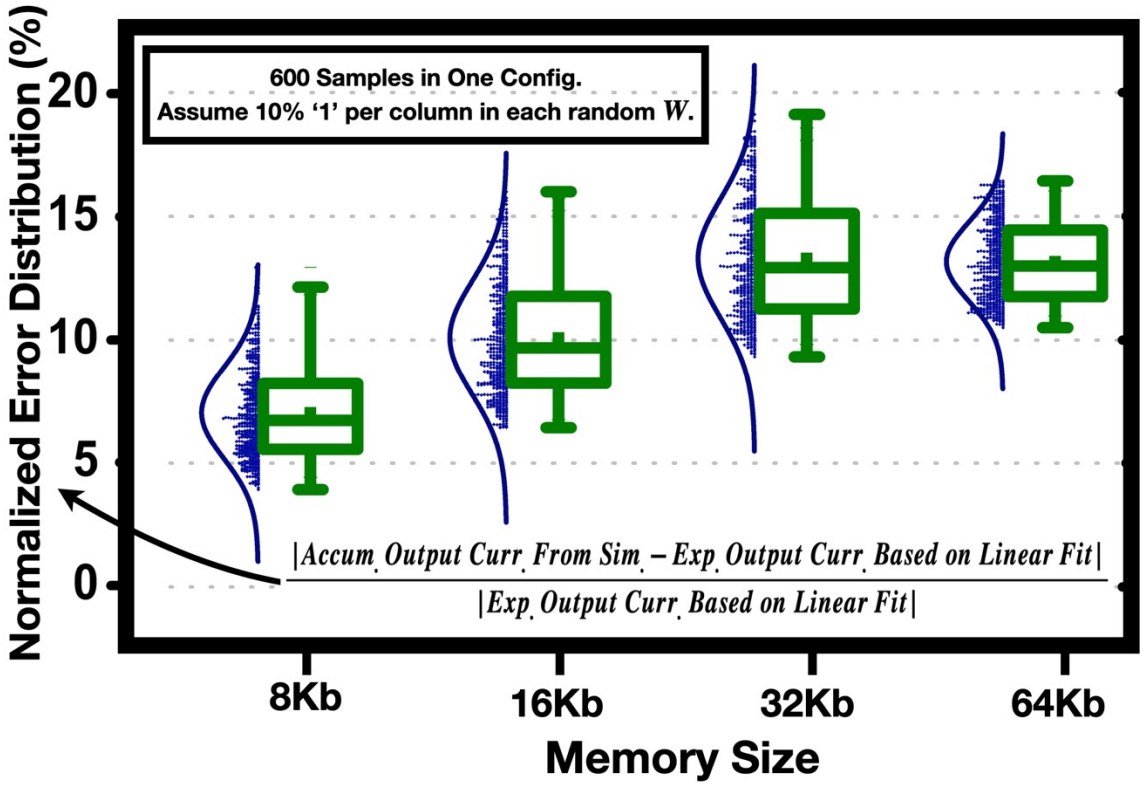


Scalability Simulation



- The power efficiency and compute density are higher for AFE-CIMs with more rows, thanks to higher parallelism in analog computation.

Scalability Simulation



- The normalized error distribution increases with the scaling number of rows due to the non-linearity introduced in the 2T read path and BCA block.

ADC Numerical Example

AN UNUSUAL ELECTRONIC ANALOG-DIGITAL CONVERSION METHOD

Blanchard D. Smith, Jr.
Melpar, Incorporated
Falls Church, Virginia

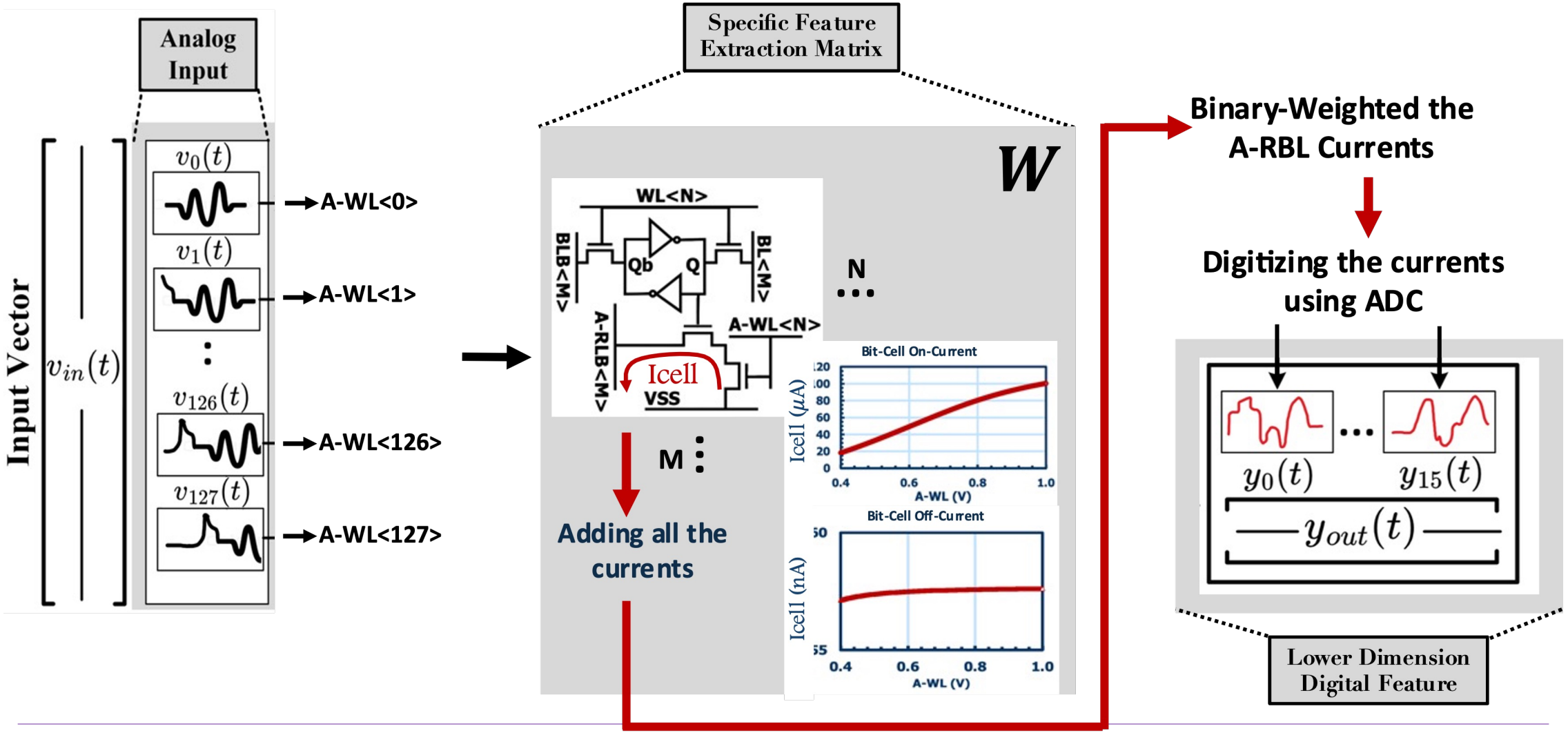
Basic Coding Method

There are a number of methods of converting a d-c voltage or current into digital form, such as counting methods[1], feedback methods[2], and coding tube methods[3]. These methods have been adequately described in the literature and are not covered here. The method presented here has not been given particular attention in the literature, although it was partly covered in a thesis by R.P. Sallen at M.I.T. in 1949. The method discussed in this paper can be seen to derive from one fundamental method of increasing the number of digits from any given coding system, namely by cascading two or more such systems. Figure 1

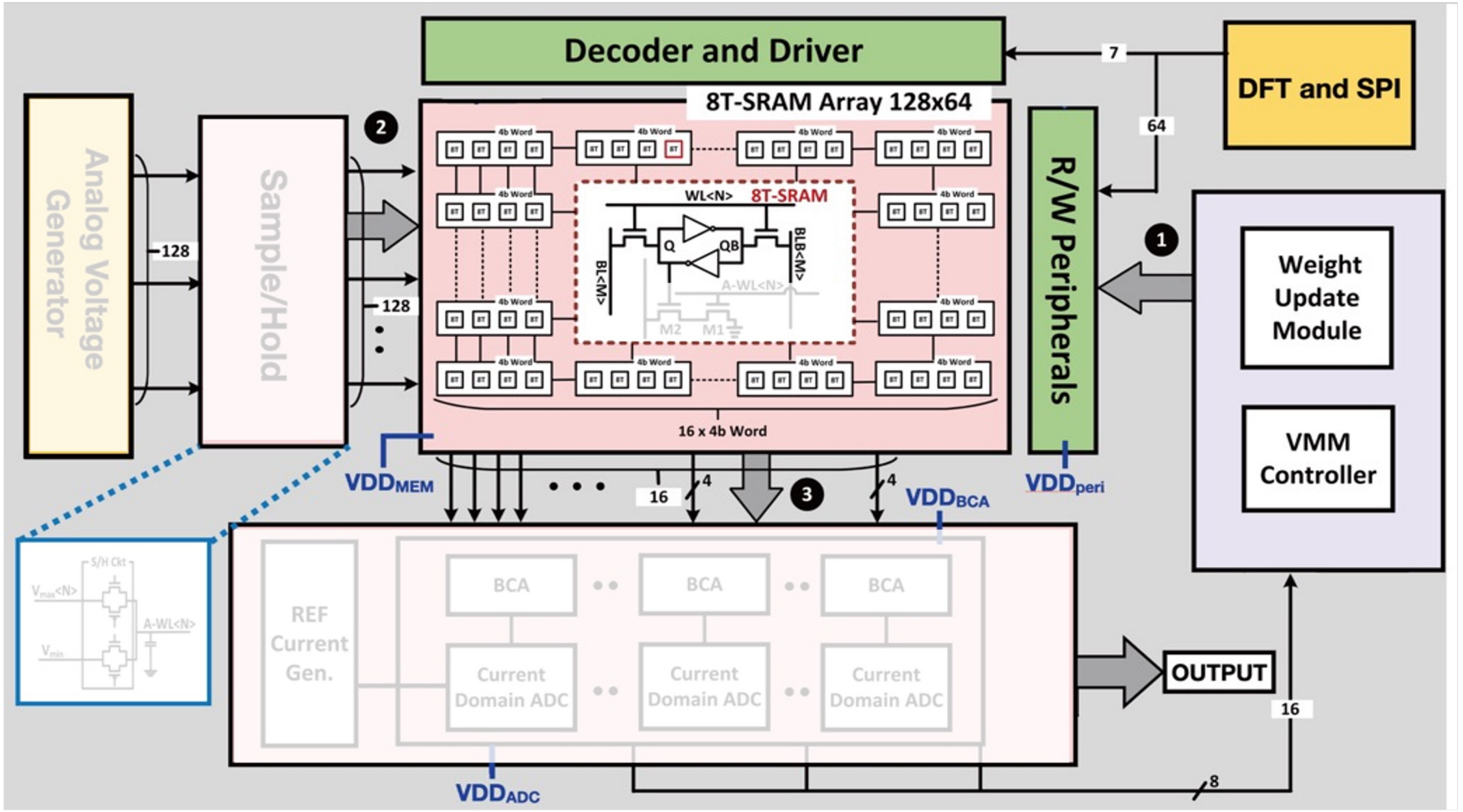
$V_1 = 21.1$	$A = 16$
$V_1 = 21.1$	$D_1 = 1$
$V_2 = 2(21.1 - 16) = 10.2$	$D_2 = 0$
$V_3 = 2(10.2) = 20.4$	$D_3 = 1$
$V_4 = 2(20.4 - 16) = 8.8$	$D_4 = 0$
$V_5 = 2(8.8) = 17.6$	$D_5 = 1$

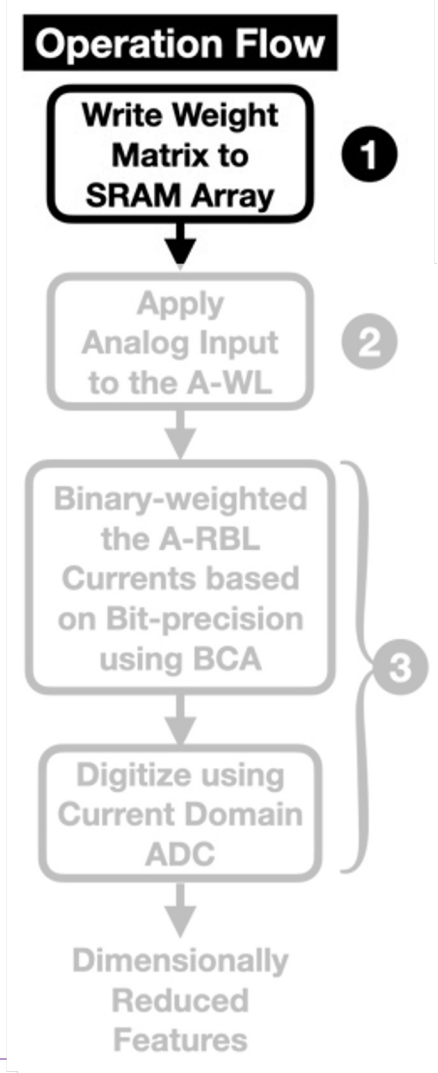
Fig. 4 - Numerical example (binary code).

Approach


Analog Input
Input Vector
$v_{in}(t)$
$v_0(t)$
$v_1(t)$
$v_{126}(t)$
$v_{127}(t)$
A-WL<0>
A-WL<1>
A-WL<126>
A-WL<127>
Specific Feature Extraction Matrix
W
WL<N>
BLB<M>
BL<M>
A-RLB<M>
A-WL<N>
Qb
Q
Icell
VSS
N
M
Adding all the currents
Bit-Cell On-Current
Icell (μA)
A-WL (V)
Bit-Cell Off-Current
Icell (nA)
A-WL (V)
Binary-Weighted the A-RBL Currents
Digitizing the currents using ADC
$y_0(t)$
$y_{15}(t)$
$y_{out}(t)$
Lower Dimension Digital Feature

Architecture





Architecture