Fundamental Limits on Energy-Delay-Accuracy of In-memory Architectures in Inference Applications

Sujan K. Gonugondla Member, IEEE, Charbel Sakr Graduate Student Member, IEEE, Hassan Dbouk Student Member, IEEE, and Naresh R. Shanbhag Fellow, IEEE

Abstract—This paper obtains fundamental limits on the computational precision of in-memory computing architectures (IMCs). An IMC noise model and associated SNR metrics are defined and their interrelationships analyzed to show that the accuracy of IMCs is fundamentally limited by the compute SNR (SNRc) of its analog core, and that activation, weight and output precision needs to be assigned appropriately for the final output SNR SNRf → SNRc. The minimum precision criterion (MPC) is proposed to minimize the ADC precision. Three in-memory compute models - charge summing (QS), current summing (IS) and charge redistribution (QR) - are shown to underlie most known IMCs. Noise, energy and delay expressions for the compute models are developed and employed to derive expressions for the SNR, ADC precision, energy, and latency of IMCs. The compute SNR expressions are validated via Monte Carlo simulations in a 65 nm CMOS process. For a 512 row SRAM array, it is shown that: 1) IMCs have an upper bound on their maximum achievable SNRc due to constraints on energy, area and voltage swing, and this upper bound reduces with technology scaling for QS-based architectures; 2) MPC enables SNRf → SNRc to be realized with minimal ADC precision; 3) QS-based (QR-based) architectures are preferred for low (high) compute SNR scenarios.

I. INTRODUCTION

In-memory computing (IMC) [1]–[4] has emerged as an attractive alternative to conventional von Neumann (digital) architectures for addressing the energy and latency cost of memory accesses in data-centric machine learning workloads. IMCs embed analog mixed-signal computations in close proximity to the bit-cell array in order to execute machine learning computations such as matrix-vector multiply (MVM) and dot products (DPs) as an intrinsic part of the read cycle and thereby avoid the need to access raw data. IMCs exhibit a fundamental trade-off between its energy-delay product (EDP) and the accuracy or signal-to-noise ratio (SNR) of its analog computations. This trade-off arises due to constraints on the maximum bit-line (BL) voltage discharge and due to process variations, specifically spatial variations in the threshold voltage \( V_t \), which limit the dynamic range and the SNR. Additionally, IMCs also exhibit noise due to the quantization of its input activation and weight parameters and due to the column analog-to-digital converters (ADCs).

Charbel Sakr, Hassan Dbouk, and Naresh R. Shanbhag and are with the Department of Electrical and Computer Engineering, at the University of Illinois at Urbana-Champaign. Sujan K. Gonugondla is with Amazon in Seattle, Washington. This work was done during his time at the University of Illinois at Urbana-Champaign.

This paper is organized as follows: Section II presents preliminaries related to signal and DP quantization. Section III proposes the IMC noise model and associated compute SNR metrics. Section IV presents analytical expressions for the compute SNR of three different IMCs. Simulation results quantify various trade-offs including those between energy and accuracy are presented in Section V. Section VI summarizes the key takeaways for designing IMCs.
II. Notation and Preliminaries

A. General Notation

We employ the term signal-to-quantization noise ratio (SQNR) when only quantization noise (denoted as \( q \)) is involved. The term SNR is employed when analog noise sources are included and use \( \eta \) to denote such sources. SNR is also employed when both quantization and analog noise sources are present.

B. The Additive Quantization Noise Model

Under the additive quantization noise model, a floating-point (FL) signal \( x \) quantized to \( B_x \) bits is represented as \( x_q = x + q_x \), where \( q_x \) is the quantization noise assumed to be independent of the signal \( x \).

If \( x \in [x_m, x_m] \) and \( q_x \sim U[-0.5\Delta_x, 0.5\Delta_x] \) where \( \Delta_x = x_m2^{-(B_x-1)} \) is the quantization step size and \( U[a, b] \) denotes the uniform distribution over the interval \([a, b] \), then the signal-to-quantization noise ratio (SQNR) is given by:

\[
\text{SQNR}_{x(x\text{dB})} = 10\log_{10}(\text{SQNR}_x) = 6B_x + 4.78 - \zeta_x(\text{dB})
\]  

(1)

where \( \text{SQNR}_x = \frac{\sigma_x^2}{\sigma_y^2} \), \( \zeta_x(\text{dB}) = 10\log_{10}(\frac{6}{\delta_w}) \) is the peak-to-average (power) ratio (PAR) of \( x \). Equation (1) quantifies the familiar 6 dB SQNR gain per bit of precision.

C. The Dot-Product (DP) Computation

Consider the FL dot product (DP) computation defined as:

\[
y_o = w^T x = \sum_{j=1}^{N} w_j x_j
\]  

(2)

where \( y_o \) is the DP of two \( N \)-dimensional real-valued vectors \( w = [w_1, \ldots, w_N]^T \) (weight vector) and \( x = [x_1, \ldots, x_N]^T \) (activation vector).

In DNNs, the dot product in (2) is computed with \( w \in [-w_m, w_m] \) (signed weights), input \( x \in [0, x_m] \) (unsigned activations assuming the use of ReLU activation functions) and output \( y \in [-y_m, y_m] \) (signed outputs). Assuming the additive quantization noise model from Section II-B, the fixed-point (FX) computation of the DP (2) with precisions \( B_w \) (weight), \( B_x \) (activation), and \( B_y \) (output), is described by:

\[
y = w_y^T x_q + q_y = (w + q_w)^T (x + q_x) + q_y
\]  

(3)

\[
y \approx w^T x + w^T q_x + q_w^T x + q_y = y_o + q_{iy} + q_y
\]  

(4)

where \( w_q = w + q_w \) and \( x_q = x + q_x \) are the quantized weight and activation vectors, respectively, \( q_{iy} \) is the total input (weight and activation) quantization noise seen at the output \( y \) (output referred quantization noise), and \( q_y \) is the additional output quantization noise due to round-off/truncation in digital architectures or from the finite resolution of the column ADCs in IMC architectures.

Assuming that the weights (signed) and inputs (unsigned) are i.i.d. random variables (RVs), the variances of signals in (4) are given by:

\[
\sigma_w^2 = N\sigma_w^2 \mathbb{E}[x^2]; \sigma_y^2 = \frac{\Delta_x^2}{12}; \sigma_{q_y}^2 = \frac{N}{12} \left( \Delta_w^2 \mathbb{E}[x^2] + \Delta_x^2 \sigma_w^2 \right)
\]  

(5)

where \( \sigma_w^2 \) is the variance of the weights, \( \Delta_w = w_m2^{-(B_w-1)} \), \( \Delta_x = x_m2^{-(B_x-1)} \) and \( \Delta_y = y_m2^{-(B_y-1)} \) are the weight, activation, and output quantization step-sizes, respectively.

III. Compute SNR Limits of IMCs

We propose the system noise model in Fig. 1 for obtaining precision limits on IMCs. Such architectures (Fig. 1(a)) accept a quantized input (\( x_q \)) and a quantized weight vector (\( w_q \)) to implement multiple FX DP computations of (2) in parallel in its analog core. Hence, unlike digital architectures, IMC architectures suffer from both quantization and analog noise sources such as SRAM cell current variations, thermal noise, and charge injection, as well as the limited headroom, which limits its compute SNR.

A. Compute SNR Metrics for IMCs

The following equations describe the IMC noise model in Fig. 1:

\[
y = y_o + q_{iy} + \eta_a + q_y; \quad \eta_a = \eta_k + \eta_h
\]  

(6)

where \( \eta_a \) is the ideal DP value defined in (2), \( q_{iy} \) is the output referred quantization noise, \( \eta_k \) is the analog noise term comprising both clipping noise \( \eta_h \) due to limited headroom and other noise sources \( \eta_k \), and \( q_y \) is the output quantization noise introduced by the ADC.

We define the following fundamental compute SNR metrics:

\[
\text{SNQR}_{q_{iy}} = \frac{\sigma_y^2}{\sigma_{q_y}^2}; \text{SNR}_x = \frac{\sigma^2}{\sigma_{iy}^2}; \text{SNQR}_{\eta_q} = \frac{\sigma_y^2}{\sigma_{\eta_q}^2}
\]  

(7)

where \( \text{SNR}_x \) is the analog SNR, \( \text{SNQR}_{\eta_q} \) is the output referred SQNR due to input (weight and activation) quantization and is given by:

\[
\text{SNQR}_{q_{iy}}(\text{dB}) = 6(B_x + B_w) + 4.8 - \left[ \zeta_x(\text{dB}) + \zeta_w(\text{dB}) \right]
\]  

(8)

where \( \zeta_x(\text{dB}) = 10\log_{10}(\frac{x_m}{\delta_x}) \) and \( \zeta_w(\text{dB}) = 10\log_{10}(\frac{x_m}{\delta_w}) \) are the PARs of the (unsigned) activations and (signed) weights, respectively, and \( \text{SNQR}_{q_{iy}} \) is the digitization SQNR solely due to ADC quantization noise and is given by:

\[
\text{SNQR}_{\eta_q}(\text{dB}) = 6B_y + 4.8 - \left[ \zeta_{\eta_q}(\text{dB}) + \zeta_w(\text{dB}) \right] - 10\log_{10}(N)
\]  

(9)

obtained by the substitutions: \( B_x \leftarrow B_y \) and \( \zeta_{\eta_q}(\text{dB}) \leftarrow \zeta_{\eta_q}(\text{dB}) = \zeta_{\eta_q}(\text{dB}) + \zeta_w(\text{dB}) + 10\log_{10}(N) \) in (1).
From (6) and (7), it is straightforward to show:

$$\text{SNR}_A = \frac{\sigma_{y}^2}{\sigma_{q_y}^2 + \sigma_{b}^2} = \left[ \frac{1}{\text{SNR}_A} + \frac{1}{\text{SQNR}_{q_y}} \right]^{-1}$$  \hspace{1cm} (10)

$$\text{SNR}_T = \frac{\sigma_{y}^2}{\sigma_{q_y}^2 + \sigma_{a}^2 + \sigma_{\eta}^2} = \left[ \frac{1}{\text{SNR}_T} + \frac{1}{\text{SQNR}_{q_y}} \right]^{-1}$$  \hspace{1cm} (11)

where SNR$_A$ is the pre-ADC SNR and SNR$_T$ is the total output SNR including all noise sources. Note: (10)-(11) can be repurposed for digital architectures by setting SNR$_A \rightarrow \infty$ since quantization is the only noise source thereby implying SNR$_A = \text{SQNR}_{q_y}$. Equations (8)-(9) indicate that SQNR$_{q_y}$ and SQNR$_{a}$ can be made arbitrarily large by assigning sufficiently high precision to the DP inputs ($B_x$ and $B_w$) and the output ($B_y$). Thus, from (10)-(11), SNR$_T$ in IMCs is fundamentally limited by SQNR$_y$ which depends on the analog noise sources as one expects.

### B. Maximizing SNR$_T$ in IMCs

Prior work based on post-training quantization [30], [31] indicates the requirement SNR$_{t(db)} > \text{SNR}_T(\text{db}) = 10\text{ dB}-40\text{ dB}$ (see Fig. 2) for the inference accuracy of an FX network to be within $1\%$ of the corresponding FL network for popular DNNs (AlexNet, VGG-9, VGG-16, ResNet-18) deployed on the ImageNet and CIFAR-10 datasets. While in-training quantization methods [32] can reduce these requirements, a precision of 4-b ($\sim 24\text{ dB}$) is generally found to be [33] sufficient. To meet this SNR$_{t(db)}$ requirement, digital architectures choose $B_x$ and $B_w$ such that SQNR$_{q_y} > \text{SNR}_T$, and then choose $B_y$ sufficiently high to guarantee SQNR$_{q_y} \gg \text{SQNR}_{y}$. So that $\text{SNR}_T \rightarrow \text{SQNR}_{y}$.

In contrast, for IMCs, we first need to ensure that SNR$_y > \text{SNR}_T^*$ so that $\text{SNR}_T$ can be made to approach SNR$_y$ with appropriate precision assignment. Such a precision assignment can be easily derived from (10)-(11) as shown below:

1) Assign sufficiently high values for $B_x$ and $B_w$ per (8) such that SQNR$_{q_y} \gg \text{SNR}_a$ so that $\text{SNR}_A \rightarrow \text{SNR}_a$ per (10).

2) Assign sufficiently high value for $B_y$ such that SQNR$_{q_y} \gg \text{SNR}_a$ so that $\text{SNR}_T \rightarrow \text{SNR}_a$ per (11).

For example, if SQNR$_{q_y}(\text{db})$, SQNR$_{q_y}(\text{db}) \geq \text{SNR}_a(\text{db}) + 9\text{ dB}$ then SNR$_a - \text{SNR}_T(\text{db}) \leq 0.5\text{ dB}$, i.e., SNR$_{t(db)}$ lies within 0.5 dB of SNR$_a (\text{db})$. In this manner, by appropriate choices for $B_x$, $B_w$, and $B_y$, IMCs can be designed such that $\text{SNR}_T \rightarrow \text{SNR}_a$, which, as mentioned earlier, is the fundamental limit on SNR$_T$.

From the above discussion it is clear that the input precisions $B_x$ and $B_w$ are dictated by network accuracy requirements, while the output precision $B_y$ needs to be set sufficiently high for the output quantization from becoming a significant noise contributor. To ensure that a sufficiently high value for $B_y$, digital architectures employ the bit growth criterion (BGC) described next.

### C. Bit Growth Criterion (BGC)

The BGC is commonly employed to assign the output precision $B_y$ in digital architectures [30], [34]. BGC sets $B_y$ as:

$$B_y^\text{BGC} = B_x + B_w + \log_2(N)$$  \hspace{1cm} (12)

Substituting $B_y = B_y^\text{BGC}$ from (12) into (9) and employing the relationship $\zeta_y(\text{dB}) = 10\log_{10}(N) + \zeta_x(\text{db}) + \zeta_w(\text{db})$, the resulting SQNR due to output quantization using the BGC is given by:

$$\text{SQNR}^\text{BGC}_y(\text{db}) = 10\log_{10}\left(\frac{\sigma_y^2}{\sigma_{q_y}^2}\right) = 6(B_x + B_w) + 4.8 - [\zeta_x(\text{db}) + \zeta_w(\text{db})] + 10\log_{10}(N).$$  \hspace{1cm} (13)

Recall that SQNR$^\text{BGC}_y \gg$ SNR$_A$ in order to ensure SNR$_T$ is close to its upper bound. Comparing (9) and (13), we see that, for high values of DP dimensionality $N$, BGC is overly conservative since it assigns large values to $B_y$ per (12). Some digital architectures truncate the LSBs to control bit growth. The SQNR of such truncated BGC (tBGC) can be obtained directly from (9) by setting the value of $B_y < B_y^\text{BGC}$.

BGC’s high precision requirements is accommodated in digital architectures by increasing the precision of arithmetic units with a commensurate increase in the computational energy, latency, and activation storage costs. However, IMCs cannot afford to use this criterion since $B_y$ is the precision of the BL ADCs which impacts its energy, latency, and area. Indeed, recent works [35] have claimed that BL ADCs dominate the energy and latency costs of IMCs. Such works employ the highly conservative BGC or tBGC to assign $B_y$.

In the next section, we propose an alternative to BGC and tBGC referred to the minimum precision criterion (MPC), that can be employed by both digital and IMC architectures to achieve a desired SQNR$_y$ with much smaller values of $B_y$.

### D. The Minimum Precision Criterion (MPC)

We propose MPC to reduce $B_y$ without incurring any loss in SQNR$_y$ compared to BGC. Unlike BGC, MPC accounts for the statistics of $y_y$ to permit controlled amounts of clipping to occur. In MPC (see Fig. 3(a)), the output $y_y$ is clipped to lie in the range $[-y_c; y_c]$ instead of $[-y_m; y_m]$ as in BGC (see Fig. 3(b)), where $y_c < y_m$ is the clipping level, and $B_y$ bits are employed to quantize this reduced range. The clipping

---

**Fig. 2.** Per-layer SNR$_{t(db)}$ requirements of DP computations in VGG-16 deployed on ImageNet.
probability $p_c = \Pr \{ |y_o| > y_c \}$ is kept to a small user-defined value, e.g., $y_c = 4\sigma_y$ ensures that $p_c < 0.001$ if $y_o \sim \mathcal{N}(0, \sigma_y^2)$. The resulting SQNR$_{q_y}$ is given by:

$$\text{SQNR}_{q_y(\text{dB})} = 6B_y + 4.8 - \gamma_{\text{MPC}} - 10 \log_{10}\left(1 + p_c \frac{\sigma_c^2}{\sigma_q^2}\right)$$

(14)

where $\gamma_{\text{MPC}} = 10 \log_{10}\left(\frac{\sigma_o^2}{\sigma_y^2}\right)$, and $\sigma_y^2 = \mathbb{E}[|y_o - y_c|^2 | |y_o| > y_c]$ is the conditional clipping noise variance. Setting $y_c = \gamma_{q_y(\text{MPC})}$ yields $\gamma_{q_y(\text{MPC})} = 10 \log_{10}(\gamma_{\text{MPC}})^2$ indicating that $p_c$ is a decreasing function of $\gamma_{q_y(\text{MPC})}$. Thus, (14) has the same form as (1) with an additional (last term) clipping noise factor.

MPC exploits a key insight (see Fig. 3(c)), which follows from the Central Limit Theorem (CLT) – in a $N$-dimensional DP computation, $\sigma_y$ grows sub-linearly (as $\sqrt{N}$) as compared to the maximum $y_m$ which grows linearly with $N$. Furthermore, (14) shows a quantization vs. clipping noise trade-off controlled by the clipping level $y_c$. We show empirically in Section III-E that SQNR$_{q_y(\text{dB})}$ in (14) is maximized when clipping level $y_c = 4\sigma_y$, if $y_o \sim \mathcal{N}(0, \sigma_y^2)$. This trade-off (see Fig. 3(b)) is absent in BGC and tBGC, and is critical to MPC's ability to realize desired values of SQNR$_{q_y}$ with smaller values of $B_y$.

Thus, we state the following MPC-based rule for maximizing the SQNR of column ADCs in IMCs:

**MPC-based SQNR Maximizing Rule**

For a Gaussian signal, setting the clipping level to four times the standard deviation will maximize the SQNR for a given precision $B_y$.

Some IMC designs do in fact allow for clipping in the column ADCs but these levels are set empirically. The MPC-based Rule in contrast quantitatively specifies the smallest ADC precision and the optimal clipping level needed to ensure that SQNR$_{q_y} \geq$ SNR$_A$. A lower bound on $B_y$ can be obtained by assuming $y_o \sim \mathcal{N}(0, \sigma_y^2)$, and substituting $y_c = 4\sigma_y$, and $p_c = 0.001$ into (14), to obtain:

$$B_y^{\text{MPC}} \geq \frac{1}{6} \left[ \text{SNR}_{A(\text{dB})} + 7.2 - \gamma - 10 \log_{10}\left(1 - 10^{-\frac{7.2}{10}}\right) \right]$$

(15)

in order for SNR$_{A(\text{dB})} - \text{SNR}_{T(\text{dB})} \leq \gamma$. For instance, the choice $\gamma = 0.5$ dB yields $B_y^{\text{MPC}} \geq \frac{1}{6} [\text{SNR}_{A(\text{dB})} + 16.3]$ which corresponds to SQNR$_{q_y(\text{dB})} \geq$ SNR$_{A(\text{dB})} + 9$ dB as discussed in Section III-B.

**E. Simulation Results**

To illustrate the difference between MPC, BGC and tBGC, we assume that SNR$_{A(\text{dB})} \geq 31$ dB, so that SNR$_{T(\text{dB})} \geq 30$ dB provided SQNR$_{q_y(\text{dB})} \geq]$ SNR$_{A(\text{dB})} + 40$ dB per (10)-(11).

We further assume DPs of varying dimension $N$ with 7-b quantized unsigned inputs and signed weights randomly sampled from uniform distributions. Substituting $B_x = B_w = 7$, $\zeta_c(\text{dB}) = -1.3$ dB, and $\zeta_w(\text{dB}) = 4.8$ dB into (14), we obtain SQNR$_{q_y(\text{dB})} \geq 41$ dB. Thus, all that remains is to assign $B_y$ such that SQNR$_{q_y(\text{dB})} \geq 40$ dB, for which there are three choices - MPC, BGC and tBGC.

Figure 4(a) compares the SQNR$_{q_y}$ achieved by the three methods. Per (15), MPC meets the SQNR$_{q_y(\text{dB})} \geq 40$ dB requirement by setting $B_y = 8$ and $\gamma_{q_y(\text{dB})} = 4$ independent of $N$. In contrast, per (12), BGC assigns $16 \leq B_y \leq 20$ as a monotonically increasing function of $N$ to achieve the same SNR$_T$ as MPC. Furthermore, tBGC meets the SQNR$_{q_y}$ requirement with $11 \leq B_y \leq 13$ but fails to do so with $B_y = 8$. Figure 4(b) shows that SQNR$_{q_y(\text{dB})}$ is maximized when $\gamma_{q_y(\text{dB})} = 4$, i.e., when clipping level $y_c = 4\sigma_y$, thereby illustrating MPC's quantization vs. clipping noise trade-off described by (14).

Figure 4 also validates the analytical expressions (8), (9), (13), and (14) (bold) by indicating a close match to ensemble-averaged values of SQNR$_{q_y}$ obtained from Monte Carlo simulations (dotted).
### IV. Analytical Models for Compute SNR

This section derives analytical expressions for SNR, of a typical IMC. We introduce compute models that form the fundamental building blocks of IMCs and present analytical expressions for circuit domain equivalents of IMCs. Typically, these models are expressed as a mapping of algorithmic variables $y_o, x_j$, and $w_j$ in (2) to physical quantities such as time, charge, current, or voltage, in order to (usually partially) realize an analog BL computation of the multi-bit DP in (2).

**A. In-memory Compute Models**

All IMCs are viewed as employing one or more in-memory compute models defined as a mapping of algorithmic variables $y_o, x_j$, and $w_j$ in (2) to physical quantities such as time, charge, current, or voltage, in order to (usually partially) realize an analog BL computation of the multi-bit DP in (2).

Furthermore, we suggest that most IMCs today employ one or more of the following three in-memory compute models:

**B. Charge Summing (QS)**

1) **The QS Model:** The QS model (see Fig. 5a) realizes the DP in (2) via the variable mapping $(y_o \rightarrow V_o, \eta_j \rightarrow I_j, x_j \rightarrow T_j)$ where the cell current $I_j$ is integrated over the WL pulse duration $T_j$ ($j = 1, \ldots, N$) on a BL (or cell) capacitor $C$ resulting in an output voltage as shown below:

$$ (y_o \rightarrow V_o) = \frac{1}{C} \sum_{j=1}^{N} (w_j \rightarrow I_j)(x_j \rightarrow T_j) $$

where $V_o$ is the DP output assuming infinite voltage headroom, i.e., no clipping. The cell current $I_j$ depends upon transistor sizes and the WL voltage $V_{WL}$, and typical values are: $C$ (a few hundred fF), $I_j$ (tens of µAs), and $T_j$ (hundreds of ps).

**Noise Models:** The noise contributions in QS arise from the following sources: (1) variations in the pulse-widths $T_j$ of...
current switch pulses $\phi_j$ (Fig. 3(a)); (2) their finite rise and fall times (see Fig. 3(b)); (3) spatial variations in the cell currents $I_j$; (4) thermal noise in the discharge RC-network; and (5) clipping due to limited voltage head-room. Thus, the analog DP output $V_o$ corresponding to $y_a = y_0 + \eta_h$ is given by:

$$
(y_a \to V_o) = (y_0 \to V_o) + (\eta_k \to v_e) + (\eta_h \to v_c),
$$

$$
v_e = v_0 + \frac{1}{C} \sum_{j=1}^{N} i_j T_j + I_j (t_j - t_{\text{rf}}),
$$

$$
v_c = \min (V_o, V_{o,\text{max}}) - V_o,
$$

where $V_{o,\text{max}}$ is the maximum allowable output voltage, and $v_e$ and $v_c$ are the voltage domain noise due to circuit non-idealities and clipping, respectively, $i_j \sim N(0, \sigma_{j}^2)$ is the noise due to (spatial) current mismatch, and $t_j \sim N(0, \sigma_{\text{t}}^2)$ is the noise due to (temporal) pulse-width mismatch, respectively, both of which are modeled as zero mean Gaussian random variables, $t_{\text{rf}}$ models the impact of finite rise and fall times of the current switching pulses, and $v_0 \sim N(0, \sigma_0)$ is the integrated thermal noise voltage. Note: $V_{o,\text{max}}$ can be as high as 0.9 V when $V_{dd} = 1$ V.

Analytical expressions to estimate the noise standard deviations $\sigma_{I_j}$, $\sigma_{T_j}$, $\sigma_{\theta}$, and $t_{\text{rf}}$, (see appendix) are provided below:

$$
\sigma_{I_j} = I_j \left( \frac{\alpha \sigma_{V_i}}{V_{WL}} - V_i \right) = I_j \sigma_0
$$

$$
t_{\text{rf}} = T_i - \left( \frac{V_{WL} - V_i}{V_{WL}} \right) \frac{T_i + T_f}{\alpha + 1}
$$

$$
\sigma_{T_j} = \sqrt{h_j \sigma_{T_0}}, \quad \sigma_0 = \frac{1}{C} \sqrt{\frac{N T_{\text{max}} g_m kT}{3}}
$$

where $\sigma_0^2$ is normalized current mismatch variance, $T_j = h_j T_0$ is the delay of a $h_j$-stage WL driver composed unit elements with delay $T_0$ each, $\sigma_{T_0}$ is the standard deviation of $T_0$, $T_i$ and $T_f$ are WL pulse rise and fall times (see Fig. 3(b)), $\alpha$ is a fitting parameter in the $\alpha$-law transistor equation, $\sigma_{V_i}$ is standard deviation of $V_i$, $k$ is the Boltzmann constant, $T$ is the absolute temperature, and $g_m$ is the transconductance of the access transistor.

Note that typically the WL voltage $V_{WL}$ is identical for all rows in the memory array with a few exceptions such as [3] which modulates $V_{WL}$ to tune the cell current $I_j$. The effects of rise/fall times and delay variations can be mitigated by carefully designing the WL pulse generators. Therefore, noise in QS is dominated by spatial threshold voltage variations. Indeed, using the typical values from Table II we find that $\sigma_{I_j}/I_j$ ranges from 8% to 25%, while $\sigma_{T_j}/T_j$ ranges from 0.5% to 3%.

**Energy and Delay Models:** The average energy consumption in the QS model is given by:

$$
E_{QS} = \mathbb{E} [V_o] V_{dd} C + E_{su}
$$

where the spatio-temporal expectation $\mathbb{E} [V_o]$ is taken over inputs (temporal) and over columns (spatial) $E_{su}$ is the energy cost of toggling switches $\phi_j$s. Equation (21) shows that the energy consumption in the QS model increases with $C$ array size, the supply voltage $V_{dd}$, and the mean value of the DP $\mathbb{E} [V_o]$.

The delay of the QS model is given by $T_{QS} = T_{\text{max}} + T_{\text{su}}$, where $T_{su}$ is the time required to precharge the capacitors and setup currents, and $T_{\text{max}} = \max \{T_j\}$ is the longest allowable pulse-width.

2) The QS Architecture (QS-Arch): The charge summing architecture (QS-Arch) in Fig. 7(b) employs a 6T [41] or 8T [38] SRAM bitcell within the QS model (see Section IV-B). This architecture implements fully-binarized DPs on the BLs by mapping the input bit $\hat{x}_{i,j}$ to the WL access pulse $V_{WL,j}$ while the weights $\hat{w}_{i,j}$ are stored across $B_m$ columns of the BCA so that the BC currents $I_{i,j} \propto \hat{w}_{i,j}$. The output $V_o = \Delta V_{BL}$ is the voltage discharge on the BL and the capacitance $C = C_{BL}$ is the BL capacitance in (16). QS-Arch sequentially (bit-serially) processes one multi-bit input vector $\mathbf{x}$ in $B_m$ memory compute cycles followed by a digital summing of the binarized DPs to obtain the final multi-bit DP $[3]$. We derive the analytical expressions of architecture-level noise models for QS-Arch using those of the QS model described in Section IV-B. In QS-Arch, clipping occurs in each of the $B_x \times B_w$ binarized DPs and contributes to the overall clipping noise variance $\sigma_{\text{clip}}^2$ at the multi-bit DP output. Circuit noise from each binarized DP is aggregated to obtain the final circuit noise variance $\sigma_{\text{circuit}}^2$. In addition, employing MPC imposed requirement on the final DP output precision $B_y$ ([95]), we obtain the lower bound on ADC precision $B_{ADC}$. Since the multi-bit DP computation in [95] is high-dimensional ($N$ can be in hundreds), it is clear that the limited BL dynamic range e.g., $V_{o,\text{max}}$ in (17), will begin to dominate $\text{SNR}_a$ in (7). It is for this reason that most, if not all, IMCs resort to some form of binarization of the multi-bit DP in [95] prior to employing one of the in-memory compute models (see Table I). Ultimately, $\text{SNR}_a$ limits the number and accuracy of BL computations per read cycle and hence the overall energy efficiency of IMCs.
TABLE III

|                  | QS-Arch                  | QR-Arch                  | CM                  |
|------------------|--------------------------|--------------------------|---------------------|
| Bitcell type     | 6T or 8T                 | 8T or 10T + MOM cap      | 6T                  |
| Analog Core      | Binarized ($B_w = B_x = 1$) | Binary-weighted ($B_w = 1$) | Multi-bit           |
| Precision        |                          |                          |                     |
| Compute model    | QS                       | QR                       | QS & QR             |
| Energy cost      | $E_{QS-Arch} = B_w B_c (E_{QS} + E_{ADC}) + E_{misc}$ | $E_{QR-Arch} = B_w (E_{QR} + NE_{m} + E_{ADC}) + E_{misc}$ | $E_{CM} = 2N E_{QS} + E_{QR}$ + $E_{misc}$ |
| Compute model    | $C \rightarrow C_{BL}$, $V_0 \rightarrow \Delta V_{BL}$, $T_j \rightarrow T_{WL,j}$ | $C_j \rightarrow C_o$ | $QS$, $C \rightarrow C_{BL}$, $V_0 \rightarrow \Delta V_{BL}$, $T_j \rightarrow T_{WL,j}$, $QR$: $C_j \rightarrow C_o$ |
| $\sigma_{m_i}$   | $\frac{1}{12} N \Delta \sigma^2_w + \frac{1}{12} N \Delta \sigma^2_e [x^2]$ | $\frac{1}{12} N \Delta \sigma^2_w + \frac{1}{12} N \Delta \sigma^2_e [x^2]$ | $\frac{1}{12} N \Delta \sigma^2_w + \frac{1}{12} N \Delta \sigma^2_e [x^2]$ |
| $\sigma_{m_w}$   | $\frac{4}{3} \left(1 - 4^{-B_w}\right) \left(1 - 4^{-B_x}\right) \sum_{k=1}^{N} \left(k - h_0\right)^2 \left(N \frac{1}{4} \frac{1}{4}(3)\right) N^{-k}$ | $\frac{4}{3} \left(1 - 4^{-B_w}\right) N \left(\frac{1}{4} + \frac{1}{4} \frac{1}{4}\right) \frac{\sigma^2_w}{\sigma^2_{BL}} + \frac{\sigma^2_e}{\sigma^2_{BL}}$ | $\frac{4}{3} \left(1 - 4^{-B_w}\right) N \frac{\sigma^2_w}{\sigma^2_{BL}}$ |
| $\sigma_{n_o}$   | $\frac{2}{3} \left(1 - 4^{-B_w}\right) N \left(\frac{1}{4} \frac{1}{4}\right) \frac{\sigma^2_w}{\sigma^2_{BL}}$ | $\frac{2}{3} \left(1 - 4^{-B_w}\right) N \left(\frac{1}{4} \frac{1}{4}\right) \frac{\sigma^2_w}{\sigma^2_{BL}}$ | $\frac{2}{3} \left(1 - 4^{-B_w}\right) N \left(\frac{1}{4} \frac{1}{4}\right) \frac{\sigma^2_w}{\sigma^2_{BL}}$ |
| $B_{ADC}$        | $\geq \min \left(\frac{\text{SNR}_{\text{dB}}+16.2}{9}, \text{log}_2(h_0) \text{log}_2(N)\right)$ | $\geq \min \left(\frac{\text{SNR}_{\text{dB}}+16.2}{9}, B_x + \text{log}_2(N)\right)$ | $\geq \frac{\text{SNR}_{\text{dB}}+16.2}{9}$ |
| $V_c$            | $\min \left(4 \sqrt{3} N \Delta V_{BL,\text{unit}}, \Delta V_{BL,\text{max}}, N \Delta V_{BL,\text{unit}}\right)$ | $8 V_{\text{dd}} \sqrt{\frac{x[x^2+x+2]}{N}}$ | $8 \sigma_{BL,\text{unit}} \Delta V_{BL,\text{max}} \sqrt{2[x^2]} \sqrt{N}$ |

$h_0 = \frac{V_{BL,\text{max}}}{\Delta V_{BL,\text{unit}}}; \sigma_D = \frac{1}{\sqrt{2}}$ is the normalized standard deviation of the bit-cell current \cite{18}; $(x)^+ = \max(x, 0)$; $\sigma_{ij}^2 = E \left[x^2\right] W/L C\text{ox}/C_0$.

C. Charge Redistribution (QR)

1) The QR model: The QR model (Fig. [5]c) is commonly employed to perform the additions in (2). The multiplications in (2) are separately computed via charging/discharging capacitor $C_j$ in proportion to the product $w_j x_j$ (j = 1, ..., N) as in [8], [12], or by employing explicit multiplier circuits such as in [6], [9]. The N capacitors share charge via a sequence of switching events (see Fig. [5]c) to generate the final voltage $V_o$ given by:

$$ (y_o \rightarrow V_o) = \sum_j C_j \sum_j C_j (w_j x_j \rightarrow V_j) \quad (22) $$

The capacitors $C_j$ are typically metal-on-metal (MOM) capacitors with values ranging from 1 fF to 10 fF [6]. [12].

Noise Models: Assuming MOM-based $C_j$s, the noise contributions in QR arise from: (1) capacitor mismatch [45]; (2) charge injection due to switching [46], and (3) thermal noise. Unlike QS, and similar to 1A, the QR model does not suffer from headroom clipping noise. Hence, the DP output $V_a$ corresponding to $y_a$ in (6) is given by:

$$ (y_a \rightarrow V_a) = (y_o \rightarrow V_o) + (\eta_e \rightarrow v_e) \quad (23) $$

$$ = \frac{1}{\sum_j (C_j + c_j)} \sum_j (C_j + c_j)(V_j + v_{\theta_j} + v_j) $$

where $v_e$ is the voltage domain noise term due to circuit nonidealities corresponding to $\eta_e$ in [6], $v_j$ is the noise due to charge injection, $c_j \sim N(0, \sigma^2_{C_j})$ is the capacitor mismatch, and $v_{\theta_j} \sim N(0, \sigma_{\theta,j})$ is the thermal noise. Furthermore,
expressions for the noise parameters in (23) can be derived as [45], [46]:

\[
sig_{C_j} = \kappa \sqrt{C_j}, \quad v_j = p \frac{WLC_{ox}(V_{dd} - V_t - V_j)}{C_j}, \quad \sig_{\theta,j} = \sqrt{\frac{kT}{C_j}}
\]

(24)

where \(\kappa\) is a technology- and layout-dependent Pelgrom coefficient [45], \(0 \leq p \leq 1\) is constant that depends on the layout of the switch transistor, \(C_{ox}\) is the gate oxide capacitance per unit area, and \(W\) and \(L\) are the width and length of the switch transistor. The effect of noise in the QR compute model can be minimized by increasing the capacitors sizes at the expense of energy consumption as seen from (25) below.

**Energy and Delay Models:** The average energy consumption in the QR model is given by:

\[
E_{QR} = \sum_j E[(V_{dd} - V_j)] V_{dd} C_j + E_{sw}
\]

(25)

where \(E_{sw}\) includes energy cost for the switches \(\phi_{sw}\).

The delay of the QR model is given by: \(T_{QR} = T_{share} + T_{sw}\), where \(T_{share}\) is the time required for charge sharing to complete, and \(T_{sw}\) is the time required to precharge the capacitors to the desired voltages \(V_j\).

2) The QR architecture (QR-Arch): The QR-Arch, e.g., [8], in Fig. (7c) employs a modified BC to include a capacitor \(C_o\) and additional switches for multiplication within the QR model. While works such as [8] employ the parasitic capacitances on the BL within the BC, an explicit MOM capacitor is assumed for simplicity. QR-Arch implements a binary weighted DP by storing the weights \(\hat{w}_{i,j}\) across \(B_w\) rows of the BCA and by providing a multi-bit analog input \(x_j\) to the multiplier. The multiplication is implemented by first charging the capacitor \(C_o\) to a voltage proportional to \(x_j\) and then discharging it based on \(\hat{w}_{i,k}\). Multiplication is followed by a QR operation across the rows so that the final voltage across the capacitors in each row is proportional to binary-weighted DP. Thus, the QR-Arch processes one multi-bit input vector \(x\) in one in-memory compute cycle to compute binary-weighted DPs that are power-of-two (POT) summed digitally to obtain the final multi-bit DP (2).

The average energy per DP \(E_{QR-Arch}\) (see Table [III]) includes \(E_{QR}\) and \(E_{misc}\) which in turn includes the energy consumption of the DACs used for converting \(x_j\) into the analog domain. The energy cost of DACs are amortized since these are shared by multiple DPs computations.

Since the QR compute model does not suffer from headroom clipping, \(\sigma_{\text{h}} = 0\) and the lower bound on \(B_{ADC}\) is dictated solely by MPC. The primary analog noise contributors (\(\sigma_{\text{y}}^2\)) are the capacitor mismatch (\(\sigma_{\text{C}}^2\)), charge injection noise (\(\sigma_{\text{ij}}^2\)) in the switches, and thermal noise (\(\sigma_{\text{Th}}^2\)) as indicated in Table [III]. An alternative QR model [43], [47], [48] is to directly switch the bottom plate of the capacitors with binary outputs of the BC multiplies. Doing so leads to greater energy efficiencies and significant reduction in charge injection noise. The analysis presented in this section can be extended to such models as well.

**D. Compute Memory**

CM [1], [6], [9] in Fig. (7c) employs a 6T SRAM BC within the QS (see Section [V-B]) and QR (see Section [V-C]) models. In the most general case, CM strives to implement a \(B_w \times B_x\)-b DP directly by mapping \(B_x\)-b inputs \(x_j\) to pulse width \(T_j\) and/or amplitude \(V_{WL,j}\) of the WL access pulses and storing \(B_w\)-b weights \(w_j\) in a column-major format across \(B_w \times N\) cells. In practice, CM realizations such as [6] employ POT weighted WL access pulse-widths for \(B_w\) rows so that the voltage discharge \(\Delta V_{BL}\) on the \(j\)-th BL is proportional to the weight \(w_j\). The product \(w_j x_j\) is realized using a per-column charge redistribution-based multiplier, followed by a QR stage to aggregate the \(N\) multiplier outputs. In this way, CM computes the \(B_w \times B_x\)-b DP (2) in analog in a single in-memory compute cycle.

The energy cost per DP for CM in Table [III] is obtained by substituting \(C = C_{\text{BL}}\) and \(V_o = \Delta V_{BL}\) in (21), and using (25) with \(C_j = C_o\). Here, \(E_{\text{mult}}\) is the energy consumption of the mixed-signal multiplier. The factor of 2 in first term accounts for discharge on both BL and BL-bar needed to realize signed weights [9]. The second term is the energy consumed in aggregating the column outputs using the QR model with identical capacitors \(C_o\).

The expression for \(\sigma_{\text{y}}^2\) neglects the impact of pulse width variations and other noise sources in QR since it is dominated by variations in the bit-cell current \(I_j\). However, the sample-accurate Monte Carlo simulations incorporate all noise sources.

The factor of \(\sqrt{N}\) in the denominator of the expression for the ADC input range \(V_d\) due to the use of QR indicates that the loss in voltage range due to charge redistribution across \(N\) capacitors.

**V. SIMULATION RESULTS**

This section describes the SNR validation methodology for validating the SNR (noise) expressions in Table [III] and simulation results for QS-Arch.

A. **SNR Validation Methodology**

Figure 8 describes the SNR validation methodology. We obtain the QS and QR model parameters (Section [V]) using Monte Carlo circuit simulations in a representative 65 nm CMOS process, with experimental validation of some of these, e.g., \(\sigma_{\text{y}},\) from our IMC prototype ICs [6], [49] when possible.

Incorporating non-linear circuit behavior along with noise models, sample-accurate Monte Carlo Python simulations are employed to numerically calculate SNR values using ensemble averaged (over 1000 instances) statistics. We compare the SNR values obtained through sample-accurate simulations
with those obtained by evaluating the analytical expressions in Table III.

The quantitative results in subsequent sections employ the QS and QR model parameter values in Table III along with QS-Arch, QR-Arch, and CM energy and noise models from Table III. An SRAM BCA with 512 rows and $C_{BL} = 270 \text{ fF}$ is assumed throughout. Energy and accuracy is traded-off of by tuning $V_{WL}$ in QS-Arch and CM, and by tuning $C_o$ in QR-Arch. We assume zero mean signed weights $w_j$ and unsigned inputs $x_j$ drawn independently from two different distributions. We set $B_x = B_w = 6$ everywhere, unless otherwise stated, so that $\text{SNR}_{\text{Q/S}(\text{dB})} > 38.9 \text{ dB} \Rightarrow \text{SNR}_{\text{ADC}(\text{dB})}$ and therefore $\text{SNR}_{A} \approx \text{SNR}_{T}$ from (10). Next, we show how $\text{SNR}_{A}$ and $\text{SNR}_{T}$ trade-off with $N$ and $B_{ADC}$.

B. SNR Trade-offs in IMCs

1) QS-Arch: Figure 9(a) shows that the maximum achievable $\text{SNR}_{A}$ increases with $V_{WL}$. Further, for a fixed $V_{WL}$, QS-Arch also exhibits a sharp drop in $\text{SNR}_{A}$ at high values of $N > N_{\text{max}}$, e.g., $\text{SNR}_{A} \approx 19.6 \text{ dB}$ for $N \leq 125$ and then drops with increase in $N$. A key reason for this trade-off is that $\sigma_{e_{\text{BL}}}^2$ decreases while $\sigma_{e_{\text{MP}}}^2$ increases as $V_{WL}$ is reduced (see Table III), and since $\sigma_{e_{\text{BL}}}^2$ limits $N$ and $\sigma_{e_{\text{MP}}}^2$ limits $\text{SNR}_{A}$. Thus, by controlling $V_{WL}$, we can trade-off $N_{\text{max}}$ with $\text{SNR}_{A}$. Specifically, $N_{\text{max}}$ increases by $2x$ for every 3 dB drop in $\text{SNR}_{A}$.

In QS-Arch, the minimum value of $B_{ADC}$ (see Table III) depends upon the minimum of: 1) the MPC term (15); 2) the headroom clipping term; and 3) the small $N$ case where BL discharge $\Delta V_{\text{BL}}$ has a finite number of discrete levels. Figure 9(b) shows that $\text{SNR}_{T} \rightarrow \text{SNR}_{A}$ of Fig. 9(a) when $B_{ADC}$ is greater than the lower bound (circled) in Table III for different values of $V_{WL}$ and $N$.

2) QR-Arch: QR-Arch demonstrates a clear energy-accuracy-area trade-off as seen in Fig. 10(a). Here, $\text{SNR}_{A}$ improves with capacitor $C_o$ size but at the expense of higher energy and area costs. For instance, increasing $C_o$ from 1 fF to 3 fF and 9 fF leads to $\text{SNR}_{A}$ improvements of $~8 \text{ dB}$ and $~12 \text{ dB}$, respectively.

Figure 10(b) shows that the expressions in Table III correctly predict the minimum value of ADC precision $B_{ADC}$ and the input range $V_c$. MPC is demonstrated to greatly reduce the ADC precision requirements as for this example 6-8 bits suffice in order to maintain $\text{SNR}_{A}$. In contrast, if BGC were to be employed, $B_{ADC} = 12$ would have been assigned.

3) Compute Memory: Figure 11(a) shows that the quantization noise term $\sigma_{q_{\text{MP}}}^2$ reduces and the headroom clipping noise $\sigma_{e_{\text{BL}}}^2$ increases as a function of $B_w$ implying an $\text{SNR}_{A}$-optimal value for $B_w$, e.g., $\text{SNR}_{A}$ peaks at $B_w = 6$ and $B_w = 7$ for $V_{WL} = 0.8 \text{ V}$ and $V_{WL} = 0.7 \text{ V}$, respectively.

Figure 11(a) also shows another interesting trade-off, this time between headroom clipping noise $\sigma_{e_{\text{BL}}}^2$ and $\sigma_{e_{\text{MP}}}^2$, e.g., when $B_w = 7$, $\text{SNR}_{A}$ is dominated by $\eta_k$ when $V_{WL} = 0.6 \text{ V}$ and by $\eta_h$ when $V_{WL} = 0.8 \text{ V}$. Furthermore, both noise sources are balanced when $V_{WL} = 0.7 \text{ V}$. In fact, one can show that the clipping threshold $k_h$ is proportional to $\sigma_D$ indicating this relationship.

Figure 11(b) shows that choosing $B_{ADC}$ using MPC (15) ensures that the $\text{SNR}_{T}$ is indeed within 0.5 dB of $\text{SNR}_{A}$ in Fig. 17(a). Once more, MPC assigns $B_{ADC} \leq 8$ when $B_x = B_w = 6$ and $N = 128$ which is much smaller than $B_{ADC} = 19$ determined via BGC.

C. Impact of ADC Precision

Minimizing the column ADC energy is critical to maintain IMC’s energy efficiency. ADCs in IMCs need to operate in a noise-limited regime due to the high PAR of high-dimensional DP outputs combined with severe area constraints imposed by column-pitch matching requirements. To estimate the ADC energy costs we use the following empirical model based on [48]:

$$E_{\text{ADC}} = k_1 \left( B_{\text{ADC}} + \log_2 \left( \frac{V_{\text{DD}}}{V_c} \right) \right) + k_2 \left( \frac{V_{\text{DD}}}{V_c} \right)^2 4^{B_{\text{ADC}}}$$

where $V_c$ is the voltage range that need to be quantized, and $k_1 = 100 \text{ fJ}$ and $k_2 = 1 \text{ nJ}$ are empirical parameters [48] based on the recent ADCs [50], [51].

From (25), it is clear that the energy consumption of ADC decreases with $V_c$ and increases with $B_{ADC}$. If BGC is employed, then $2^{B_{\text{ADC}}} \propto N$ resulting in ADC energy increasing with $N$ when $V_c$ is constant, as in the case of QR-Arch and in CM (see Fig 12(b) and Fig 12(c)). However, in QS-Arch, $V_c \propto N$ therefore the ADC energy consumption in QS-Arch remains constant with $N$ (see Fig 12(a)).

On the other hand, if MPC is employed, $B_{ADC}$ remains constant with $N$ (Table III), and hence $E_{\text{ADC}}$ only depends on $V_c$. In QS-Arch, $E_{\text{ADC}}$ reduces with $N$ as $V_c \propto \sqrt{N}$, while in QR-Arch and CM $E_{\text{ADC}}$ increases with $N$ as $V_c \propto 1/\sqrt{N}$ (see Fig 12(a)). Note that for QR-Arch and CM, MPC leads to significant energy savings over BGC criterion since $E_{\text{ADC}} \propto N^2$ in BGC while $E_{\text{ADC}} \propto N$ using MPC.

D. Impact of Technology Scaling

One expects IMCs to exhibit improved energy efficiency and throughput in advanced process nodes due to lower capacitance and lower supply voltage. However, the impact of technology scaling on the analog noise sources also needs to be considered. To study this trade-off, we employ the SNR and energy models from Section IV (see Table III) with parameters scaled as per the ITRS roadmap [52]. FDSOI technology is assumed for the 22 nm, 11 nm and 7 nm nodes.

For a specific node, Fig. 13 shows that the energy cost reduces by $2 \times$ in CM and QS-Arch, and $4 \times$ in QR-Arch for every $6 \text{ dB}$ drop in $\text{SNR}_{A}$. QS-Arch suffers a catastrophic drop in $\text{SNR}_{A}$ before reaching the input quantization noise limit set by (8). This drop occurs due to an increase in the clipping noise variance $\sigma_{e_{\text{BL}}}^2$. In contrast, QR-Arch is able to approach quantization noise limits as it does not suffer from headroom clipping noise.

Across technology nodes, the maximum achievable $\text{SNR}_{A}$ in QS-Arch and CM reduces as technology scales from 65 nm down to 7 nm due to: 1) increased clipping probability caused by lower supply voltages, and 2) increased variations in BL
Fig. 9. SNR trade-offs in the QS-Arch with $B_x = B_w = 6$: (a) $\text{SNR}_{\text{A(dB)}}$ vs. $N$ for different values of $V_{\text{WL}}$, and (b) $\text{SNR}_{\text{T(dB)}}$ vs. $B_{\text{ADC}}$ showing that the expression in Table III correctly predicts the minimum ADC precision $B_{\text{ADC}}$ (circled). Close match is achieved between expressions in Table III (E) and simulations (S) of (17).

Fig. 10. SNR trade-offs in the QR-Arch with $B_w = 7$, and $N = 64$: (a) $\text{SNR}_{\text{A(dB)}}$ as a function of $B_x$ for different values of $C_w$ showing that the SNR improves with $C_w$, and (b) $\text{SNR}_{\text{T(dB)}}$ as a function of $B_{\text{ADC}}$ for different values of $C_w$ with $B_x = 6$ and $B_w = 7$, showing that the expression in Table III correctly predicts the minimum ADC precision $B_{\text{ADC}}$ (circled). Here, ‘E’ and ‘S’ correspond to the evaluation of the expressions in Table III and sample-accurate simulations of (23), respectively.

Fig. 11. SNR trade-offs in CM with $B_x = 6$ and $N = 128$: (a) $\text{SNR}_{\text{A(dB)}}$ vs. $B_w$ indicating the existence of an optimal value of $B_w$ that balances quantization and headroom clipping noise, and (b) $\text{SNR}_{\text{T(dB)}}$ vs. $B_{\text{ADC}}$ with $B_w = 6$. Here, ‘E’ and ‘S’ correspond to the evaluation of the expressions in Table III and sample-accurate simulations using (17) and (23), respectively.

Fig. 12. ADC energy in (a) QS-Arch, (b) QR-Arch and (c) CM with $B_x = B_w = 6$ as a function of $N$ when $B_{\text{ADC}}$ chosen according to BGC (12), and the MPC criterion (Table III) such that $\text{SNR}_{\text{T(dB)}}$ is within 0.5 dB of $\text{SNR}_{\text{A(dB)}}$. $V_{\text{WL}} = 0.8$ V in CM, $V_{\text{WL}} = 0.7$ V in QS-Arch, and $C_o = 3$ fF in QR-Arch.

Fig. 13. Impact of CMOS technology scaling on the compute SNR vs. energy trade-off in (a) QS-Arch (swept parameter: $V_{\text{WL}}$), (b) QR-Arch (swept parameter $C_w$), and (c) CM (swept parameter: $V_{\text{WL}}$) with $B_x = 3$, $B_w = 4$, and $N = 100$. 
discharge voltage $\Delta V_{BL}$ due to smaller $V_{bd}/V_i$ ratio. As a result, Fig. [13] also shows that the energy consumption, at the same SNRA, is in fact higher in 11 nm and 7 nm nodes as compared to the 22 nm node in QS-Arch and CM due to the need to employ a higher values of $V_{th}$ to control variations in $\Delta V_{BL}$, implying the technology scaling may not be friendly to IMCs based on the QS compute model.

VI. CONCLUSIONS AND SUMMARY

Based on the results presented in the earlier sections, we provide the following IMC design guidelines:

- For IMCs to be useful in realizing DNNs, the compute SNR of their analog core (SNRa) needs to be the range 10 dB – 40 dB or greater depending on the layer. This is because the total SNR (SNRp) of DP computations implemented on IMCs is upper bounded by SNRa.
- In order for SNRF $\rightarrow$ SNRa with minimal energy and latency costs: 1) use [9] to ensure that the weight and activation precisions are sufficiently (e.g., 9 dB) below analog noise sources; and 2) use the MPC-based Precision Assignment Rule [15] to assign the ADC precision.
- QS-based architectures tend exhibit lower energy cost at low compute SNRs. Meanwhile, QR-based architectures are preferred when the compute SNR requirements are higher.
- For the QS-Arch, given an array size, there exists a trade-off between the maximum achievable SNRa and the maximum realizable DP dimension $N$. Multi-bank IMCs will be required for high-dimensional DPs in order to boost the overall compute SNR.
- Technology scaling will have an adverse impact on the maximum achievable SNRa and the energy cost incurred for a fixed SNRa for both QS-Arch and CM.
- When MPC is employed, the ADC energy increases with the DP size $N$ for both QR-Arch and CM due to the decreasing signal variance in the QR compute model. The opposite trend is observed for QS-Arch where the ADC energy decreases as $N$ is increased.
- CM avoids incurring large ADC energy consumption by realizing multi-bit DPs instead of binarized ones.

An overarching conclusion of this paper is that the drive towards minimizing energy and latency using IMCs, runs counter to meeting the compute SNR requirements imposed by applications. This paper quantifies this trade-off through analytical expressions for compute SNR and energy-delay models. It is hoped that IMC designers will employ these models as they seek to optimize the design of IMCs of the future, including the use of algorithmic methods for SNR boosting such as statistical error compensation (SEC) [53].

APPENDIX A

SQNR expressions

We present the derivation of the expressions of SQNR in [8], SQNR in [9], SQNR in [13], and SQNR in [14].

Derivation of SQNR in [8]:
Substituting $\Delta w = w_m^2 - B_w + 1$ and $\Delta x = x_m 2^{-B_x}$ in (5) yields:

$$\sigma^2_{y_q} = \frac{N}{3} \left( \frac{\sigma_w^2}{\frac{1}{4}} 2^{-2B_w} + \mathbb{E}[x^2] w_m^2 2^{-2B_w} \right)$$

which we substitute into the expression of SQNR in (7), along with the expression of $\sigma_{y_q}$ from (3) to obtain:

$$\text{SQNR}_{y_q} = \frac{N \mathbb{E}[x^2] \sigma_w^2}{3 \times 2^{2B_w} \left( \frac{2^{2B_w} + \frac{1}{4} 2^{-2B_w} \right)}$$

Dividing both numerators and denominators by $\frac{N}{3} \mathbb{E}[x^2] \sigma_w^2$, (28) can be written as:

$$\text{SQNR}_{y_q} = \frac{3 \times 2^{2B_w} + \frac{1}{4} 2^{-2B_w} \left( \frac{2^{2B_w} + \frac{1}{4} 2^{-2B_w} \right)}$$

The result for SQNR in [9] follows by taking $\text{SQNR}_{y_q}(dB) = 10 \log_{10}(\text{SQNR}_{y_q})$, with SQNR given by (29).

Derivation of SQNR in [13]:
From the SQNR definition in (1), we have

$$\text{SQNR}_{y_q}(dB) = 6B_y + 4.78 - \zeta_y(yq)$$

Thus, it suffices to show that $\zeta_y(yq) = \zeta_x(yq) + \zeta_w(yq) + 10 \log_{10}(N)$ for the result in (1) to follow. For simplicity, let us assume signed integer inputs and weights. Since $y_o = w^{T}x$, we have $y_m = N x_m w_m$, (no clipping) and $y_m = \sqrt{N} \sigma_x w_o$. Thus, $\zeta_y = \frac{y_m}{\sigma_y} = \sqrt{N} \sigma_x \zeta_w$ with $\zeta_x = \frac{x_m}{\sigma_x}$ and $\zeta_w = \frac{w_m}{\sigma_w}$. The result follows by writing $\zeta_y = 20 \log_{10}(\zeta_y)$.

Derivation of SQNR in [13]:
The result follows by replacing $B_y$ in (9) by $B_x + B_w + \log_2(N)$ which is assigned by BGC as per (12). Note that $6 \log_2(N) \approx 20 \log_{10}(N)$ hence we obtain

$$\text{SQNR}_{y_q}(db) = 6(B_x + B_w + \log_2(N))$$

as listed in [13].

Derivation of SQNR in [14]:
In MPC we have:

$$\text{SQNR}_{y_q} = \frac{\sigma^2_{y_q}}{\sigma^2_{y_q} + \rho_o \sigma^2_{cc}} = \frac{\sigma^2_{y_q}}{\sigma^2_{y_q} + \rho_o \sigma^2_{cc}}$$

with $\sigma^2_{y_q} = \Delta y^2 = w_m^2 - 2y_m$, $\rho_o = \mathbb{P} \{ |y_o| > y_c \}$, and $\sigma^2_{cc} = \mathbb{E} \{ (y_o - y_c)^2 \}$. The above can be re-written as

$$\text{SQNR}_{y_q} = \frac{3 \times 2^{2B_w}}{\left( \zeta_{yq} \right) 2 \left( 1 + \rho_o \sigma^2_{y_q} \right)}$$

with $\zeta_{yq} = \frac{w_m}{\sigma_y}$. The result for SQNR in [14] follows by taking $\text{SQNR}_{y_q}(db) = 10 \log_{10}(\text{SQNR}_{y_q})$, with SQNR given by (30).
APPENDIX B
ANALOG NOISE MODELS EXPRESSIONS

We present the derivation of expressions (18), (19), expressions for noise variances \((\eta_h, \eta_e)\) and ADC input range \(V_c\) listed in Table III.

**Derivation of (18):**

Employ the \(\alpha\)-law transistor I-V equation below to model the SRAM cell current \(I_j\) (see Fig. 6(a)):

\[
I_j = \frac{W}{L} k(V_{WL} - V_i)^\alpha
\]

In the presence of threshold voltage variations, (31) transforms into:

\[
I_j + i_j = \frac{W}{L} k'(V_{WL} - (V_i + \Delta V_i))^\alpha
\]

where \(v_i\) is the threshold voltage variation and \(i_j\) is the resulting cell current variation. Using a 1st-order Taylor series expansion, we get:

\[
i_j \approx v_i \frac{\partial I_j}{\partial V_i} = -v_i \frac{\alpha I_j}{V_{WL} - V_i}
\]

Assuming \(v_i\) is a zero mean random variable with standard deviation \(\sigma_{v_i} = \sqrt{\text{Var}(v_i)}\) leads to (18) with \(\sigma_{I_j} = \sqrt{\text{Var}(i_j)}\).

**Derivation of (19):**

To model the impact of finite rise/fall time on the total voltage discharge associated with \(j\)-th bitcell \(V_{o,j}\), we integrate the total charge accumulated on the bitline cap \(C\):

\[
V_{o,j} + v_j = \frac{1}{C} \int_{t=0}^{t=T_j} I_j(t) dt
\]

where \(V_{o,j} = I_j T_j / C\) is the total discharge assuming an ideal \(V_{WL}\) pulse \((T_i = T_f = 0 \rightarrow I_j(t) = I_j)\), and \(v_j\) is the voltage drop that accounts for these effects. Modeling the SRAM cell current as an ideal current source with value set by \(I_j\), we get:

\[
V_{o,j} + v_j = \frac{W k' i_j}{C} \int_{t=0}^{t=T_j} (V_{WL} - V_i)^\alpha dt
\]

To simplify the analysis, we employ a linear approximation (red curve) to a realistic \(V_{WL}\) waveform in Fig. 6(b)). Evaluating (35) with the linear approximation results in:

\[
V_{o,j} + v_j = \frac{I_j}{C} \left[ T_j - T_i + \left( \frac{V_{WL} - V_i}{V_{WL}} \right) \frac{T_i + T_f}{\alpha + 1} \right]
\]

Thus, we have (19) to account effect of rise and fall times of the WL pulse.

**Derivation of (20):**

To derive the thermal noise variance in (20), we employ the bit-cell model in Fig 6(a). The access transistors in the \(j\)-th bitcell contribute thermal noise \(i_{\theta,j}\). The final BL thermal noise voltage \(v_{\theta}\) is obtained by integrating the thermal noise current \((i_{\theta,j})\) contributions from the \(N\) bit-cells attached to the BL, on the output capacitor \(C\) as follows:

\[
v_{\theta} = \frac{1}{C} \sum_{j=1}^{N} w_j \left[ \int_{t=0}^{t=T_j} i_{\theta,j} dt \right]
\]

Assuming the access transistors are in saturation, the two-sided power spectral density of \(i_{\theta,j}\) is given by \(S_{i_{\theta}}(f) = \frac{4}{3} g_m kT\). Therefore:

\[
E \left[ \left( \int_{t=0}^{t=T_j} i_{\theta,j} dt \right)^2 \right] = T_j S_{i_{\theta}}(0) = T_j \frac{4}{3} g_m kT
\]  

We assume \(T_j = c x T_{max}\) are independent and the integrated \(i_{\theta,j}\) is independent and zero mean. Further, assuming \(w_j \in \{0, 1\}\) are Bernoulli distributed with parameter 0.5, we obtain:

\[
\sigma_{\theta}^2 = E[{w_j}^2] = \frac{1}{C^2} \sum_{j=1}^{N} E[{w_j}^2] E[T_j] \frac{4}{3} g_m kT
\]

\[
= \frac{1}{C^2} \sum_{j=1}^{N} \frac{1}{2} \frac{T_{max} N}{2} \frac{3}{4} g_m kT = \left( \frac{T_{max} N}{C^2} \right) \frac{g_m kT}{3}
\]

**Derivation of \(\sigma_{\eta_h}^2\) in CM:**

We write the headroom clipping noise in CM as:

\[
\eta_h = \sum_{j=1}^{N} \left( w_j - \min(|w_j|, w_h) \text{sign}(w_j) \right) x_j = \lambda^T x
\]

where \(w_h = k_h \Delta w\) is the smallest value of \(|w_j|\) that leads to clipping, and \(\lambda\) is the clipping noise vector. The clipping noise terms can be assumed to be independent from each other and from the inputs. Furthermore, by virtue of the weights having a symmetric distribution, the clipping noise has zero mean, so that:

\[
\sigma_{\eta_h}^2 = N E[\lambda^2] E[\lambda^2].
\]

In addition, the headroom clipping noise term variance is given by:

\[
E[\lambda^2] = \text{Pr}(|w| \geq w_h) \frac{(w - w_h)^2}{|w| > w_h}
\]

Then, we use the bound \(\text{Pr}(|w| \geq w_h) \leq \frac{\sigma_{\eta}^2}{w_h^2}\) by virtue of Chebyshev’s inequality, and we evaluate \(E[(w - w_h)^2] = (1 - w_h)^2\). Substituting into the above, we obtain an estimate for the headroom clipping noise term variance:

\[
E[\lambda^2] \approx \frac{1}{12} \sigma_{\eta}^2 k_h^2 2^{2B_w} \left( 1 - 2 k_h^2 2^{2B_w} \right)^2
\]

which we plug into (42) to obtain the expression for the total clipping noise variance \(\sigma_{\eta_h}^2\) in CM as listed Table III.

**Derivation of \(\sigma_{\eta_e}^2\) in CM:**

We write the electrical noise in CM as: \(\eta_e = \sum_{j=1}^{N} x_j \delta_{w,j}\) where \(\delta_{w,j}\) is the electrical noise term corresponding to the discharge of weight \(w_j\). By virtue of inputs being independent from electrical noise terms, and assuming the latter are identically distributed, we have:

\[
\sigma_{\eta_e}^2 = N E[\lambda^2] E[\delta_{w,j}^2].
\]
Assuming weight and input bits to be independent and equally distributed with zero-mean. Further, cross-correlations of headroom clipping of the individual headroom clipping noise terms being non-zero makes the overall headroom clipping noise zero-mean in spite of the individual headroom clipping noise terms being non-zero-mean. Further, cross-correlations of headroom clipping noise terms are neglected, so that the total headroom clipping noise variance is given by:

$$\sigma^2_{h_b} = \sum_{i=1}^{B_w} \sum_{j=1}^{B_s} 2^{1-i-j} \lambda_{i,j}$$

where $$\lambda_{i,j}$$ is the headroom clipping noise term for every bit-wise DP. Note that the nature of two’s complement arithmetic makes the overall headroom clipping noise zero-mean in spite of the individual headroom clipping noise terms being non-zero-mean. Further, cross-correlations of headroom clipping noise terms are neglected, so that the total headroom clipping noise variance is given by:

$$\sigma^2_{h_b} = \sum_{i=1}^{B_w} \sum_{j=1}^{B_s} 2^{1-i-j} \lambda_{i,j}$$

In addition, for independent, identically distributed headroom clipping noise terms, we obtain:

$$\sigma^2_{h_b} = \sum_{i=1}^{B_w} \sum_{j=1}^{B_s} 2^{1-i-j} \lambda_{i,j}$$

which we plug into (50) to obtain the expression for $$\sigma^2_{h_b}$$ in QS-Arch as listed in Table III.

**Derivation of $$V_c$$ in CM:**

In CM, BL discharge [43] is: $$\Delta V_{BL} = 2^{B_w-1} V_{BL\text{-unit}}$$, and therefore after multiplications and aggregations via charge sharing voltage at the input of the ADC is given by:

$$\Delta V_o = \frac{2^{B_w-1} V_{BL\text{-unit}}}{N} \sum_{i=1}^{N} w_i x_i$$

Therefore, $$V_c = \frac{\sqrt{N} \sigma}{\sigma}$$, we obtain the expression for $$V_c$$ in CM as listed in Table III.

**Derivation of $$\eta_h$$ in QS-Arch:**

We write the headroom clipping noise in QS-Arch as:

$$\eta_h = \sum_{i=1}^{B_w} \sum_{j=1}^{B_s} 2^{1-i-j} \delta_{i,j,k}$$

Further, for identically distributed electrical noise terms, the above simplifies to:

$$\sigma^2_{h_b} = \frac{4}{9} N \left(1 - 4^{-B_w}ight) \left(1 - 4^{-B_s}\right) \text{Var}(\delta)$$

where $$\delta$$ is the electrical noise per bit-cell discharge whose variance is:

$$\text{Var}(\delta) = \frac{1}{4} \sigma^2_{D}$$

with the $$\frac{1}{4}$$ term being due to the necessity of both input and weight bits to equal 1. This value of $$\text{Var}(\delta)$$ is plugged into (51) to obtain the expression for $$\sigma^2_{h_b}$$ in QS-Arch as listed in Table III.

**Derivation of $$V_c$$ in QR-Arch:**

In QR-Arch, we estimate binary-weighted DP in each column using charge sharing, as:

$$V_e = \frac{\sqrt{N}}{N} \sum_{i=1}^{N} x_i w_i$$

Since in VR-Arch we need its standard deviations to estimate $$V_c$$. Since $$w_i$$ is binary-valued, $$\text{Var}[x] = 0.5 \delta$$, and $$\text{Var}[V - 0.5 \delta]$$ is obtained using (18). Assuming weight bits to equal 1. This value of $$\text{Var}(\delta)$$ is plugged into (51) to obtain the expression for $$\sigma^2_{h_b}$$ in VR-Arch as listed in Table III.

REFERENCES

[1] M. Kang et al., “An energy-efficient VLSI architecture for pattern recognition via deep embedding of computation in SRAM,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 8326–8330.

[2] N. Shanbhag et al., “Compute memory,” US Patent 9,697,877, Issued July 4th 2017.

[3] M. Kang et al., *Deep In-Memory Architectures for Machine Learning*, Springer, 2020.

[4] N. Verma et al., “In-memory computing: Advances and prospects,” *IEEE Solid-State Circuits Magazine*, vol. 11, no. 3, pp. 43–55, 2019.

[5] J. Zhang et al., “In-memory computation of a machine-learning classifier in a standard 6T SRAM array,” *IEEE Journal of Solid-State Circuits*, vol. 52, no. 4, pp. 915–924, April 2017.

[6] M. Kang et al., “A multi-functional in-memory inference processor using a standard 6T SRAM array,” *IEEE Journal of Solid-State Circuits*, vol. 53, no. 2, pp. 642–655, 2018.

[7] Z. Jiang et al., “XNOR-SRAM: In-memory computing SRAM macro for binary/ternary deep neural networks,” in *2018 IEEE Symposium on VLSI Technology*. IEEE, 2018, pp. 173–174.

[8] A. Biswas and A. P. Chandrakasan, “Conv-RAM: An energy-efficient SRAM with embedded convolution computation for low-power CNN-based machine learning applications,” in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2018, pp. 488–490.

[9] S. K. Gonugondla et al., “A variation-tolerant in-memory machine learning classifier via on-chip training,” *IEEE Journal of Solid-State Circuits*, vol. 53, no. 11, pp. 3163–3173, 2018.
C. Sakr and N. Shanbhag, “An analytical method to determine minimum attention in-memory processor for keyword spotting,” in 2020 IEEE Custom Integrated Circuits Conference (CICC). IEEE, 2020, pp. 1–4.

W.-S. Khwa et al., “A 65nm 4Kb algorithm-dependent computing-in-memory SRAM unit-macro with 2.3 ns and 55.8 TOPS/W fully parallel product-sum operation for binary DNN edge processors,” in IEEE International Solid-State Circuits Conference (ISSCC), 2018, pp. 496–498.

H. Valavi et al., “A mixed-signal binaryzied convolutional-neural-network accelerator integrating dense weight storage and multiplication for reduced data movement,” in 2018 IEEE Symposium on VLSI Circuits. IEEE, 2018, pp. 141–142.

Kim et al. “Area-efficient and variation-tolerant in-memory BNN computing using 6T SRAM array,” in 2019 IEEE Symposium on VLSI Circuits. IEEE, 2019, pp. 118–119.

Q. Dong et al., “A 351 TOPS/W and 372.4 GOPS compute-in-memory SRAM macro in 7nm FinFET CMOS for machine learning applications,” in IEEE International Solid-State Circuits Conference (ISSCC), 2020, pp. 242–243.

J.-W. Su et al., “A 28nm 64Kb inference-training two-way transpose multibit 6T SRAM compute-in-memory macro for AI edge chips,” in IEEE International Solid-State Circuits Conference (ISSCC), 2020, pp. 240–241.

X. Si et al., “A 28nm 64Kb 6T SRAM computing-in-memory macro with 8b MAC operation for AI edge chips,” in IEEE International Solid-State Circuits Conference (ISSCC), 2020, pp. 246–247.

A. Jaiswal et al., “9T SRAM cell as a multi-bit dot product engine for beyond von-neumann computing,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 11, pp. 2556–2567, 2019.

M. Ali et al., “IMAC: In-memory multi-bit multiplication and accumulation in 6T srarn array,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 8, pp. 2521–2531, 2020.

X. Si et al., “A dual-split 6T SRAM-based computing-in-memory unit-macro with fully parallel product-sum operation for binarized DNN edge processors,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 11, pp. 4172–4185, 2019.

Z. Liu et al., “NS-CIM: A current-mode computation-in-memory architecture enabling near-sensor processing for intelligent IoT vision nodes,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 9, pp. 2909–2922, 2020.

S. Zhang et al., “A robust 8-bit non-volatile computing-in-memory core for low-power parallel MAC operations,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 6, pp. 1867–1880, 2020.

M. Gong et al., “A 65nm thermometer-encoded time/charge-based compute-in-memory neural network accelerator at 0.735pJ/MAC and 0.41pJ/Update,” IEEE Transactions on Circuits and Systems II: Express Briefs, pp. 1–1, 2020.

A. Agrawal et al., “Xcel-RAM: Accelerating binary neural networks in high-throughput SRAM compute arrays,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 8, pp. 3064–3076, 2019.

A. Jaiswal et al., “i-SRAM: Interleaved wordlines for vector boolean operations using srams,” IEEE Transactions on Circuits and Systems I: Regular Papers, pp. 1–9, 2020.

S. Yin et al., “Vesti: Energy-efficient in-memory computing accelerator for deep neural networks,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 1, pp. 48–61, 2020.

S. Srinivasa et al., “ROBIN: Monolithic-3D SRAM for enhanced robustness with in-memory computation support,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 7, pp. 2533–2545, 2019.

M. Abu Lebdeh et al., “An efficient heterogeneous memristive XNOR for in-memory computing,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 64, no. 9, pp. 2427–2437, 2017.

M. Kang et al., “Deep in-memory architectures for machine learning - accuracy vs. efficiency trade-offs,” IEEE Transactions on Circuits and Systems - Part I, vol. 67, no. 5, pp. 1627–1639, January 2020.

S. K. Gonugondla et al., “Fundamental limits on the precision of in-memory architectures,” in IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2020.

C. Sakr et al., “Analytical guarantees on numerical precision of deep neural networks,” in International Conference on Machine Learning, 2017, pp. 3007–3016.

C. Sakr and N. Shanbhag, “An analytical method to determine minimum per-layer precision of deep neural networks,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 1090–1094.

H. Jia et al., “A microprocessor implemented in 65nm CMOS with configuruable and bit-scalable accelerator for programmable in-memory computing,” arXiv preprint arXiv:1811.04047, 2018.

S. Okumura et al., “A ternary based bit scalable, 8.80 TOPS/W CNN accelerator with many-core processing-in-memory architecture with 890k synapses/mm2,” in 2019 IEEE Symposium on VLSI Circuits. IEEE, 2019, pp. 248–249.

R. Guo et al., “A 5.1pJ/neuron 127.3us/inference RNN-based speech recognition processor using 16 computing-in-memory SRAM macros in 65nm CMOS,” in 2019 IEEE Symposium on VLSI Circuits. IEEE, 2019, pp. 120–121.

J. Yue et al., “A 65nm computing-in-memory based CNN processor with 2.9-to-35.8TOPS/W system energy efficiency using dynamic-sparsity performance-scaling architecture and energy-efficient inter/intra-macro data reuse,” in IEEE International Solid-State Circuits Conference (ISSCC), 2020, pp. 234–235.

Z. Jiang et al., “CS3RAM: An in-memory-computing sram macro based on robust capacitive coupling computing mechanism,” IEEE Journal of Solid-State Circuits, 2020.

M. Kang et al., “An energy-efficient memory-based high-throughput VLSI architecture for convolutional networks,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2015.

V. Tripathi and B. Murmann, “Mismatch characterization of small metal fringe capacitors,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 61, no. 8, pp. 2236–2242, 2014.

G. Wegmann et al., “Charge injection in analog mos switches,” IEEE Journal of Solid-State Circuits, vol. 22, no. 6, pp. 1091–1097, 1987.

D. Bankman and B. Murmann, “An 8-bit, 16 input, 3.2 pJ/op switched-capacitor dot product circuit in 28-nm FinFET CMOS,” in Solid-State Circuits Conference (A-SSCC), 2016 IEEE Asian. IEEE, 2016, pp. 21–24.

B. Murmann, “Mixed-signal computing for deep neural network inference,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp. 1–11, 2020.

S. K. Gonugondla et al., “A 42pJ/decision 3.12 TOPS/W robust in-memory machine learning classifier with on-chip training,” in IEEE International Solid-State Circuits Conference (ISSCC), 2018, pp. 490–492.

B. Murmann. (2019) ADC performance survey 1997-2019. [Online]. Available: https://web.stanford.edu/~murmann/adcsurvey.html

J. Reki et al., “Analogue/digital converter trends: Power dissipation, scaling and digitally assisted architectures,” in 2008 IEEE Custom Integrated Circuits Conference. IEEE, 2008, pp. 105–112.

ITRS-collaborations, “ITRS roadmap tables,” ITRS, 2015. [Online]. Available: http://www.itrs.net/itrs-reports.html

N. R. Shanbhag et al., “Shannon-inspired statistical computing for the nanoscale era,” Proceedings of the IEEE, vol. 107, no. 1, pp. 90–107, 2018.