IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

Received 18 April 2023; revised 14 July 2023 and 30 August 2023; accepted 8 September 2023.
Date of publication 13 September 2023; date of current version 5 December 2023.

Digital Object Identifier 10.1109/JXCDC.2023.3315134

The Impact of Analog-to-Digital Converter Architecture and Variability on Analog Neural Network Accuracy

MATTHEW SPEAR[1,2] (Member, IEEE), JOSHUA E. KIM[1], CHRISTOPHER H. BENNETT[2] (Member, IEEE), SAPAN AGARWAL[2] (Member, IEEE), MATTHEW J. MARINELLA[1] (Senior Member, IEEE), and T. PATRICK XIAO[2] (Member, IEEE)

[1]School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA
[2]Sandia National Laboratories, Albuquerque, NM 87123 USA

CORRESPONDING AUTHOR: M. SPEAR (j.b.mspear@gmail.com)

This work was supported by the Laboratory Directed Research and Development Program at Sandia National Laboratories, a multimission laboratory managed and operated by the National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under Contract DE-NA0003525.

This article has supplementary downloadable material available at https://doi.org/10.1109/JXCDC.2023.3315134, provided by the authors.

ABSTRACT The analog-to-digital converter (ADC) is not only a key component in analog in-memory computing (IMC) accelerators but also a bottleneck for the efficiency and accuracy of these systems. While the tradeoffs between power consumption, latency, and area in ADC design are well studied, it is relatively unknown which ADC implementations are optimal for algorithmic accuracy, particularly for neural network inference. We explore the design space of the ADC with a focus on accuracy, investigating the sensitivity of neural network outputs to component variability inside the ADC and how this sensitivity depends on the ADC architecture. The compact models of the pipeline, cyclic, successive-approximation-register (SAR) and ramp ADCs are developed, and these models are used in a system-level accuracy simulation of analog neural network inference. Our results show how the accuracy on a complex image recognition benchmark (ResNet50 on ImageNet) depends on the capacitance mismatch, comparator offset, and effective number of bits (ENOB) for each of the four ADC architectures. We find that robustness to component variations depends strongly on the ADC design and that inference accuracy is particularly sensitive to the value-dependent error characteristics of the ADC, which cannot be captured by the conventional ENOB precision metric.

INDEX TERMS Analog computing, analog-to-digital conversion, in-memory computing (IMC), machine learning, neural network, process variations.

I. INTRODUCTION

Machine learning is increasingly being integrated into modern life. It is being used to defeat humans at chess and go, target ads at consumers, and produce realistic writing and art [1]. As these algorithms become more complex to meet new demands, they also demand more computing power. Neural network processing is dominated by linear algebra, and inference workloads are dominated in particular by matrix–matrix multiplication and matrix-vector multiplication (MVM) [2]. Improved efficiency in the MVM kernel over current digital hardware would therefore enable significant reduction in the power consumption of machine learning systems.

Analog in-memory computing (IMC) is an alternative approach to compute an MVM that allows for significant energy savings, potentially by orders of magnitude [3], [4]. Fig. 1 shows a diagram of an analog MVM. The inputs are applied as voltages on the rows. The weight matrix is mapped to the conductances in an array of programmable resistive memory devices (e.g., resistive RAM, flash memory, electrochemical RAM, and others) [5], [6], [7]. Ohm's law is used to multiply the inputs and weights, and Kirchoff's law down the columns sums the products to compute a dot product. These currents are typically converted to a voltage using a transimpedance amplifier (TIA) or integrator. Analog MVMs can also be computed inside an array of SRAM or DRAM

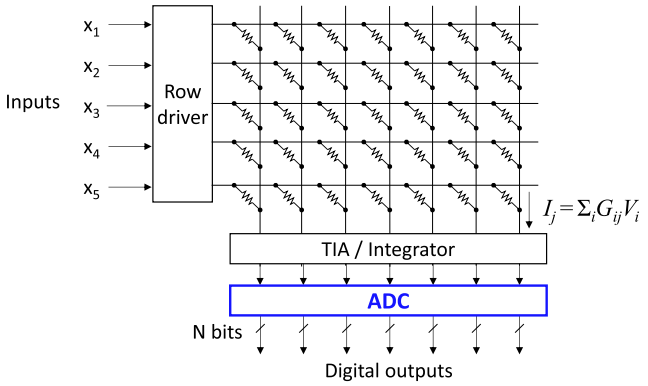

FIGURE 1. **Analog MVM using a resistive memory array.**

memory [8], [9]. In either case, the analog outputs are passed into an analog-to-digital converter (ADC) to interface with digital components, such as the on-chip network. In addition to the intrinsic efficiency and parallelism of analog computation, this scheme eliminates the energy cost of moving weight data between the processor and memory.

Despite the advantages of analog IMC, ensuring the accurate computation of MVMs remains a challenge. Notably, all nonvolatile memories have some variability in their conductance states. In analog IMC, this directly affects the result of the MVM computation and can lead to a reduction in the accuracy of the target algorithm. Other nonidealities such as the series resistance of the metal wires and errors induced by variability in peripheral circuits can also affect accuracy [10].

To date, the ADC's effect on neural network accuracy has only been analyzed through the lens of generic models for ADC quantization, clipping, and output noise [10], [11], [12], [13], [14], [15], [16], [17]. While this analysis is useful for optimizing the resolution and range of the ADC, it abstracts away the additional sources of error that are inside the ADC circuitry and offers little practical guidance on how to improve its design. A more detailed sensitivity analysis of errors that account for the ADC implementation has not yet been conducted. This detailed analysis is needed to set the circuit or process-level requirements on the ADC components, as well as to determine what type of ADC is best suited for an analog IMC application. This work investigates how the accuracy of analog neural network inference is affected by capacitance mismatch, insufficient amplifier gain, and comparator offset inside the ADC, and how these effects depend on the ADC circuit architecture.

II. ADC ARCHITECTURES

There are a variety of ADC architectures that implement different approaches to analog-to-digital conversion. Four ADC architectures are evaluated in this work: a pipeline ADC, a cyclic ADC, a ramp ADC, and a successive approximation register (SAR) ADC [18], [19], [20]. The pipeline and cyclic ADCs chain several low-resolution ADC stages together to produce a higher resolution ADC. The ramp and SAR ADCs

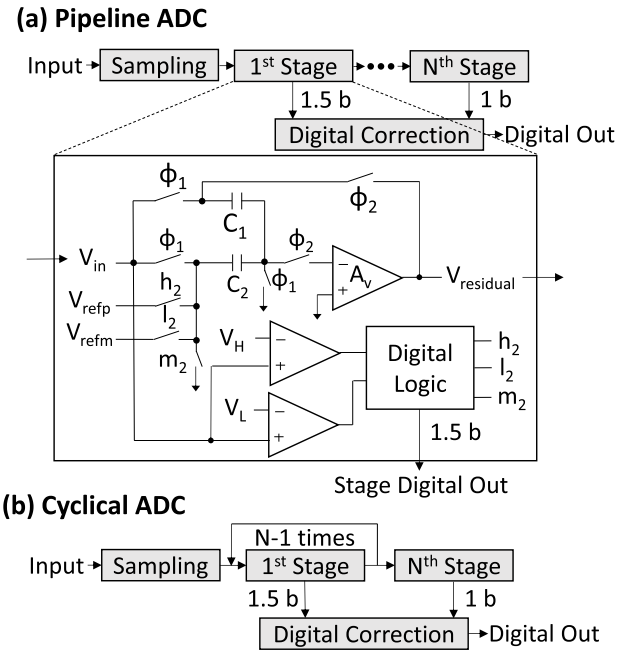

FIGURE 2. **(a) *N*-stage pipeline ADC. (b) Cyclic ADC both with 1.5-bit stages.**

compare the analog input against reference voltages that are generated by a digital-to-analog converter (DAC). All four architectures are suitable for use with an analog MVM but differ in their sensitivity to component variability.

There are design tradeoffs between speed, energy, and area for the four architectures. The exact tradeoffs are dependent on the implementation of the ADCs and will not be analyzed in detail in this work, which will mainly compare their sensitivity to component errors and their effect on neural network accuracy. The pipeline ADC has the highest conversion throughput but has a large area cost. The ramp ADC has a long latency but can efficiently convert all of the array's outputs in parallel using a global DAC and has the smallest per-column area cost. The SAR ADC is an intermediate case. It has a much shorter latency than a ramp ADC but has larger area and is harder to parallelize because the DAC cannot be used for multiple conversions simultaneously. Compared to the pipeline ADC, the SAR ADC has lower throughput but uses less area [18], [19], [20].

A. PIPELINE ADC

The pipeline ADC [shown in Fig. 2(a)] is a multistage ADC that uses a succession of 1.5-bit stages to enable high-resolution digitization of the analog input [18]. For example, in an 8-bit ADC, there are seven 1.5-bit stages and a final 1-bit stage. Each 1.5-bit stage decides whether an input voltage is high (h_2), low (l_2), or medium (m_2) relative to two reference voltages (subscript denotes that these signals are in phase with Φ_2). The redundancy of using three output levels rather than two ensures that the ADC's output is resilient to offsets in the comparators. Each stage then applies an

interstage gain (ideally equal to 2) and a shift that depends on the 1.5-bit output. The resulting residual voltage is then passed as the input to the next stage. The digital correction logic reduces the three redundant options from each stage down to two, using the results from the less significant stages. A pipeline ADC is a popular choice for medium- to high-resolution ADCs with high accuracy and throughput, owing to its robustness to comparator offsets and the ability to process multiple samples simultaneously in different stages.

Each 1.5-bit stage is composed of two subcircuits: the analog components producing the residual to be passed to the next stage and the digital logic producing the 1.5-bit output. The residual is given by

$$V_{\text{residual}} = \frac{V_{\text{in}}\left(1+\frac{C_1}{C_2}\right) - V_{\text{dig}}\left(\frac{C_1}{C_2}\right)}{1+1/A_v\beta_p} \tag{1}$$

where A_v is the operational amplifier gain, V_{in} is the input voltage, V_{dig} is one of the three reference voltages [V_{refp}, V_{refm}, 0] that is chosen by the digital logic, and $\beta_p = C_1/(C_1 + C_2 + C_{\text{par}})$. C_{par} is the parasitic capacitance at the negative input to the operational amplifier. In this case, $V_{\text{refp}} = -V_{\text{refm}} = V_{\text{REF}}$, and the range is $2V_{\text{REF}}$. In an ideal pipeline 1.5-bit stage, the following conditions are met: 1) $C_1 = C_2$ and 2) the operational amplifier has large enough gain A_v to accurately compute the residual based on the capacitance ratio.

B. CYCLIC ADC

A cyclic ADC is built using the same 1.5-bit stage as the pipeline, but instead of having N unique stages, it cycles through the same stage $N-1$ times and then has a comparator for the final stage. The advantage of the cyclic ADC compared to the pipeline ADC is a significant reduction in area due to the reuse of a single stage. The disadvantage is conversion throughput. The cyclic ADC has a throughput of one input per N clock cycles, compared to the pipeline ADC's throughput of one input per one clock cycle.

C. RAMP ADC

A ramp ADC uses a comparator to compare the input voltage to a reference voltage ramp signal that is synchronized to an N-bit digital counter [19]. When the comparator flips, the value of the N-bit counter at that moment is latched to an N-bit register, and this value is the digital output. In the implementation shown in Fig. 3, an N-bit counter is passed into a DAC to produce the reference signal. A ramp ADC is well suited for analog IMC because the counter and DAC can be shared across the whole array, and only a comparator and an N-bit latch are needed per column [21]. This allows the analog outputs from the entire array to be digitized in parallel, thus reducing the energy and area footprint per column.

D. SAR ADC

An SAR ADC works similar to a ramp ADC. It uses a comparator and a DAC to find the N-bit digital value that most closely approximates the analog input signal. However, while the ramp ADC checks every possible N-bit digital value in ascending order until it reaches one that matches the analog input, the SAR ADC successively refines the digital value one bit at a time, effectively implementing a binary search algorithm [20]. An SAR ADC is desirable because it is relatively fast, with a latency of only N cycles (verusu 2^N cycles for the ramp ADC), with similar accuracy to the ramp. The downside of the SAR ADC is that it cannot simultaneously share the DAC across all the columns like a ramp ADC can, but it still consumes less area than the pipeline ADC.

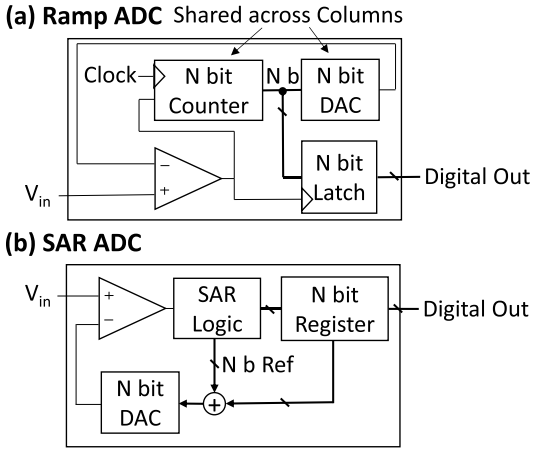

FIGURE 3. **(a) *N*-bit ramp ADC. (b) *N*-bit SAR ADC.**

E. DAC DESIGN

We consider three different switched-capacitor DACs in our analysis, shown in Fig. 4, which are similar in principle but are different in their implementation [22], [23]. All three use a radix-2 array of capacitors, where C_1 is the minimum-sized unit capacitor. The most commonly used capacitive DAC design is shown in Fig. 4(a). During the first half cycle, each bit of the digital input to the DAC controls whether each of the N capacitors is charged. In the second half cycle, the charge is transferred to the output capacitor to produce an output voltage given by

$$V_{\text{out}} = \frac{\frac{1}{2^N C_1}\left(V_{\text{refp}} - V_{\text{refm}}\right)\sum_i^N b_i 2^{N-i} C_1 + V_{\text{refm}}}{1+1/A_v\beta_1} \tag{2}$$

where $b_1, \ldots, b_N$ are the input bits from most significant bit (MSB) to least significant bit (LSB), N is the number of bits, V_{refp} and V_{refm} are the positive and negative reference voltages, and $\beta_1 = 2^N C_1/[2^N C_1 + \Sigma_i(b_i 2^{N-i} C_1)]$. For the pipeline ADC, $V_{\text{refp}} = -V_{\text{refm}} = V_{\text{REF}}$, and the range is $2V_{\text{REF}}$. The 2^N digital levels are ordered by treating the N-bit digital value as an unsigned integer. The capacitance matching and the amplifier gain are the two main considerations to produce an accurate DAC. Fig. 5 (blue) shows how errors induced by capacitance mismatch vary as a function of the analog output value. The error increases monotonically with the output value because larger inputs cause more capacitors

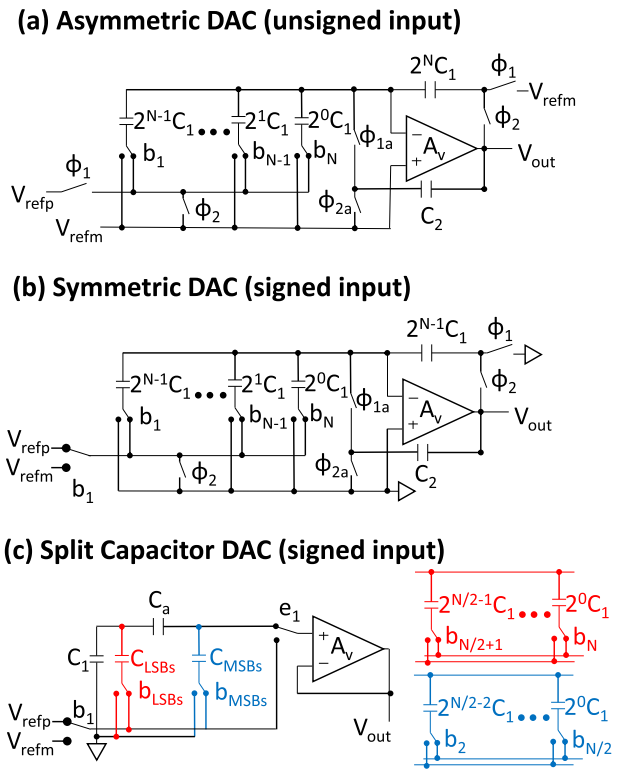

FIGURE 4. **(a) *N*-bit switched-capacitor asymmetric DAC. (b) Symmetric DAC. (c) Split capacitor DAC. Red/blue components on the left are shorthand for the subcircuits on the right.**

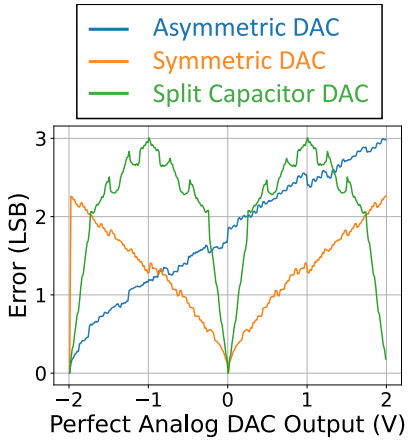

FIGURE 5. **Average error for 100 8-bit DACs with 16% standard deviation of capacitance mismatch as a function of a perfect 8-bit DAC output.**

to be switched on, and thus, a larger error accumulates from the individual capacitor errors.

As we will explain in Section IV, for neural network applications, it is highly desirable for the error to be minimized for analog values in the middle of the DAC's range. To achieve this, we use the modified DAC design in Fig. 4(b), which we will refer to as the "symmetric" DAC. First, the N-bit digital input is treated as a two's complement signed value rather than an unsigned value. Second, instead of always referencing V_{refp} during the input period and V_{refm} during the output period, either V_{refp} or V_{refm} is used during the input period (depending on the signed bit) and 0 V is used as a reference during the output period. Finally, logic is used to take the magnitude of the digital input and use those bits to control the capacitors. This results in minimum errors for inputs close to 0 V due to no capacitors being switched on, and errors increasing with the magnitude of the input in either direction as additional capacitors are switched on. This can be seen in Fig. 5 (orange), where the errors for the symmetric DAC are perfectly symmetric around a 0-V analog value.

The output voltage of this DAC is given by

$$V_{\text{out}} = \frac{\frac{1}{2^{N-1}C_1}\sum_i^N b_i 2^{N-i} C_1 V_{\text{dig}}}{1 + 1/A_v\beta_2} \tag{3}$$

where $b_1, \ldots, b_N$ are the input bits from MSB to LSB in a two's complement representation, N is the number of bits, V_{dig} is one of the reference voltages [V_{refp}, V_{refm}] chosen by the MSB, and $\beta_2 = 2^{N-1}C_1/[2^{N-1}C_1 + \Sigma_i(b_i 2^{N-i}C_1)]$. Notice that if each of the bits is zero, then none of the capacitors affect the output and the output is 0 V.

The split capacitor DAC [Fig. 4(c)] significantly reduces the size of the capacitors needed [23]. This has a large impact for the SAR ADC, though less important for the ramp ADC, which shares one DAC across all columns. The capacitance reduction is done by splitting the capacitors into two groups: an LSB half and an MSB half. These two groups are joined by an attenuation capacitor, which allows the LSB and MSB halves to use similarly sized capacitors, despite the difference in their bit significance.

This DAC takes a two's complement digital input, similar to the DAC in Fig. 4(b). The output voltage is governed by

$$V_{\text{out}} = \frac{\frac{C_{\text{lsbs}}C_a + C_{\text{msbs}}(C_{\text{LSB}}+C_a)}{(C_{\text{LSB}}+C_a)(C_{\text{MSB}}+C_a) - C_a^2} V_{\text{dig}}}{1 + 1/A_v} \tag{4}$$

where C_{lsbs} (C_{msbs}) is the sum of the capacitances selected in the LSB (MSB) group, C_{LSB} (C_{MSB}) is the total capacitance in the LSB (MSB) group, C_a is the attenuation capacitor, and V_{dig} is one of the reference voltages [V_{refp}, V_{refm}] chosen by the MSB. There is one exception that is handled by the switch controlled by e_1, when the minimum code is passed. The attenuation capacitor is set by the following:

$$C_a = \frac{2^{\text{floor}\left(\frac{N}{2}\right)}}{2^{\text{floor}\left(\frac{N}{2}\right)} - 1} C_1. \tag{5}$$

III. ADC SIMULATION

For all four ADC topologies analyzed in this work, the accuracy is affected by capacitance mismatch, finite operational amplifier gain, and input offset voltage in the comparator.

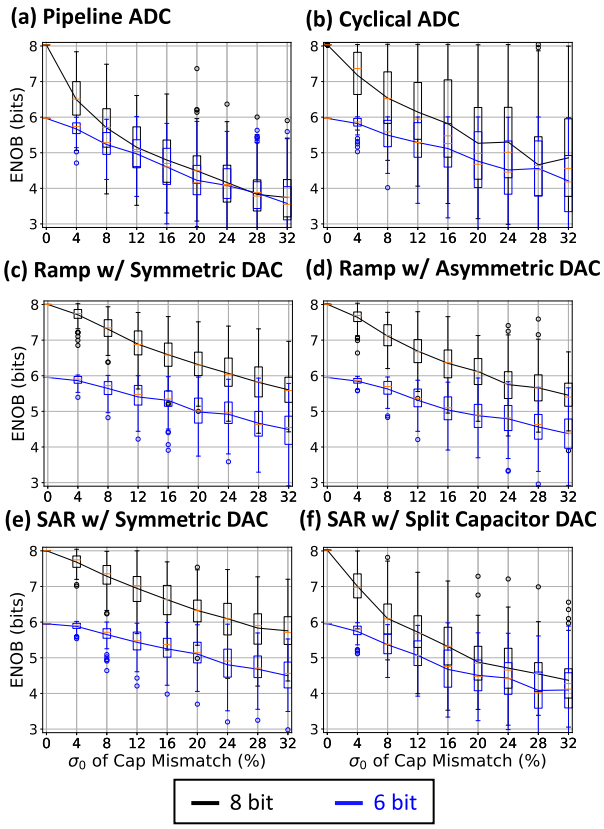

FIGURE 6. ENOB versus standard deviation of capacitance for (a) pipeline ADC, (b) cyclical ADC, (c) ramp ADC with symmetric DAC, (d) ramp ADC with asymmetric DAC, (e) SAR ADC with symmetric DAC, and (f) SAR ADC with split capacitor DAC. Box shows median, lower, and upper quantile. Whiskers show range excluding outliers (circles).

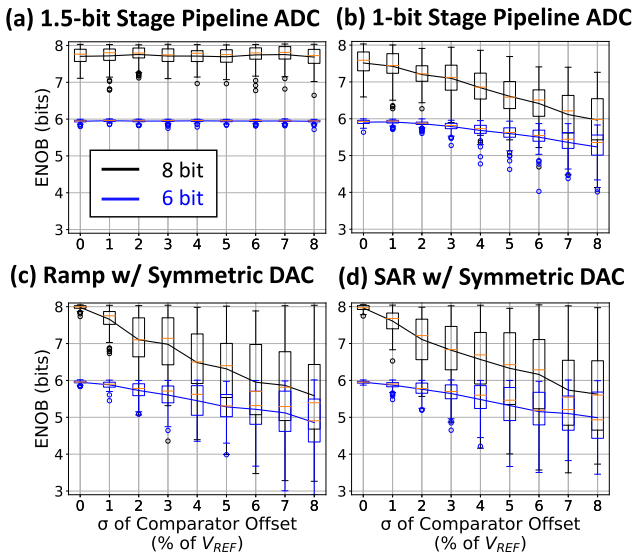

FIGURE 7. ENOB versus standard deviation of comparator offset for (a) pipeline ADC with 1.5-bit stages, (b) pipeline ADC with 1-bit stages, (c) ramp ADC with symmetric DAC, and (d) SAR ADC with symmetric DAC, for both 8-bit and 6-bit designs.

Fig. 6 shows how each ADC's effective number of bits (ENOB) depends on capacitance mismatch, while Fig. 7 plots the ENOB versus comparator offset. The effect of gain is shown in the supplementary information. A gain larger than 50 dB provides good accuracy for all 8-bit topologies. ENOB is the number of bits the ADC can resolve above the noise floor; for example, if an 8-bit ADC has an ENOB of 7 bits, the LSB can be treated as noise. To calculate ENOB, the signal-to-noise-and-distortion ratio (SNDR) was calculated for each ADC and converted to ENOB using the following equation [24]:

$$\text{ENOB} = \frac{\text{SNDR (dB)} - 1.76}{6.02}. \tag{6}$$

Notably, the ENOB is an aggregate metric over the entire input range, and it does not resolve the dependence of ADC errors on the input voltage.

A. CAPACITANCE MISMATCH

To simulate the sensitivity of each ADC to capacitance mismatch, the values of the component capacitors were randomly sampled from a normal distribution, whose standard deviation was varied. In a typical foundry process, the capacitance mismatch scales inversely with the square root of width times length [25]

$$\sigma_{\text{mismatch}}(\%) \approx \sigma_0(\%) \times \frac{\sqrt{W_0 * L_0}}{\sqrt{W * L}} \tag{7}$$

where σ_0 is the standard deviation of mismatch for the minimum capacitor with dimensions W_0 and L_0. In the following results, we refer to the value of σ_0 as the capacitance mismatch and this mismatch is scaled for larger capacitors in the DAC according to (7). While the standard deviation of capacitance mismatch was swept, the gain was held constant at 70 dB and the comparator offset was 0 V. Each ADC was evaluated at 6-bit and 8-bit resolution, which were chosen based on the evaluation neural network in Section IV (ResNet50 on ImageNet). Previous work has shown that the accuracy of ResNet50 is nearly unaffected by output quantization to 8-bits, and it remains within a few percentage points of ideal at 6-bits, as long as the ADC's range is optimized [4].

As shown in Fig. 6, the SAR ADC with the symmetrical DAC and the two ramp ADCs are the least sensitive to capacitance mismatch. The pipeline, cyclic, and SAR with the split capacitor DAC are the most sensitive. This is due to the size of the capacitors used in each ADC. The three that perform well all have large capacitors (that scale exponentially with N). The pipeline and cyclic ADCs only use minimum sized capacitors, and the SAR with the split capacitor DAC uses an attenuation capacitor that is very close to minimum size. Equation (7) shows that the capacitance mismatch decreases for the larger capacitors. Therefore, the ADCs with larger capacitors experience a smaller loss of precision for a given level of capacitance mismatch σ_0.

There is no difference in sensitivity between the SAR and ramp ADCs with the symmetric DACs. All the capacitance

error in these ADCs is from the DACs, so because they use the same DAC, they have the same ENOB. The pipeline and cyclic ADCs also have very little difference, because they have the same fundamental stages. The variability of the cyclic is higher than the pipeline as the cyclic ADC only has one set of capacitors, while the pipeline has $N - 1$ sets of capacitors. The additional capacitors help reduce the variability in the pipeline ADC.

B. COMPARATOR OFFSET

To evaluate the effect of comparator offset, each comparator's input offset voltage was assigned a random value from a normal distribution with zero mean. The standard deviation of the offset voltage for all comparators was varied to produce the results in Fig. 7. Based on the offset, the comparators would then switch at a voltage that is lower or higher than the correct value. For the ramp and SAR ADCs, this directly impacts ENOB as the output is directly shifted by this offset. The 1.5-bit stage pipeline ADC is designed to mitigate the effect of significant comparator offsets. To illustrate this advantage, a 1-bit stage pipeline ADC, which has smaller area but lacks the redundancy provided by the 1.5-bit stage, was included for comparison.

The effects of comparator offset on the 1-bit stage pipeline, ramp, and SAR ADCs are all very similar, as seen in Fig. 7. Comparator offsets directly impact the outputs of these ADCs, as explained above. The 1.5-bit stage pipeline ADC is the outlier as it is largely unaffected by the comparator offset. The reason is that the 1.5-bit stages resolve each bit into three categories: high, middle, and low. The high and low categories correspond to one and zero, respectively, while the middle category corresponds to an uncertain state. The uncertain state is resolved using digital correction, which considers the lower bits. This benefits from the lower stages' higher total gain. The only way for comparator offset to affect the 1.5-bit stage would be if the offset was larger than the range of the middle category. In other words, the comparator offset needs to be large enough to switch a zero to a one or vice versa, completely skipping the uncertain category. The 1-bit stage pipeline ADC lacks this critical resilience to comparator offset. Therefore, for the pipeline ADC, we consider only the design with 1.5-bit stages in Section III-B.

IV. NEURAL NETWORK SIMULATION

We now investigate the effect of the nonidealities in Section III on neural network inference using an analog accelerator. The accuracy simulations of analog IMC hardware were conducted using CrossSim [26]. The compact models for the ADCs, based on the equations in Section II, were integrated into CrossSim as behavioral models and were used to digitize every intermediate MVM result in the neural network.

The ResNet50 network on the ImageNet dataset, shown in Fig. 8(a), was used as a benchmark [27], [28]. Two other datasets (MNIST and CIFAR-10) were compared in the supplementary information, and ImageNet was the most

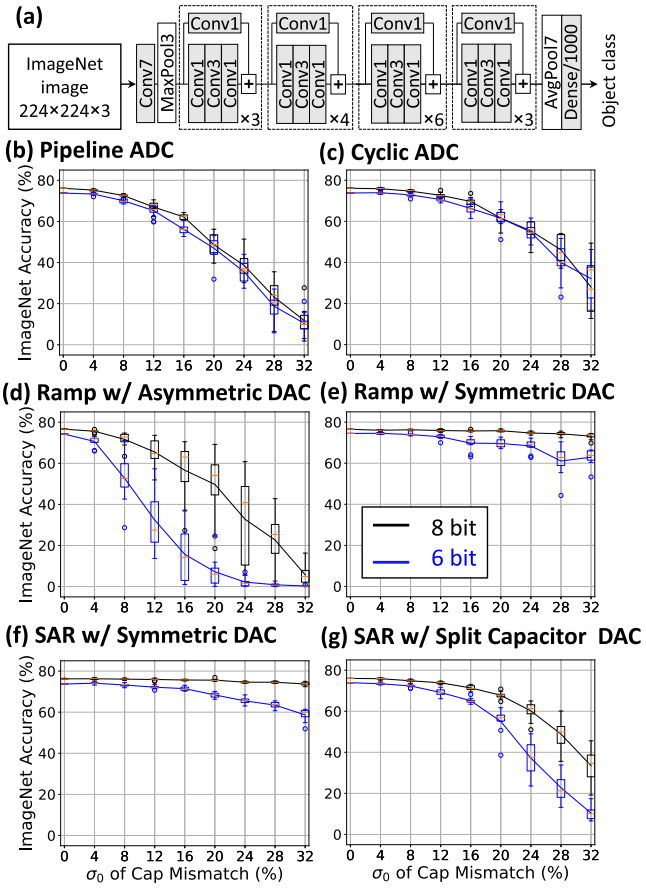

FIGURE 8. (a) ResNet50 architecture for ImageNet classification. Mean ImageNet accuracy (ten runs) versus capacitance mismatch with (b) pipeline ADC, (c) cyclic ADC, (d) ramp ADC with asymmetric DAC, (e) ramp ADC with symmetric DAC, (f) SAR ADC with symmetric DAC, and (g) SAR ADC with a split capacitor DAC.

susceptible to component variation in the ADC. For computational tractability, the following studies considered the classification of 1000 images from the ImageNet test set.

In order to isolate the effect of ADC errors, we assumed that weights are quantized to 8 bits and then mapped without error to memory device conductance values. We also assumed that positive and negative weights are encoded using a difference of two conductances and that these contributions are subtracted in analog prior to the ADC. The ADC input range (or equivalently, level spacing) for each layer was optimized via calibration [4]. Each array was limited to 1152 rows, with larger matrices being partitioned across multiple arrays with separate ADCs that produce digitized partial sums. The partial sums are then summed digitally. For ResNet50, this mapping results in a total of 1.2×10^7 ADC conversions simulated per input image.

Notably, our simulations include only the nonidealities in the ADC described in Section III and assume that no errors are introduced by adjacent components in the signal path, such as switches and column multiplexers, whose properties

are highly implementation dependent. The amplifier gain is held at 70 dB for all of the following results, which ensures that the accuracy is unaffected by the gain, as shown in the supplementary information.

It may be possible to digitally correct for ADC errors if the data dependence of the error in each fabricated ADC is characterized and stored in a lookup table. Our work did not assume that any digital correction was done, but additional work on digital correction schemes may help to reduce the effect of ADC component errors. The overhead of digital correction may be significant, so the improved accuracy may come at the cost of area, power, and speed.

A. CAPACITANCE MISMATCH

Fig. 8(b)–(g) shows the ImageNet accuracy using the pipeline ADC, cyclic ADC, ramp ADC (with symmetric and asymmetric DAC), and the SAR ADC (with symmetric DAC and split capacitor DAC) under varying degrees of capacitance mismatch. The value of all capacitances is randomly drawn, separately for every ADC in every array used to implement ResNet50, at the start of each inference simulation on 1000 images. To obtain the variability in accuracy due to capacitance mismatch, the simulation was repeated ten times per value of mismatch (σ_0), where all the capacitances were resampled on each run. We assumed that each pipeline or SAR ADC is shared across ten columns via time multiplexing. For these results, the comparator offset was set to 0 V.

For the pipeline ADC, there is very little difference in error sensitivity between the 6-bit implementation and the 8-bit implementation. Both start out at ∼75% accuracy before gradually decreasing with increasing capacitance mismatch. The two implementations have significantly different ENOBs only at small capacitance mismatch when the ENOB $>$ 5 bits, as shown in Fig. 6(a). In this regime, the ImageNet accuracy is not strongly sensitive to ENOB. This is consistent with prior studies, which showed that the accuracy of ResNet50 is relatively insensitive to a quantization resolution of 5 bits or more [4]. Then, when the capacitance mismatch is large enough to affect the accuracy, the two implementations have converged to the same ENOB [as shown in Fig. 6(a)], thus causing the similarities in accuracy in Fig. 8(b). The cyclic ADC is slightly more resilient than the pipeline ADC but is overall very similar.

The accuracy of the 8-bit pipeline ADC and the 8-bit ramp ADC with the asymmetric DAC responds similar to capacitance mismatch. This is surprising, since over the same range of mismatch, the 8-bit ramp ADC's ENOB drops much less than the 8-bit pipeline ADC's ENOB. This is due to the difference in the input value dependence of the errors in the two ADCs, which is not captured by the ENOB metric. Fig. 9 shows the mean error of the output of the ADCs as a function of the input voltage. As mentioned in Section II-D, the distribution of activation values in many neural networks follows a heavily skewed distribution with most of the values close to zero [29]. During inference, this means that the inputs to each ADC are also heavily clustered around zero volts [10].

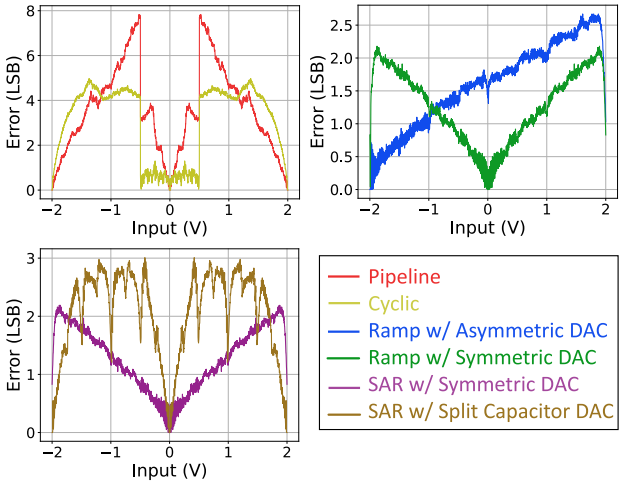

FIGURE 9. Output error for 8-bit ADCs with 16% standard deviation of capacitance mismatch as a function of the input voltage. The error is a mean of 100 runs with randomized capacitance values.

The ramp ADC with the asymmetric DAC has a lower overall error than the pipeline ADC (as expected from Fig. 6), but close to zero, the pipeline ADC actually has less error.

When a symmetric DAC is used, both the ramp ADC and the SAR ADC are minimally affected by the capacitance mismatch. There is still a difference between the 6-bit and 8-bit implementations, but the overall tolerance to capacitance mismatch has increased, so the difference is less. The reason for this improvement is explained by Fig. 9, where the use of the symmetric DAC with the error profile shown in Fig. 5 leads to a minimum in the ADC error around a 0-V input.

The SAR ADC with the split capacitor DAC is more strongly affected by the capacitance mismatch than the SAR ADC with the symmetric DAC. This is due to the reduction in capacitance size of the split capacitor DAC and sensitivity to the value of the small attenuation capacitor. The cost of reducing the area is decreased resilience to capacitance mismatch, which can also be seen in the larger errors in Fig. 9 compared to the other DAC designs.

B. COMPARATOR OFFSET

We investigated the effect of comparator offset on the ImageNet accuracy for the 8-bit implementations of the pipeline ADC, ramp ADC with symmetric DAC, and the SAR ADC with symmetric DAC, using the method described in Section III-B. The capacitor mismatch standard deviation was held constant at 1% for these tests. These results are shown in Fig. 10.

For the ramp and SAR ADCs, the ImageNet accuracy is strongly affected by the comparator offset. Both the ramp and the SAR ADCs utilize the comparators to directly resolve the digital level, so any offset in the comparator will directly impact the resolution of the ADC. For a standard deviation of comparator offset of 3.5% V_{REF}, the ImageNet accuracy falls to only 10%. By comparison, the ImageNet accuracy for

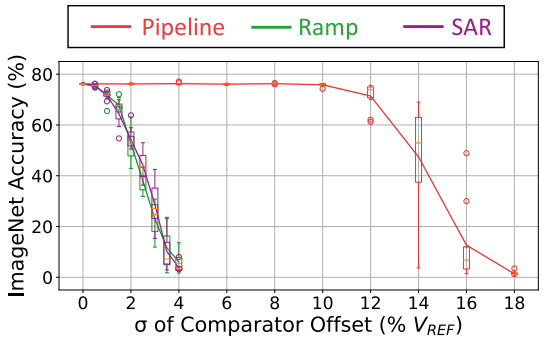

FIGURE 10. **Mean accuracy of ten runs of ImageNet with an 8-bit pipeline ADC, ramp ADC with symmetric DAC, and SAR ADC with symmetric DAC at different comparator offsets.**

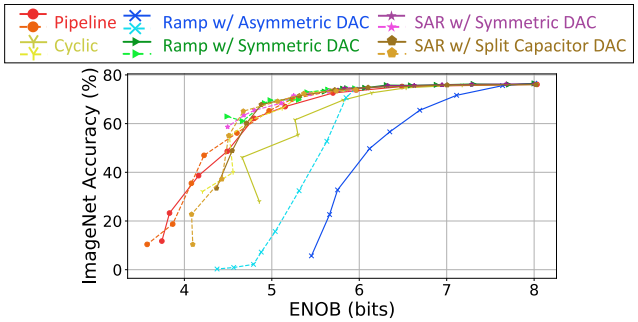

FIGURE 11. **Mean accuracy of ten runs of ImageNet compared to the ENOB for all the ADC topologies in Fig. 8. For each of the six topologies, the upper label in the legend is 8-bit and the lower label is 6-bit.**

the pipeline ADC is unaffected by comparator offset up to a standard deviation of comparator offset of 10% V_{REF}. This is due to the robustness to comparator offsets provided by the 1.5-bit pipeline stages, as discussed in Section III-C.

C. ENOB AS A PREDICTOR OF INFERENCE ACCURACY

Fig. 11 shows the relationship between the ENOB of the ADC and the ImageNet accuracy, by combining the results in Figs. 6 and 8(b)–(g). The pipeline ADC, cyclic ADC, ramp ADC with symmetric DAC, and both SAR ADCs have similar accuracy as a function of ENOB, and the dependence is similar to what has been shown in previous work that quantified ImageNet accuracy versus the number of ADC quantization bits [4]. The accuracy for these five ADC topologies is not significantly affected above an ENOB of 5 bits and then rapidly decreases with a projected $<1\%$ accuracy at 3 bits.

The ImageNet accuracy as a function of ENOB for the two ramp ADCs with an asymmetric DAC, however, is very different. These have a lower accuracy for the same ENOB compared to the other cases. Both the 8-bit and 6-bit version have low accuracy at 5 bits of ENOB. This deviation from the trend is due to the fact that unlike the other topologies, the ramp ADC with an asymmetric DAC does not have an error minimum at 0 V. This can be seen in Fig. 9. More generally, these examples show that the ENOB can be a poor metric to predict the effect of ADC errors on the accuracy of neural network inference. This is because the ENOB is a measure of the average error across the entire range of analog inputs, while the ImageNet accuracy depends critically on the distribution of these errors over the input values. Specifically, neural networks more heavily weigh errors around zero, due to the statistics of the intermediate data values present inside the algorithm. The ENOB is only useful as a predictor of neural network accuracy when comparing ADCs that are known to have a similar distribution of errors across input values. In the supplementary information, we propose the Gaussian weighted error, an alternative figure-of-merit that better accounts for the typical distribution of errors in a neural network and has a stronger correlation with accuracy.

V. CONCLUSION

The choice of the ADC implementation in analog IMC systems should consider not only speed, power consumption, and area but also the effect of the ADC on the accuracy of the algorithm. There are tradeoffs between accuracy and the other metrics, and how they should be weighted in the design of the ADC is highly dependent on the use case.

Our results showed that when considering the end-to-end neural network inference accuracy, accelerators built from different ADC architectures have different sensitivities to component mismatch inside the ADCs. For most other applications, the precision of an ADC can be adequately described using the widely known ENOB metric. However, as shown by our results, the ENOB is not generally a good predictor of the accuracy of neural network inference, because it gives equal weight to all the possible input values to the ADC. For neural network inference applications, the ADCs in an analog IMC system should preferentially minimize the error in the middle of its input range, which provides better matching to the statistics of intermediate data inside these algorithms. With a suitable DAC design, the ramp ADC and SAR ADC can both provide this key property. The pipeline and cyclic ADCs also intrinsically have this characteristic, while also offering superior robustness to comparator voltage offsets.

ACKNOWLEDGMENT

The employee owns all right, title, and interest in and to the article and is solely responsible for its contents. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this article or allow others to do so, for U.S. Government purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan https://www.energy.gov/downloads/doe-public-access-plan.

REFERENCES

[1] D. Silver et al., "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017.

[2] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.

[3] T. P. Xiao, C. H. Bennett, B. Feinberg, S. Agarwal, and M. J. Marinella, "Analog architectures for neural network acceleration based on non-volatile memory," *Appl. Phys. Rev.*, vol. 7, no. 3, Sep. 2020, Art. no. 031301.

[4] T. P. Xiao et al., "An accurate, error-tolerant, and energy-efficient neural network inference engine based on SONOS analog memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 4, pp. 1480–1493, Apr. 2022.

[5] A. Chen, M. Meneghini, D. Van Blerkom, F. Schanovsky, and T. Shaw, "Mechanisms and performance of metal oxide resistive RAM (RRAM)," in *Proc. IEEE Int. Integr. Rel. Workshop Final Rep.*, South Lake Tahoe, CA, USA, Oct. 2013, pp. 187–189.

[6] K. Seki et al., "An 80-ns 1-Mb flash memory with on-chip erase/erase-verify controller," *IEEE J. Solid-State Circuits*, vol. 25, no. 5, pp. 1147–1152, Oct. 1990.

[7] S. Lin et al., "Electrochemical simulation of filament growth and dissolution in conductive-bridging RAM (CBRAM) with cylindrical coordinates," in *IEDM Tech. Dig.*, San Francisco, CA, USA, Dec. 2012, pp. 26.3.1–26.3.4.

[8] Q. Dong et al., "15.3 A 351TOPS/W and 372.4 GOPS compute-in-memory SRAM macro in 7 nm FinFET CMOS for machine-learning applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2020, pp. 242–244.

[9] R. Genov and G. Cauwenberghs, "Charge-mode parallel architecture for vector-matrix multiplication," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 10, pp. 930–936, Oct. 2001.

[10] T. P. Xiao et al., "On the accuracy of analog neural network inference accelerators [feature]," *IEEE Circuits Syst. Mag.*, vol. 22, no. 4, pp. 26–48, 4th Quart., 2022.

[11] B. Jacob et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 2704–2713.

[12] V. Joshi et al., "Accurate deep neural network inference using computational phase-change memory," *Nature Commun.*, vol. 11, no. 1, p. 2473, May 2020.

[13] S. K. Gonugondla, C. Sakr, H. Dbouk, and N. R. Shanbhag, "Fundamental limits on the precision of in-memory architectures," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, San Diego, CA, USA, Nov. 2020, pp. 1–9.

[14] A. S. Rekhi et al., "Analog/mixed-signal hardware error modeling for deep learning inference," in *Proc. 56th ACM/IEEE Design Autom. Conf. (DAC)*. New York, NY, USA: Association for Computing Machinery, Jun. 2019, pp. 1–6.

[15] T.-J. Yang and V. Sze, "Design considerations for efficient deep neural networks on processing-in-memory accelerators," in *IEDM Tech. Dig.*, San Francisco, CA, USA, Dec. 2019, pp. 22.1.1–22.1.4, doi: 10.1109/IEDM19573.2019.8993662.

[16] H. Jiang, W. Li, S. Huang, S. Cosemans, F. Catthoor, and S. Yu, "Analog-to-digital converter design exploration for compute-in-memory accelerators," *IEEE Des. Test. Comput.*, vol. 39, no. 2, pp. 48–55, Apr. 2022.

[17] M. Le Gallo et al., "Precision of bit slicing with in-memory computing based on analog phase-change memory crossbars," *Neuromorphic Comput. Eng.*, vol. 2, no. 1, Mar. 2022, Art. no. 014009.

[18] P. Ghoshal and S. K. Sen, "Realization of a 1.5 bits/stage pipeline ADC using switched capacitor technique," in *Proc. Int. Conf. Intell. Control Power Instrum. (ICICPI)*, Kolkata, India, Oct. 2016, pp. 65–66.

[19] B. Gordon, "Linear electronic analog/digital conversion architectures, their origins, parameters, limitations, and applications," *IEEE Trans. Circuits Syst.*, vol. CS-25, no. 7, pp. 391–418, Jul. 1978.

[20] A. Shafiee et al., "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *Proc. ACM/IEEE 43rd Annu. Int. Symp. Comput. Archit. (ISCA)*, Seoul, South Korea, Jun. 2016, pp. 14–26.

[21] M. J. Marinella et al., "Multiscale co-design analysis of energy, latency, area, and accuracy of a ReRAM analog neural training accelerator," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 1, pp. 86–101, Mar. 2018.

[22] M. Saberi, R. Lotfi, K. Mafinezhad, and W. A. Serdijn, "Analysis of power consumption and linearity in capacitive digital-to-analog converters used in successive approximation ADCs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 8, pp. 1736–1748, Aug. 2011.

[23] Md. T. Shahed and A. B. M. H. Rashid, "Design of a 10 bit low power split capacitor array SAR ADC," in *Proc. IEEE Int. Conf. Signal Process., Inf., Commun. Syst. (SPICSCON)*, Dhaka, Bangladesh, Dec. 2021, pp. 74–77.

[24] B. Murmann. (2022). *Introduction to ADCs/DACs: Metrics, Topologies, Trade Space, and Applications*. [Online]. Available: https://github.com/bmurmann/ADC-survey

[25] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1439, Oct. 1989.

[26] T. P. Xiao et al. *CrossSim: Accuracy Simulation of Analog In-Memory Computing*. Accessed: Jul. 13. [Online]. Available: https://github.com/sandialabs/cross-sim

[27] *MLPerf Inference ImageNet Calibration Set*. Accessed: Jul. 13. [Online]. Available: https://github.com/mlcommons/inference/blob/master/calibration/ImageNet/cal_image_list_option_1.txt

[28] V. J. Reddi et al., "MLPerf inference benchmark," in *Proc. ACM/IEEE 47th Annu. Int. Symp. Comput. Archit. (ISCA)*, May 2020, pp. 446–459.

[29] D. Miyashita, E. H. Lee, and B. Murmann, "Convolutional neural networks using logarithmic data representation," 2016, *arXiv:1603.01025*.