Adaptive Multiply-Accumulate Unit with Scalable Critical Path for Aggressive Voltage Underscaling DNN Accelerators

Tongjing Wu[1], Siting Liu[2], Tong Li[1], Hui Wang[1], Zhigang Mao[1], Jie Han[3], Honglan Jiang[1*]
[1]School of Integrated Circuits, Shanghai Jiao Tong University, Shanghai, China
[2]School of Information Science and Technology, ShanghaiTech University, Shanghai, China
[3]Department of Electrical and Computer Engineering, University of Alberta, Alberta, Canada
{tongjing_wu, tongli_licu, ihuiwang, maozhigang, honglan}@sjtu.edu.cn
liust@shanghaitech.edu.cn, jhan8@ualberta.ca

ABSTRACT

Deep neural network (DNN) accelerators have widely been investigated for efficient inference, where matrix multiplication commonly implemented by multiply-accumulate (MAC) units dominates the power consumption. Underscaling the supply voltage of MACs can effectively reduce the power dissipation of DNN accelerators yet induces timing errors that may degrade the inference accuracy. In this paper, we propose a bit-width adjustment circuit (denoted as ADA) for arbitrary MAC units under aggressive voltage underscaling, which adaptively tunes the critical path as per input data patterns. Incurring 16.1% area overhead, the proposed ADA enables a MAC array to achieve zero accuracy loss under voltage underscaling ratios of 0.71 and 0.72 in the inferences of multi-layer perceptron (MLP) and convolutional neural network (CNN), respectively. While preserving accuracy, the MAC array equipped with ADA achieves up to 50.89% power reduction compared to that without ADA under nominal voltage operation. Furthermore, we propose ADA-Plus to specifically optimize the MACs in output stationary systolic arrays, which reduces the area of ADA by 23%. Compared with state-of-the-art error resilient circuit designs for voltage underscaling, to achieve a similar accuracy, the proposed adaptive MAC achieves the highest voltage underscaling ratio, resulting in 14.26%–48.44% power savings for the MAC array.

Keywords

DNN accelerator, multiply-accumulate (MAC) unit, voltage underscaling, low-power

1 INTRODUCTION

Deep neural networks (DNNs) have shown remarkable achievements across diverse areas such as computer vision [19], natural language processing [2], and speech recognition [11]. Accelerating the DNN inference is becoming increasingly crucial; thus, numerous accelerators have been developed, with TPU being a representative example [12]. The systolic array consisting of multiply-accumulate (MAC) units plays a key role in the compute engine of DNN accelerators, enabling high-throughput matrix multiplications for DNN workloads, which dominates the power consumption [13].

Owing to the enormous demand for MACs, energy efficient MAC units have widely been investigated to reduce the overall power consumed by DNN accelerators. A variety of techniques have been explored to enhance the hardware efficiency of MAC units from various perspectives, including logarithmic domain computing [8], quantization [7], and voltage underscaling [4]. Among them, voltage underscaling is particularly effective for power reduction since power dissipation decreases quadratically with supply voltage. However, when the voltage is aggressively lowered, the slowed transistor switching induces severe timing errors that can eventually cause catastrophic accuracy degradation in practical applications. A mainstream technique to address this issue is error detection and correction (EDaC), which detects timing errors and corrects them under voltage underscaling.

Many prior studies have shown that EDaC and EDaC-like circuits can effectively recover or mitigate timing errors in aggressive voltage underscaling DNN accelerators [3, 5, 9, 10, 18, 20, 22]. These circuits can generally be categorized into two types, gate-level and transistor-level. EDaC circuits in gate-level are easy to implement and highly portable across different designs. In [22], based on the observation that omitting a few multiplication results causes less degradation on accuracy than propagating incorrect partial sums, the outputs of processing elements (PEs) that encounter timing errors are excluded from the accumulation process. Instead of dropping the wrong multiplication results, [9] uses a delayed clock to sample the correct result after half of a clock period. However, the extensive use of Razor flip-flops [6] and Multiplexers (MUXs) in gate-level designs may incur a large area overhead. Moreover, buffer insertion is usually required in these designs for short path padding, leading to considerable area and power overheads [20]. On the other hand, EDaC circuits in transistor-level incur a much smaller area overhead. In [5], a self-calibrating tunable replica critical path system is proposed, which dynamically tracks real path delays across PVT variations to enable accurate timing error detection. Rather than tracking and recovering timing errors like most EDaC circuits, [20] predicts timing errors with internal signals in the MAC unit and switches the register clock to a delayed one. Transition detectors (TDs) used by these two designs for timing error detecting and predicting are implemented with customized transistor-level circuits, thereby significantly reducing the area overhead. Nevertheless, compared with designs fully implemented using gate-level standard cells, those employing customized transistor-level circuits are more complex to realize and less portable in chip implementation. In addition, both gate-level and transistor-level EDaC circuits depend on a delayed clock for timing error correction, which may incur many timing issues in practice.

In this paper, we propose two bit-width adjustment circuits (referred to as ADA and ADA-Plus) for MAC units under aggressive voltage underscaling, which fundamentally shorten the critical paths and thereby prevent timing errors at low voltages without relying on the

*Corresponding author
This work was supported in part by the National Natural Science Foundation of China under Grant 62374108.

DAC '26, Long Beach, CA, USA

ACM ISBN 979-8-4007-2254-7/2026/07
https://doi.org/10.1145/3770743.3804327

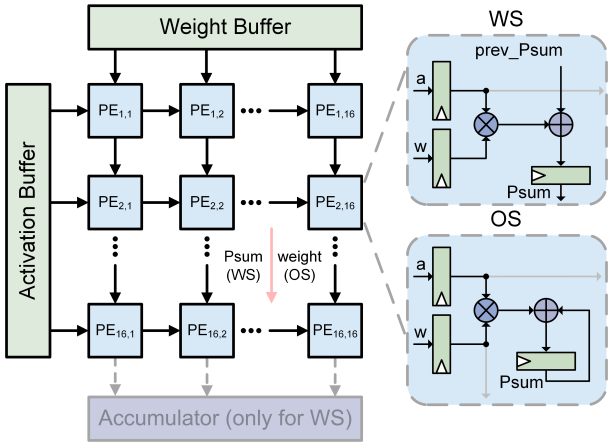


Figure 1: The schematic of systolic array and dataflow.

delayed clock or Razor flip-flops as in EDaC circuits. Specifically, the contributions of this work are as follows.

- We empirically show that sign transitions in the accumulation results can lead to long carry chains in the synthesized MAC netlist under voltage underscaling, even when the corresponding input patterns intuitively imply a short carry propagation chain.
- ADA is devised to activate the minimum bit width of adders as per the leading one or zero positions of the input addend, thereby shortening the critical path and avoiding timing errors of the MAC unit. ADA is completely implemented by gate-level standard cells and does not require a delayed clock, resulting in 30.3%–75.88% less area overhead than other gate-level EDaC designs.
- Taking advantage of the characteristics of the output stationary dataflow, we propose ADA-Plus that adjusts the bit width of adders for the next MAC operation based on the current accumulation result. ADA-Plus incurs 23% less area than ADA, with equal or higher performance in accuracy.
- We evaluate the proposed designs across different neural networks and datasets, achieving state-of-the-art hardware efficiency and accuracy. To the best of our knowledge, this is the first work to achieve adaptive critical path in MAC units operating at aggressively low voltages.

2 BACKGROUND AND MOTIVATION

2.1 Dataflow of Systolic Array

A systolic array is composed of a regular grid of PEs that perform MAC operations in a pipelined and highly parallel manner, as illustrated in Figure 1. In systolic arrays, two typical dataflow schemes are widely adopted: output stationary (OS) and weight stationary (WS). As shown in Figure 1, in the WS dataflow, weights are preloaded into local registers while the input activations flow horizontally. The partial sums are propagated vertically and accumulated across PEs while the final results are continuously produced from the bottom row of the array. In the OS mode, the input activations and weights are aligned in a parallelogram and propagate horizontally and vertically across the array each cycle, respectively. Each PE multiplies its activation and weight and meanwhile accumulates the multiplication result with the locally stored partial sum.

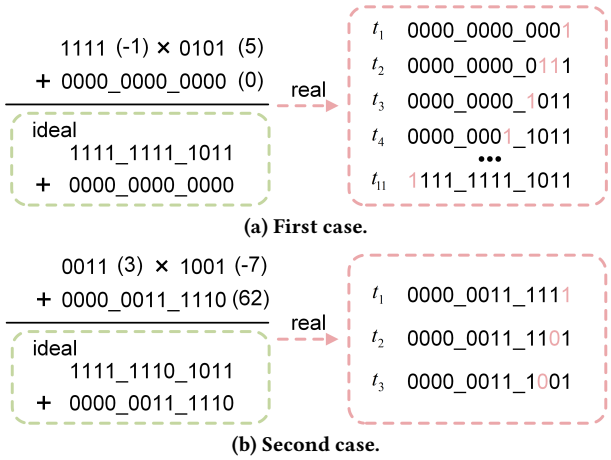


Figure 2: Diagrams of the carry chains in a synthesized MAC unit with a 4×4 multiplication and a 12-bit addition.

2.2 Carry Chain in Synthesized MAC Unit

In a MAC unit, the critical path is often determined by the large bit width addition. To investigate the carry chain, we perform AMS simulations on a MAC spice netlist with a 4×4 multiplication and a 12-bit addition under a voltage underscaling ratio of 0.89. Figure 2 shows two representative carry propagation behaviors.

In the first case shown in Figure 2a, ideally, the output of the multiplication is -5 and the carry chain in the adder should be short since no carry is generated at any bit. However, since the MAC netlist is flattened by the synthesis tool and the multiplier and adder are merged into a single carry-propagate structure to ensure high optimization, the actual carry behavior deviates from the ideal case. The simulation result shows that the zeros in the higher bit positions flip sequentially, resulting in a long carry chain. In the second case shown in Figure 2b, if the product were sign-extended before accumulation as in an ideal MAC, a long carry chain would appear. However, the six significant bits of the accumulation result are initially zero, and no transitions occur afterwards. In this case, the accumulation result settles within a short time.

These two cases demonstrate that due to logic optimization, the synthesized netlist under voltage underscaling may exhibit a different carry chain behavior from the ideal MAC implementation. A direct solution is to reduce the bit width of the accumulation result, thereby shortening the critical path fundamentally.

2.3 Necessity of Dynamic Bit-Width

We evaluate the distribution of required bit widths of accumulation results in MACs by deploying a six-layer CNN on a 16×16 systolic array for CIFAR-10 [14] classification. The distribution is obtained when executing the first convolution layer, where the required bit width for representing the accumulation result of each MAC operation is recorded. Also, we record the frequency of the long carry chain occurring for each required accumulation bit width. The long carry chain is defined as a path whose output becomes stable after three-quarters of a clock cycle. As shown in Figure 3, when a long carry chain occurs, the required bit width reaches up to 12 bits. However, most accumulation results require more than 12 bits in general. This observation aligns with the analysis in Section 2.2, i.e., small accumulation values along with sign inversions can produce long carry chains. Nevertheless, fixing the bit width to 12 bits in this layer to shorten the critical path would cause overflow for most results. Therefore, dynamic bit-width adjustment is essential.

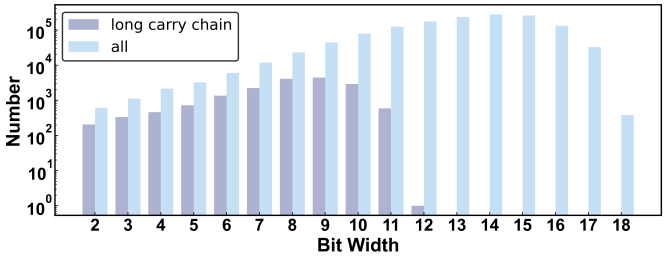
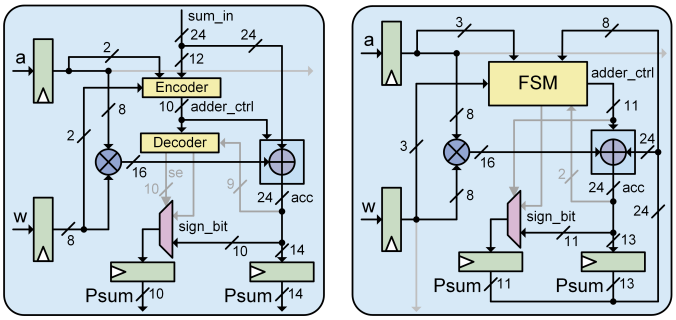


Figure 3: Distribution of the required bit widths for accumulation outputs in a convolution layer and the occurring frequency of long carry chains for each bit width.



(a) ADA. **(b) ADA-Plus.**

Figure 4: The overview of ADA and ADA-Plus.

3 ADA and ADA-Plus DESIGN

Figure 4 illustrates the MAC unit equipped with ADA and ADA-Plus, where the multiplication inputs are 8-bit and the addend input and output are 24-bit.

3.1 ADA Design for Universal Dataflows

As shown in Figure 4a, ADA consists of an encoder and a decoder. The encoder detects the patterns of the inputs, including the leading one and leading zero of the addend "sum_in," and generates the control signal "adder_ctrl" for the adders. The decoder generates the accumulation result selection signal "se" and the selected sign "sign_bit" for the accumulation result.

Figure 5 shows the design details of ADA. First, bit-XNOR operation is performed on the sign bit ("sum_in[23]") and the 12th to the 22nd bits of "sum_in" ("sum_in[22:12]") in the encoder, producing 1 when a bit matches the sign bit and 0 otherwise. The XNOR result between the "sum_in[22]" and the sign bit is denoted as "adder_ctrl[10]," while the XNOR results of the 14th to the 21st bits are denoted as the 0th to 7th bits of signal "LOZ." These signals are then fed into the leading zero detector (LZD). Inside the LZD, they are cascaded from the most significant bit to the least significant bit through a sequence of AND operations. This structure ensures that if a leading one or leading zero exists among "sum_in[22:14]," the corresponding XNOR result drives the associated AND output to 0, and all subsequent AND outputs are forced to 0.

The output of the LZD is an 8-bit "adder_ctrl" signal, which is used to control the activation and deactivation of the full adders (FAs). As illustrated in Figure 5, the lower 14 FAs from 0th to 13th are always on, while each FA from 14th onward is supplied through three PMOS switches controlled by the corresponding "adder_ctrl" signals. For example, the 14th FA is controlled by "adder_ctrl[2:0]," while the 15th FA is controlled by "adder_ctrl[3:1]." As long as any one of the corresponding three control signals is 0, the associated FA works normally. This ensures that when a leading one or leading zero is detected, the

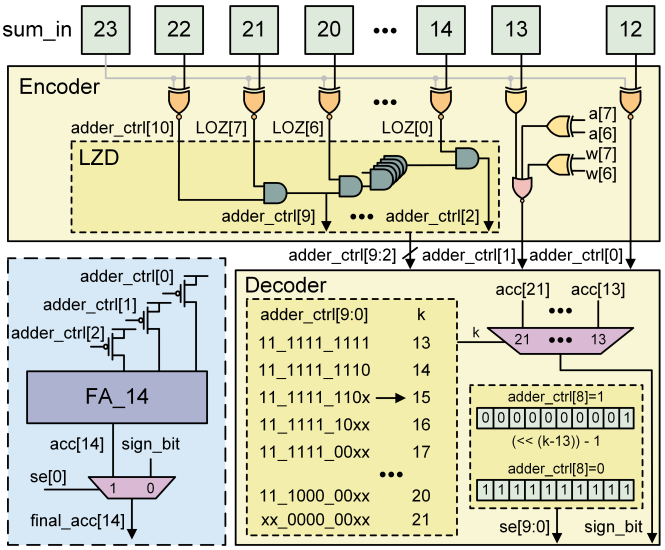


Figure 5: Circuit design of ADA.

FA corresponding to that bit position is activated along with its two higher-bit FAs and all lower-bit FAs. This effectively reserves one sign bit for the leading one or leading zero in "sum_in" and extends it by an additional bit to accommodate the multiplication output, thereby preventing overflow. For example, when a leading one appears at the 14th bit of "sum_in," the LZD generates an "adder_ctrl" signal of "1111_1110." In this case, the 14th to the 16th FAs are activated, while the FAs higher than the 16th bit remain disabled. Specifically, the 23rd FA shares the same control signals as the 22nd FA. This means that when a leading one appears at the 20th bit of "sum_in" or above, all FAs are activated.

It is noteworthy that "adder_ctrl[1]" and "adder_ctrl[0]" are not generated by the LZD. Instead, they are directly produced by applying an XNOR operation between their corresponding "sum_in" bits and the sign bit. This implies that when a leading one or leading zero appears within the lower 12 bits of "sum_in," all controllable FAs are disabled.

However, when only 14 FAs are activated, the valid range of the accumulation output "acc" is [-8192, 8191]. Since the inference network generally adopts symmetric quantization, the ranges of the input activations and weights are [-127, 127]. Therefore, the corresponding output range of the multiplication is [-16129, 16129], which requires 15 FAs to be computed correctly. Therefore, additional logic is incorporated in the generation of "adder_ctrl[1]" by checking the two most significant bits of the input activation "a" and weight "w." When these two bits are different, "adder_ctrl[1]" is forced to 0, which activates 16 FAs. In this case, the valid range of the accumulation output "acc" becomes [-32768, 32767], which ensures that no overflow occurs. When the condition for forcing "adder_ctrl[1]" to 0 is not met and all 14th to 23th FAs remain disabled, the output range of the multiplication and the range of "sum_in" are respectively [-4032, 4096] and [-4096, 4095]. Since the valid range of "acc" is [-8192, 8191] in this case, no overflow will occur in the accumulation result.

Signals "adder_ctrl[9:0]" are then fed into the decoder, which determines the sign bit selection signal "k." The value of "k" increments from 13 to 21, corresponding to the output bits from the 13th to the 21st FAs. Based on "k" and "adder_ctrl[8]," the decoder generates the accumulation result selection signal "se." The "se[9:0]" correspond to the selection signals for the final accumulation result "final_acc."

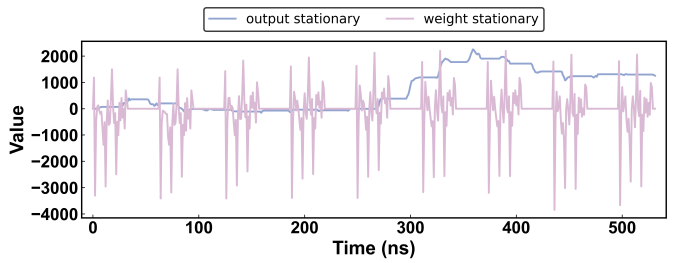


Figure 6: Trend of accumulation results in WS and OS PE.

When a bit of "se" is 0, the corresponding "final_acc" bit is assigned the value of "sign_bit." Otherwise, the corresponding "final_acc" bit is obtained from the associated FA. The generation logic of "se" is:

$$se = \begin{cases} (00\_0000\_0001 << (k-13)) - 1 & \text{if adder\_ctrl[8] = 1} \\ 11\_1111\_1111 & \text{if adder\_ctrl[8] = 0} \end{cases} \quad (1)$$

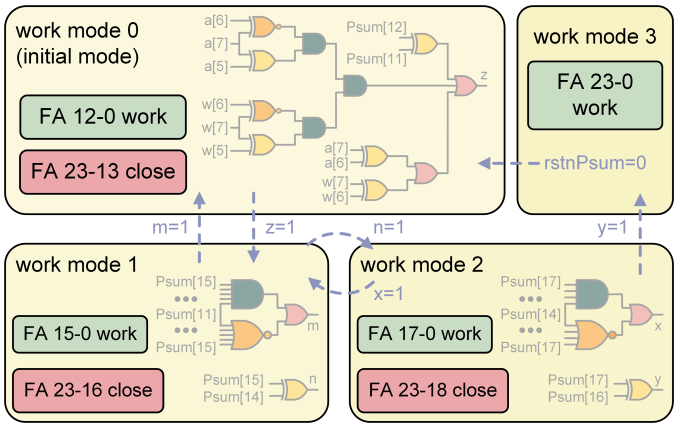
This ensures that the output of the highest activated FA is treated as the "sign_bit," and all accumulation bits above it are directly assigned this "sign_bit" rather than waiting for the corresponding FAs to produce valid results, thereby effectively shortening the critical path. As shown in Figure 5, when all the controllable FAs are disabled, "sign_bit" comes from the output of the 13th FA.

3.2 ADA-Plus Design for OS Dataflow

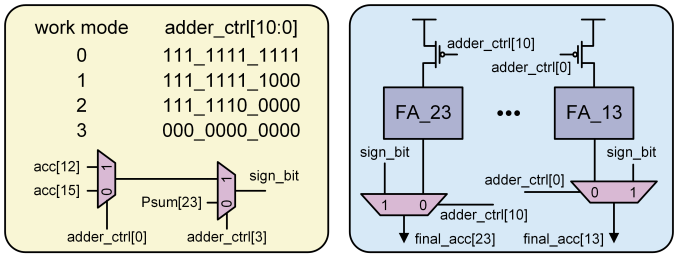
The dataflow mode of a systolic array affects the data patterns of the accumulation outputs. Figure 6 records the value trend of the accumulation output of a PE inside a 16×16 systolic array during the inference of a three-layer MLP on the MNIST dataset under both WS and OS modes. In the WS mode, the addend inputs are received from the PEs above, resulting in weak correlation among them. Consequently, significant fluctuations can be observed in the accumulation results. Since the array would wait for the bottom-row PEs to output all results before updating the weights, the accumulation results periodically drop to zero during inference. In the OS mode, the addend input of each PE originates from its accumulation result of the previous clock cycle, leading to a continuous transition of the partial sums. This continuity allows the bit width of the accumulation output to be adjusted gradually based on the current partial sum "Psum," without requiring a per-cycle leading one or leading zero detection as in ADA. Therefore, we propose ADA-Plus, which inserts a small finite state machine (FSM) into the MAC unit.

As shown in Figure 4b, "adder_ctrl" and "sign_bit" signals are generated by the FSM in ADA-Plus based on the "a," "w," "acc" and "Psum." In ADA-Plus, the 0th to 12th FAs remain always on, while the subsequent 11 FAs are gated by "adder_ctrl." Moreover, all the "adder_ctrl" signals are directly reused as the selection signals for the "final_acc" in the 13th to 23rd FAs, eliminating the need for the "se" signal used in ADA.

Figure 7a shows the details of the FSM in ADA-Plus. It consists of four work modes in total, where the initial mode enables only the 0th to 12th FAs, denoted as work mode 0. When the two bits of the registered accumulation result "Psum[12]" and "Psum[11]" are different, the 0th to 15th FAs are enabled. In addition, to prevent overflow caused by excessively large multiplication results, 16 FAs are enabled in the current cycle when either the magnitude of weight or activation is larger than or equal to 64 (i.e., the two most significant bits are different), or both larger than or equal to 32 (i.e., the three most significant bits are 110 or 001). In these cases, the FSM transitions to work mode 1 in the following cycle. In work mode 1, when "Psum[15]" and "Psum[14]" are different, the active FA range is expanded to the 0th to 17th bits by transitioning the FSM to work mode 2. Conversely,



(a) State transition diagram of the FSM.



(b) Adder control and sign bit selection logic.

Figure 7: Design details of ADA-Plus.

when "Psum[11]" to "Psum[15]" are all equal in work mode 1, the FSM returns to work mode 0 to shorten the critical path whenever possible. In work mode 2, the state transition conditions follow the same principle as in work mode 1. When "Psum[17]" and "Psum[16]" are different, the FSM advances to work mode 3. Conversely, when "Psum[14]" to "Psum[17]" are all equal, the FSM falls back to work mode 1. In work mode 4, based on the experimental results shown in Figure 3 and Figure 6, we assume that once "Psum" is large enough to trigger this mode, it will not decrease to the value that requires a small bit width. Therefore, work mode 4 does not revert to lower modes; instead, it remains active until the accumulation is completed and the "Psum" reset signal "rstnPsum" is received, after which the FSM returns directly to work mode 0.

Figure 7b shows the "sign_bit" selection logic and power control logic of FA in ADA-Plus. We assume that the sign bit of "Psum" remains unchanged when the FSM operates in work mode 2 or work mode 3, therefore "Psum[23]" is directly assigned to "sign_bit" in these modes. In contrast, when the FSM is in work mode 0 or work mode 1, "sign_bit" is taken from the output of the 12th or 15th FA, depending on the value of "adder_ctrl[0]." Moreover, in ADA-Plus, each controllable FA is gated by a PMOS switch driven by its corresponding "adder_ctrl" signal. As a result, "final_acc" can be selected directly using "adder_ctrl," leading to a significant area reduction.

4 EXPERIMENTS

In this section, we evaluate ADA and ADA-Plus both in an isolated MAC unit and in a systolic array applied for DNN acceleration. In section 4.1, we examine the FA gating and also compare the area overhead of ADA, ADA-Plus, and other EDaC designs. In section 4.2, we evaluate systolic arrays based on different MAC designs under voltage underscaling and assess accuracy, power, and PVT robustness. The baseline MAC unit is implemented in RTL with an 8-bit signed "a," an 8-bit signed "w" and a 24-bit signed "Psum." It is synthesized

Table 1: Minimum operating voltage and power reduction results across PVT corners for the proposed MAC-based systolic arrays.

corner	temp	$V_{\text{min, proposed}}$ (V)			$V_{\text{min, origin}}$ (V)		Power reduction (%)		
		ADA (OS)	ADA-Plus (OS)	ADA (WS)	OS	WS	ADA (OS)	ADA-Plus (OS)	ADA (WS)
tt	−40°C	0.68	0.67	0.68	0.76	0.76	19.59	18.24	16.55
	25°C	0.63	0.62	0.63	0.70	0.70	19.35	17.82	15.46
	125°C	0.58	0.57	0.58	0.67	0.67	25.64	25.32	21.94
ss	−40°C	0.76	0.75	0.76	0.84	0.84	17.68	16.35	14.97
	25°C	0.71	0.70	0.71	0.80	0.80	20.99	19.74	17.73
	125°C	0.65	0.64	0.65	0.76	0.76	27.02	26.05	24.53
ff	−40°C	0.60	0.60	0.60	0.67	0.67	19.81	16.66	15.15
	25°C	0.55	0.54	0.55	0.60	0.60	16.43	15.59	10.98
	125°C	0.49	0.48	0.49	0.58	0.58	29.19	28.96	24.48

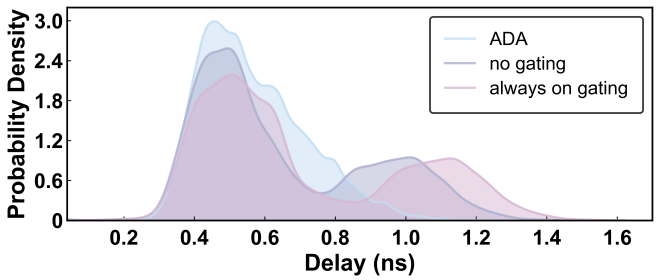


Figure 8: KDE curves of MAC output delay.

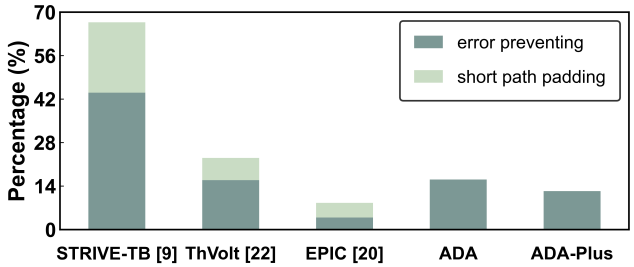


Figure 9: Area overhead of voltage underscaling schemes.

using Synopsys Design Compiler with a CMOS 28 nm standard cell library, whose nominal supply voltage is 0.9 V. All designs are derived from this baseline MAC netlist.

4.1 Evaluation of ADA and ADA-Plus

To control the activation of the FAs in ADA and ADA-Plus, PMOS power-gating switches are employed. To verify whether this gating affects the FA delay and evaluate the effectiveness of ADA, we perform AMS simulations on an ADA-based MAC spice netlist under a voltage underscaling ratio of 0.72. A total of 100,000 random input combinations are applied to the MAC, and the time required for the accumulation result to stabilize in each cycle is recorded. We then analyze the distribution of these delays and plot their Kernel Density Estimation (KDE) curves, as shown in Figure 8. The three KDE curves correspond to ADA, no gating, and always on gating, representing normal ADA operation, the original design without any gating, and the ADA-based circuit with PMOS switches permanently on, respectively. The results show that employing PMOS switches slightly increases latency when all switches are in the ON state. In addition, the critical path is effectively shortened with ADA, as evidenced by a substantial reduction in the proportion of large-delay cases in the KDE curve.

To compare the area overhead of different MAC designs for voltage underscaling, we synthesize their RTL implementations into gate-level netlists and estimate the resultant area relative to the baseline MAC. For STRIVE-TB [9] and ThVolt [22], we use Synopsys PrimeTime to insert buffers for short path padding under a 1 GHz timing constraint and then estimate their area. For EPIC [20], we estimate the area based on the size of the transistors used in the customized transistor-level circuits and the transmission gates used for short path padding. As shown in Figure 9, STRIVE-TB exhibits a substantially larger area of error preventing circuit than the other designs due to the extensive use of Razor flip-flops and MUXs. Moreover, although STRIVE-TB can correct timing errors on the full 24-bit accumulation result, it requires numerous buffers to avoid hold time violations, increasing its area compared with the other EDaC designs. ThVolt protects only the upper 8 bits of the accumulation result that achieves the highest inference accuracy on our evaluation. Consequently, its area is smaller than that of STRIVE-TB. Since no short path padding is required, ADA reduces area overhead by 75.88% compared with STRIVE-TB and by 30.3% compared with ThVolt, while incurring only a 16.1% total overhead. By removing the leading one and leading zero detection logic and simplifying the sign bit selection logic, ADA-Plus reduces the area overhead by 23% compared with ADA, resulting in a total overhead of 12.4%. Owing to the inherent area advantage of the customized transistor-level design, EPIC exhibits the smallest area overhead among all designs.

4.2 Applications

To assess MAC performance under voltage underscaling in DNN inference, this work follows the evaluation methodology proposed in [20]. The delay of each standard cell at various supply voltages is extracted using Synopsys HSPICE, which is then incorporated into Synopsys VCS for post simulation to obtain the final computation results.

We first evaluate the behavior of ADA and ADA-Plus under different PVT corners using VCS_XA for AMS simulation. Since AMS simulation is significantly slower than digital simulation, we deploy a $784 \times 16 \times 16 \times 10$ MLP on an 8×8 systolic array to perform inference on 8 images in MNIST for power evaluation. The power consumption of all PEs during the inference process is accumulated to obtain the final power. Using the methodology described earlier, the inference accuracy of the aforementioned systolic array under different PVT corners is evaluated. Table 1 shows the minimum operating voltage and power reduction results across PVT corners for the proposed MAC-based systolic arrays. The minimum operating voltages for the proposed and original systolic arrays, $V_{\text{min, proposed}}$ and $V_{\text{min, origin}}$, are defined as the voltages at which the accuracy loss on MNIST remains below 1%. Table 1 shows that the systolic array equipped with ADA maintains inference accuracy under substantial voltage underscaling in both OS and WS dataflows. Compared with the original design, it achieves up to 29.19% power reduction. For the OS dataflow, the ADA-Plus results in lower minimum operating voltages than ADA in most PVT corners,

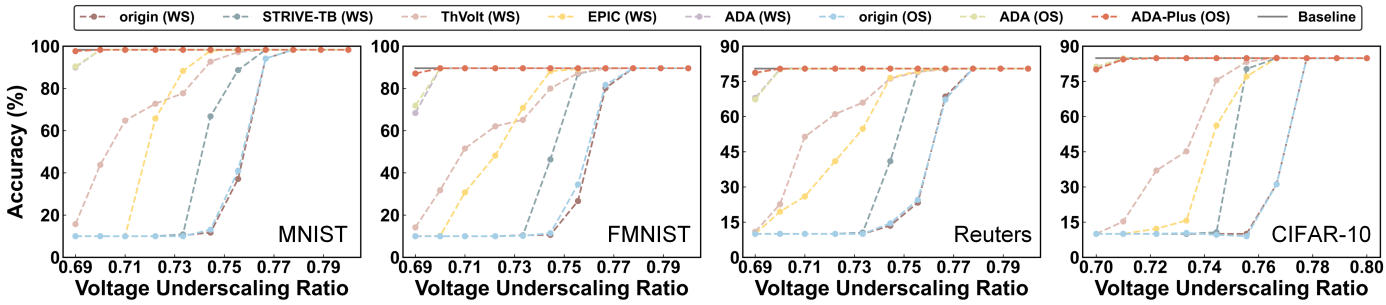


Figure 10: Inference accuracy of systolic array on DNN benchmarks with voltage scaling.

Table 2: DNN models and their error-free accuracy.

Benchmarks			Acc (%)
Dataset	Architectures		
MNIST [15]	MLP	784×256×256×10	98.30
FMNIST [21]		784×256×256×10	89.56
Reuters [1]		2048×256×256×46	80.45
CIFAR-10 [14]	CNN	CONV: (32, 32, 32)×(32, 32, 32) CONV: (16, 16, 64)×(16,16, 64) CONV: (8, 8, 128)×(8, 8, 128) FC: 2048×10	84.90

achieving up to 28.96% power reduction. It is also worth noting that the minimum operating voltage decreases with the increase of the temperature, due to the temperature inversion effect.

In addition, we deploy different neural networks on 16×16 systolic arrays based on various MAC designs and evaluate their inference accuracy under voltage underscaling, where all systolic arrays equipped with EDaC designs operate in the WS dataflow. The clock frequency is fixed at 1 GHz. Model configurations and their error-free accuracies are summarized in Table 2. In the CNN experiments, batch normalization is not used, which results in relatively low accuracy after quantization. Moreover, image-to-column (im2col) is performed in the CNN testbench so that the systolic array only computes matrix multiplications. Due to the large data volume generated by im2col, running inference on the full 10k-image test set would require more than one week. Therefore, we randomly select 1000 images for evaluation, where the error-free accuracy for this subset is 84.9%. As shown in Figure 10, both ADA-Plus and ADA applied to the two dataflows significantly outperform other designs under aggressively low voltages. With ADA and ADA-Plus, the systolic array sustains error-free inference at voltage underscaling ratios of 0.71 for MLPs and 0.72 for CNNs. In the MLP experiments, ADA-Plus delivers the best performance, which results in nearly no accuracy degradation even at a voltage underscaling ratio of 0.69, whereas the EDaC-based designs have already become completely unusable due to severe accuracy loss. In the CNN experiments, ADA-Plus and ADA deliver comparable performance, both achieving a minimum voltage underscaling ratio of 0.7. In contrast, systolic arrays employing prior EDaC designs can only operate reliably down to a voltage underscaling ratio of 0.76.

We then evaluate the power consumption of these designs using the same methodology as in our PVT-corner experiments. As shown in Figure 11, the shaded gray region in each power curve denotes the voltage range where the systolic array equipped with the corresponding MAC design incurs more than 1% accuracy loss on the MNIST dataset, which implies ineffective power consumption. Comparing the minimum effective power of each design, the systolic arrays incorporating ADA (OS), ADA (WS), and ADA-Plus (OS) reduce power consumption by 19.35%, 15.46%, and 17.82%, respectively, relative to the original

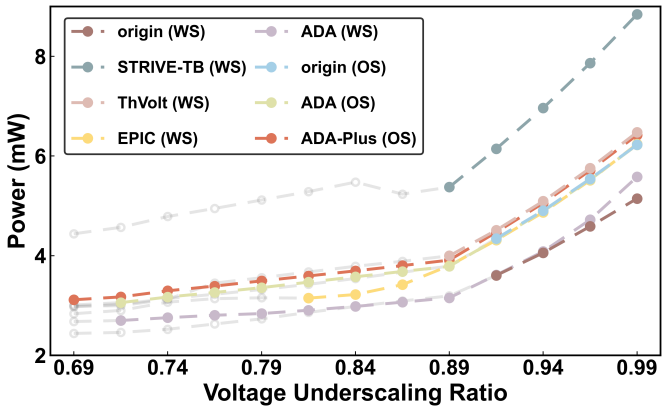


Figure 11: Power of systolic arrays consisting of the considered MAC designs under different voltage underscaling ratios.

systolic array. Compared with the original systolic array operating at the nominal supply voltage, the three designs achieve power savings of 50.89%, 47.55%, and 49.96%, respectively. Moreover, ADA achieves 48.44% and 28.72% lower power consumption than STRIVE-TB and ThVolt at their respective minimum effective voltages. By supporting a lower operating voltage, ADA reduces the power consumption of the systolic array by 14.26% relative to EPIC. Furthermore, to verify that using ADA in the MAC at low voltages is better than directly employing high speed MACs, we synthesize a MAC incorporating high speed multipliers and adders based on Synopsys DesignWare[16, 17]. The corresponding systolic array achieves a minimum voltage underscaling ratio of 0.76, with a power consumption of 3.04 mW that is 12.6% higher than that of the ADA-based systolic array.

5 CONCLUSION

This work presents two bit-width adjustment circuits that enable MAC units to operate reliably under aggressive voltage underscaling. By inferring the effective input operand range from the leading one or leading zero position and activating only the necessary FA bits, ADA shortens the critical path of MAC units using purely standard cells. It reduces the area overhead by 30.3%–75.88% versus prior gate-level EDaC schemes while adding only 16.1% area to a baseline MAC. An ADA-based systolic array sustains error-free inference for MLP and CNN at voltage underscaling ratios of 0.71 and 0.72 and delivers up to 50.89% power savings over the original design under nominal supply voltage. Leveraging output stationary dataflow, ADA-Plus further predicts the required bit width from the current partial sum, trimming ADA's area by 23% with equal or better accuracy. Across diverse networks and datasets, the array equipped with adaptive MACs achieves the lowest voltage underscaling ratio and 14.26%–48.44% array-level power reduction compared to other designs.

REFERENCES

[1] 2021. Reuters-21578 dataset. http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

[2] Chengwei Dai, Kun Li, Wei Zhou, and Songlin Hu. 2025. Capture the key in reasoning to enhance CoT distillation generalization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 441–465.

[3] Lishuo Deng, Junyi Qian, Zhengguo Shen, Cai Li, Jingchen Wang, Zhangrui Qian, Ziyu Li, Longning Qi, and Weiwei Shan. 2025. A Compound Timing Detection of Both Data Transition and Path Activation for Reliable In Situ Error Detection and Correction. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 33, 10 (2025), 2772–2782. doi:10.1109/TVLSI.2025.3578426

[4] Ronald G Dreslinski, Michael Wieckowski, David Blaauw, Dennis Sylvester, and Trevor Mudge. 2010. Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits. *Proc. IEEE* 98, 2 (2010), 253–266.

[5] Yuxuan Du, Junyi Qian, Zhengguo Shen, Chengjun Wu, Weiwei Shan, and Xi Wang. 2024. DSC-TRCP: Dynamically Self-Calibrating Tunable Replica Critical Paths Based Timing Monitoring for Variation Resilient Circuits. *IEEE Journal of Solid-State Circuits* (2024).

[6] Dan Ernst, Nam Sung Kim, Shidhartha Das, Sanjay Pant, Rajeev Rao, Toan Pham, Conrad Ziesler, David Blaauw, Todd Austin, Krisztian Flautner, et al. 2003. Razor: A low-power pipeline based on circuit-level timing speculation. In *Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003. MICRO-36.* IEEE, 7–18.

[7] Xinkuang Geng, Siting Liu, Leibo Liu, Jie Han, and Honglan Jiang. 2024. QUQ: Quadruplet Uniform Quantization for Efficient Vision Transformer Inference. In *Proceedings of the 61st ACM/IEEE Design Automation Conference* (San Francisco, CA, USA) *(DAC '24)*. Association for Computing Machinery, New York, NY, USA, Article 272, 6 pages. doi:10.1145/3649329.3656516

[8] Xinkuang Geng, Yunjie Lu, Hui Wang, and Honglan Jiang. 2025. Segment-Wise Accumulation: Low-Error Logarithmic Domain Computing for Efficient Large Language Model Inference. In *2025 Design, Automation & Test in Europe Conference (DATE)*. IEEE, 1–7.

[9] Noel Daniel Gundi, Zinnia Muntaha Mowri, Andrew Chamberlin, Sanghamitra Roy, and Koushik Chakraborty. 2023. STRIVE: Enabling Choke Point Detection and Timing Error Resilience in a Low-Power Tensor Processing Unit. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*. 1–6. doi:10.1109/DAC56929.2023.10247879

[10] Sangsu Jeong, Huiwon Yun, Dongkwon Lee, Sunwoo Lee, Minyoung Kang, and Dongsuk Jeon. 2025. An 83.16-TOPS/W Voltage-Scalable Time-Domain CNN Accelerator with Full-Swing Delay Cell and Gray-Code TDC in 28-nm CMOS. In *2025 IEEE Custom Integrated Circuits Conference (CICC)*. 1–3. doi:10.1109/CICC63670.2025.10982877

[11] Ya Jiang, Hongbo Lan, Qing Wang, and Shutong Niu. 2025. Multi-modal Streaming ASR in Cross-talk Scenario for Smart Glasses. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[12] Norman Jouppi, Cliff Young, Nishant Patil, and David Patterson. 2018. Motivation for and Evaluation of the First Tensor Processing Unit. *IEEE Micro* 38, 3 (2018), 10–19. doi:10.1109/MM.2018.032271057

[13] Sehoon Kim, Coleman Hooper, Thanakul Wattanawong, Minwoo Kang, Ruohan Yan, Hasan Genc, Grace Dinh, Qijing Huang, Kurt Keutzer, Michael W. Mahoney, Yakun Sophia Shao, and Amir Gholami. 2023. Full Stack Optimization of Transformer Inference: a Survey. arXiv:2302.14017 [cs.CL] https://arxiv.org/abs/2302.14017

[14] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[15] Y. LeCun and C. Cortes. 2010. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/.

[16] Synopsys. 2024. DesignWare DW01_add. https://www.synopsys.com/dw/ipdir.php?c=DW01_add.

[17] Synopsys. 2024. DesignWare DW02_mult. https://www.synopsys.com/dw/ipdir.php?c=DW02_mult.

[18] Yamilka Toca-Díaz, Rubén Gran Tejero, and Alejandro Valero. 2024. Ensuring the Accuracy of CNN Accelerators Supplied at Ultra-Low Voltage. In *2024 IEEE 42nd International Conference on Computer Design (ICCD)*. 92–95. doi:10.1109/ICCD63220.2024.00024

[19] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. 2025. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 5294–5306.

[20] Tongjing Wu, Xiaolu Hu, Tong Li, Siting Liu, Hui Wang, Weifeng He, Zhigang Mao, and Honglan Jiang. 2025. EPIC: Error PredIction and Correction for Power-Efficient Voltage Underscaling Multiply-Accumulate Unit. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*. 1–7. doi:10.1109/DAC63849.2025.11132540

[21] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).

[22] Jeff Zhang, Kartheek Rangineni, Zahra Ghodsi, and Siddharth Garg. 2018. Thundervolt: enabling aggressive voltage underscaling and timing error resilience for energy efficient deep learning accelerators. In *Proceedings of the 55th Annual Design Automation Conference*. 1–6.