A Low-Power Mixed-Precision Integrated Multiply-Accumulate Architecture for Quantized Deep Neural Networks

Xiaolu Hu*, Xinkuang Geng*, Zhigang Mao*, Jie Han†, Honglan Jiang*
* Department of Micro-Nano Electronics, Shanghai Jiao Tong University, Shanghai, China
† Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada
{huxiaolu, xinkuang, maozhigang}@sjtu.edu.cn, jhan8@ualberta.ca, honglan@sjtu.edu.cn

***Abstract*—As mixed-precision quantization techniques have been widely considered for balancing computational efficiency and flexibility in quantized deep neural networks (DNNs), mixed-precision multiply-accumulate (MAC) units are increasingly important in DNN accelerators. However, conventional mixed-precision MAC architectures support either signed×signed or unsigned×unsigned multiplications. The signed×unsigned multiplication enhancing the computing efficiency of DNNs with ReLU activations has never been considered in the design of mixed-precision MAC. Thus, this work proposes a mixed-precision MAC architecture supporting six operation modes, int8×int8, int8×uint8, two int4×int4, two int4×uint4, four int2×int2, and four int2×uint2. In this design, to balance the power and delay of different modes, the multiplication is implemented based on four precision-split 4×4 multipliers (PS4Ms). The accumulation is integrated into the partial product accumulation of the multiplication to eliminate redundant switching activities in separate compression. With 10% area reduction, the proposed MAC denoted as PS4MAC, reduces the power by over 35%, 42%, and 56% for 8-bit, 4-bit, and 2-bit operations, respectively, compared with the design based on the Synopsys DesignWare (DW) multipliers. Additionally, it achieves over 23% power savings for 8-bit operations compared to state-of-the-art (SotA) mixed-precision MAC designs. To save more power, an approximate computing mode for 8-bit multiplication is further designed, resulting in a MAC unit enabling eight operation modes, referred to as PS4MAC_AP. Finally, output-stationary systolic arrays (SAs) are explored using the above-mentioned MAC designs to implement DNNs operating under a 1 GHz clock. Our designs show the highest energy efficiency and outstanding area efficiency in all 8-bit, 4-bit, and 2-bit operation modes. Compared with the traditional SA with high-precision-split multipliers, PS4MAC_AP improves the energy efficiency for 8-bit operations by 0.6 TOPS/W, and PS4MAC achieves 0.4 TOPS/W - 0.7 TOPS/W improvement for all operation modes.**

***Index Terms*—mixed-precision, low-power, integrated MAC, accuracy-configurable**

I. Introduction

While a new era of artificial intelligence (AI) evolves through large language models (LLMs), traditional AI applications like image classification and speech recognition continue to deeply impact our daily lives. With the growth in application complexity and accuracy requirements, deep neural network (DNN) architecture is becoming increasingly large. Multiply-accumulate (MAC) operations account for over 99% of the computations [1] in a DNN implementation, making it a dominant factor in the power efficiency of DNN accelerators. Therefore, the development of power-efficient MAC units is crucial for DNN deployment, especially for edge applications.

MAC designs have been tailored to both floating-point and fixed-point computations [2]–[4]. With quantization techniques being extensively explored to reduce DNN model size and computational complexity, MAC architectures targeting low bit-width fixed-point data have gained significant attention. For instance, SmoothQuant [5] enables int8 quantization for weights and activations in matrix multiplications of LLM, and PD-Quant [6] achieves a classification accuracy of 53.14% (resnet-18 on ImageNet dataset) using uint2 weights and activations without retraining. Additionally, PQDE [7] improves the accuracy of a quantized mobilenetV2 in int4, surpassing its full-precision counterpart on Cifar-10 and Cifar-100 datasets. Without considering the characteristics of computations in DNNs, both weights and activations are usually quantized into either int or unit format, requiring signed×signed and unsigned×unsigned multiplications, respectively. However, commonly used activation functions such as ReLU and Sigmoid produce non-negative activations. In this case, int quantization would waste a sign bit that may lead to low accuracy, whereas uint quantization would introduce bias parameters that complicate the computations. Thus, int quantization for weights and uint quantization for activations perfectly fit this scenario, necessitating signed×unsigned multiplications. However, state-of-the-art (SotA) mixed-precision MAC designs rarely support signed×unsigned operations. Consequently, it is crucial to devise a mixed-precision MAC unit supporting both signed×signed and signed×unsigned multiplications to guarantee the computational efficiency and accuracy of quantized DNNs.

Bit-serial approaches [8], [9] successfully realize the mixed-precision MAC in the temporal domain yet with a significantly high time overhead. In the spatial domain, maximizing the reuse of computational cells by sacrificing a portion of the configuration area is commonly performed to enhance energy efficiency. Existing mixed-precision multiplication architectures can be classified into two typical types, low-precision-combination

This work was supported in part by the National Key Research and Development Program of China under grant number 2022YFB4500200, and in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant numbers RES0048688, RES0051374 and RES0054326.

978-3-9826741-0-0/DATE25/© 2025 EDAA

(LPC) and high-precision-split (HPS), as per their precision configurability in spatial domain [4], [10], [11]. Based on LPC, BitFusion [1] implements mixed-precision multiplication by an array of 2x2 multipliers whose results are fused through configurable shifter and adder networks. BitBlade [12] proposes a bit-wise summation instead of shift-add operations to reduce the area overhead for the bit-width scaling. Although LPC architectures achieve a high hardware utilization rate, the varied input bandwidth for different precision modes complicates the memory access and data flow, reducing compatibility across different system topologies.

In HPS architectures, a lower-precision multiplication is realized by setting some partial products (PPs) of a higher-precision multiplier to zero, without changing the multiplier structure yet compromising the hardware utilization rate. The higher-precision multiplier can be either Booth multiplier [13], [14] or Baugh-Wooley multiplier [15], [16]. While the Booth multiplier reduces the number of PPs directly, it is inefficient for low-precision due to unnecessary control complexity and delays in simple operations. In the Baugh-Wooley multiplier, sum-separate (SS) and sum-together (ST) are two common schemes with similar throughput for implementing mixed-precision functions. Although the traditional SS scheme in [15] demonstrates superior energy and area efficiency compared to ST mode at frequencies above 200 MHz, redundant switching activities inducing extra dynamic power are not considered. Specifically, redundant switching activities will occur when the inputs for a component arrive at different times, i.e., the switching activities that happen before the inputs stabilize are redundant.

This work aims to develop a power-efficient mixed-precision MAC to enable efficient deployment of quantized DNNs. Our new contributions are summarized as follows.

- A novel precision-split 4×4 multiplier (denoted as PS4M) is devised by dividing the PP array into two specific parts that are compressed separately, eliminating redundant switching activities for 2-bit operations. A mixed-precision multiplier is then obtained by combining four PS4Ms, with a thorough tradeoff between circuit reuse and control complexity.
- To further tradeoff between delay, area, and power, an integrated MAC (IMAC) architecture is utilized in the mixed-precision MAC design (PS4MAC), where the multiplication is merged with a 32-bit hybrid accumulator involving a 32-bit sum and a 16-bit carry. By reducing redundant activities in compression, PS4MAC can reduce over 23% power for 8-bit operations, compared with SotA design [15].
- To enable more granularity in DNN accuracy and power efficiency, an accuracy configuration circuit is integrated into PS4MAC, obtaining another MAC supporting approximate 8-bit multiplication (AP), referred to as PS4MAC_AP. Under AP mode, PS4MAC_AP can reduce the power by 29% compared to SotA design [15], achieving similar accuracy in Cifar-10 and Cifar-100 datasets.
- An output-stationary (OS) systolic array (SA) is constructed based on the proposed MAC units. Compared with the traditional systolic array with SS strategy, PS4MAC achieves about 0.55 TOPS/W, 0.40 TOPS/W, and 0.70 TOPS/W improvement for energy efficiency for 8-bit, 4-bit and 2-bit operations, respectively.

II. Mixed-Precision Integrated MAC Design

A. Precision-Split 4×4 Multiplier (PS4M)

The Baugh-Wooley algorithm is widely used to efficiently restructure the partial product generation of signed multiplication, avoiding sign extension operations. Based on the Baugh-Wooley algorithm, we propose the precision-split 4×4 multiplier (PS4M) for the design of a mixed-precision MAC, as shown in Fig. 1. In this design, the 4×4 PP array is split into two distinct regions that are compressed separately, supporting a 4×4 signed×signed multiplication (Fig. 1(a)), a 4×4 signed×unsigned multiplication (Fig. 1(b)), two 2×2 signed×signed multiplications (Fig. 1(c)), or two 2×2 signed×unsigned multiplications (Fig. 1(d)). Different PP accumulation schemes are devised as per their positions in the 8×8 multiplication (Fig. 2); this will be introduced in the next subsection.

The region on the upper right, outlined in red, includes a 2×3 PP array consisting of low-order PPs, while the lower left region in blue contains the remaining PPs. This split strategy is obtained based on a thorough tradeoff between configuration complexity, delay, and power consumption. Specifically, compared with the conventional PP compression scheme in HPS that compresses all PPs as a whole, this split reduces the redundant switching activities for 2×2 multiplications because 0s are concentrated in a PP accumulator (the blue region). When the bit-width (*bw*) of the multiplication is 4 (Figs. 1(a) and (b)) or 8 (Fig. 2), each PP in the red region is simply generated by using an AND gate. When *bw* is 2 (Figs. 1(c) and (d)), the PPs in the red 2×3 region are adjusted for supporting both 2×2 signed×signed and signed×unsigned multiplications. The blue region contains the remaining higher-order partial products (PPs) that are compressed into 8-bit output, $p_{7:0}$. Note that p_1 and p_0 are always zero, and p_3 and p_2 are zero when $bw = 2$.

B. Mixed-Precision Integrated MAC

As shown in Fig. 2(a), our basic mixed-precision multiplier is constructed by using four PS4Ms named HH, HL, LH, and LL. The intermediate accumulation results (IARs) from HL and LH contribute only to the operation mode where $bw = 8$. Otherwise, they are always zero. Therefore, the PPs in the red regions of HL and LH are compressed by using two high-speed parallel prefix adders to shorten the critical path delay. On the contrary, HH and LL are utilized for every operation mode; the PPs in their red regions are compressed by two ripple-carry adders (RCAs) to save area and power. When performing signed×signed and signed×unsigned 8-bit multiplication, the PP generation varies slightly in each PS4M as per the Baugh-Wooley algorithm. However, the format of the compression for each PS4M remains identical, with two outputs $p_{7:0}$ and $c_{4:0}$ that are represented as circles and triangles in Fig. 2(b).

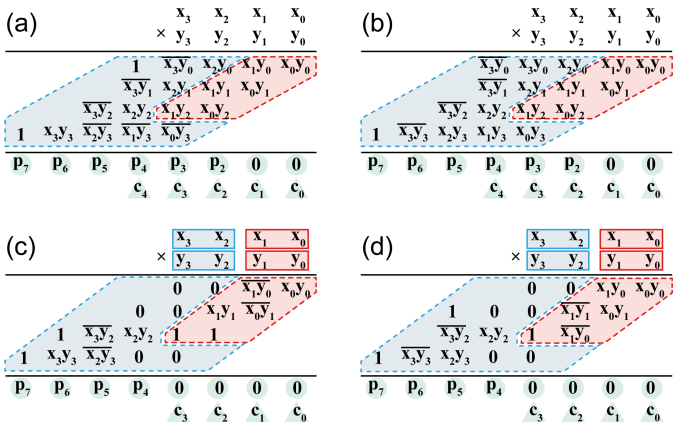

Fig. 1. The partial product arrays of the proposed 4×4 multiplier for (a) a 4×4 signed×signed multiplication, (b) a 4×4 signed×unsigned multiplication, (c) two 2×2 signed×signed multiplications, and (d) two 2×2 signed×unsigned multiplications.

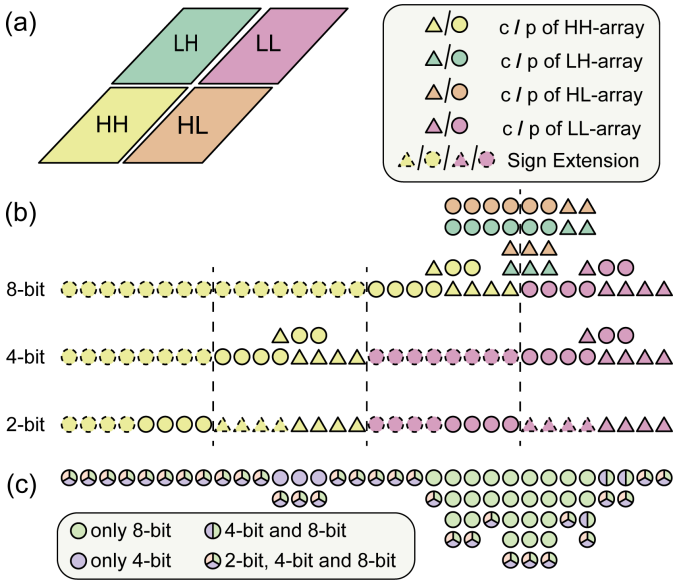

Fig. 2. (a) The proposed mixed-precision multiplication scheme achieved by the combination of four PS4Ms. (b) The array layouts of IARs from PS4Ms with sign extensions for 8-bit, 4-bit and 2-bit multiplications, respectively and (c) the total array of the mixed-precision multiplier.

In general, a MAC unit is constructed by a standalone multiplier and adder, which introduces a large number of redundant switches in the adder caused by two required carry propagation adders (CPAs). Factored systolic array (FSA) [17], [18] has been devised to merge multiplication and accumulation together as an integrated MAC (IMAC) with a fixed precision. In this design, each IMAC involves a hybrid accumulation that uses a CPA on the lower half bits and a carry-save adder (CSA) on the upper half bits. It shows an effective tradeoff between power efficiency and area efficiency.

Traditional HPS-based MAC designs compress the total PP array by a fixed structure, which not only introduces more redundant switches for low-precision operations but also makes it impossible to merge the accumulation into the multiplication. Thus, HPS-based MAC implements the accumulation by adding

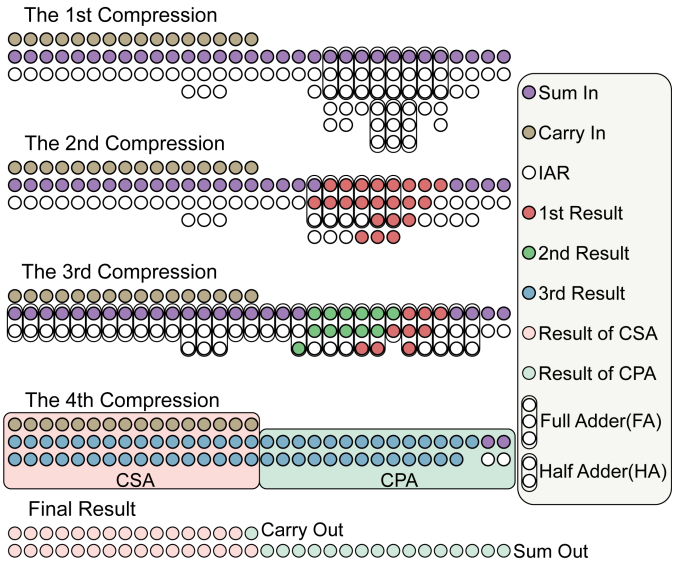

Fig. 3. The compression architecture of PS4MAC.

the final multiplication result with the addend of large bit-width, leading to a long carry propagation chain. The proposed mixed-precision multiplication design releases the strong bound of the PPs and enables the multiplication to merge with the final accumulation. In this design, the accumulation margin is double the *bw* of the multiplication inputs. Thus, the total output bit-width of the mixed-precision MAC is 32-bit. The IARs from four PS4Ms are obtained and then arranged with sign extensions as Fig. 2(b) according to the *bw* of multiplication. Fig. 2(c) merges the three PP array layouts in Fig. 2(b) into the array of the mixed-precision multiplier.

To ensure high speed and low power, the accumulation is merged into PP compression based on the IMAC scheme introduced in [17], obtaining the proposed mixed-precision MAC referred to as PS4MAC. Fig.3 shows the compression architecture for PS4MAC, where the outputs include a 32-bit sum and a 16-bit carry. To diminish redundant switches in the compression, the inputs for FAs and HAs in the Dadda tree are carefully grouped, to balance the delay and power. Note that the IARs in Fig. 2(c) are denoted as white circles in Fig. 3. As the IARs of HL and LH are parallelly generated by two PS4Ms with the same PP compression scheme, the bits at the same position can be obtained simultaneously. Thus, they are grouped and compressed together in the 1st compression. In the 4th compression, a 16-bit CSA and a 16-bit CPA are utilized for upper IARs and lower IARs, respectively. Consequently, the dynamic power consumed by the 32-bit Sum register and 16-bit Carry register is also reduced compared with traditional MAC units. This is because the number of switches within registers in PS4MAC is minimized by the multi-level accumulation.

C. Mixed-Precision MAC with Approximate Mode

Approximate computing is commonly applied in DNNs due to their inherent error resilience [19]. Compared with conventional precision adjustment, approximate computing enhances the granularity of accuracy and hardware efficiency. Building

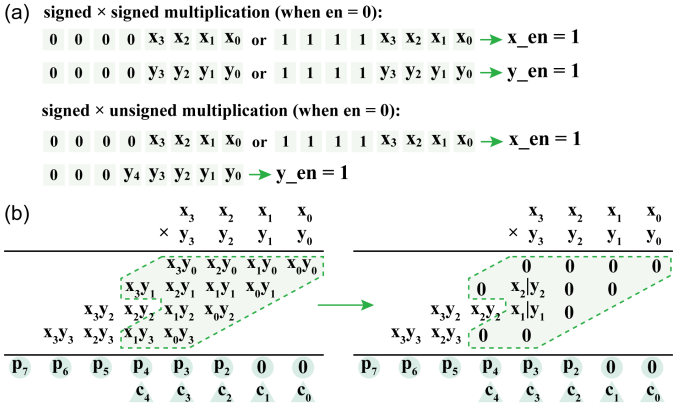

Fig. 4. (a) The detection scheme of signals x_en and y_en, and (b) the partial product array of the LL multiplier in 8-bit approximate mode.

upon PS4MAC, an 8-bit approximate multiplication mode is developed, producing a MAC unit with eight configurable operation modes, denoted as PS4MAC_AP.

To guarantee a high accuracy for DNN applications, we propose to approximate the compression of less significant PPs when the absolute values of the multiplication inputs are sufficiently large. The magnitudes of the input data x and y are identified via leading-one/zero detection. Fig. 4(a) shows the generation scheme for the flag signals x_en and y_en, where $en = 0$ denotes that the approximate mode is enabled. In the AP mode, when the leading-one (for positive inputs) or leading-zero (for negative inputs) is located at the lower 4 (or 5 for unsigned numbers) bits, x_en (or y_en) is set to 1, indicating that the magnitude of the input x (or y) is small; otherwise, x_en (or y_en) is set to 0. When both x_en and y_en are 1, the multiplication is performed without any approximation. The final configuration signal ll_en is given by

$$ll\_en = x\_en \ \& \ y\_en. \tag{1}$$

When the MAC executes 8-bit operations, $ll\_en = 0$ enables the approximation.

Figure 4(b) illustrates the proposed approximation scheme for the LL PS4M shown in Fig. 2. In the right green region, the PPs are approximately compressed into two values, $x_1|y_1$ and $x_2|y_2$, located at the third column. This approximation strategy is derived from exhaustive simulations, balancing the hardware while maintaining high accuracy in DNN applications. The HH, HL, and LH remain unchanged.

III. HARDWARE EVALUATION

This section evaluates the circuit characteristics of both MAC units and systolic array architectures with PS4MAC, PS4MAC_AP, and other state-of-the-art designs employing different multi-precision techniques. This section evaluates the circuit characteristics of the proposed PS4MAC, PS4MAC_AP, and other state-of-the-art mixed-precision MAC designs. Furthermore, the systolic array designs based on the considered mixed-precision MAC architectures are evaluated in terms of power and area efficiency.

A. Mixed-Precision MAC Designs

Except for the mixed-precision MAC design based on DW multipliers [20], we compare our designs with two MAC units based on the Booth-Wallace (BW) multiplier [13], [14] and the SS multiplier [15], respectively. To ensure a fair comparison, each MAC unit receives two 8-bit multiplication inputs, along with a 32-bit sum and a 16-bit carry, producing an updated 32-bit sum and 16-bit carry. In addition, we modified all MAC units to make them support signed×signed and signed×unsigned computations based on their own strategy. As the Booth algorithm offers minimal power efficiency for 2-bit computations, BW is scaled to support only 4-bit and 8-bit operations.

All MAC designs are synthesized by Synopsys Design Compiler (DC) using 28nm technology in the typical condition with high-density cells under the condition of 25°C and 0.9V supply voltage. In addition, Verilog Compile Simulator (VCS) tools are applied to verify operating functions and create waveform files used as the stimuli of MAC units to estimate the power dissipation. Each waveform file involves a million Gaussian-distributed random input combinations with 1GHz frequency in different bit-width operation modes, simulating the data statistics of quantized DNNs. The unsigned inputs follow a truncated Gaussian distribution, constrained to positive values.

TABLE I
HARDWARE COMPARISON OF MERGED MULTI-PRECISION MAC DESIGNS

MAC Designs		DW [20]	SS [15]	BW [13], [14]	PS4MAC	PS4MAC_AP AC	PS4MAC_AP AP
Delay (ns)		0.8784	0.9975	0.9997	0.9974	0.9986	
Area (μm^2)		275.18	213.05	233.34	230.59	248.04	
Power (μW)	8-bit ss*	325.1	269.1	277.9	**206.7**	206.8	**188.9**
	8-bit su#	319.8	253.1	278.0	**194.8**	196.6	**179.8**
	4-bit ss	308.4	178.3	246.6	**176.4**	180.8	–
	4-bit su	309.5	174.1	242.3	**170.7**	175.2	–
	2-bit ss	249.5	98.6	–	**87.1**	91.3	–
	2-bit su	280.0	130.6	–	**121.8**	125.0	–

* ss represents signed×signed multiplication.
# su represents signed×unsigned multiplication.

Table I shows the results for critical path delay, area, and power consumption of each operation mode for the considered mixed-precision MAC designs. DW performs mixed-precision multiplications using several standalone multipliers with different bit-widths. This approach minimizes the critical path delay but results in increased area and power dissipation due to the lack of circuit reuse. From an overall perspective, given an operation bit-width, the power consumptions of the considered MAC designs for signed×signed and signed×unsigned operations are nearly identical except for 2-bit operations. This is because the sign adjustment takes an increasingly high portion of power consumption in the 2-bit operating circuits. SS attains minimal area, while its PP compression produces many redundant switching activities, leading to high power consumption for 8-bit operations. The proposed PS4MAC outperforms DW and BW in terms of area, and power in every operating mode. Compared with SS, PS4MAC exhibits a similar delay

and slightly increased area. However, it can reduce power consumption by 23.2%, 1.1%, and 11.7% for 8-bit, 4-bit, and 2-bit operations with signed×signed multiplication, and 23.0%, 2.0%, and 6.7% for those with signed×unsigned multiplication. Note that the power reductions for 4-bit and 2-bit operations are much lower than those for 8-bit operations. This is because the PP compression is optimized to eliminate the redundant switches of 8-bit operations, which would introduce additional switches for 4-bit and 2-bit operations, counteracting the power gains due to PS4M. Although with slightly more delay and area costs than SS due to additional control circuits, PS4MAC_AP achieves 29.8% and 29.0% reductions in power consumption for 8-bit signed×signed and signed×unsigned approximate MAC operations, respectively, with comparable results in 4-bit and 2-bit scenarios.

B. Systolic Array

In general, an output-stationary SA consists of a grid of processing elements (PEs), as shown in Fig. 5(a). Each PE implements a two-stage pipeline, separating multiplication and accumulation across two clock periods, as shown in Fig. 5(b). The propagate signal *prop* dictates whether PEs perform MAC operations locally or pass the accumulated results to neighboring PEs for further processing, enabling efficient data flow.

In contrast, a different architecture referred to as CPA-factored SA (FSA) [17] shows higher power and area efficiency, as shown in Fig. 5(c). Fig. 5(d) shows the corresponding PE that is implemented by the considered MAC units in this work. The approximation-enable signal *en*, represented by the dashed line, is only utilized for PS4MAC_AP. Once propagated to the final row of the SA, the accumulated sum and carry are summed by using a 16-bit RCA, yielding the final result for general matrix multiplication (GEMM). While the RCA increases the computational area, it significantly reduces power consumption and shortens the critical path of the integrated MAC units. Furthermore, as the size of SA increases, the number of PEs grows quadratically, while the number of extra RCAs grows linearly, indicating that extra RCAs occupy only a small fraction of the total area in large-scale CPA-factored SAs.

Figure 6 presents the energy and area efficiency of SAs operating at 1GHz clock frequency, where N denotes the size of the SA. Both traditional SA and FSA contain N×N PEs, while FSA incorporates additional N CPAs. SAs with $N = 16$ and $N = 32$ are evaluated to assess the impact of size. As the SS strategy results in the minimal area for the MAC unit, the traditional SS-based SA (SS_1O), outputting a 32-bit accumulated sum, is synthesized for comparison. The other SAs are constructed by using the previously discussed mixed-precision MAC designs, with the SS-based design referred to as SS_2O. Figure 6 shows that our designs achieve superior energy and area efficiency across 8-bit, 4-bit, and 2-bit operations. While SS_1O exhibits higher energy efficiency than SS_2O, indicating the compatibility of the SS strategy with traditional SA architectures. The proposed PS4MAC-based SA outperforms SS_1O by approximately 0.55 TOPS/W, 0.40 TOPS/W, and

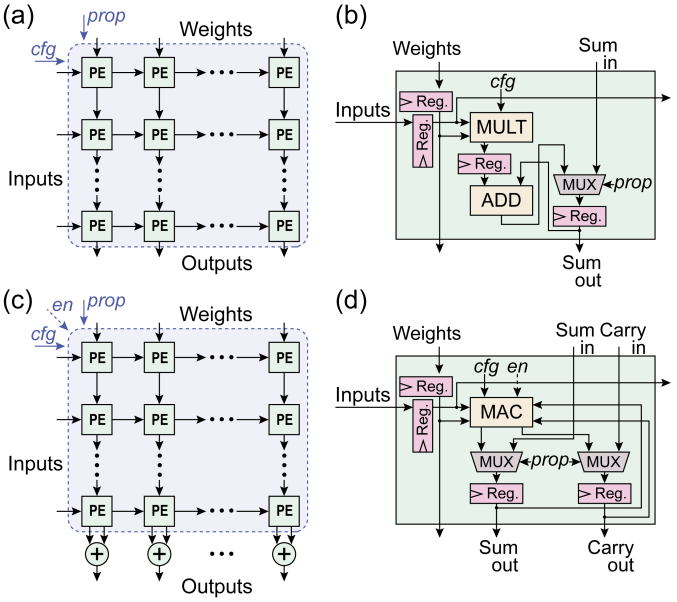

Fig. 5. (a) The traditional output-stationary SA, and (b) the internal architecture of its PE with a two-stage pipeline. (c) The CPA-factored output-stationary SA [17], and (d) the internal architecture of its PE.

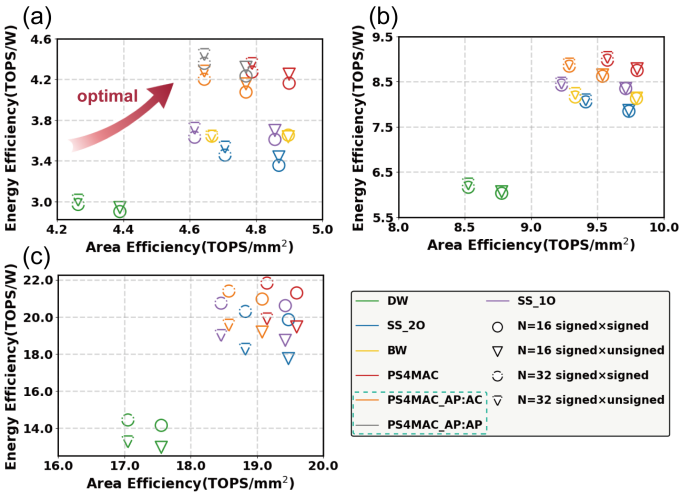

Fig. 6. Energy efficiency and area efficiency of systolic arrays for (a) 8-bit, (b) 4-bit, and (c) 2-bit operations.

0.70 TOPS/W in 8-bit, 4-bit, and 2-bit operations, respectively. With a lower area efficiency, PS4MAC_AP demonstrates above 0.60 TOPS/W improvements in energy efficiency for 8-bit approximate operations at $N = 16$ and $N = 32$, compared to SS_1O. Although the BW-based SA offers comparable area efficiency to PS4MAC, it does not support 2-bit operation.

IV. QUANTIZED DNN APPLICATIONS

To show the efficiency of the proposed mixed-precision MAC designs, the accuracy of several quantized DNNs is accessed, utilizing models of resnet, densenet, and ror with different depths and Min-Max quantization on three typical image classification datasets, Cifar-10, Cifar-100, and ImageNet. The

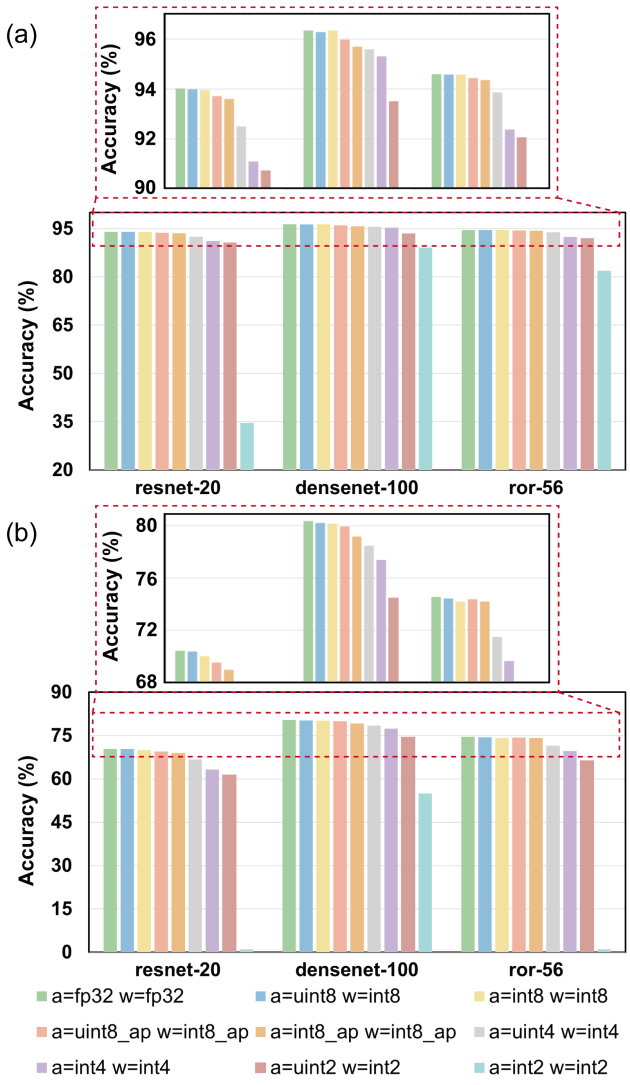

Fig. 7. The classification accuracy of various exact and quantized DNNs on the (a) Cifar-10 and (b) Cifar-100 datasets.

proposed approximate multiplication is tested by replacing the accurate multiplication with it in GEMM.

For Cifar-10 and Cifar-100, 32-bit floating-point (fp32) and 8-bit fixed-point DNNs are obtained through post-training quantization (PTQ), while 2-bit and 4-bit fixed-point DNNs are achieved using quantization-aware training (QAT) over 20 and 200 epochs, respectively. Additionally, 2-bit quantization employs per-channel and LSQ techniques [21] to improve the accuracy. As shown in Fig. 7, for a same bit-width, signed×unsigned multiplication outperforms signed×signed multiplication, particularly in 2bit computations. Thus, a low-bit-width signed×unsigned multiplication can replace higher-bit-width multiplication to achieve greater energy efficiency while maintaining high accuracy. The accuracy of PS4MAC_AP in AP mode closely matches the exact computation without the need for retraining. Although 4-bit computations achieve enhanced accuracy due to QAT, their accuracy cannot beat that in AP mode. In these cases, PS4MAC_AP offers a more energy-efficient alternative while keeping high accuracy.

TABLE II
THE ACCURACY (%) OF EXACT AND QUANTIZED RESNET-34 MODELS ON THE IMAGENET DATASET

act.	fp32	uint8	int8	uint8_ap	int8_ap	uint4	int4	uint2	int2
wei.	fp32	int8	int8	int8_ap	int8_ap	int4	int4	int2	int2
acc.	75.16	75.02	75.07	1.06	0.10	73.63	72.21	25.91	2.31

For ImageNet dataset, the quantization techniques remain unchanged, but the number of training epochs for 2-bit computation is reduced to 20 due to the complexity and time demands for retraining on large-scale datasets. Table II indicates the limitations of 2-bit computations on the accuracy, necessitating the use of higher-bit-width models. When using PS4MAC_AP for GEMM, the AP mode leads to very low accuracy in the classification of ImageNet. In this case, the AC mode should be enabled to improve the accuracy.

V. CONCLUSION

Aiming at quantized DNNs, this paper proposes an energy-efficient mixed-precision integrated MAC unit (denoted as PS4MAC) that supports six operation modes, i.e., an int8×int8, an int8×uint8, two int4×int4, two int4×uint4, four int2×int2, and four int2×uint2 with corresponding accumulations. Based on PS4MAC, PS4MAC_AP is devised by integrating an approximate mode for 8-bit multiplications. The approximation mode in PS4MAC_AP achieves over 42% and 29% reductions in power dissipation compared with DW and SS, respectively. Compared with the traditional SA using SS under a 1 GHz clock, the FSA constructed by using PS4MAC_AP improves the energy efficiency for 8-bit operations by 0.6 TOPS/W and achieves a higher area efficiency when the size of SA is 32×32. PS4MAC achieves 0.4 TOPS/W - 0.7 TOPS/W improvement for all modes with the highest area efficiency. Finally, the accuracy of various quantized low-bit DNNs is evaluated, showing the necessity of supporting mixed-precision and signed×unsigned multiplications.

REFERENCES

[1] H. Sharma, J. Park, N. Suda, L. Lai, B. Chau, J. K. Kim, V. Chandra, and H. Esmaeilzadeh, "Bit fusion: Bit-level dynamically composable architecture for accelerating deep neural network," in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pp. 764–775, 2018.

[2] W. Mao, K. Li, Q. Cheng, L. Dai, B. Li, X. Xie, H. Li, L. Lin, and H. Yu, "A configurable floating-point multiple-precision processing element for hpc and ai converged computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 2, pp. 213–226, 2022.

[3] S. B. Ali, S.-I. Filip, and O. Sentieys, "A stochastic rounding-enabled low-precision floating-point mac for dnn training," in *2024 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1–6, 2024.

[4] M. M. H. Shuvo, S. K. Islam, J. Cheng, and B. I. Morshed, "Efficient acceleration of deep learning inference on resource-constrained edge devices: A review," *Proceedings of the IEEE*, vol. 111, no. 1, pp. 42–91, 2023.

[5] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, "SmoothQuant: Accurate and efficient post-training quantization for large language models," in *Proceedings of the 40th International Conference on Machine Learning*, vol. 202 of *Proceedings of Machine Learning Research*, pp. 38087–38099, PMLR, 23–29 Jul 2023.

[6] J. Liu, L. Niu, Z. Yuan, D. Yang, X. Wang, and W. Liu, "Pd-quant: Post-training quantization based on prediction difference metric," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24427–24437, June 2023.

[7] Z. Yue, R. Wu, L. Ma, C. Fu, and C.-W. Sham, "Pqde: Comprehensive progressive quantization with discretization error for ultra-low bitrate mobilenet towards low-resolution imagery," in *2024 IEEE 6th International Conference on AI Circuits and Systems (AICAS)*, pp. 452–456, 2024.

[8] J. Lee, C. Kim, S. Kang, D. Shin, S. Kim, and H.-J. Yoo, "Unpu: An energy-efficient deep neural network accelerator with fully variable weight bit precision," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 173–185, 2018.

[9] S. Sharify, A. D. Lascorz, K. Siu, P. Judd, and A. Moshovos, "Loom: Exploiting weight and activation precisions to accelerate convolutional neural networks," in *Proceedings of the 55th Annual Design Automation Conference*, pp. 1–6, 2018.

[10] V. Camus, C. Enz, and M. Verhelst, "Survey of precision-scalable multiply-accumulate units for neural-network processing," in *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pp. 57–61, IEEE, 2019.

[11] V. Camus, L. Mei, C. Enz, and M. Verhelst, "Review and benchmarking of precision-scalable multiply-accumulate unit architectures for embedded neural-network processing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 4, pp. 697–711, 2019.

[12] S. Ryu, H. Kim, W. Yi, E. Kim, Y. Kim, T. Kim, and J.-J. Kim, "Bitblade: Energy-efficient variable bit-precision hardware accelerator for quantized neural networks," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 6, pp. 1924–1935, 2022.

[13] B. Moons, R. Uytterhoeven, W. Dehaene, and M. Verhelst, "14.5 envision: A 0.26-to-10tops/w subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28nm fdsoi," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 246–247, IEEE, 2017.

[14] S. Zhang, J. Gu, S. Yin, L. Liu, and S. Wei, "A multiple-precision multiply and accumulation design with multiply-add merged strategy for ai accelerating," in *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, pp. 229–234, 2021.

[15] L. Mei, M. Dandekar, D. Rodopoulos, J. Constantin, P. Debacker, R. Lauwereins, and M. Verhelst, "Sub-word parallel precision-scalable mac engines for efficient embedded dnn inference," in *2019 IEEE international conference on artificial intelligence circuits and systems (AICAS)*, pp. 6–10, IEEE, 2019.

[16] E. Manca, L. Urbinati, and M. R. Casu, "Star: Sum-together/apart reconfigurable multipliers for precision-scalable ml workloads," in *2024 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1–6, 2024.

[17] K. Inayat and J. Chung, "Hybrid accumulator factored systolic array for machine learning acceleration," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 7, pp. 881–892, 2022.

[18] D. N. Devi, G. Ajay Kumar, B. G Gowda, and M. Rao, "Integrated mac-based systolic arrays: Design and performance evaluation," in *Proceedings of the Great Lakes Symposium on VLSI 2024*, GLSVLSI '24, (New York, NY, USA), p. 292–295, Association for Computing Machinery, 2024.

[19] X. Hu, A. Liu, X. Geng, Z. Wei, K. Jiang, and H. Jiang, "A configurable approximate multiplier for cnns using partial product speculation," in *2024 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1–6, 2024.

[20] Synopsys, "Designware DW02_mult." https://www.synopsys.com/dw/ipdir.php?c=DW02_mult, 2024.

[21] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," *CoRR*, vol. abs/1902.08153, 2019.