A 13-34 TOPS/W Edge-AI Processor Featuring Booth-Value-Confined Accelerator, Near-Memory Computing, and Contiguity-Aware Mapping

Quan Cheng[1*], Longyang Lin[2*], Mingqiang Huang[2*], Qiufeng Li[2], Zhengke Yang[2], Liuyao Dai[2], Hao Yu[2], Yu-Jen Chen[3], Yiyu Shi[3], Masanori Hashimoto[1]

[1]Kyoto University, Japan

[2]Southern University of Science and Technology, China

[3]University of Notre Dame, USA

Email: hashimoto@i.kyoto-u.ac.jp, [*]Equally contributed authors

2024 IEEE Asian Solid-State Circuits Conference (A-SSCC) | 979-8-3503-7632-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/A-SSCC60305.2024.10849341

Recently, edge artificial intelligence (AI) devices, particularly those with in-memory-computing (IMC)/near-memory-computing (NMC) architecture, have demonstrated superior power efficiency compared with conventional von-Neumann architecture [1]-[7]. To further improve the power efficiency, approximate computing techniques have been investigated [8], which inevitably incur accuracy loss as such approximations cannot be evaluated during the NN training stage on GPU (Fig. 1). In addition, despite the exceptionally high power efficiency at the macro-level, the often-overlooked resource consumption for inter-/intra-layer data alignment and reshape can actually lead to substantial overhead of power consumption and latency. This is caused by 1) inefficient memory bandwidth utilization due to irregular or unbalanced read/write patterns (e.g., [3][4]), and 2) severe memory space discontinuity due to inefficient neural network (NN) mapping strategy (e.g., [6][7]), as shown in Fig. 1. Hence, when considering the end-to-end NN execution, aspects of off-chip memory traffic and network mapping strategy require further investigations at the architectural level.

To address the above-mentioned issues, an edge-AI processor is proposed in this work, featuring 1) power-/area-efficient processing element (PE) with Booth-value-confined (BVC) approximate multipliers and a shared adder tree, supporting BVC-based NN training with no accuracy loss, 2) NMC-friendly data flow supporting regular and balanced read/write operation and optimized data-reuse, and 3) memory space contiguity-aware (MSCA) NN mapping strategy with hardware-software co-design for minimum memory access latency, as shown in Fig. 1.

Fig. 2 presents the architecture of the proposed edge-AI processor, consisting of a full RISC-V sub-system with rich peripherals as the main controller, a digital PLL, a programmable pooling unit supporting both max and average pooling, and a programmable MatrixConv unit with 16 near-memory-engines (NMEs) for accelerating the convolutional and matrix computations. Each NME contains a BVC PE with a 16KB weight memory directly connected for NMC. The BVC PE supports 3 customized precision, namely BVC8/6/3, detailed in Fig. 3. The NMC-friendly data flow accommodates two main data streams: 1) inter-NME data flow for activation reuse, and 2) intra-NME data flow for weight reuse, detailed in Fig. 4.

To further improve power efficiency, multi-precision (MP) capability is integrated into the proposed radix-8 Booth-based BVC approximate PE, supporting three different custom precisions (BVC3/6/8) with offline Booth encoding, as shown in Fig. 3. The BVC weights inherently exclude "±3X" cases after BVC-based NN training on GPU with minimal accuracy loss. MP parallel computation is achieved by the configurable shifters and multiplexer for different combinations of partial product accumulations, with the adder trees shared in all precisions. The overall reductions in power and area are 82% and 70% respectively, compared with conventional MP Booth-based PE [2]. Moreover, BVC3 case only generates one partial product per weight, resulting in a 2X throughput compared to other BVC cases. Besides, the quantity of unique value of BVC3/6/8 is 7, 36, 144, respectively.

To fully exploit the benefits of NMC, both intra- and inter- NME data paths with NMC-friendly data flow are incorporated, allowing data reuse for both activations and weights, as shown in Fig. 4(a). A Partial Sum Scheduler is utilized to schedule the accumulation process of temporal NME outputs. 16 data in each buffer group represent the output of 16 different output channels, while data under the same demultiplexer represents the output of the same output channel. As shown in Fig. 4(b), each weight memory updates data every 1-to-16 cycles, which means that each weight can be reused 1-to-16 times in each PE. In addition, activations can also be reused 1-to-16 times as the number of NME is 16. Moreover, at the system-level, three data-loading options are incorporated: 1) updating weights, 2) updating activations, and 3) updating both, as the weights or activations used in the current computation round might be called upon again in the next round depending on the mapped NN structure. Fig. 4(c) shows the proposed MSCA NN mapping strategy. The input shape of each layer is determined by *Nm* (i.e., the number of multipliers in each PE), while the output shape of each layer is determined by *Ne* (i.e., the number of PEs). Hence, the primary idea is to ensure that the input shape matches the output shape so that no data reshaping is required, specifically *Nm* = *Ne*. However, fulfilling the above condition might not always be feasible due to limited chip area and power budgets. In such cases, it is preferable to ensure that *Nm* is divisible by *Ne*, thereby eliminating the required buffer and logic for weight reshaping. In addition, the memory space contiguity is strictly maintained when the off-chip data storage is organized in line with *Nm*-based data shape, meaning that the high-volume burst access is always feasible, resulting in minimum memory access latency.

Fig. 5 illustrates the BVC-based NN training flow. Original weights in floating-point (FP) format are first quantized to INT8 and then extended to MP INT weights via Neural Architecture Search (NAS) [9]. Then, the exact MP INT weights are converted to MP BVC weights through a lookup table during each re-training iteration until a minimal accuracy loss is achieved. Note that the NN computation on our architecture is identical to that in re-training on GPU. Also, weight encoding is adopted to reduce the computation complexity of the BVC PE on chip. Finally, data reshaping is applied to satisfy the proposed NMC-friendly data flow. Besides, the performance of each layer of the VGG16 and ViT-Tiny models trained based on the above strategy is shown below. Due to the low weight reuse rate in transformer-based NNs, the power efficiency of ViT-Tiny is relatively lower compared to convolutional NNs like VGG16.

Fig. 6 illustrates the end-to-end NN executions of VGG16 and ViT-Tiny, performed with off-chip DRAM (I/O power consumption is excluded for a fair comparison), achieving an average power efficiency of 10.14-15.11 TOPS/W, peak power efficiency of 12.92-29.11 TOPS/W, and 23.3%-31.4% memory latency reduction. Moreover, Fig. 6 compares the proposed design with prior arts [3]-[7]. Compared with [3]-[5] where only an accelerator is implemented, the proposed design is capable of end-to-end NN executions with competitive peak performance and power efficiency. Compared with [6]-[7] in real NN evaluations, the proposed design improves the power efficiency by 1X-5X and area efficiency by 4.6X-35.7X at a similar process node. The proposed edge-AI processor is fabricated on 22nm CMOS process (Fig. 7). The measured power is 13.61-210.67mW at 230-1020MHz frequency under a voltage of 0.6-1.1V. The peak power efficiency is 33.98TOPS/W with Activation(INT4) x Weight(BVC3) case at 0.6V and 230MHz.

Acknowledgement

This work was supported by JST CREST Grant Number JPMJCR18K2, Japan.

References:

[1] J. Heo et al., "PRIMO: A Full-Stack Processing-in-DRAM Emulation Framework for Machine Learning Workloads," *2023* IEEE/ACM ICCAD, San Francisco, CA, USA, 2023, pp. 1-9.

[2] M. Huang et al., "A High Performance Multi-Bit-Width Booth Vector Systolic Accelerator for NAS Optimized Deep Learning Neural Networks," IEEE TCAS-I, vol. 69, no. 9, pp. 3619-3631, 2022.

[3] S. Ryu et al., "BitBlade: Energy-Efficient Variable Bit-Precision Hardware Accelerator for Quantized Neural Networks," in IEEE JSSC, vol. 57, no. 6, pp. 1924-1935, 2022.

[4] F. Tu et al., "A 28nm 15.59μJ/Token Full-Digital Bitline-Transpose CIM-Based Sparse Transformer Accelerator with Pipeline/Parallel Reconfigurable Modes," ISSCC, pp. 466-468, 2022.

[5] Y. Wang et al., "An Energy-Efficient Transformer Processor Exploiting Dynamic Weak Relevances in Global Attention," IEEE JSSC, vol. 58, no. 1, pp. 227-242, 2023.

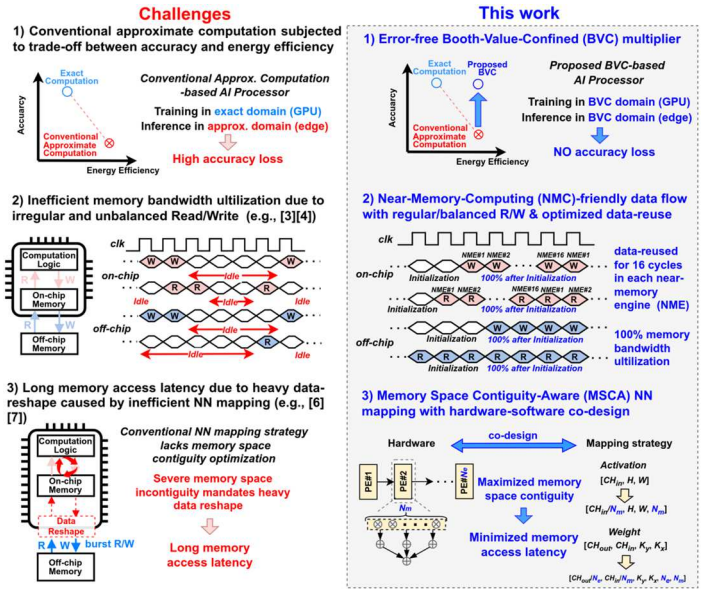


Fig. 1. Challenges in edge-AI processor and proposed solutions.

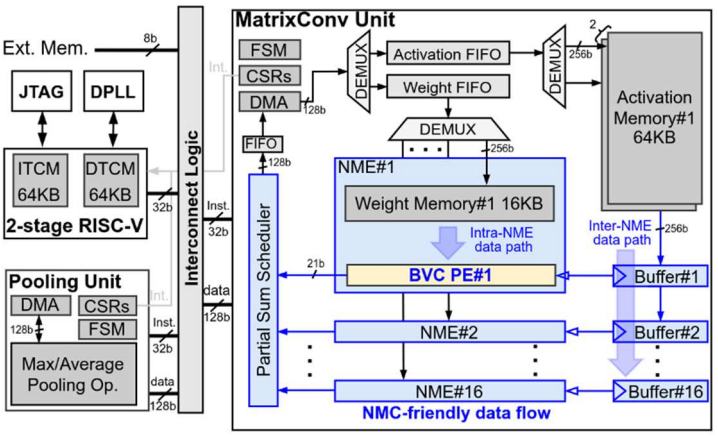


Fig. 2. The overall edge-AI processor architecture with near-memory-computing-friendly data flow and Booth-value-confined multi-precision processing elements.

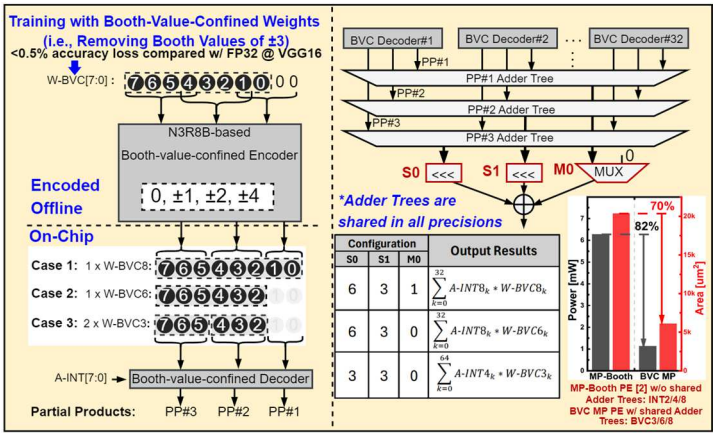


Fig. 3. Radix-8 Booth-based Booth-value-confined multi-precision processing element.

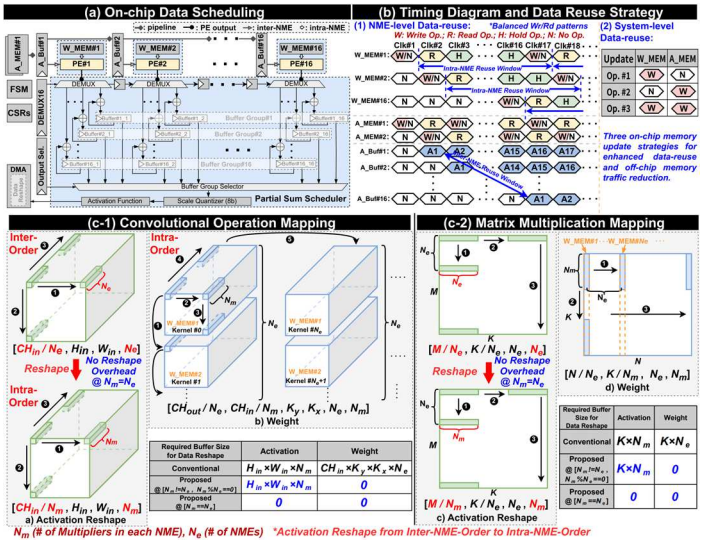


Fig. 4. (a) On-chip data scheduling within intra-/inter-NME data paths; (b) Timing diagram and data-reuse strategy; (c) Memory space contiguity-aware NN mapping strategy.

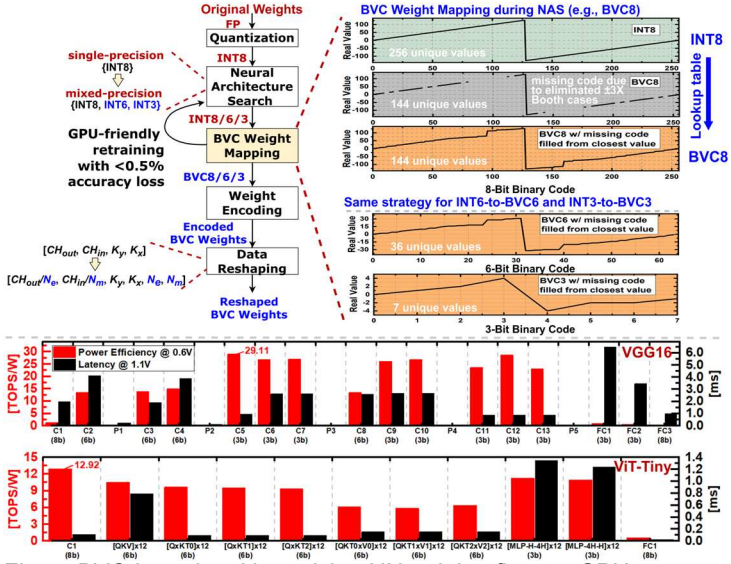


Fig. 5. BVC-based multi-precision NN training flow on GPU; Measurement results on different NN models.

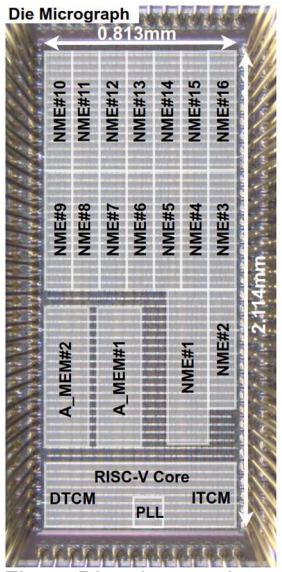
Model	VGG16	ViT-Tiny
Dataset	ImageNet	ImageNet
Accuracy [%]	71.17	69.77
Weight / Activation Sparsity [%]	16.11 / 69.31	10.88 / 52.89
Latency @ 1.1V [ms]	40.19	4.25
Average Power Effciency @ 0.6V [TOPS/W]	15.11	10.14
Peak Power Effciency @ 0.6V [TOPS/W]	29.11	12.92
Latency Reduction[1]	31.4%	23.3%

Assuming external bandwidth of N_e bytes/clock
[1]Compared to an identical design w/o MSCA NN mapping strategy (i.e., inefficient burst access)

	JSSC22 [3]	ISSCC22 [4]	JSSC22 [5]	ISSCC23 [6]	ISSCC23 [7]	This Work
Process	28nm	28nm	28nm	28nm	22nm FDX	22nm
Architecture	Digital Accel.	Digital CIM Accel.	Digital Accel.	Digital Accel.	1x RV32IMCFXpulp+ 16x RV32IMCFXpupnn+ RBE Accel.	1x RV32IMAC+ Digital Accel.
NN Mapping	NA	NA	NA	YES[1]	YES[2]	YES[3]
Supply [V]	0.6-1.0	0.6-1.0	0.56-1.1	0.66-1.3	0.5-0.8	0.6-1.1
Frequency [MHz]	44-195	80-240	50-510	100-500	420	230-1020 (Accel.) 10-200 (RISC-V)
Area [mm^2]	0.71	6.83	6.82	7.81	18.7	Core Area: 1.719 Die Area: 2.750
Power [mW]	7.8-74	27.04-118.21	12.06-272.8	17-174	12.8 - 123	13.61-210.67
Memory [KB]	144	192	336	1120	1152	512
Precision	INT2/4/8	INT8/16	Approx. INT12	INT8	INT2/4/8/16/32(RISC-V) INT2-8(RBE)	INT8/16/32(RISC-V) BVC3/6/8(Accel.)
Peak Performance [TOPS]	1.42@INT2	0.37@INT16 1.48@INT8	0.52-4.07	1.024	0.637@RBE2x2	2.09@BVC3 1.04@BVC6 1.04@BVC8
Area Efficiency [TOPS/mm^2]	3.30@INT2	0.054@INT16 0.217@INT8	0.597	0.131	0.034@RBE2x2	1.215@BVC3 0.608@BVC6 0.608@BVC8
Peak Power Efficiency [TOPS/W]	44.1@INT2	20.5@INT8 5.1@INT16	27.56 3.98@GPT-2	11.2@InceptionV3 10.7@ResNet50 10.5@MobieNetV1	12.4@RBE2x2 6.38@RBE Mixed ResNet20 5.83@RBE4x4 ResNet18	33.98@BVC3 18.51@BVC6 15.91@BVC8 29.11@VGG16 12.92@ViT-Tiny

[1]Mainly focus on Multiply-and-Accumulate utilization; [2]Supports only 3x3 and 1x1 convolutions; [3]Fully programmable

Fig. 6. Measurement and comparison with state-of-the-art designs.



	Specification
Process	22nm CMOS
Supply [V]	0.6-1.1
Frequency [MHz]	230-1020 (Accelerator) 10-200 (RISC-V)
Chip Area [mm^2]	2.75
Core Area [mm^2]	1.719
Power [mW]	13.61-210.67
On-chip Memory [KB]	128 (ITCM & DTCM) 128 (AMEM) 256 (WMEM)
Multiply-and-Accumulate	1024 @ A-INT4 × W-BVC3 512 @ A-INT8 × W-BVC6 512 @ A-INT8 × W-BVC8
Precision	INT8/16/32 (RISC-V) Weight: BVC3/6/8 (Accel.) Activation: INT4/8 (Accel.)
Peak Performance @ 1.1V [TOPS]	2.09 @ A-INT4 × W-BVC3 1.04 @ A-INT8 × W-BVC6 1.04 @ A-INT8 × W-BVC8
Area Efficiency @ 1.1V [TOPS/mm^2]	1.215 @ A-INT4 × W-BVC3 0.608 @ A-INT8 × W-BVC6 0.608 @ A-INT8 × W-BVC8
Peak Power Efficiency @ 1.1V [TOPS/W]	10.74 @ A-INT4 × W-BVC3* 5.48 @ A-INT8 × W-BVC6* 4.96 @ A-INT8 × W-BVC8*
Peak Power Efficiency @ 0.6V [TOPS/W]	33.98 @ A-INT4 × W-BVC3* 18.51 @ A-INT8 × W-BVC6* 15.91 @ A-INT8 × W-BVC8*
	29.11 @ VGG16 12.92 @ ViT-Tiny

*25% / 75% sparsity for weight / activation, 100% PE utilization

Fig. 7. Die micrograph and summary table.

Additional References:

[6] C. -Y. Du et al., “A 28nm 11.2TOPS/W Hardware-Utilization-Aware Neural-Network Accelerator with Dynamic Dataflow,” ISSCC, pp. 1-3, 2023.

[7] F. Conti et al., “22.1 A 12.4TOPS/W @ 136GOPS AI-IoT System-on-Chip with 16 RISC-V, 2-to-8b Precision-Scalable DNN Acceleration and 30%-Boost Adaptive Body Biasing,” ISSCC, pp. 21-23, 2023.

[8] A. G. M. Strollo et al., “Approximate Multipliers Using Static Segmentation: Error Analysis and Improvements,” IEEE TCAS-I, vol. 69, no. 6, pp. 2449-2462, June 2022.

[9] B. Wu et al., “FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search," 2019 IEEE/CVF CVPR, Long Beach, CA, USA, 2019, pp. 10726-10734.