15.2 A 2.75-to-75.9TOPS/W Computing-in-Memory NN Processor Supporting Set-Associate Block-Wise Zero Skipping and Ping-Pong CIM with Simultaneous Computation and Weight Updating

Jinshan Yue[1,2], Xiaoyu Feng[1], Yifan He[1], Yuxuan Huang[1], Yipeng Wang[2], Zhe Yuan[1], Mingtao Zhan[1], Jiaxin Liu[1], Jian-Wei Su[3], Yen-Lin Chung[3], Ping-Chun Wu[3], Li-Yang Hung[3], Meng-Fan Chang[3], Nan Sun[1], Xueqing Li[1], Huazhong Yang[1], Yongpan Liu[1]

[1]Tsinghua University, Beijing, China
[2]Pi2star Technology, Beijing, China
[3]National Tsing Hua University, Hsinchu, Taiwan

2021 IEEE International Solid- State Circuits Conference (ISSCC) | 978-1-7281-9549-0/20/$31.00 ©2021 IEEE | DOI: 10.1109/ISSCC42613.2021.9365958

Computing-in-memory (CIM) is an attractive approach for energy-efficient neural network (NN) processors, especially for low-power edge devices. Previous CIM chips [1-5] have demonstrated macro and system-level design enabling multi-bit operations and sparsity support. However, several challenges exist, as shown in Fig. 15.2.1. First, though a previously proposed block-wise sparsity strategy [5] can power off ADCs, zeros still contributed to storage requirements, and power gating was not applied to computing resources. Second, on-chip SRAM CIM macros are not large enough to hold all weights. Updating weights between computing operations leads to significant performance loss. Finally, the limited sensing margin incurs poor accuracy for large NN models on practical datasets, such as ImageNet. The precision and power of the ADCs should be optimized and adjusted.

To overcome these challenges, this work proposes a CIM NN processor with 2.75-to-75.9TOPS/W system energy efficiency. The main contributions include: 1) A set-associate block-wise zero-skipping architecture (SABZA) that skips zero-valued activations and weights to reduce power, storage and execution time; 2) A ping-pong CIM (PP-CIM) architecture enabling simultaneous computing and weight update operations; 3) An energy-efficient PP-CIM circuit design and a digital-predictor-assisted adaptive 0/2/4b ADC to support accurate inference on practical datasets.

Figure 15.2.2 shows the overall CIM processor, consisting of two global activation/weight SRAMs, four homogeneous CIM cores, an instruction SRAM with top controller, a set-associate adder array and a result SRAM. Each CIM core contains an activation buffer, a sparsity scheduling module (SSM), a weight index SRAM, a PP-CIM macro and a local adder array. Weights are transformed offline into block-wise sparse pattern and stored in a set-associate compact format. The top controller sends weights from the global weight SRAM to CIM macros and executes multiply-accumulation (MAC) operations in weight-stationary mode. The exchangeable CIM banks alternately execute computation and weight update operations, enabling full-time utilization. The SSM utilizes weight indexes and online activation sparsity to select activated CIM columns and ADC precision. The 64 set-associate adders receive CIM results with indexes and accumulate them to the correct results. A 32b accumulation adder dynamically skips zeros and accumulates the partial-sums in the result SRAM.

Figure 15.2.3 presents the SABZA and the digital-predictor-assisted ADC sensing mechanism. In CIM macros, each two adjacent CIM banks compose a set-associate group to calculate results for 8 output channels. Each group of *N B*-bit weight data (B=1-8b) is accompanied by a 7b index, illustrating sparsity, row and column number. The 112b indexes for 16 CIM banks are sent to SSM, which dynamically detects activation sparsity. The weight/activation sparsity is classified into 3 patterns: all-zero, <4 non-zeros and ≥4 non-zeros. The all-zero activation group is skipped. When an all-zero weight group appears due to the misalignment or a very sparse weight distribution, the corresponding ADCs are powered off. The <4 and ≥4 non-zeros patterns are calculated under 2/4b ADC sensing mode.

Every two ADCs in the same CIM bank are for 1b activation (H1b/L1b), and the results are accumulated in the local adder array. The set-associate group contains two CIM banks and eight set-associate adders. Results from the local adders are selected to be added to specific registers by recognizing the corresponding output channel indexes. Fig. 15.2.3 presents a 2-row 8-output set-associate case to illustrate the compact weight format and workflow. As the activation ID and the second weight index in the first column is inconsistent due to the compact weight storage format, one more clock cycle (Cycle 3) is needed. This situation can be optimized during the NN network training stage. In this example with 75% sparsity, a 4× macro storage reduction and 3.2× cycle reduction are achieved by zero skipping.

Figure 15.2.4 shows the ping-pong CIM (PP-CIM) architecture. The ~100Kb CIM macro cannot store all weight data for large NN models, often requiring >1MB. The PP-CIM architecture supports simultaneous weight updates and computing operations. In Phase 1, weights are written into the alternate banks of the CIM macros. In Phase 2, computing and weight update banks exchange. The CIM banks with updated weights are used for computing, while the outdated computing banks start weight updates for the next workload. In case the weight transfer is complete while the current CIM operations are not finished, the weight SRAM goes into the retention mode to save power. Fig. 15.2.4 shows the PP-CIM improvement on different NN layers and models, illustrating different performances with varying ratios of weight-update and computing operations. For a ResNet18 model on CIFAR-10, 1.26× overall performance improvement is observed by reducing idle weight-update cycles. The improvement can be 1.94× in several layers. Over the models on the CIFAR-10 and ImageNet dataset, the PP-CIM architecture shows a 1.02-to-1.30× performance gain and 46-to-98% weight SRAM retention time. Compared with a simple double-buffering strategy requiring an additional 100% macro area, the PP-CIM only increases the macro area by approximately 15%.

Figure 15.2.5 presents the details of the PP-CIM macro comprising the SRAM cell array, input driver, normal IO and 32 ADCs. The 128×128 cell array is divided into 16×128 PP-CIM banks. Each bank contains two bitwise multiply units (A-BMU, B-BMU) and one local computing cell (LCC). Each BMU contains four compact 6T SRAM cells. With the additional B-GBL/B-GBLB pair, this macro supports simultaneous computing and weight-update operations. When one BMU is selected to run the computing operation, the other one can perform weight updates. In the SRAM read/write mode, the SRAM cell is accessed through local bitline pairs (LBL/LBLB), and connected to the vertical global bitline (B-GBL) with HWL=V_{DD}. In the CIM mode for A-BMU, A-HWL turns off M1 and M2. M3-M6 run the multiply operation (IN•W), wherein weight data stored in an SRAM cell is accessed via LBL/LBLB, and two separated 1b activation inputs are rail-to-rail voltage and accessed via A-GBL/A-GBLB. The results go through the horizontal global bitline (HGBL/HGBLB) to two ADCs to get the final results. In each cycle, the CIM macro activates a maximum of 32 columns for a CIM operation, with no overflow ensured by the digital predictor of SSM. The HGBL/HGBLB pair achieves 2.07× better sensing margin compared with a previous 5b ADC [4]. Each ADC can switch between 2b/4b sensing and power off (0b) modes to reduce power consumption.

This processor is fabricated in 65nm CMOS technology and measurement results are shown in Fig. 15.2.6. This chip has separate power domains for the CIM macros and other digital circuits. This chip can work at 25-100MHz with 0.62-1.0V digital voltage and 1.0V CIM voltage. The tested ADC current shows a 1.33× average power reduction using the digital predictor. The deployed NN models include VGG and ResNet on the CIFAR-10 and ImageNet dataset. This chip achieves a 3.91-to-4.57× on-chip storage reduction, with 2.75-to-18.0TOPS/W system energy efficiency for real NN models considering all on-chip power and execution time. The peak system-level energy efficiency is 75.9TOPS/W at 37.5MHz with 0.65V digital and 1.0V CIM voltage. This work demonstrates a CIM chip with both zero-activation and zero-weight skipping and a PP-CIM architecture with simultaneous computing and weight-update capability and shows reasonable accuracy on larger datasets CIFAR-10 (2.24% loss) and ImageNet (3.45% loss) using a digital-predictor-assisted adaptive-precision ADC. An energy-efficiency breakdown shows 6.35× improvement compared with state-of-the-art [5]. Fig. 15.2.7 shows the summary table and die photo.

Acknowledgement:
This work was supported in part by National Key R&D Program 2018YFA0701500; NSFC Grant 61674094, 61934005, 61720106013; Beijing National Research Center for Information Science and Technology; and Beijing Innovation Center for Future Chips. Corresponding Author: Yongpan Liu.

References:

[1] X. Si et al., "A Twin-8T SRAM Computation-In-Memory Macro for Multiple-Bit CNN-Based Machine Learning," *ISSCC*, pp. 396-397, 2019.
[2] J. Yang et al., "Sandwich-RAM: An Energy-Efficient In-Memory BWN Architecture with Pulse-Width Modulation," *ISSCC*, pp. 394-395, 2019.
[3] J. Su et al., "A 28nm 64Kb Inference-Training Two-Way Transpose Multibit 6T SRAM Compute-in-Memory Macro for AI Edge Chips," *ISSCC,* pp. 240-241, 2020.
[4] X. Si et al., "A 28nm 64Kb 6T SRAM Computing-in-Memory Macro with 8b MAC Operation for AI Edge Chips," *ISSCC*, pp. 246-247, 2020.
[5] J. Yue et al., "A 65nm Computing-in-Memory-Based CNN Processor with 2.9-to-35.8TOPS/W System Energy Efficiency Using Dynamic-Sparsity Performance-Scaling Architecture and Energy-Efficient Inter/Intra-Macro Data Reuse," *ISSCC*, pp. 234-235, 2020.

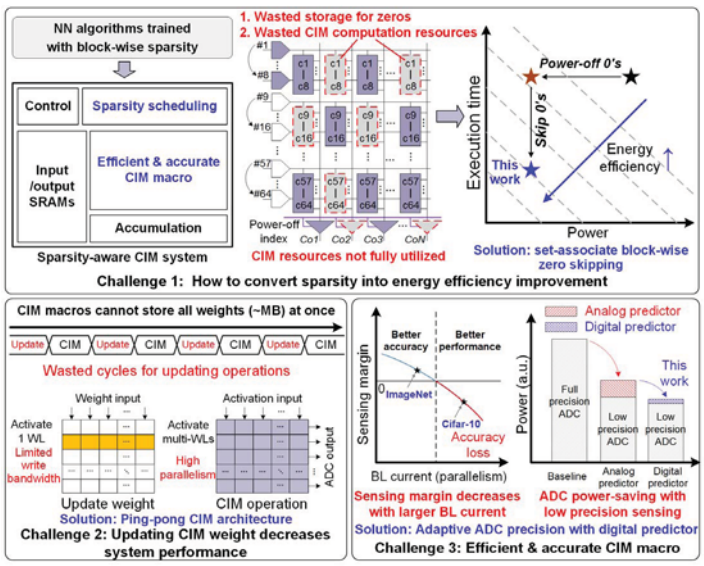

Figure 15.2.1: Design challenges for energy-efficient sparsity-aware CIM neural-network processor.

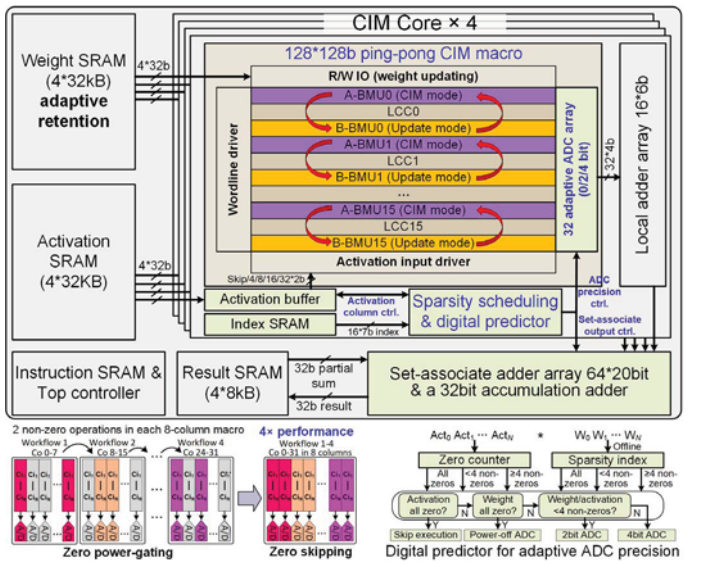

Figure 15.2.2: Overall CIM processor architecture with set-associate block-wise zero-skipping and ping-pong CIM.

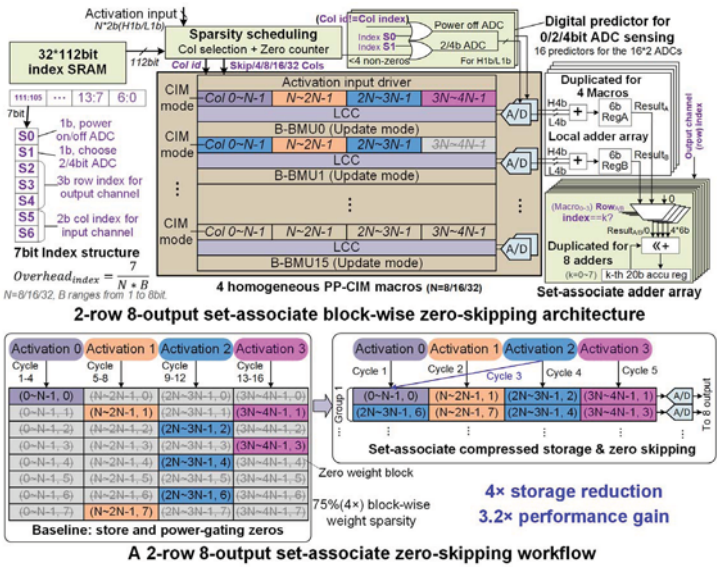

Figure 15.2.3: Workflow of set-associate block-wise zero-skipping and digital-predictor-assisted adaptive ADC sensing.

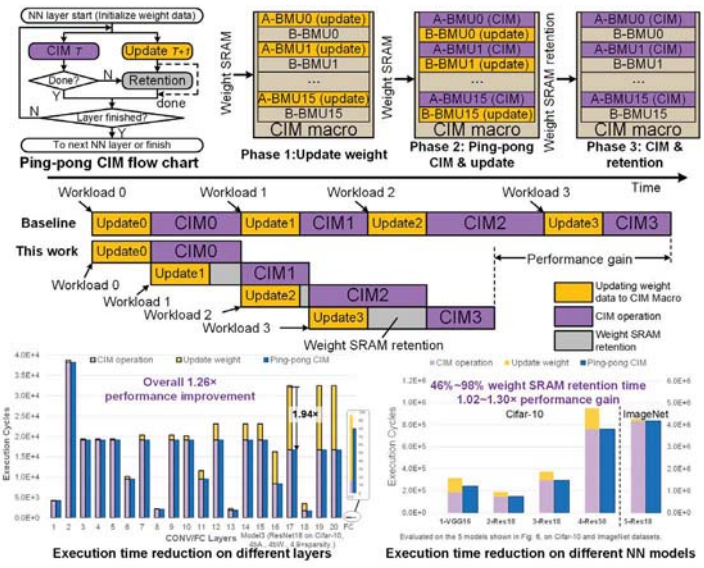

Figure 15.2.4: Ping-pong CIM architecture and performance improvement on different NN layers and models.

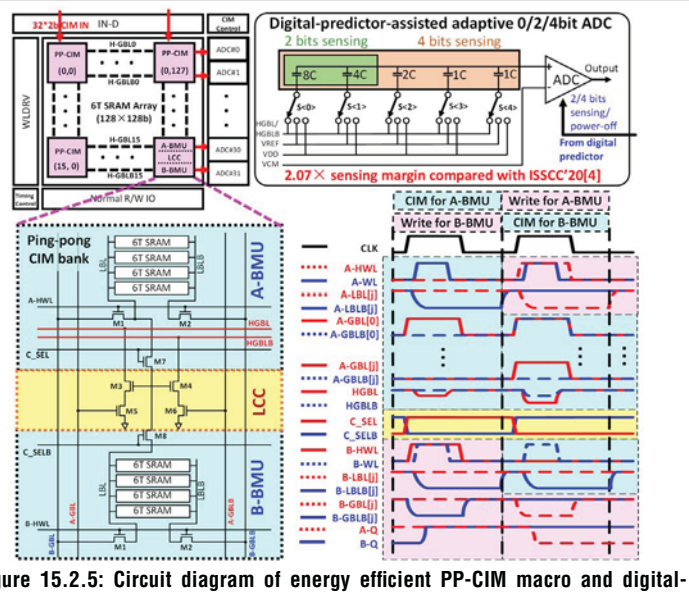

Figure 15.2.5: Circuit diagram of energy efficient PP-CIM macro and digital-predictor-assisted adaptive 0/2/4b ADC.

Dataset	Cifar-10				ImageNet
Model	1-VGG16	2-ResNet18	3-ResNet18	4-ResNet50	5-ResNet18
Sparse block size (N)	32	32	32	32	32
Sparsity compression	4.2×	4.9×	4.9×	4.1×	4.2×
Activation precision	4bit				8bit
Weight precision	4bit	2bit	4bit	4bit	4bit
Top-1 accuracy *2	91.56%	91.71%	92.52%	92.65%	66.31%
Accuracy loss	-2.24%	-3.14%	-2.33%	-2.38%	-3.45%
Weight+index storage compression	3.96×	4.32×	4.57×	3.91×	3.94×
Execution time *3	6.51ms	3.98ms	7.95ms	20.4ms	112ms
CIM macro energy eff. (TOPS/W)	16.73	48.24	24.14	22.03	7.32
System energy eff. (TOPS/W) *3	6.20	18.00	9.01	8.22	2.75

Test and evaluation results on different NN models *1

*1: At 0.65V(digital)+1.0V(CIM macro), 37.5MHz. Off-chip memory accesses and operations (batch normalization, pooling) are required. Off-chip/IO power is not included. Results are evaluated based on the tested typical layers. *2: Accuracy of ImageNet models are evaluated with 2bit mode ADC and worst variation assumption. *3: Include all on-chip execution time and on-chip digital and CIM power.

	ISSCC'19 [1]	ISSCC'19 [2]	ISSCC'20 [3]	ISSCC'20 [4]	ISSCC'20 [5]	This work
Application	CNN/FC	CNN	CNN/FC	CNN/FC	CNN/FC	CNN/FC
Technology	65nm	28nm	28nm	28nm	65nm	65nm
Activation	1/2/4 bit	8 bit	2/4/6/8 bit	4/8 bit	2/4/6/8 bit	2/4/6/8 bit
Weight	2/5 bit	1 bit	4/8 bit	4/8 bit	4/8 bit	1-8 bit
SA/ADC	3/5 bit	1 bit	5 bit	5 bit	5 bit	0/2/4 bit
Output	3/5/7 bit	16 bit	10-20 bit	12/16/20 bit	16/32 bit	32 bit
Digital Storage	N/A	2k bit	N/A	N/A	164KB	294KB
Sparsity Support	N/A	N/A	N/A	N/A	power-off zero execution	skip zero execution
Ping-pong CIM	×	×	×	×	×	√
Cifar100/ImageNet Accuracy	N/A	N/A	N/A	√	N/A	√
Energy Efficiency (TOPS/W) *1†	72.1 (macro)	119.7(macro)	61.1 (macro)	68.4 (macro)	158.7 (macro) 35.8 (system)	370.4 (macro) 75.9 (system)

*1: 1 operation (OP) = 1 multiplication/addition. Presented OP data are equivalent to the non-sparse models. *2: [5] is for MNIST and does not include time for updating weight into CIM macros. *3: include updating time. 2b activation, 1b weight. Macro efficiency@1.0V,100MHz. System @0.65V(digital)+1.0V (CIM), 37.5MHz.

Comparison with state-of-the-art CIM processors

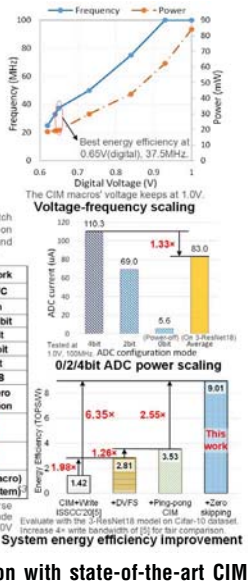

Figure 15.2.6: Measurement results and comparison with state-of-the-art CIM processors.

Technology	TSMC 65nm GP
Chip Area	3.0mm×4.0mm
Core Area	2.44mm×3.41mm
CIM Macro Area	0.69mm×0.62mm×4
Application	CNN / FC
Weight Bit-precision	1-8 bit
Activition Bit-precision	2/4/6/8 bit
Total Digital SRAM	294kB
CIM SRAM	16kbit × 4
Supply Voltage (Digital)	0.62-1.0V
Supply Voltage (Analog)	1.0V
Frequency	25-100MHz
CIM Macro Power*1	2.55mW × 4
System Power	18.6-84.1mW
Performance*2	0.10-3.16TOPS
CIM Macro*3 Energy Efficiency	**7.32-370TOPS/W**
System *3 Energy Efficiency	**2.75-75.9TOPS/W**

*1: Test at 1.0V, 100MHz.
*2: 25-100MHz, from non-sparse models to 4× sparse models, 2bit activation, 1bit weight.
*3: Off-chip and IO power is not included. Off-chip operation time (batch normalization, pooling) is not included.

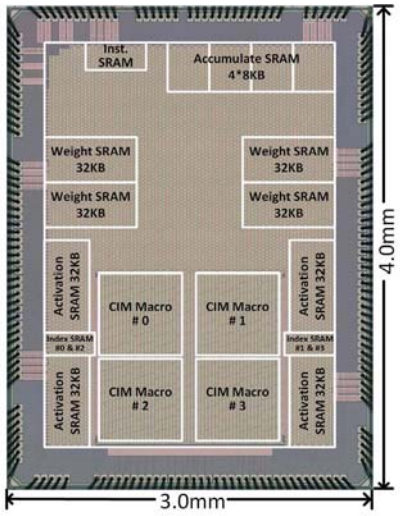

Figure 15.2.7: Die photo and metrics.

978-1-7281-9549-0/21/$31.00 ©2021 IEEE