A 10T SRAM Computing-in-Memory Unit-Macro with Multibit MAC Operation for AI Edge Processors

Van Truong Nguyen, Nguyen Bui Huu, Xuan Thanh Pham, and Jong-Wook Lee
Department of Electronics Engineering, Kyung Hee University
E-mail: *nguyenvantruong.bk@gmail.com, nguyencv90@khu.ac.kr, thanhpham@khu.ac.kr, jwlee@khu.ac.kr*

Abstract

Computing-in-memory (CIM) is a promising approach to reduce latency and improve the energy efficiency of multiply-and-accumulate (MAC) operation under a memory-wall constraint for artificial intelligence (AI) edge processors. This design presents an energy-efficient static random access memory (SRAM) – CIM unit-macro using: 1) a 10T bit-cell SRAM array stores multiple-bit filter weight; 2) a multiple-bit MAC operation scheme with up to 4b input, 4b weight, and 8b output precision for CNN applications; 3) a successive approximation analog to digital converter (SAR-ADC) within cell array to reduce area overhead and power consumption.

I. INTRODUCTION

Deep neural networks (DNNs) have achieved breakthroughs in a wide variety of artificial intelligence (AI) and machine learning (ML) applications, including image classification [1], speech recognition [2], and facial recognition [3]. While DNN promises significant benefits for the "Internet of Thing" (IoT) devices, it also has specific requirements. The DNN processors and accelerators run the computing algorithms that must be energy efficient to extend the battery life of these IoT devices.

Convolutional neural networks (CNNs) provide state of the art results in a wide variety of AI/ML applications, ranging from image classification [1] to speech recognition [2]. However, the conventional all-digital implementation of CNNs [4]-[6] has shown that energy consumption and delays are dominated by the frequent movement of input data, weights, and intermediate data between the processor and memory. This issue is referred to as the von Neumann bottleneck or memory wall. Multiply-and-accumulate (MAC) operations are essential to CNN accelerators.

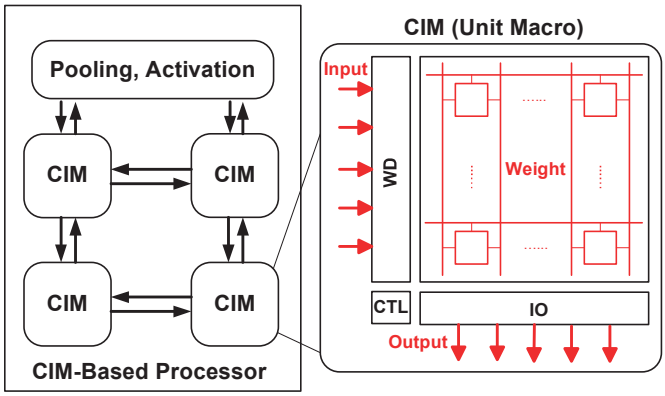

Figure 1. Conceptual of SRAM-CIM for DNN processors.

Computing-in-Memory (CIM) methods [7]-[10] have been developed to reduce the energy consumption of CNN processors by enabling parallel data processing within memory, as shown in Fig. 1. Rather than accessing raw data row by row in each column (as in conventional memory), CIM allows the execution of MAC operations in multiple rows simultaneously. This property significantly reduces the amount of intermediate data that is generated and facilitates highly parallel computation. Up to this point, some silicon verified SRAM-based CIM devices had been reported, including an error-adaptive binarized classifier for Mixed National Institute of Standards and Technology (MNIST) dataset [7], a Conv-RAM for binary weight and 6-bit input/output neural networks [8], a Xcell-RAM for binary neural networks [9], and an XNOR-SRAM for binary/ternary DNNs [10]. These SRAM-CIM works have demonstrated various benefits of CIM in terms of functionality and improved energy efficiency. However, advanced AI edge processors require multibit input (IN), weight (W), and output (OUT) for CNN MAC operations to achieve an inference accuracy that is sufficient for practical applications.

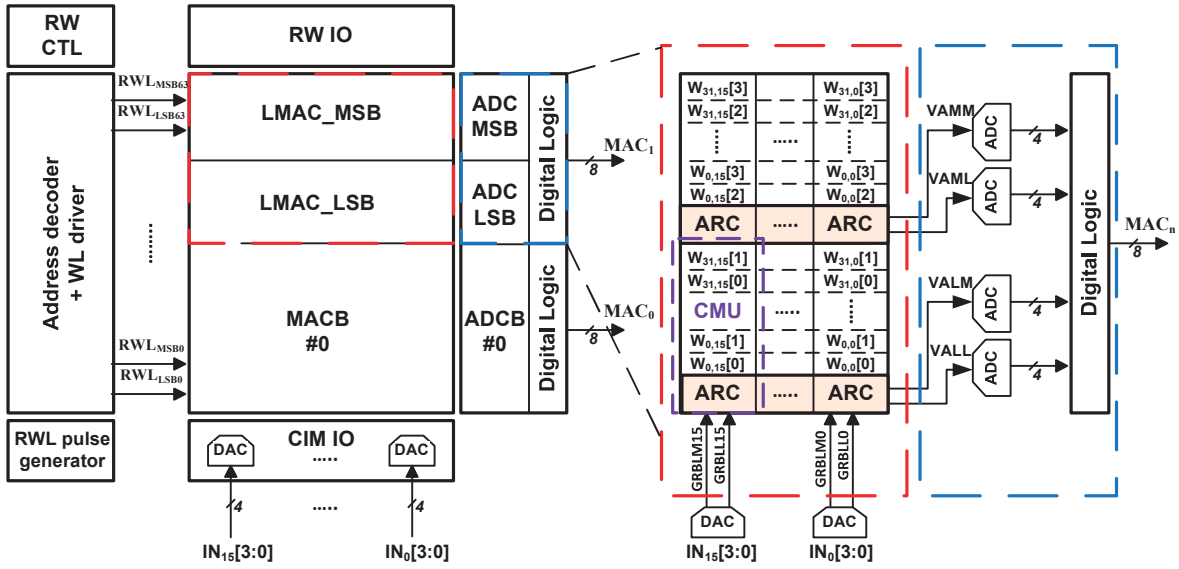

Figure 2. The overall structure of the proposed 10T SRAM-CIM unit-macro.

In this design, we proposed a configurable 10T SRAM-CIM unit-macro with 4-bit inputs, 4-bit weights, and 8-bit outputs for various multibit CNN applications. A 2-Kb 10T SRAM-CIM unit-macro was designed for prototyping.

II. DESIGN

2.1 Macro Architecture

Fig. 2 presents the structure of the proposed 10T SRAM-CIM unit-macro. It consists of 128 rows x 16 columns of 10T SRAM bit-cells separated into two MAC block (MACB), column-wise digital to analog converter (DAC), read word-line (RWL) pulse generator, two ADC blocks (ADCB) and digital logic blocks. Each MACB is divided into two local MAC block storing 4b-weights Wi,j[3:0] (2b MSB Wi,j[3:2] is stored in LMAC_MSB and 2b LSB Wi,j[1:0] is stored in LMAC_LSB) in the same row-column position. Each LMAC block comprises 16 column-wise multiplication units (CMUs). Each CMU has 16 10T pair cells and one accumulation and reference cell (ARC), which supports two multiplication channels for 2b-input and 2b-weight in each channel.

The 10T SRAM CIM unit-macro can be operated in two modes: SRAM mode and CIM mode. In SRAM mode, the stored weights are accessed using a standard read/write peripheral circuit via a single active word-line. In CIM mode, each CMU computes input-weight-product (IWP) between an activated pair 2b weights and 2b inputs on two channels simultaneously, and then 16 column-wise multiply results are accumulated through ARC to generate two analog voltage (VA) outputs in each LMAC block. Consequently, VA outputs are sent to ADCB and then digital logic block for weight combination to obtain 8b outputs MAC. The proposed 10T SRAM-CIM unit-macro supports two parallel multibit MAC operations on two MACBs.

2.2 Cell Design

We use a 10T-SRAM cell as the basic memory unit. Two 10T-SRAM cells form a pair: the most-significant 10T (M10T) and the least-significant 10T (L10T), as shown in Fig. 3. Each 10T-SRAM cell contains the basic 6T-cell as the storage unit, along with transistors N0-N1 and N2-N3, forming the differential read ports, respectively. The overhead area ratio compared with the 6T bit-cell is 1.52. In the SRAM mode, the 10T-SRAM cell works functionally similar to the standard 6T cell through the ports (WWL_M/WLL_L, BL, BLB). In the CIM mode, a 4b-input (IN[3:0]) is split into two groups and applied to DACs to pre-charge RBL_M (IN[3:2]) and RBL_L (IN[1:0]). Each RBL_M/RBL_L uses four voltage levels to represent a 2b-input. The read current (I_{RC}) on two read bit-lines (RBL_M/RBL_L) is proportional to the multiplication between

the number of RWL_M/RWL_L pulse and 2-bit weight (W[1:0]) stored in 10T pair cell. As a result, the output voltage on each RBL_M/RBL_L represents the IWP of 2b-input and 2b-weight (IWP_{RBLM} = IN[3:2] x W[1:0], IWP_{RBLL} = IN[1:0] x W[1:0]).

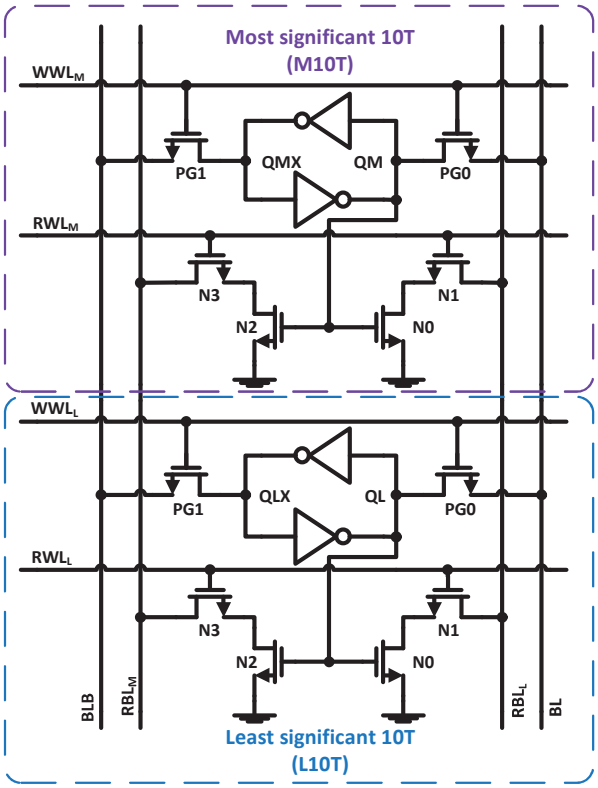

Figure 3. Schematic of 10T SRAM bit-cell.

2.3 Current Steering DAC

During the first phase of the multibit MAC operation, the 4-bit inputs (IN[3:0]) are split into two groups and converted to the analog voltage on global read bit-line $GRBL_M$ and $GRBL_L$ corresponding to IN[3:2] and IN[1:0] using current steering DAC. Fig. 4 shows the schematic of the proposed current steering DAC. It consists of two groups of binary-weighted current sources using cascode pMOS stack biased in the saturation region. The $GRBL_M$/$GRBL_L$ voltage level is proportional to the digital input applied to its column.

2.4 Multiply-and-Accumulate (MAC) Scheme

Fig. 5 shows the detailed circuit for the multiplication of 4b inputs and the 4b weights stored in the array. After completing the first phase, local RBL_M/RBL_L is pre-charged through $GRBL_M$/$GRBL_L$ to an analog voltage proportional with 4b input. The second phase starts by

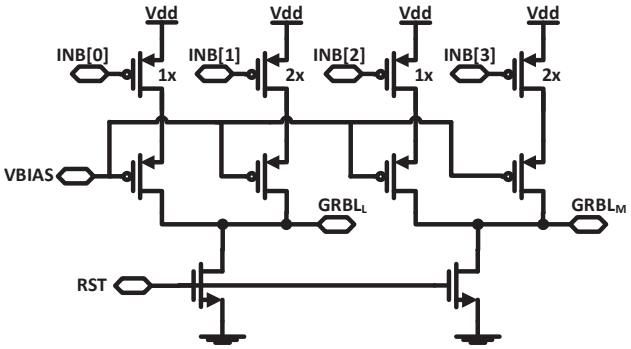

Figure 4. Schematic of the current steering DAC.

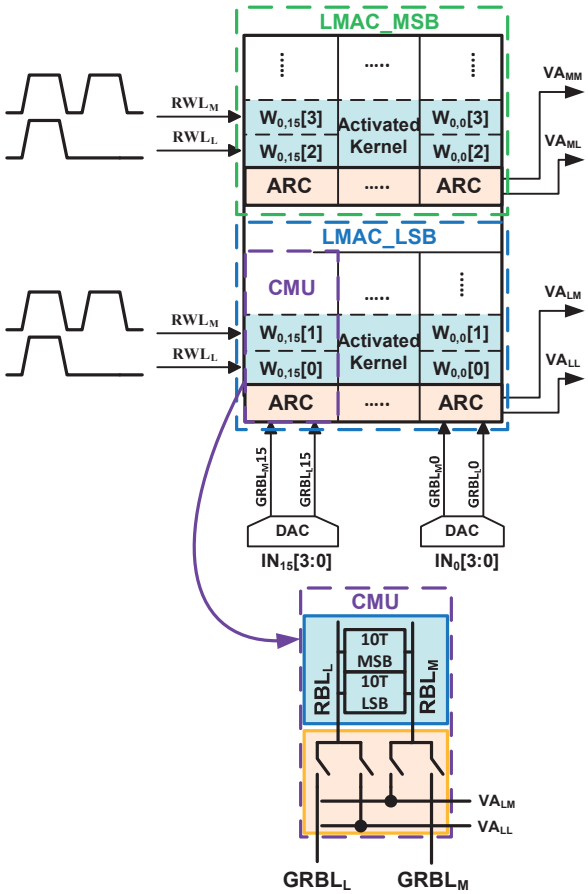

Figure 5. Multibit MAC Scheme.

activating RWL_M/RWL_L appropriated with the selective row. The multibit weights realized by the number of RWL pulses. After multiplying 2b input and 2b weight on the RBL_M/RBL_L, the results are accumulated and generate four analog voltage outputs VA_{LL}, VA_{LM}, VA_{ML}, and VA_{MM}. The output VA_{LL} represents $\sum$(INi[1:0] x Wp,i[1:0]), VA_{LM} represents $\sum$(INi[1:0] x Wp,i[3:2]), VA_{ML} represents $\sum$(INi[3:2] x Wp,i[1:0]), and VA_{MM} represents $\sum$(INi[3:2]

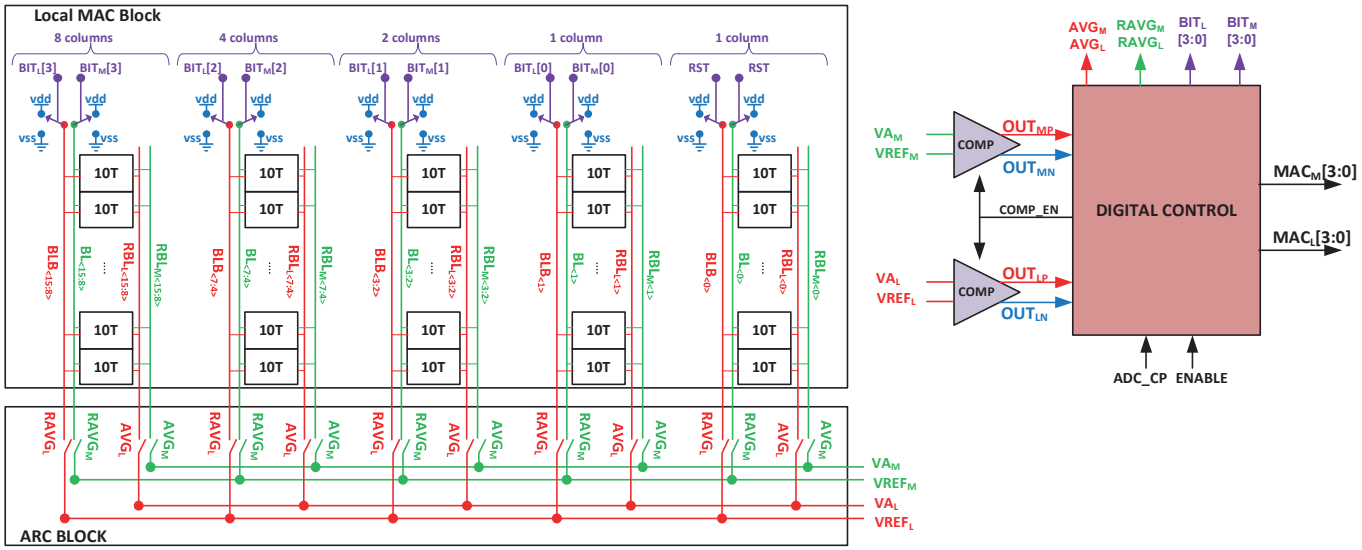

Figure 6. Architecture for the SAR ADC for one local array of the SRAM-CIM.

x Wp,i[3:2]). These outputs are sent to ADCB and then shift register and adder block to obtain the final 8b MAC output representing ∑(INi[3:0] x Wp,i[3:0]).

2.5 Successive Approximation ADC

The last phase of the multibit MAC operation is the analog-to-digital conversion with 4b outputs resolution. This phase is done in parallel for all LMAC blocks, producing outputs corresponding to different filters simultaneously. Fig. 6 shows the proposed SAR ADC architecture correlated to each LMAC block. It comprises three main parts: two comparators for two-channel outputs from ARC block, a digital control block, and a capacitive DAC using the inherent capacitance of 16 BL/BLB pairs in the local array. Utilizing the distributed intrinsic BL/BLB capacitance for this architecture is the essential technique to reduce the overhead area problem of SAR ADR. The comparator has two standard StrongARM latches using nMOS and pMOS devices for the input differential pair, respectively. The digital control block generates timing signals (BIT_L[3:0], BIT_M[3:0], AVG_L/AVG_M, $RAVG_L$/$RAVG_M$) to perform the binary search algorithm on two channels $VREF_M$ and $VREF_L$, starting from the MSB to the LSB. Once this is done, the conversion is complete, and two 4-bit outputs are available for two channels, MAC_M[3:0] and MAC_L[3:0], respectively. It takes 4 ADC clock cycles to complete the conversion of 4-bit output resolution.

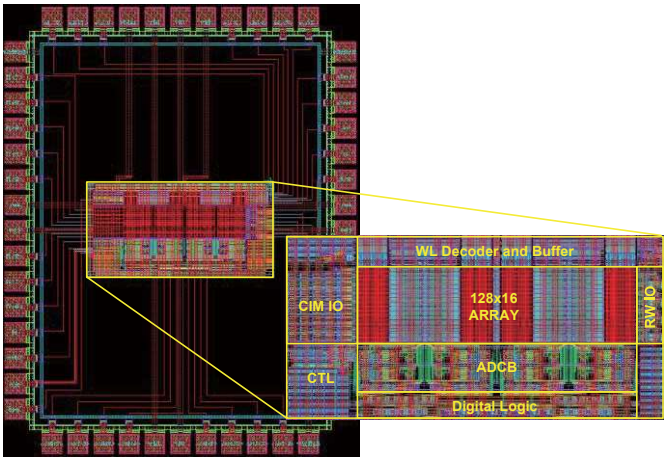

Figure 7. Test-chip layout.

III. RESULTS

The 2-Kb 10T SRAM-CIM was implemented in a 180-nm CMOS logic process. The chip photo and summary are presented in Fig. 7 and Table I, respectively.

IV. CONCLUSIONS

A 10T SRAM-CIM unit-macro to support multibit MAC operation for CNNs has been presented. The proposed SRAM-CIM unit-macro includes 10T SRAM array, a current steering DAC, a multibit MAC operation scheme with up to 4b input, 4b weight, and 8b output precision for CNN applications, and a successive approximation ADC. A

128x16 SRAM-CIM unit-macro was designed and shown a prospect result with 1.28 GOPS throughput, and energy efficiency of 0.496 TOPS/W.

TABLE I
TEST-CHIP SUMMARY

Technology	180-nm
Unit-macro size	2Kb
Bit-cell	10T
Bit-cell size	1.48μm x 7.19μm
Input precision (bit)	4
Weight precision (bit)	4
Output precision (bit)	8
Main Clock	20 MHz
ADC Clock	160 MHz
Throughput (GOPS)	1.28
Energy Efficiency (TOPS/W)	0.496

VI. REFERENCES

[1] A.Krizhevsky, I.Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Process. Syst., 2012, pp.1097-1105.

[2] G. Hinton et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal Process. Mag., vol. 29, no.6, pp. 82-97, Nov. 2012.

[3] Y. Taigman, M. Yang, M.'A. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in Proc. IEEE CVPR, pp. 1701-1708, Jun. 2014.

[4] Y. H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy efficient reconfigurable accelerator for deep convolutional neural networks," IEEE J. Solid-State Circuits, vol. 52, no. 1, pp. 127-138, Jan. 2017.

[5] B. Moons and M. Verhelst, "An energy-efficient precision-scalable ConvNet processor in 40-nm CMOS," IEEE J. Solid-State Circuits, vol. 52, no. 4, pp. 903-914, Apr. 2017.

[6] N. Whatmough, S. K. Lee, H. Lee, S. Rama, D. Brooks, and G. Y. Wei, "A 28 nm SoC with a 1.2 GHz 568 nJ/prediction sparse deep neural network engine with > 0.1 timing error rate tolerance for IoT applications," IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, pp. 242-243, Feb. 2017.

[7] J. Zhang, Z. Wang, and N. Verma, "In-memory computation of a machine-learning classifier in a standard 6T SRAM array," IEEE J. Solid-State Circuits, vol. 52, no. 4, pp. 915-924, Apr. 2017.

[8] A. Biswas and A. P. Chandrakasan, "Conv-RAM: An energy efficient SRAM with embedded convolution computation for low-power CNN-based machine learning applications," IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, pp. 488-490, Feb. 2018.

[9] A. Agrawal et al., "Xcell-RAM: Accelerating binary neural networks in high-throughput SRAM compute arrays," Jul. 2018, arXiv:1807.00343. [Online]. Available: https://arxiv.org/abs/1807.00343

[10] Z. Jiang, S. Yin, M. Seok, and J.-S. Seo, "XNOR-SRAM: In-memory computing SRAM macro for binary/ternary deep neural networks," in Proc. IEEE Symp. VLSI Technol., Honolulu, HI, USA, Jun. 2018, pp. 173-174.