15.3 A 65nm 3T Dynamic Analog RAM-Based Computing-in-Memory Macro and CNN Accelerator with Retention Enhancement, Adaptive Analog Sparsity and 44TOPS/W System Energy Efficiency

Zhengyu Chen, Xi Chen, Jie Gu

Northwestern University, Evanston, IL

Computing-In-Memory (CIM) techniques which incorporate analog computing inside memory macros have shown significant advantages in computing efficiency for deep learning applications. While earlier CIM macros were limited by lower bit precision, e.g. binary weight in [1], recent works have shown 4-to-8b precision for the weights/inputs and up to 20b for the output values [2, 3]. Sparsity and application features have also been exploited at the system level to further improve the computation efficiency [4, 5]. To enable higher precision, bit-wise operations were commonly utilized [3, 4]. However, there are limitations in existing solutions using the bit-wise operations with SRAM cells. Fig. 15.3.1 shows the summary of challenges and solutions in this work. First, all existing solutions utilize 6T/8T/10T SRAM as a CIM cell, which fundamentally limits the size of the CIM array. In this work, we replace the commonly used SRAM cell with a 3-transistor (3T) analog memory cell, referred as dynamic-analog-RAM (DARAM) which represents a 4b weight value as an analog voltage. This leads to ~10× reduction in transistor count and achieves an effective CIM single-bit area smaller than the foundry-supplied 6T SRAM cell. Secondly, as no bit-wise calculation is needed in this work, only single-phase MAC operation are performed, removing the throughput degradation associated with previous multi-phase approaches and digital accumulation in [3, 4]. Furthermore, analog linearity issues are mitigated by highly linear time-based activation, removal of matching requirements for critical multi-bit caps [4, 6], and a special read current compensation technique. Thirdly, to mitigate the power bottleneck of ADC or SA, this work applies analog sparsity-based low-power methods, which include a compute-adaptive ADC skipping operation when the analog MAC value is small (or "sparse") and a special weight-shifting technique, leading to an additional ~2× reduction in CIM-macro power. We demonstrate the proposed techniques using a 65nm CIM-based CNN accelerator showing state-of-art energy efficiency.

Figure 15.3.2 shows the 3T dynamic-analog-RAM (DARAM). Similar to a conventional CIM bit cell, the charge drawn to BL_R is proportional to the multiplication of read current I_{mem} from the read-access transistor M1 and the time-pulse duration of RE through switch M2. A 4b weight is stored as an analog voltage on the internal "MEM" node generating a read current proportional to the weight value. Due to the 4b lumped analog weight, a 4b MAC operation is realized by a single read of the DARAM, which is considerably simpler than prior bit-wise operation approaches. Designed with regular logic transistors, the critical read-access transistor M1 is sized with larger W and L to reduce device variation. The DARAM cell has an area 1.9× larger than a previous 8T CIM cell and 3× larger than a foundry 6T SRAM cell leading to an effective bit area of 47% of the 8T CIM cell and 75% of the foundry 6T SRAM cell [3]. During write, write-access transistor M3 is used to write the analog voltage from BL_W to the "MEM" node from a column-wise DAC with an adjustable voltage range from 0.45-1V. Each write can be finished within one clock cycle with a total of 64 clock cycles to write the entire CIM macro. Subthreshold and gate leakage are minimized to maintain a constant analog voltage during the life cycle of stationary weights for the CNN operation. As shown in Fig. 15.3.1, the weight stationary cycles of CNN models (e.g. VGG16 or ResNet18) vary from tens of cycles to thousands of cycles for a single image and increase proportionally with the batch size, driving the retention requirements for the analog voltage. A special 3D inter-layer and inter-digit metal capacitor using M1 to M5 interleaving MEM and GND nodes vertically and horizontally is added inside each DARAM cell to enhance the storage capacitance by 3×. As shown in Fig. 15.3.2, during CNN inference, separate biasing of BL_W at 0.8V leads to about a 20× reduction in subthreshold leakage current. This allows a retention time of ~41k cycles (for a voltage drift less than half of a single bit) at typical corner and more than 5k cycles at a fast corner. As a result, a batch size of 5-to-40 images can be processed without a rewrite (refresh) operation with negligible accuracy loss. For a larger batch size, a 64-cycle DARAM refresh operation is needed at every 5.5-to-41k cycles, leading to a throughput overhead of less than 1.2% or a CIM macro energy overhead of less than 0.4%. Note for a smaller batch size or CNN layers with less stationary weights, refresh is not needed.

Figure 15.3.3 shows the architecture of the CNN accelerator with 4 CIM macros. Each CIM macro contains a 64×32 DARAM array. A row-wise digital-time-converter (DTC) is used to convert a 4b activation into a time pulse with 50ps resolution. A 5b SAR ADC and a 4b current DAC are implemented at each column to provide MAC read-out and analog write-in. The design natively supports 4b/4b input/weight operation and can also support 8b/8b by combining two DARAM cells and operating in successive two cycles. Similar to prior schemes, global SRAMs are used to store weight and input/output activation data before being fetched into CIM macro. An ASIC core is used to manage data sequencing and pre/post-processing including (a) offsetting of data values due to the non-2's complementary format of weights in comparison with the support of both non-2's and 2's complement formats in prior works [3, 4]. The offset calculation has negligible overhead as it is commonly shared by all the columns; (b) 4-to-8b conversion if needed; (c) accumulation at the inter-macro loop similar to [4]. An additional three features are introduced in this work: (1) An input-stationary operation mode is supported, which is more efficient for later layers in VGG/ResNet. (2) A special analog weight shifting technique is introduced where the weights are shifted down whenever the weight range in a column is not fully utilized, thereby reducing MAC energy consumption which favors lower weight values. The shifted weights are pre-determined off-chip according to the weights being used and the associated MAC offsets are added back in the ASIC to restore the values. As shown in Fig. 15.3.3, an average of 3b weight shifting is achieved, providing a 1.5× energy reduction for MAC operations. (3) Input sparsity is leveraged by detecting zero inputs from the ASIC and disabling row-wise DTC and the associated MAC operations in the CIM macro.

Figure 15.3.4 presents the ADC skipping technique exploiting "analog sparsity" in MAC operations to save the dominant ADC power in the CIM macro. As shown in the histogram of the bitline voltage drop, i.e. the analog MAC value, based on the VGG model, over 60% of the cases have a bitline voltage drop less than 27% of full swing leading to the possibility of merging two or more MAC accumulations without activating the ADC and bitline precharge with small accuracy degradation of 0.1-0.4% arising from occasional overflow. This differs from [2] which only reduces the ADC conversion steps at low MAC values. This work skips the entire ADC operations leading to higher energy savings: an average ADC power reduction of 2.4×. In addition, we invoke early termination of a MAC operation based on the ReLU function, i.e. the accumulation has become negative enough that the sign of accumulation results cannot be flipped by the remaining MAC operations. The detection is performed in the ASIC according to a preset negative threshold. Combining both approaches, an average of about 2.9× savings can be achieved in ADC energy consumption. Figure 15.3.4 also shows a nonlinearity compensation scheme, where the nonlinear relationship between the bitline current and MEM voltage from the read transistor M1 is compensated by a non-linear analog voltage generated from the DAC. As a result, a highly linear I_{mem} vs. weight is achieved.

A 65nm CMOS test chip was fabricated to demonstrate the DARAM in a CNN accelerator running at 105MHz at 1V. Calibration was performed to remove variation impacts, e.g. ADC, DAC offset, etc. by adding small offsets in the ASIC. As shown in the measurement results in Fig. 15.3.5, a retention time of up to 0.36ms (38k cycles) without refresh was observed with negligible accuracy degradation supporting a batch size of 37 images in VGG16. With larger batch size, the refresh operations incurred up to 0.17% throughput overhead. The ADC skipping scheme brings a 65% saving of ADC energy with less than 0.4% accuracy impact using a 27% of the bitline full swing as the skipping threshold. Combining all sparsity features, the macro power was reduced by 2.1×, on average, for the VGG16 model. The CNN accelerator was measured from 1.1V down to 0.85V showing a system efficiency from 29TOPS/W to 37TOPS/W without sparsity enhancement. A comparison with prior work is shown in Fig. 15.3.6. Compared to the closest system implementation in [4], at 4b weight/input operation, an 8× system energy efficiency improvement at 44.7TOPS/W is achieved along with 3× area reduction in macro size. Overall, this work achieves a macro efficiency of 217TOPS/W at 4b, which is 3× higher than those reported in closer technologies and is only 32% lower than that reported in a recent 7nm technology. In addition, the effective bit cell area is smaller than the foundry-supplied 6T SRAM. Figure 15.3.7 shows the die photo and additional information.

Acknowledgements:
This work was supported in part by the National Science Foundation under grant number CCF-1846424.

References:

[1] W.-S. Khwa et al., "A 65nm 4Kb Algorithm-Dependent Computing-in-Memory SRAM Unit-Macro with 2.3 ns and 55.8 TOPS/W Fully Parallel Product-Sum Operation for Binary DNN Edge Processors," *ISSCC*, pp. 496-497, 2018.

[2] X. Si et al., "A 28nm 65Kb 6T SRAM Computing-in-Memory Macro with 8b MAC Operation for AI Edge Chips", *ISSCC*, pp. 246-247, 2020.

[3] X. Si et al., "A Twin-8T SRAM Computation-In-Memory Macro for Multiple-Bit CNN-Based Machine Learning", *ISSCC*, pp. 396-397, 2019.

[4] 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.

[5] R. Guo et al., "A 5.1pJ/Neuron 127.3μs/Inference RNN-based Speech Recognition Processor using 16 Computing-in-Memory SRAM Macros in 65nm CMOS, *IEEE. Symp. VLSI Circuits*, pp. C120-C121, 2019.

[6] Q. Dong et al., "A 351TOPS/W and 372.4GOPS Compute-in-Memory SRAM Macro in 7nm FinFET CMOS for Machine-Learning Applications", *ISSCC*, pp. 242-243, 2020.

 978-1-7281-9549-0/21/$31.00

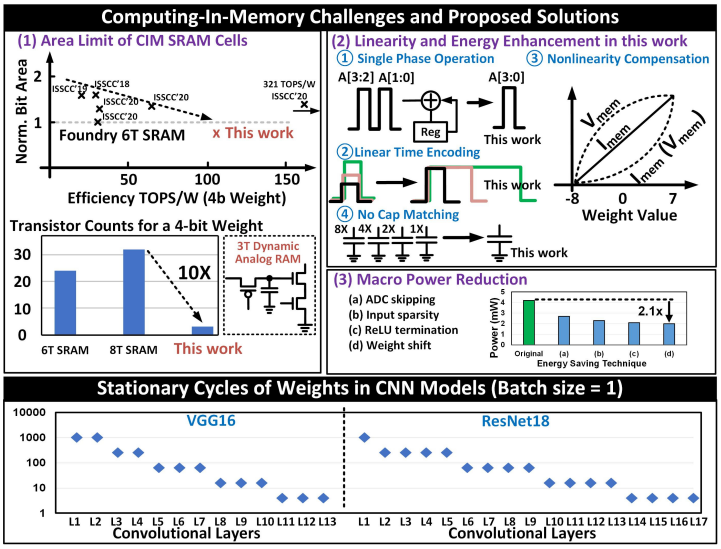


Figure 15.3.1: Challenges of existing computing-in-memory designs and proposed area and energy-efficient solutions using 3T dynamic analog RAM.

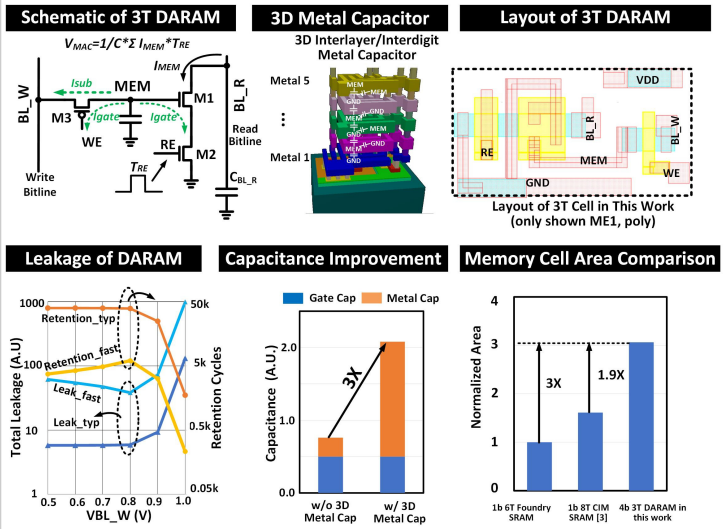


Figure 15.3.2: Design of 3T dynamic analog memory cell with internal 3D metal capacitor, area comparison with prior CIM cell and simulated leakage/retention performance versus write bitline bias.

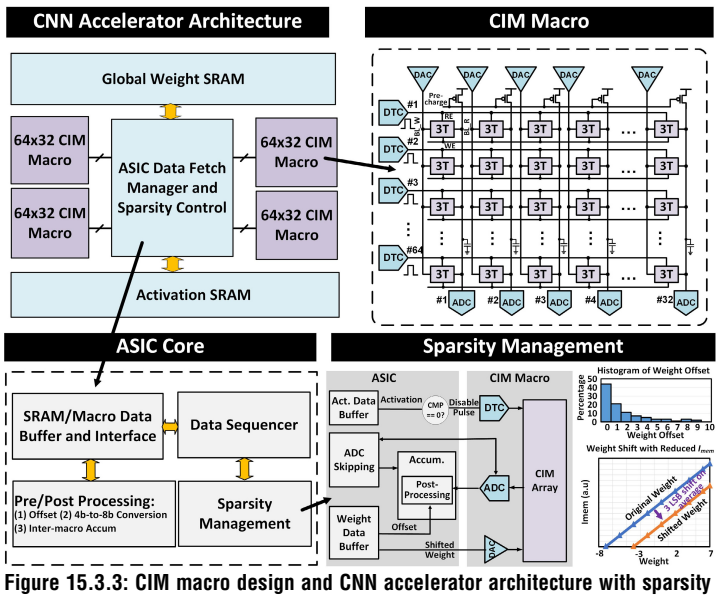


Figure 15.3.3: CIM macro design and CNN accelerator architecture with sparsity management.

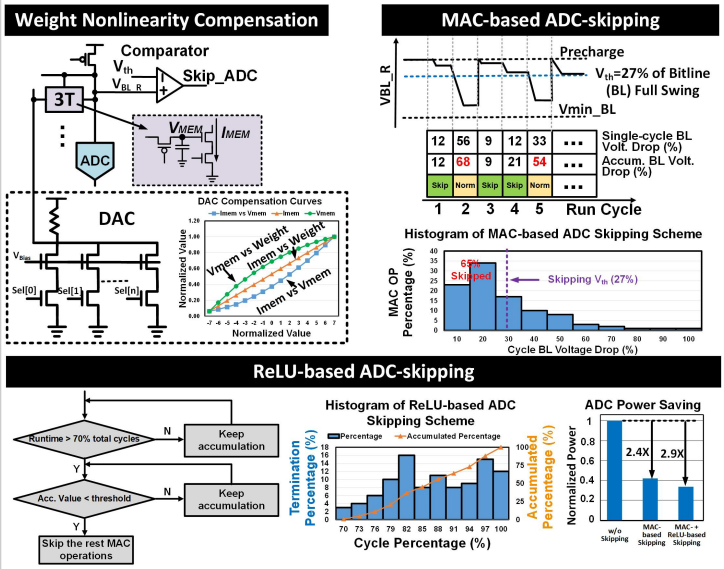


Figure 15.3.4: Compute-adaptive ADC skipping techniques and nonlinearity compensation using non-linear analog memory voltages.

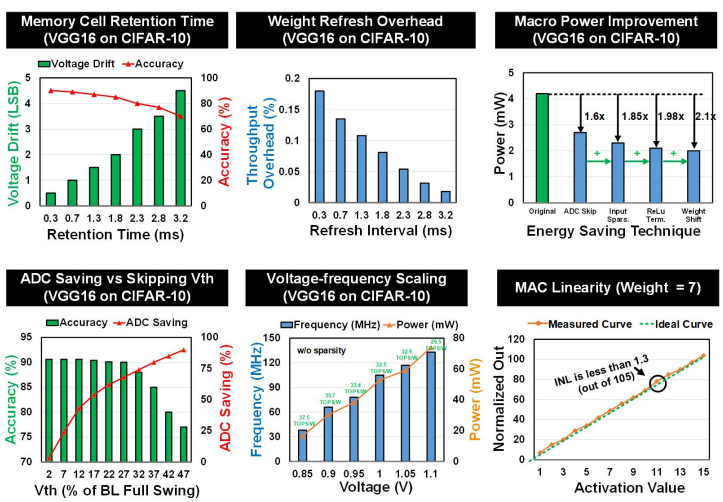
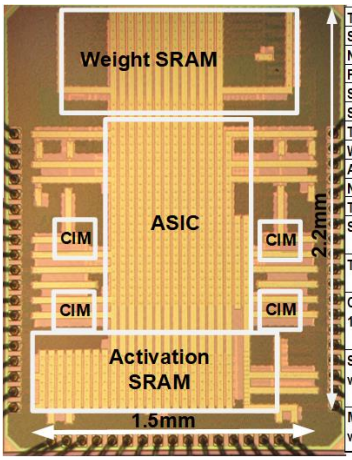


Figure 15.3.5: Measurements results on memory retention time, weight refresh overhead, power improvements through sparsity techniques, ADC skipping Vth impact, voltage-frequency scaling, and MAC linearity.

	[6] ISSCC 2020	[2] ISSCC 2020	[3] ISSCC 2019	[4] ISSCC 2020	This work
Memory Bit	8T SRAM	6T SRAM	Twin-8T SRAM	8T SRAM	3T Analog RAM
Tech. (nm)	7	28	55	65	65
Frequency (MHz)	222	240	-	100	105
System Area (mm2)	-	-	-	9	3.3
Size of Macro (bit)	64x64	512x64	64x60	64x64	4x64x32
Area of Macro (mm2)	0.0032	-	-	0.148	0.05
Activation Precision (bit)	4	4/8	1/2/4	2/4/6/8	4/8
Weight Precision (bit)	4	4/8	2/5	4/8	4/8
ADC Precision (bit)	4	5	5	5	5
Digital Storage	-	-	-	164KB	172KB
Sparsity Support	-	-	-	Activation + weight	Activation + Weight + ADC
Power of CIM Macro (mW)	-	-	-	3.8	4.2 (raw) * 1.98 (w/ sparsity saving)
Power of System (mW)	-	-	-	65	52.8 (raw) * 38.42 (w/ sparsity saving)
Energy Efficiency of CIM Macro at 4bit Weight/Input (TOPS/W)	321	68.44	22.96	25.83	102.2 (raw) * 216.8 (w/ sparsity saving)
Energy Efficiency of System at 4bit Weight/Input (TOPS/W)	-	-	-	5.83	32.5 (raw) * 44.7 (w/ sparsity saving)

* Energy efficiency is reported based on 4b (activation) by 4b (weight) VGG model on CIFAR-10; 1 operation (OP) is either 1 multiplication or 1 addition. Reported energy efficiency includes refresh overhead and excludes off-chip/IO related energy and time. Power and energy efficiency are reported at nominal supply voltage of 1V.

Figure 15.3.6: Comparison table.



Technology	65nm LP
Supply voltage	0.85V-1.1V; (1V nominal)
Nominal Freqency	105MHz
Processor area	1.5mm x 2.2mm
Single CIM macro area	0.23mm x 0.22mm
Single CIM macro array size	64 x 32
Total effective CIM bits	33kb (=4x64x32x4)
Weight bit resolution	4b or 8b
Activation bit resolution	2b or 4b or 8b
Number of channels	64
Total digital SRAM	172KB
Single macro power	4.2mW (w/o sparsity) 2.03mW (w/ sparsity)
Total system power	52.8mW (w/o sparsity) 38.4mW (w/ sparsity)
CIM macro energy efficiency at 1V w/o and w/ sparsity (TOPS/W)	w=4b, in=4b: 102.2 / 216.8 w=8b, in=4b: 53.1 / 112.2 w=8b, in=8b: 26.1 / 54.8
System energy effciency at 1V w/o and w/ sparsity (TOPS/W)	w=4b, in=4b: 32.5 / 44.7 w=8b, in=4b: 16.1 / 22.3 w=8b, in=8b: 8.3 / 10.9
Measured accuracy (CIFAR-10) with 4b weight and input	90.6% (VGG16) 91.2% (ResNet18)

Figure 15.3.7: Die micrograph.

978-1-7281-9549-0/21/$31.00