CIM-SECDED: A 40nm 64Kb Compute In-Memory RRAM Macro with ECC Enabling Reliable Operation

Brian Crafton[1], Samuel Spetalnick[1], Jong-Hyeok Yoon[2], Wei Wu[3], Carlos Tokunaga[3], Vivek De[3], Arijit Raychowdhury[1]

[1]Georgia Tech, Atlanta, USA. [2]DGIST, Daegu, Republic of Korea. [3]Intel, Hillsboro, USA.

Resistive RAM (RRAM) is a promising candidate for compute in-memory (CIM) applications owing to its natural multiply-and-accumulate structure in a 1T-1R bitcell, high-bit density, non-volatility, and voltage and process compatibility. These properties seek to advance applications such as AI with higher throughput and bit-density. However, due to process, temperature, and write-to-write variations the resistive state of each RRAM undergoes both spatial and temporal variations. Significant effort has been made to reduce the impact of device variation using iterative write verify (IWV) or training-aware approaches [1]. Unfortunately, traditional ECC is not compatible with CIM when multiple cells are read simultaneously on the same bitline. To address this issue at the circuit level, this paper presents a 64Kb RRAM macro in 40nm CMOS supporting SECDED (single error correction, double error detection) scheme compatible with CIM for any number of parallel row accesses. Compared to prior work, our results indicate that CIM-SECDED (1) improves bit error rate (BER) by up to 69.2× for compute in-memory (2) relaxes the constraints on resistance variations and directly lowers IWV and write voltages. As a result, when applied to AI workloads we achieve (1) 24.4% (29.9%) accuracy improvement on the CIFAR10 (ImageNet) dataset (2) and consequently, improved endurance though lowering write voltage requirements [2].

Fig 1. shows the proposed CIM RRAM macro supporting 8-bit weight networks using 8 adjacent 1-bit RRAM cells. The details of peripheral design of the macro have been discussed in [4]. Here we present the circuit and architecture that enables error detection and correction logic for CIM-SECDED. The BL during RD is selected by the 8:1 BL/SL MUX and connected with a 4-bit ADC-based readout circuit. Integrated and programmable IWV is used to obtain a high ratio between the HRS and LRS states and low device variation when reading up to 8 wordlines at a time (Fig. 1). However, the IWV policy must be considered carefully as high write voltage and successive writes lowers endurance. The CIM-SECDED detection and correction technique uses hamming codes just as traditional SECDED. CIM-SECDED is implemented using digital logic and operates on the output from the ADCs. After performing CIM-SECDED, shift and add logic are applied for implementation of vector-matrix multiplication (VMM).

Fig. 2. shows single cell read and multi-row (8 row) read data collected from the array. LRS and HRS distributions are collected after writing the devices with 3 different write voltages. While higher write voltages yield tighter distributions and higher resistance ratio, it lowers device endurance [2]. Despite achieving a tight resistance window, multi-row read significantly increases the chance of error due to accumulated variation from several cells. To quantify this error rate, we randomly program the cells such that a uniform distribution of values is achieved (0-8 LRS) and perform multi-row read. Fig. 2. shows this result in the form of a confusion matrix where the expected ADC output code is on y-axis and the actual ADC output code is on the x-axis. Each bin shows the percent of actual ADC output codes were obtained for the expected ADC output code. When the number of LRS cells is low (<4) the result is always correct for the experiment's sample size (8192 total). When more LRS cells are read, errors occur with increasing frequency. However, we note that errors are always constrained to ±1 errors (i.e., |*measured – expected ADC code*| <= 1). This observation occurs because resistance variations follow a normal distribution and thus neighboring ADC codes (±1) are exponentially more likely. This property has special implications for both error correction and detection. Like traditional SECDED, a ±1 error can be detected and localized using a hamming code.

The proposed encoding, decoding, and correction steps for CIM-SECDED are shown in Fig. 3 along with 1 encoding example and 2 decoding examples. The example shown uses a (4, 3) hamming code for each row and requires an additional 2 bits for double error detection and sign detection (±1). Compared to traditional SECDED, CIM-SECDED requires only 1 additional bit which enables sign detection. For example, to protect 32 bits of data SECDED requires 7 parity bits and CIM-SECDED requires 8 bits (7+1). The encoding for CIM-SECDED is shown in the top left, where the localization (parity) and sign bits are computed as a function of the 4-bit data. The 3 parity bits are computed the same way as traditional SECDED. The first sign bit (S_0) also serves as double error detection. The second sign bit (S_1) is computed as mod-4 and shifting by 1. Combined, the two bits serve as a checksum for the sum of the data and parity bits. This enables sign detection and correction. Although mod-3 would enable simpler sign detection, mod-4 is used because it preserves double error detection. One example problem is shown in the top right, where we compute the parity and sign bits and show the mapping to the RRAM CIM macro. The error is localized and double errors are checked using the same method as traditional SECDED. For localization we need only consider the LSB of each ADC readout (i.e. 0→0; 1→1; 2→0; 3→1) because errors are constrained to ±1. Then the sign of the error is computed using the computed checksum and the checksum encoded in the sign bits. The sign is obtained using a LUT containing 16 entries for the various possible outcomes shown in the bottom left. Lastly, the address is decoded and the error (±1) is applied to the victim ADC output.

Fig 4. shows the CIM-SECDED flow, starting from encoding the weights with parity bits to decoding and correcting checksums in the on-chip macro. For most CIM applications encoding can be done off-chip, and only decoding is required on-chip. Our complete (32, 8) CIM-SECDED design is shown in the bottom as a fully digital implementation. Like standard SECDED, an XOR tree is used for each of the localization bits. To compute the calculated mod-4 checksum a single 2-bit no-carry adder tree is used. Using the output of localization and sign detection, double error detection and the sign lookup are performed. And lastly a state machine interprets the address, DED flag, and sign to update the ADC output codes.

In Fig. 5 we quantify BER and its impact on template AI applications. We apply our data in Fig. 2 to standard DNN benchmarks. The bar charts reveal the error breakdown for each layer in a prototypical VGG11 network. For 7.1% LRS variation (V_{BL}=1.7 in Fig. 2), we observe up to 4 errors per parallel ADC readout (40=32+8 codes). For 3.7% LRS variation (V_{BL}=1.9 in Fig. 2) we observe up to 2 errors. The significant majority of the CIM operations result in either 0 error or 1 error. When using CIM-SECDED the operations resulting in only a single error are successfully corrected, thus reducing the BER 69.2× over prior work. This translates to accuracy improvement in the CNN classification tasks shown in the table below. CIM-SECDED reduces error for all test conditions. Specifically, it yields 0% (0.4%) accuracy loss on ImageNet compared to an unacceptable 3.9% (16.7%) accuracy loss without it. Furthermore, it offers very low error on CIFAR10 and thus enables lower WR voltage and tolerates higher resistance variation. By reducing write voltage, we can both improve endurance [2] and reduce power by up to 24.9%.

Fig 6. shows a breakdown of the area and power in the RRAM CIM macro. The CIM-SECDED logic accounts for only 3.5% of the total area and 15% of the total power. However, 20% (8/40) of the RRAM cells and read circuit must be allocated as parity bits and account for an additional 13.3% total area and 16.4% total power consumed. While CIM-SECDED incurs overhead, we still find it achieves competitive efficiency of 43.1 TOPS/W at 100 MHz. A comparison with the state-of-the-art CIM architectures [3-8] illustrates competitive metrics while addressing key technological challenges. The die-shot and the chip-characteristics are shown in Fig. 7.

Acknowledgement: This work was funded by the Semiconductor Research Corporation under the Center for Brain-Inspired Computing (C-BRIC) under Grant 2777.005 and 2777.006, The authors would also like to thank Win-San Khwa, Yu-Der Chih, and Meng-Fan Chang at TSMC for technical discussions and chip fabrication support.

References:

[1] Crafton, Brian, et al. "Merged Logic and Memory Fabrics for Accelerating Machine Learning Workloads." Design & Test (2020).

[2] Nail, C., et al. "Understanding RRAM endurance, retention and window margin trade-off using experimental results and simulations." 2016 IEDM.

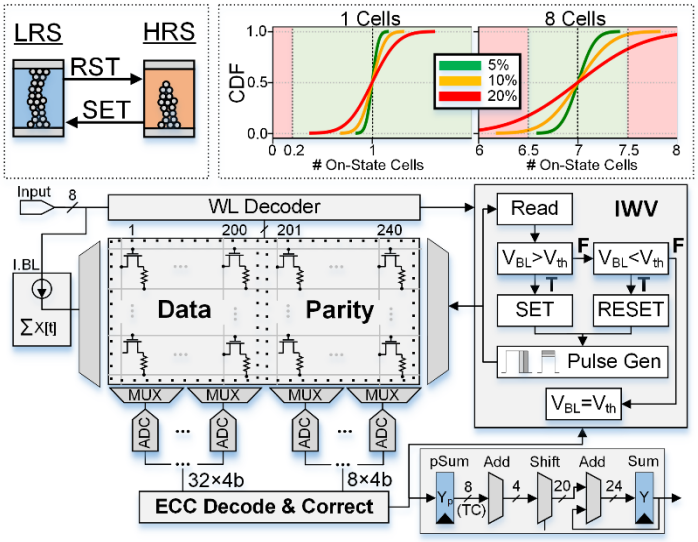


Fig. 1. Overall architecture of the compute in-memory RRAM macro with CIM-SECDED & iterative write verify control.

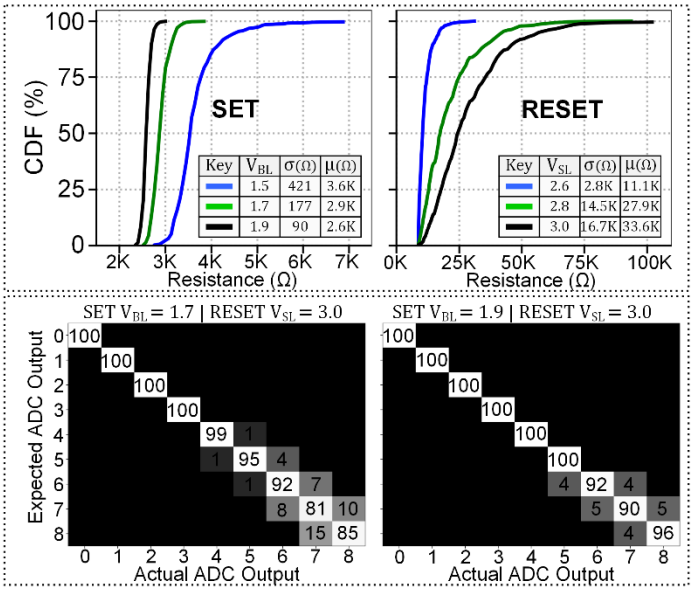


Fig. 2. Measured (A) LRS & HRS distributions (B) CIM confusion matrix.

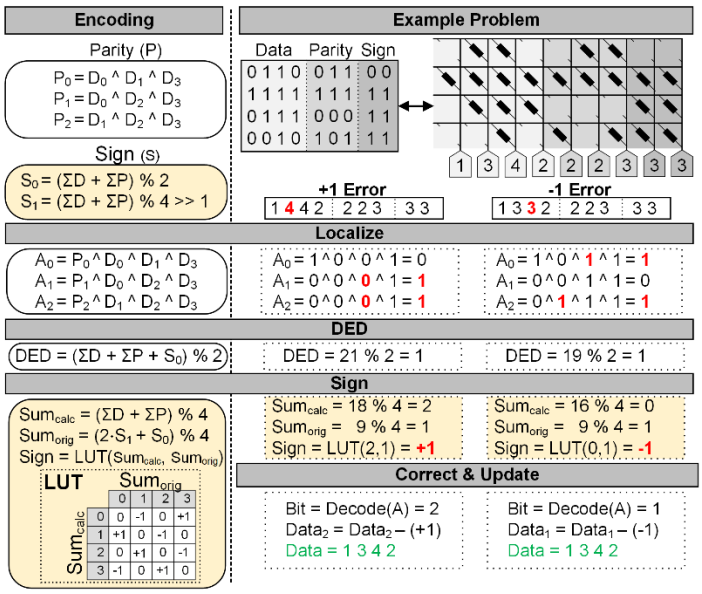


Fig. 3. CIM-SECDED encoding and decoding with examples.

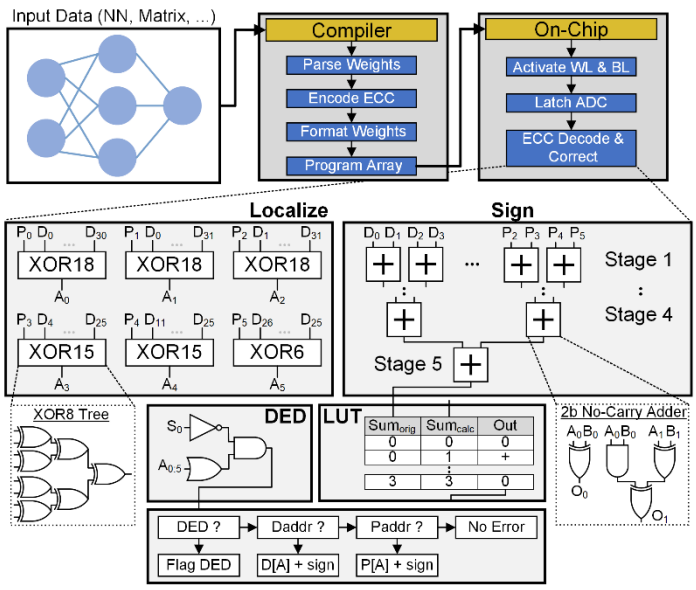
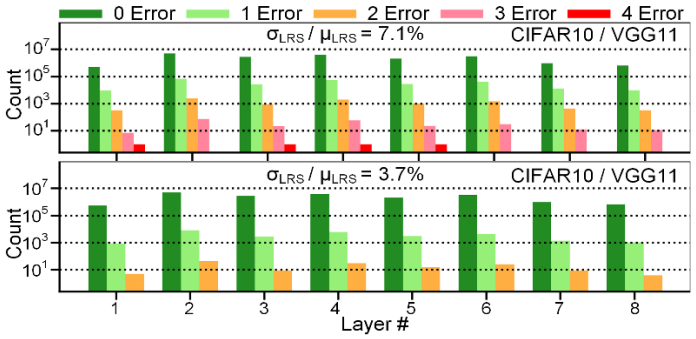
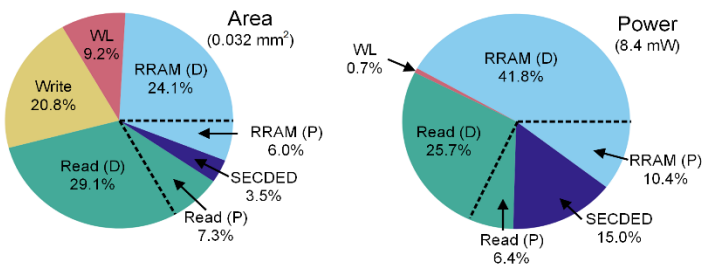


Fig. 4. Software flow and digital decoding and correction logic.



Task		Variation (%)		Accuracy Loss (%)	
Dataset	Network	WR Voltage	σ_{LRS} / μ_{LRS}	No ECC	ECC
CIFAR10	VGG11	V_{BL} = 1.9	3.7%	0.1%	0%
		V_{BL} = 1.7	7.1%	4.7%	0%
		V_{BL} = 1.5	11%	34.2%	9.8%
ImageNet	ResNet18	V_{BL} = 1.9	3.7%	3.9%	0%
		V_{BL} = 1.7	7.1%	16.7%	0.4%
		V_{BL} = 1.5	11%	63.9%	34.0%

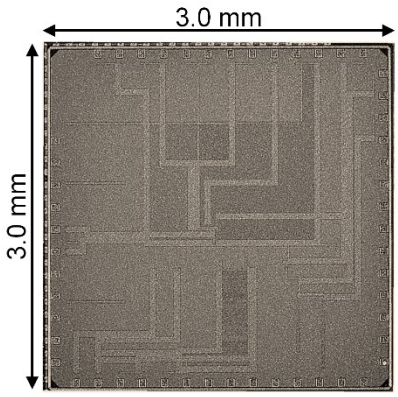
Fig. 5. (A) Bit error rate (BER) and (B) accuracy loss on CIFAR10 & ImageNet with CIM-SECDED using LRS variation from Fig 2.



Comparison Table NR – Not Reported * – Computed using data

Reference	ISSCC 2018 [5]	ISSCC 2019 [6]	ISSCC 2020 [7]	ISSCC 2020 [8]	ISSCC 2021 [3]	CICC 2021 [4]	This Work
Technology	65nm	55nm	22nm	130nm	40nm	40nm	40nm
Memory	RRAM	RRAM	RRAM	RRAM	RRAM	RRAM	RRAM
Supply	1.0V	1.0V	0.9V	1.8V	0.9V	0.9V	0.9V
Sense Mode	Current	Current	Current	I&F	Voltage	Voltage	Voltage
Write Verif	N	N	N	N	Y	Y	Y
TOP / W	19.2	53.17	121.38	148	56.67	118.44	43.1
ECC	N	N	N	N	N	N	Y
BER	NR	NR	NR	NR	*3.6×10^{-5}	*9.0×10^{-4}	5.2×10^{-7}

Fig. 6. (A) Area and power breakdown by module (B) Comparison with state-of-the-art compute in-memory macros.



Technology	40nm RRAM & CMOS
Frequency	100 MHz
Digital VDD	0.9V
Analog VDD	0.8V
I/O VDD	3.3V
TOP / W	43.1
Package	QFN48

Fig. 7. Micrograph of the test chip and summary of performance

Additional References:

[3] Yoon, Jong-Hyeok, et al. "A 40nm 100Kb 118.44 TOPS/W Ternary-weight Computein-Memory RRAM Macro with Voltage-sensing Read and Write Verification for reliable multi-bit RRAM operation." 2021 CICC.

[4] Yoon, Jong-Hyeok, et al. "29.1 A 40nm 64Kb 56.67 TOPS/W Read-Disturb-Tolerant Compute-in-Memory/Digital RRAM Macro with Active-Feedback-Based Read and In-Situ Write Verification." 2021 ISSCC.

[5] Chen, Wei-Hao, et al. "A 65nm 1Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors." 2018 ISSCC.

[6] Xue, Cheng-Xin, et al. "24.1 a 1Mb multibit ReRAM computing-in-memory macro with 14.6 ns parallel MAC computing time for CNN based AI edge processors." 2019 ISSCC.

[7] Xue, Cheng-Xin, et al. "15.4 A 22nm 2Mb ReRAM Compute-in-Memory Macro with 121-28TOPS/W for Multibit MAC Computing for Tiny AI Edge Devices." 2020 ISSCC.

[8] Wan, Weier, et al. "33.1 a 74 tmacs/w cmos-rram neurosynaptic core with dynamically reconfigurable dataflow and in-situ transposable weights for probabilistic graphical models." 2020 ISSCC.