A 65nm 8-bit All-Digital Stochastic-Compute-In-Memory Deep Learning Processor

Jiyue Yang, Tianmu Li, Wojciech Romaszkan, Puneet Gupta, Sudhakar Pamarti.

University of California, Los Angeles, CA.

High compute density improves the data reuse and is the key to reducing off-chip memory access and achieving high energy efficiency in ML accelerators. Compute-in-Memory (CIM) promises high compute density but requires ADCs, DACs that add to the macro's energy and area [1][2] limiting its compute density. Besides, CIM's analog compute is sensitive to process variability and mismatches. The transistor nonlinearity also significantly degrades the compute accuracy. Stochastic Computing (SC), which represents numbers as probability of 1s in random binary streams, is a digital-compute scheme that uses tiny MACs and offers high compute density without ADC/DAC (Fig. 1). Simulations show 4x reduction of memory access compared to 8b fixed-point digital accelerators due to the massive parallelism it can achieve using the tiny digital logic gates (AND/OR). However, SC suffers from high cost of binary to stochastic number conversion and compute error.

In this work, we combine SC and CIM to achieve much higher compute density and solve their respective problems. The proposed Stochastic Compute-In-Memory (SCIM) machine learning processor embeds SC MAC logic onto the SRAM's bitline and only makes 1-bit decisions that don't need DACs/ADCs. By storing the pre-generated random bitstreams: Stochastic Numbers (SNs) in the memory, stochastic number generation (SNG) cost is amortized over reuse by 32x. The SNG uses maximum-length LFSR and accurately converts binary numbers. A computation skipping technique is implemented that reduces the required bitstream length by 4x. A programmable SCIM-based CNN processor is demonstrated in 65nm and achieves 6x higher macro density and system energy efficiency of 7.96TOPS/W and macro energy efficiency of 20 TOPS/W for 8-b compute. A training algorithm that accounts for the specifics of SC is employed and accuracy comparable to 8b fixed point networks is demonstrated for MNIST and CIFAR10 datasets.

Previous in-situ SC accelerator [3] proposed to embed SC's multiplication in the memory and perform accumulation externally, which significantly limits the macro's throughput. The proposed SCIM accelerator uses 1-bit OR-based accumulation, which is 0 when all the inputs are 0 and is 1 when at least one input is 1. The OR-based accumulation approximates the summation. For example, for two inputs A and B, the output becomes A+B-AxB, Fig.1. However, the error term -AxB can be accounted for during the gradient descent of the neural network's backpropagation to update the weights. Comparable neural network accuracy to the fixed-point implementation is achieved [4]. The OR-based accumulation enables the SC MAC to be efficiently realized using the bitline of a memory array (Fig.1). Each SRAM bit cell adds two nMOS transistors that AND the bitcell (storing inputs) and the compute wordline (sending weights), while the shared compute bitline realizes a wired-OR operation between N cells.

Given their structural similarities, we compare SCIM with CIM. Since the SC MAC produces only 1-bit at a time, a simple SA is sufficient. The result is a DAC/ADC-free, digital architecture that can follow a traditional digital design flow. Note that each 1b SCIM MAC needs to be evaluated 2^N times to achieve N-bit computation precisions whereas CIM makes a single N-bit decision. A computation skipping technique is employed in conjunction with average pooling to cut down the SC's stream length by 4x [4]. Consider an average pooling of a 2x2 window. It is realized by a 4:1 mux that selects each input for ¼ of the output sequence (Fig.2). With computation skipping, the unselected input streams are not computed, resulting in 4x fewer SC computations. The reduced computation latency along with the low energy SC MAC unit embedded in the memory results in 10x energy reduction of the 8-bit MAC compared to CIM (Fig.2).

A bottleneck with conventional SC is the large energy cost of the conversion from binary to SNs: an SNG consumes 25x more energy than a SC MAC unit (Fig. 2). The proposed SCIM amortizes activation SNG costs by storing pre-generated activation SNs in the SRAM. They are reused between multiple sliding windows of the filter. It amortizes weight SNG costs by sharing the weight SNs between 32 different MAC rows. Overall SNG cost is reduced by 32x. Note that alternatively weight SNs could be stored in the SRAM and activation SNs streamed in.

The proposed SCIM CNN accelerator (Fig.3) is highly programmable and provides a large on-chip buffer to store all the weights. It has an on-chip FSM that supports different layer types, sizes and batch norm/pool options. It employs 32 SCIM cores in parallelly. Each SCIM core has a 32x256 CIM macro, SNGs and a near memory compute unit. The activations are stored inside the macro since they only have positive values and require half the storage. Each cell supports two MACs, with positive and negative weights on CPWLP/N. A simple inverter-based sense amp detects ON or OFF state and converts to a 1b result.

Note that storing SNs inside the memory requires generating SNs in parallel. This is achieved by duplicating the SNGs in each SCIM core but programming the LFSRs to start at different phases, Fig. 3. All SCIM cores receive the same binary numbers and convert them to full SNs in one cycle. MACs' output SNs are generated in one cycle. The convolution layer uses the row-serial data flow (Fig.4). Each input row is stored in separate rows and each weight row is applied to SCIM macro serially. After compute, the MAC output is shifted and accumulated to produce the result of a 2D convolution. Those results are converted back to fix-point domain by parallel counters. Two pipeline stages are followed to perform average pooling, batchnorm and ReLU, but they can be bypassed. The training process of the SC network is also shown in Fig. 4. It accounts for the OR-based accumulation and the randomness of the LFSR to improve the accuracy.

The test chip is fabricated in TSMC 65nm GP technology and occupies 9.4mm^2. SCIMA chip has 520Kb SC MAC embedded in the memory, achieving 6x higher macro density than state-of-art CIMs due to simple operation without the ADC/DAC. Fig.5 summarizes accelerator's network performance measured at 0.8V supply. The clock frequency is 4MHz and can be much higher given a stronger on-chip power delivery network. A 4- and 5-layer NN for CIFAR10 and MNIST are trained and tested on multiple chips. All parameters are loaded on chip once and no off-chip memory accesses are made during operations. The measured CIFAR-10 and MNIST accuracy are comparable to the same network trained in 8b fixed point. The SC matrix vector multiplier (MVM) has 100% utilization and achieves the peak energy efficiency of 5.75TOPS/W. Activations and weights have 8-bit precision. A MAC is defined as 2 operations and accounts for the processing the full 256b SNs. Fig.5 also shows the die photo and the area breakdown of the chip. The energy efficiency is measured over different voltage and frequency conditions. The chip performs robustly over 0.7-1.05V supply voltage with the best system efficiency of 7.96TOPS/W and macro efficiency of 20 TOPS/W at 0.7V and 3.2MHz. Fig.6 shows a performance comparison with state-of-the-art in CIM. For a fair comparison, prior art's reported efficiency numbers are scaled to a 65nm node and to equivalent 8b precision. The proposed SCIM achieves 2.5x higher peak energy efficiency and it is the only work that achieves robust operation under a wide range of supply voltages.

References:

[1] H. Jia et al., "A Programmable Neural-Network Inference Accelerator Based on Scalable In-Memory Computing," ISSCC, 2021.

[2] J. Yue et al., "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," ISSCC, 2021.

[3] S. Li et al., "SCOPE: A Stochastic Computing Engine for DRAM-Based In-Situ Accelerator" MICRO, 2018.

[4] W. Romaszkan, T. Li, T. Melton, S. Pamarti and P. Gupta, "ACOUSTIC: Accelerating Convolutional Neural Networks through Or-Unipolar Skipped Stochastic Computing," DATE, 2020.

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

[6] Y. Chen et al., "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," ISSCC, 2016.

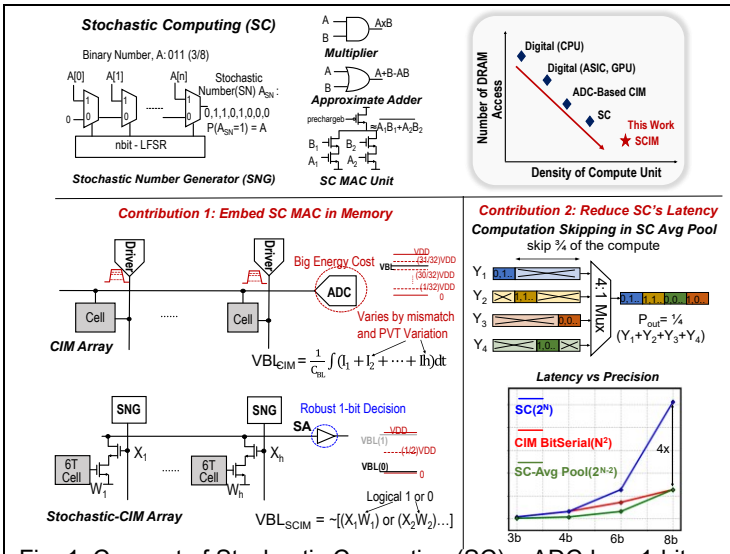

Fig. 1. Concept of Stochastic Computing (SC); . ADC-less 1-bit OR-based SC accumulation embedded in memory; Computation skipping reduce latency

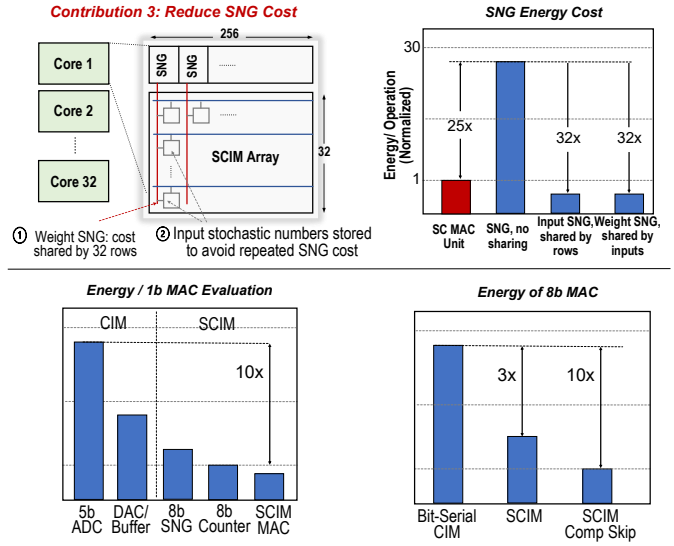

Fig. 2. SCIM reduces SNG cost; Energy comparison vs CIM.

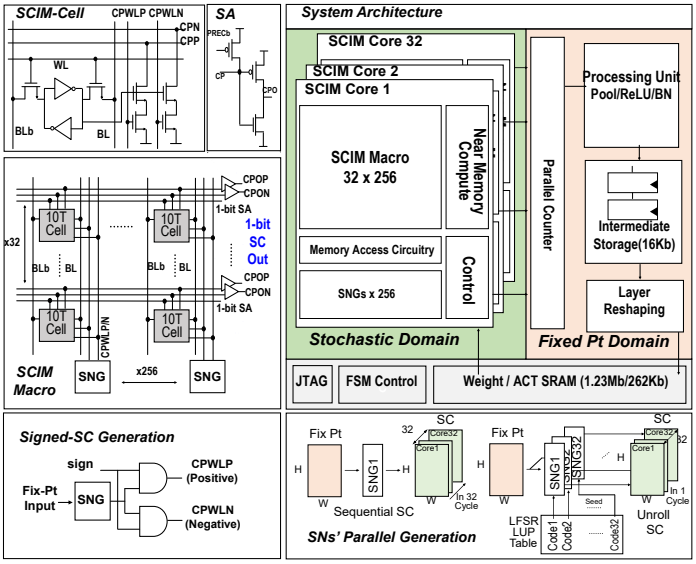

Fig. 3. System architecture of the SCIMA; Details of the SCIM macro; Illustration of unrolling SC in space.

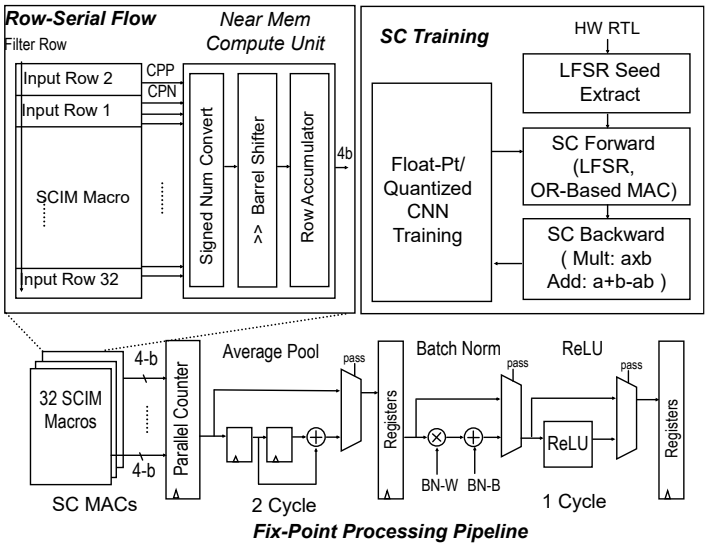

Fig. 4. Row-serial dataflow and near memory compute unit; SC training procedures; Fixed point processing pipeline.

Measurement Result Summary (MNIST, CIFAR10,MVM)

Dataset	MNIST	CIFAR-10	MVM
Network Topology	1. Conv 5x5x1x6, Pool 2x2 2. Conv 5x5x6x16, Pool 2x2 3. FC 5x5x16x120 4. FC 120x64 5. FC 64 x10	1. Conv 5x5x3x32, Pool 2x2 2. Conv 5x5x32x32, Pool 2x2 3. Conv 5x5x32x64, Pool 2x2 4. FC 4x4x64x10	Matrix: 32x256 Vector: 1x256
Bit Precision	8bit Weight, 8bit Act		
Training Accuracy	99.2%	75%	N/A
Measured Accuracy	99.1%	73.5% (Top-1), 98.4%(Top-5)	N/A
Peak Hardware Utilization	5.1%	31.3%	100%
VDD/CLK	0.8V/4Mhz		
Batch Size	1		N/A
Energy/Classification	18.3μJ	35.6μJ	N/A
Peak System Energy Efficiency*	0.45TOPS/W	1.84TOPS/W	5.75TOPS/W
Latency	5.85msec/frame	11.5msec/frame	1usec

* 1 SC MAC is 2 Ops = Processing 256 SC bits

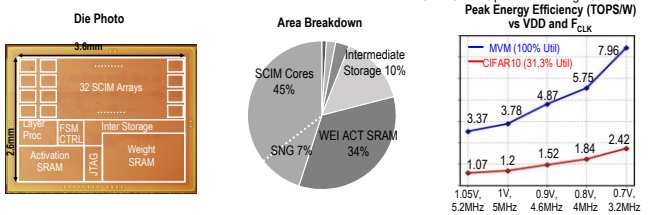

Fig. 5. Measurement result; Die Photo; Area breakdown; Energy efficiency vs Vdd and CLK.

	Chen, ISSCC 16[6]	Dong, ISSCC 20 [5]	Jia, ISSCC 21[1]	Yue, ISSCC 21 [2]	This Work
Technology	Digital, 65nm	CIM, 7nm	CIM, 16nm	CIM-SRAM, 65nm	SCIM, 65nm
Size	16mm^2	0.0032mm^2	25 mm^2	12 mm^2	9.4 mm^2
Voltage	0.8-1.2V	1V	0.8V	0.65V(Digital),1V(CIM)	0.7-1.05V (CIM) , 0.8V(SRAM)
On-Chip CIM Capacity	Non CIM	4Kb (CIM)	4.5Mb (CIM)	64Kb (CIM)	520Kb (CIM)
CIM Macro Density[1]	Non CIM	16.6 Kb/mm^2	22 Kb/mm^2	16 Kb/mm^2	130Kb/mm^2
Precision	16 bit	4b Fix-Pt	8b Fix-Pt	8b Fix-Pt	256b SC (8b Fix-Pt)
Peak System Energy Efficiency (TOPS/W) (Scaled to 8b)[2]	0.332 (System)	0.98 (Macro)	1.88 (System)	2.3 (System) 6 (Macro)	7.96 (System)[3] 20 (Macro)
Peak System Throughput.[2,3] (GOPS, Scale to 8b)	480	114	3K	25-790	58.8 (MVM)[3]
NN Performance (MNIST)	N/A	N/A	N/A	N/A	99.1%(T-1), 5.85msec/fr, 18.3uJ/fr
NN Performance (CIFAR10)	N/A	N/A	91.5%, 0.13msec/fr, 310.4uJ	91.56%, 6.51ms/fr	73.5%(T-1), 98.4%(T5) 11.5msec/fr, 35.6uJ/fr

[1] scaled to 65nm [2] # of OP is scaled with (bit precision) [3] Peak efficiency comparison done at 100% hardware util, corresponding to MVM's energy efficiency

Fig. 6. Comparison Table