Realizing Direct Convolution in Memory with Systolic-RAM

Jacob N. Rohan, Jaydeep P. Kulkarni

The University of Texas at Austin, Austin, TX

Abstract:

A 12.8Kbit Static Random Access Memory (SRAM) array is demonstrated in 40nm CMOS for charge-domain vector-matrix multiplication (VMM). While conventional compute-in-memory (CIM) approaches rely on the indirect convolution algorithm, the proposed Systolic-RAM performs a form of direct convolution which eliminates the need for data duplication and near-memory registers. For this purpose, bitcells feature additional read/write ports configured to move data directly from one neighboring bitcell to the next. Circuit details for implementing signed analog multiplication within the array are discussed. Quantized neural network training methods are used to effectively mitigate non-ideal analog effects and achieve test accuracy near that of a floating-point network. The 12.8Kbit VMM test chip configured for 8-bit 5x5 convolution achieves 175(113) peak(continuous) multiply-accumulate (MAC) operations per clock cycle and consumes 3.0mW at 100MHz.

Motivation:

Data movement significantly impairs power performance in von Neumann systems when large amounts of data are exchanged between computer memory and processing units (referred to as the memory wall bottleneck). CIM approaches attempt to reduce data movement energy and latency overheads by performing key computations in parallel within the memory array (Fig.1). Although adequate bit-resolution is commonly considered a leading measure of CIM performance [1], few works have elaborated on the data movement power, duplication, and restructuring required to realize convolution within CIM macros [2]. Duplication occurs since the indirect convolution method required for VMM-based accelerators uses an image-to-column (IM2COL) transformation [3,4]. This means conventional methods do not physically adopt the concept of sliding kernel (the stride) within hardware and require significant data caching at the CIM array periphery to support peak throughput. The data overhead becomes more detrimental for large kernels. For a K-by-K kernel with stride=1, each activation will belong to K^2 unique stride locations and would be duplicated in K^2 columns of the resulting IM2COL matrix. For example, convolution using a large kernel (such as 11x11) will require data to be duplicated (>100 times) within an activation stationary CIM array, significantly reducing its data density and performance. Such data duplication and data pre-conditioning should be considered as a substantial factor in determining the energy efficiency of convolution computations.

Systolic-RAM Design:

Systolic-RAM computes convolutions without data duplication by re-cycling data between neighboring SRAM bit-cells. The process consists of two alternating phases: Φ1 (data movement) and Φ2 (compute). Fig.2A illustrates how adjacent stride-regions are computed simultaneously. After computation, vertical stride is achieved in a Φ1V phase by cycling data through buffered-6T (B6T) bitcells to exchange K pixels from one stride location to the next (Fig.2B) while reusing K*(K-1) pixels from the previous computation. When a horizontal kernel translation is required (Fig.2C), Φ1H phase is used to insert new data from 8T bitcells into the B6T datapath. Fig.2D illustrates both these translations within the B6T/8T cell structure.

This digital data movement allows Systolic-RAM to perform several MAC products every clock cycle. Seven unique 5x5-pixel regions of the input image are computed simultaneously for a total of 175 operations per cycle (Ops/Cy). This corresponds to the highlighted regions in Fig.2E. Total effective Ops/Cy varies based on application. For example, configuring this design for a ResNet layer (which requires padding for input/output sizes to be identical) achieves a continuous 113 [Ops/Cy] (24025 effective operations in 155 compute cycles + 56 digital write cycles) when padded for 31x31 input/output images. In these cases, additional rows/columns can be added for increased parallelism.

Analog MAC is achieved using multiplicative digital-to-analog (MDAC) structures positioned in the back-end-of-line (BEOL) above the array. In the Φ2 (compute) phase, the kernel data is broadcast as an analog differential bit-line voltage to modulate the MDAC. The MDAC's inputs are selectively switched to either bit-line based on the digital data in the bitcells. The resulting output charge produced is proportional to the product of signed 8-bit kernel and signed 8-bit input data. Fig.3 demonstrates the DAC and large-signal ring amplifier used to drive the bit-line capacitive load, while Fig 4. depicts the layout of the capacitive MDAC structure above the array. While the differential analog datapath rejects common mode interference, parasitic capacitance in the MDAC results in significant non-linearity. To mitigate this effect, 3D parasitic extraction is performed and notches in the MOM-CAP structure are adjusted to tune input capacitances and preserve final output linearity [5,6].

Measured Results:

The measured multiplication characteristics (Fig.5a) demonstrate a 74% reduction in worst-case DAC DNL with only 45% reduction in amplitude. Least squares regression was used to extract the relative significance of each bit and demonstrate the linearity improvement (Fig.5b). Noise and nonlinearity with respect to input and weight were modeled as a differential convolution layers in Pytorch to match the characteristics in (Fig.5a). A gradient-blocking technique for quantization was used for autograd and network re-training [7]. Pretrained ResNet-18 convolutional neural network (CNN) using float32 demonstrated 91.9% test accuracy on CIFAR-10 dataset [4,8,9]. Immediately after 8-bit quantization, test accuracy was 90.3% using the calibrated MDAC and 81.6% with the uncalibrated MDAC. After retraining for 3 epochs, test accuracy recovered to 91.6% (0.3% below float32 baseline) with calibration but only 86.2% for the uncalibrated MDAC (Fig.5c).

SystolicRAM performs convolution at 14.4 bit-TOPS/W for 100MHz, 1.1V (Fig.6). We found the largest contributor of power in SystolicRAM to be the ring amplifier topology chosen since the Φ1 reset (RST) phase requires the input and output of inverter-like structures to be shorted together (Fig.4). Changing devices in this design to have a high-threshold voltage (HVT) is estimated to yield a static power reduction of 25x for digital elements (logic and bitcells) and 2x for analog components resulting in a projected FOM of 35.8 bit-TOPS/W. The proposed approach can improve data/energy-efficiency, bit-precision, and supported kernel size of VMM macros used for convolution computations.

Conclusion:

Systolic-RAM demonstrates the first in-memory direct convolution engine as an all-in-one approach to data-efficient convolution. Systolic-RAM makes good use of BEOL wiring for analog multiplication and charge sharing over the SRAM with little silicon-area overhead. This works demonstrates the importance of DAC calibration and use of state-of-the-art quantization neural network methods to recover near-ideal CNN classification performance in analog compute systems.

Acknowledgments:

Authors would like to thank TSMC for test chip fabrication.

References:

[1] T. Yang et al., "Design Considerations for DNN," IEDM 2019. [2] M. Cho, D Brand, "Memory-Efficient Convolution for DNN," PLMR 2017. [3] M. Dukhan "The Indirect Convolution Algorithm," arXiv:1907.02129 [cs], 2019 [4] A. Paszke et al., "Automatic differentiation in PyTorch," arXiv:1907.02129 [cs], 2019 [5] H. Balasubramaniam et al., "Floating Shield DAC" SCS 2009. [6] P. Harpe, "SAR ADC with Passive FIR Filter," CICC 2018. [7] C.N. Coelho, et. al. "Automatic deep heterogeneous quantization of DNN," arXiv:2006.10159 [physics.ins-det], 2020 [8] H. Phan, huyvnphan/PyTorch_CIFAR10. Zenodo, 2021. doi: 10.5281/ZENODO.4431043 [9] A. Krizhevsky, "CIFAR-10, Learning Multiple Layers of Features," 2009 [10] B. Hershberg et al., "Ring Amplifiers," ISSCC 2012 [11] H. Balasubramaniam et al., "Floating Shield DAC" SCS 2009.

978-1-6654-4350-0/21/$31.00 ©2021 IEEE

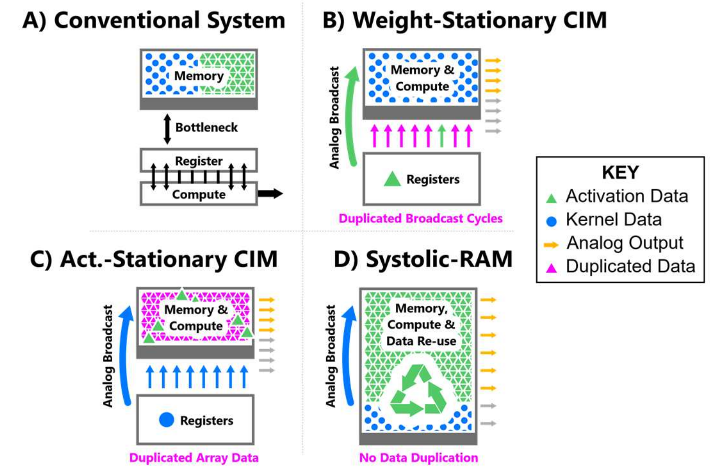


Fig. 1: **(A)** Conventional systems suffer from von Neumann bottleneck. **(B-C)** Weight/activation-stationary CIM requires duplicating or buffering of data. **(D)** Systolic-RAM requires minimal near-memory circuitry and eliminates need for data duplication.

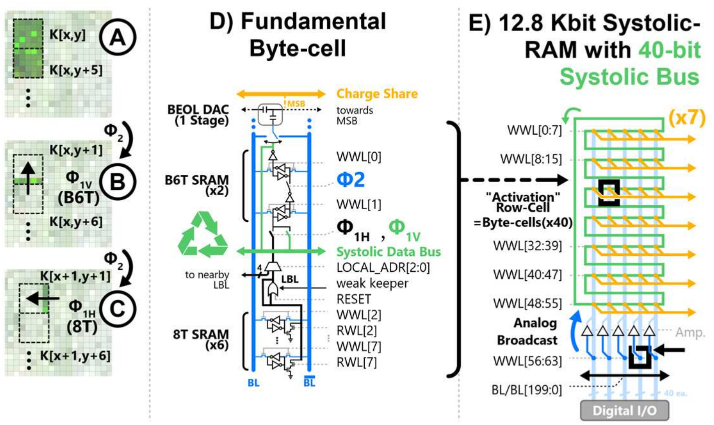


Fig. 2: **(A-C)** Vertical and horizontal translation of the kernel corresponds to **(D)** data movement between neighboring bitcells. Kernel data is broadcast along bit-lines (blue) and modulate the BEOL MDAC to perform multiplication with data in the bitcells. **(E)** The resulting charge output represents the multiplicative product and is accumulated horizontally along 7 charge share lines.

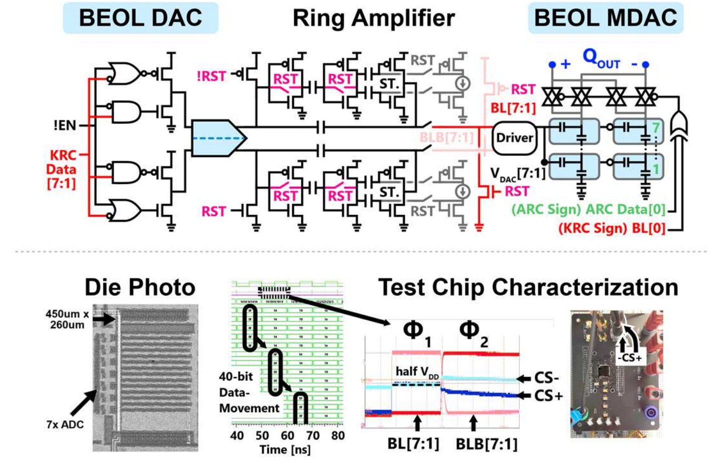


Fig. 3.: (TOP) Schematic for large-signal ring amplifier for broadcast of analog data on bit-lines [10]. Output stage is biased using current mirrors to mitigate PVT effects. (BOTTOM) Test chip showing in-memory data movement and differential multiplication waveforms.

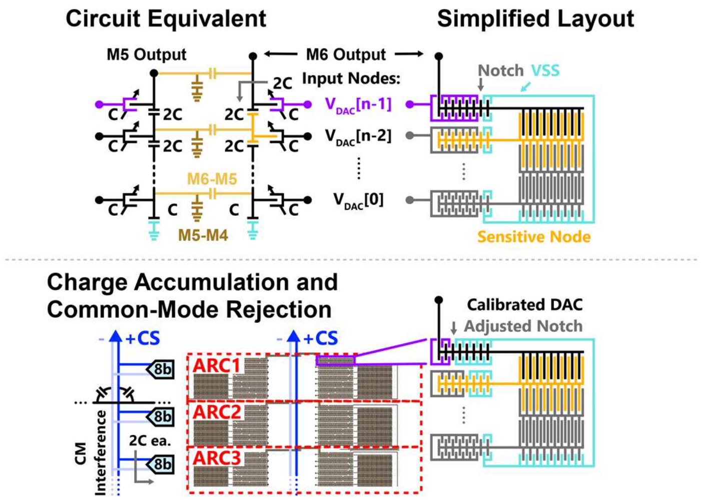


Fig. 4. (TOP) Circuit equivalent for C2C DAC with floating voltage shield [11]. (Bottom) Notches are adjusted for improved linearity. Adjacent DACs share a single output for charge accumulation.

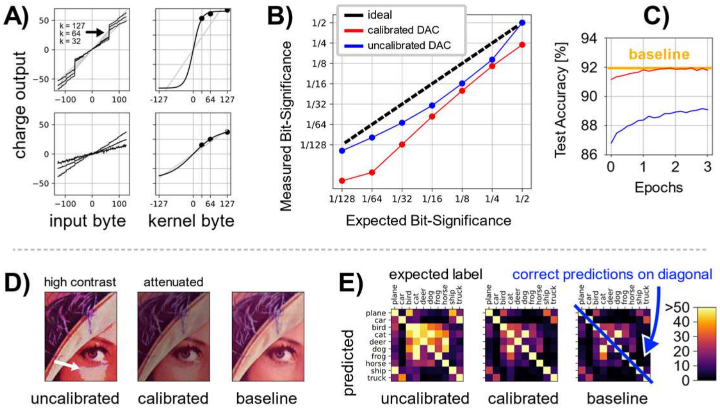


Fig. 5. **(A)** Measured multiplication characteristics and **(B)** corresponding bit-significance for calibrated and uncalibrated DACs demonstrate significant linearity improvement. **(C)** Re-training curves demonstrate good recovery to baseline performance after 3 epochs with calibration. **(D)** Visualized effect of convolution on test image. **(E)** Effect of non-linearity on CIFAR-10 predictions prior to retraining, based on 10,000 test images.

Merit	Units	Projected	This Work
Tech.	[nm] (HVT)	40 (HVT)	40 (LVT)
Supply	[V]	1.1	1.1
SRAM	[Kbyte]	1.56	1.56
SRAM Bit-cell		8T, B6T	8T, B6T
Method		SystolicRAM	SystolicRAM
Weight Precision	[bits]	8	8
Area	[mm^2]	0.12	0.12
Power	[mW]	1.2	3.0
Throughput	[GOPS]	5.4	5.4
Area Efficiency	[GOPS/mm^2]	44.8	44.8
Power Efficiency	[TOPS/Watt]	4.5	1.8
FOM	**[bit-TOPS/Watt]**	**35.8**	**14.4**

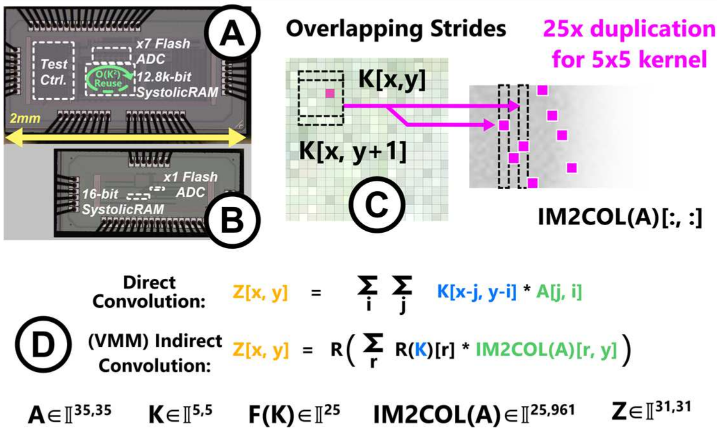
Static Power		
RA (RESET)	**1.37**	**[mW]**
↳ Static Leakage	=54.8	[uW/RA]
↳ Multiplier	* 25	# RA
Logic + Bit-cells	**0.66**	**[mW]**
↳ Static Leakage	51.95	[nW / bit]
↳ Multiplier	* 12800	[bit]
ADC (4bit Flash)	**0.84**	**[mW]**
↳ Static Leakage	119.9	[uW/ADC]
↳ Multiplier	* 7	#ADC

Energy Per Operation		
φ1V Data Movement	**5.2**	**[fJ/bit]** (*80%)
φ1H Data Movement	**219.8**	**[fJ/bit]** (*20%)
↳ Address decode	= 35.2	
↳ LBL Reset	+ 90.8	
↳ MUX	+ 88.6	
↳ Latching SRAM	+ 5.2	
φ2 Data Movement	**4.9**	**[fJ/bit]** (*100%)

* Activity Factor

Dynamic Power		
Data Movement	**74.2**	**[uW@100MHz]**
↳ Dynamic energy	=53.0	[fJ/bit] (avg.)
↳ Multiplier	*1400	[bit/cycle]
Analog Broadcast	**0.7**	**[uW@100MHz]**
↳ Dynamic energy	=28.0	[nW/RA/Cycle]
↳ Multiplier	* 25	# RingAmp (RA)
ADC Power	**2.2**	**[uW@100MHz]**
↳ Dynamic energy	=312.0	[fJ/Cycle/ADC]
↳ Multiplier	* 7	#ADC

Fig. 6. Detailed energy/performance breakdown considers figure of merit (FOM) as bit-resolution times terra-operations per second per watt (TOPS/Watt). Majority of power is consumed by short-circuit current in the ring amplifier (RA) [10].



Direct Convolution:

$$Z[x, y] = \sum_{i} \sum_{j} K[x-j, y-i] * A[j, i]$$

(VMM) Indirect Convolution:

$$Z[x, y] = R\left(\sum_{r} R(K)[r] * IM2COL(A)[r, y] \right)$$

$A \in \mathbb{I}^{35,35}$ $K \in \mathbb{I}^{5,5}$ $F(K) \in \mathbb{I}^{25}$ $IM2COL(A) \in \mathbb{I}^{25,961}$ $Z \in \mathbb{I}^{31,31}$

Fig. 7.: **(A)** 12.8k-bit test chip and **(B)** separate 16-bit test structure. **(C)** Demonstration of how a single pixel is duplicated in the IM2COL matrix and **(D)** corresponding equations for direct and indirect convolution. Matrix dimensions relevant to this work are provided. Reference 4 provides detailed description for dimensionality and memory impact of IM2COL; see “torch.nn.Unfold”.