Compute-in-Memory with Emerging Nonvolatile-Memories: Challenges and Prospects

Shimeng Yu, Xiaoyu Sun, Xiaochen Peng, Shanshi Huang
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332
Email: shimeng.yu@ece.gatech.edu

***Abstract*—This invited paper surveys the recent progresses of compute-in-memory (CIM) prototype chip designs with emerging nonvolatile memories (eNVMs) such as resistive random access memory (RRAM) technology. 8kb to 4Mb CIM mixed-signal macros (with analog computation within the memory array) have been demonstrated by academia and industry, showing promising energy efficiency and throughput for machine learning inference acceleration. However, grand challenges exist for large-scale system design including the following: 1) substantial analog-to-digital (ADC) overhead; 2) scalability to advanced logic node limited by high write voltage of eNVMs; 3) process variations (e.g. ADC offset) that degrade the inference accuracy. Mitigation strategies and possible future research directions are discussed.**

Keywords—in-memory computing; hardware accelerator; non-volatile memory; deep learning

I. Introduction

Deep learning is remarkably powerful in a variety of intelligent information processing applications such as image and speech recognition. Though GPU has been the mainstream platform to accelerate the deep learning at the cloud, there are growing interests to develop application-specific integrated-circuit (ASIC) chips for further improving the energy-efficiency for deep learning workloads. Digital multiply-and-accumulate (MAC) arrays are generally employed for ASIC solutions for deep learning [1]. Data flow is often optimized to increase the data reuse on-chip. Nevertheless, most weights and inputs/outputs are moved across MAC arrays and from global buffers. Therefore, it is more attractive to embed the MAC computation into the memory array itself, namely *compute-in-memory (CIM)* [2], to minimize the data transfer. In CIM, the vector-matrix multiplication is executed in parallel (with analog computation) where the input vectors activate multiple rows. The dot-product is obtained as the multiplication of input voltage and cell conductance, and the partial sum is added up by the column current. Analog-to-digital converter (ADC) at edge of the array generally converts the partial sum to binary bits for digital processing (e.g. shift-and-add, activation, and pooling).

To implement CIM, mature SRAM technologies (possibly with modified bit cell) have been proposed [3-5]. However, SRAM is inherently volatile, and consumes significant standby leakage power, especially for the dynamic power gating often used in the edge devices. In this sense, emerging non-volatile memory (eNVM) technologies [6] are better suited for the area/power constraint platforms, as they could be turned on and off instantly without losing the stored weights. eNVMs of interests here include resistive random access memory (RRAM), phase change memory (PCM), spin-transfer-torque magnetic random access memory (STT-MRAM) and ferroelectric field effect transistor (FeFET). In the recent years, industry has heavily invested in eNVM technologies with even commercial fabrication processes available, e.g. TSMC's 40nm RRAM [7] and Intel's 22nm RRAM [8], TSMC's 40nm PCM [9], Intel's 22nm STT-MRAM [10] and Samsung's 28 nm STT-MRAM [11], while doped HfO_2 based FeFET technology is also emerging, e.g. Globalfoundries' FeFET at 22nm [12].

Capitalizing on these progresses, eNVM based CIM designs have also become viable. In this paper, we will first have a survey of the prototype chips that monolithically integrate eNVMs with CMOS periphery for deep learning. Then, we will discuss the critical challenges that these designs may face with and possible mitigation strategies and future research needs.

II. Survey of CIM Prototypes with eNVMs

We survey the eNVM based CIM prototype chips in the past few years in Table 1. In 2015, IBM pioneered in a PCM design [13] with a software-hardware co-evaluation approach - the weights are read-out (row-by-row) from the PCM array with post-processing of accumulation and activation in software. The design is capable of *in-situ* training, but the accuracy for a toy-model - MNIST, is rather limited due to the non-ideal effects of the devices such as asymmetry and variability in the weight update. Since then, there are more reported designs [14-16] using RRAM with more functionalities built on-chip, e.g. ADC periphery and parallel read-out to realize nature of CIM. The recent macro by NTHU [17] and ASU/GaTech [18] presented state-of-the-art designs. NTHU's design employs single-level-cell but groups multiple cells to represent higher weight precision, however, the parallelism is limited as only 9 rows are turned on at one time. ASU/GaTech's design enables a fully parallel operation by turning on all the rows simultaneously.

It is realized that *in-situ* training with eNVMs is still premature due to the asymmetric weight update [19] and relatively large write latency/energy of eNVMs (compared to SRAM), thus the recent demonstrations focused on inference engine with relatively low bit precision (1 to 3 bit per cell), aiming at the edge computing applications. The latest progresses of the eVNM based CIM designs have scaled the technology node to 40nm and increased the capacity of the macro to Mb level. With optimized weight mapping strategies [20], these macros are capable to process a moderate CIFAR-10 dataset with reasonable inference accuracy 80%~90% with impressive energy efficiency ~50 TOPS/W.

978-1-7281-6031-3/20/$31.00 ©2020 IEEE

TABLE I. SURVEY OF RECENT eNVM BASED CIM MACRO PROTYPE CHIPS

	Tech node	Array size	Multi-bit per cell	Parallel read-out	ADC periphery	TOPS /W	Dataset	Accuracy	Training or inference
IBM (PCM) [13]	180 nm	500 × 611	4 bit	No	No	N/A	MNIST	82.9%	Training
UMass (RRAM) [14]	2 µm	128 × 64	7 bit	Yes	No	N/A	MNIST	91.7%	Training
Umich (RRAM) [15]	180 nm	54 × 108	7 bit	Yes	Yes	0.187	N/A	N/A	Training
Panasonic (RRAM) [16]	40 nm	1Mb (4Mb total)	3 bit	Yes	Yes	66.5	MNIST	90.8%	Inference
NTHU (RRAM) [17]	55 nm	256 × 512 (1Mb total)	1 bit	Partial (9-row)	Yes	55.8	CIFAR-10	81.8%	Inference
ASU/GaTech (RRAM) [18]	90 nm	128 × 64	1 bit	Yes (fully)	Yes	61	CIFAR-10	84.5%	Inference

III. DESIGN AND TECHNOLOGICAL CHALLENGES

We take the ASU/GaTech's RRAM CIM macro [18] as an example to illustrate the general design and technological challenges for the inference engines. The die photo and layout is shown in Fig. 1. We can see that 1) the level shifter to boost the voltage domain from logic power supply to the RRAM write voltage occupies noticeable area; 2) the 3-bit ADC area is significantly larger than the RRAM array itself, and the column pitch matching is difficult even with 8:1 column MUX sharing; 3) from Table I above, the accuracy for CIFAR-10 is a few percentage less than the software baseline, mainly due to the ADC offset. These observations suggest the following three key challenges in either device engineering or circuit design.

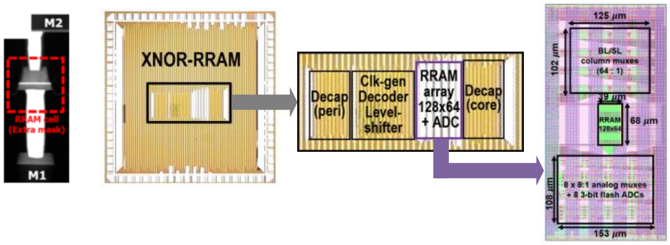

Fig. 1 ASU/GaTech RRAM based CIM macro [18].

A. Device low on-state resistance and high write voltage

Since CIM adds analog current along the column, the eNVM cells in multiple rows are in parallel. From the technological perspective, the on-state resistance (Ron) has to be engineered higher than the normal value for single-bit read-out in memory application. However, most of the eNVM technologies today exhibit Ron around 10 kΩ or below. Low Ron will contribute too large column current and cause noticeable IR drop along the interconnect wire, which may affect the analog read-out accuracy. In addition, the analog MUX at end of the column needs to be significantly sized up to avoid substantial voltage drop. Another technological challenge is the write voltage (Vw) for eNVMs like RRAM and PCM is still much higher than logic power supply. Therefore, significant area is spent on the level shifter or charge pump circuitry.

To evaluate the potential technological benefits of increasing Ron and decreasing Vw, we use the well calibrated NeuroSim framework [21] to compare the following 4 cases of RRAM at 32nm: 1) Ron=10kΩ and Vw=3.3V (as the baseline); 2) Ron=100kΩ and Vw=3.3V; 3) Ron=10kΩ and Vw=1V; 4) Ron=100kΩ and Vw=1V. The other device parameters are assumed as constant: read voltage Vr=0.5V; write pulse =75 ns. Read pulse width will be calculated by bitline RC delay plus ADC sensing. Here we consider a sub-array of CIM macro with array size 128×128, column mux sharing 8:1, and 3-bit current-mode Flash-ADC. Read operation is assumed that only 1 input vector with 50% row activity, and write operation is assumed with 50% weight updates. The area, read latency/energy, write latency/energy is shown in Table II and the normalized data is shown in Fig. 2. We could draw the following conclusions. First, increasing Ron could reduce the column MUX area by using smaller size transistors. Second, lowering Vw could further reduce the bit cell size from 36 F^2 (using I/O transistors) to 16 F^2 (using logic transistors) and elimination of the level shifter, resulting in significant area reduction. Apparently, increasing Ron could save read/write energy, while has minimal impact on read/write latency. Lowering Vw could directly reduce write energy, and indirectly reduce read latency due to the shorter bitline length by the reduced bit cell size and the removal of level shifter. Overall, Ron>100 kΩ and Vw<1V should be the targets for future device engineering.

TABLE II. NEUROSIM BENCHMARK AT 32NM NODE

Array Size = 128*128	**Ron=10kΩ**		**Ron=100kΩ**	
	Vw=3.3V	***Vw=1.0V***	***Vw=3.3V***	***Vw=1.0V***
Area (µm²)	3,613	2,263	2,543	1,452
Read Latency (ns)	22	13	21	14
Read Energy (pJ)	34	31	11	8
Write Latency (ns)	9,732	9,664	9,732	9,664
Write Energy (pJ)	617,550	56,657	62,900	5,787

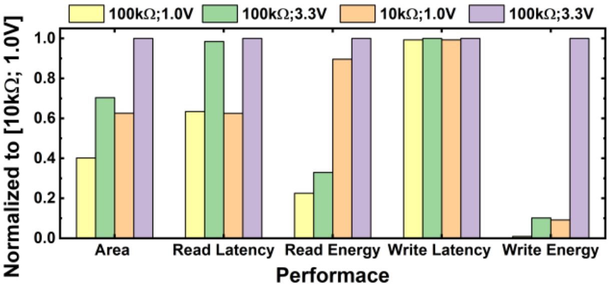

Fig. 2 Normalized sub-array level CIM performance of different RRAM technology parameters respect to Ron=10kΩ and Vw=3.3V.

B. ADC area and power bottleneck

As shown in most reported designs, ADC is still a major bottleneck for CIM. It should be noted that the ADC requirement for CIM is unique: it does not require super-high resolution or bandwidth, 3-5 bit and <1 Gbps are typically sufficient for inference engine [22]. However, there are stringent requirements

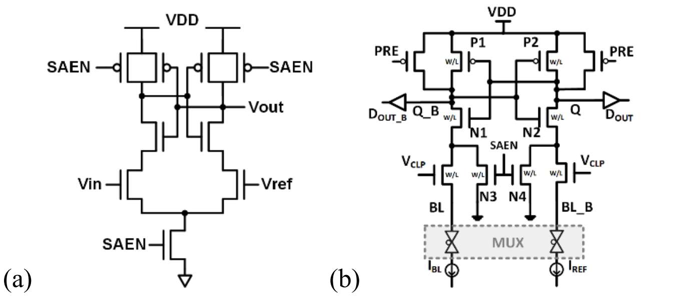

Fig. 3 (a) An example of simple voltage-mode comparator. (b) An example of simple current-mode comparator.

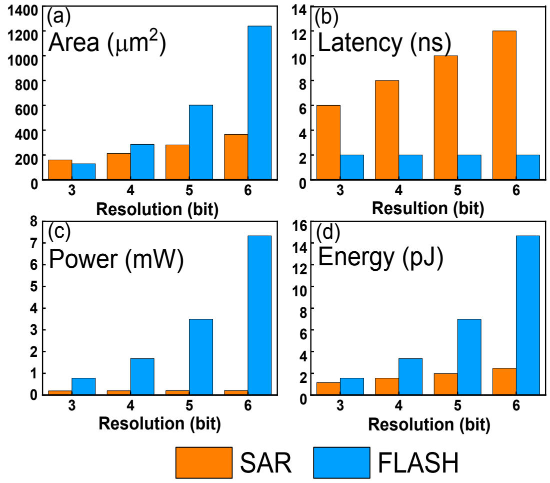

Fig. 4 The comparison between SAR-ADC and Flash-ADC on (a) area, (b) latency, (c) power, and (d) energy simulated at 40nm.

on the area of ADC, as ideally each column needs to be equipped with one ADC to maximize the parallelism of CIM. It is very difficult to achieve the column pitch matching from the layout point of view due to the relatively large size of ADC and small column pitch of eNVM array. Considering these requirements, Flash-ADC and successive approximation ADC (SAR-ADC) are of interests. ADCs could be built with voltage-mode or current-mode comparators with different references [23].

We employ a simple voltage-mode comparator as shown in Fig. 3 (a) and discuss the trade-offs between Flash-ADC and SAR-ADC based on such comparator. Fig. 4 shows the SPICE simulation results of the comparison between SAR-ADC and Flash-ADC in terms of area, latency, power, and energy for different resolutions. The simulations are performed using TSMC 40nm PDK with the following assumptions: array size is 128 × 128, Ron is 100kΩ, and on/off ratio is 100. To make a fair comparison, a thermometer-to-binary encoder is included in the Flash-ADC design while the output of SAR-ADC is naturally binary code. As shown in Fig. 4 (a) and (c), the area and power of SAR-ADC only slightly increases when the resolution goes up due to the additional overhead from the logic control module. Unsurprisingly, the area and power of Flash-ADC increases exponentially due to the exponential growth of the number of comparators deployed in the design. Nevertheless, Flash-ADC holds the advantage of short latency, i.e., potentially higher throughput, compared to SAR-ADC. As shown in Fig. 4 (b), the latency of Flash-ADC remains the same while SAR-ADC's latency increases linearly with the resolution bitwidth, leading to an enlarged gap with higher resolutions. Considering the energy-efficiency of the conversion per sample, Fig. 4(d) compares the energy consumption between two designs where the energy number is averaged from various input samples (i.e., the number of Ron cells along the column varies from 1 to 128). It can be observed that the average energy consumption of Flash-ADC is slightly larger than SAR-ADC when the resolution is 3-bit and the difference dramatically increases with higher resolutions. Overall, the results indicate that SAR-ADC is generally more preferable in terms of area- and energy-efficiency, making it suitable for area- and power-constrained platforms. However, taking the balance between energy consumption and throughput into consideration, i.e., the energy-delay product metric (EDP), Flash-ADC could be a better option when the required resolution is not high (≤ 4-bit) where the area and power of Flash-ADC are not yet unacceptably too large to be practical. For binary neural network, 3-bit Flash-ADC could be a good choice [24], for 8-bit (weights/activations) neural network, 5-bit SAR-ADC could be a good choice [22].

C. Process variations with analog computation

With analog computation in the CIM, inference accuracy could be degraded by the process variations. The primary variation sources include the cell-to-cell Ron variation and ADC offset. Cell-to-cell variation could be minimized by iterative write-verify technique, and sigma of conductance distribution <1% is achievable [18]. The more critical challenge is with the ADCs which quantize the partial sum. It is suggested that either nonlinear quantization is needed (for low ADC resolution) [25] or linear quantization is needed (for high ADC resolution). Moreover, the ADC offset may further degrade the inference accuracy and cause different chip instances having different inference results even for the same input.

We employ a simple current-mode comparator as shown in Fig. 3 (b) and discuss the impact of ADC offset on the inference accuracy. A 5-bit Flash-ADC based on the schematic in Fig. 3 (b) is evaluated by SPICE Monte Carlo simulations using TSMC 40nm PDK. The sense pass rate decreases with increase of the partial sum (or increase of the column current), because the real difference between I_{BL} and I_{ref} becomes small as the MUX transistor dominates the conductance. To obtain different offset magnitude, we size the W/L being 1,2 and 4 as examples, and larger W/L results in smaller process variations (and smaller MUX resistance) thus better pass rate. We use a VGG-like 8-layer network for CIFAR-10, and incorporate the sense pass rate statistics for partial sum into the PyTorch simulations. The ideal software inference accuracy is 92.04%. Considering the ADC offset, the accuracy drops to 77.79% for W/L=1, 88.49% for W/L=2 and 91.65% for W/L=4. This few percent drop of accuracy corroborates the experimental results shown in the CIM macros [17-18].

To mitigate the impact of process variations, there are possible algorithmic techniques. One technique is to introduce noise during the training phase aiming to converge the network to some local minima with shallower valley in the loss function landscape [26], though it could result in a lower baseline software accuracy. The other technique is to apply the retraining after calibrating the ADC offset in the fabricated chips. Using the actual weighted sum assuming a specific ADC offset pattern, we use 50,000 images from CIFAR-10 training dataset and retrain the entire network. After one epoch, we could recover the accuracy to 86.62% (91.40% or 91.82%) for W/L=1 (2 or 4) case. The retrained model is specified to the certain chip thus will see accuracy drop again if applied to another chip with different variation, making on chip fine-tune necessary for each chip. These results are summarized in Fig. 5 (b). We see that the accuracy could not be fully recovered, especially for the small size case since it has bigger variation, necessitating future research into this problem. From circuit design's perspective, advanced offset cancellation techniques are possible [27], however, this will add additional overhead on the area constraint that CIM is already facing with.

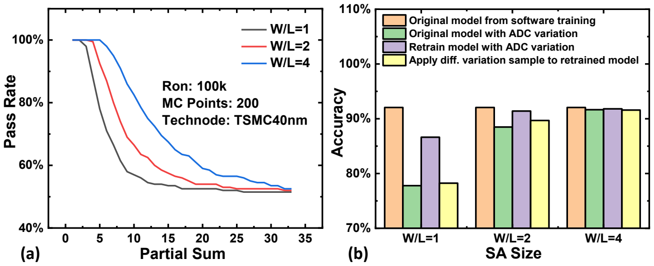

Fig. 5 (a) MC simulation results of sense pass rate for a 5-bit Flash-ADC respect to the partial sum. (b) Inference accuracy of CIFAR-10 dataset after considering the ADC offset, and after applying the retraining technique.

IV. Summary and Outlook

Significant progresses have been made to integrate eNVMs on CMOS platform recently. The tangible application for CIM inference engine is the edge computing, with significantly improved throughput and energy efficiency. Challenges down the road include: 1) device engineering is required to scale eNVM technologies to advanced logic nodes beyond 22nm; 2) compact ADC designs are needed for area and power efficiency; 3) algorithmic or circuit techniques are preferred to mitigate the impact of process variations. Suggested possible future research direction is the monolithic 3D integration, e.g. eNVMs remain at legacy node at the top tier and ADC periphery and other logic circuitry is pushed to more advanced node at the bottom tier.

Acknowledgment

This work is supported by NSF-CCF-1903951, ASCENT, one of the SRC/DARPA JUMP Centers and NSF/SRC E2CDA.

References

[1] Y.-H. Chen, et al. "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE International Conference on Solid-State Circuits (ISSCC)*, 2016.

[2] S. Yu, "Neuro-inspired computing with emerging non-volatile memory," *Proceedings of the IEEE*, vol. 106, no. 2, pp. 260-285, 2018.

[3] S. Yu, et al. "Emerging memory technologies: recent trends and prospects," *IEEE Solid State Circuits Magazine*, vol. 8, no. 2, pp. 43-56, 2016.

[4] J. Zhang, et al. "A machine-learning classifier implemented in a standard 6T SRAM array," *Symp. VLSI Circuits*, 2016.

[5] S. K. Gonugondla, et al. "A 42 pJ/decision 3.12 TOPS/W robust in-memory machine learning classifier with on-chip training," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2018.

[6] X. Si, et al. "A twin-8T SRAM computation-in-memory macro for multiple-bit CNN-based machine learning," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2019.

[7] C.-C. Chou, et al. "An N40 256K×44 embedded RRAM macro with SL-precharge SA and low-voltage current limiter to improve read and write performance," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2018.

[8] J. Pain, et al. "A 3.6Mb 10.1Mb/mm^2 embedded non-volatile ReRAM macro in 22nm FinFET technology with adaptive forming/set/reset schemes yielding down to 0.5V with sensing time of 5ns at 0.7V," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2019.

[9] J. Y. Wu, et al. "A 40nm low-power logic compatible phase change memory technology," *IEEE International Electron Devices Meeting (IEDM)*, 2018.

[10] L. Wei, et al. "A 7Mb STT-MRAM in 22FFL FinFET technology with 4ns read sensing time at 0.9V using write-verify-write scheme and offset-cancellation sensing technique," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2019.

[11] Y. J. Song, et al. "Demonstration of highly manufacturable STT-MRAM embedded in 28nm logic," *IEEE International Electron Devices Meeting (IEDM)*, 2018.

[12] S. Dunkel et al. "A FeFET based super-low-power ultra-fast embedded NVM technology for 22 nm FDSOI and beyond," *IEEE International Electron Devices Meeting (IEDM)*, 2017.

[13] G. W. Burr, et al. "Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element," *IEEE Trans. Electron Devices*, vol. 62, no. 11, pp. 3498-3507, 2015.

[14] C. Li, et al. "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," *Nature Communications*, vol. 9, 2385, 2018.

[15] F. Cai, et al. "A fully integrated reprogrammable memristor– CMOS system for efficient multiply–accumulate operations," *Nature Electronics*, vol. 2, pp. 290–299, 2019.

[16] R. Mochida, et al. "A 4M synapses integrated analog ReRAM based 66.5 TOPS/W neural-network processor with cell current controlled writing and flexible network architecture," *Symp. VLSI Technology*, 2018.

[17] C.-X. Xue, et al. "A 1Mb multibit ReRAM computing-in-memory macro with 14.6ns parallel MAC computing time for CNN based AI edge processors," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2019.

[18] S. Yin, et al. "Monolithically integrated RRAM and CMOS based in-memory computing for efficient deep learning," *IEEE Micro*, 2019.

[19] X. Sun, S. Yu, "Impact of non-ideal characteristics of resistive synaptic devices on implementing convolutional neural networks," *IEEE J. Emerg. Sel. Topics Circuits Syst. (JETCAS)*, vol. 9, no. 3, pp. 570-579, 2019.

[20] X. Peng, et al. "Optimizing weight mapping and data flow for convolutional neural networks on RRAM based processing-in-memory architecture," *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019.

[21] P.-Y. Chen, X. Peng, S. Yu, "NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning," *IEEE Trans. CAD*, vol. 37, no. 12, pp. 3067-3080, 2018.

[22] X. Peng, et al. "DNN+NeuroSim: An end-to-end benchmarking framework for compute-in-memory accelerators with versatile device technologies," *IEEE International Electron Devices Meeting (IEDM),* 2019.

[23] M.-F. Chang, et al. "Challenges and circuit techniques for energy-efficient on-chip nonvolatile memory using memristive devices," *IEEE J. Emerg. Sel. Topics Circuits Syst. (JETCAS)*, vol. 5, no. 2, pp. 183-193, 2015.

[24] X. Sun, et al. "XNOR-RRAM: A scalable and parallel synaptic architecture for binary neural networks," *Design, Automation & Test in Europe (DATE)*, 2018.

[25] R. Liu, et al. "Parallelizing SRAM arrays with customized bit-cell for binary neural networks," *Design Automation Conference (DAC)*, 2018.

[26] Y. Long, et al. "Design of reliable DNN accelerator with un-reliable ReRAM," *Design, Automation & Test in Europe (DATE)*, 2019.

[27] C.-P. Lo, et al. "Embedded 2Mb ReRAM macro with 2.6 ns read access time using dynamic-trip-point-mismatch sampling current-mode sense amplifier for IoE applications," *Symp. VLSI Circuits*, 2017.