Energy-Efficient Deep In-memory Architecture for NAND Flash Memories

Sujan K. Gonugondla*, Mingu Kang*, Yongjune Kim*, Mark Helm†, Sean Eilert†, Naresh Shanbhag*
*University of Illinois at Urbana-Champaign, †Micron Technology Inc.

***Abstract*—This paper proposes an energy-efficient deep in-memory architecture for NAND flash (DIMA-F) to perform machine learning and inference algorithms on NAND flash memory. Algorithms for data analytics, inference, and decision-making require processing of large data volumes and are hence limited by data access costs. DIMA-F achieves energy savings and throughput improvement for such algorithms by reading and processing data in the analog domain at the periphery of NAND flash memory. This paper also provides behavioral models of DIMA-F that can be used for analysis and large scale system simulations in presence of circuit non-idealities and variations. DIMA-F is studied in the context of linear support vector machines and k-nearest neighbor for face detection and recognition, respectively. An estimated 8×-to-23× reduction in energy and 9×-to-15× improvement in throughput resulting in EDP gains up to 345× over the conventional NAND flash architecture incorporating an external digital ASIC for computation.**

I. Introduction

Computing platforms that efficiently perform inference and data mining applications have attracted significant interest today. These applications require the implementation of complex machine learning algorithms that operate on large volumes of data. Conventional computing architectures are designed such that, computation and storage are inherently separated. Such systems often suffer from large latency and data transfer costs. Though processor speed and storage density have seen exponential growth over time, data transfer rates between the processor and memory have seen limited growth. Hence, techniques that radically reduce the data access costs are imperative.

Memory access energy cost has two major components: 1) memory read, and 2) data transfer. One way to reduce the data transfer costs is to bring computation near the memory. There have been many attempts at integrating memory and computation such as [1], [2], where Hamming distance based search are performed near memory. Similar approaches were proposed for storage class memories such as [3], where search algorithms were performed on solid state drives (SSDs). On other hand, works such as Minerva [4], and XSD [5] tackle more general processing tasks inside the SSD by introducing a GPU or a specialized processor to accelerate large vector computations. These techniques can successfully minimize traditional storage I/O limitations. However, since storage and logic process technologies are usually not compatible, these techniques cannot be easily applied in the same technology.

Alternatively, deep in-memory architecture (DIMA) [6]–[13] was proposed to drastically reduce memory read costs by embedding mixed signal computations in the periphery of a SRAM bitcell array. DIMA on SRAM is useful in tackling problems in the kB scale. However, to address large-scale machine learning problems (in the GB/TBs), in-memory computing architectures for high-density storage technologies are essential. NAND flash memories are an industry standard for large-scale storage. However, its throughput and energy consumption are primarily limited by its off-chip I/O interface and bandwidth (limited to 800MB/s [18]). Furthermore, the external bandwidth of a typical SSD is 16× smaller than the internal bandwidth as observed in [3]. Hence, techniques that enable processing on-chip would thus achieve large throughput and energy savings by minimizing the need to transfer data off-chip. This makes in-memory computing on NAND flash an attractive proposition. However, there are various challenges that need to be overcome in order to enable such techniques. These challenges primarily arise from large threshold voltage variation, small pitch, and technology limitations.

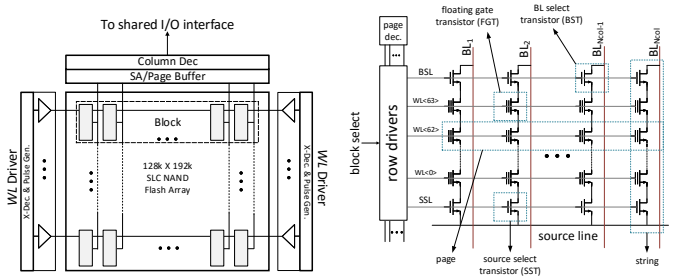
This paper proposes deep in-memory architecture for NAND flash (DIMA-F) which brings computing functionality into NAND flash memories. DIMA-F reads the stored data and processes highly parallel dot-products on single-level cell (SLC) NAND flash memories in the analog domain. This architecture can be used to perform classification, compression, filtering, and feature extraction. This paper employs behavioral models for the proposed architecture that account for circuit non-idealities in order to estimate the accuracy of inference. DIMA-F is evaluated in the context of face detection and face recognition on the Caltech 101 database [14] and the Extended Yale B database [15], respectively. System level simulations show marginal degradation in accuracy as compared to fixed point implementations, while achieving between $8\times$-to-$23\times$ energy savings, $9\times$-to-$15\times$ throughput gain, and $72\times$-to-$345\times$ improvements in energy delay product (EDP) compared to the conventional NAND flash architecture incorporating an external digital ASIC for computation.

II. Background

A. Deep in-memory architecture (DIMA)

DIMA was originally proposed for pattern recognition on SRAM [6]. Subsequent works [9], [10] show IC implementations and demonstrate up to $53\times$ EDP improvement [10] over conventional architectures.

DIMA is successful in implementing highly parallel vector operations such as dot products and vector distances, that are essential for machine learning. It has two major features. The first is multi-row functional read (MR-FR), where multiple bits across rows are read in the analog domain on the bit-lines (BLs) weighted using pulse width/amplitude modulated



(a) (b)

Fig. 1: Architecture of a conventional SLC NAND flash: a) *plane*, and b) a *block*.

(PWAM) word-line (WL) signals. The PWAM-WLs enable functionality such as digital to analog conversion (D/A) or multiplications as shown in [10] and [9], respectively. The second is bit-line processing (BLP) and Cross BLP (CBLP). While the BLP performs scalar operations such as multiplication, absolute difference, and additions on the data read via MR-FR, CBLP aggregates BLP outputs by charge-sharing. By processing data low-swing analog domain on the BLs, DIMA [10] achieves improvements in throughput and energy efficiency of 5.3× and 10×, respectively.

B. SLC NAND flash memory architecture

NAND flash is a non-volatile storage/memory architecture that uses floating gate (FG) transistors as the basic storage cell. Data in NAND flash memories is stored as threshold voltages of the FG transistors which are induced by the charges on the FGs. In SLC NAND flash memory, the FG transistors have two states, i.e., an erased state (low threshold voltage) and a programmed state (high threshold voltage), corresponding to a single logical bit.

A NAND flash chip contains a memory array, a control unit, high voltage generation circuitry for read and write operations, buffers to store/transmit data, and I/O interface circuitry. NAND flash memory is organized as multiple memory banks referred to as *planes*. Figure 1(a) shows the typical architecture of a NAND flash plane. Each plane is further horizontally divided into *block*. A block is in turn divided into *pages* horizontally and *strings* vertically. NAND flash strings typically contain 64-128 FG transistors connected serially as shown in Fig. 1(b). They are accessed through a group to 64-128 word-lines. The data stored across the FG transistors sharing a single WL is called a page.

III. Deep In-Memory Architecture for NAND Flash (DIMA-F)

Though bringing computational functionality to NAND flash memory is highly beneficial in relaxing the constraints imposed by I/O circuitry, there are major challenges to be addressed. These challenges arise from small bitcell pitch, high variability of NAND flash memories, and limitations in speed of NAND flash technologies for computations.

Figure 2 shows the proposed architecture for DIMA-F. It consists of the following blocks: a) a memory array that allows *multi-column functional read* (MC-FR) that converts a W-bit word to an analog voltage on a capacitor, b) a *multi-bit-line processor* (MBLP) pitch-matched to BLs read in MC-FR that performs scalar multiplication, c) a buffer that stores a reference vector or weights used during the MBLP operations, d) a cross bit-line processor (CBLP) that performs dimensionality reduction by summation to implement dot product, and e) an ADC or slicer that converts the analog output of the CBLP into the digital domain.

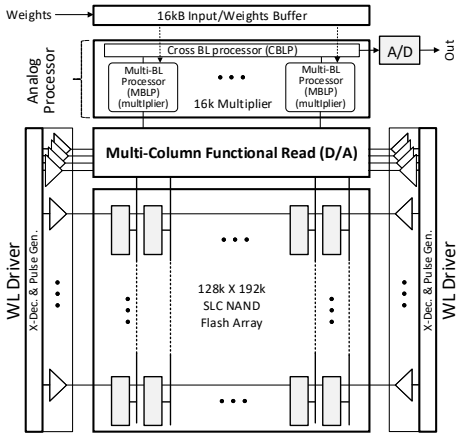


Fig. 2: Proposed deep in-memory architecture SLC NAND flash (DIMA-F).

A. Multi-column functional read (MC-FR)

Consider a data word D stored as a W-bit binary vector $\mathbf{d} = \{d_{W-1}, d_{W-2}.., d_0\}$. The goal of the MC-FR operation is to read in analog voltage proportional to the decimal value of the data stored in the flash array ($\Sigma_{n=0}^{W-1} 2^n d_n$). To enable this the bits are stored horizontally in a page as shown in Fig. 3(a). Hence, N_{BL}/W words per plane are read in parallel using MC-FR, where N_{BL} is the number of BLs in a plane.

Figure 3 shows the architecture and the timing for MC-FR. During MC-FR, the WL associated with the page being read is set to a voltage V_{read}, while other WLs in that block are set to voltage V_{pass}. In the precharge phase, the gate voltages of all SEL transistors are set to $V_{pre} + V_{TH}$ and the gate voltage of PCH transistors is set to $V_{dd} + V_{TH}$, charging the BLs and OUT nodes to approximately V_{pre} and V_{dd}, respectively. In the evaluation phase, PCH transistors are effectively turned off allowing the discharge of the C_{OUT} capacitor. The SEL transistors act as clamp transistors and are pulse width modulated such that the overall discharge on C_{OUT} is proportional to the decimal value of D. The current through string $I_{s,i} \approx I_{on}$ if $d_i = 1$ else $I_{s,i} \approx 0$. Choosing $T_i = 2^i T_0$ enables a D/A operation via MC-FR resulting in $\Delta V_{\text{OUT}} = \frac{I_{on} T_0}{C_{\text{OUT}}} \sum_{i=0}^{W-1} 2^i d_i$.

B. Multi-BL and cross BL processing (MBLP and CBLP)

A capacitive multiplier similar to the one introduced in [8], [10] is used here (see Fig. 4). The analog value obtained from MC-FR is multiplied by a digital number stored in the input/weight buffer. The values from the input/weight buffer are given sequentially to the multiplier. The switching sequence

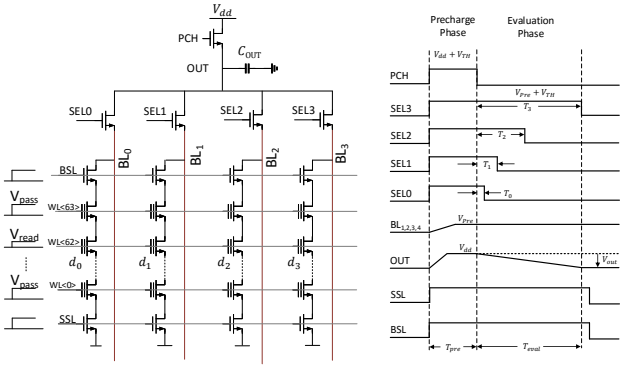


Fig. 3: Proposed MC-FR technique for $W = 4$: a) architecture, and b) timing.

for multiplication is shown in Fig. 4. The multiplication process involves sequential charge sharing based on the digital inputs p_i. The effective voltage discharge of V_M is proportional to the product, $\Delta V_M = \sum_i (0.5)^i p_i \Delta V_{\text{OUT}}$. Multiplier outputs across the plane are charge-shared to perform an average/addition operation on one of the two CBLP rails based on the sign of the outputs.

IV. Behavioral Models

Behavioral models are required to perform large-scale application level simulations. These models need to account for non-idealities such as threshold voltage variations, read and program disturbance, diffusion, inter-cell interference (ICI), and back pattern dependency. We propose behavioral models that estimate effective string resistance by using long channel approximation. These models are able to capture the behavior of NAND flash array at a string level which is sufficient for application level simulations. Effective resistance of a cell during read $R(V_G)$ is estimated at super threshold by,

$$R(V_G) \approx \big(k(V_G - V_{TH})\big)^{-1}, \tag{1}$$

and at near threshold by,

$$R(V_G) \approx \left(I_S \Big(\frac{W}{L}\Big) e^{\frac{V_G - V_{TH}}{nV_T}} \frac{V_{DS}}{V_T} \right)^{-1}. \tag{2}$$

Equations (1) and (2) allows us to estimate the effective string current (I_s) as a function of V_{read} using,

$$I_s(V_{read}) = \frac{V_{BL}}{R_k(V_{read}) + \sum_{i=0, i\neq k}^{63} R_i(V_{pass})}, \tag{3}$$

where R_i is the effective resistance of the cell i, V_{BL} is the BL voltage, and k is the cell being read. While effects such as back-pattern dependencies are captured by the model, other variations are accounted by modeling the threshold voltage as a Gaussian random variable ($\mathbf{V}_{TH}$) [16]. Thus, cell resistance and the overall string current during the read operations are also treated as random variables, $\mathbf{R}_i$, and $\mathbf{I}_s$. Hence, the output of the MC-FR is also a random variable:

$$\Delta \mathbf{V}_{\text{OUT}}(V_{read}) = \sum_{i=0}^{W-1} \frac{\mathbf{I}_{s,i}(V_{read}) T_i}{C_{\text{OUT}}}. \tag{4}$$

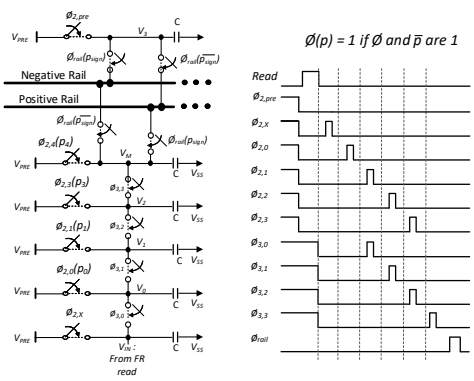


Fig. 4: Signed multiplier and associated timing.

V_{read} is chosen to minimize the mean squared error from the ideal output.

V. Energy models

We employ and build upon the energy models proposed in [17]. Energy consumption during MC-FR is dominated by the energy to charge BLs (E_{BL}), the energy to toggle WLs (E_{WL}) and the energy dissipation due to the string currents (E_s).

In a conventional NAND flash suffers large stall times between page reads due to the limited speed of the shared I/O bus. The MC-FR technique reduces the read energy consumptions as compared to conventional current sensing. Here, the lack of stall times prevent BLs from completely discharging between the consecutive read cycles which reduce E_{BL}. This reduces precharge times allowing throughput improvements, and reduction of E_s. Further the lack of stall times between consecutive reads within a block reduces the number of WL transitions between V_{read} to V_{pass} to two, as other WLs need not be discharged between reads. Thus the average energy for a page read of DIMA-F E_{DIMA-F} is,

$$E_{DIMA-F} = E_{BL} + E_{WL} + E_s, \tag{5}$$

$$E_{BL} = 0.5 N_{BL} C_{BL} \Delta V_{BL} V_{dd}, \tag{6}$$

$$E_{WL} = C_{WL} V_{pass} (V_{pass} - V_{read}) / \eta_{WL}, \tag{7}$$

$$E_s = V_{dd} I_{s,avg} (T_{pre} + T_{eval}), \tag{8}$$

where $I_{s,avg}$ is the average string current, C_{BL} and C_{WL} are BL and WL capacitances, η_{WL} are the efficiency of the charge pump driving the WLs while T_{pre} and T_{eval} are the precharge and evaluation times, respectively.

VI. Evaluation Methodology

DIMA-F is tested over the following applications to demonstrate its benefits.

Linear support vector machines (SVM): Linear SVM is a simple and widely used classifier, which uses dot-product for decisions. We use this in the case of face detection on Caltech 101 database [14]. Linear SVM classification involves computing, $y = \boldsymbol{w}^T \boldsymbol{x} + b$, where $\boldsymbol{w}$ and b are pre-trained weights and $\boldsymbol{x}$ is the image vector to be classified. The image is classified as *face* if $y > 0$ and as *non-face* otherwise.

Cross correlation based detection (CC): Cross-correlation is a useful metric to measure the similarity between two data

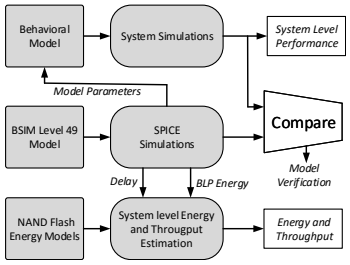
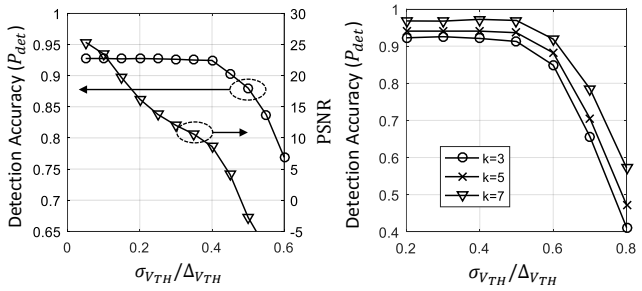


Fig. 5: Simulation methodology.



(a) (b)

Fig. 6: Detection accuracy P_{det}: a) SVM, and b) CC based k-NN algorithm as a function of threshold voltage variation.

vectors. We use CC as a distance metric in k-nearest neighbor (k-NN) algorithm for face recognition on Extended Yale B database [15]. It has 2336 test images with 28 classes.

For simulations in this paper, NAND flash memory on 32nm node with 16kB per pages, 64 pages per block, 3000 blocks per plane and 4 planes per IC is used. The images under test are scaled to 200×320 dimensions, where each pixel is represented in 8-bit fixed-point. Additionally, Extended Yale B images are pre-normalized for the CC based algorithms. Each image is rearranged into a 64k pixel vector and stored on 4 pages across 4 planes. Input/weight buffer stores weights in the case of SVM and reference images in the case of CC. The dot product outputs are converted into digital domain via 8-bit ADC for post-processing. Simulation methodology is described in Fig. 5. System level simulations were performed using behavior models described in Section IV with model parameters obtained from SPICE simulations of a NAND flash array. Two architectures for a conventional baseline are considered to demonstrate the benefits of DIMA-F: a) single NAND IC with an off-chip processor, and b) a standard solid state drive (SSD) containing 16 ICs with an external processor.

VII. Simulation Results

Figure 6 shows the detection accuracy (P_{det}) of DIMA-F as a function of $\sigma_{V_{TH}}/\Delta_{V_{TH}}$, where $\sigma_{V_{TH}}$ is the variance of the threshold voltage and $\Delta_{V_{TH}}$ is the mean threshold voltage difference between programmed and erased cells. Peak signal-to-noise ratio (PSNR) of the image read via MC-FR degrades with increasing threshold voltage variation. The SVM algorithm is relatively more robust to threshold voltage variation. Threshold voltage variations ($\sigma_{V_{TH}}/\Delta_{V_{TH}}$) in a typical NAND flash memory ranges between 0.2 to 0.3. The detection accuracy of the SVM algorithm is 92% in this range.

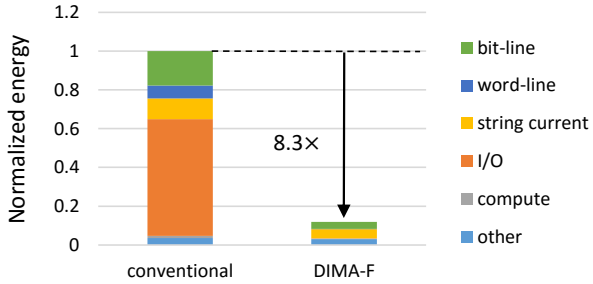


Fig. 7: Estimated energy savings in the single IC scenario.

Detection accuracy of the CC algorithm is about 88% for Top-1 and increases to 95% for Top-3 case in the $\sigma_{V_{TH}}/\Delta_{V_{TH}}$ range of 0.2 to 0.5, where Top-k is the accuracy when the correct label is among the top k candidates. The accuracy improved under the application of k-NN algorithm using CC as the distance metric. Detection accuracy of at least 92% for $k = 3$ and 95% for $k = 5$ is observed in the $\sigma_{V_{TH}}/\Delta_{V_{TH}}$ range of 0.2 to 0.4.

A. Throughput and Energy

The throughput of individual conventional NAND IC is limited by the I/O that has a data transfer rate of 800MB/s (ONFI 4 standard [18]). In an SSD the throughput is further limited by the PCIe bandwidth of 8GB/s [3]. Since DIMA-F based SSD does not have such I/O limitations, it can read at a rate of 7.56GB/s/IC. Therefore, a throughput improvement of 9× and 15× is achieved compared to a conventional system with a single IC and SSD scenario, respectively.

Energy estimates were obtained from models described in Section V with parameters obtained via SPICE simulations. I/O energy is estimated conservatively such that the device would meet the typical ONFI 4 standard. We observe that I/O energy is the dominating component of the conventional system's energy consumption. In a single IC scenario, the I/O load is conservatively estimated to include the load of a single NAND IC connected to a bus. However, in an SSD scenario, multiple ICs are connected to the bus and would proportionally increase the I/O load. An overall 8.3× and 23× energy savings are achieved compared to single IC scenario and SSD scenario, respectively. Energy breakdown for the single IC scenario is shown in Fig. 7.

VIII. Conclusion

Scaling trends in memory density and bandwidth suggest that the memory access problem will get worse over time. In-memory computing provides an alternative approach to address this problem. In this paper we propose DIMA-F, which achieves 8×-to-23× energy savings and 9×-to-15× throughput improvements by overcoming the I/O barriers on SLC NAND flash memory. Future work includes extending DIMA-F to MLC NAND flash memories and other non-volatile memories.

Acknowledgment

This work was supported in part by Systems On Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.

REFERENCES

[1] H. J. Mattausch, T. Gyohten, Y. Soda, and T. Koide, "Compact associative-memory architecture with fully parallel search capability for the minimum Hamming distance," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 2, pp. 218–227, 2002.

[2] Y. Oike, M. Ikeda, and K. Asada, "A high-speed and low-voltage associative co-processor with exact Hamming/Manhattan-distance estimation using word-parallel and hierarchical search architecture," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 8, pp. 1383–1387, 2004.

[3] E. Doller, A. Akel, J. Wang, K. Curewitz, and S. Eilert, "DataCenter 2020: Near-memory acceleration for data-oriented applications," in *IEEE Symposium on VLSI Circuits Digest of Technical Papers*, 2014, pp. 1–4.

[4] A. De, M. Gokhale, R. Gupta, and S. Swanson, "Minerva: Accelerating data analysis in next-generation SSDs," in *IEEE International Symposium on Field-Programmable Custom Computing Machines*, 2013, pp. 9–16.

[5] B. Y. Cho, W. S. Jeong, D. Oh, and W. W. Ro, "XSD: Accelerating mapreduce by harnessing the GPU inside an SSD," in *Proceedings of the 1st Workshop on Near-Data Processing*, 2013.

[6] M. Kang, M.-S. Keel, N. R. Shanbhag, S. Eilert, and K. Curewitz, "An energy-efficient VLSI architecture for pattern recognition via deep embedding of computation in SRAM," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2014, pp. 8326–8330.

[7] N. Shanbhag, M. Kang, and M.-S. Keel, "Compute memory," US Patent No. 9,697,877 B2, Issued July 4 2017.

[8] M. Kang, S. K. Gonugondla, M.-S. Keel, and N. R. Shanbhag, "An energy-efficient memory-based high-throughput VLSI architecture for convolutional networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 1037–1041.

[9] J. Zhang, Z. Wang, and N. Verma, "A machine-learning classifier implemented in a standard 6T SRAM array," in *IEEE Symposium on VLSI Circuits Digest of Technical Papers*, 2016, pp. 1–2.

[10] M. Kang, S. Gonugondla, A. Patil, and N. Shanbhag, "A 481pJ/decision 3.4 M decision/s Multifunctional Deep In-memory Inference Processor using Standard 6T SRAM Array," *arXiv preprint arXiv:1610.07501*, 2016.

[11] M. Kang, S. K. Gonugondla, and N. R. Shanbhag, "A 19.4 nJ/decision 364K decisions/s In-memory Random Forest Classifier in 6T SRAM Array," in *IEEE European Solid-State Circuits Conference (ESSCIRC)*, Sept 2017, pp. 263–266.

[12] S. K. Gonugondla, M. Kang, and N. Shanbhag, "A 42pJ/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training," in *IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2018, pp. 490–491.

[13] M. Kang, S. K. Gonugondla, A. Patil, and N. R. Shanbhag, "A multi-functional in-memory inference processor using a standard 6T SRAM array," *IEEE Journal of Solid-State Circuits (JSSC)*, 2018.

[14] L. Fei-Fei, R. Fergus and P. Perona, "One-Shot learning of object categories," *IEEE Trans. Pattern Recognition and Machine Intelligence.*, 2004.

[15] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[16] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Threshold voltage distribution in MLC NAND flash memory: Characterization, analysis, and modeling," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013*. IEEE, 2013, pp. 1285–1290.

[17] V. Mohan, T. Bunker, L. Grupp, S. Gurumurthi, M. R. Stan, and S. Swanson, "Modeling power consumption of NAND Flash memories using Flashpower," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 7, pp. 1031–1044, 2013.

[18] Open NAND flash Interface, "Open NAND Flash Interface Specification Version 4," *http://www.onfi.org/media/onfi/specs/onfi_4_0-gold.pdf*, 2014.