Exploiting Intrinsic Variability of Filamentary Resistive Memory for Extreme Learning Machine Architectures

Manan Suri, *Member, IEEE*, and Vivek Parmar, *Graduate Student Member, IEEE*

***Abstract*— In this paper, we show for the first time how unavoidable device variability of emerging non-volatile resistive memory devices can be exploited to design efficient low-power, low-footprint Extreme Learning Machine (ELM) architectures. In particular we utilize the uncontrollable Off-state resistance (Roff/HRS) spreads, of nanoscale *filamentary*- resistive memory devices, to realize random input weights and random hidden neuron biases; a characteristic requirement of ELM. We propose a novel RRAM-ELM architecture. To validate our approach, experimental data from different *filamentary*- resistive switching devices (CBRAM, OXRAM) is used for full network simulations. Learning capability of our RRAM-ELM architecture is illustrated with the help of two real world applications- (i) diabetes diagnosis test (classification) and (ii) *SinC* curve fitting (regression).**

***Index Terms*—Cognitive Computing, Resistive Memory, OXRAM, CBRAM, RRAM, Extreme Learning Machine, Machine Learning, Neuromorphic, Brain-Inspired, stochastic computing.**

I. Introduction

Reaseach and development activity in the field of dedicated hardware concepts for cognitive, natural or bio-inspired computing has spiked phenomenally over the last few years. A broad underlying common factor in many such approaches is the idea of co-integrating hybrid CMOS architectures with emerging nanoscale devices for computing applications [15]. Such emerging nanoscale devices cover a vast breadth of concepts and materials ranging from the more exploratory nano-organic transistors [17] (NOMFETs) to graphene nanoribbons [16] to even close-to industrial STT-MRAM [18] (spin-transfer torque magnetic memory) devices.

Emerging resistive memory technology (also sometimes referred to as *memristors* or RRAM/ReRAM) is one strong promising class of nanodevices, widely proposed for implementing synaptic/weighting functionality in bio-inspired nanoscale architectures [19]. RRAM devices have several inherent advantages, such as- (i) high integration density (ultra-compact feature size – $4F^2$), (ii) full CMOS compatibility (iii) non-volatility (iv) high cycling endurance, (v) stable retention characteristics (vi) multi-level programming (vii) possible 3D integration, and (viii) low operating voltages/currents and low-leakage [20]. RRAM technologies can be classified in to different sub families such as- Phase-change memory (PCM), metal-oxide based memory (OXRAM) and conductive-bridge memory (CBRAM), differing in terms of the active layer materials and the underlying physics governing the switching behavior. In this paper we will mainly focus on *filamentary*- RRAM devices (OXRAM and CBRAM). We will show how undesirable nanoscale variability of such devices can be rather exploited to design Extreme Learning Machine (ELM) architectures in an energy/area efficient manner. We choose the ELM learning framework since ELMs are assumed to be bio-inspired. In particular, ELMs share the similarity of the brain's independence in tuning/selecting input layer parameters irrespective of the environment or nature of the input data being processed [7]. ELMs offer significant advantages such as fast learning speed, ease of implementation, and minimal human intervention, for many regression/multi-class classification applications when compared to other supervised/unsupervised learning paradigms such as back propagation [1] and support vector machines (SVM) [7], which tend to be more computation intensive for large datasets. Till very recently the hardware realization of ELMs [5] [12] was mainly described in the full digital domain using multipliers and DDR memory which makes the design inherently slower as well as more power consuming and '*always-on*' in some cases. Hybrid neuro-memristive ELM designs were recently proposed in literature that mainly emphasize on use of memristors for implementing the output weights [4]. In this work, we show a broader application and scope of RRAM devices in ELM hardware extending their use to realize random input/hidden weights and random neuron biases.

Section II explains the basics of *filamentary*- resistive memory (RRAM) devices and discusses the experimentally observed variability that we exploit for our design. Section III outlines the basic principle of ELM and explains our proposed RRAM-ELM architecture. Section IV discusses the results of two real-world applications that we simulated to validate our proposed architecture.

II. Nano-scale Filamentary RRAM

A. Basic Principle and Working

RRAM devices are two-terminal MIM-type structures (metal-insulator-metal) sandwiching an active insulator layer, between metallic electrodes (see Fig. 1). The active layer exhibits reversible non-volatile switching behavior on application of appropriate programming current/voltage across the device terminals. In the case of *filamentary*- RRAM

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNANO.2015.2441112, IEEE Transactions on Nanotechnology

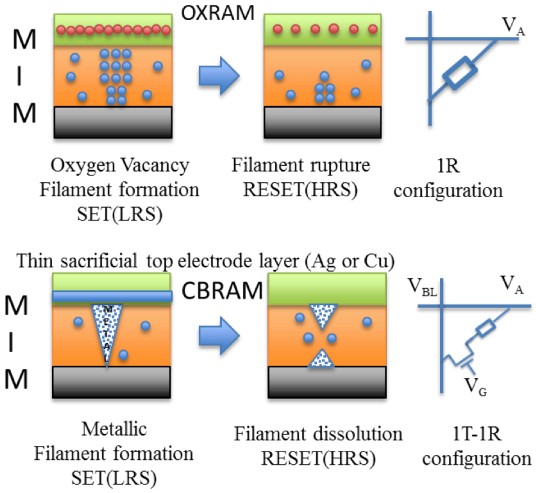
OXRAM
M
I
M
Oxygen Vacancy Filament formation SET(LRS)
Filament rupture RESET(HRS)
1R configuration
V_A
Thin sacrificial top electrode layer (Ag or Cu)
CBRAM
M
I
M
V_BL
V_A
V_G
Metallic Filament formation SET(LRS)
Filament dissolution RESET(HRS)
1T-1R configuration

Fig. 1: Basic switching principle of filamentary RRAM devices. 1R, 1T-1R configurations are also shown. Selector device can either be a diode or transistor.

devices, formation of a conductive filament in the active layer, leads the device to a low-resistance (LRS/On) SET-state, while dissolution of the filament puts the device in a high-resistance (HRS/Off) RESET-state. For OXRAM devices, the conductive filament is composed of oxygen vacancies and defects [21], while in the case of CBRAM it consists of reduced metal ions sourced from a thin sacrificial metal anode [22]. For both CBRAM and OXRAM devices, the SET-state resistance (LRS) values can be well defined by controlling the dimensions of the conductive filament [8], [21], or the conduction mechanism (specific to some OXRAM devices) [23], which depends on the amount of current flowing through the active layer. Current flowing through the active layer is controlled either by externally imposed current compliance or by using an optional selector device that is used to drive the RRAM device (i.e. 1R-1T/1D configurations).

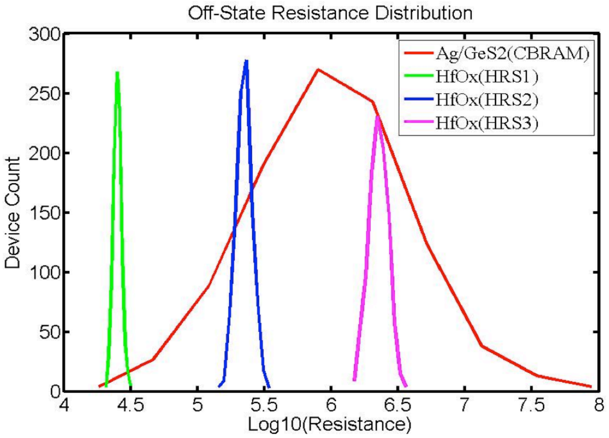

Fig. 2: Extracted HRS (Roff) log-normal distributions for filamentary RRAM devices listed in Table 1.

Table 1: Extracted HRS parameters for filamentary-RRAM devices. V_G,V_{BL} correspond to gate-voltage and bit-line voltage in 1T-1R configuration. V_A corresponds to Anode voltage in 1R configuration.

Device, Technology, reference	**Mean (kΩ)**	**Variance ($\log_{10}$R)**	**Reset programming conditions (configuration)**
CBRAM, (Ag/GeS$_2$),[8]	892.86	0.6	V_G= 2 V, V_{BL} = 2 V, 10 µs, (1R-1T)
OXRAM, (HfO$_x$), [14]	25.12	0.03	V_A= -2.4 V, 50 ns (1R)
OXRAM, (HfO$_x$), [14]	221.82	0.06	V_A=- 2.7 V, 50 ns (1R)
OXRAM, (HfO$_x$), [14]	2238.72	0.07	V_A= -3 V, 50 ns (1R)

B. *Off-State Variability*

RESET-state resistance values (HRS) of filamentary-RRAM devices generally present a large variable dispersion. The spread in HRS values arise due to stochastic breaking or uncontrolled dissolution of the conductive filament during the reset process. Different nanoscale factors, such as presence of unavoidable defects close to the filament, interface effects, active-layer/electrode non-uniformity, material degradation/morphological changes on cycling, percolation paths, and process variations may lead to HRS variability in cycle to cycle or device to device realizations [13], [24], [25]. Table 1 lists the HRS variability parameters (mean-µ, standard deviation-σ) for HfO$_x$ based OXRAM, and Ag/GeS$_2$ based CBRAM devices that were extracted from [14] and [8] respectively. We chose different RESET programming conditions for HfO$_x$ based OXRAM devices to study a wide range of mean HRS values. Device-to-device HRS spreads, for CBRAM and OXRAM, are shown in Fig. 2, generated by applying a log-normal distribution to the parameters listed in Table 1. While HRS variability profiles like the ones shown in Fig. 2, are undesired for implementing *multi-level* memory states, in the following sections we show they can be exploited as an advantage in ELM architecture design. The reset-switching energy for Ag/GeS$_2$ CBRAM devices is approximately 1.8 nJ/event [8], while that for the OXRAM devices is ~ 15 pJ/event [14].

III. Extreme Learning Machines

A. *Basics of Extreme Learning Machines (ELM)*

An ELM is basically a SLFN (Single layer feed forward Neural Network) consisting of- hidden layer synapses with randomly assigned weights, a hidden neuron layer with an infinitely differentiable activation function, and an output layer with synaptic weights determined by the learning rule shown in Fig. 3. Unlike other algorithms that try to assign hidden layer synaptic weights to some predetermined values while solving a QPP (Quadratic Programming Problem) for the output synaptic weights, (ex- Support Vector Machines), or improve them over successive iterations (ex-Back propagation) [7], ELMs use random distributions of input weights and hidden layer neuron biases, that remain fixed during learning [1]. Output synaptic weights in ELMs are determined through a simple L1-minimization scheme i.e. using a matrix inversion [9]. Use of fixed random input layer weights allows the ELM to obtain a very good generalization behavior, compared to other gradient based neural networks which explicitly try to tune all parameters [10]. ELM's simple learning algorithm gives it a strong advantage in terms of speed when compared to SVMs and other bio-inspired algorithms [1].

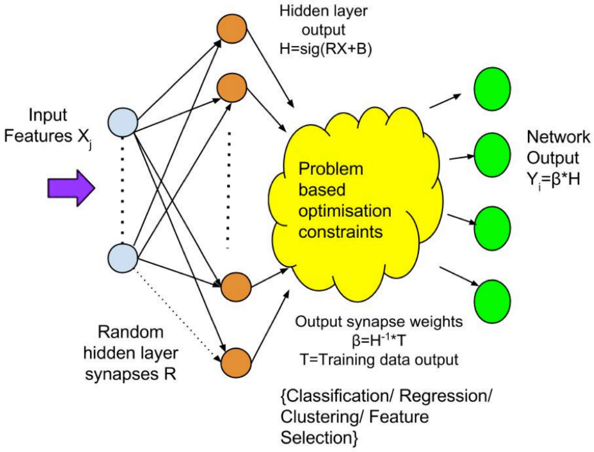

Fig. 3: Basic ELM Framework and governing equations used to calculate output synaptic weights during training. 'B' denotes random hidden neuron biases.

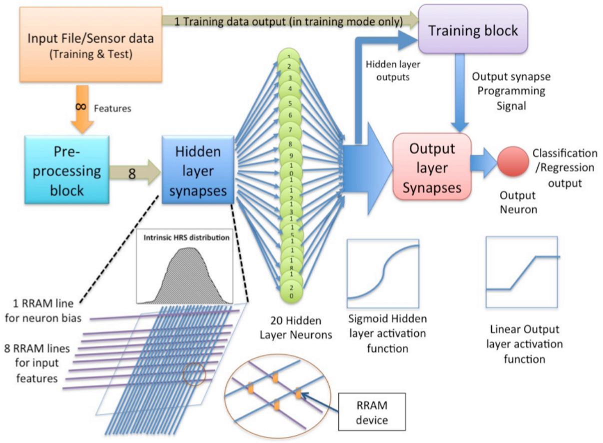

Fig. 4: Proposed RRAM-ELM Architecture. For diabetes diagnosis classification 8 input feature vectors and 20 hidden layer neurons were used. HRS variability of RRAM matrix was exploited for implementing both- random input weights and random neuron biases.

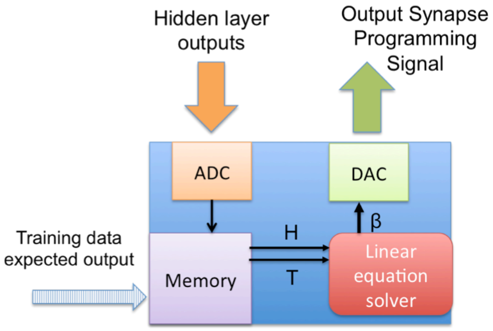

Fig. 5: Training block schematic

B. Our RRAM-ELM Architecture

Our proposed RRAM-ELM architecture is shown in Fig. 4. The system is operated in *training*- and *test*- modes. Block wise description is as follows-

1) *Pre-Processing block*

Real-time input data from sensors or stored information is first pre-processed. Based on the nature of the raw data (audio/visual/digital) the pre-processing block may include filtering, digital-to-analog conversion (DAC) and normalization steps. Output of the pre-processing block is in the form of voltage signals, fed directly to a network of hidden layer synapses. The magnitude of the output voltage signals of the pre-processing block, is scaled in the range of '*read'* voltages of the RRAM synaptic devices. This ensures that output of pre-processing stage doesn't program the input layer synapses, when not required.

2) *Hidden layer synapses*

Hidden layer synapses are realized using a crossbar (1R) or matrix (1T-1R) configuration of filamentary RRAM devices. The minimum size of the crossbar should be **(N+1) x M**; where N denotes the total number of input features and M denotes the total number of hidden layer neurons. ELM architectures require fixed random input weights. A RESET operation is performed initially for the RRAM synaptic matrix, before the launch of *training*-mode, to obtain the type of HRS distribution described in Section II-B. The variable HRS spreads give rise to random input layer synaptic weights. It is also worth noting that cycle-to-cycle device variation will not impact the ELM learning performance as for each dataset and application, the input/output stage synaptic weights in our ELM is programmed just once. Exploiting intrinsic HRS spreads is area and energy efficient as use of costly extrinsic techniques such as random number generator or PRNG (pseudo-random) circuits is avoided [4] [12]. Mean (μ) value of the HRS distribution can be controlled, by tuning the reset programming conditions (pulse widths/amplitude) as shown in Table 1 and Fig. 2.

3) *Hidden layer neurons*

Current flowing through the RRAM synaptic matrix is weighted by the resistances of the RRAM devices, and constantly integrated in the hidden layer neuron block. The ELM learning algorithm can work with many infinitely differentiable activation functions such as sine, radial basis function, etc. We chose standard sigmoid function as the hidden neuron for our architecture, based on the circuit implementation described in [2]. For all simulations described in this paper, we used a sigmoid gain factor of 10^4.

Random neuron biasing can also be achieved in an extremely area efficient manner by exploiting the hidden layer RRAM synaptic matrix. (N+1) th line in the RRAM matrix (assuming the system has N input features) can be biased using a constant voltage source, applied across the top terminal of all the RRAM devices. The bottom terminals of each individual device in the (N+1) th line is directly connected to the individual hidden layer neurons. Since the resistance of all RRAM devices in the matrix is a point in one of the HRS distributions shown in Fig. 2, the resultant current being fed in to the hidden layer neurons from the (N+1) th RRAM line also follows a similar distribution. This implementation avoids the need of any external bias randomization circuit.

4) *Training block*

This block is active only during the *training*-mode operation of the network. For each training data point, the hidden layer neuron output (H) and the expected output (T) are stored inside the training block (see Fig. 5). Thus the minimum size of training block memory is given by the following expression.

*Training block memory = No. of training samples x (No. of hidden neurons + 1) * Data bit width*

A matrix inversion operation (H^{-1}*T) is performed in the training block as described in [1], in order to solve the linear system of equations, and generate the output synaptic weights. The matrix inversion can be implemented using algorithms such as Gauss-Jordan elimination or QR decomposition using the tool discussed in [11].

5) *Output layer synapses*

These are programmed only at the end of the training mode. As the output layer weights are generated from a matrix inversion operation they will have a wide dynamic range. Combinations of multiple RRAM devices can be used to realize such synapses, as also described in [4].

6) *Output neuron*

In contrast to the hidden layer, which uses a sigmoid activation function, for the output neuron we make use of a linear activation function, since the output layer is for a general purpose architecture that can perform both classification and regression. During the training phase the output layer is switched off in order to minimize power dissipation.

IV. Results & Discussion

In order to validate the learning capability of the proposed RRAM-ELM architecture (Fig. 4), we simulated the full architecture (i.e. all blocks described in Section III B) in MATLAB for two real world applications (regression and classification). For each learning application, we used 4 different experimentally obtained HRS distributions shown in Fig. 2 and Table 1 to model the RRAM synapses.

A. Diabetes Diagnosis (Classification Problem)

The goal of this test was to classify whether a patient is diabetic or not, based on publicly available clinical information (provided in [6]). Table 2, shows the 8 different input features that were used to characterize every patient. A sample size of 576 patients was used for training the network initially. A sample of 192 patients was used for the test mode. The simulated network consists of 8 input nodes (8: features), 180 hidden synapses, 20 hidden neurons, 20 output stage synapses and 2 output neurons.

Table 2, shows the mean classification accuracy and standard deviation of our system over 20 test cycles for all the HRS distributions listed in Table 1. The overall average classification accuracy was found to be 77.7%. We also simulated an ideal ELM, i.e. one without RRAM based synapses [1], to compare the accuracy. In the case of non-RRAM based ELM the accuracy was found to be approximately the same ~77.7%. Directly correlating the μ, σ values of Table 1 and Table 3, remain an on-going activity at this time.

Table 2: Diabetes input feature dataset description with 8 different clinical parameters [6].

Sr. No.	Feature Name	Mean	Std. Deviation
1	Number of times pregnant	3.8	3.4
2	Plasma glucose concentration	120.9	32
3	Diastolic blood pressure	69.1	19.4
4	Triceps skin fold thickness	20.5	16.4
5	2-Hour serum insulin	79.8	115.2
6	Body mass index	32	7.9
7	Diabetes pedigree function	0.5	0.3
8	Age	33.2	11.8

Table 3: Diabetes diagnosis classification performance for our RRAM-ELM architecture performed over 20 test cycles

Device/ (mean HRS value)	Test accuracy (mean)	Test accuracy (Std. Dev.)
Ag/GeS$_2$ CBRAM, μ = 892.86 kΩ	77.64	1.06
HfO$_x$, μ = 25 kΩ	77.79	1.29
HfO$_x$ μ = 222 kΩ	77.69	1.09

HfO_x μ = 2239 kΩ	77.70	0.88
Ideal ELM	77.74	1.23

B. Sinc-Curve fitting (Regression Problem)

The goal of this experiment was to emulate the functionality of a Sinc function generator using the RRAM-ELM architecture. The data provided to the network comprised of uniformly distributed samples of input and output values of an ideal Sinc function over the interval of -10 to 10. Training data size as well as test data size was 5000 sample points. The simulated network consists of 1 input node, 40 hidden synapses, 20 hidden neurons, 20 output stage synapses and 1 output neuron.

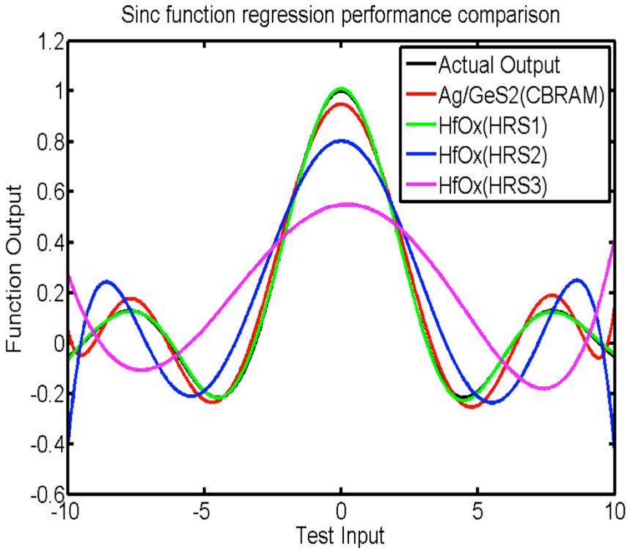

Fig. 6: Sinc Function Regression results obtained using RRAM-ELM architecture.

Table 4: Sinc Regression Performance Comparison for RRAM-ELM.

Device (mean HRS value)	Mean square error
Ag/GeS_2 CBRAM, μ = 892.86 kΩ	0.05
HfO_x, μ = 25 kΩ	0.006
HfO_x μ = 222 kΩ	0.13
HfO_x μ = 2239 kΩ	0.28

As evident from Table 4 and Fig. 6, the OXRAM HRS distribution with μ = 25 kΩ, gave the best Sinc approximation results, with a low MSE of 0.006. The experiment validates the possibility of using RRAM-ELM network for approximating complex functions by using a relatively small, general-purpose circuit.

The two applications simulated in this paper were for proof of concept purpose. In real-world data intensive problems, the memory requirements of ELM training blocks would also increase significantly with number of data points. Having RRAM and CMOS on the same die in such cases would be highly beneficial due to RRAM's high integration density ($4F^2$) and possibility of easy backend 3D integration. Not only RRAM will help to realize random weights/biases, but also in implementing any standard digital memory requirement on board, in an extremely efficient manner. Unlike purely digital ELM implementations the proposed RRAM-ELM also offers non-volatility. In digital ELMs trained weights need to be computed/stored externally in a memory bank and then re-loaded on the system, or the system always needs to be in '*on-state*', thus making it more power hungry. Synaptic programming speeds are also maximized with the use of RRAM as these devices can be switched at nanosecond timescales [20]. Table 5 presents a qualitative comparison of our RRAM-ELM approach with other hardware ELM implementations from literature. From Table 5 it is evident that the proposed RRAM-ELM implementation has the lowest synaptic area requirements.

Table 5: Qualitative comparison of hardware ELM implementations.

Design type [ref]	**Synaptic Area**	**Power/Energy** Analog Vs. Digital	**Speed**
RRAM-ELM (This paper)	**Lowest Area** (1-device per input synapse) ~ $4F^2$	**Low** Train Mode: (Digital) Test Mode: (Analog)	(Ultra-fast Nanosecond timescale switching)
Neuro memristive ELM [4]	**Moderate Area** (Multiple devices used per input synapse ~ N x ($4F^2$))	**Low** Train Mode: (Analog) Test Mode: (Analog)	(Sequential updates)
Silicon transistor based ELM [5]	**Moderate Area** 4 transistors/synapse for hidden layer and counter based summation for output layer	**Medium** Train Mode: (Analog) Test Mode: (Analog)	(Transistors increase overall capacitance on path)
FPGA based ELM [12]	**Large Area** (Memory interface + Adder + MAC unit/digital arithmetic pipeline)	**High** Train and Test Mode: Digital	(Depends completely on digital h/w)

A more fair quantitative comparison is possible only by realizing all the blocks mentioned in Section III B on a hybrid mixed-signal ASIC. Full ASIC realization is an on-going activity and beyond the scope of this paper. Another on going and logical future direction of work is to analyze the correlations between application learning performance and actual device variability parameters.

V. Conclusion

In this paper we outlined a new methodology to design an efficient hardware ELM architecture (RRAM-ELM) exploiting the intrinsic HRS variability of filamentary-RRAM devices. We show that HRS distributions can be used to realize input stage random synaptic weights and random neuron biases. Our approach was validated by two test case applications – classification of suspected diabetes patients and regression/curve fitting of Sinc function. Mean classification accuracy of the system was found to be ~77.7 % while best case regression MSE was found to be 0.006.

REFERENCES

[1] Huang, Guang-Bin, Qin-Yu Zhu, and Chee-Kheong Siew. "Extreme learning machine: theory and applications." *Neurocomputing* 70, no. 1 (2006): 489-501

[2] Shi, Bingxue, Lu Chen, and Chun Lu. "Current controlled sigmoid neural circuit." U.S. Patent 6,664,818, issued December 16, 2003.

[3] Cambria, Erik, Guang-Bin Huang, Liyanaarachchi Lekamalage Chamara Kasun, Hongming Zhou, Chi-Man Vong, Jiarun Lin, Jianping Yin et al. "Extreme Learning Machines." *IEEE Intelligent Systems* 28, no. 6 (2013): 30-59

[4] Merkel, Cory, and Dhireesha Kudithipudi. "A current-mode CMOS/memristor hybrid implementation of an extreme learning machine." In *Proceedings of the 24th edition of the great lakes symposium on VLSI*, pp. 241-242. ACM, 2014.

[5] Basu, Arindam, Sun Shuo, Hongming Zhou, Meng Hiot Lim, and Guang-Bin Huang. "Silicon spiking neurons for hardware implementation of extreme learning machines." *Neurocomputing* 102 (2013): 125-134.

[6] Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[7] Huang, Guang-Bin. "An insight into extreme learning machines: random neurons, random features and kernels." *Cognitive Computation* (2014): 1-15.

[8] Suri, Manan, Damien Querlioz, Olivier Bichler, Giorgio Palma, Elisa Vianello, Dominique Vuillaume, Christian Gamrat, and Barbara DeSalvo. "Bio-inspired stochastic computing using binary CBRAM synapses." *Electron Devices, IEEE Transactions on* 60, no. 7 (2013): 2402-2409.

[9] Yang, Allen Y., Shankar S. Sastry, Arvind Ganesh, and Yi Ma. "Fast ℓ 1-minimization algorithms and an application in robust face recognition: A review." In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pp. 1849-1852. IEEE, 2010.

[10] Huang, Guang-Bin, Qin-Yu Zhu, and Chee-Kheong Siew. "Extreme learning machine: a new learning scheme of feedforward neural networks." In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 2, pp. 985-990. IEEE, 2004.

[11] Irturk, Ali, Bridget Benson, Shahnam Mirzaei, and Ryan Kastner. "An FPGA design space exploration tool for matrix inversion architectures." In *Application Specific Processors, 2008. SASP 2008. Symposium on*, pp. 42-47. IEEE, 2008.

[12] Decherchi, Sergio, Paolo Gastaldo, Alessio Leoncini, and Rodolfo Zunino. "Efficient digital implementation of extreme learning machines for classification." *Circuits and Systems II: Express Briefs, IEEE Transactions on* 59, no. 8 (2012): 496-500.

[13] Raghavan, Nagarajan. "Performance and reliability trade-offs for high-κ RRAM." *Microelectronics Reliability* (2014).

[14] Yu, Shimeng, Ximeng Guan, and H-SP Wong. "On the switching parameter variation of metal oxide RRAM—Part II: Model corroboration and device design strategy." *Electron Devices, IEEE Transactions on* 59, no. 4 (2012): 1183-1188.

[15] Panchapakeshan, P.; Vijayakumar, P.; Narayanan, P.; Chi On Chui; Koren, I.; Moritz, C.A., "3-D integration requirements for hybrid nanoscale-CMOS fabrics," *Nanotechnology (IEEE-NANO), 2011 11th IEEE Conference on* , vol., no., pp.849,853, 15-18 Aug. 2011

[16] Khasanvis, Santosh, K. M. Habib, Mostafizur Rahman, Pritish Narayanan, Roger K. Lake, and Csaba Andras Moritz. "Hybrid Graphene Nanoribbon-CMOS tunneling volatile memory fabric." In Proceedings of the 2011 IEEE/ACM International Symposium on Nanoscale Architectures, pp. 189-195. IEEE Computer Society, 2011.

[17] Alibart, Fabien, Stéphane Pleutin, David Guérin, Christophe Novembre, Stéphane Lenfant, Kamal Lmimouni, Christian Gamrat, and Dominique Vuillaume. "An organic nanoparticle transistor behaving as a biological spiking synapse." Advanced Functional Materials 20, no. 2 (2010): 330-337.

[18] Vincent, A. F., J. Larroque, W. S. Zhao, N. Ben Romdhane, O. Bichler, C. Gamrat, J-O. Klein, S. Galdin-Retailleau, and D. Querlioz. "Spin-transfer torque magnetic memory as a stochastic memristive synapse." In Circuits and Systems (ISCAS), 2014 IEEE International Symposium on, pp. 1074-1077. IEEE, 2014.

[19] Indiveri, Giacomo, Robert Legenstein, George Deligeorgis, and Themistoklis Prodromakis. "Integration of nanoscale memristor synapses in neuromorphic computing architectures." Nanotechnology 24, no. 38 (2013): 384010.

[20] DeSalvo, B.; Sousa, V.; Perniola, L.; Jahan, C.; Maitrejean, S.; Nodin, J-F; Cagli, C.; Jousseaume, V.; Molas, G.; Vianello, E.; Charpin, C.; Jalaguier, E., "Emerging memory technologies: Challenges and opportunities," *VLSI Technology, Systems, and Applications (VLSI-TSA), 2012 International Symposium on* , vol., no., pp.1,2, 23-25 April 2012

[21] Wong, H-SP, Heng-Yuan Lee, Shimeng Yu, Yu-Sheng Chen, Yi Wu, Pang-Shiu Chen, Byoungil Lee, Frederick T. Chen, and Ming-Jinn Tsai. "Metal–oxide RRAM." Proceedings of the IEEE 100, no. 6 (2012): 1951-1970.

[22] C. Gopalan, Y. Ma, T. Gallo, J. Wang, E. Runnion, J. Saenz, F. Koushan, P. Blanchard, and S. Hollmer, "Demonstration of conductive bridging random access memory (CBRAM) in logic CMOS process," Solid-State Electron., vol. 58, no. 1, pp. 54–61, 2011.

[23] Su, Yu-Ting, Kuan-Chang Chang, Ting-Chang Chang, Tsung-Ming Tsai, Rui Zhang, J. C. Lou, Jung-Hui Chen et al. "Characteristics of hafnium oxide resistance random access memory with different setting compliance current." Applied Physics Letters 103, no. 16 (2013): 163502.

[24] Fantini, Andrea, Ludovic Goux, Robin Degraeve, D. J. Wouters, N. Raghavan, G. Kar, Attilio Belmonte, Y-Y. Chen, Bogdan Govoreanu, and Malgorzata Jurczak. "Intrinsic switching variability in HfO 2 RRAM." In Memory Workshop (IMW), 2013 5th IEEE International, pp. 30-33. IEEE, 2013.

[25] Chen, An, and Ming-Ren Lin. "Variability of resistive switching memories and its impact on crossbar array performance." *Reliability Physics Symposium (IRPS), 2011 IEEE International*. IEEE, 2011.

Dr. Manan Suri received his PhD in Nanoelectronics and Nanotechnology from Institut Polytechnique de Grenoble (INPG), France in 2013. He obtained his M.Eng. (2010) and B.S (2009) degrees, in Electrical & Computer Engineering, from Cornell University, USA. Currently he is working as an Assistant Professor with the department of Electrical Engineering at the Indian Institute of Technology- Delhi (IIT-D), India. He worked as a Senior Scientist with NXP Semiconductors, Belgium from 2013-14. His research interests include semiconductor devices, non-volatile memory and bio-inspired computing. He holds 5 US & European patents. He has authored one book chapter and more than 15 papers in reputed international journals and conferences.

Vivek Parmar (S'913) received the B.E. degree in electronics and communication engineering from the S.V.I.T.-Vasad, India, in 2013. He is currently pursuing M.Tech in Electrical Engineering from the Indian Institute of Technology- Delhi (IIT-D), India. His interests include Embedded Systems, VLSI design, and Machine Learning. He currently holds the position of Chapter Treasurer, IEEE CASS-CS, Delhi Section.