SCRAMBLE: A Secure and Configurable, Memristor-Based Neuromorphic Hardware Leveraging 3D Architecture

Nikhil Rangarajan
New York University Abu Dhabi (NYUAD)
nikhil.rangarajan@nyu.edu

Satwik Patnaik
Texas A&M University
satwik.patnaik@tamu.edu

Mohammed Nabeel
NYUAD
mtn2@nyu.edu

Mohammed Ashraf
NYUAD
mohammed.ashraf@nyu.edu

Shubham Rai
TU Dresden
shubham.rai@tu-dresden.de

Gopal Raut
IIT Indore
phd1701102005@iiti.ac.in

Heba Abunahla
Khalifa University (KUSTAR)
heba.abunahla@ku.ac.ae

Baker Mohammad
KUSTAR
baker.mohammad@ku.ac.ae

Santosh Kumar Vishvakarma
IIT Indore
skvishvakarma@iiti.ac.in

Akash Kumar
TU Dresden
akash.kumar@tu-dresden.de

Johann Knechtel
NYUAD
johann@nyu.edu

Ozgur Sinanoglu
NYUAD
ozgursin@nyu.edu

***Abstract*—In this work we present SCRAMBLE, a configurable neuromorphic architecture that provides security against different threats by employing memristors for critical parts and functions. More specifically, we employ memristive memory cells – that are 3D stacked on top of the configurable neuromorphic hardware – to securely hold the weights as well as activation functions of any model processed on the generalized architecture. Thus, programmable memristive cells enable reconfiguration of the architecture to thwart both model stealing and hardware IP stealing attacks. We implement a proof-of-concept for the proposed architecture and analyze its security metrics. We also benchmark it against selected prior art for neuromorphic architectures to quantify the security-performance trade-offs.**

I. Introduction

Hardware neuromorphic systems employing crossbar arrays for vector matrix multiplication (VMM) have proliferated in recent times owing to the advancements in the memristive materials and devices space. The ease of the analog multiply-and-accumulate (MAC) operations in such crossbar arrays, without the need for complex digital circuits, has paved the way for their adoption in various applications ranging from artificial intelligence (AI) and machine learning (ML) to edge computing, imaging, and sensing [1]–[3]. Further, several emerging device-based memristive technologies enable in-memory computing (IMC) to circumvent the *von Neumann* bottleneck, and render modern neuromorphic systems power and performance efficient [4].

Memristor Device Advancements: The early concerns about the endurance, slow operation, and footprint of memristors have been alleviated to a certain extent in the past few years due to the discovery and demonstration of the memristive phenomenon in several new material systems and device topologies [5]. For instance, high performance memristors, with a TiN/AlN/Pt stack, were demonstrated in [6]. These nitride-based memristors exhibit ultra-fast switching speeds ($\sim$85 ps) and low switching currents. An Ag/BaTiO$_3$/Nb:SrTiO$_3$ ferroelectric tunnel junction (FTJ)-based memristor capable of implementing five bits per cell, with a 600 ps switching speed, was showcased in [7]. An ultra-small Pt nanofin-based memristor with 2 nm feature size and a single layer density of 4.5 terabits per square inch was introduced in [8]. Two-dimensional materials have garnered significant attention for nanoelectronic devices due to their interesting electronic and optical properties. For example, the authors in [9] fabricated a 2D hexagonal boron nitride (h-BN)-based memristive crossbar array with Au/h-BN/Au and Ag/h-BN/Ag structures, to achieve zeptojoule switching energy with considerable endurance numbers. A high density memristive crossbar array with 50 nm feature size and 50 ns switching speed was achieved in [10] using p-type van der Waals SnS in an Ag/SnS/Pt structure.

Security Vulnerabilities in Neuromorphic Architectures: Though the promising progress at the device-level has encouraged and accelerated the development of memristor-based neuromorphic systems further, there are still looming concerns over the security vulnerabilities in these emerging architectures as follows (see also Fig. 1(a)).

1) Model replication attacks repeatedly query the neuromorphic circuit and collect a large enough I/O dataset to successfully train a new neural network (NN) model to achieve the same functional behavior as the original design [11]. This attack was impeded in [12] by exploiting the obsolescence effect-induced resistance drift in memristors, however, the proposed defense was shown to be inadequate in the face of input magnitude scaling attacks as well as temperature manipulation [13].

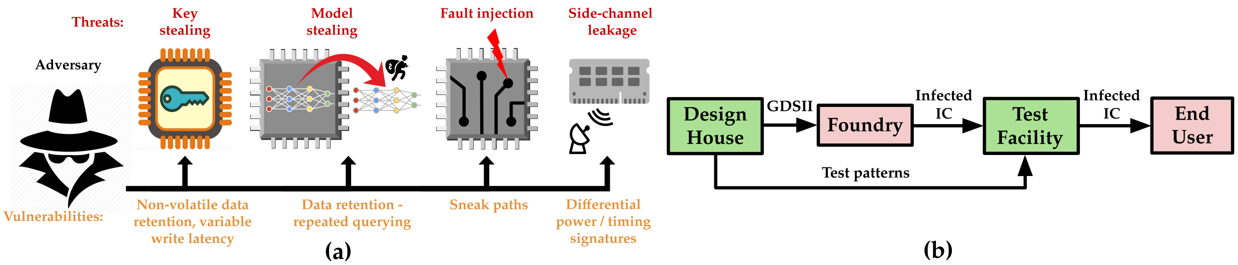

Fig. 1: (a) Typical threat landscape for memristor-based neuromorphic systems. Vulnerabilities are highlighted in orange and their corresponding threats in red. (b) The specific threat model considered for the proposed secure neuromorphic architecture. The foundry and end user (red) are considered untrusted while all other entities are trusted (green).

2) The data retention in non-volatile memristive cells presents a window of opportunity for an attacker to launch cold boot attacks as well as physical probing attacks after powering off the neuromorphic system [14].
3) Memristor crossbar architectures without access transistors for cell gating suffer from the problem of current sneak paths parallel to the intended path [15]. These sneak paths can become a potential attack surface for an attacker to exploit and induce undesired bias and precision loss [16]. This attack leverages the controlled application of input sequences, either externally or through an implanted hardware Trojan (HT), to slowly build up charge along the sneak paths and cause faults, resulting in unintended neuronal firing.
4) The variable write latency in certain memristor crossbar implementations, which is a direct by-product of sneak currents, can result in timing variability in the write operations. This variability can be employed by an attacker to launch data/key retrieval attacks [17].
5) A side-channel attack on the IMC crossbar architecture of memristor-aided logic (MAGIC) [18] was proposed in [19]. This attack leverages the differential power signatures and operating times of the AND and OR arrays in MAGIC.

In this paper, we review prior neuromorphic architectures, discuss their common security vulnerabilities, and then propose a secure 3D neuromorphic solution by amalgamating the benefits of a configurable memristor-based weight array in the top (memory) layer, with a configurable digital activation function circuit in the bottom (logic) layer. Such a programmable 3D arrangement allows us to protect against hardware and model intellectual property (IP) stealing attempts both at fabrication time and runtime. The threat model for the proposed architecture is highlighted in Fig. 1(b).

II. Background

In this section, we briefly discuss the fundamental aspects of the memristor device and then provide an overview of state-of-the-art neuromorphic architectures, along with some security challenges for those.

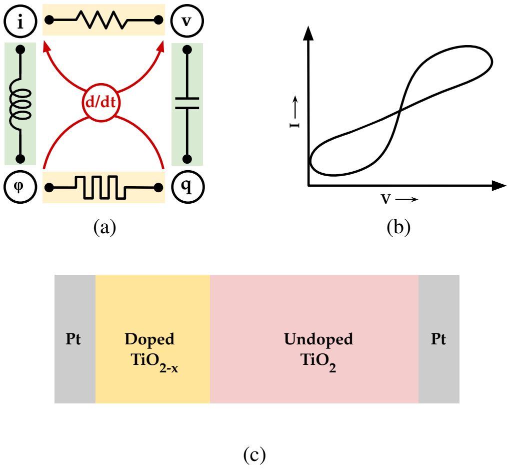

Fig. 2: (a) Memristor as the fourth fundamental circuit element. (b) Theoretical current-voltage characteristics of a bipolar memristor [23]. (c) Memristor structure from its original MIM implementation [21].

A. Memristor: Construction and Working

The memristor, a two-terminal element which connects magnetic flux to electric charge (Fig. 2a), was proposed by *Chua* [20] in 1971 to be the fourth fundamental circuit element, besides RLC elements. The memristive element exhibits a pinched hysteresis behaviour [21] (Fig. 2b), a characteristic that requires at least two equations to describe [22]. Memristors retain their internal resistive state (memristance or memory resistance) according to the history of the applied voltage and current, as described by the following equation.

$$M(q) = \frac{\mathrm{d}\Phi(q)}{\mathrm{d}q} \equiv \frac{v(t)}{i(t)} \tag{1}$$

The first memristor was realized using a metal-insulator-metal (MIM) structure, by sandwiching two metal oxide layers between two metallic electrodes as shown in Fig. 2c. A layer of doped TiO_{2-x} with oxygen vacancies, on an undoped layer of TiO_2, further sandwiched between platinum electrodes, was used by *Williams* for this first memristor implementation [21]. This prototype exhibited all the expected properties as proposed by *Chua* in [20]. Other suitable insulating layers for the

MIM memristor include, e.g., chalcogenides [24] and metal oxides [25].

A memristor can be in one of two possible states, viz. high-resistance state (HRS) and low-resistance state (LRS), an attribute that corresponds to the movement of the dopant ion in the sandwiched metal oxide. Initially, a newly fabricated memristor behaves linearly and requires activation [26]. The SET and RESET operations involve switching of the memristor from HRS to LRS and vice versa, respectively. Mathematical models of memristance ($M(t)$) have been well documented – starting with the linear model, which lacks the boundary behaviour [27], followed by the introduction of window functions that help capture the boundary behavior [28]. A more sophisticated model of the memristance is the exponential model, capable of capturing its highly nonlinear behaviour as well [29]. These models are well studied using SPICE modeling and benchmarked against device specifications [30].

B. Prior Neuromorphic Architectures

First of all, we like to note that prior neuromorphic architectures are vulnerable to one or more attack vectors highlighted in the threat landscape of Fig. 1(a). While CMOS-based designs may fall prey to model replication via repeated querying or advanced side-channel attacks, memristor-based implementations are particularly susceptible to key stealing attacks exploiting non-volatile data retention and variable write latency in the crossbar array.

Next, we review selected CMOS-based neuromorphic architectures.

The reconfigurable on-line learning spiking neuromorphic processor (*ROLLS*) was demonstrated as a low-power neural computing system in [31]. The proposed architecture employs electronic neuromorphic circuits to mimic neuronal physics and emulate the dynamics and learning properties of real neurons and synapses. *ROLLS* used digital components to modify the synaptic and somatic properties, as well as set the topology of the NN. This architecture was developed to implement short-term and long-term plasticity, with the capability of on-line learning for adapting to incoming stimuli.

The spiking neural network architecture (*SpiNNaker*) [32] was developed as a massively parallel neural computing system for handling large neuroscience experiments and simulating various neural algorithms. *SpiNNaker* comprises of a custom multiprocessor with a globally asynchronous, locally synchronous system. It uses asynchronous handshaking protocols to communicate (via analog spikes) between individual systems using a digital bus [33].

The *Neurogrid* mixed-signal multichip [34] takes advantage of the benefits of energy efficiency and sparsity in the analog domain, for performing large neural simulations. Salient features of *Neurogrid* include a) emulation of all neural elements (except the soma) with shared electronic circuits to maximize synaptic connections, b) analog implementation of electronic components (except for axonal arbors) to improve energy efficiency, and c) interconnection of neural arrays in a tree network to optimize the throughput. The axon branching and inter-neural communication is realized using Field-Programmable Gate Arrays (FPGA) and SRAM banks. The *Neurogrid* architecture consists of 16 Neurocores with 1M neurons and billions of synaptic connections.

Another mixed-signal neuromorphic architecture called *Braindrop*, which uniquely leverages the variability, mismatch and heterogeneity in analog/mixed-signal processes, was proposed in [35]. *Braindrop* uses a novel neural engineering framework to exploit process variability-induced mismatch for error-tolerant computation. The heterogeneous neuronal gains and biases, resulting from transistor threshold-voltage mismatch, are forged to form a diverse set of basis functions for function approximation. Further, it uses accumulative thinning, a digitally implemented linear-weighted-sum operation, to reduce communication traffic and sparsify signals.

A programmable digital 3D neuromorphic architecture, the *Neurocube*, was introduced in [36]. This design consists of a logic layer with a 2D mesh network of processing engines (PE) connected to router units. The PE elements are capable of performing MAC operations and come with a dedicated cache. The entire logic layer is integrated in 3D with multiple tiers of high density DRAM. The DRAM tiers are partitioned into cells and concentric cells over multiple DRAM tiers form parallel memory channels called vaults. Using memory-centric neural computing and a programmable neurosequence generator, the *Neurocube* is able to implement a wide range of NN models, similar to a neuromorphic FPGA.

Moving away from traditional CMOS-based neuromorphic architectures, we now briefly review some of the seminal memristive neuromorphic systems in the literature.

One of the first memristive neuromorphic architectures, *CrossNets*, was proposed in 2003 by *Likharev et al.* [37]. It employs a hybrid design, combining regular CMOS technology with self-assembled molecular devices, to create a CMOL (hybrid CMOS/nanowire/molecular) circuit capable of realizing synaptic weighting functionality [33]. *CrossNets* consisted of perpendicular axionic and dendritic nanowires, and a uniform field of bistable two-terminal memristive switches (synapses) at the intersections of these nanowires. The somas were implemented using CMOS-based opamps with a sigmoid-type activation. This architecture was shown to achieve image recognition and classification tasks.

A phase-change memristor (PCM)-based neuromorphic architecture was developed at *IBM* [38] with neurons and synapses realized by leveraging the PCM physical properties and state dynamics. They utilize $Ge_2Sb_2Te_5$ as the phase-change material and employ level-tuned neuronal characteristics to preferentially process input information. This particular implementation was shown to be adept at unsupervised learning and correlation detection.

III. Proposed Architecture

The *SCRAMBLE* architecture consists of two tiers in a monolithic 3D stack as shown in Fig. 3. The top layer (L2) comprises a programmable memristive crossbar array and the

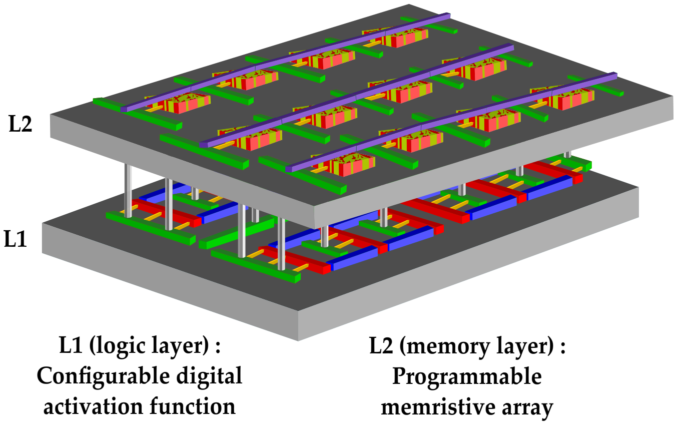

Fig. 3: Representative illustration (not to scale) of the proposed secure 3D neuromorphic architecture with (i) a configurable digital activation function logic in the bottom layer and (ii) a programmable memristive crossbar array in the top layer.

bottom layer (L1) contains a configurable digital activation IP module and peripherals.

A. Programmable Memristor Crossbar Array

For our analysis, we consider a 16×16 array of the Ag/SnS/Pt memristors in [10]. Note that these devices have a 50 nm^2 footprint with 50 ns write/read times. They exhibit conductances in the range of 20-100 μS (linear conductance potentiation), and require read voltages of ~60-80 mV. A single neuron in this array consists of a chain (column) of 16 memristive weights that are multiplied with the corresponding row-read voltages and accumulated via Kirchoff's current addition. Hence, the 16×16 memristor array in L2 consists of 16 neurons, each with their own instance of digital activation function circuitry in L1. To implement NN models with a larger number of hidden layers, more memristive crossbar array tiers can be added to the monolithic 3D stack [39], [40].

B. Configurable CORDIC-based Activation Function

A typical neural computation requires an activation function to model the non-linearity of an application. Popular activation functions used in various learning applications are *sigmoid*, *tanh*, and *Rectified Linear Unit* (ReLU) functions. To implement the proposed architecture, a desirable feature is to have a configurable activation function that provides flexibility to select the activation function as per the application requirement, while affording security at the same time.

Reconfigurable implementation of the architecture considering various activation functions is done using the Coordinate Rotation Digital Computer (CORDIC) algorithm [41], which offers various modes to realizes different functions. Note that CORDIC requires conversion from rectangular to polar coordinates [42] to solve trigonometric relationships. Without loss of generality, we use the CORDIC algorithm in hyperbolic mode to compute either the *sigmoid* or the *tanh* activation function for our proof-of-concept.

The CORDIC algorithm uses an iterative approach for approximate numerical calculations. The number of iterations represents a design-time constraint; it can be revised to, e.g., manage hardware area footprint or increase the throughput of the overall computation. As can be seen in Fig. 4, the CORDIC unit outputs the $\sinh(z)$ or $\cosh(z)$ activation function, where z is the MAC output. Additional adder and divider blocks are required to compute the two activation functions as follows:

$$e^z = sinh(z) + cosh(z) \tag{2}$$

$$tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} = \frac{sinh(z)}{cosh(z)} \tag{3}$$

$$sigmoid(z) = \frac{1}{1 + e^{-z}} = \frac{e^z}{1 + e^z} \tag{4}$$

Equation 2 is computed using a adder. The resulting e^z term is required to further compute both *sigmoid* and *tanh* functions as shown in Equations 3 and 4, respectively. Both equations/activation functions require a divider. More details on the hardware implementation of the CORDIC-based activation functions can be found in [41].

For the reconfigurable overall architecture, the two select lines shown in Fig. 4 are used to switch between the three different activation functions. Note that the ReLU function is added separately to the CORDIC module. As can bee seen in Fig. 3, the logic of the reconfigurable activation functions occupies the lower layer (L1) of the 3D design. The memristor crossbar in the upper layer L2 computes the MAC result, which is then fed to L1 to calculate the activation outputs. It is important to note that the select signals and other constant inputs (X_{in}, Y_{in}) are also stored in L2.

C. Performance and Security Analysis

We implement a proof-of-concept design of *SCRAMBLE* using *Verilog* and synthesize it using *Synopsys DC* for the *TSMC 22nm ULL* library. Note that we leverage the low-power SAR ADC in [43], configured for simple 8-bit encoding of positive floating numbers, for processing the activation function outputs. Also note that, although the memristor array size for the MAC operation is only 16×16, the overall memristor array is much larger, namely 65×16, in order to securely hold the select signals and constants for the CORDIC activation function module.

In Table I, we present a power-performance-area (PPA) comparison of *SCRAMBLE* using a 16×16 MAC array against a similarly sized *Neurocube* design. Note that all metrics are end-to-end, i.e., they consider contributions from the overall 65×16 memristive-array, the ADC, and the activation function module with its peripherals. We observe that *SCRAMBLE* exhibits competitive power and area metrics, but its overall delay is large due to the considerable read time required for memristors as compared to DRAM. However, with ongoing device and materials research for memristors, this read delay is expected to reduce significantly in future.

For a security analysis, we consider the following scenario. An adversary not having access to the weights and configuration stored in the memristor cells – i.e., any adversary during

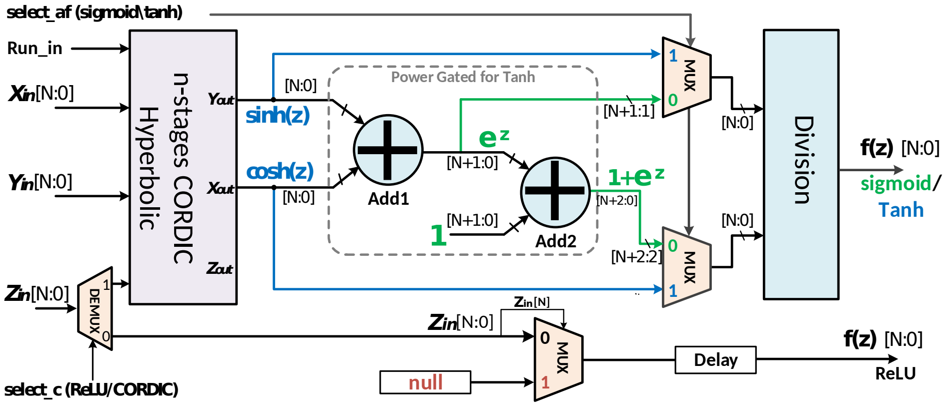

Fig. 4: Configurable CORDIC-based activation function module.

TABLE I: PPA comparison of SCRAMBLE with similarly-sized Neurocube [36].

Neuromorphic architecture	**Power (mW)**	**Delay (ns)**	**Area (mm^2)**
Neurocube (28 nm) [36]	15.6	3.33	0.1936
SCRAMBLE (22 nm)	10.075	52.143	0.032736

the design and manufacturing, or an adversary in the field without probing capabilities – tries to steal the design IP and mimic the functional behaviour of *SCRAMBLE*.

For that, Fig. 5 provides box-plots for Hamming Distance (HD)-represented mismatch in functional behaviour for NNs processing on different memristor array sizes of illegitimate *SCRAMBLE* instances as follows. First, three golden conductance/weight matrices of sizes 16×16, 32×32, and 64×64 for some exemplary but arbitrary neural functions are initialized. Then, we launch a random-guessing attack, effectively randomizing the conductance values as well as the selection of the activation function implemented by the CORDIC module. This exercise is repeated 1,000 times for each array size, and we report the corresponding HD values in Fig. 5. Here, we observe that the corruptibility offered by the reconfigurable memristor array and activation functions is ∼30–60%, which implies good security against foundry-based and end-user adversaries.

Going beyond our proof-of-concept here, note that advanced memristor implementations like [44], which are capable of reconfiguration between drift and diffusive modes, can further aid the security and protection of proprietary weights by transforming a non-volatile weight array to a volatile array that is susceptible to memristor obsolescence and weight drift.

Further, the tiered 3D structure of the *SCRAMBLE* architecture naturally lends itself to split manufacturing [45], [46] – memristors and peripherals are implemented in upper metal layers, separated from the logic in the lower layers. Thus, the full design of the architecture can be protected against foundry-based adversaries. For the memristor crossbar array structure in this work, attacks based on spatial correlation between active components and their interconnects [47], [48] are not of concern. This is because, to an adversary having access only to the lower digital layer, any of the incomplete interconnects only point to regular array structures without revealing the actual configuration/programming of the array.

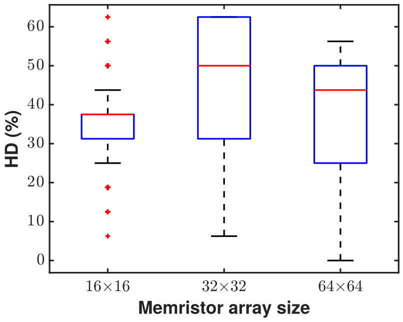

Fig. 5: Hamming distance (HD) for mismatch in functional behaviour of NNs running on illegitimate instances of *SCRAMBLE*, for different memristor array sizes.

IV. Conclusion

Threats against neuromorphic computing systems are of particular interest nowadays, owing to the increased popularity and use of such systems for various critical applications. Neural networks for specialized tasks can take a lot of resources to train and, as such, their internal IP can be considered proprietary and worth protecting against model and IP stealing attacks. In this work, we demonstrate *SCRAMBLE*, a secure and configurable memristor-based neuromorphic architecture to thwart attacks launched by adversaries at the foundry or regular end-users. We envision a two-tiered 3D implementation for the proposed *SCRAMBLE* architecture, with a programmable memristive array holding the weights and configuration settings in the top layer, and a configurable digital activation-function module in the bottom layer. This way we can reconfigure, transform and thus protect the entire neural network that runs in *SCRAMBLE*. We quantify *SCRAMBLE*'s security (HD for corruptibility) and PPA, and contrast the latter against a state-of-the-art CMOS-based reconfigurable neuromorphic architecture.

References

[1] B. Sun *et al.*, “Synaptic devices based neuromorphic computing applications in artificial intelligence,” *Materials Today Physics*, vol. 18, p. 100393, 2021.

[2] P. Date *et al.*, “Efficient classification of supercomputer failures using neuromorphic computing,” in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 242–249.

[3] O. Krestinskaya *et al.*, “Neuromemristive circuits for edge computing: A review,” *IEEE transactions on neural networks and learning systems*, vol. 31, no. 1, pp. 4–23, 2019.

[4] C. Li *et al.*, “In-memory computing with memristor arrays,” in *2018 IEEE International Memory Workshop (IMW)*. IEEE, 2018, pp. 1–4.

[5] Y. Li *et al.*, “Review of memristor devices in neuromorphic computing: materials sciences and device challenges,” *Journal of Physics D: Applied Physics*, vol. 51, no. 50, p. 503002, 2018.

[6] B. J. Choi *et al.*, “High-speed and low-energy nitride memristors,” *Advanced Functional Materials*, vol. 26, no. 29, pp. 5290–5296, 2016.

[7] C. Ma *et al.*, “Sub-nanosecond memristor based on ferroelectric tunnel junction,” *Nature communications*, vol. 11, no. 1, pp. 1–9, 2020.

[8] S. Pi *et al.*, “Memristor crossbar arrays with 6-nm half-pitch and 2-nm critical dimension,” *Nature nanotechnology*, vol. 14, no. 1, pp. 35–39, 2019.

[9] S. Chen *et al.*, “Wafer-scale integration of two-dimensional materials in high-density memristive crossbar arrays for artificial neural networks,” *Nature Electronics*, vol. 3, no. 10, pp. 638–645, 2020.

[10] X. F. Lu *et al.*, “Exploring low power and ultrafast memristor on p-type van der Waals SnS,” *Nano Letters*, vol. 21, no. 20, pp. 8800–8807, 2021.

[11] B. Liu *et al.*, “Security of neuromorphic systems: Challenges and solutions,” in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2016, pp. 1326–1329.

[12] C. Yang *et al.*, “Security of neuromorphic computing: thwarting learning attacks using memristor’s obsolescence effect,” in *Proceedings of the 35th International Conference on Computer-Aided Design*, 2016, pp. 1–6.

[13] D. Rajasekharan *et al.*, “SCANet: Securing the weights with superparamagnetic-MTJ crossbar array networks,” *IEEE transactions on neural networks and learning systems*, 2021.

[14] X. Pan *et al.*, “Nvcool: When non-volatile caches meet cold boot attacks,” in *2018 IEEE 36th International Conference on Computer Design (ICCD)*. IEEE, 2018, pp. 439–448.

[15] M. A. Zidan *et al.*, “Memristor-based memory: The sneak paths problem and solutions,” *Microelectronics journal*, vol. 44, no. 2, pp. 176–183, 2013.

[16] R. JS *et al.*, “Neuromorphic security,” in *Emerging Topics in Hardware Security*. Springer, 2021, pp. 257–279.

[17] V. R. Kommareddy *et al.*, “Are crossbar memories secure? new security vulnerabilities in crossbar memories,” *IEEE Computer Architecture Letters*, vol. 18, no. 2, pp. 174–177, 2019.

[18] S. Kvatinsky *et al.*, “MAGICmemristor-aided logic,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 895–899, 2014.

[19] S. S. Ensan *et al.*, “SCARE: Side channel attack on in-memory computing for reverse engineering,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 12, pp. 2040–2051, 2021.

[20] L. Chua, “Memristor-the missing circuit element,” *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, Sep. 1971.

[21] R. S. Williams, “How we found the missing memristor,” *IEEE Spectrum*, vol. 45, no. 12, pp. 28–35, Dec 2008.

[22] B. Mohammad *et al.*, “Robust hybrid memristor-CMOS memory: Modeling and design,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 11, pp. 2069–2079, Nov 2013.

[23] N. R. McDonald *et al.*, “Analysis of nonpolar resistive switching exhibited by Al/Cu$_x$O/Cu memristive devices created via room temperature plasma oxidation,” in *2011 International Semiconductor Device Research Symposium (ISDRS)*, Dec 2011, pp. 1–2.

[24] A. S. Oblea *et al.*, “Silver chalcogenide based memristor devices,” in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, July 2010, pp. 1–3.

[25] L. Goux *et al.*, “Coexistence of the bipolar and unipolar resistive-switching modes in NiO cells made by thermal oxidation of Ni layers,” *Journal of Applied Physics*, vol. 107, no. 2, p. 024512, 2010. [Online]. Available: https://doi.org/10.1063/1.3275426

[26] J. Rajendran *et al.*, “An approach to tolerate process related variations in memristor-based applications,” in *2011 24th Internatioal Conference on VLSI Design*, Jan 2011, pp. 18–23.

[27] Y. N. Joglekar *et al.*, “The elusive memristor: properties of basic electrical circuits,” *European Journal of Physics*, vol. 30, no. 4, p. 661675, May 2009. [Online]. Available: http://dx.doi.org/10.1088/0143-0807/30/4/001

[28] Z. Biolek *et al.*, “SPICE model of memristor with nonlinear dopant drift.” *Radioengineering*, vol. 18, no. 2, 2009.

[29] D. B. Strukov *et al.*, “Exponential ionic drift: fast switching and low volatility of thin-film memristors,” *Applied Physics A*, vol. 94, pp. 515–519, 2009.

[30] S. Benderli *et al.*, “On SPICE macromodelling of TiO$_2$ memristors,” *Electronics Letters*, vol. 45, no. 7, pp. 377–379, March 2009.

[31] N. Qiao *et al.*, “A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses,” *Frontiers in neuroscience*, vol. 9, p. 141, 2015.

[32] E. Painkras *et al.*, “SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation,” *IEEE Journal of Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, 2013.

[33] R. A. Nawrocki *et al.*, “A mini review of neuromorphic architectures and implementations,” *IEEE Transactions on Electron Devices*, vol. 63, no. 10, pp. 3819–3829, 2016.

[34] B. V. Benjamin *et al.*, “Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations,” *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699–716, 2014.

[35] A. Neckar *et al.*, “Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model,” *Proceedings of the IEEE*, vol. 107, no. 1, pp. 144–164, 2018.

[36] D. Kim *et al.*, “Neurocube: A programmable digital neuromorphic architecture with high-density 3D memory,” *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 380–392, 2016.

[37] K. Likharev *et al.*, “CrossNets: High-performance neuromorphic architectures for CMOL circuits,” *Annals of the New York Academy of Sciences*, vol. 1006, no. 1, pp. 146–163, 2003.

[38] A. Pantazi *et al.*, “All-memristive neuromorphic computing with level-tuned neurons,” *Nanotechnology*, vol. 27, no. 35, p. 355205, 2016.

[39] K.-T. T. Cheng *et al.*, “3D CMOS-memristor hybrid circuits: Devices, integration, architecture, and applications,” in *Proceedings of the 2012 ACM International Symposium on International Symposium on Physical Design*, ser. ISPD ’12. New York, NY, USA: Association for Computing Machinery, 2012, p. 3340. [Online]. Available: https://doi.org/10.1145/2160916.2160925

[40] H. An *et al.*, “Opportunities and challenges on nanoscale 3D neuromorphic computing system,” in *2017 IEEE International Symposium on Electromagnetic Compatibility Signal/Power Integrity (EMCSI)*, 2017, pp. 416–421.

[41] G. Raut *et al.*, “A CORDIC based configurable activation function for ann applications,” in *2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2020, pp. 78–83.

[42] J. E. Volder, “The CORDIC trigonometric computing technique,” *IRE Transactions on Electronic Computers*, vol. EC-8, no. 3, pp. 330–334, 1959.

[43] D. Cordova *et al.*, “A 0.8 V 875 MS/s 7b low-power SAR ADC for ADC-based wireline receivers in 22nm FDSOI,” in *2020 IFIP/IEEE 28th International Conference on Very Large Scale Integration (VLSI-SOC)*. IEEE, 2020, pp. 52–57.

[44] R. A. John *et al.*, “Reconfigurable halide perovskite nanocrystal memristors for neuromorphic computing,” *Nature Communications*, vol. 13, no. 1, pp. 1–10, 2022.

[45] C. McCants. (2016) Trusted integrated chips (TIC) program. [Online]. Available: https://www.ndia.org/-/media/sites/ndia/meetings-and-events/divisions/systems-engineering/past-events/trusted-micro/2016-august/mccants-carl.ashx

[46] K. Vaidyanathan *et al.*, “Building trusted ICs using split fabrication,” in *2014 IEEE international symposium on hardware-oriented security and trust (HOST)*. IEEE, 2014, pp. 1–6.

[47] H. Li *et al.*, “Deep learning analysis for split-manufactured layouts with routing perturbation,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 10, pp. 1995–2008, 2020.

[48] J. Magaña *et al.*, “Are proximity attacks a threat to the security of split manufacturing of integrated circuits?” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 12, pp. 3406–3419, 2017.