

Ultra-Compact, Scalable, Energy-Efficient VO_2 Insulator-Metal-Transition Oxide Based Spiking Neurons for Liquid State Machines

Samiran Ganguly, Nikhil Shukla, Avik W. Ghosh

Charles L. Brown Dept. of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA, USA

{sganguly, ns6pf, ag7rq} @virginia.edu

Abstract—Spiking neural networks, inspired by biological neural systems, could process immense volumes of spatio-temporal data by representing them as spikes. Here, we propose to implement compact, scalable, energy-efficient spiking neurons based on the unique insulator-metal transition in Vanadium dioxide (VO_2) which interact through memristive synapses, to emulate a Liquid State Machine (LSM). Further, we demonstrate the implementation of this recurrent neural network as a temporal auto-encoder, and adaptive channel equalizer for application in neuromorphic signal processing. Our approach provides a pathway to reduce component count ($50 - 100X$) and improve energy efficiency ($> 50X$) over conventional CMOS based implementations.

Index Terms—Insulator-Metal-Transition, Spiking Neurons, Liquid State Machine, Reservoir Computing, Extended Kalman Filter

I. INTRODUCTION

High-performance computing has historically developed around the von Neumann architecture and the Boolean computing paradigm, executed on Silicon Complementary Metal Oxide Semiconductor (CMOS) hardware. Software development model has for decades has worked around the CMOS fabric that has singularly dictated our choice of materials, devices, circuits and architecture to build our technology. Over the last decade however, Moore's law for hardware scaling has significantly slowed down with its end predicted soon, primarily due to the prohibitive energy cost of computing with increasingly smaller devices. At the same time, software development has taken off with Machine Learning and Artificial Intelligence dominating the roost. This has given rise to the notion of 'neuromorphic computing' which promises to radically transform this model of computing to match the ultimate computer in existence, the brain.

It is well known that the temporal dynamics of biological neurons encode immense volumes of processing power [1], [2] in extremely small footprint: $\sim 10^{12}$ neurons and $\sim 10^{15}$ synapses or neural interconnection in a volume of $\sim 1200cm^3$, and power consumption: $10 - 15W$. It has been argued that spike encoded information (spike interval/density modulation) can be energy-efficient as information is encoded in the dead times between sparse spikes [3]–[5], and their phase sequencing [6] and time multiplexing [7] in reservoir computing.

978-1-7281-3915-9/19/\$31.00 © 2019 IEEE

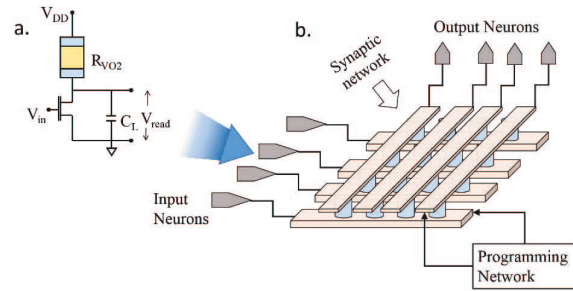


Fig. 1. a. The proposed hardware unit of the spiking neuron built from a 1T-1R pull-up pull-down configuration of a VO_2 layer and an n-channel MOSFET. b. Large scale neural networks can be fabricated with spiking neurons by coupling them with dense cross-bar arrays of programmable memristors forming a synaptic network between various layers of a neural network.

Spiking networks can have higher representation power and computational capacity compared to other neural networks [8]–[10], and are shown to be optimally entropy-efficient (maximal # of bits/Joule) [11], [12] in biological neurons. Commercial silicon spiking nets like IBM's TrueNorth chip [13]–[15] are seen to be $\sim 800X$ more energy efficient than silicon multiprocessors, although lacking integrated on-chip training. The true challenge with CMOS only hardware substrate is to find a physical mechanism to initiate, stabilize and terminate the spikes in a controllable way to leverage the advantages of scalable networks of spiking neurons, with small energy-area footprint, in particular for low size, weight and power (SWaP) applications.

In this work we propose to use Vanadium Dioxide's (VO_2) unique electrical volatile characteristics arising from the Insulator-Metal transition (IMT) property to build ultra-compact and highly energy-efficient hardware units that behaves as spiking neuron (fig. 1a) that can encode information in the spike intervals/frequency and when coupled with programmable memristive synapses, say in a cross-bar array arrangement (fig. 1b), can directly implement dense spiking recurrent neural networks in hardware. In this work we specifically illustrate the networks of these spiking neurons to build a Liquid State Machine (LSMs), the spiking version of a

Reservoir Computer. We then show that these LSMs built from spiking neurons can perform non-linear signal processing tasks such as an adaptive channel equalizer, and learn to predict a complex time series running blind, i.e. without an external input. We then conclude by discussing the relative reduction in component count and power consumption provided by these hardware units over a purely CMOS implementations of such spiking neural networks.

II. VO_2 INSULATOR-METAL-TRANSITION: ULTRA-COMPACT SPIKING NEURON

A. Insulator-Metal-Transition, VO_2 Characteristics and Controllable Spiking Oscillator

VO_2 shows a sharp Insulator Metal Phase Transition (IMT) [16]–[19] when driven thermally, optically and electrically (fig. 2a). It has been shown that IMT at 340K shows a typical ON-OFF ratio $\sim 1-5 \times 10^3$ to a max of 10^5 , with a switching speed $\sim 10ns$ electronically, $\sim 10fs$ optically, and measured switching reliability over 10^9 cycles [20]. The underlying mechanism is still in debate and seems to be a combination of purely electronic Mott transition (abrupt opening of a Coulomb gap) and a simultaneous structural change between monoclinic and tetragonal/rutile configurations [21]. The main feature of its output characteristics is a unique volatile hysteresis loop (fig. 2a,b) with two stable states (solid horizontal arms) and two unstable states (dashed vertical arms) which is distinct from pinched hysteresis memristor curves used as selectors (e.g. Ag/HfO_2), or magnetic hysteresis curves used in MRAM technology where both the stable states are accessible at $V \sim 0$ point of the characteristics.

This volatile hysteresis has its center shifted from $V = 0$ to half the transition voltages V_H, V_L which proves quite important for our application. This characteristics allows the device to move naturally from an oscillatory spiking to a non-spiking state and vice-versa, simply by adjusting a load line using an external transistor in the hardware unit as shown in the schematic fig. 2b. Fig. 3c,d show experimental data and simulations on the device whose characteristics are presented in fig. 2a, showing the non-spiking and spiking behavior and is obtained by a slight shift in the load line (determined by the resistance of the transistor).

When the transistor load line crosses the stable high resistance state section of the I-V curve, the IMT neuron remains in the resting state (solid blue line). However as gate voltage V_{in} increases the transistor load-line (solid orange line) now periodically crosses both the unstable (black dashed lines) arms, and the neuron fires and generates an oscillatory spike train, as can be seen in fig. 2e. Due to stochastic cycle-to-cycle variations in V_{IMT} , firing occurs with probability $p < 1$ leading to probabilistic spiking of the IMT neuron. Further, the spiking probability can be modulated using the gate voltage, V_{in} , which decides the fraction of the voltage range of V_T intersected by the transistor load line.

This unique volatile characteristic generating spike trains, along with in-built activation and deactivation (much like a real biological neuron) is naturally suited to provide ultra-compact

and efficient, controlled spiking behavior. Additionally, the built-in stochasticity of the phase transition in VO_2 naturally provides controllable noise in the device. It is well recognized that noise is an important feature in building robust neural network models by preventing over-fitting to training data and can provide a hardware mechanism to build a natural ‘echo-state property’ feature useful in recurrent neural networks like Reservoir Computing as we discuss later.

B. Spike Encoder and Spiking Stochastic Leaky-Integrate-and-Fire Neuron

We can build an efficient spike-interval/density modulator using the proposed VO_2 unit as shown in fig. 3a. The input signal is scaled appropriately using an impedance divider circuit to reflect the movement of the load-line between the stable high resistance state to the oscillatory state using the gate voltage applied on the input MOSFET. The simulation presented in fig. 3b shows the normalized output of the circuit measured across the load capacitor. A “high” signal, in this scheme, shifts the load-line to the oscillatory stage by reducing the transistor resistance and the unit keeps firing, till the input gate voltage reduces shifting the load-line back to the stable high R state, causing the spikes to end. Therefore, a simple $1R - 1T$ unit can generate an efficient and compact spike modulator based on the intrinsic electrical characteristics of the device itself.

The same unit can also be used to build a spiking neuron, where the input signal gets integrated over time in the gate capacitor and if it reaches a threshold, the neuron fires and generates spikes. Our proposed unit naturally embodies a spike generator based on the input signal at the input terminal. Instead of a smooth analog signal from some sensor (as in case of the spike encoder), if we provide spiking signals using a resistive-capacitive metallic interconnect to the input terminal, the gate capacitor can integrate the charge provided by these spikes naturally over a few characteristic RC time constants of the interconnect + gate capacitor and can move the unit from non-spiking to spiking state. Being a volatile device, this built-up charge will leak away. Therefore if the input spike train is stopped, the unit will move back to a non-spiking state. This illustrates the principle of the same spike encoder effectively working as a leaky-integrate-and-fire (LIF) neurons. Additionally, the inherent stochastic nature of the transition voltage V_H and V_L discussed previously, changes the spiking voltage/charge threshold creating inherent stochasticity in the activation/spike generation.

III. LIQUID STATE MACHINES FROM VO_2 SPIKING NEURONS

A. Dynamical Recurrent Neural Networks: Reservoir Computing

Reservoir Computing [22] is a biologically inspired model of computing where a set of non-linear units or neurons are assembled together in recurrent fashion, i.e. with bidirectional feedbacks between various neurons (fig. 4a). To this “reservoir” of neurons a temporal signal is imposed, giving rise

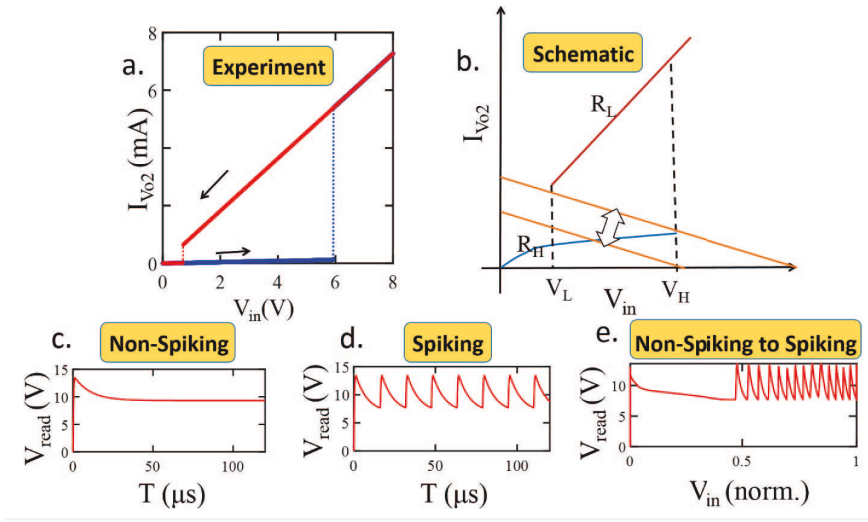


Fig. 2. a. Experimental characteristics of the a VO_2 film showing its shifted volatile hysteresis I-V curve. b. Schematic of the I-V curve to illustrate the utilization of a controllable load-line to go from non-spiking to spiking states. c. Non-spiking behavior of the neuron. d. Spiking behavior of the neuron. e. Movement from non-spiking to spiking behavior by moving the load-line using the gate input gate voltage to the device. It can be seen that spike frequency or spike density increases with the increasing input voltage.

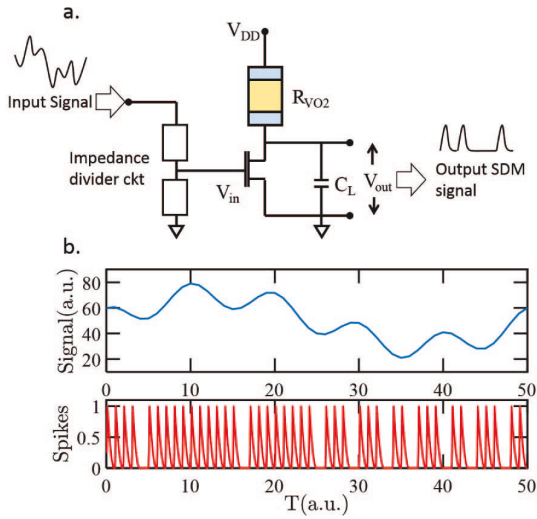


Fig. 3. a. Spike encoder using the proposed unit along with an impedance divider circuit at the input to appropriately scale the input voltage to control the movement of the load-line created by the transistor. b. Example of a spike interval/density encoding of an arbitrary input signal.

to collective excitations. An inference is made by harvesting or sampling these collective reservoir states using a readout layer which can be a linear or logistic readout, or even a deep network. In this model of computing, the reservoir is never trained - only the readout layer(s) is trained (Fig. 2b). This model of computing also allows attaching different readouts to the same reservoir to carry out different inferencing tasks, embodying the inherent parallelism of biological neural

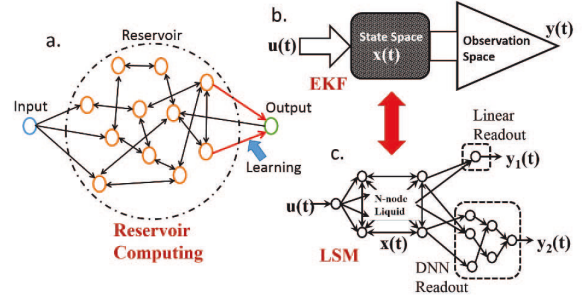


Fig. 4. a. A schematic of a reservoir computing. b. Comparison between the mathematical model of an Extended Kalman Filter (EKF) and a Liquid State Machine (LSM).

systems (fig. 4c). Reservoir computing is particularly suited to spatio-temporal inferencing tasks due to the recurrence, as the signature of an input signal provided to the reservoir continues to persist in the reservoir for a certain time period (also called the echo-state property) which allows the readout to harvest different time samples of the same input signal by reading from different nodes at the same time. Reservoir computing is presented in two different forms in the literature: if the neurons are logistic and the signal analog, it is called an Echo State Network (ESN) [23], whereas if the neurons are spiking and signals are encoded in spike domain, the model is called a Liquid State Machine (LSM) [24]–[26].

B. Reservoir Computing and Signal Processing

Kalman filters [27], [28] are a class of adaptive filters that can be used to learn and predict the states of a dynamical system from their histories and extract independent variables

from dependent variables of state of such a system. A Kalman filter consists of a dynamical system composing a “state space”, and an “observer system” that samples the state space to generate an estimate or an inference. Being a dynamical system, the state and therefore the observations are correlated in time (fig. 4b). In a regular Kalman filter, the state space is a linear function of the input signal and the observation is made by sampling the state of the dynamical system to yield the observed signal. Extended Kalman Filters (EKF) extend conventional Kalman filters to nonlinear signal processing. EKFs are the de-facto standard in GPS tracking, navigation systems and nonlinear state estimation theory.

It can be seen that there are strong parallels between Reservoir computing and EKFs. The reservoir corresponds to the non-linear dynamical state space, while the readout corresponds to the observer (fig. 4b,c). ESNs have been presented in literature [29]–[31] as an approach to train EKFs composed of logistic neurons, whereas LSMs were developed as a plausible model of biological cognitive systems where extremely rich cognitive tasks can be carried out with rather small sets of neurons and limited localized learning. This obvious connection makes VO_2 based Liquid State Machines work as hardware embodiment of the mathematical model of an EKF to perform signal processing and control tasks. Compactness and energy efficiency is achieved by directly implementing the reservoir in the dynamics of the network of spiking neurons and programmable synapses built from memristive interconnects at the readout layer. Our proposed unit and their networks embed the paradigm of spiking neurons and EKF inherently in their physics and make them particularly suited for signal processing tasks as we show next.

IV. TEMPORAL PROCESSING TASKS

In this section we discuss two example temporal data/signal processing tasks using the LSMs by leveraging their connection to EKFs. These examples show that it is possible to implement highly compact and energy-efficient signal processors using small spiking neural networks built from VO_2 neurons.

A. Temporal Predictor

An autoencoder learns the underlying representation or generating function of a set of a data. When applied to a time series data it can be useful in predictive tasks such as navigation and control. In this mode the encoder predicts the next few steps of the sequence which can be used to carry out the navigation and control task. Best in class systems for such tasks are EKFs which can be used to predict the “next step” for a moving object given a history of the past and the current trajectory. Therefore, EKFs can work as temporal autoencoder, i.e. an encoder for time-series data. We implement a temporal auto-encoder using a VO_2 LSM based EKF, shown in fig. 5a. In this task we attempt to learn and reproduce this signal blindly, i.e. once it is learned we want to test if we can generate the “next steps” purely from the LSMs self generated output and how well this predicted signal matches with the original signal generator.

A periodic and subtly chaotic source is provided to the system as the input signal. The input signal is first converted into the spike domain and then provided to a 20-node LSM. A linear weighted sampler is used to readout the reservoir states and thereafter demodulated to generate the output signal. After harvesting a large number of reservoir states, we adjust the sampler weight using a linear regression technique as popular in the reservoir computing community.

During the testing phase, we disconnect the source and connect the readout of the LSM back to its own input, providing the input signal for the next time-step. Therefore, in this testing step the LSM tries to keep generating the source signal blindly looking at its own output, without any corrective feedback or knowledge of the testing signal, generated from the same source that was used during the training phase. Our simulations show that the LSM can do this predictive task reasonably well for 10-15 steps for the given function, however it starts to diverge as the number of steps increase without a corrective feedback (fig. 5b,c). This is to be expected due to the finite memory capacity of the LSM. Even excellent cognitive machines like humans cannot predict such trajectories indefinitely.

During our simulations, we found that extremely small reservoirs (# of nodes = 1-5) find it difficult to learn complex signals, however as the network size is increased the accuracy begins to improve as the state space of the LSM expands with the size of the network allowing it to remember more complex ‘high frequency’ components of the input signal.

B. MMSE Channel Equalizer

A communication channel can inherently inject all kinds of non-linear distortions, inter symbol interference, and noise in a signal. The task of equalization is to recover the underlying “clear” signal from the distorted signal. An equalizer can be constructed in multiple ways depending on how they are supposed to be used. We implement an EKF based MMSE (minimum mean square error) equalizer using VO_2 LSM which attempts to reconstruct a signal which has been distorted by a telecommunication media (fig. 5d).

In this task the LSM tries to generate output $y(t)$ that is as close to input $d(t)$ from the distorted signal $u(t)$ as possible. The original signal $d(t)$ passes through a medium which introduces inter-symbol interference, non-linear distortions and noise to generate the distorted signal $u(t)$. This distorted signal is then converted into the spike domain using VO_2 spike modulator, which is then provided to a 20 node LSM built from assembling together VO_2 neurons. Then we generate a weighted sample of the reservoir states, which is then demodulated to recover the original signal. During the training phase, we adjust the weighted sampling matrix using linear regression. After successful training, we test the equalizer which can now recover the original signal from the distorted one successfully as shown in fig. 5e. In our simulations we found that we could achieve a symbol recovery rate between 85 – 100% depending on the degree of distortion and LSM network size.

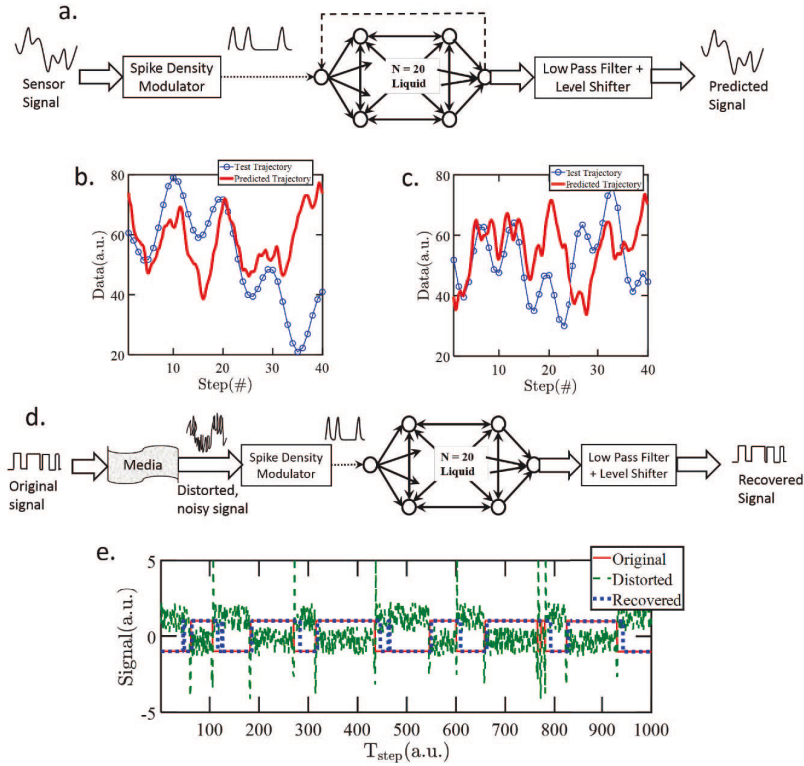


Fig. 5. a. Temporal Predictor setup using a VO_2 LSM. The input signal is spike encoded and provided to the reservoir and readout using a linear sampler and decoder built from low pass filter and level shifter. During the testing phase, the reservoir's own output is fed back to the input and the encoder is decoupled from reservoir. b. & c. Two example simulations of blind signal prediction task. The network reproduces the signal reasonably well for first 15-20 steps but diverges after that without corrective feedback. d. Channel Equalizer setup using a VO_2 LSM. The LSM attempts to reverse the effect of the distortion created by the media on an input signal. e. Original, distorted, and recovered signal obtained from the equalizer shown together.

V. VO_2 SPIKING NEURONS VS. BOOLEAN CMOS NEURONS: A COMPARATIVE ASSESSMENT

Spiking Neural Networks like LSMs can be emulated numerically on any Turing Machine, including a Boolean CMOS processor. However, the hardware primitives of such a computer do not match the algorithmic model of these networks. A network composed of the VO_2 based hardware unit directly implements a reservoir, where the collective dynamics of VO_2 switching itself gives rise to reservoir states being used for the computation. Therefore, the compactness, energy efficiency, and scalability that can be obtained from these units can never be matched, even in principle, by a Boolean emulator. Our proposed unit works in the analog domain as a controllable spike generator and is composed of just one transistor with VO_2 grown directly on the drain terminal so its area cost of just 1 transistor. The power dissipation and energy-delay product of this unit is comparable to a CMOS inverter due to high ON-OFF ratio, gain, and steep sub-threshold swing of the device and this has been previously demonstrated experimentally in a similar structure called the HyperFET [32]. Therefore, the energy cost of a single self-contained VO_2 spiking neuron is equivalent to the smallest

unit of computing in CMOS circuits. Additionally, since the signals are represented as spikes in time, it is an inherently analog mode of computation where a signal can be carried by just 1 wire, instead of multiple wires as required in a Boolean implementation, which further reduces the area and energy cost significantly.

As an illustrative example, consider the work referenced in [33] as an example of a liquid state machine implementation using a stochastic Boolean accelerator in a ring-like reservoir topology, demonstrating a similar temporal prediction task as this work, required ≈ 13 logical elements, ≈ 13 combinational functions, and ≈ 6 logic registers per neuron. While without knowing the exact detailed design of the FPGA used it is difficult to estimate the total component count, an efficient design presented in another work [34], demonstrating similar signal processing tasks as this work, required 4 4-LUT structures per neuron and a simple count will suggest this translates to 420 CMOS inverters or 840 MOSFETs. Therefore, it can be seen that *a Boolean implementation can easily require $\sim 500X$ more components over the presented unit to implement a spiking neural network*. Even assuming great leaps in designs of Boolean accelerators for neural networks

in future, we can safely promise at least 50 – 100X reduction in component count. Considering that the dissipation in VO_2 neurons is similar to a CMOS inverter, we expect energy-efficiency to scale approximately linearly with the component count improvement, i.e. $\sim 500X$ in the best case scenario for VO_2 neurons or $\sim 50 - 100X$ in best case scenario for CMOS implementations. In [35] it was found that the VO_2 neurons had $\sim 30X$ decrease in power dissipation over an equivalent CMOS implementation in a classification task. This strongly supports our lower bound estimates for energy-efficiency improvement of VO_2 spiking neurons over CMOS implementations.

It should be noted that the estimates provided here are preliminary and cursory. One of the future directions will be extending these benchmarks to a wider set of implementations of LSMs and Kalman filters in FPGAs and ASICs, as well to other material systems, such as low energy-barrier magnets which have also been used to implement spiking neurons. The central goal of this paper is to demonstrate, using proof-of-concept simulations, that ultra-compact, energy-efficient LSMs can be fabricated directly from IMT spiking neurons, and elicit research and exploration interest in the same from the community.

REFERENCES

- [1] D. V. Buonomano and M. M. Merzenich, "Temporal information transformed into a spatial code by a neural network with realistic properties," *Science*, vol. 267, no. 5200, pp. 1028–1030, 1995.
- [2] W. Maass and H. Markram, "On the computational power of circuits of spiking neurons," *Journal of computer and system sciences*, vol. 69, no. 4, pp. 593–616, 2004.
- [3] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of physiology*, vol. 117, no. 4, pp. 500–544, 1952.
- [4] R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane," *Biophysical journal*, vol. 1, no. 6, pp. 445–466, 1961.
- [5] J. Nagumo, S. Arimoto, and S. Yoshizawa, "An active pulse transmission line simulating nerve axon," *Proceedings of the IRE*, vol. 50, no. 10, pp. 2061–2070, 1962.
- [6] G. Buzsáki and A. Draguhn, "Neuronal oscillations in cortical networks," *science*, vol. 304, no. 5679, pp. 1926–1929, 2004.
- [7] D. Marković, N. Leroux, M. Riou, F. Abreu Araujo, J. Torrejon, D. Querlioz, A. Fukushima, S. Yuasa, J. Trastoy, and P. Bortolotti, "Reservoir computing with the frequency, phase, and amplitude of spin-torque nano-oscillators," *Applied Physics Letters*, vol. 114, no. 1, p. 012409, 2019.
- [8] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural networks*, vol. 10, no. 9, pp. 1659–1671, 1997.
- [9] W. Maass, "Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons," *Advances in Neural Information Processing Systems*, vol. 9, pp. 211–217, 1997.
- [10] R. Legenstein and W. Maass, "What makes a dynamical system computationally powerful," *New directions in statistical signal processing: From systems to brain*, pp. 127–154, 2007.
- [11] P. Crotty and W. B. Levy, "Energy-efficient interspike interval codes," *Neurocomputing*, vol. 65, pp. 371–378, 2005.
- [12] W. B. Levy and R. A. Baxter, "Energy efficient neural codes," *Neural computation*, vol. 8, no. 3, pp. 531–543, 1996.
- [13] D. S. Modha, "Introducing a brain-inspired computer," *Published online at <http://www.research.ibm.com/articles/brain-chip.shtml>*, 2017.
- [14] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, and Y. Nakamura, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.
- [15] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, and G.-J. Nam, "Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 10, pp. 1537–1557, 2015.
- [16] N. Shukla, A. Parihar, E. Freeman, H. Paik, G. Stone, V. Narayanan, H. Wen, Z. Cai, V. Gopalan, and R. Engel-Herbert, "Synchronized charge oscillations in correlated electron systems," *Scientific reports*, vol. 4, p. 4964, 2014.
- [17] M. Brahlek, L. Zhang, J. Lapano, H.-T. Zhang, R. Engel-Herbert, N. Shukla, S. Datta, H. Paik, and D. G. Schlom, "Opportunities in vanadium-based strongly correlated electron systems," *MRS Communications*, vol. 7, no. 1, pp. 27–52, 2017.
- [18] F. J. Morin, "Oxides which show a metal-to-insulator transition at the Neel temperature," *Physical review letters*, vol. 3, no. 1, p. 34, 1959.
- [19] Z. Yang, C. Ko, and S. Ramanathan, "Oxide electronics utilizing ultrafast metal-insulator transitions," *Annual Review of Materials Research*, vol. 41, pp. 337–367, 2011.
- [20] L. A. Ladd and W. Paul, "Optical and transport properties of high quality crystals of V_2O_4 near the metallic transition temperature," *Solid State Communications*, vol. 7, no. 4, pp. 425–428, 1969.
- [21] H.-T. Kim, B.-G. Chae, D.-H. Youn, S.-L. Maeng, G. Kim, K.-Y. Kang, and Y.-S. Lim, "Mechanism and observation of Mott transition in VO_2 -based two- and three-terminal devices," *New Journal of Physics*, vol. 6, no. 1, p. 52, 2004.
- [22] B. Schrauwen, D. Verstraeten, and J. Van Campenhout, "An overview of reservoir computing: theory, applications and implementations," in *Proceedings of the 15th european symposium on artificial neural networks*, p. 471–482 2007, 2007, pp. 471–482.
- [23] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, no. 34, p. 13, 2001.
- [24] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.
- [25] —, "Computational models for generic cortical microcircuits," *Computational neuroscience: A comprehensive approach*, vol. 18, pp. 575–605, 2004.
- [26] W. Maass, P. Joshi, and E. D. Sontag, "Computational aspects of feedback in neural circuits," *PLoS computational biology*, vol. 3, no. 1, p. e165, 2007.
- [27] S. Haykin, *Kalman filtering and neural networks*. John Wiley & Sons, 2004, vol. 47.
- [28] M. S. Grewal, *Kalman filtering*. Springer, 2011.
- [29] H. Jaeger, *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach*. GMD-Forschungszentrum Informationstechnik Bonn, 2002, vol. 5.
- [30] —, "Adaptive nonlinear system identification with echo state networks," in *Advances in neural information processing systems*, 2003, pp. 609–616.
- [31] C. Sheng, J. Zhao, H. Leung, and W. Wang, "Extended kalman filter based echo state network for time series prediction using mapreduce framework," in *2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks*. IEEE, 2013, pp. 175–180.
- [32] N. Shukla, A. V. Thathachary, A. Agrawal, H. Paik, A. Aziz, D. G. Schlom, S. K. Gupta, R. Engel-Herbert, and S. Datta, "A steep-slope transistor based on abrupt electronic phase transition," *Nature communications*, vol. 6, p. 7812, 2015.
- [33] M. L. Alomar, V. Canals, A. Morro, A. Oliver, and J. L. Rossello, "Stochastic hardware implementation of Liquid State Machines," in *2016 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2016, pp. 1128–1133, ISSN: 2161-4407.
- [34] B. Schrauwen, M. D'Haene, D. Verstraeten, and J. V. Campenhout, "Compact hardware liquid state machines on FPGA for real-time speech recognition," *Neural Networks*, vol. 21, no. 2, pp. 511–523, Mar. 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0893608007002353>
- [35] M. Jerry, A. Parihar, B. Grisafe, A. Raychowdhury, and S. Datta, "Ultra-low power probabilistic IMT neurons for stochastic sampling machines," in *2017 Symposium on VLSI Technology*, Jun. 2017, pp. T186–T187, ISSN: 2158-9682.