8-2018

Efficient Neuromorphic Computing Enabled by Spin-Transfer Torque: Devices, Circuits and Systems

Abhronil Sengupta
Purdue University

EFFICIENT NEUROMORPHIC COMPUTING ENABLED BY SPIN-TRANSFER TORQUE: DEVICES, CIRCUITS AND SYSTEMS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Abhronil Sengupta

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2018

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Kaushik Roy, Chair

School of Electrical and Computer Engineering

Dr. Anand Raghunathan

School of Electrical and Computer Engineering

Dr. Vijay Raghunathan

School of Electrical and Computer Engineering

Dr. Byunghoo Jung

School of Electrical and Computer Engineering

Approved by:

Dr. Venkataramanan Balakrishnan

Head of the School Graduate Program

Dedicated to my parents

without whose love and support none of this would have been possible

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere thanks and gratitude to my Ph.D. advisor, Prof. Kaushik Roy, whose mentorship and support was instrumental in making this dissertation come to reality. His guidance and broad viewpoint on the implications and significance of a research project has shaped my outlook as a researcher and helped me formulate various research projects during my Ph.D. I would like to thank Prof. Roy again for being ever-excited about my research and presenting my work at various research conferences and meetings. I am greatly indebted to him for the critical feedback he has always provided regarding my work and my presentation style, for helping shape my academic career and for being a source of constant help, support and guidance to choose my career path ahead.

Next, I would like to thank the members of my doctoral dissertation committee: Prof. Anand Raghunathan, Prof. Byunghoo Jung and Prof. Vijay Raghunathan. Their invaluable feedback have helped me to shape the approach and direction of my dissertation research at various times. I would also like to thank Prof. Amit Konar and Prof. Chayanika Bose from my undergraduate institution Jadavpur University, India for instilling my interest in machine learning and electronic devices respectively. My research on spintronics enabled neuromorphic computing, lying at the intersection of these two fields, would not have been possible without their guidance. Many thanks to Prof. Konar for inspiring and assisting me to undertake graduate studies.

I am extremely fortunate to have performed two industrial summer research internships directly related to my dissertation at Facebook Reality Labs and Intel Labs. I would like to thank my internship managers Dr. Chiao Liu (Facebook Reality Labs) and Dr. Ram Krishnamurthy (Intel Labs) for guiding the internship projects, for providing critical feedback and for strongly supporting my internship project and my academic career even after my internship. I will also take this opportunity to thank

my internship mentors who helped make my internship projects successful: Dr. Yuting Ye and Dr. Robert Wang (Facebook Reality Labs); Dr. Gregory K. Chen, Dr. Raghavan Kumar, Dr. Huseyin Ekin Sumbul and Dr. Phil Knag (Intel Labs).

This work would not have been possible without the help and support of my seniors and fellow members of the Nanoelectronics Research Lab at Purdue University. I would like to thank my senior and good friend Dr. Sri Harsha Choday (Intel Labs) who helped me to start off my Ph.D. research. He served as an amazing mentor and a collaborator. I would also like to acknowledge the tremendous help and support I have received from Dr. Xuanyao Fong (National University of Singapore) and Dr. Yusung Kim (Intel Corporation). They have helped shape my fundamentals on spintronic device physics. I also want to thank my collaborators throughout my Ph.D. for their assistance and valuable inputs. I would also take this opportunity to thank members of the Nanoelectronics Research Lab who made life at lab and at Purdue fun: Dr. Karthik Yogendra (IBM), Dr. Woo-Suhl Cho (Apple), Aparajita Banerjee (Intel), Akhilesh Jaiswal, Aayush Ankit, Indranil Chakraborty, Deboleena Roy, Chamika Mihiranga Liyanagedera, Parami Wijesinghe and others whom I may have mistakenly not mentioned here.

A lot of my research has been enabled by work performed by various colleagues and research labs around the world and I would like to acknowledge them here as well. My spintronic device proposals were enabled by domain wall motion based experiments in magnetic materials performed by Prof. Geoffrey Beach's group at MIT and Prof. Sayeef Salahuddin's group at UC Berkeley. I would also like to thank Dr. Peter U. Diehl (ETH Zurich) for helping us setup his code framework on Spiking Neural Networks.

I would also like to acknowledge funding support for my various research projects: Center for Spintronic Materials, Interfaces, and Novel Architectures (C-SPIN), a MARCO and DARPA sponsored StarNet center, the Semiconductor Research Corporation, the National Science Foundation, Intel Corporation, the US DoD Vannevar Bush Faculty Fellowship and the Center for Brain-inspired Computing Enabling

Autonomous Intelligence (C-BRIC) - a Joint University Microelectronics Program (JUMP) center.

Finally and most importantly, I would like to express my greatest thanks to my parents who have been my pillars of strength throughout. I am forever indebted to them for their love, support, sacrifices and encouragement.

I would like to thank all my well-wishers and everyone who have directly or indirectly helped me through my Ph.D. journey and I hope I am able to live up to everyone's expectations in my career ahead.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

ABSTRACT

Sengupta, Abhronil Ph.D., Purdue University, August 2018. Efficient Neuromorphic Computing Enabled by Spin-Transfer Torque: Devices, Circuits and Systems. Major Professor: Kaushik Roy.

Present day computers expend orders of magnitude more computational resources to perform various cognitive and perception related tasks that humans routinely perform everyday. This has recently resulted in a seismic shift in the field of computation where research efforts are being directed to develop a neurocomputer that attempts to mimic the human brain by nanoelectronic components and thereby harness its efficiency in recognition problems. Bridging the gap between neuroscience and nanoelectronics, this thesis demonstrates the encoding of biological neural and synaptic functionalities in the underlying physics of electron spin. Description of various spin-transfer torque mechanisms that can be potentially utilized for realizing neuro-mimetic device structures is provided. A cross-layer perspective extending from the device to the circuit and system level is presented to envision the design of an All-Spin neuromorphic processor enabled with on-chip learning functionalities. Device-circuit-algorithm co-simulation framework calibrated to experimental results suggest that such All-Spin neuromorphic systems can potentially achieve almost two orders of magnitude energy improvement in comparison to state-of-the-art CMOS implementations.

1. INTRODUCTION

Although the brain is not yet fully understood, neuromorphic computing that attempts to emulate some facets of its functionalities and inter-connectivity, are becoming increasingly popular on machine learning tasks, and are surpassing humans at multiple cognitive tasks more than ever before. For instance, recently Google DeepMind beat a professional human champion at a 19×19 Go board game [1]. The key inspiration behind the development of algorithms and computing paradigms with high degree of bio-fidelity is driven by the expectation that by emulating some attributes of the human brain, we would be able to approach the brain's highly efficient and low-power cognitive abilities. For instance, implementation of bio-realistic "spiking" neural computing paradigms have recently enabled low-power event-driven neuromorphic hardware equipped with on-chip local spike-timing dependent synaptic learning functionalities.

While these neuro-inspired computing models are still implemented in von-Neumann architectures consisting of Boolean logic and memory circuits, the brain's "computing fabric" is highly parallel, interconnected and enabled with in-situ synaptic memory storage. Further CMOS transistors, that form the underpinnings of current computing systems, are on-off switches that are naturally suited for Boolean computing but may not inherently map to the "computational primitives" of neuro-mimetic algorithms. Limited by this mismatch between the computational units and the underlying hardware, CMOS based neuromorphic architectures consume resources and power that are orders of magnitude higher than that involved in the biological brain [2]. Bridging this gap necessitates the exploration of devices, circuits and architectures that provide a better match to biological processing and which require a significant rethinking of traditional von-Neumann based computing.

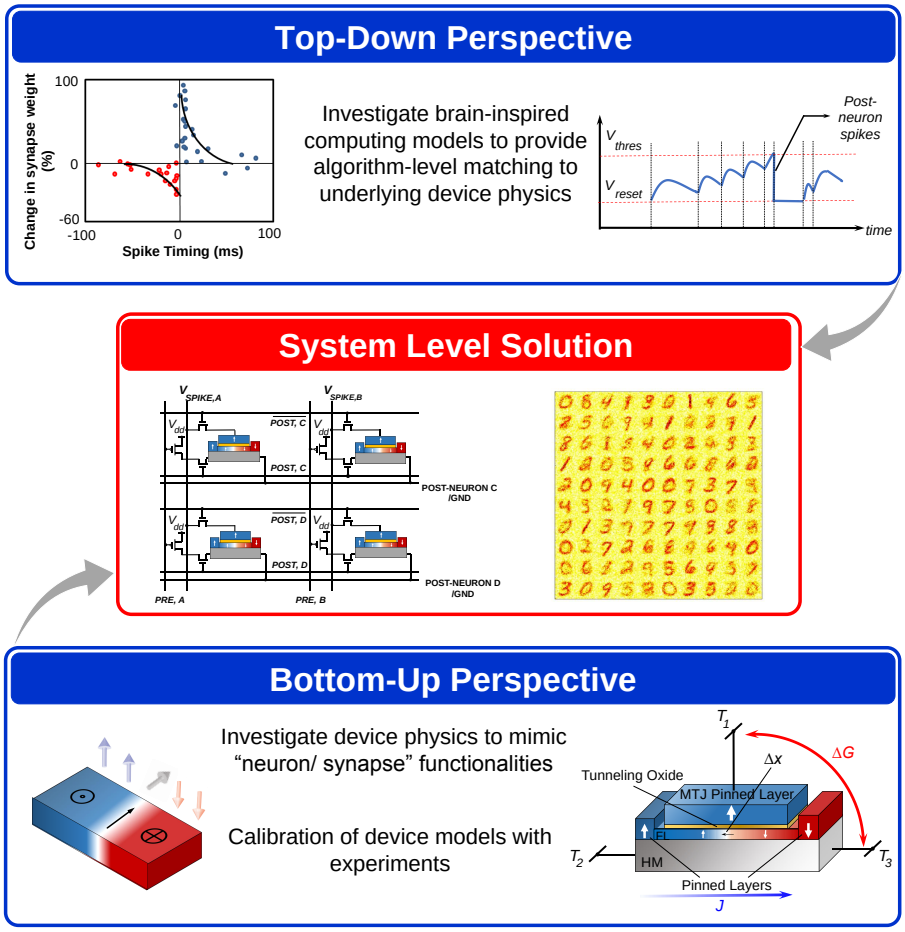

Fig. 1.1. Cross-layer research effort across the stack of materials, devices, circuits and algorithms to provide system-level solutions for enabling cognitive intelligence. A "top-down" perspective to provide algorithm-level matching to the underlying device physics of spintronic devices is complemented by a "bottom-up" approach where recent experiments in spintronics are leveraged to propose device structures that can directly mimic neural and synaptic functionalities.

While usage of spintronic devices in memory applications have achieved maturity and is close to the market [3], recent experiments in domain wall motion based devices [4, 5] and probabilistic switching characteristics of scaled nanomagnets [6, 7] are revealing immense possibilities of implementing a plethora of neural and synaptic functionalities by single spintronic device structures that can be operated at very low

terminal voltages. Simple engineering of the device dimensions or biasing region of the operating transistors can enable the emulation of functionalities that can range from neuron spiking behavior to synaptic learning abilities in the same magnetic stack. While other emerging devices such as resistive memories have also been explored for neuromorphic computing, they are limited by the variety of neural or synaptic functionalities that they can emulate along with high energy requirements for programming [8, 9] (which is an essential component of learning and neural inference). The prospect of large improvements in integration density and energy consumption and concurrently providing in-memory computing possibilities (due to their inherent non-volatility) can potentially make spintronic devices a promising path towards realizing "brain-like" nanoelectronic computing. This thesis attempts to provide a multi-disciplinary perspective across the entire stack of materials, devices, circuits, systems and algorithms where understanding of the underlying device physics of spintronic devices ("bottom-up approach") is complemented by efforts to adapt neuromorphic computing models to the unique characteristics of spintronic devices ("top-down approach") to construct cognitive networks of interconnected spintronic neural and synaptic components (Fig. 1.1) [10, 11].

2. SPINTRONIC DEVICES: UNDERLYING PHYSICAL PHENOMENA

Several spintronic device structures have been proposed in literature to mimic different neuronal or synaptic functionalities. However, in order to understand the mapping of biological functions to the operation of such spin devices, an understanding of the underlying physical phenomena is necessary. This section provides a brief overview of major spin-torque effects in nanomagnets that can be engineered to realize such neuromimetic computations.

The two main physical phenomena that are exploited to construct neuromimetic spin devices are the spin-torque effect ("write" mechanism) and the Tunneling Magneto-Resistance or the TMR effect ("read" mechanism). The manipulation of magnetization state without the assistance of any external magnetic field through spin-transfer torque effect was first predicted by Slonczewski [12] and Berger [13] in 1996. Several experiments demonstrating spin-transfer torque induced magnetization reversal have been demonstrated henceforth [14–16]. On the other hand, sensing the magnetization state through the TMR effect was first experimentally observed by Julliére in 1975 in Fe/Ge-O/Co stacks [17].

2.1 Device Fundamentals

A nanomagnet is characterized by two collinear but oppositely directed stable magnetization directions, termed as the "easy" axis, such that in the absence of any external perturbation (magnetic field or input spin current) the magnetization would relax to either of the stable magnetization states. The stability of the magnet in

the presence of thermal noise is maintained by virtue of a barrier height, E_B, that is determined by the uniaxial anisotropy, K_{u2}, of the magnet as [18],

$$E_B = K_{u2}V \tag{2.1}$$

where V is the volume of the magnet. The lifetime of the magnet in absence of thermal agitation is related exponentially to the magnitude of the barrier height. For instance, a barrier height of $40k_BT$ (k_B is Boltzmann constant) ensures a magnet lifetime of ~ 7.4 years [18].

The uniaxial anisotropy of the magnet, and hence the direction of magnet "easy-axis", can be in-plane (IMA) when shape anisotropy dominates the resultant anisotropy of the magnet [3, 19, 20]. In this case, the magnet cross-sectional area would be an ellipse with the "easy-axis" being in the direction of the longer dimension. In contrast, in perpendicular magnetic anisotropy (PMA) materials, the magnetocrystalline anisotropy dominates over the shape anisotropy in order to make the out-of-plane direction as the "easy-axis" direction [3, 21, 22]. Hence, PMA magnets are usually of circular cross-sectional area.

In order to read the magnetization state of the nanomagnet, a Vertical Spin Valve (VSV) structure is utilized as shown in Fig. 2.1(a). It is referred to as the Magnetic Tunnel Junction (MTJ) [16, 17, 23] where a thin oxide acts as the tunneling barrier between two nanomagnets. The resistance of the MTJ depends on the relative orientation of the magnetization directions of the two nanomagnets. In order to provide a reference, the magnetization of one of the magnets is pinned to a particular direction (usually achieved by coupling to an antiferromagnetic layer), $\widehat{\mathbf{m}}_P$, while the magnetization of the other layer, $\widehat{\mathbf{m}}$, can be determined by the resistance of the MTJ stack. The two layers are referred to as the "pinned" layer (PL) and "free" layer (FL) respectively. The difference in resistance of the MTJ with relative magnetic orientations of the FL and PL can be explained from the concept of "spin-filtering" [3, 24]. When $\widehat{\mathbf{m}}_P$ and $\widehat{\mathbf{m}}$ are parallel to each other (Parallel configuration: P), electrons with that corresponding spin orientation can easily tunnel through the oxide since the filled

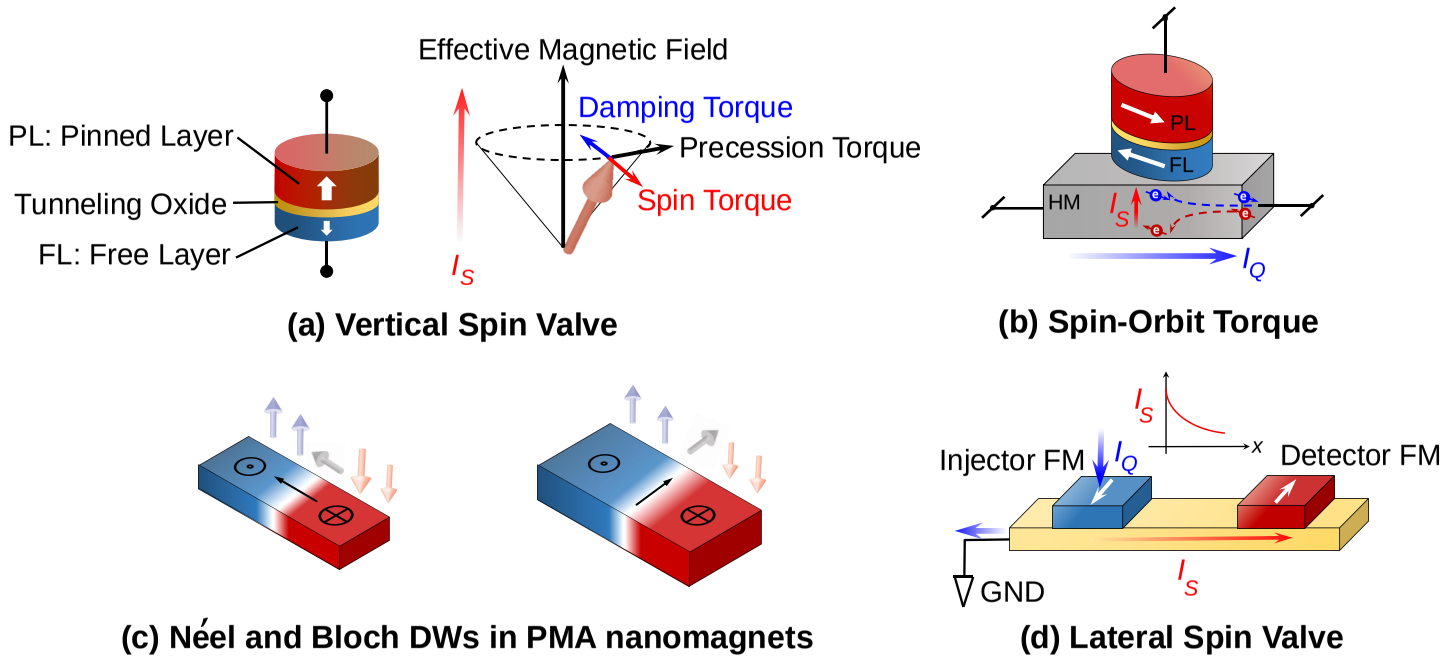

Fig. 2.1. (a) Vertical Spin Valve: A Magnetic Tunnel Junction consists of two ferromagnets, namely the "free" layer (FL) and the "pinned" layer (PL) separated by a tunneling oxide barrier. The magnetization dynamics evolves under the influence of the damping torque, precession torque and spin-torque due to an input spin current, I_S, (b) Néel and Bloch domain walls (DWs) observed in narrow and wider nanostrips with Perpendicular Magnetic Anisotropy (PMA) respectively, (c) Spin-orbit torque is generated on a nanomagnet due to charge current flow (I_Q) through an underlying Heavy Metal (HM) layer due to spin-Hall effect, (d) Lateral Spin Valve based structure where an injector and detector ferromagnet are located on top of a non-magnetic channel. The detector ferromagnet can be switched due to non-local spin-torque effect exerted by charge current flowing through the injector magnet to the ground contact lying beneath the magnet. The magnitude of the injected spin current, I_S, reduces exponentially with the distance between the injector and detector FMs.

states in the band structure of one contact corresponding to that particular spin orientation is well matched to empty states for the same spin in the other contact. On the contrary, when $\widehat{\mathbf{m}}_P$ and $\widehat{\mathbf{m}}$ are oppositely directed (Anti-Parallel configuration: AP), the band structures of either spin configuration are not well-matched for the two contacts, thereby resulting in higher resistance. The metric utilized to measure the difference between the P (R_P) and AP (R_{AP}) MTJ resistances is referred to as the Tunneling Magnetoresistance Ratio (TMR) defined as,

$$TMR = \frac{R_{AP} - R_P}{R_P} \times 100\% \tag{2.2}$$

It is worth noting here that the MTJ P and AP resistances are a function of the oxide thickness and applied voltage across the MTJ which can be formulated using the Non-Equilibrium Green's Function based transport simulation framework [25]. Considering that the FM has a uniform magnetization direction, the MTJ resistance (R) is a function of the spacer (MgO) thickness (t_{MgO}), relative angle between the magnetizations of the FM and the pinned layer (θ), and the voltage across the MTJ (V_{MTJ}). The variation can be described by the following equations [25],

$$R \propto \quad e^{a_0 t_{MgO} + b_0} + \sum_{m=1}^{c} \left((-1)^{m-1} V_{MTJ}^{2m} e^{a_m t_{MgO} + b_m} \right)^{-d} \tag{2.3}$$

$$R(\theta) = \quad \frac{1}{R_P} \left(\cos\left(\frac{\theta}{2}\right) \right)^2 + \frac{1}{R_{AP}} \left(\sin\left(\frac{\theta}{2}\right) \right)^2 \Bigg)^{-1} \tag{2.4}$$

Here, R_P and R_{AP} represent the parallel ($\theta = 0$) and anti-parallel resistances ($\theta = \pi$) of the MTJ respectively. The fitting parameters a_m, b_m, c and d can be determined by calibrating the simulation framework with experimental data. For an extensive description of the NEGF based simulation framework, readers are referred to Ref. [25].

The discussion so far has been limited to sensing the magnetization state of a nanomagnet. Let us now discuss the mechanism of manipulating the magnetization direction of a magnet. One of the most common mechanisms is by passing a charge current through the MTJ stack due to spin-transfer torque effect [12–16]. When charge current flows from the FL to the PL, electrons are injected into the FL from

the PL that are spin-polarized in the direction of $\widehat{\mathbf{m}}_P$. The magnitude of injected spin current is determined by the polarization of the magnet. Hence the injected spins attempt to orient the FL in the direction of $\widehat{\mathbf{m}}_P$. For a sufficient magnitude of the current flowing from the FL to the PL, the MTJ is switched to the P configuration. On the other hand, when current flows from the PL to the FL, the FL attempts to inject spins into the PL. However, due to "spin-filtering", only electrons with spin parallel to $\widehat{\mathbf{m}}_P$ can tunnel easily to the PL from the FL. Hence the remaining spins anti-parallel to $\widehat{\mathbf{m}}_P$ remain in the FL and exert a torque to orient the MTJ in the AP state.

The temporal evolution of magnetization dynamics can be described by Landau-Lifshitz-Gilbert equation [26] with additional terms to account for the effect of spin-transfer torque [27] as follows,

$$\frac{d\widehat{\mathbf{m}}}{dt} = -\gamma(\widehat{\mathbf{m}} \times \mathbf{H}_{eff}) + \alpha(\widehat{\mathbf{m}} \times \frac{d\widehat{\mathbf{m}}}{dt}) + \frac{1}{qN_s}(\widehat{\mathbf{m}} \times \mathbf{I}_s \times \widehat{\mathbf{m}}) \tag{2.5}$$

where $\widehat{\mathbf{m}}$ is the unit vector of FL magnetization, $\gamma = \frac{2\mu_B\mu_0}{\hbar}$ is the gyromagnetic ratio for electron, α is Gilbert's damping ratio, $\mathbf{H}_{eff}$ is the effective magnetic field, $N_s = \frac{M_s V}{\mu_B}$ is the number of spins in free layer of volume V (M_s is saturation magnetization and μ_B is Bohr magneton), and $\mathbf{I}_s$ is the input spin current generated by the HM underlayer. Thermal noise is included by an additional thermal field [28], $\mathbf{H}_{thermal} = \sqrt{\frac{\alpha}{1+\alpha^2}\frac{2k_B T_K}{\gamma\mu_0 M_s V \delta_t}}G_{0,1}$, where $G_{0,1}$ is a Gaussian distribution with zero mean and unit standard deviation, k_B is Boltzmann constant, T_K is the temperature and δ_t is the simulation time-step.

In the absence of any input current stimulus, the magnet is subjected to a field-torque (that causes it to precess in the direction of the effective magnetic field) and a damping torque (that attempts to stabilize the magnet along the initial equilibrium state). The effective magnetic field includes any external applied field, magnetic uniaxial anisotropy field along with a thermal fluctuation field [28, 29] that lends a stochastic behavior to the switching process. The impact of input current on the magnetization dynamics is usually described by a Slonczewski-like torque [27] that acts in the plane of the damping torque and stabilizes the magnet along either of

the two stable magnetization directions depending on the direction of the input spin current. Although some experiments have reported contributions from a field-like torque to the resultant spin-torque due to the input current [30], its magnitude is usually much less in comparison to the Slonczewski-like torque in tunneling junctions.

2.2 Domain Wall Motion

Mono-domain magnets where the entire FL magnetization is uniformly polarized can represent only two binary states. More than two states can be represented by multi-domain magnets that are fabricated with elongated shape to stabilize a transition region (termed as domain wall, DW) between two regions of opposite magnetic polarizations. The device state can be then represented by the position of the DW or the relative proportion of the two oppositely polarized magnetic domains. The manner of magnetization transition at the DW location depends on the anisotropy and shape of the magnet. While IMA nanowires are characterized by transverse (thin and narrow nanostrips) or vortex DWs (wider and thicker nanostrips) [31], PMA materials exhibit Néel (narrow nanostrips) or Bloch DWs (wider nanostrips) [32]. Due to lower switching current requirements, we will consider PMA nanomagnets in this text. Fig. 2.1(c) depicts the magnetic orientations of Néel and Bloch DWs observed in PMA magnetic strips. The domain wall is termed as a Néel wall when the magnetization direction at the wall location rotates in a plane perpendicular to the plane of the wall and is typically observed for nanowires with width less than $100nm$ (owning to shape anisotropy) [33]. For wider nanowires, the wall magnetization rotates in the plane of the wall and is termed as the Bloch wall [33]. Charge current flowing through the magnetic strip can displace the domain wall in the direction of electron flow due to STT effect. Current induced DW motion in the direction of electron flow was predicted [34] and also observed in multiple experiments [35,36]. DW motion due to charge current flow through the magnet can be attributed to spin-torque generated due to local magnetization tracking of electrons flowing through the magnet.

2.3 Spin-Orbit Torque

Spin current generated by STT effect is always limited by the polarization strength of the injector magnet. Recent experiments on Insulator-Ferromagnet-Heavy Metal (I-FM-HM) multilayer structures have opened up the possibility of much greater spin injection efficiencies due to strong spin-orbit interaction (SOI) [37] observed in such multilayer structures. When a charge current flows through the underlying HM, spin-orbit torque (SOT) is generated at the FM-HM interface. Although the cause of SOT can be attributed to two possible origins, namely the Rashba field due to structural inversion asymmetry [38] and the spin-Hall effect (SHE) [39], we will consider SHE to be the dominant underlying physical phenomena for this text. As shown in Fig. 2.1(c), due to the flow of charge current through the HM, electrons with opposite spins scatter on the top and bottom surfaces of the HM. The spin-polarization is orthogonal to both the directions of charge current and injected spin current. These electrons experience spin-scattering repeatedly while traveling through the HM and thereby transfer multiple units of spin angular momentum to the FM lying on top. The magnitude of injected spin current density (J_s) is proportional to the magnitude of input charge current density (J_q), with the proportionality factor being defined as the spin-Hall angle [39] ($\theta_{SH} < 1$). Hence, the input charge to spin current conversion is governed by the following relation,

$$I_s = \theta_{SH}. \left(\frac{W_{FM}}{t_{HM}}\right) I_q \tag{2.6}$$

where I_s and I_q are the input spin current and charge current magnitudes respectively, W_{FM} is the width of the FM lying on top of the HM, and t_{HM} is the HM thickness. By ensuring $W_{FM} >> t_{HM}$, high spin injection efficiencies greater than 100% ($I_s > I_q$) can be achieved. Typical HMs with high spin-orbit coupling under exploration are Pt, β-W and β-Ta. An important point to note is that the injected spins at the FM-HM interface have in-plane spin polarization due to SHE. Hence, SOT induced magnetization reversal is only possible for IMA magnets while an external magnetic field is required to switch PMA magnets in presence of SOT [40–42].

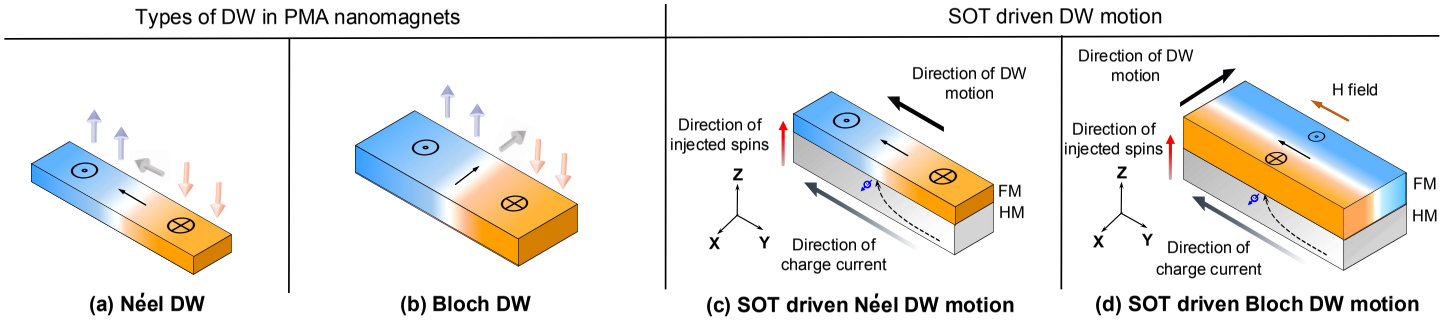

Fig. 2.2. (a) & (b) Néel and Bloch DW observed in PMA nanomagnets. The domain wall is termed as a Néel wall when the magnetization direction at the wall location rotates in a plane perpendicular to the plane of the wall and Bloch wall when the wall magnetization rotates in the plane of the wall, (c) & (d) SOT driven Néel and Bloch DW motion in transverse and longitudinal DWs respectively.

Energy efficient SOT induced DW motion has been also observed in FM-HM bilayers [43–45]. Consider the multilayer structures shown in Fig. 2.2(c-d). Input charge current flowing along the y-direction will cause injection of x-axis directed spins at the FM-HM interface. A general principle to determine the DW movement direction is to calculate the cross-product between the injected spin direction at the FM-HM interface and the magnetization direction at the wall location. The cross product direction signifies the final magnetization state of the magnet, and hence, the DW motion direction. Regarding the orientation of the DW, there can be two alternatives, namely a longitudinal wall (parallel to the length of the magnet) or a transverse wall (perpendicular to the length of the magnet). However, in both cases the wall magnetization needs to be along the y-axis in order to achieve any DW movement. This implies that a Bloch wall configuration is required for the longitudinal wall and a Néel wall orientation is required for the transverse wall. Let us first discuss the case for the longitudinal wall. Shape anisotropy of the magnet (assuming sufficient magnet width, typically above $100nm$) will cause the stabilization of Bloch wall in the FM [46]. However, an in-plane magnetic field is required to retain the stability of the wall in the presence of injected spins due to current flow in the underlying HM [46]. On the other hand, the Néel wall can be stabilized by an effect termed as the Dzyaloshinskii-Moriya exchange interaction (DMI), which is normally associated with such FM-HM bilayers due to spin-orbit coupling and broken inversion symmetry of such magnetic heterostructures [47–49]. As a matter of fact, the DMI strength in certain multilayers like CoFe-Pt or CoFe-Ta [48, 49] has been observed to be strong enough to impose Néel wall configuration even for wider nanomagnets where conventional magnetostatics would have yielded a Bloch configuration. Note that Bloch wall stabilization in the former case (longitudinal DW) discussed before is possible in samples with negligible DMI [46]. The strength of the effective DMI field at the wall location is enough to stabilize the Néel wall magnetization even in the presence of in-plane injected spins due to current flow through the underlying HM. Hence no external magnetic field is required for DW propagation in such magnetic

multilayers with inherent DMI effect and consequently more attractive from scalability point of view. As a result we will focus on device structures based on the latter case for the remainder of this text.

The DMI effect can be modeled by including an additional field ($\mathbf{H}_{DMI}$) in the calculation of the effective field $\mathbf{H}_{eff}$ and is given by,

$$\mathbf{H}_{DMI} = -\frac{2D}{\mu_0 M_s}\left[\frac{\partial m_z}{\partial x}\widehat{x} + \frac{\partial m_z}{\partial y}\widehat{y} - \left(\frac{\partial m_x}{\partial x} + \frac{\partial m_y}{\partial y}\right)\widehat{z}\right] \tag{2.7}$$

where D represents the effective DMI constant and determines the strength of DMI field in such multilayer structures. A positive sign of D implies right-handed chirality and vice versa. In the presence of DMI, the boundary conditions at the edges of the sample is given by,

$$\frac{\partial \widehat{\mathbf{m}}}{\partial n} = \frac{D}{2A}\widehat{\mathbf{m}} \times (\widehat{\mathbf{n}} \times \widehat{z}) \tag{2.8}$$

where A is the exchange correlation constant and $\widehat{\mathbf{n}}$ represents the unit vector normal to the surface of the FM.

2.4 Lateral Spin Valves

Spin current injection can also occur in Lateral Spin Valve (LSV) structures, as depicted in Fig. 2.1(d), where an injector and a detector ferromagnet are situated on top of a non-magnetic channel. When electrons flow through the injector magnet to the ground contact of the channel lying below the magnet, a large number of spins oriented in the same direction as the magnetization of the injector magnet are accumulated in the channel region underneath the magnet. The gradient of this spin potential difference between the two spin orientations causes one type of spin to flow along the channel, thereby exerting non-local spin-torque on the detector magnet. The magnitude of injected spin current decays exponentially with distance between the two ferromagnets due to spin-flip processes. Apart from choosing appropriate materials with longer spin-flip lengths [50, 51], a tunneling barrier can be inserted between the magnet and channel to achieve better spin injection [51]. Recent experiments

have demonstrated non-local spin-torque induced magnetization reversal in Py/Au nanopillars located on top of a Cu wire [52].

2.5 Towards More Efficient Devices

Improving the efficiency of operation of spin devices, and notably the "write" and "read" mechanisms is key to achieving scalable, compact and low-power neuromimetic devices. Using PMA materials is one possible alternative to reduce the critical switching current density for magnetization reversal [40–42] or DW displacement [43, 45, 49, 53]. Other physical mechanisms like voltage-controlled magnetic anisotropy [54], magnetoelectric effect [55, 56] or topological insulator induced spin current generation [57, 58] are also under exploration that can potentially serve as replacements for HM induced magnetization switching. Innovations in the material stack, for instance using Heusler alloys [59] or anti-ferromagnetic materials [60, 61] may lead to further energy benefits. Multi-level information encoded by DW position in magnets can be also potentially replaced by current induced skyrmion displacement [62, 63]. While the discussion in this article will be mainly based on single-domain or DW motion based multi-domain devices with HM underlayers, the concepts can be easily extended to incorporate innovations in the material stack or the underlying physical mechanism utilized for switching [64–67].

Additionally, improving the TMR effect is crucial to achieving more efficient synapses that can offer higher distinguishability for the scaling operation of the neuron inputs. While the theoretical limit of the AP and P resistance ratios is near 300 [68], experiments have achieved a maximum variation of 600% till date [69]. A roadmap issued by the IEEE Magnetics Society has predicted a variation of 1000% in a time period of ten years [70].

3. NEUROMORPHIC COMPUTATION: PRELIMINARIES

In this section, we will first describe the functionality of the major units of such neural computing models. We will also discuss different variants of neuron models (with varying degrees of bio-fidelity) and synaptic learning mechanisms. Relationship of such models to neuroscience mechanisms observed in the brain will be also established.

The main functional units of such neuromimetic computations are the neuron and the synapse. Synapses are adaptive or plastic junctions between neurons that modulate the strength of the signal being transmitted from the pre-neuron to the receiving or post-neuron. Computational tasks like pattern recognition are therefore performed by virtue of plasticity of the synapses in response to signals being transmitted between the neurons since they encode the importance level of different inputs being received by a particular neuron. Fig.3.1(a) depicts a particular synaptic connection between a pre- and a post-neuron. Neuromorphic computation relies on the abstraction of the plasticity of the synaptic junction (governed by neuro-transmitter release at the synapse due to the incoming action potential from the pre-neuron) and the neuroscience mechanisms occurring in the post-neuron (to generate an outgoing signal to the next layer of neurons).

3.1 Neural Computation

Each neural computing unit receives a set of inputs from other pre-neurons through synaptic junctions. The weighted contribution from all the neurons is then summed up and processed by the neurons. The bio-fidelity level at which the "artificial" neuron is modeled has gradually evolved over the last few years from simple perceptrons to more biologically realistic spiking neurons [71]. Irrespective of the details of the neural model, it is worth noting the nature of neuromorphic computation being real-

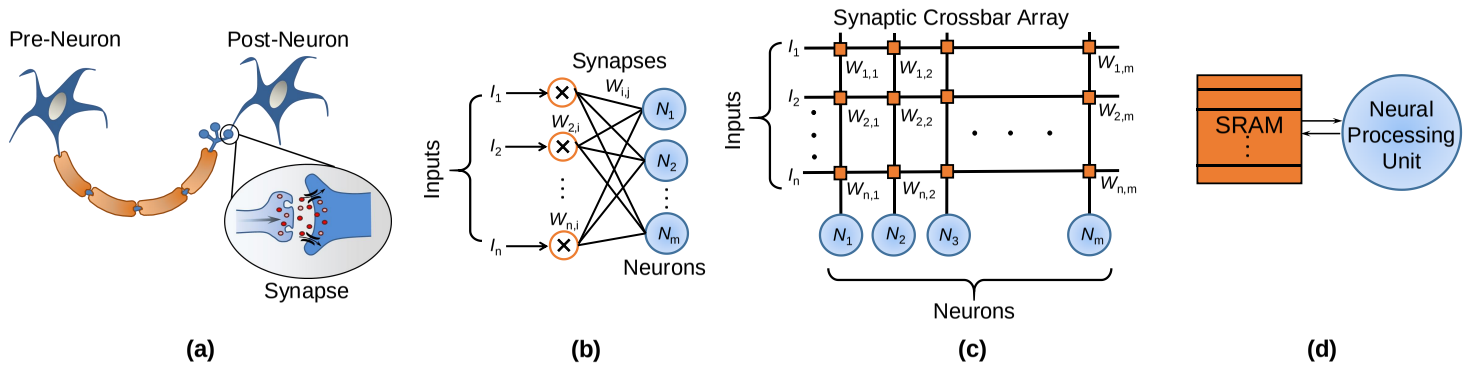

Fig. 3.1. (a) A pre-neuron transmits signals to a post-neuron through a synaptic junction, (b) Computation in a particular layer of a fully connected network can be mapped to a parallel dot-product operation between the inputs and the synaptic weights followed by neural processing for each neuron in the layer, (c) Such a computing kernel can be implemented in a crossbar array structure where programmable resistive devices encoding the synaptic weight are present at each cross-point. Input voltages applied along the rows get weighted by the synaptic conductance and provide the resultant input current (dot-product of applied voltages and synaptic conductances) to the neuron for processing, (d) In contrast, a CMOS architecture would consist of an SRAM module for synaptic weight storage. Memory access and memory leakage due to data transfer between the SRAM module and the computation core (Neural Processing Unit) constitute a significant portion of the total energy consumption.

ized in such networks. Considering a set of neurons in a particular layer receiving a set of inputs through synaptic weights, the computation can be mapped to a parallel dot-product operation between the inputs and synaptic weights followed by neural processing for each neuron in the layer (Fig. 3.1(b)). Such a computing kernel is inherently suited for "in-memory" computing platforms based on crossbar arrays of memristive devices as shown in Fig. 3.1(c) [72, 73]. A memristor is a nanoscale non-volatile programmable resistor. Input voltages drive the rows of the crossbar array where a resistive device encoding the synaptic weight is present at each cross-point joining a particular input to the corresponding neuron. The current flowing through a particular memristive synapse is scaled by the device conductance (synaptic scaling operation) and all such currents gets summed up along the column of the array, according to Kirchhoff's law, and passes as the resultant input to the neuron. Additionally, due to non-volatility of the crossbar memristive elements, such architectures do not suffer from leakage concerns. In contrast, digital CMOS implementations like the IBM TrueNorth involves an architecture depicted in Fig. 3.1(d), where synaptic weights would be fetched from a Static-Random-Access-Memory (SRAM) bank to the neuron computing core [74, 75]. The inefficiency of such architectures results from the memory access and leakage energies (which usually constitutes $\sim 60-80\%$ of the total energy consumption in typical pattern recognition workloads for fully connected networks) and the overall system performance is memory bandwidth limited.

Let us now describe the details of neural processing across different generations. Perceptron networks consist of neurons having "step" transfer function (relationship between the output and input signals), i.e. they generate a high output signal if the weighted summation of neuron input crosses a particular threshold [71]. However, since their success was limited to only a very small set of simple problems, they were replaced by the "second" generation of "artificial" neurons where the transfer function of the neuron was "non-step", i.e. the neuron produced an analog output in response to the input stimulus [71]. Such neurons offer high recognition accuracies in a vast category of large-scale recognition tasks and are routinely utilized today

as a basic building block of deep neural networks. The scalability of such neurons to more "difficult" problems can be attributed to the fact that a greater degree of information can be encoded in the analog neuron output in contrast to the encoded binary information in perceptron networks. A second and equally important contributing factor is the gradient of the neuron transfer function. Backpropagation [76], which is the underlying algorithm for training networks of such neural units, relies on the computation of the partial derivative of the error function (difference between the network output and the desired output) with respect to the synaptic weights, which in turn, is dependent on the gradient of the neuron transfer function. Hence, while a "non-step" neuron transfer function offers gradient information during error backpropagation, perceptrons offer gradient information only at the threshold point. A few popular "non-step" neuron transfer functions are the Sigmoid and Rectified Linear Unit (ReLU) functions.

A more recent paradigm shift in neural computing has been the "spiking" neuron model, encoding a much higher degree of bio-fidelity [77]. A principal biological information that was completely ignored in the first two neuron generations was the mode of neural communication. Biological neurons communicate with each other through binary signals or spikes [77, 78]. Hence, in order to account for neuron communication by means of spikes and simultaneously overcome the bottlenecks of perceptrons (neuron providing '0' - no spike and '1' signal - spike), such "spiking" neurons consider the input as a time-series event instead of a single value as in previous generations. The input is usually encoded in a series of time-steps and provided to the neuron. A common form of input encoding is that of a Poisson spike train, where the probability of spike generation at a particular time-step is proportional to the value of the input. This is usually referred to as "rate" encoding [79] in literature, since the number of spikes transmitted over a given timing window is proportional to the value of the input.

The most common "spiking" neuron model is that of the Leaky-Integrate-Fire (LIF) neuron [78], whose temporal dynamics is given by,

$$C_{mem}\frac{dV_{mem}}{dt} = -\frac{V_{mem}}{R_{mem}} + \sum_i w_i.\delta(t - t_{f,i}) \tag{3.1}$$

where V_{mem} is the membrane potential, R_{mem} is the membrane resistance, C_{mem} is the membrane capacitance, w_i is the synaptic weight for the i-th input, and $\delta(t - t_{f,i})$ is the spiking event occurring at time-instant $t_{f,i}$. When the neuron's membrane potential V_{mem} crosses the threshold V_{th}, the membrane potential gets reset to V_{reset} and does not vary for a time duration termed as the refractory period [78]. Note that more bio-plausible neural models account for the modeling of a post-synaptic current that increases every time a spike is received and then decays exponentially [78]. This post-synaptic current is then integrated by the LIF neuron instead of the spikes as mentioned in Eq.3.1.

It is worth noting here that "spiking" neuron models are not only limited to being more biologically plausible, but offers a host of advantages from hardware implementation perspective. One of the most important breakthrough has been in the arena of unsupervised adaptive local learning enabled by Spike-Timing Dependent Plasticity (STDP) which has made it possible for learning functionalities to be enabled "on-chip". We will discuss synaptic learning in details in the next sub-section. Additionally, since such networks are 'spike' or 'event driven' and can perform pattern recognition by sparse distribution of spikes, they can potentially lead to sparse, event-driven hardware that exploits power-gating functionalities [74,75]. For instance, synaptic weights can be now fetched from the SRAM bank only upon the receipt of an input event or 'spike' (unlike non-spiking nets where all the synaptic weights are required to be fetched to the computing core for each input). Asynchronous event-driven communication techniques at the architecture level like Address Event Representation (AER) are also under exploration [80, 81]. At the circuit level, an additional benefit is achieved due to the replacement of a multiplier by a multiplexer for each synaptic scaling operation. Since the inputs are binary, they do not need to be multiplied by

the synaptic weights but can be transmitted to the neural computing core in case a 'spike' is received [82]. Note that the loss of information due to binary inputs is compensated by temporal encoding over the time-steps of the spike train. However, the advantages due to reduced power consumption of spiking networks (event-driven hardware) far outweigh the cost of increased delay for inference (temporal encoding) [82, 83]. Further, the sparsity of neural spiking activations increases drastically with network depth [84]. Hence, the power and energy benefits improve further with larger sized networks imperative for complex machine learning tasks [84].

Due to such inherent advantages of Spiking Neural Networks (SNNs) at the hardware level, there has been significant interest in recent years to convert non-spiking nets to SNNs by replacing the original neurons by "spiking" neurons after training [84, 85]. The main motivation behind the conversion stems from the fact that while non-spiking nets can be trained with very high classification accuracies at large-scale recognition tasks using backpropagation, achieving similar accuracies in STDP trained spiking networks is still an active research area. The "spiking" neuron model typically used for such conversion schemes has been the Integrate-Fire (IF) model which is equivalent to the LIF neuron without any leak term in the membrane potential. Such an IF neuron without any refractory period has been shown to be a firing-rate approximation of the ReLU unit mentioned previously [86]. This is apparent from the fact that higher the value of the input for the ReLU, higher is the value of the neuron output. Similarly, for the IF neuron, higher is the rate of input spikes, higher is the number of transmitted output spikes. Recently, deep layered SNNs with VGG and Residual network architectures (trained using such ReLU-IF spiking neuron conversion mechanism) have demonstrated competitive accuracies over complex datasets like CIFAR [87] and ImageNet [88] (see Appendix A).

However, note the fact that the above "spiking" neuron computing models are completely deterministic and do not account for the noisy probabilistic neural computation that actually occurs in the human brain. Recent proposals have investigated stochastic neural models that abstract the neural computation by a probability dis-

tribution function that varies as a function of the input being received by the neuron at each time-step of computation [89–92]. The variation is usually characterized by a non-linear functionality. Such probabilistic neural computation has been observed in 'pyramidal' spiking neurons in the cortex and recent research proposals have investigated the possibility of performing Bayesian computation in cortical microcircuits of stochastic neurons [91, 92]. Additionally, such stochastic neural computational units have been also used in Restricted Boltzmann Machines and Deep Belief Networks [93] trained by Contrastive Divergence [94]. Such probabilistic "spiking" neural models are particularly interesting for spintronic device applications since such devices are inherently characterized by a time-varying thermal noise leading to stochastic behavior.

We would like to conclude this section on neural computation by a brief discussion on an additional neuroscience mechanism termed as homeostasis [95] that is also routinely utilized in SNN based pattern recognition systems. It is a spike frequency adaption mechanism wherein the neuron threshold increases by a specific amount every-time the neuron spikes. This ensures that as a neuron starts to dominate the spiking pattern in a particular pool of neurons, it also becomes progressively difficult for that particular neuron to spike in the future. We will discuss the manner in which such homeostasis effects assist in performing pattern recognition.

3.2 Spike-Timing Dependent Plasticity

As mentioned in the previous section, prior to the advent of SNNs, synaptic learning was achieved primarily by backpropagation algorithm [76]. This is a supervised training algorithm where the neural network is trained with a particular set of inputs that are associated with specific class labels or categories. The algorithm aims at finding the optimal set of synaptic weights by minimizing the error function (difference between class labels and actual network outputs) using gradient descent algorithm. Readers are referred to Ref. [76] for details on the backpropagation algorithm. A

few key points worth noting is the supervised nature of the training algorithm and the synaptic weight update scheme which is not only dependent on the outputs of neurons in other layers of the network but also require a backward pass of the gradient computation through the entire network. This has broadly limited the scope of specialized hardware to implement backpropagation on-chip due to expensive power and area requirements of the underlying hardware.

The number of applications requiring some form of intelligence in present day Internet of Things (IoT) technologies like mobiles and wearables are huge and often require embedded on-chip intelligence since it is often not possible to transmit data in real-time to cloud for computing. Further, it is also not practical to have supervised learning algorithms to implement pattern recognition systems since real-time data will be mostly unlabeled (without any specific categories). Hence, unsupervised hardware-inexpensive synaptic learning mechanisms is a key requirement for the implementation of on-chip learning.

A more bio-realistic and hardware-friendly approach to synaptic learning in comparison to backpropagation is the STDP learning rule in SNNs, which is based on measurements obtained from rat hippocampal glutamatergic synapses [96] (Fig. 3.2(a)). According to this theory, the synaptic weight is modulated depending on the spiking patterns of the pre-neuron and post-neuron. The synaptic weight increases (decreases) if the pre-neuron spikes before (after) the post-neuron. Intuitively, this signifies that the synapse strength should increase if the pre-neuron spikes before the post-neuron as the pre-neuron and post-neuron appear to be temporally correlated. The relative change in synaptic strength decreases exponentially with the timing difference between the pre-neuron and post-neuron spikes. The STDP characteristics can be formulated in a mathematical framework as follows,

$$\begin{aligned} \Delta w &= A_+ \exp\left(\frac{-\Delta t}{\tau_+}\right) \begin{cases} \Delta t > 0 \\ \end{cases} \\ &= -A_- \exp\left(\frac{\Delta t}{\tau_-}\right) \begin{cases} \Delta t < 0 \\ \end{cases} \end{aligned} \tag{3.2}$$

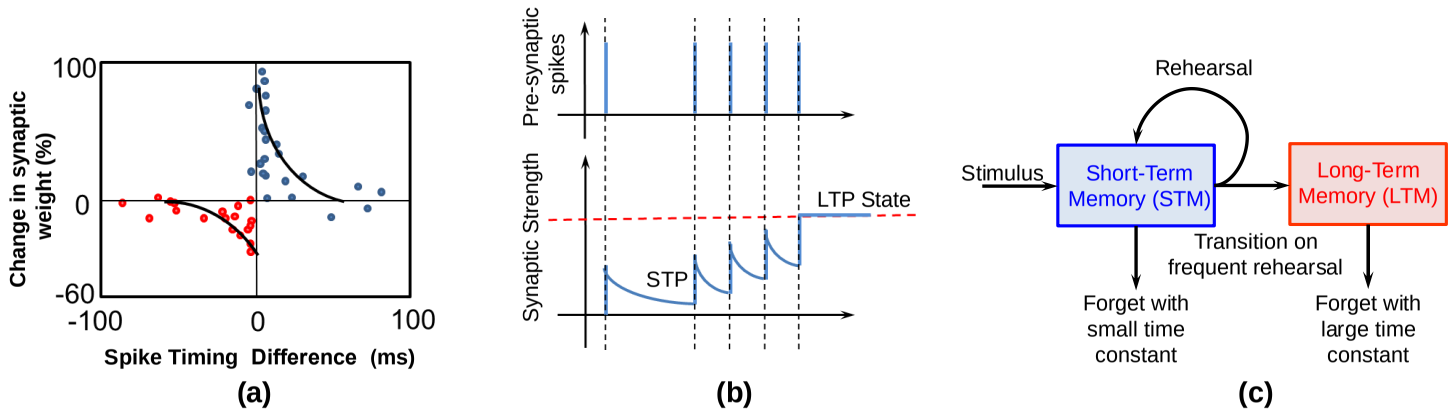

Fig. 3.2. (a) Spike-Timing Dependent Plasticity (STDP) measurements obtained from rat hippocampal glutamatergic synapses [96]. STDP learning rule can be formulated by considering that the synaptic weight potentiates (depresses) if the pre-neuron spikes before (after) the post-neuron. The variation is exponential with spike-timing difference, (b) The synaptic strength increases momentarily on the receipt of a pre-synaptic spike but starts decaying back to the initial value in the absence of spikes. This is referred to as Short-Term Plasticity (STP). On frequent stimulation, the synapse strengthens to a long-term stable state. This is referred to as Long-Term Potentiation (LTP), (c) STP-LTP is often correlated to the concept of Short-Term Memory (STM) and Long-Term Memory (LTM). While information is initially stored in the STM, it gets transferred to the LTM on frequent rehearsal of the input stimulus.

Here, A_+, A_-, τ_+ and τ_- are constants and $\Delta t = t_{post} - t_{pre}$, where t_{pre} and t_{post} are the time-instants of pre- and post-synaptic firings respectively. We will refer to the case of $\Delta t > 0$ ($\Delta t < 0$) as the positive (negative) time window for learning for the rest of this text. Note that this learning mechanism is unsupervised since no prior information about input class or label is necessary. Further, synaptic weight update is completely local since it is modulated depending on the activities of only the neurons it connects. This has enabled learning functionalities to be implemented on-chip at much lower hardware costs. Although pattern recognition systems with high accuracies based on STDP learning are still in preliminary stage, competitive accuracies in typical digit recognition and sparse encoding workloads have been already achieved [95]. Note that the above STDP learning rule is referred to as anti-symmetric STDP and has been the most popular learning mechanism for training SNNs. However, other variants of STDP have been also observed in neuroscience studies and have been utilized in different genres of recognition tasks [97].

We will discuss STDP implementation in spintronic synapses in later sections. However, a primary concern for such spintronic synapses, and in general any memristive synapse technology, is the bit resolution at aggressively scaled device dimensions. Driven by this fact, researchers have proposed variants of STDP learning based on single-bit synapses [7, 98, 99] where the multi-bit requirement is replaced by probabilistic synaptic weight update. It has been already mentioned that spintronic devices exhibit an inherent stochasticity during the switching process which has been mainly attributed to the time-varying thermal noise [28]. Hence, the STDP framework described in Eq. 3.2 can be modified in this scenario as the probability of binary synaptic state change (instead of analog weight change) to offer a direct correspondence to stochastic switching behavior of single-bit nanoelectronic synapses. Stochastic single-bit synaptic learning achieving competitive accuracies in digit recognition applications has been recently demonstrated in SNNs [7].

3.3 Volatile Synaptic Learning

The exact mechanisms that underlie learning or plasticity of synapses is highly debated and still unknown. While STDP has been a popular viewpoint of explaining synaptic plasticity, there has been some research studies that attempt to explain synaptic plasticity from an alternative volatile learning plasticity viewpoint. This is referred to in literature as Short-Term Plasticity (STP) and Long-Term Potentiation (LTP) [100, 101]. The theory postulates that synapses undergo inherent volatile state changes upon receipt of incoming action potentials (due to release of neurotransmitters). In case the action potentials are received infrequently, the neurotransmitter concentration decays to the background value after the action potential is removed and hence the synaptic plasticity remains unchanged (STP). However, as more frequent action potentials are received, the ionic neurotransmitter concentration starts increasing and ultimately the synapse switches to a stable long-term state (LTP). Hence, while STDP is a form of non-volatile synaptic learning, STP-LTP models synaptic plasticity as a form of frequency-dependent volatile synaptic learning. While adoption of STP and LTP concepts in SNNs for usage in pattern recognition is still an area of active research, it offers the promise of adaptive learning where the network might be able to unlearn itself in response to changing environments, which might not be possible to achieve by non-volatile STDP learning rule.

Such a learning mechanism is in accordance to the volatile forgetting nature of human memory and has been often correlated to Short-Term Memory (STM) and Long-Term Memory (LTM) psychological models proposed by Atkinson and Shiffrin [102, 103]. The model is equivalent to STP and LTP where the synaptic element can be viewed to be analogous to human memory. Input information is received and stored in the STM and only gets transferred to LTM if the input is received with sufficient frequency. The characteristic difference between STM and LTM is that while information is stored for a limited period in STM (analogous to volatile meta-stable synaptic state change in response to input stimulus), LTM retains the

information for a much longer period of time (analogous to long-term stable synaptic state). Fig. 3.2(b) and (c) illustrates the concepts of STP-LTP and STM-LTM respectively. It is worth noting here that psychological STM-LTM concepts have been also harnessed to model the computational units of Recurrent Neural Network (RNN) architectures [104].

3.4 Network Connectivity

The discussion so far has been limited broadly to the functionalities exhibited by the fundamental units in neuromorphic systems. However, in order to construct pattern recognition systems based on these units, specific network connections and topologies are necessary. Initial studies in neural networks mainly focused on fully-connected nets (FCNs), where neurons are arranged in different layers and connected in an all-to-all fashion, as shown in Fig. 3.1(b). However, such simple network connectivity failed to be invariant to translation or scaling of input patterns. Further, FCNs with larger number of neurons/layers implies storage of a huge set of synaptic weights along with higher degree of neuron connectivity between layers which limits its scalability to large-scale cognitive tasks.

Deep networks based on convolution operations have been able to overcome most of these challenges. The inspiration behind such a connectivity is based on the seminal work of Hubel and Wiesel which revealed that the animal cortex consists of cells which are sensitive to specific areas of the entire visual field (implying a local connectivity for each neuron) and that they function as filters for that particular receptive field [105]. Further, a certain category of cells were found to be sensitive to edge-like features in the visual field while another category of cells were found to be invariant to the location of the pattern in the receptive field [105]. Such mechanisms served as the main motivation behind the structure of Convolutional Neural Networks (CNNs).

Fig. 3.3(a) shows the CNN structure. Drawing inspiration from the hierarchical arrangement of layers in the visual cortex, CNNs consist of a number of cascaded

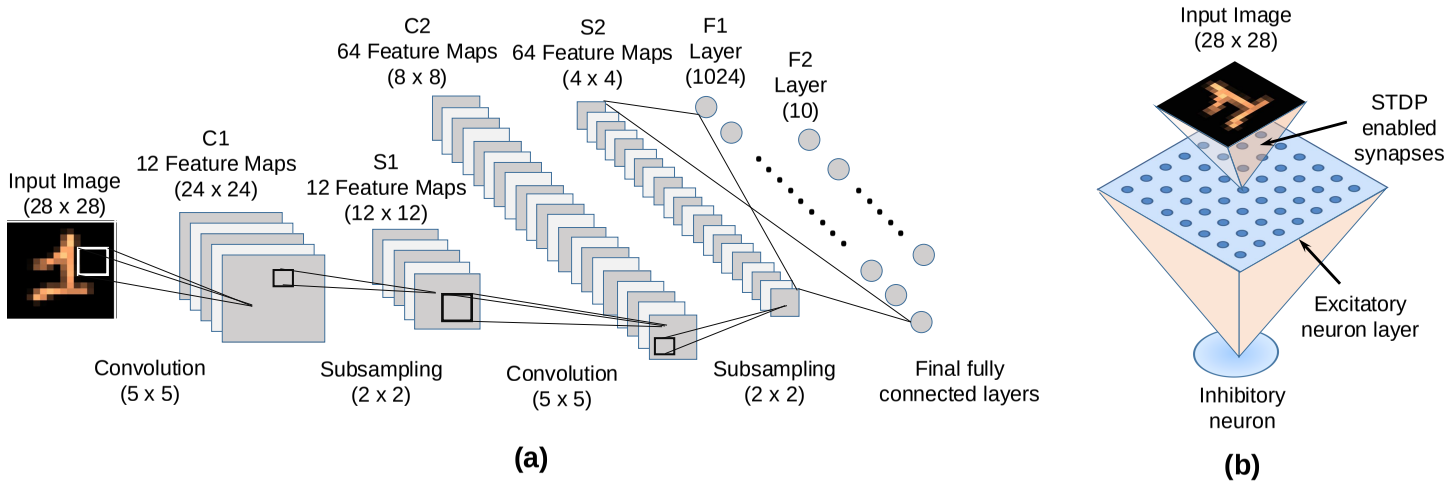

Fig. 3.3. (a) A Deep Convolutional Neural Network (CNN) consists of alternate cascaded layers of convolution and subsampling terminated by a fully connected output layer. The figure depicts a typical CNN network used for digit recognition (28x28-12c5-2s-64c5-2s-10o). (b) A network typically used for studying STDP is shown. Such connections have been observed in cortical microcircuits of pyramidal neurons in the brain. It consists of an excitatory layer of neurons that receives spike trains from the input in an all-to-all fashion. Lateral inhibition and homeostasis promotes STDP learning in such single layer networks.

stages where each stage consists of a convolution layer (C) followed by a sub-sampling layer (S). Each C layer is characterized by a set of trained weight kernels that is used to convolve with the input maps for that particular layer. For instance, in an image recognition system the input map for the first layer of a network would be the entire image being classified. Each kernel is then convolved with the entire image to produce an equivalent number of output maps. Each neuron in the output map therefore has limited connectivity (equal to the size of the convolution kernel). Additionally, the network offers resiliency to image translation and scaling due to the convolution operation. The C stage is usually followed by an S layer which performs an averaging operation over non-overlapping subsampling windows of each output map to reduce their dimensionality. As the depth of the layer increases, the number of maps increases with decreasing dimensionality. Ultimately the final two layers are usually fully connected and the number of neurons in the output layer equals the number of classes in the recognition problem. Due to the limited fan-in of each neuron, sparse neural connectivity is achieved. Additionally number of synaptic weights to be learnt during training is also reduced, due to the shared weight kernel being convolved across the entire map, thereby resulting in significantly reduced training time.

An alternative network architecture that has been popular in the domain of STDP learning enabled SNNs has been shown in Fig. 3.3(b) [95]. Such connections are again inspired from cortical microcircuits of pyramidal neurons observed in the brain. The network consists of a layer of neurons that receive input spike trains through excitatory (positive) synaptic weights in an all-to-all fashion. The network is also associated with a lateral inhibitory signal that triggers a negative spike signal whenever one of the neurons in the layer spikes. In order to prevent single neurons from dominating the spiking pattern due to lateral inhibition, the "spiking" neurons are enabled with homeostasis functionality. STDP in the excitatory synaptic connections in such networks can assist each neuron to respond selectively to specific classes of input patterns. Note that training deeper networks enabled by STDP is still an area of active research.

While the discussion in this section mainly focused on feedforward networks without any directed loops, RNN architectures are also becoming increasingly popular for sequence learning tasks like language modeling [106], handwriting prediction and generation [107], speech recognition [108], among others. The only difference between RNNs and standard feedforward networks is the fact that the computational units or neurons receive its own output from the previous time-step as its input in the current time-step (in addition to external inputs). Such a memory effect in RNNs enables it to perform context learning in sequential inputs. However, note that the main functionalities of the computational units – the neurons and synapses remains unaltered, thereby allowing the same synaptic/neural spin-devices to be used in these different algorithmic architectures. This will be discussed in details in the next section.

4. SPINTRONIC DEVICE PROPOSALS AND CORRESPONDENCE TO NEURAL AND SYNAPTIC FUNCTIONALITIES

Nanoscale programmable resistive devices mimicking neural and synaptic functionalities is imperative towards the realization of energy-efficient neuromorphic architectures. The field of neuromorphic computing, wherein research effort is directed to mimic neural and synaptic mechanisms by the underlying device physics, was pioneered by Carver Mead in the 1980s [109]. He proposed that CMOS transistors operating in subthreshold region can be utilized to implement neuromimetic computations since the main mechanism of carrier transport in that operating regime is diffusion, thereby emulating the mechanism of ion flow in biological neuron channels [109]. Although such sub-threshold CMOS neuron and synapse designs are still being investigated by various research groups [110], they require multiple transistors and feedback mechanisms to mimic the functionality of neurons/synapses. The first work on spintronic neuromorphic computing can be traced back to the work of Krysteczko *et al.* where they explored the possibility of implementing memristive functionalities in MTJ structures through voltage induced switching phenomena [111].

4.1 Spin-Torque Neuristors

In this section, we will review different spintronic device structure proposals that can potentially offer a direct correspondence to neuronal computations with varying degrees of bio-fidelity. Fig. 4.1 depicts various spintronic devices mimicking neurons of different computing generations from “step” to “spiking” neurons. We will begin our discussion on the neuronal devices by considering it receives a resultant weighted

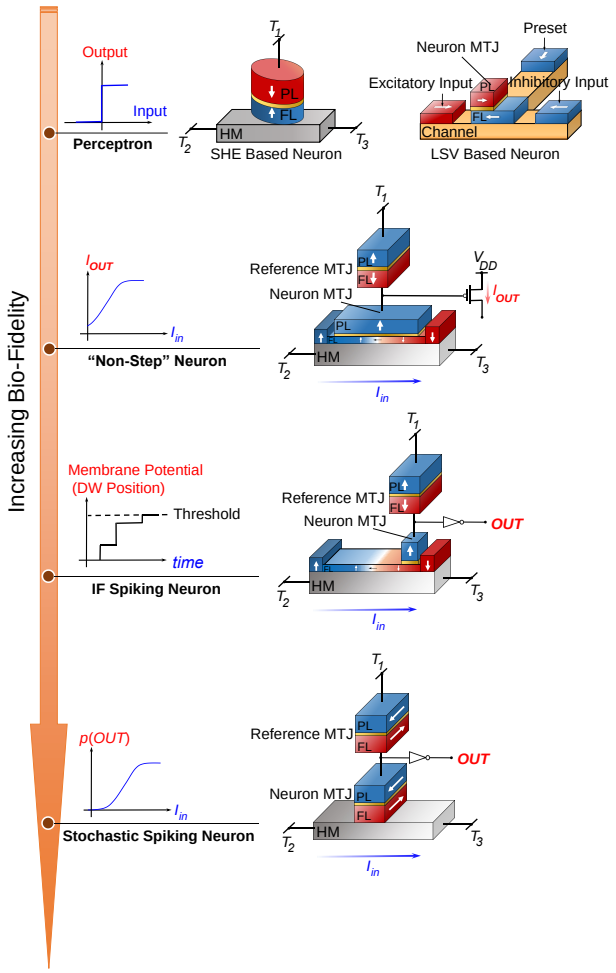

Fig. 4.1. Spin-torque neuristors with different degrees of bio-fidelity are shown. Perceptron or "step" neurons can be implemented in SHE based neuron structures where a current flowing through an underlying HM layer orients a PMA magnet lying on top along the unstable "hard-axis". Subsequently the direction of current flowing through the PL orients the magnet to either of the stable "easy-axis" directions. A complementary device structure can be envisioned using the LSV concept by injecting spins oriented along the "hard-axis" in a non-magnetic channel using a "Preset" magnet. "Non-step" neuron functionalities can be implemented in DW motion based device structures by interfacing the Neuron MTJ with a Reference MTJ. A similar device structure with the MTJ located at the edge of the FL can be used to implement an IF "spiking" neuron. Stochastic "spiking" neuron functionalities can be implemented in mono-domain neural device structures by exploiting the underlying probabilistic MTJ magnetization dynamics.

synaptic current input. Synaptic device structures and interfacing of synaptic arrays with neuronal devices for generating the input synaptic current will be discussed in the next sections.

4.1.1 Perceptron

Let us begin this section by noting the functional similarity between a "step" neuron transfer function and a mono-domain MTJ switching event. The MTJ switches between the two stable P and AP states provided the switching current magnitude is greater than a particular threshold. Consequently, in order to emulate the "step" neuron functionality with neuron threshold at the origin, the input current to an MTJ neuron has to be greater than the switching current requirement, which in turn, increases the operating voltage of the MTJ. Ref. [112] investigated the design of an MTJ based neuron for the implementation of a "step"-transfer function neural network. In order to reduce the input synaptic current magnitude, the MTJ was initialized to the AP state and provided with a bias current that was equal to the critical current requirement for MTJ switching to the P state. Hence, a small magnitude of synaptic current (positive or negative) would ensure MTJ switching to either the P state or remaining in the original AP state. However, due to the high bias and reset current requirements, energy improvements for such MTJ "step"-neuronal devices was highly limited [112]. Note that in this work, the focus point has been the mapping of simply the MTJ switching event to a neuron functionality while the internal time-domain magnetization dynamics has not been considered. As we will show later, utilization of the stochastic MTJ switching dynamics due to time-varying thermal noise to model neural computations can lead to "spiking" neuron implementations with higher bio-fidelity and enhanced recognition performances for computing platforms.

Continuing our discussion on simply the MTJ switching event to mimic a "step" neuron, the energy consumption can be drastically reduced in case the MTJ is initialized to an unstable magnetization state prior to the switching process. This would

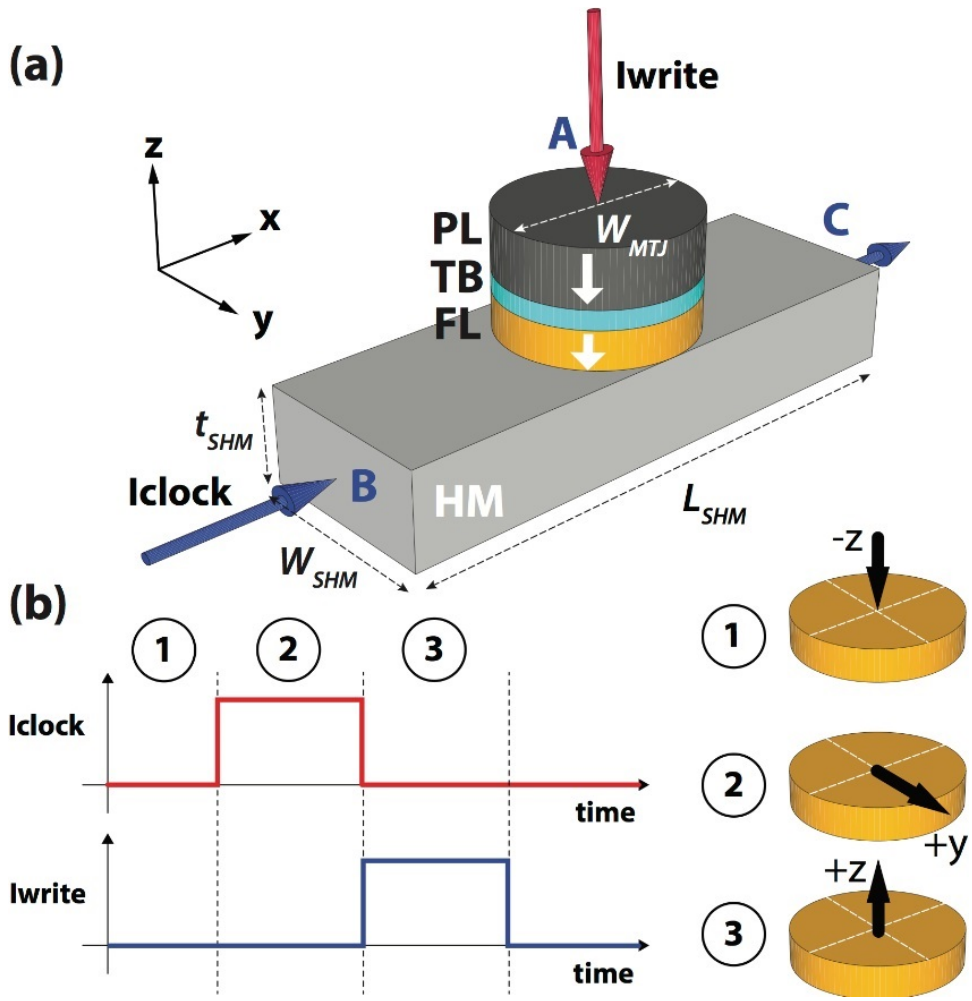

Fig. 4.2. (a) The three terminal thresholding device for spin-neuron consists of an MTJ structure on top of a HM layer, (b) The two-step switching scheme consists of a clocking current I_{clock} flowing through HM from terminals B to C followed by the synaptic current I_{write} flowing between terminals A and C, (c) The clocking current I_{clock} orients the ferromagnet along "hard-axis" while the current I_{write} causes deterministic "easy-axis switching".

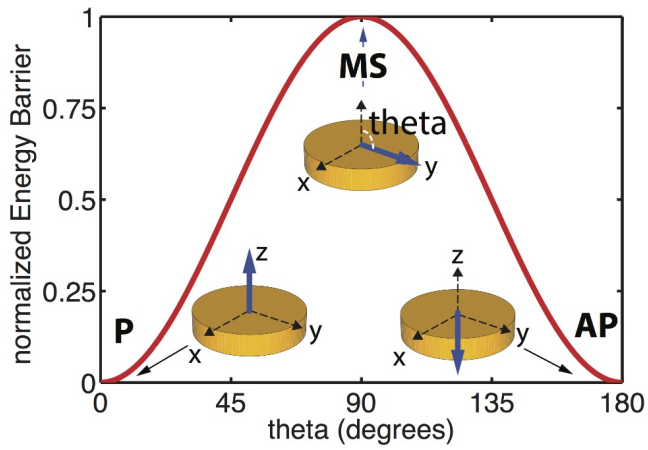

Fig. 4.3. Normalized energy landscape of a nanomagnet with a uniaxial anisotropy in out-of-plane direction. The two energy minima points in the P and AP configuration are separated by an anisotropy barrier.

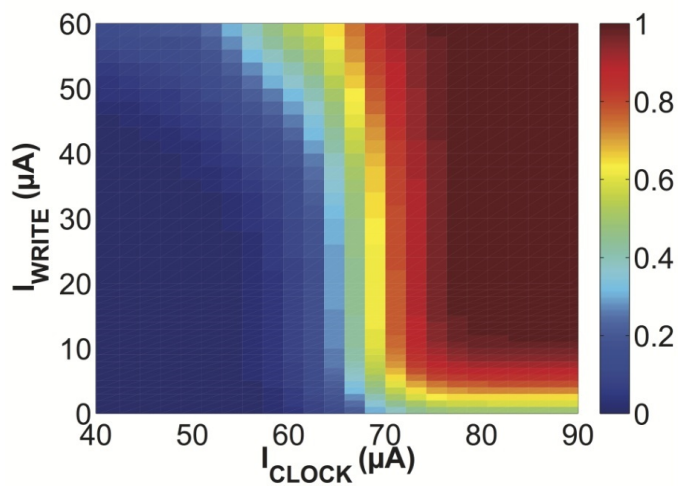

Fig. 4.4. Switching phase diagram showing probability of switching for a range of clock and write currents. The figure depicts that for sufficient magnitude of clocking current, the probability of deterministic switching by write currents is ~ 1 for current magnitudes of the order of a few μA.

assist in reducing the critical current requirement for the switching process, since a very small magnitude of input synaptic current can now enable the switching process to either of the two stable states (depending on the input spin current direction) by overcoming thermal fluctuations. This concept can be utilized in a spintronic device structure (shown in Fig. 4.2(a) [113]) where a PMA magnet lies on top of a HM and is operated in two subsequent stages of "Preset" and "Evaluation". Note that in Section 2.3, we mentioned that PMA magnets cannot be switched solely in presence of SOT since in-plane spins are injected by current flowing through the underlying HM into the PMA nanomagnet lying on top (see Fig. 4.3 which depicts the energy landscape of FM with uniaxial anisotropy, which could originate from shape, interface, or bulk magneto-crystalline anisotropy). Two-step switching schemes have been utilized previously in magnetic quantum-dot cellular automata (MQCA) [114], All-Spin Logic (ASL) [115], SHE-assisted-memory bit-cell [116] and Spin Amplifier [117]. The operation of the device is discussed in details next.

As illustrated in Fig. 4.2(b), for the first step, a charge current (I_{clock}) is supplied through the HM (between terminals B and C) which generates a torque to align the FL magnetization in $\pm y$ direction. In other words, I_{clock} aligns the FL magnetization along the hard-axis of the magnet i.e. the unstable point in the energy landscape (labeled as MS in Fig. 4.3). Let us define this switching stage as "hard-axis switching". Subsequently in the second step, the electronic synapses drive a charge current (I_{write}) between terminals A and C, as illustrated in Fig. 4.2(b). The net synaptic current (I_{write}) flowing through the MTJ exerts a torque on the magnetization which will align the magnet to either one of the easy axis direction along ($\pm z$). This step is referred to as "easy-axis switching". The direction of torque generated by I_{write} depends on the polarity of the net synaptic current. If the synaptic current is a positive value, the sign of torque is such that the FL's magnetization becomes AP to that of the PL. On the other hand, a negative synaptic current places the FL's magnetization P to that of PL. The P and AP states of the MTJ correspond to the low and high (binary) outputs of the neuron. The proposed thresholding device is

functionally similar to a biological neuron 'firing' a pulse when the synaptic signal exceeds a certain threshold.

To determine the appropriate magnitude of clock and write currents for the proposed device, the switching phase diagram for a range of clock and write currents is constructed as shown in Fig. 4.4. The device structure is an elliptic PMA magnet of dimensions $\frac{\pi}{4} \times 40 \times 40nm^2$ for CoFe($1.5nm$)-W($2nm$) bilayer stack. The device parameters are mentioned in Table 4.1. For each set of clock and write currents, $\sim 100,000$ stochastic LLG simulations were carried out to obtain the statistics of switching. For simplicity, the rise and fall times of the pulses were set to zero and the pulse width for clock and write currents are set to $2ns$ and $1ns$, respectively. As it can be observed from the figure, when clock current is large enough, the amount of write current needed to achieve successful switching is on the order of few μA, just enough to overcome thermal fluctuations and tilt the magnet in the desired direction. Although some amount of the synaptic current flows through the HM, the spin-orbit torque generated due to this minimal current is expected to have negligible impact on the magnetization of the FL. Thus the proposed device facilitates fast and energy-efficient threshold operation by utilizing spin-Hall effect for "hard-axis switching" and minimal synaptic current for deterministic "easy-axis switching".

For the first stage of the switching process, a charge current of $\sim 85\mu A$ (from Fig. 4.4)) was used to orient the nano-magnet in the hard-axis position within a duration of $2ns$, resulting in a power consumption of $\sim 7.22\mu W$ per neuron. The fast and energy efficient "hard-axis switching" is mainly attributed to a spin injection efficiency of 4.71 resulting from SOT. In the next step, the net synaptic charge current drives the magnet to one of its stable magnetization states. The operating supply voltages of the synaptic devices were limited by the minimum current required to deterministically switch the spin neuron in the appropriate direction (Fig. 4.4).

Additionally, the functionality of the proposed device due to the presence of a finite delay between the I_{clock} and I_{write} signals was assessed by determining the variation of the write error rate of the FL with the synaptic current, corresponding

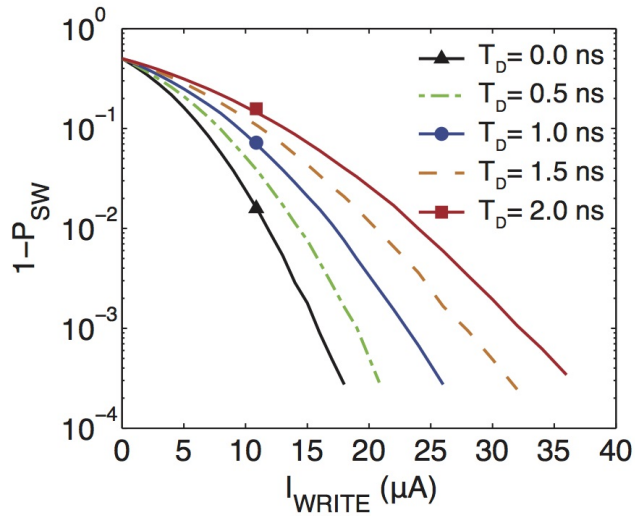

Fig. 4.5. The figure depicts the variation of the write error rate (1 - switching probability) of the FL with the synaptic current, corresponding to a clocking current of $85\mu A$ for different values of delay (T_D) between the clocking and synaptic currents.

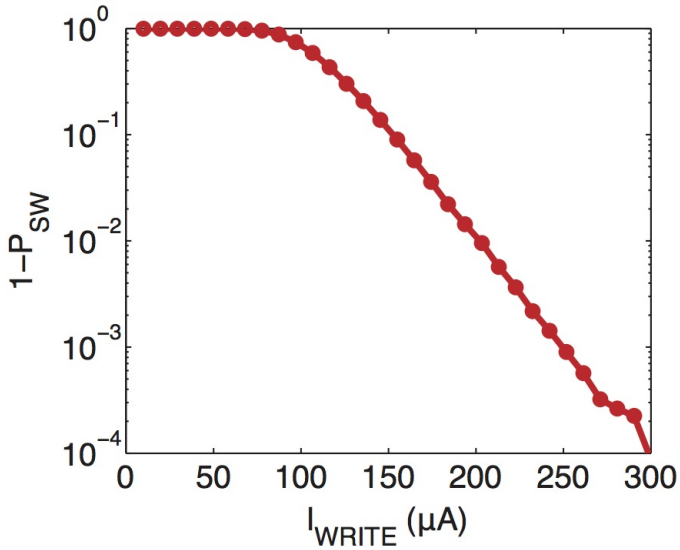

Fig. 4.6. The figure depicts the variation of the write error rate of the FL with no applied clocking current.

Table 4.1.
Typical Device Parameters for CoFeB-W Samples [118]

Parameters	**Value**
Saturation Magnetization	1000 KA/m
Spin-Hall Angle	0.3
Spin-Hall Metal Resistivity	200 $\mu\Omega.cm$
Gilbert Damping Factor, α	0.0122

to a clocking current of $85\mu A$ (Fig. 4.5) by performing $\sim 100,000$ stochastic LLG simulations. Once the magnetization is put in its "hard-axis", its relaxation to "easy-axis" can be described by a characteristic relaxation time constant, $\tau_D = \frac{1+\alpha^2}{\alpha\gamma H_K}$, where H_K is effective anisotropy field. Using simulation parameters used in this work, the relaxation time constant τ_D is calculated as $3.5ns$. As a result, if the delay time between I_{clock} and I_{write} is less than τ_D, then the functionality of the proposed neuron would not be significantly affected. A worst case simulation of the feed-forward ANN with an average delay of $1ns$ between the clocking and synaptic currents for each neuron in the network showed insignificant degradation in classification accuracy. The inherent error resiliency of such neural computing algorithms helps in nullifying the effect of delay between clocking and synaptic currents to a large extent. In order to quantify the advantage of using spin-Hall effect to clock the neuron, the switching probability curve for the neuron with no prior clocking current is shown in Fig. 4.6. The synaptic current required to achieve the same write error rate is almost one order of magnitude lower for the proposed clocking scheme of the spin-orbit torque neuron.

4.1.2 "Non-Step" Neurons

Let us now proceed to the implementation of "non-step" neuron functionalities in spintronic devices. Note that since an MTJ with a mono-domain FL consists of two stable states, only two distinct neuron outputs can be represented by such a device structure. However, for a multi-domain FL, where the magnet consists of two oppositely polarized magnetization regions separated by the DW, the device can exhibit multi-resistive states.

As shown in Fig. 4.7, our proposed device structure consists of an MTJ structure where the FL is a DW magnet (magnet having a transitory DW region) lying on top of a HM layer (for energy efficiency) [119, 120]. The underlying device physics for transverse Néel DW motion in such PMA magnetic multilayers due to charge current flow through the HM has been discussed in Section 2. Note that a complementary device structure utilizing spin-orbit torque induced Bloch DW motion was also investigated in Ref. [121]. Although the discussion henceforth will be based on Néel wall motion, the concepts are equally valid for device structures utilizing Bloch DW motion. The FL is surrounded by two PLs on either side to ensure that the DW stabilizes at the opposite edges of the FL for large magnitudes of the current flowing through the underlying HM. A multi-level DW motion based resistive device was recently shown to exhibit 15-20 intermediate resistive states [122].

The operation of such a multi-terminal device occurs in two subsequent "write" and "read" stages. During the "write" stage, the magnitude of current flowing through the HM ("write" current) programs the position of the DW in the FL of the MTJ structure. The DW displacement increases linearly with the magnitude of the input synaptic current flowing through the underlayer (I_{in}) between terminals T_2 and T_3. After the "write" phase, terminal T_1 is activated instead of T_2 which enables the "read" current path in the device between terminals T_1 and T_3. Such decoupled "read" and "write" current paths not only assist in optimizing the "write" and "read" peripheral circuits independently but also enable a low value of resistance in the path

of the "write" current (mainly the resistance of the underlying HM layer). As we will discuss in a later section, a crucial functionality that is required for nanoelectronic neurons is low input "write" resistance for proper operation of neuromorphic crossbar arrays. It is the decoupled nature of the "write" and "read" current paths of such multi-terminal devices that have made it possible for spintronic devices to be utilized not only as a synapse, but also as a neuron.

The DW position of the FL is sensed by a simple resistive divider, as shown in Fig. 4.7, where the neuronal device is interfaced with a Reference MTJ which is always fixed to the AP state. The "read" current can be maintained to sufficiently low magnitudes by ensuring proper oxide thickness of the neuronal and Reference MTJs which assists in achieving "disturb-free read" of the neuron MTJ. This resistive divider drives a transistor operating in saturation regime (in order to ensure that the supplied current to the fan-out resistive synapses is independent of the magnitude of the interfaced synaptic resistances). As the magnitude of the input current I_{in} increases, the resistance of the neuronal device reduces due to decrease in the proportion of the AP domain in the MTJ device. This, in turn, causes the current provided by the output transistor (I_{out}) to increase. It can be shown that the transfer function (relationship between I_{out} and I_{in}) of such a device is approximately linear by performing a device-circuit co-design discussed next. Note that a biological neuron's output is transmitted via the axon to fan-out neurons. Similarly, the spintronic neuron receives a resultant synaptic current which is the weighted summation of its inputs. This resultant current input flowing through the heavy metal of the spintronic neuron generates an output which is transmitted via the CMOS transistor, acting as the axon, to the next stage. After every "read" cycle, the neuron is "reset" for the next operation by passing a current through the HM in the opposite direction to initialize the DW at the opposite edge of the MTJ.

Fig. 4.8(a) shows the domain wall displacement in a CoFe sample with cross-section of $160nm \times 0.6nm$ for a charge current density of $J = 0.1 \times 10^{12} A/m^2$. The grid size was taken to be $4 \times 4 \times 0.6nm^3$. Fig. 4.8(b) depicts the variation of

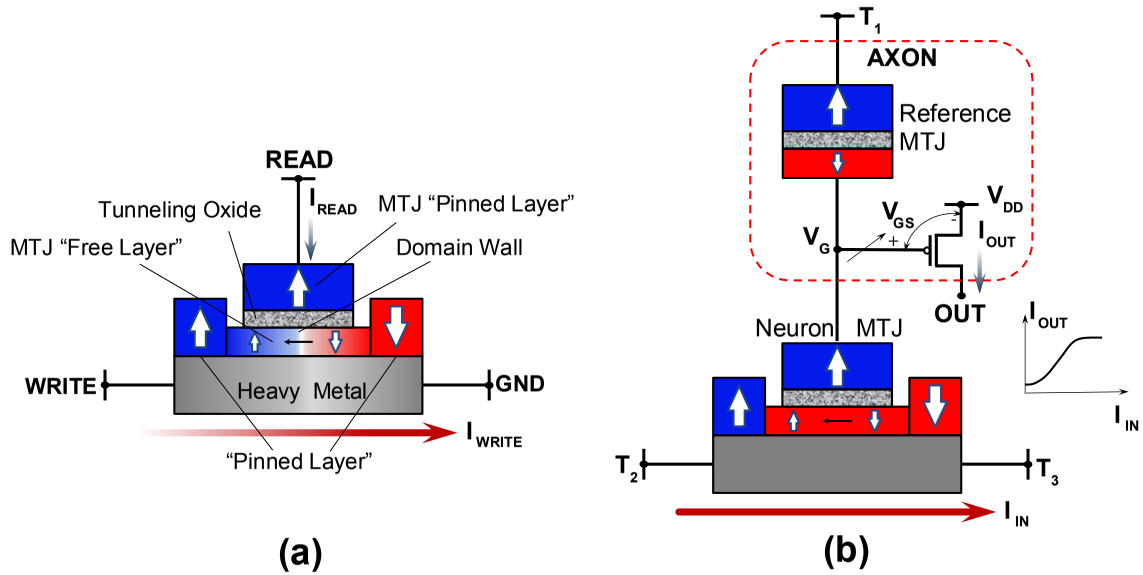

Fig. 4.7. (a) Three terminal device structure acting as the basic building block for the All-Spin ANN. Spin-orbit torque (SOT) generated by current, I_{WRITE}, through the heavy metal programs the domain wall position in the MTJ "free layer". The domain wall position encodes the device conductance between terminals READ and GND, (b) Operation of the spintronic device as a neuron. Initially the neuron is "reset" such that the domain wall position is initialized to the left edge of the "free layer". Then the resultant synaptic input current programs the domain wall position. Subsequently, during the "read" phase, the "Reference MTJ" and PMOS transistor serve as the axon to propagate the neuron output to its fan-out neurons. The transfer function of the neuron is characterized by the relationship between I_{OUT} and I_{IN}.

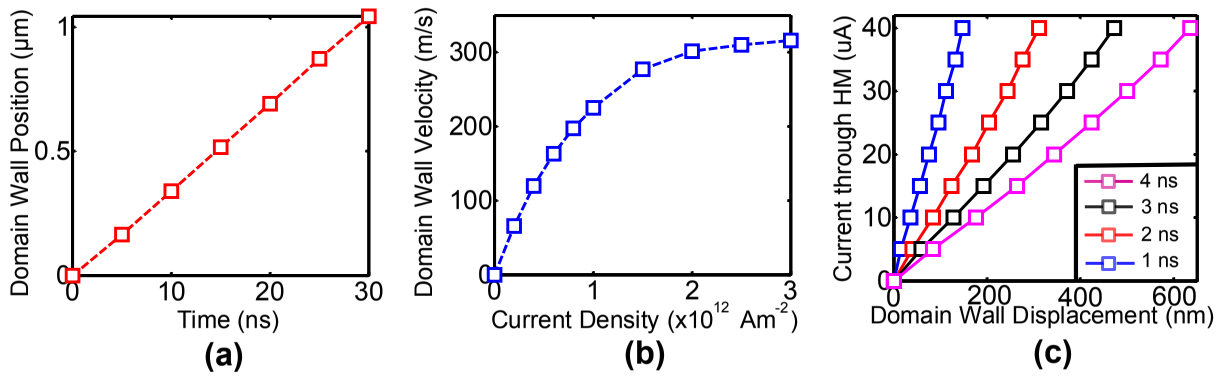

Fig. 4.8. (a) Domain wall displacement as a function of time for a CoFe strip of cross-section $160nm \times 0.6nm$ due to the application of a charge current density, $J = 0.1 \times 10^{12} A/m^2$, (b) Domain wall velocity as a function of current density. The domain wall displacement increases linearly with the magnitude of the charge current density and ultimately saturates to a maximum value. The simulation parameters (given in Table 4.2) were obtained experimentally from magnetometric measurements of Ta ($3nm$) / Pt ($3nm$) / CoFe ($0.6nm$) / MgO ($1.8nm$) / Ta ($2nm$) nanostrips [43, 48]. The graphs are in good agreement with [48], (c) Domain wall displacement is directly proportional to the programming current for a fixed duration of the programming pulse.

Table 4.2.
Typical Device Parameters for CoFe-Pt Nanostrips (DW Motion) [49]

Parameters	**Value**
Ferromagnet Thickness	$0.6nm$
Heavy Metal Thickness	$3nm$
Domain Wall Width	$7.6nm$
Saturation Magnetization, M_s	$700\ KA/m$
Spin-Hall Angle, θ_{SH}	0.07
Gilbert's Damping Factor, α	0.3
Exchange Correlation Constant, A	$1 \times 10^{-11} J/m$
Perpendicular Magnetic Anisotropy, K_{u2}	$4.8 \times 10^5 J/m^3$
Effective DMI Constant, D	$-1.2 \times 10^{-3} J/m^2$

the domain wall velocity with input charge current density. The velocity increases linearly with the current density and ultimately reaches a saturation velocity. The graphs are in good agreement with results illustrated in [48] for the same multilayer structure described in this section. Fig. 4.8(c) illustrates the fact that the domain wall displacement is directly proportional to the magnitude of the programming current (for domain wall velocities below the saturation regime).

It is worth noting here that for a given duration of the current through the heavy metal, the domain wall displacement is directly proportional to the magnitude of the current (considering input current range to be less than the saturation regime). The simulations were performed in MuMax3, a GPU accelerated micromagnetic simulation framework [124]. Fig. 4.9 shows the temporal motion of the DMI stabilized

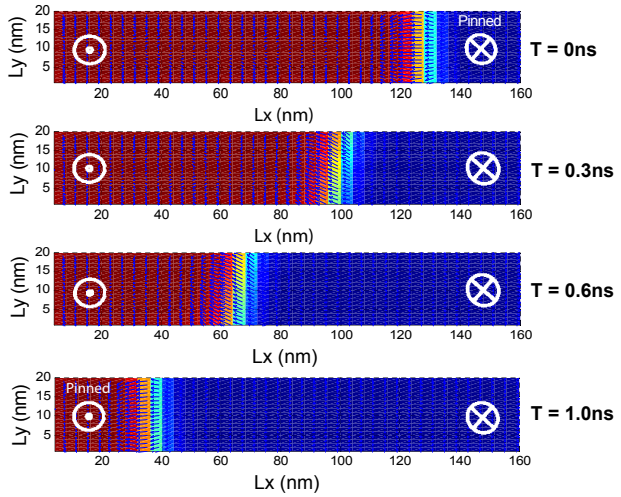

Fig. 4.9. Domain wall motion in the device due to programming current of $25\mu A$ flowing through the HM underlayer for a duration of $1ns$. The FM was taken to be $120nm$ in length surrounded by pinned layers of length $20nm$ on either side. The domain wall is displaced entirely from one edge of the FM to the other edge.

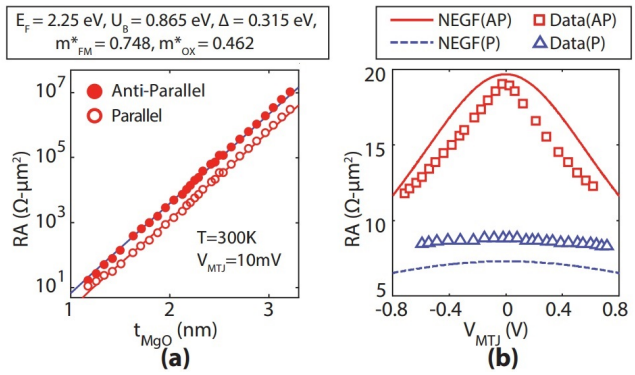

Fig. 4.10. The NEGF based transport simulation framework was calibrated to experimental results illustrated in [16,123]. (a) Device resistance increases with increase in oxide thickness, (b) The AP MTJ resistance decreases with increase in the applied voltage across the MTJ. However, for sufficiently low values of applied voltage ($< 100mV$), the AP resistance variation is extremely small.

domain wall in the device due to a programming current flowing through the HM for a duration of $1ns$.

The tunneling junction simulation framework was calibrated to experimental results illustrated in [16, 123]. For determining the MTJ resistance for a FM with a domain wall separating two oppositely polarized magnetized domains, the NEGF based simulator [25] was modified by considering the parallel connection of three MTJs. The magnetization direction of the FL of the three MTJs were considered parallel, anti-parallel and perpendicular (domain wall) to the pinned layer magnetization. The length of the first two MTJs was varied according to the position of the domain wall while the width of the third MTJ was taken to be equal to the domain wall width. Additionally, as shown in Fig. 4.10, the resistance range of the device can be varied by varying the oxide thickness.

Fig. 4.11 illustrates the variation of the output current provided by the axon transistor with input current provided to the neuron. As the magnitude of input current flowing through the heavy metal underlayer of the neuron increases, the gate voltage, V_G, of the axon transistor decreases as the pull-down resistance of the resistive divider network decreases. The supply voltage of the PMOS axon transistor was taken to be $650mV$. The supply voltage of the resistive divider network ($0.9V$) was optimized such that the corresponding swing in the gate voltage resulted in maximum swing of the output current. As shown in Fig. 4.11(c), the output current provided by the axon transistor increases almost linearly with the input current to the neuron.

Micromagnetic simulations based on typical device parameters obtained experimentally from magnetometric measurements of CoFe-Pt nanostrips [49] demonstrate that the DW can be completely displaced from one edge of a FL (dimension: $80nm \times 20nm$) to the other by $10.6\mu A$ charge current in a duration of $2ns$, thereby resulting in a total "write" and "reset" energy consumption of $0.1fJ$. Such energy-efficient SHE induced DW motion in magnetic multilayer devices can potentially lead to neuronal device structures that would be able to achieve multi-level neuronal states and thereby provide improved cognitive functionalities. It is worth noting here that such

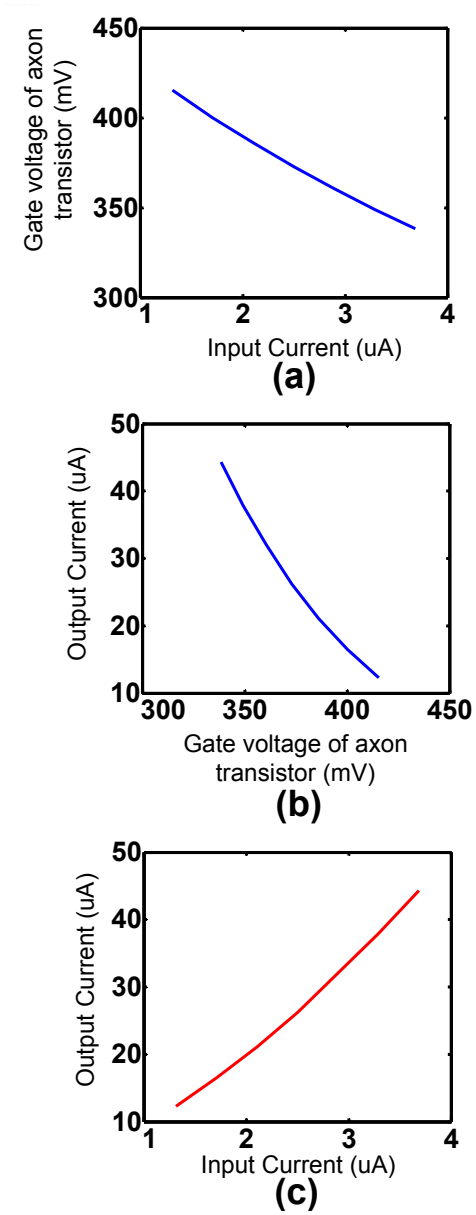

Fig. 4.11. (a) Gate voltage of axon transistor decreases with increase in magnitude of neuron input current, (b) Output current provided by axon transistor reduces with increase in the gate voltage, (c) Output current provided by the axon transistor increases almost linearly with the input current to the neuron. Hence, the neuron transfer function was taken to be linearly increasing with the input, ultimately saturating at a maximum value.

device structures can be also used as multi-level memory units for on-chip cache applications [125] and as receivers for long-distance charge based interconnects [126].

4.1.3 Integrate-Fire "Spiking" Neuron

Let us begin the discussion on "spiking" neurons by noting the similarity between the current integrating property of DW motion and the functionality of an IF "spiking" neuron. Considering input spikes (current pulses) flowing through the HM layer of an FM-HM bilayer structure at different time-steps, the DW would be displaced by an amount proportional to the magnitude of the input current pulse at each time-step whenever a spike is received. The IF functionality can be easily implemented in a slightly modified device structure, shown in Fig. 4.1, where the MTJ is located at the extreme edge of the FL and triggers an output spike (high voltage level at the output inverter) corresponding to the time-step when the DW reaches the other edge of the FL (analogous to neuron membrane potential crossing a particular threshold) [120]. The leak functionality can be implemented by passing a current through the HM in the opposite direction at every time-step.

4.1.4 Stochastic "Spiking" Neuron

As mentioned previously, multi-level neuron states provided by DW motion based spintronic devices can be replaced by binary neuron states obtained from single-domain MTJ structures in case the time-domain magnetization variation of the magnet is considered. The magnetization dynamics of a nano-magnet described by Eq. 2.5 can be reformulated by simple algebraic manipulations as,

$$\frac{1+\alpha^2}{\gamma}\frac{d\widehat{\mathbf{m}}}{dt} = -(\widehat{\mathbf{m}} \times \mathbf{H}_{eff}) - \alpha(\widehat{\mathbf{m}} \times \widehat{\mathbf{m}} \times \mathbf{H}_{eff}) + \frac{1}{q\gamma N_s}(\alpha(\widehat{\mathbf{m}} \times \mathbf{I}_s) - (\widehat{\mathbf{m}} \times \widehat{\mathbf{m}} \times \mathbf{I}_s)) \tag{4.1}$$

Considering the device magnetization to represent the neuron membrane potential, the above equation bears resemblance to LIF characteristics of a "spiking" neuron

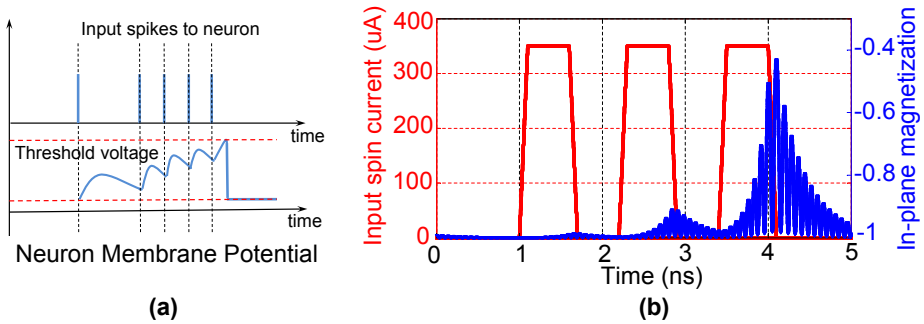

Fig. 4.12. (a) The membrane potential of a biological neuron integrates input spikes and leaks when there is no input. It spikes when the membrane potential crosses the threshold, (b) MTJ neuron dynamics due to the application of three input pulses. The in-plane magnetization starts integrating due to the pulses and then starts leaking once the pulse is removed. The MTJ structure was an elliptic disk of volume $\frac{\pi}{4} \times 100 \times 40 \times 1.5nm^3$ with saturation magnetization of $M_s = 1000KA/m$ and damping factor, $\alpha = 0.0122$.

described in Eq. 3.1. The first two terms on the RHS of Eq. 4.1 represent the leak term in the magnetization state while the last term denotes the integrating term for an input spin current stimuli. Hence, in the presence of an input spike (current pulse), the magnetization starts integrating (switching) towards the opposite stable magnetization state. However, in case the pulse is removed before the entire switching event can take place, the magnetization starts leaking back toward the original magnetization state. In order to reduce the critical switching current requirement and to reduce the input "write" resistance of the neuron, we will consider SHE-induced MTJ switching due to charge current flow through an underlying HM layer (Fig. 4.1). Fig. 4.12 illustrates the leak and integration components of the neuron dynamics for an MTJ elliptic disk due to the application of three successive pulses.

Once the magnet switches to the opposite magnetization state, the neuron has to be "reset" due to the occurrence of the "firing" event. Hence, in order to sense the neuron state, the device is required to be operated in successive "read" and "write" cycles. Each "write" cycle can correspond to a particular time-step of operation of the spiking network. The neuron receives weighted summation of the spike currents

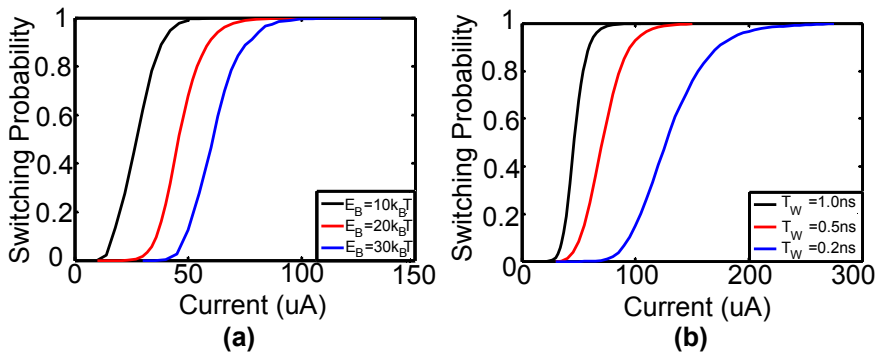

Fig. 4.13. Switching probability of an elliptic IMA magnet of dimensions $\frac{\pi}{4} \times 100 \times 40nm^2$ for CoFe ($1.2nm$) - W ($2nm$) MTJ in response to an input synaptic current at $T = 300K$ (assuming $\sim 50\%$ polarization of spin current generated by the MTJ PL). Such a switching behavior is a direct mapping to the stochastic spiking nature of cortical neurons. (a) The switching probability characteristics shifts to the right with increase in the barrier height. The data have been plotted for $E_B = (10, 20, 30)k_BT$ corresponding to FL thickness values, $t_{FL} = (0.8, 1.2, 1.5)nm$, for pulse width, $T_w = 1ns$ (duration of the "write" cycle), (b) The probability characteristics undergo more dispersion with decrease in the pulse width. The data have been plotted for $T_w = (0.2, 0.5, 1)ns$ corresponding to $E_B = 20k_BT$. The device parameters are mentioned in Table 4.1.

as its input. Since the magnetization dynamics of the MTJ is characterized by thermal noise at non-zero temperatures (in addition to the LIF characteristics discussed previously), the MTJ neuron functionality can be abstracted as a stochastic "spiking" neuron observed in the cortex [89–92], where the neuron "spikes" (switches its state) probabilistically depending on its resultant synaptic input. The variation of spiking probability with input synaptic current is usually described by a non-linear dependence [89–92], similar to the MTJ switching characteristics shown in Fig. 4.13. The switching characteristics of the MTJ neuron in response to the input synaptic current can be varied by changing the energy barrier (or equivalently the FL thickness) and the duration of the synaptic current as illustrated in Fig. 4.13. Unsupervised [6]/ supervised [127] networks enabled by such probabilistic neurons will be discussed in later sections. The "write" cycle is followed by a "read" stage to determine the MTJ

resistance (using the resistive divider driving an inverter described previously). The MTJ is "reset" in case a spike is generated.

Note that most of the current "neuro-mimetic" algorithms are based on deterministic computational units - driven by the fact that the underlying CMOS hardware used to implement such algorithms are deterministic in nature. Past research on hardware implementation of spiking neurons have mainly focused on deterministic neural models, like the Hodgkin-Huxley [78] and Leaky-Integrate-Fire [78] models. Emulation of such neural characteristics require area-expensive CMOS implementations involving more than 20 transistors [128,129] and a direct mapping of spiking neuronal characteristics to a single nanoelectronic device is still missing. However, stochasticity observed in the switching of spintronic technologies can open up new possibilities of envisioning probabilistic neural hardware enabled by stochastic devices. Interestingly, it is believed that the brain is also characterized by noisy stochastic neurons and synapses that perform probabilistic computation [130]. Hence, exploration of such stochastic neuromorphic platforms might open up new avenues at mimicking the biological brain. Note that CMOS based stochastic neural models might be possible [131] but involve significant silicon area and power consumption since they do not offer a direct mapping to the underlying neuroscience mechanisms.

The potential advantages of such a computing framework from hardware implementation perspective is manifold. They allow neural/synaptic state compression (in turn, leading to scaled device implementations) due to the additional time-domain encoding of information probabilistically. In other words, traditionally used multi-bit deterministic neural/synaptic units can be now replaced by single-bit units (enabled by stochastic magnetic devices) where the single-bit device state is updated probabilistically over time. This is also advantageous from scaling perspective since it is expected that the multi-domain spin devices might lose their multi-bit state representation property and therefore may only exhibit binary states. Note that computation using single bit neural activation can be achieved because the loss in information due to bit compression can be encoded in the probabilistic transitions of the single-bit unit

observed over a period of time. Simultaneously, they allow for sub-threshold operation of devices (in order to exploit the stochastic switching regime these devices have to be operated below the critical current requirement for deterministic switching), thereby leading to energy consumption reductions.

This work was the first proposal on using a magnet as a "stochastic bit" (exploiting the entire range of analog probabilistic switching regime of a nanomagnet) – behaving as a conditional random number generator producing a probabilistic output pulse stream with the probability being conditioned on the magnitude of the input stimulus and can be found in Ref. [6] for neural inference applications. Thereafter, this was followed by a plethora of work exploring several neuromorphic as well as other unconventional computing paradigms enabled by such magnetic "stochastic bits" [7, 127, 132–137]. The inherent stochasticity of spin devices can also potentially find use as on-chip temperature sensors [138] (discussed in Appendix B) and in logic implementation [139, 140]. However, note that the delay incurred in probabilistic logic implementation using such stochastic magnets would be significantly higher than a corresponding deterministic CMOS logic implementation since the average output of the logic has to be observed over a large enough time window to infer the output with maximum probability.

Let us consider the energy consumption of such a stochastic neuron. The average neuronal energy consumption determined for the input current ($\sim 71\mu A$) necessary to switch an elliptic IMA magnet of dimensions $\frac{\pi}{4} \times 100 \times 40nm^2$ for CoFe ($1.2nm$) - W ($2nm$) MTJ with a probability of 0.5 is evaluated to be $\sim 1fJ$ for a "write" cycle duration of $0.5ns$ [6]. In contrast, state-of-the-art designs of CMOS neurons result in energy consumption in the range of pJ per spike ($267pJ$ reported in Ref. [141] and $41.3pJ$ reported in Ref. [142]).

Proof-of-concept experiments demonstrating stochastic magnetization switching in ferromagnet-heavy metal bilayer structures have been also demonstrated [134]. Fig. 4.14(a) depicts a $1.2\mu m$ wide Hall-bar structure consisting of Ta ($10nm$) / CoFeB ($1.3nm$) / MgO ($1.5nm$) / Ta ($5nm$) (from bottom to top) material stack

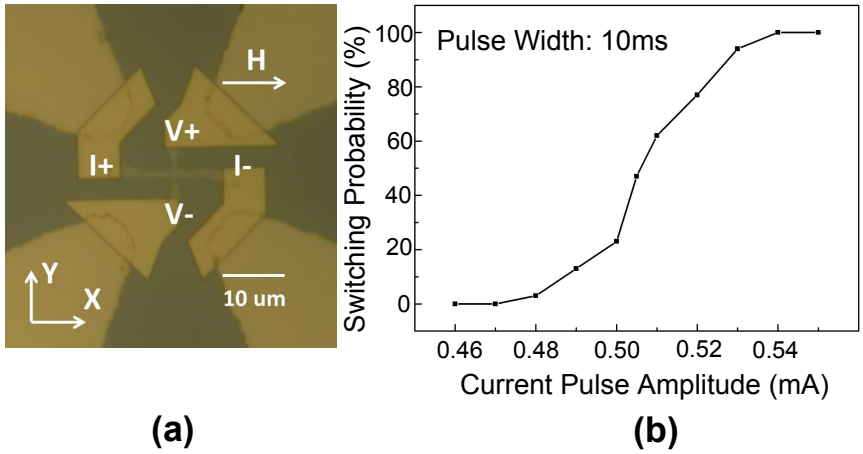

Fig. 4.14. (a) Hall-bar structure consisting of Ta ($10nm$) / CoFeB ($1.3nm$) / MgO ($1.5nm$) / Ta ($5nm$) (from bottom to top) material stack [134]. Input current flows between terminals $I+$ and $I-$ while the magnetization state is detected by change in the anomalous Hall-effect resistance measured between terminals $V+$ and $V-$, (b) Experimental measurements of the switching probability of the Hall-bar with variation in amplitude of the current pulse flowing through the heavy metal underlayer for a fixed pulse width of $10ms$ [134].

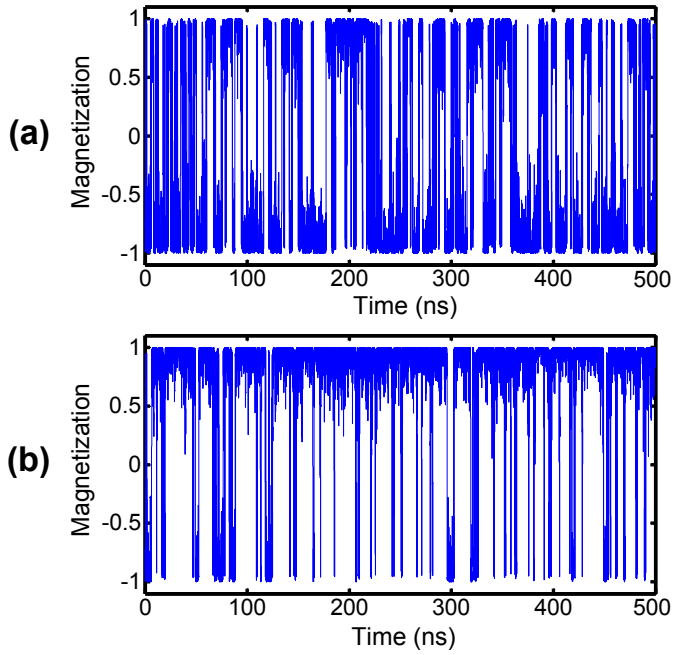

Fig. 4.15. Simulation study of the random telegraphic switching of a superparamagnet of barrier height $1k_BT$ under (a) no bias and (b) under a bias current of $1.5\mu A$ [143].

with perpendicular magnetic anisotropy. Input charge current flows between $I+$ and $I-$ terminals while the final stable magnetization state is determined by the anomalous Hall effect resistance between terminals $V+$ and $V-$. Note that the switching is performed in the presence of an external in-plane magnetic field since the perpendicular anisotropy magnet cannot be solely switched by in-plane spins generated by current flowing through the heavy metal underlayer. Fig. 4.14(b) represents the experimental measurements for the switching probability of the magnetic stack with variation in the magnitude of the current pulse being used for switching (with pulse width being fixed at $10ms$). Note that the non-linear variation of the switching probability of the magnet with the magnitude of the current pulse flowing through the heavy metal underlayer resembles theoretical simulations depicted in Fig. 4.13. Such proof-of-concept experiments can be easily extended to device structures where a Tunnel Junction is used as the read-out mechanism (exhibiting 2-3 times larger resistance variation in comparison to Hall-bar structures) for compatibility with peripheral CMOS circuitry.

The barrier height of the magnet (defined as the product of the magnetic anisotropy and the magnet volume) determines the current range that can be used for stochastic magnet switching. As the magnet volume is scaled down, the magnitude of the current range useful for stochastic switching reduces, thereby increasing the energy efficiency of the device. However, in highly scaled devices having barrier height $\sim 1k_BT$, the magnet undergoes random telegraphic switching in the nano-second time scale. Fig. 4.15(a) depicts the magnetization dynamics of a $1k_BT$ magnet under no bias current flowing through the HM. The average magnetization over a long enough time window is approximately 0. On the other hand, the dwell time in either one of the stable states can be modulated in the presence of an external bias current (Fig. 4.15(b)). Note that such superparamagnetic MTJs operating in the telegraphic regime has been referred to as "p-bits" by authors in Refs. [139, 140]. Experiments have demonstrated telegraphic switching in MTJ stacks [136, 144, 145], with barrier height as low as

$\sim 11k_BT$ [146]. Scaling magnets to even lower barrier heights ($< 5k_BT$) might be difficult from fabrication perspective.

The potential advantage of utilizing random telegraphic switching as the stochastic computing element lies in its energy efficient operation. While $\sim 71\mu A$ current is required for $0.5ns$ to switch a $20k_BT$ barrier height magnet with 50% probability [127], thereby leading to an I^2Rt energy consumption of $\sim 1fJ$, zero bias current is required to achieve 50% switching probability in a $\sim 1k_BT$ device. Note that, in practical device implementation, 50% switching probability may not be achieved exactly at zero bias current due to presence of device imperfections, stray fields and magnetic coupling between elements. Also, the device being highly sensitive to noise and variations, require appropriate peripheral circuits for proper functionality. These design tradeoffs will be explained in details in the succeeding sections.

We would like to conclude this section by noting the two main device structures that will be used for the rest of this discussion - the DW motion based bilayer structure used as a "non-step"/IF "spiking" neuron and the single-domain MTJ based device used as a stochastic "spiking" neuron. These devices will be used to implement deterministic/probabilistic STDP in multi-/single-bit synapses respectively in the next section.

4.2 Spin-Torque Synapses

4.2.1 Spike-Timing Dependent Plasticity

The mechanism that lends cognitive capabilities to networks of interconnected neurons is the plasticity of the synaptic junctions. For a vast majority of these plasticity mechanisms, the synaptic conductance is modulated depending on the time-difference between the spikes of the neurons it connects. Let us first consider the implementation of STDP in the DW motion based device structure introduced in the previous section. The device conductance between terminals T_1 and T_3 is dominated by the MTJ conductance which varies linearly with the domain wall position. Let us

denote the conductance of the device when the FM magnetization is P(AP) to the PL as $G_P(G_{AP})$, i.e. the domain wall is at the extreme right (left) of the FM. Thus, for an intermediate position of the domain wall at a location x from the left-edge of the MTJ, the device conductance between terminals T_1 and T_3 is given by,

$$G_{eq} = G_P.\frac{x}{L} + G_{AP}.\left(1 - \frac{x}{L}\right) + G_{DW} \tag{4.2}$$

where L denotes the length of the MTJ excluding the domain wall width and G_{DW} represents the conductance of the wall region. For a given time duration, it can be shown from micromagnetic simulations that the programming current magnitude, J, is directly proportional to the DW displacement, Δx [4, 119, 120]. Since, $\Delta G \propto \Delta x \propto J$, the programming current should vary in a similar manner as the variation of the synaptic plasticity (ΔG variation) with spike timing difference of connecting neurons. Such an intuitive variation of programming current variation for synaptic plasticity implementation is again a functionality offered by the decoupled "write" and "read" current paths of the proposed device structure. The programming current flows through the constant HM resistance and is not impacted by the present synaptic MTJ conductance magnitude. This results in simple peripheral circuit design as well for implementing the desired plasticity rule. In contrast, conductance change in traditional two terminal memristors depend on the history of the programming pulses.

The operating mode of the synapse, i.e. the spike transmission ("read") or the programming ("write") mode is determined by the control signal POST. The access transistors causes the isolation of the appropriate device terminals during "write"/"read" operations. When the POST signal is deactivated, terminals T_1 and T_3 of the device are activated and spike voltage signals can be transmitted from the pre-neuron (V_{SPIKE}) signal through the MTJ conductance to provide an equivalent amount of synaptic current to the post-neuron circuit (connected to terminal T_3). When the POST signal is activated the "write" current path through terminals $T_2 - T_3$ gets activated and the device state is updated depending on the amount of synaptic current being supplied by the interfaced M_{STDP} transistor. Note that the terminal T_3 is con-

nected to GND during "write" mode of operation of the device and is disconnected from the post-neuron.

Let us now consider the learning mechanism in the spintronic device in more details. The most common learning rule dictates an exponential reduction in conductance change with increase in the value of spike timing difference. The exponential variation of current through the HM can be obtained by biasing the interfaced transistor M_{STDP} in the sub-threshold regime ($V_{gs} < V_t$ and $V_{ds} > 4U_T$, V_t: threshold voltage and U_T: thermal voltage) since the current flowing through the transistor will vary exponentially with the gate to source voltage. Thus, for a linear increase of the gate voltage (PRE signal) every time a pre-neuron spikes, the peripheral programming transistor will be driven from cut-off to the sub-threshold saturation region when the POST signal is activated and an appropriate programming current (magnitude varying exponentially with timing difference of pre- and post-neuron spikes) should flow through the HM. The duration of the programming current is determined by the duration of the POST signal and the magnitude is determined by the current supplied by the bias-point (PRE signal) of the M_{STDP} transistor. It is worth noting here that the relationship $\Delta G \propto \Delta x \propto J$ is valid when the magnitude of the programming current J remains constant during the programming duration. This is achieved by ensuring that the rise time of the gate voltage PRE of the M_{STDP} transistor, or equivalently the STDP time constants, are much longer than the programming time durations (duration of POST signal) such that the current flowing through the HM of the spintronic synapse remains approximately constant. We consider STDP timing constants in the range of $\sim \mu s$ whereas the duration of the POST signal was $1ns$. For a linearly rising gate voltage from 0.2 to $0.6V$ of the M_{STDP} transistor (drain voltage being at $0.6V$), exponential current dynamics was observed due to transistor operation in the sub-threshold saturation regime. The linearly rising gate voltage can be easily implemented by charging a capacitor with a constant input current source everytime a pre-neuron spikes [5]. Fig. 4.17 shows the response of the programming circuit for the case when the programming current path is active

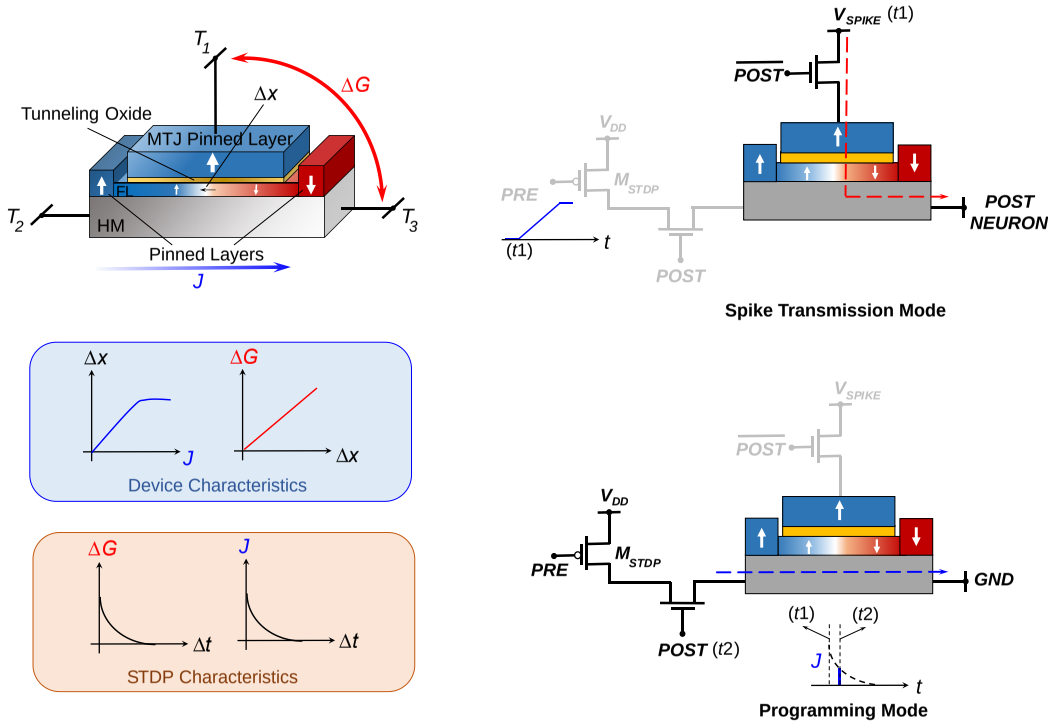

Fig. 4.16. Spike-Timing Dependent Plasticity: Magnitude of current flowing through the underlying HM, J, causes a proportionate displacement, Δx, in the DW position, which causes a change, ΔG, in the device conductance between terminals T_1 and T_3. The device characteristics illustrate that the programming current magnitude is directly proportional to the amount of conductance change, provided the DW velocity is below the saturation value. STDP characteristics is implemented by biasing the transistor M_{STDP} in subthreshold saturation regime in order to achieve the exponential current dynamics through the HM layer. The spike transmission and programming current modes are depicted in the right hand panel where the PRE and V_{SPIKE} signals are activated at pre-neuron firing event at time t_1. POST signal, activated at post-neuron firing event at time t_2, samples the appropriate amount of programming current corresponding to the spike timing difference.

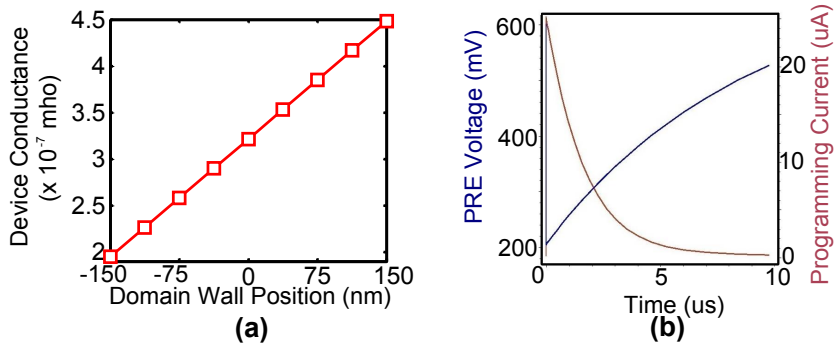

Fig. 4.17. (a) Linear variation of device conductance with domain wall position, (b) Programming circuit simulation to generate the STDP characteristics in the proposed spintronic synapse.

throughout the simulation time. The duration of the time window can be varied by changing the capacitance value. From device simulations, it was determined that a maximum current of $\sim 80\mu A$ is required to displace the domain wall from one edge of the FM to the other edge (for a synapse of dimensions $320nm \times 20nm$. Hence the maximum amount of energy consumption involved in synapse programming is $\sim 48fJ(600mV \times 80\mu A \times 1ns)$ per synaptic event.

The discussion so far has been limited only to the implementation of the positive timing window of the STDP curve. In order to implement both the timing windows, an additional NMOS transistor is utilized in parallel to the PMOS transistor M_{STDP}. Two separate learning circuitries are utilized for each of the timing windows which consists of a capacitor being charged by a current source. Every-time the pre-neuron spikes, the circuit for the negative timing window is reset first such that the gate voltage of the NMOS transistor starts increasing with time. Since the drain of the NMOS transistor is negative (in order to pass current through the HM in the opposite direction for the negative timing window), the current supplied by the NMOS transistor increases as the delay of activation of the POST signal increases. In order to account for both the timing windows, the POST signal is activated after a delay of the negative timing window in order to sample the programming current

contributions from the learning circuits for the positive and negative timing windows. Hence if the post-neuron spikes before the pre-neuron (negative window), the programming path will be activated during the time duration the gate voltage of the NMOS transistor is rising to pass a negative current through the device and thereby reduce the device conductance. After the duration of the negative timing window, the learning circuit for the positive timing window is reset and the POST signal is activated during this window only for a potentiation event, i.e. post-neuron "spiking" after pre-neuron. Note that the learning circuitry which consists of the capacitor and the current source transistors can be shared across all the synapses being driven by the same pre-neuron. Discussions of crossbar arrays of such spintronic synapses for SNN implementations with on-chip learning capabilities will be discussed in the next section along with more detailed timing diagrams to explain the implementation of the positive and negative timing windows. Detailed operations explaining the implementation of synaptic plasticity is explained in Fig. 4.16.

As discussed previously, the "read" operation of the spintronic device or the synaptic scaling operation is a direct consequence of Kirchoff's law. For a constant magnitude of the spike signal, V_{SPIKE}, the current flowing through the synapse gets multiplied by the synaptic conductance. However, it is worth noting here that the conductance of the device is a function of the applied voltage as well. The resistance in the AP state is a much stronger function of the applied voltage than the P state and reduces by a significant amount as the applied voltage increases. Hence, higher the magnitude of the spike signal lower is the ratio of the maximum to the minimum synaptic conductance achievable. Note that higher synaptic weight ratios are desirable for achieving higher accuracy in pattern recognition workloads. Hence in order to maximize the discrimination between the two synaptic states, it is important to operate the synapses at low operating voltages less than $100mV$. This can be easily achieved by interfacing such synapses with magneto-metallic spin neurons (which inherently require low currents for switching) [6] or CMOS neurons operating in the subthreshold saturation regime [129]. Operating the synapses at lower voltages is

Table 4.3.
Comparison of proposed spintronic synapse with other CMOS and post-CMOS implementations

Device	**Dimensions**	**Programming Energy/ Operating Voltage**	**Programming Time**	**Terminals**	**Programming Mechanism**
GeSbTe memristor [8]	$40nm$ mushroom and $10nm$ pore	Average $2.74pJ$/ event	$60ns$	2	Programmed by Joule heating (Phase change)
GeSbTe memristor [9]	$75nm$ electrode diameter	$50pJ$ (reset) & $0.675pJ$ (set)	$10ns$	2	Programmed by Joule heating (Phase change)
Ag/AgInSbTe/Ag chalcogenide memristor [147]	$100\mu m$ x $100\mu m$	Threshold voltage - $0.3V$	$5\mu s$	2	Programmed by Joule heating (Phase change)
Ag-Si memristor [148]	$100nm$ x $100nm$	Threshold voltage - $2.2V$	$300\mu s$	2	Movement of Ag ions
FeFET [149]	Channel length - $3\mu m$	Maximum gate voltage - $4V$	$10\mu s$	3	Gate voltage modulation of ferroelectric polarization
Floating gate transistor [150]	$1.8\mu m/0.6\mu m(0.35\mu m$ CMOS technology)	$V_{dd} - 4.2V$ & Tunneling Voltage -$15V$	$100\mu s$ (injection) & $2ms$ (tunneling)	3	Injection and tunneling currents
SRAM synapse [128]	$0.3\mu m^2$ ($10nm$ CMOS technology)	Average $328fJ$ (4-bit synapse)	-	-	Digital counter based circuits
Spintronic synapse	Ferromagnet dimensions - $320nm$ x $20nm$	Maximum $48fJ$/ event	$1ns$	3	Spin-orbit torque

more important for "non-spiking" networks since the neuron inputs need to be analog in nature. Hence the voltages applied across the synapse would be different for different inputs, thereby causing the synaptic weight to be a function of the applied input. Thus it is imperative to operate the synapses at low voltages from a functional perspective. Lower operating voltage assists in reducing the maximum "read" current flowing through the device which, in turn, determines the device width. Assuming that the main spin torque exerted on the FL due to the "read" current being from SOT generated by the HM, the device width can be scaled up to ensure that no DW depinning occurs for the maximum allowable magnitude of the "read" current. The length of the synapse would be determined by the maximum number of states required from algorithm perspective.

Table 4.3 provides a comparative analysis of our spintronic synapse (calibrated to experiments performed in Ref. [49]) with other proposed synaptic devices. Synaptic device structures based on emerging post-CMOS technologies [8, 9, 147, 148] are usually two-terminal devices and do not offer de-coupled programming and read current paths. Three terminal synaptic devices based on FeFET [149] and floating gate transistors [150] have been also proposed. However, the programming in such devices is usually accomplished through the gate terminal and a high gate voltage is usually applied across a very thin oxide [149, 150] leading to reliability issues, in addition to associated high power consumption. Programming is also relatively slow in such three terminal synaptic devices [149, 150]. SRAM based synapses have been also proposed for digital CMOS based SNN design [128]. However, for implementing 1 bit of the synapse, an 8-T SRAM cell has to be used, thereby leading to significant area overhead for implementation of a single synapse [128]. In addition, learning circuits will involve multiple digital counters and will be more area/power consuming than our proposed design. As shown in Table 4.3, such SOT induced plastic CoFe-Pt synapses demonstrate programming energies per synaptic event which is an order of magnitude lower than programming energies reported for a 4-bit SRAM synapse at $10nm$ technology node [128]. Interestingly, analysis performed by Rajendran *et al.* revealed that

although analog neuromorphic systems based on typical emerging memristive technologies will provide area benefits at scaled technologies, power consumption would be twice as high in comparison to its digital counterpart [128]. This is because resistive technologies like GeSbTe [8, 9]/Ag-Si [148] devices are usually characterized by high threshold voltages $\sim V$ and involve much higher programming energies in the range of $\sim pJ$ and programming time durations in the range of $\sim \mu s$. Low-power on-chip learning enabled by such spintronic synapses can potentially bridge this energy in-efficiency gap.

4.2.2 Probabilistic Synaptic Learning

The complementary version of single-bit probabilistic STDP can be similarly implemented using the single-domain MTJ-HM bilayer structures discussed previously. While Vincent *et al.* explored a simplified version of probabilistic STDP where the probability of synaptic state change was constant for positive and negative timing windows [99], we proposed crossbar architectures of such MTJ-enabled stochastic learning where the update probability varied exponentially with spike timing in accordance to original STDP formulations [7]. As explained in Fig. 4.18, this can be achieved by a similar framework described for the DW motion based devices where an additional interfaced transistor M_{STDP}, biased in the saturation regime, is driven by a linearly increasing gate voltage every time the pre-neuron spikes [7]. Another potential advantage of probabilistic learning is below-threshold operation of devices. Since the update probability is maintained typically below 0.1 to maintain "non-greedy" learning [7], operating current and voltage requirements of such devices are significantly reduced.

4.2.3 Volatile Synaptic Learning

In order to implement frequency dependent volatile synaptic learning, a nanoelectronic device is required that exhibits only two stable resistive states and undergoes

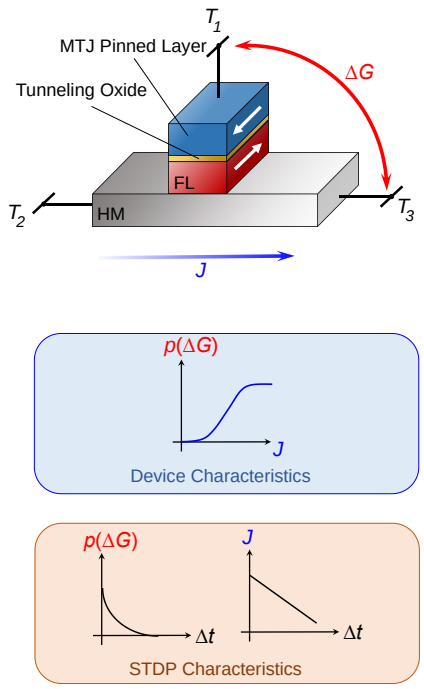

Fig. 4.18. Probabilistic STDP learning: This can be achieved in a similar fashion in mono-domain MTJ synapses by exploiting sigmoidal stochastic device switching characteristics. In the low switching probability regime (for ensuring non-greedy learning), the "write" current reduces linearly with spike timing to emulate exponential probabilistic STDP characteristics. This is ensured by biasing M_{STDP} in the saturation regime.

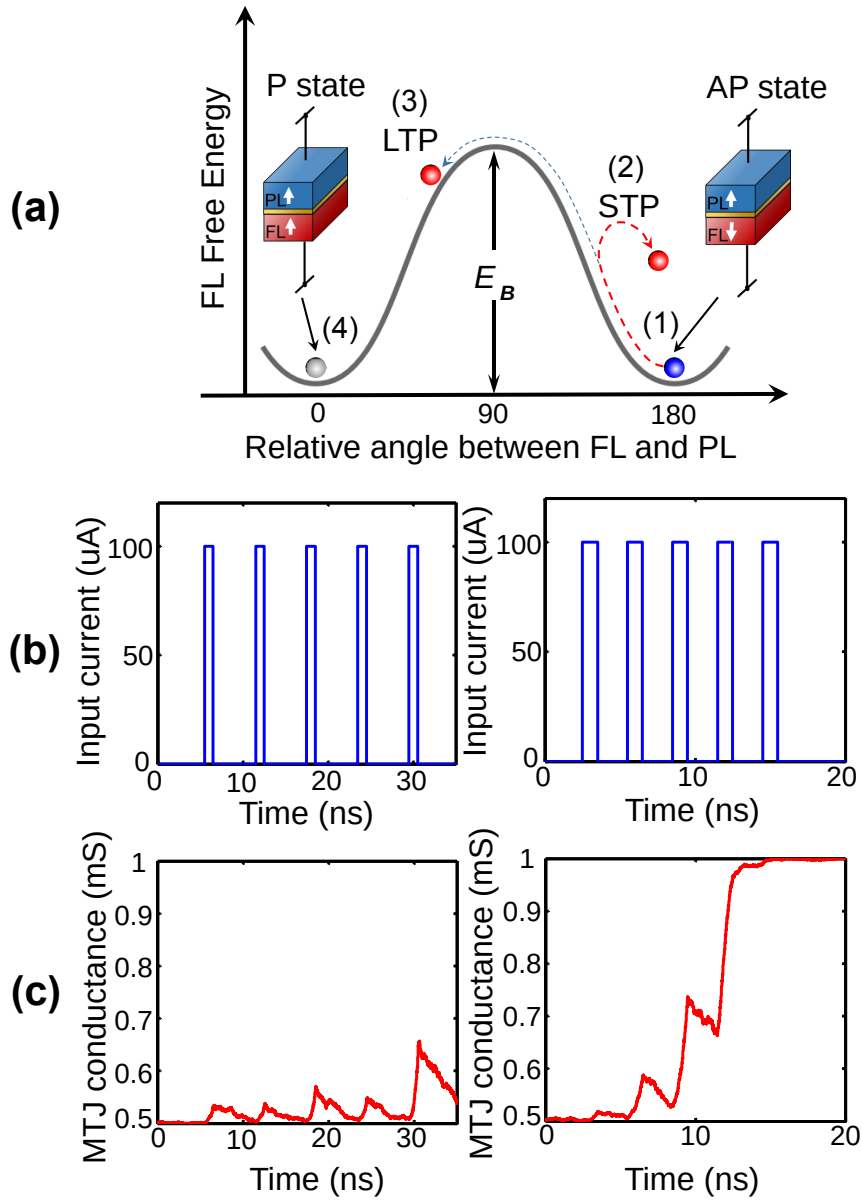

Fig. 4.19. Frequency dependent volatile synaptic learning: A mono-domain MTJ is characterized by two stable states separated by an energy barrier E_B. If the frequency of the input stimuli is not enough, the MTJ is unable to cross the metastable position at 90° relative angle between FL and PL and stabilizes back to the initial magnetization state, exhibiting STP. As the stimuli frequency increases, the MTJ exhibits a much higher probability of switching to the other stable state, thereby exhibiting LTP [151].

meta-stable state transitions whenever an input stimulus is received. The apparent spintronic device that can be directly mapped to such a functionality is the mono-domain MTJ where the spin-polarization of incoming electrons can be thought to be analogous to the release of neurotransmitters in a biological synapse.

The STP and LTP mechanisms exhibited in the MTJ due to the spin-polarization of the incoming electrons can be explained by the energy profile of the FL of the MTJ. Let the angle between the FL magnetization, $\widehat{\mathbf{m}}$, and the PL magnetization, $\widehat{\mathbf{m}}_P$, be denoted by θ. The FL energy as a function of θ has been shown in Fig. 4.19(a) where the two energy minima points ($\theta = 0^0$ and $\theta = 180^0$) are separated by the energy barrier, E_B. During the transition from the AP state to the P state, the FL has to transition from $\theta = 180^0$ to $\theta = 0^0$. Upon the receipt of an input stimulus, the FL magnetization proceeds "uphill" along the energy profile (from initial point 1 to point 2 in Fig. 4.19(a)). However, since point 2 is a meta-stable state, it starts going "downhill" to point 1, once the stimulus is removed. If the input stimulus is not frequent enough, the FL will try to stabilize back to the AP state after each stimulus. However, if the stimulus is frequent, the FL will not get sufficient time to reach point 1 and ultimately will be able to overcome the energy barrier (point 3 in Fig. 4.19(a)). It is worth noting here, that on crossing the energy barrier at $\theta = 90^0$, it becomes progressively difficult for the MTJ to exhibit STP and switch back to the initial AP state. This is in agreement with the psychological model of human memory where it becomes progressively difficult for the memory to "forget" information during transition from STM to LTM. Hence, once it has crossed the energy barrier, it starts transitioning from the STP to the LTP state (point 4 in Fig. 4.19(a)). The stability of the MTJ in the LTP state is dictated by the magnitude of the energy barrier. The lifetime of the LTP state is exponentially related to the energy barrier [18]. For instance, for an energy barrier of $31.44k_BT$ used in this work, the LTP lifetime is $\sim$ 12.4 hours while the lifetime can be extended to around $\sim$ 7 years by engineering a barrier height of $40k_BT$. The lifetime can be varied by varying the energy barrier, or equivalently, volume of the MTJ. The phenomena can be also

explained by the leaky-integrate time-varying LLG dynamics of the magnetic FL. In the presence of an input spike (current pulse), the magnetization starts integrating (switching) towards the opposite stable magnetization state. However, in case the pulse is removed before the entire switching event can take place, the magnetization starts leaking back towards the original magnetization state. It is worth noting here that, like traditional semiconductor memories, magnitude and duration of the input stimulus will definitely have an impact on the STP-LTP transition of the synapse. However, frequency of the input is a critical factor in this scenario. Even though the total flux through the device is same, the synapse will conditionally change its state if the frequency of the input is high. We verified that this functionality is exhibited in MTJs by performing LLG simulations (including thermal noise at $300K$) for a magnet of dimensions $\frac{\pi}{4} \times 40 \times 40 \times 1.5nm^3$ and parameters mentioned in Table 4.1. While we are not considering spin-orbit torque induced switching in these simulations, the results can be easily extended to FM-HM multilayers. 50% spin polarization strength was considered by the PL of the MTJ. The P and AP conductance states of the MTJ was considered to be $0.5mS$ and $1mS$. As shown in Fig. 4.19(b), the MTJ conductance undergoes meta-stable transitions (STP) and is not able to undergo LTP when the time interval of the input pulses is large ($6ns$). However, on frequent stimulations with time interval as $3ns$, the device undergoes LTP transition incrementally. Fig. 4.19(b) and (c) illustrates the competition between memory reinforcement and memory decay in an MTJ structure that is crucial to implement STP and LTP in the synapse.

We demonstrate simulation results to verify the STP and LTP mechanisms in an MTJ synapse depending on the time interval between stimulations. The MTJ was subjected to 10 stimulations, each stimulation being a current pulse of magnitude $100\mu A$ and $1ns$ in duration. As shown in Fig. 4.20, the probability of LTP transition and average device conductance at the end of each stimulation increases with decrease in the time interval between the stimulations. The dependence on stimulation time interval can be further characterized by measurements corresponding to

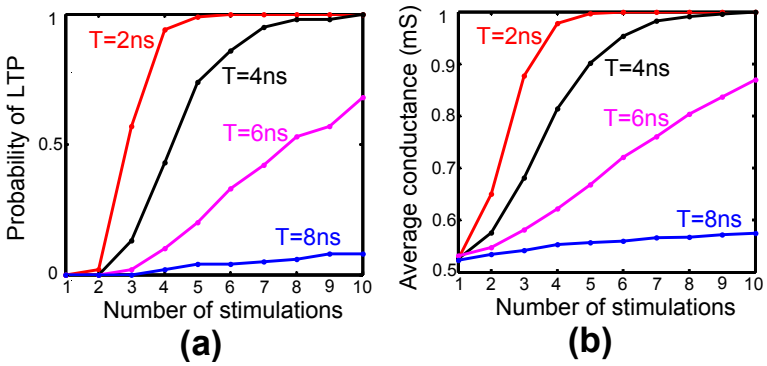

Fig. 4.20. (a) Stochastic LLG simulations with thermal noise performed to illustrate the dependence of stimulation interval on the probability of LTP transition for the MTJ. The MTJ was subjected to 10 stimulations, each stimulation being a current pulse of magnitude $100\mu A$ and $1ns$ in duration. However, the time interval between the stimulations was varied from $2ns$ to $8ns$. While the probability of LTP is 1 for a time interval of $2ns$, it is very low for a time interval of $8ns$, at the end of the 10 stimulations, (b) Average MTJ conductance plotted at the end of each stimulation. As expected, the average conductance increases faster with decrease in the stimulation interval. The results have been averaged over 100 LLG simulations.

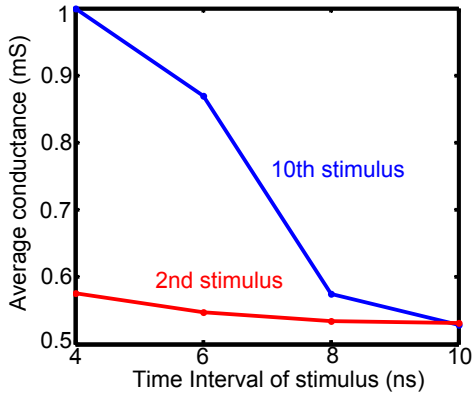

Fig. 4.21. PPF (average MTJ conductance after 2nd stimulus) and PTP (average MTJ conductance after 10th stimulus) measurements in an MTJ synapse with variation in the stimulation interval. The results are in qualitative agreement to PPF and PTP measurements performed in frog neuromuscular junctions [152, 153].

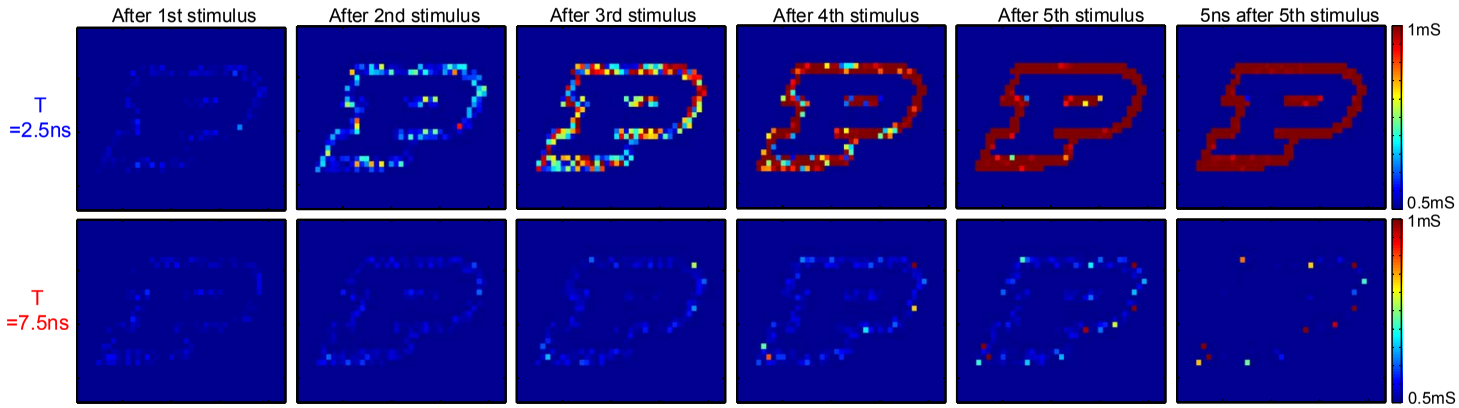

Fig. 4.22. STM and LTM transition exhibited in a 34×43 MTJ memory array. The input stimulus was a binary image of the Purdue University logo where a set of 5 pulses (each of magnitude $100\mu A$ and $1ns$ in duration) was applied for each ON pixel. While the array transitioned to LTM progressively for frequent stimulations at an interval of $T = 2.5ns$, it "forgot" the input pattern for stimulation for a time interval of $T = 7.5ns$.

paired-pulse facilitation (PPF: synaptic plasticity increase when a second stimulus follows a previous similar stimulus) and post-tetanic potentiation (PTP: progressive synaptic plasticity increment when a large number of such stimuli are received successively) [152, 153]. Fig. 4.21 depicts such PPF (after 2nd stimulus) and PTP (after 10th stimulus) measurements for the MTJ synapse with variation in the stimulation interval. The measurements closely resemble measurements performed in frog neuromuscular junctions [152] where PPF measurements revealed that there was a small synaptic conductivity increase when the stimulation rate was frequent enough while PTP measurements indicated LTP transition on frequent stimulations with a fast decay in synaptic conductivity on decrement in the stimulation rate. Hence, stimulation rate indeed plays a critical role in the MTJ synapse to determine the probability of LTP transition.

The psychological model of STM and LTM utilizing such MTJ synapses was further explored in a 34×43 memory array. The array was stimulated by a binary image of the Purdue University logo where a set of 5 pulses (each of magnitude $100\mu A$ and $1ns$ in duration) was applied for each ON pixel. The snapshots of the conductance values of the memory array after each stimulus have been shown for two different stimulation intervals of $2.5ns$ and $7.5ns$ respectively. While the memory array attempts to remember the displayed image right after stimulation, it fails to transition to LTM for the case $T = 7.5ns$ and the information is eventually lost $5ns$ after stimulation. However, information gets transferred to LTM progressively for $T = 2.5ns$. It is worth noting here, that the same amount of flux is transmitted through the MTJ in both cases. The simulation not only provides a visual depiction of the temporal evolution of a large array of MTJ conductances as a function of stimulus but also provides inspiration for the realization of adaptive neuromorphic systems exploiting the concepts of STM and LTM.

There have been recent proposals of other emerging devices that can exhibit such STP-LTP mechanisms like Ag_2S synapses [154] and WO_X memristors [153, 155]. However, it is worth noting here, that input stimulus magnitudes are usually in the

range of volts (1.3V in [153] and 80mV in [154]) and stimulus durations are of the order of a few msecs (1ms in [153] and 0.5s in [154]). In contrast, similar mechanisms can be exhibited in MTJ synapses at much lower energy consumption (by stimulus magnitudes of a few hundred μA and duration of a few ns). We believe that this work will stimulate proof-of-concept experiments to realize such MTJ synapses that can potentially pave the way for future ultra-low power intelligent neuromorphic systems capable of adaptive learning.

5. SPIN BASED NEUROMORPHIC CIRCUITS AND SYSTEMS

5.1 All-Spin Neural Networks for Deterministic Inference

Irrespective of the network connectivity (FCN/CNN) the main computing kernel involved in such computing schemes can be mapped to a parallel dot-product implementation followed by neural processing. Let us begin the discussion in this section by considering spintronic synapses to be the multi-bit DW motion based device structures driving similar IF "spiking" neurons discussed in the previous section. For this subsection, we will assume offline learning of such networks where the synaptic weights are pre-determined by backpropagation [84, 85] and on-chip learning functionality is not involved. Enabling on-chip intelligence in SNNs will be illustrated in the next subsection.

The main underlying principle for implementation of the parallel-dot product computing kernel is based on the very simple and intuitive application of Kirchoff's laws. Considering a dot-product operation between m inputs and n outputs, the computation can be represented by a crossbar array of dimension $m \times n$ (Fig. 5.1). At each cross-point of the array, a spintronic synaptic device is present whose conductance encodes the value of the corresponding synaptic weight. Whenever a "spike" is received at a particular input, a high voltage signal is applied along the row while a no "spike" is represented by a low voltage signal. Assuming all the vertical lines of the array to be at ground potential, the current flowing through each crosspoint will be weighted by the synaptic conductance and get summed up along the column to provide a resultant input current (representing the dot product) to the neuron for further processing. Note that this is a major advantage of such "in-memory" computing architectures since the synaptic weights can be stored locally in the non-volatile

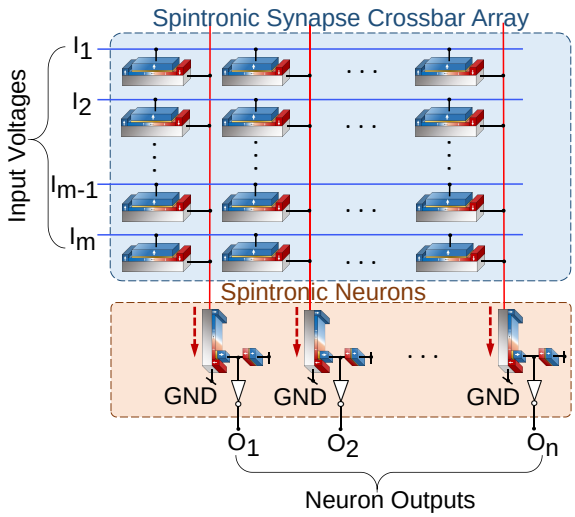

Fig. 5.1. All-Spin Neural Networks: A particular layer of a neural network with m inputs and n outputs can be mapped to a crossbar array of dimension $m \times n$. At a particular time-step, the rows corresponding to those inputs which have spiked are asserted a HIGH voltage level while zero voltage is applied along the rows for the "non-spiking" inputs. Since the input "write" resistance of the magneto-metallic spin-neurons is low, the resultant current provided by each column of the crossbar array as input to the corresponding spin-neuron equals approximately the dot-product of the neuron inputs and the corresponding synaptic weights.

resistive states of the spintronic devices arranged in a crossbar fashion. In contrast, CMOS based neuromorphic architectures involve significant energy consumption due to memory leakage and memory access in order to fetch the synaptic weight values to the neural computing core for each input spike.

In order to maintain the vertical columns at ground potential, prior work has mostly considered interfacing the crossbar arrays with analog CMOS neurons that can maintain the vertical columns at virtual ground [73]. Note that the basic functionality that we are exploiting in the design of spintronic neuronal device structures is also that of a programmable resistor. However, the main reason such device structures are suitable for neural as well as synaptic operations is due to the decoupled nature of the "write" and "read" current paths. The input resistance of the device

during the "write" operation is mainly the low HM resistance and hence the synaptic input current from the crossbar array is not required to flow through the MTJ oxide. Further such magneto-metallic spin-neurons are characterized inherently by low switching current requirement thereby minimizing the terminal voltage drop across such devices. This is the main reason attributed to the usage of other two terminal resistive memories [8, 9, 148] primarily as synaptic devices. Interfacing such two terminal memristive crossbar arrays with two terminal memristive neurons would be potentially difficult resulting in erroneous dot product computation since the vertical columns of the array would be no longer maintained at ground potential (due to the high threshold voltages and resistances of such memory technologies). In addition to providing the flexibility of implementing neuronal and synaptic devices by the same technology, spintronic neurons enable low power operation of the spintronic crossbar array due to low switching current requirements of such magneto-metallic devices. In contrast, analog CMOS neuron implementations typically require the crossbar arrays to be run at a much higher voltage.

Let us now consider the operation of the crossbar array in more details. Each time-step of SNN operation consists of a neuron "write" cycle followed by the "read" and "reset" cycles. In order to implement bipolar weights, two rows (V_{i+} and V_{i-}) are used for each input V_i. When the input V_i assumes a logic value of '0'(no "spike"), then '0' voltage level is applied to both the inputs. However, when V_i assumes a logic value of '1'("spike"), then voltage V_o (less than $100mV$) is applied to the row corresponding to V_{i+} and $-V_o$ is applied to the row corresponding to V_{i-}. If the weight $w_{i,j}$ for the j-th neuron and input V_i is positive, then the conductance corresponding to V_{i+} is programmed to $G_{i,j+} = w_{i,j}.G_o$ (G_o is the mapped conductance for unity synaptic weight), while the conductance, $G_{i,j-}$ corresponding to V_{i-} is programmed to high OFF resistive state and vice versa. Let us consider the input conductance of the spintronic neuron during the "write" operation (mainly the HM conductance of the neuron) to be G_s and the voltage drop across the neuron to be V_s. Equating the current supplied by the resistive synapses to the current flowing through the neuron,

we get $\sum_i (G_{i,j+}.(V_{i+} - V_s) + G_{i,j-}.(V_{i-} - V_s)) = G_s.V_s$ which indicates that the net synaptic current supplied to the spintronic neuron is given by,

$$\begin{aligned} I_j &= G_s.V_s \\ &= \frac{G_s.\sum_i (G_{i,j+}.V_{i+} + G_{i,j-}.V_{i-})}{G_s + \sum_i (G_{i,j+} + G_{i,j-})} \\ &= \frac{\sum_i (G_{i,j+}.V_{i+} + G_{i,j-}.V_{i-})}{1+\gamma} \end{aligned} \tag{5.1}$$

As mentioned previously, it is imperative to run spintronic crossbar arrays at low operating voltages from functionality viewpoint. However, lower the operating voltage, higher is the range of synaptic conductances (which can be appropriately tuned by choosing a proper value of MTJ oxide thickness) required to ensure sufficient current requirement for DW displacement from one edge to another in the FM of the spintronic neurons. Hence lower crossbar operating voltage results in the increment of the ratio, $\gamma = \sum_i (G_{i,j+} + G_{i,j-})/G_s$, which in turn, results in non-ideal operation of the neuron. In order to ensure that $\gamma << 1$ for a given crossbar operating voltage, the duration of the "write" cycle can be adjusted accordingly since the current required to achieve a specific DW displacement scales linearly with the duration of the "write" current. The output signals of the inverters from a particular array can be stored in a latch and used to communicate input signals to the fan-out neurons being implemented in the crossbar array for the succeeding stage. Note that the latched neuron outputs can be also used to drive input rows of the same crossbar array (inputs for the next time-step) to implement recurrent neuron connections in RNN architectures.

Ref. [120] evaluated the circuit-level performance of such an All-Spin SNN based design against a baseline CMOS implementation at $45nm$ technology node for a benchmark digit recognition problem. A hybrid device-circuit-algorithm co-simulation framework was utilized for this work. Micro-magnetic simulations to model the domain wall dynamics in presence of charge current input through the HM were performed in MuMax3 [124]. Subsequently, a behavioral model of the device was employed to develop a SPICE model for the neurocomputing fabric. The performance

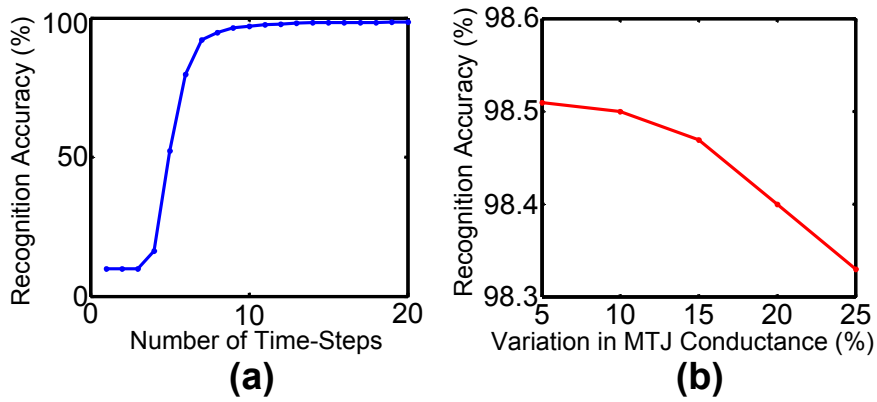

Fig. 5.2. (a) Recognition accuracy over the testing set of the MNIST dataset as a function of the time-steps of simulation, (b) Degradation in recognition accuracy with variation in the MTJ resistances (expressed as % σ variation).

of this design was evaluated for a standard digit recognition problem on the MNIST dataset [156]. The Deep Spiking Neural Network architecture (28x28-12c5-2s-64c5-2s-10o) used for this work, consists of two convolution layers and two subsampling layers arranged alternatively. The training is based on the work performed by authors in Ref. [85]. Our design falls into the category of offline learning where the synaptic weights are learnt off-chip and are programmed to corresponding resistive states of the spintronic synapses once the training is accomplished.

It is imperative to determine the optimum bit discretization necessary in the neurons and synapses of the network in order to minimize the costs for a corresponding hardware implementation. Insignificant degradation in classification accuracy was observed for 4-bit (16 levels) discretization in the synapses and 2-bit (4 levels) discretization in the neurons. Considering that the DW location can be displaced and sensed over a minimum distance of $20nm$, the length of the synapse was taken to be $320nm$, while the length of the neuron was $80nm$. The neuron width was fixed at $20nm$.

As mentioned previously, the optimum "write" cycle duration for the spintronic neurons need to be adjusted in order to minimize the ratio γ. It was observed that

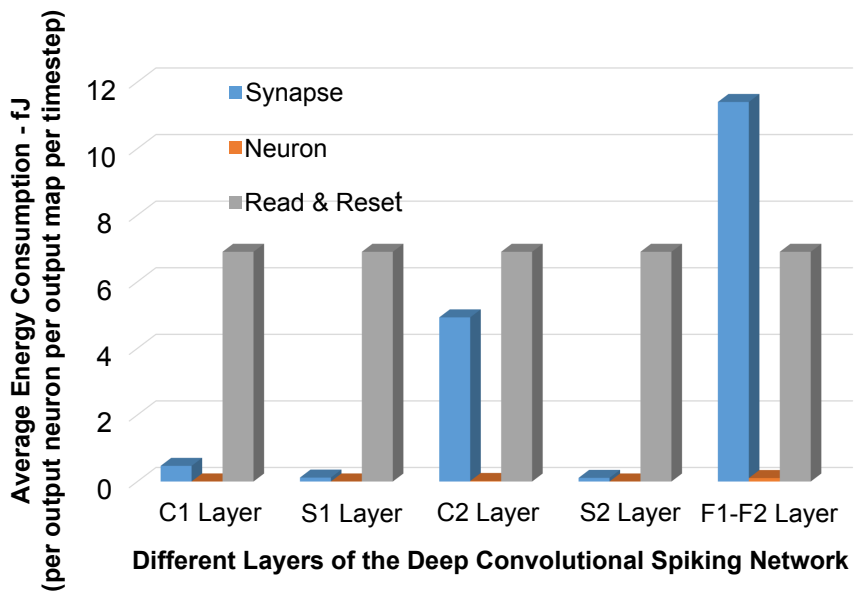

Fig. 5.3. Energy consumption (averaged per output neuron per output map per time-step) for different layers of the spintronic network.

for a "write" cycle duration of $2ns$, there was insignificant impact on the network performance. Fig. 5.2(a) shows the classification accuracy as a function of the time-steps for simulation of the network. An accuracy of 98.5% (measured over the entire testing set) was achieved at the end of 20 time-steps including the effect of such non-idealities.

Variation of DW pinning can be also overcome by suitably having notches along the length of the magnet [157]. However, impact of variation in the MTJ resistances on the performance of the network is an important point of consideration. Fig. 5.2(b) demonstrates that the network performs robustly in terms of classification accuracy, even with 25% σ variation in the MTJ resistances. This is mainly attributed to the error-resilient and self-adaptive nature of such neural algorithms.

An intuitive understanding of the power benefits that could be potentially offered by such spintronic neural network designs can be obtained from device-level simulations. As mentioned previously, micromagnetic simulations reveal that $\sim 10.6\mu A$ current is required to displace the DW from one edge of the neuron to another (di-

mension $80nm \times 20nm$) in a duration of $2ns$. The current flows through the FM-HM bilayer resistance resulting in an energy consumption of $0.05fJ$ (I^2Rt energy consumption). In addition, the spintronic synapses providing input currents to each neuron are operated at ultra-low terminal voltages of $100mV$. Fig. 5.3 depicts the energy consumption (averaged per output neuron per output map per time-step) for different layers of the spintronic network. The "Synapse" and "Neuron" components involve the average energy consumption in the spintronic crossbar array and the interfaced spintronic neurons respectively during the "write" duration of $2ns$. Subsequently, the "read" circuit for the neuron is activated. An oxide thickness of $2nm$ was considered for the neuron MTJ and the "Reference" MTJ to minimize the magnitude of the average "read" current to $31.7nA$. The output of the resistive divider drives an inverter, whose output is stored in a latch, resulting in a pipelined design. In case a spike is generated, the neuron is reset by passing a current through the HM in the opposite direction for a duration of $1ns$. The "Read & Reset" component includes the energy consumption involved in the neuron resistive divider, inverter and the latch design. As expected, the "Synapse" energy consumption increases significantly as the number of fan-in-synapses per neuron start increasing progressively along the layers of the network ($C1 \rightarrow C2 \rightarrow F1 - F2$). The "Neuron" energy component is relatively lower due to ultra-low current switching of magneto-metallic spintronic neurons which in-turn enables the ultra-low voltage operation of the spintronic crossbar array. The "Synapse" and "Neuron" energy components are lower for the sub-sampling layer due to the less-power intensive averaging operation over a 2×2 subsampling window.

A corresponding implementation of the network architecture was synthesized in commercial $45nm$ CMOS technology for comparative purposes. The design consisted of input multiplexers to transmit synaptic weights to the output only if spikes are received. Subsequently the multiplexer outputs were added up to generate the resultant contribution to the neuron membrane potential per time-step. A comparator was utilized to compare the membrane potential value to a specific threshold and determine the corresponding spiking activity. A pipelined design with power-gating (to

exploit the advantage of event-driven operation of the network) was considered with the same bit-discretization mentioned previously. Simulation studies indicate that the proposed spintronic design can potentially achieve 250× improvement in energy consumption and 56× improvement in EDP over the baseline CMOS implementation. Note that this is a circuit level comparison work. Memory access overhead for CMOS based architectures would further increase the energy benefits offered by such All-Spin SNN designs.

5.2 Deterministic STDP Learning

For clarity, the learning circuitry for SNN was omitted in the above discussion. To better understand device, circuit and system level efficiencies with spin-synapses in the context of learning, let us consider the STDP-enabled single layer SNNs discussed in Section 3.4. The network functionality can be mapped to a crossbar array as shown in Fig. 5.4 where spike signals transmitted along the rows from the pre-neurons get summed up along the columns to the post-neurons. The spintronic synapses are programmed only when the post-neuron spikes (with a delay of the negative timing window) and are switched off from the post-neuron circuit during the programming phase using the POST control signal. Each cross-point consists of a spin-synapse interfaced with access transistors and M_{STDP} transistor. An additional programming transistor is also present at each cross-point for the negative timing window but is not shown in Fig. 5.4 for illustrative purposes. Let us consider the circuit primitives and its operation for STDP learning with more details next.

The circuit involved in generating the PRE signal is discussed in this section. Fig. 5.5 shows the sub-threshold CMOS circuit used to generate the PRE signal for pre-neuron A connecting to post-neurons C and D. We discuss the mechanism for generating the signal for the positive time window. A similar design can be used to generate the programming current for the negative time window. The circuit was originally proposed in [158] as a reset and discharge synapse. However it failed to

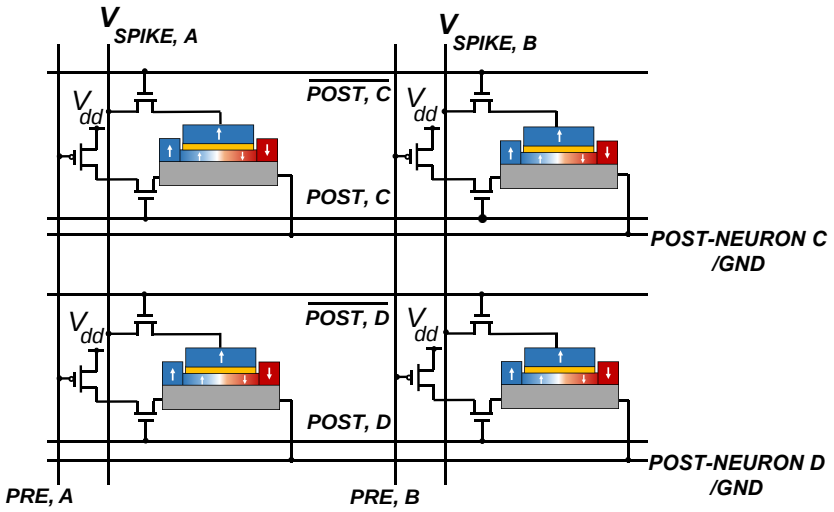

Fig. 5.4. Detailed hybrid spintronic-CMOS crossbar array is depicted for the implementation of STDP learning. Each spintronic synapse is interfaced with programming and access transistors. The 2×2 array connects pre-neurons A and B to post-neurons C and D.

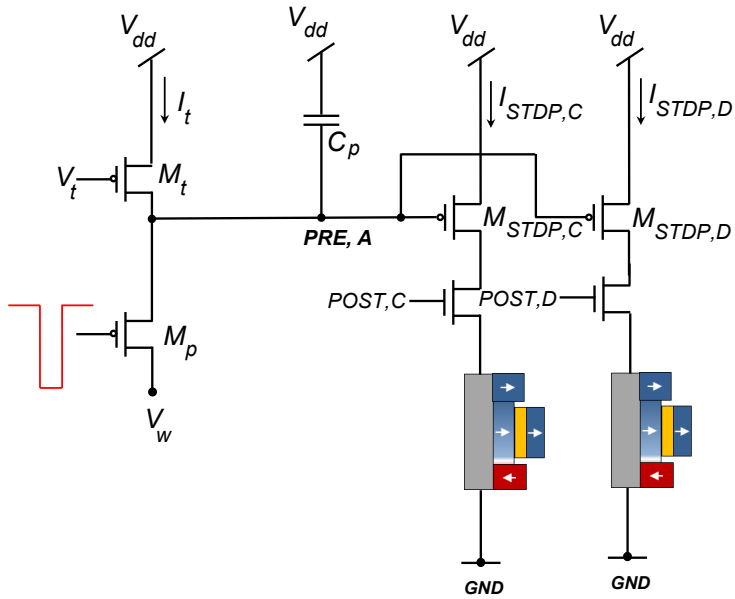

Fig. 5.5. Sub-threshold CMOS circuit utilized for generating the programming current involved in STDP learning (circuit for positive time window shown) for pre-neuron A connecting to post-neurons C and D.

emulate the post-synaptic dynamics of biological synapses as the circuit response depends only on the previous input spike [159]. In this work, we employ this circuit to implement STDP learning in our proposed device.

The transistor M_p acts as a switch. When the positive time window starts, the transistor M_p receives a low-active pulse and gets turned ON. As a result, the node PRE, A is set to the bias voltage V_w. After the transistor M_p is switched OFF, the transistor M_t, operating in sub-threshold saturation regime, provides a constant current to linearly charge the capacitor C_p at a rate $\frac{I_t}{C_p}$. Hence, if the transistor M_{STDP} is operated in sub-threshold saturation, exponential dynamics will be observed in the output current I_{STDP}. The current flowing through transistor M_{STDP} for an input pulse at time $t = t_n$ is given by,

$$I_{STDP} = I_0 e^{\frac{-U_T C_p (t-t_n)}{k I_t}} \tag{5.2}$$

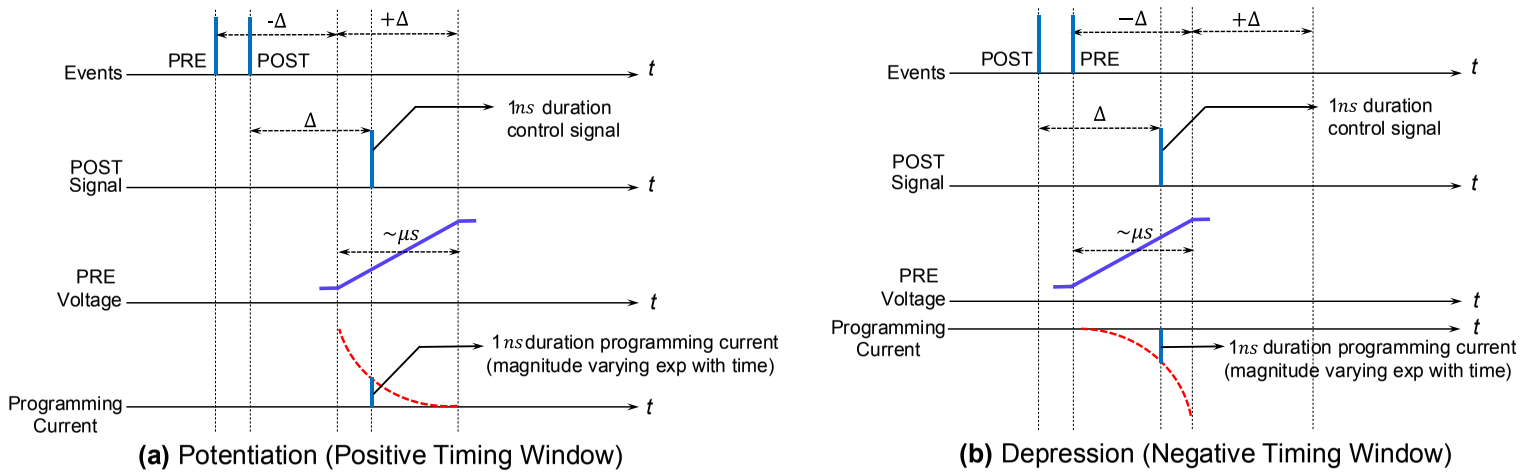

Fig. 5.6. Detailed timing diagrams demonstrating the implementation of (a) potentiation (positive timing window) and (b) depression (negative timing window) in the spintronic synapse. POST is the control signal that is activated during programming while PRE is the gate voltage of the M_{STDP} transistor that implements synaptic plasticity. Duration of the programming current is determined by the duration of the POST signal while the magnitude is determined by the value of the PRE signal when the POST signal is high.

where, k is the sub-threshold slope factor and U_T is the thermal voltage. Hence, whenever the pre-neuron spikes, the circuits for generating the STDP characteristics for the negative and positive time windows are activated sequentially. When learning starts for the positive timing window, a short pulse is applied to the gate of the transistor M_p so that the circuit is reset and the node PRE, A is charged to V_w. When the post-neuron does not spike, the transistor M_{STDP} is in cut-off since the POST signal is deactivated and the access transistors for programming are turned OFF. Once the post-neuron spikes, the programming current path gets activated and the transistor M_{STDP} switches to the sub-threshold saturation regime and transmits the necessary amount of programming current through the device. Unsupervised multi-bit STDP learning with MTJ "spiking" neurons has been demonstrated in Ref. [6].

The operation is discussed in details in Fig. 5.6. Let us first describe the case for the positive timing window, i.e. post-neuron spiking after the pre-neuron (Fig. 5.6(a)). $(-\Delta)/(+\Delta)$ represents the duration during which the learning circuits for the negative/positive timing windows are activated sequentially for the corresponding pre-neuronal firing event. The control signal POST is activated after a duration (Δ) the post-neuron spikes. As described in the figure, magnitude of the programming pulse is determined by the current being passed by the programming transistor M_{STDP} (value of the PRE voltage when the POST signal is active) and the duration is determined by the duration of the POST signal. Since the PRE signal varies in $\sim \mu s$ time scale and does not almost change during the programming time duration ($\sim ns$ time scale), it ensures that the programming current magnitude is almost constant and is equal to the sampled value from the exponential STDP dynamics corresponding to the appropriate spike timing difference. As mentioned previously, since the programming current magnitude is directly proportional to the amount of change in the MTJ conductance, exponential STDP characteristics is implemented in the spintronic device. Similar discussions are valid for the negative timing window (Fig. 5.6(b)) where the post-neuron spikes before the pre-neuron. In this case, the POST

signal is activated during the negative window ($-\Delta$) and the NMOS transistor passes an appropriate amount of programming current in the opposite direction through the device. Circuit-level simulations confirming the proposal have been demonstrated in Fig. 4.17(b).

In order to simulate the SNN implementation based on the proposed spintronic synapse, a hierarchical simulation framework was utilized. Device-level simulations of the spin-orbit torque induced domain wall motion was performed in MuMax. A behavioral model of the device was developed for subsequent simulation of such synapses interfaced with CMOS neurons and learning circuits. The circuit level simulations were performed in HSPICE using a standard cell library in commercial $45nm$ CMOS technology. The device and circuit simulations were utilized to generate models of the plastic synapses and spiking neurons to perform system level simulations of a network of spiking neurons using Brian simulator [160].

The input images (28×28 pixels) used for training was taken from the MNIST dataset [156]. The images were rate encoded and an array of 100 excitatory neurons was used to simulate the self-learning functionality of synapses in SNNs. Synapses present at the crosspoints joining the inputs to the excitatory neurons can be programmed depending on the temporal spiking patterns of the pre- and post-neuron. The inhibitory functionality in such networks can be implemented by an additional row in the crossbar array that is driven by a negative voltage. The row should be activated whenever any of the neurons generate an output spike to prevent multiple neurons from learning the same pattern.

Fig. 5.7 (a)-(b) depicts synapse weights plotted in 28×28 array (same as input images) for each of the 100 neurons used for the recognition purpose. Initially all the weights are random. However, as learning progresses the synapses of each neuron start learning generic representations of the various digits. Thus a particular neuron becomes more sensitive to the digit whose generic representation is being stored in its synapse weights since it will fire more if input spike trains are received at the pixel locations corresponding to high synaptic weights. The various system level simulation

parameters have been outlined in Table 5.1. The parameters were tuned to achieve learning ability in the synapses. The units of the time constants are with respect to the duration of each timestep in the simulation. For this work, the circuits were designed to operate in $\sim \mu s$ time scale as mentioned before. It is worth noting here that the manner in which the time constants and other parameters can be tuned in the circuit level simulations have been discussed in the previous section. The numbers in braces represent the value corresponding to the inhibitory neuron.

Table 5.1.
Spiking Neural Network Parameters for STDP Learning

Parameters	**Value**
No. of excitatory/inhibitory neurons	100
Probability of input spike per timestep	$0 - 0.06375$
Number of timesteps per image	350
STDP time constants	100(1)
Neuron time constants	10(10)
Post-synaptic current time constants	1 (2)

Additionally, we would like to mention here, that such neuromorphic systems are significantly robust to imprecision due to device mismatch, variability and noise effects due to the adaptive nature of such computations involving plasticity, homeostasis and feedback mechanisms [110]. Further, authors in Ref. [161] demonstrate the immunity of such single layer SNNs based on crossbar arrays of resistive synapses with lateral inhibition and homeostasis effects to variations and non-idealities in typical resistive synaptic devices and CMOS neuron circuits. In particular, we performed an analysis of the impact of variations in the oxide thickness/MTJ synaptic conductances on the classification accuracy of the system. Almost no degradation in classification accuracy

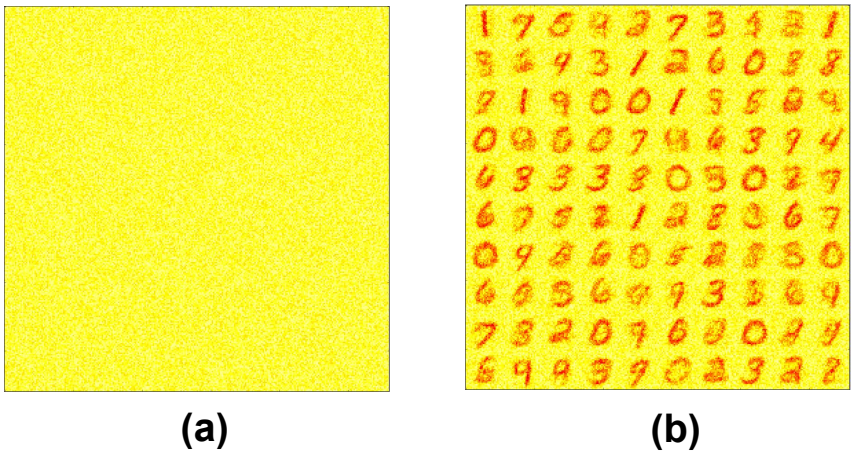

Fig. 5.7. (a) SNN topology used for digit recognition arranged in a crossbar array fashion, (b) Initial random synapse weights plotted in a 28×28 array for 100 neurons in the excitatory layer, (c) Representative digit patterns start getting stored in the synapse weights for each neuron after 1000 learning epochs.

was observed for the 100-neuron network even with 25% variation in the resistances of the spintronic synapses.

Interested readers are referred to Ref. [162] for a discussion on the practical implementation of arrays of such spintronic devices interfaced with CMOS transistors. The size limitation of crossbar arrays of such spintronic devices is determined by the driving capabilities of rows of the array by input voltages in the presence of parasitics. In addition, sneak paths also become a potential issue for large crossbar arrays in order to implement on-chip learning. These are concerns that are equally valid for spin-devices and other memristive technologies, in general. However, it is worth noting here that computation occurring in a large crossbar can be distributed easily among smaller crossbar arrays by simply replacing the large unit by an equivalent number of smaller crossbar units using peripheral control circuitry [163].

5.3 All-Spin Neural Networks for Stochastic Inference

While the above discussion in Section 5.1 considered offline trained Deep ANNs driven by deterministic DW motion based IF "spiking" neurons, similar SNN networks can be trained for stochastic "spiking" neurons enabled by single-domain MTJs. Ref. [127] explored an approach of training deep ANNs with sigmoid transfer function neurons using backpropagation and subsequently utilizing the offline trained weights to implement an SNN where the neurons generate output spikes at each time-step using sigmoid probability distribution functions. The advantages of such an approach is driven solely by the fact that complex neural operations (like sigmoid transfer functions) required to achieve high recognition accuracies can be now implemented by simple device structures consisting of mono-domain magnets by leveraging the underlying device stochasticity. The details of the algorithm and device-circuit primitives for designing such networks enabled by stochastic neurons are provided next.

Let us consider an ANN neural unit that receives an input I through a synapse of weight w. The neuron generates an output y by passing the weighted input through a non-linearity $f(.)$. We will consider the function $f(.)$ to be the sigmoid function $\left(f(x) = \frac{1}{1+e^{-x}}\right)$ in this work, due to its popularity in traditional ANN networks for achieving high accuracy in complex recognition problems [164] along with the possibility of enabling this functionality by MTJ devices, as will be explained next. Hence, for the ANN neuron, the corresponding output y will be given by,

$$y = \frac{1}{1 + e^{-w.I}} \tag{5.3}$$

It is worth noting here that the input $I \in [0, 1]$, since it represents the inputs coming from normalized values of external stimuli (image pixels for image recognition systems) or from other neuron outputs in previous layers (which lie in the range $[0, 1]$ due to the limited range of sigmoid function).

Next, let us describe the proposed conversion process from ANN to SNN (Fig. 5.8(a)). In the spiking mode of communication, the input I can be rate encoded as a Poisson spike train $\widetilde{I}(t)$. The train consists of a sufficiently large number of time-

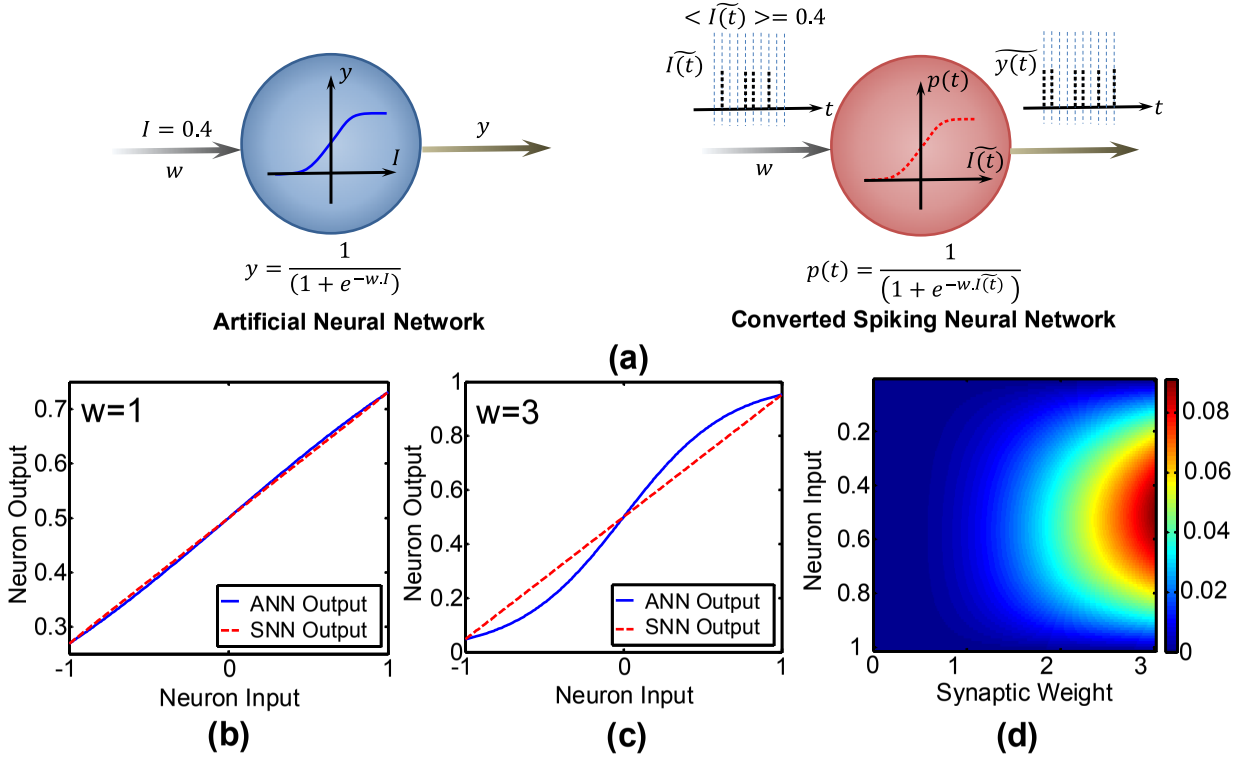

Fig. 5.8. (a) The ANN is converted to SNN computing model by interpreting the neuron transfer function as the neuron spiking probability in the SNN mode, (b) and (c) ANN and SNN outputs are plotted over the entire input range for weight magnitudes, $w = 1$ and $w = 3$ (maximum weight) respectively, (c) Error contour plot between the ANN output and the converted SNN output with variation in both neuron input and synaptic weight magnitudes. The error increases with increasing weight but remains bounded within reasonably low values.

steps, T_N, where the probability of generating a spike at each time-step equals the input I. It can be proved that the resulting process is a homogeneous (probability of spike generation constant over time-steps) Poisson process where the average firing rate, i.e. average number of spikes generated over the entire train duration, is given by [165],

$$< \widetilde{I}(t) >= \frac{\sum_t \widetilde{I}(t)}{T_N} = I \tag{5.4}$$

The spiking neuron processes the input spikes and generates a set of output spikes $\widetilde{y}(t)$. The response of the neuron is determined by its average firing activity over the T_N time-steps, $< \widetilde{y}(t) >$. Note that such input encoding and neuron output measurement schemes are standard norms for SNNs and is not an additional requirement/overhead for our proposal. Our proposal concerns the manner in which the neuron will process and generate the output spike train $\widetilde{y}(t)$. In order to achieve near lossless (with respect to accuracy) conversion from ANN to SNN, $< \widetilde{y}(t) >$ should approximate y reasonably well.

Our conversion mechanism follows from the very intuitive observation that the analog activation output of the ANN neuron in the range $[0, 1]$ can be mapped to the probability of spike generation, $p(t)$, of the spiking neuron at each time-step. Hence, at each time-step t, the neuron receives the input spike train, $\widetilde{I}(t)$, and generates an output spike with probability $p(t) = f(\widetilde{I}(t))$.

Now, let us provide a mathematical analysis to justify that such a mapping is able to approximate the original ANN neural unit to a reasonable degree of precision. It follows from Eq. 5.4, that the spike train consists of $I.T_N$ number of spiking events and $(1 - I).T_N$ number of non-spiking events, on the average, over the entire duration of time-steps, T_N. The output spike train is generated according to an inhomogeneous Poisson process [165] (spike generation probability varies over time), where the probability of spike generation is equal to $p(t|\widetilde{I}(t) = 1) = \frac{1}{1+e^{-w}}$ whenever there is an input spike and $p(t|\widetilde{I}(t) = 0) = \frac{1}{1+e^{0}} = \frac{1}{2}$ in the case of no spike. Hence, the inhomogeneous Poisson process can be decomposed into two homogeneous Poisson processes corresponding to spiking (of duration $I.T_N$ time-steps) and non-spiking

events (of duration $(1-I).T_N$ time-steps). Hence, the average firing activity of the neuron will be given by the sum of the firing activities of the individual Poisson processes averaged over the total number of time-steps, T_N. Following Eq. 5.4, we can state that the average firing rate of the output spike train, $\widetilde{y}(t)$, is given by,

$$\begin{aligned} <\widetilde{y}(t)> &= p(t|\widetilde{I}(t)=1).I + p(t|\widetilde{I}(t)=0).(1-I) \\ &= \frac{I}{1+e^{-w}} + \frac{1-I}{1+e^{0}} \\ &= \frac{1}{2} + \frac{I}{2}\left(\frac{1-e^{-w}}{1+e^{-w}}\right)\Bigg(\end{aligned} \tag{5.5}$$

Closer inspection of the above equation reveals that $<\widetilde{y}(t)>$ is a linear approximation of the sigmoid function in the range $I \in [0,1]$. Fig. 5.8(b) and (c) represents a plot of the outputs, y (ANN) and $<\widetilde{y}(t)>$ (SNN) with variation in the input I and for synaptic weight magnitudes $w=1$ and $w=3$ respectively (3 being the maximum weight for the synapses in our network). Note that the negative range for I represents the case for negative synaptic weight. As can be concluded from the figure, the error between the functions is almost negligible for $w=1$ and increases slightly as the magnitude of the weight increases. However, even for the maximum weight $w=3$, the error remains bounded below reasonably low values over the entire approximation range. This fact is reinstated by Fig. 5.8(d) which represents a contour plot of the error magnitude between the two expressions y and $<\widetilde{y}(t)>$ with variation in both I and w. Note that since we are trying to encode information in the analog sigmoid output of the neural units, weights obtained as a result of backpropagation training typically remain bounded below values that ensure that the neuron outputs do not fall in the saturation regime of the sigmoid function. As can be observed from Fig. 5.8(c), for a weight magnitude of 3, almost the entire range of the sigmoid function is being used and hence it is expected that synaptic weights should converge to such limited ranges after the training process. Additionally neural nets, being inspired from computational mechanisms observed in the biological brain, are characterized by an inherent tolerance to variations in the neural and synaptic units and hence such

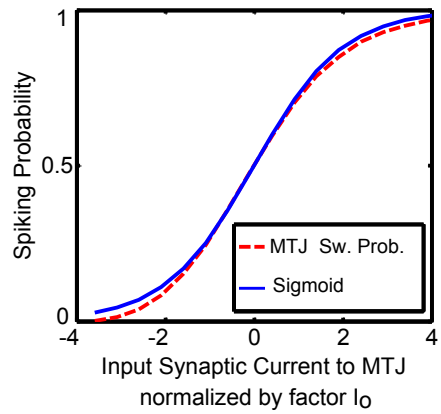

Fig. 5.9. (a) Switching probability characteristics of an MTJ of volume $\frac{\pi}{4} \times 100 \times 40 \times 1.2nm^3$ at $T = 300K$ re-plotted for $T_w = 0.5ns$ as a function of the input synaptic current, I_{syn}, normalized by factor $I_o = 10\mu A$. The data closely resembles a sigmoid probability density function.

minor variation between y (ANN) and $< \widetilde{y}(t) >$ (SNN) is not expected to impact the network performance.

The device simulation parameters have been outlined in Table. 4.1 and are based on experimental measurements performed in Ref. [118]. A barrier height of $20k_BT$ was chosen since the MTJ is being used as a computing element in this application. Fig. 4.13 depicts the switching probability of the MTJ with variation in the magnitude of input current. The probability switching characteristics undergoes more dispersion with decrease in the duration of the input "write" current, T_w. While more dispersion in the characteristics results in increased robustness of the system in presence of variations, power consumption of the network increases. These tradeoffs will be discussed in details later. In order to map such switching probability characteristics of the MTJ to the sigmoid probability function for spike generation discussed in the previous section, the MTJ is considered to be driven by two input currents, namely I_{bias} and I_{syn}. The current I_{bias} provides the necessary current to the MTJ to bias it at a probability of 0.5. The current I_{syn} is the resultant input synaptic current to the neuron. Hence, in absence of I_{syn}, the MTJ has 50% probability of switching

similar to the sigmoid characteristics. Fig. 5.9 illustrates the switching probability characteristics of the MTJ with variation in input synaptic current, I_{syn} (normalized by a factor, I_o, which encodes the degree of dispersion of the MTJ switching probability characteristics). The switching characteristics match the sigmoid variation to a reasonable degree of approximation. Also, note that such neuromorphic algorithms are highly error-resilient and such small approximations in the neuron output will not cause significant changes in the network performance. We will validate our claims by presenting results for a convolutional neural network in the next section. The mapping of the normalization factor in the input synaptic current, I_o, to the hardware implementation of a synaptic crossbar array will be discussed later.

In order to implement a neural network, neurons need to be interfaced with synapses. The basic computing core in any neural network architecture, even for deep networks, consists of a dot product implementation where each of the neural inputs are initially multiplied by synaptic weights, and are subsequently processed by the neuron. Such a functionality can be directly mapped to a crossbar architecture, as discussed in an earlier section. The operation of the crossbar array is exactly similar as described in the previous section (along with the associated terminologies) except for the fact that the MTJ receives bias current I_{bias} along with the current from the crossbar array. Equating the current supplied by the resistive synapses along with the input bias current, I_{bias}, to the current flowing through the neuron, we get $\sum_i (G_{i,j+}.(V_{i+} - V_s) + G_{i,j-}.(V_{i-} - V_s)) + I_{bias} = G_s.V_s$ which indicates that the net synaptic current supplied to the spintronic neuron is given by,

$$\begin{aligned} I_j &= \frac{G_s.\left(\sum_i (G_{i,j+}.V_{i+} + G_{i,j-}.V_{i-}) + I_{bias}\right)}{G_s + \sum_i (G_{i,j+} + G_{i,j-})} \\ &= \frac{\sum_i (G_{i,j+}.V_{i+} + G_{i,j-}.V_{i-}) + I_{bias}}{1+\gamma} \end{aligned} \tag{5.6}$$

Note that the resultant weighted synaptic input is scaled by a factor $G_o.V_o$ (in the current domain). Hence, in order to map the functionality to the sigmoid probability characteristics, the scaling factor in the MTJ switching characteristics discussed pre-

viously, I_o has to be equal to $G_o.V_o$. In other words, the resultant synaptic current being supplied by the crossbar array needs to be adjusted according to the dispersion of the switching probability characteristics of the MTJ in order to maintain consistency with the computational model described previously.

As mentioned in the previous subsection, the input resistance of the neuronal device has to be sufficiently low in order to ensure that most of the input voltage drops across the resistive synapses and the voltage drop across the neurons are negligible, i.e. to minimize the effect of γ. Hence, a sufficient value of the spike voltage, V_o (which dictates the value of G_o), has to be maintained to ensure that $\gamma << 1$. Duration of the input "write" current also has an impact on the choice of V_o and G_o. With more duration of input current and hence, less dispersion in the switching characteristics, I_o decreases resulting in decrease of G_o and hence γ. However, robustness of the system to variations in the bias current and synaptic conductances suffer. These design space explorations will be considered in details next. Operation of each time-step of the SNN takes place through three cycles. In the first phase or the "write" cycle, the MTJ neuron receives the bias current and the input synaptic current from the crossbar array and switches probabilistically. Note that the bias current can be provided by an additional row of the crossbar array consisting of PMOS transistors biased in saturation. After the "write" cycle, the "read" terminals of the neuron are activated. As mentioned before, the "read" circuit consists of a resistive divider network with a "Reference" MTJ (whose state is fixed to the AP state). Hence a spike (logic value '1') is generated at the output inverter in case the MTJ switches to the P state. In case a spike is generated, the MTJ is switched back to the AP state by passing a sufficiently high magnitude of current through the HM in the opposite direction during a subsequent "reset" phase to ensure normal MTJ operation during the next time-step.

The performance of the network was assessed for a deep learning network architecture [164] (28x28-6c5-2s-12c5-2s-10o) on a standard digit recognition problem based on the MNIST dataset [156].The network is trained using 60,000 training samples

based on the methodology outlined in Ref. [164]. Once the training is accomplished, the learnt weights are mapped to the synaptic conductances using the scheme mentioned in the previous section. All recognition accuracies mentioned in this text are with respect to the 10,000 test samples in the dataset. The baseline ANN network was trained with an accuracy of 98.56% over the testing set. During the operation of the converted SNN, the image pixels are converted to Poisson spike trains where the average number of spikes generated over a given time window encode the corresponding pixel intensity.

Note that a convolutional architecture is being used in this work since it has achieved high recognition accuracies in a large number of complex datasets. Further the architecture only dictates the manner in which the neurons and synapses are connected to form the network. However, our proposal holds true for any neural network topology since the basic computational elements and their mapping to crossbar architectures remain equally valid. We would also like to point out that improved training algorithms/network architectures to enhance the performance of the network in terms of recognition accuracy can be performed. However, the goal of this work is to demonstrate the applicability of the MTJ as a probabilistic spiking neuron that can potentially enable near-lossless (with respect to classification accuracy), low-power, low latency SNNs converted from trained ANNs.

Let us first describe the impact of "write" cycle duration on the performance of the network. With increase in the duration of the "write" cycle, the switching probability characteristics become sharper. Hence the synaptic current requirement from the crossbar array reduces. Further, the bias current magnitude also reduces since spin-orbit torque is exerted on the magnet for a longer duration of time. Hence, power consumption of the network is expected to reduce with increase in the magnitude of the "write" cycle duration. However, this occurs at the expense of delay since the network has to be operated over a number of time-steps and each time-step duration is directly related to the duration of the "write" cycle.

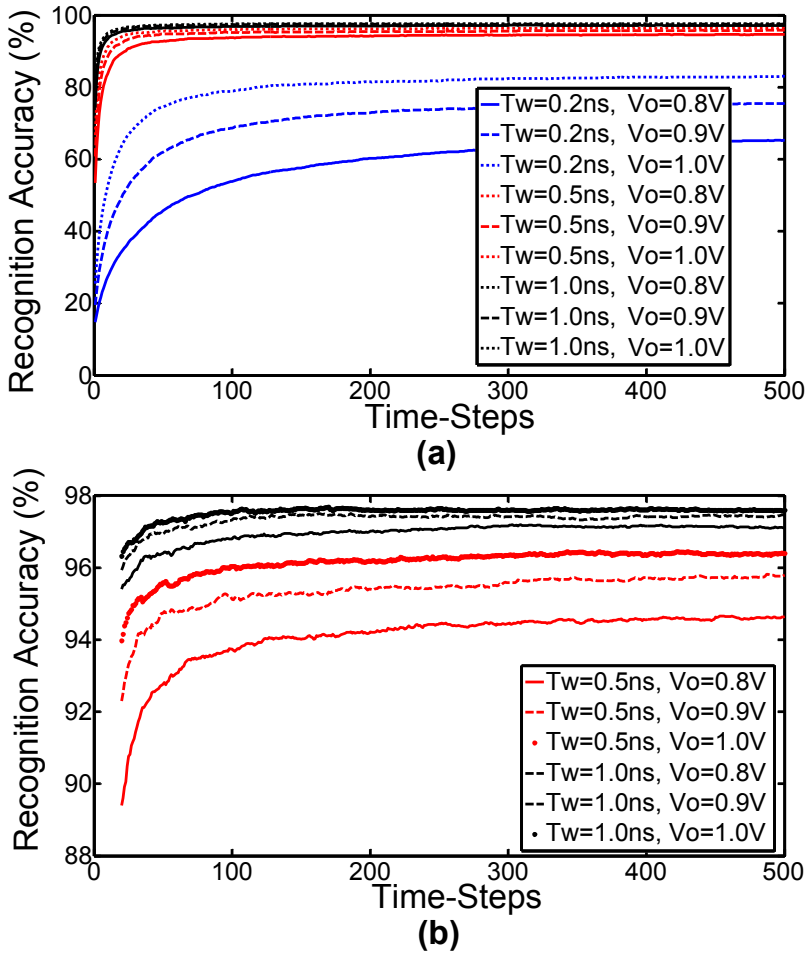

Fig. 5.10. (a) Recognition accuracy as a function of time-steps with variation in the "write" cycle duration ($T_w = 0.2, 0.5$ and $1ns$) and crossbar supply voltage ($V_o = 0.8, 0.9$ and $1V$), (b) Zoomed-in depiction of plot (a) from 50-500 time-steps for $T_w = 0.5$ and $1ns$. Near-lossless SNN conversion can be achieved by maintaining a sufficient duration of the "write" cycle, even with scaling of crossbar supply voltage.

However, decrease in the "write" cycle duration, i.e. increase in the dispersion of the probability switching characteristics of the MTJ will result in increase of the factor γ, as discussed previously, thereby leading to non-ideal network operation. Fig. 5.10 depicts the classification accuracy as a function of the time-steps of simulation of the SNN with varying "write" cycle durations (T_w), namely 0.2, 0.5 and $1ns$. As expected, for a fixed supply voltage, classification accuracy improves with increase in the "write" cycle duration. While the network accuracy reaches 97.6% and 96.4% for $T_w = 1ns$ and $0.5ns$ respectively, it saturates at 83% for $T_w = 0.2ns$ at the end of 500 time-steps. An interesting point to note is the low latency in the performance of the network. The accuracy reaches 96.3% and 93.8% at the end of just 20 time-steps for $T_w = 1ns$ and $0.5ns$ respectively. This is a crucial advantage offered by our ANN-SNN conversion scheme since although SNN implementations are ideal for low-power neural network implementations, they incur penalty in terms of the delay since the network outputs need to be observed over a number of time-steps to generate sufficient confidence in the inference process. With our proposed conversion scheme, network accuracies close to the original trained ANN baseline can be achieved only within a few tens of time-steps of the spiking network operation.

Scaling the supply voltage, in turn, results in increment of the factor γ, thereby leading to more errors in the network performance. However, it is worth noting here that the drop in recognition accuracy is minimal for sufficiently large durations of the "write" cycle. For instance, the accuracy drop is insignificant (97.1% and 94.6% for $T_w = 1ns$ and $0.5ns$ respectively) even with the crossbar supply voltage being scaled down to $0.8V$. The key point we would like to stress from this section is that by maintaining a sufficient duration of the "write" cycle, it is possible to achieve near-lossless SNN operation with minimal delay coupled with the possibilities of voltage scaling for reduction in power consumption. It is also worth noting here that the analysis performed in this section includes non-idealities arising from hardware mapping of the SNN to a synaptic resistive crossbar array interfaced with MTJ neurons (includ-

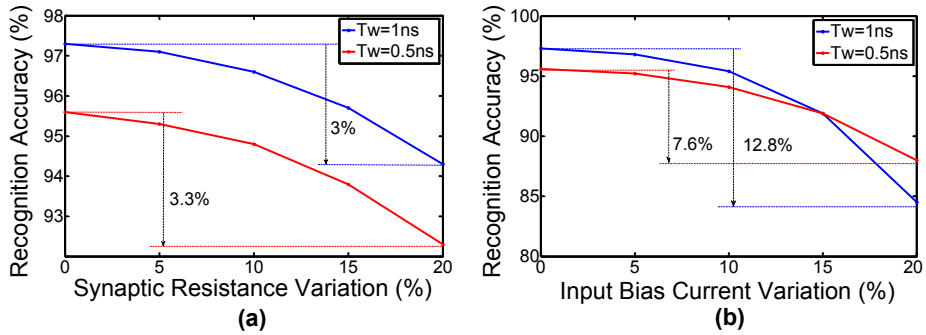

Fig. 5.11. Average recognition accuracy (measured over 50 independent Monte Carlo simulations for each of the 10,000 test images in the dataset) with variations (expressed as % σ variation) in (a) resistances in the synaptic crossbar array and, (b) input bias current to the MTJ. The results have been measured at the end of 50 time-steps of SNN operation for crossbar supply voltage, $V_o = 1V$.

ing non-ideality factor γ and deviations of MTJ switching probability characteristics from ideal sigmoid function).

Although increase in the "write" cycle duration helps to reduce the non-ideality in the network (by reduction of factor γ), it is associated with increased performance loss in presence of random variations due to sharper probability switching characteristics of the MTJ. In this section we will investigate the impact of random variations in the synaptic resistances of the crossbar array along with variations in the input bias current of the MTJ (Fig. 5.11). The average classification accuracy was determined by performing 50 independent Monte Carlo simulations of the network for each of the 10,000 test images in the dataset.

Fig. 5.11(a) depicts the average classification accuracy of the network with variations in the synaptic resistances of the crossbar array. Since the range of synaptic resistances are adjusted according to the dispersion of the MTJ switching probability characteristics (through the relation $I_o = V_o.G_o$ discussed previously), the impact of synaptic resistance variation is expected to be similar for different "write" cycle durations. An additional point to note is that, even with $\sigma = 20\%$ variation in the synaptic resistances, only 3% ($T_w = 1ns$) and 3.3% ($T_w = 0.5ns$) degradation in classification

accuracy was observed with respect to the original network (without variations) at the end of 50 time-steps. Such robustness to variations in the input synaptic current can be attributed to the error-resiliency of such neuromorphic computing systems.

However, the input bias current of the MTJ is a more critical parameter (with respect to variations) that ensures proper functionality of the network. Variations in the input bias current can skew the probabilistic MTJ operation in one direction, thereby causing degradation in recognition accuracy. Hence, sharper MTJ probability switching characteristics would result in more errors during the recognition process with variations in the input bias current. Fig. 5.11(b) illustrates that while 12.8% reduction in accuracy was observed for $\sigma = 20\%$ over the ideal network at the end of 50 time-steps for $T_w = 1ns$, only 7.6% degradation was observed for $T_w = 0.5ns$. These results signify the fact that it is crucial to choose an optimal value of "write" cycle duration that simultaneously achieves near-lossless SNN conversion along with robustness to random variations in the input bias and synaptic currents. Note that a precise value of input bias current can be maintained by utilizing CMOS reference current generators that would exhibit σ variations much less than 20%. However, impact on network performance with such high degree of variations was performed to establish that the network is highly error-resilient along with the fact that a judicious choice of the "write" cycle duration can enable robustness of the network even to large variations in the more sensitive MTJ input bias current.

Additionally, we considered the impact of variation in the chip operating temperature by running a worst-case simulation where all the MTJs in the network were assumed to operate at $400K$ instead of the design temperature, $300K$. A recognition accuracy of 96.73% was achieved at the end of 50 time-steps of network operation, thereby confirming that the proposed probabilistic neural computing framework is resilient to temperature variations as well.

In order to evaluate the energy consumption of the network, SPICE simulations were performed to determine the energy consumption involved in "write", "read" and "reset" operations. In addition to providing a compact implementation of a

spiking neuron, the MTJ enables low-power operation of the synaptic crossbar array. This is due to the fact that only input current magnitudes of a few tens of μA need to be supplied by the crossbar array on either side of the bias current. Note that the dominant power consumption of the network is involved in the synaptic crossbar array (since the number of synapses typically outnumber the number of neurons in such deep neural networks by two to three orders of magnitude), and such magneto-metallic spintronic neurons enable the low-power operation of the crossbar architectures. For the energy analysis, we considered the optimal "write" and "reset" cycle duration to be $0.5ns$ due to the possibilities of achieving near-lossless SNN conversion along with robustness to input bias current variations. As mentioned previously, an intuitive insight to the power efficiency of the network can be obtained by considering the fact that only $71\mu A$ of input current is required to bias the MTJ at 50% switching probability ($T_w = 0.5ns$). This current flowing through a HM resistance of 400Ω, results in an I^2Rt energy consumption of $\sim 1fJ$ in the neuron. Considering the resultant energy consumption in the "write", "read" and "reset" cycles of the network over a duration of 50 time-steps (since competitive classification accuracy can be obtained at the end of a few tens of time-steps), the total energy consumption of the proposed MTJ based SNN network was evaluated to be $19.5nJ$ per image classification.

An interesting point to note is that there is an additional delay overhead involved in the SNN operation. On the other hand, ANN operation (for instance, resistive crossbar array driven by analog CMOS neurons) would require a single time-step for recognition. However, the delay overhead (few tens of time-steps) is much smaller than the corresponding reduction in power consumption due to event (spike)-driven hardware operation. For example, the average energy consumption of an analog CMOS neuron is estimated to be $\sim 700fJ$ [166] which would still be an order of magnitude greater than the average energy consumption of an MTJ neuron ($\sim 1fJ$) operated over a duration of 50 time-steps.

In order to compare with a baseline digital CMOS implementation, a deep spiking network consisting of Integrate-Fire (IF) neurons converted from a corresponding trained ANN was used based on the methodology proposed in Ref. [85] for the same network architecture (28x28-6c5-2s-12c5-2s-10o) being considered in this work. The network was synthesized using a standard cell library in $45nm$ commercial CMOS technology. The design consisted of digital adders to sum up the synaptic weights in case of a spiking event (enabled by multiplexers). A comparator was utilized to compare the accumulated synaptic contributions to a specific threshold (IF functionality) and determine the corresponding spiking activity. A pipelined design with power-gating (to exploit the advantage of event-driven operation of the network) was considered with the same bit-discretization in the synaptic weights as mentioned previously. The average energy consumption involved in the network per image classification was evaluated to be $391nJ$ ($20\times$ more energy consumption than the proposed MTJ based spiking architecture).

Analysis on the scaling effects of stochastic spin devices for neuromorphic computing have been performed in Ref. [143]. As mentioned previously, scaling magnetic device dimensions results in reduced energy consumption for stochastic operation. However, as the scaling tends to the "super-paramagnetic" regime, the magnet undergoes volatile telegraphic switching. Such a volatile device operation entails "asynchronous" mode of network operation since parallel "read" and "write" operations are now required for the MTJ (unlike the synchronous clocked "write" and "read" cycles used to operate the MTJ for non-superparamagnetic MTJs). The "read" and "write" ports of the neuron MTJ are activated simultaneously due to the low data retention time of the magnet. The system is not driven in a synchronous fashion by any clock signal and spikes generated by the neuron output inverters drive the next set of fan-out neurons in an asynchronous fashion. Note that asynchronous parallel "read" and "write" operations are also not suited for high barrier height magnets in the non-telegraphic regime ($10 - 20k_BT$) from delay perspective since telegraphic switching would occur in the $\sim \mu s - ms$ timescale in this scenario. As the barrier

height is scaled, the retention failure probability of the magnet during a specified "read" cycle will increase. Analysis performed in Ref. [143], reveal that the barrier height of the magnet should be greater than $4.6k_BT$ to ensure that the retention failure probability is less than 1% during a "read" time cycle of $1ns$ (required time for worst-case corner simulations of the "read" circuit in $45nm$ technology node). Hence magnets with barrier heights less than $5k_BT$ is more suited for the asynchronous scheme of operation mentioned above.

The lower power consumption in superparamagnets as neural inference elements is achieved at the expense of reduced error resiliency. Since the "write" and "read" operations occur in parallel for magnets switching in the telegraphic regime, the "read" current can significantly bias the probabilistic switching of the device. Magnetic fields generated by nearby electric currents may also serve to bias the device stochasticity. The situation is worsened by the fact that the "write" and "read" currents are in the same range due to significantly lower "write" current requirement for stochastic switching in such scaled devices. Hence the "read" circuit for the neuron MTJ needs to be highly optimized such that the read current is maintained at the minimal value. Note that this is not a design issue in higher barrier height magnets since "read" and "write" cycles are de-coupled in time. Further, the gradient or the rate of change of switching characteristics of such magnets in response to input current magnitude is extremely high. For instance, the stochastic switching characteristics undergo a full swing from 0 to 1 approximately in the range of $\pm 1\mu A$ for a $1k_BT$ magnet [143]. In other words, the stochastic switching characteristics are highly sensitive to variations in the magnitude of the external bias input current which, in turn, results in reduced classification accuracy or similar performance metric of any pattern recognition system with variations in the supply voltage, synaptic conductances or CMOS peripherals [143]. For instance, variation analysis performed in Ref. [143] for a standard digit recognition problem on a two-layer convolutional neural network architecture enabled by asynchronous operation of $1k_BT$ barrier height magnets reveal $\sim 5\%$ accuracy degradation for 20% variation in the synaptic resistive elements,

$\sim 6\%$ accuracy degradation for $25mV$ variation in crossbar supply voltage and 3% accuracy decrement for worst-case corner simulation with 2σ variations in the CMOS read circuit. In contrast, the synchronous implementation with higher barrier height magnets are resilient to variations in the crossbar supply voltage and read circuit while a small degradation of $\sim 3\%$ classification accuracy is observed for variations in the synaptic elements of the resistive crossbar array. Note that such sensitive operation in response to noise and other non-idealities is not specific to a $1k_BT$ magnet but is valid for superparamagnets operating in the telegraphic switching regime (barrier height in the range $1 - 5k_BT$).

5.4 Probabilistic STDP Learning

The multi-bit STDP formulation can be modified in the stochastic single-bit scenario to represent the probability of synaptic state change in response to spike timing difference [7]. The synaptic state change probability can be modulated by appropriate peripheral circuitry (similar to the one described for the domain wall motion based devices) that ensures proper variation of the programming current magnitude with spike timing difference. The operation of the crossbar array of stochastic synapses driving stochastic neurons is similar to the array described for domain wall motion based devices (depicted in Fig. 5.1) except that the core neuron and synaptic devices have single bit resolution in contrast to the domain wall motion based devices. The biasing region of the M_{STDP} transistor is determined to ensure that the current flowing through the heavy metal varies in such a manner that the switching probability of the MTJ varies exponentially with the spike timing difference. Probabilistic STDP based on spintronic synapses in such single layer networks have been demonstrated in Ref. [7] and have been able to achieve $\sim 80\%$ recognition accuracy over the MNIST [156] training set for a set of 225 excitatory neurons. Such networks have been shown to achieve competitive recognition accuracies by increasing the neuron count beyond 1000. Interested readers are referred to Ref. [167] for an overview of All-Spin

Stochastic SNNs where stochastic synaptic learning is accomplished by probabilistic neural inference, both enabled by single-domain MTJ devices. It is worth mentioning here that such stochastic computing paradigms are equally valid for magnets scaled to the super-paramagnetic regime. However, appropriate circuit considerations need to be accounted for due to the telegraphic switching behavior of such low barrier magnets [143]. Note that such networks, in principle, are "Binary Networks" being characterized by binary neuron and binary synaptic units.

5.5 System Level Benchmarking

We also performed a rigorous system-level benchmarking of a reconfigurable neuromorphic architecture based on such All-Spin SNNs [168]. In this section, we discuss our spintronic "in-memory" computing architecture (referred to as "Spintronic Architecture" in Fig. 5.15) that is used to analyze the system-level benefits of spintronic devices for SNN acceleration. As discussed earlier, a Spintronic Crossbar Array (SCA) stores the trained weight (connectivity) matrix and computes the inner-product between the input and the weight matrix. This obviates the frequent data transfer requirements between memory and computation core. Furthermore, the SCA is interfaced with spintronic neurons that allow low-power inner-product and neuron computations.

The size of an SCA is typically limited by the driving capability of the voltage drivers and the fan-in limitation of a spin-neuron. However, the neuron fan-in in a typical neural network is of the order of several hundreds. Hence mapping such a connectivity (weight) matrix requires partitioning the matrix across multiple SCAs to provide input to the same output neuron. The output neuron computation is done by time-multiplexing the SCA current integrations on the neuron as shown in Fig. 5.12. Fig. 5.12(a) shows a feed-forward neural network with neuron fan-in of 4. Fig. 5.12(b) shows the mapping of the network on SCAs of size 2×2. Each column of the SCA corresponds to an output neuron. Two weights for each output neuron

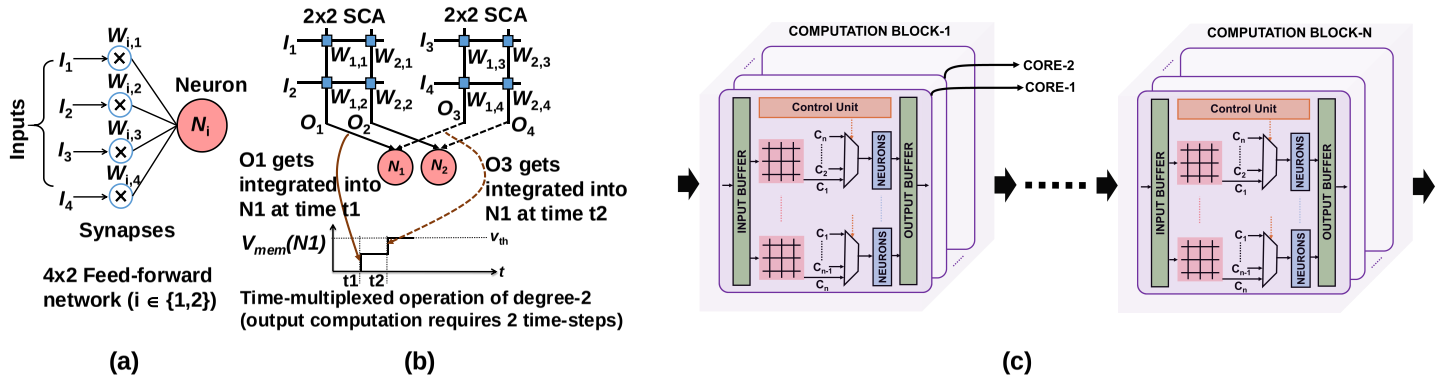

Fig. 5.12. (a) 4×2 feedforward neural network, (b) Time-multiplexed execution of the 4×2 network on 2×2 SCAs, (c) Organization of Spintronic "In-Memory" Computing Architecture for SNNs.

are mapped on each SCA. As shown in Fig. 5.12(b), the final neuron output (for instance 'N1') is computed by time-multiplexed integration of crossbar currents ('O1' and 'O3') on N1.

Fig. 5.12(c) shows the logical organization of our spintronic architecture. The SNN realization is achieved by multiple computation blocks connected back to back. Further, each computation block is composed of multiple computation cores (CORE in Fig. 5.12(c)). As shown in Fig. 5.12(c), the computation core is a pool of SCAs, associated neurons, input and output buffers coupled together with a control unit. Such a computation core efficiently realizes the partitioned connectivity matrices by mapping them across multiple SCAs locally within a computation core. The control unit realizes the time-multiplexed integrations depending on a neuron's fan-in. Thus the core is the computation primitive in our spintronic architecture. Eventually multiple such computation cores are employed to map a layer of SNN depending on the number of neurons and synapses contained in the layer. Different layers of the SNN are mapped across multiple computation blocks to map all the neurons and synapses in an SNN. One sequential dataflow throughout these computational blocks (that spatially map the layers) realizes one-time step of the SNN implementation.

Here, we describe our CMOS baseline architecture for SNNs. SNeuE is a many-core architecture which utilizes the data sharing patterns in SNN processing to enable their energy-efficient acceleration. SNeuE consists of two parts, namely: (1) SRAM to store the trained weights and inputs, and (2) computation core to perform the inner-product between the inputs and weights fetched from SRAM along with neuron computations.

Here, we explain the logical dataflow between different components in SNeuE (shown in Fig. 5.13). Weights stored in the SRAM are fetched and stored into the weight FIFOs present in the computation core. Each Neuron Unit (NU) receives its weights from a dedicated weight FIFO. The input FIFO streams input data across the NU array that allows data sharing and reduces the memory (SRAM) fetches associated with inputs, thereby resulting in energy efficiency. This is a direct consequence

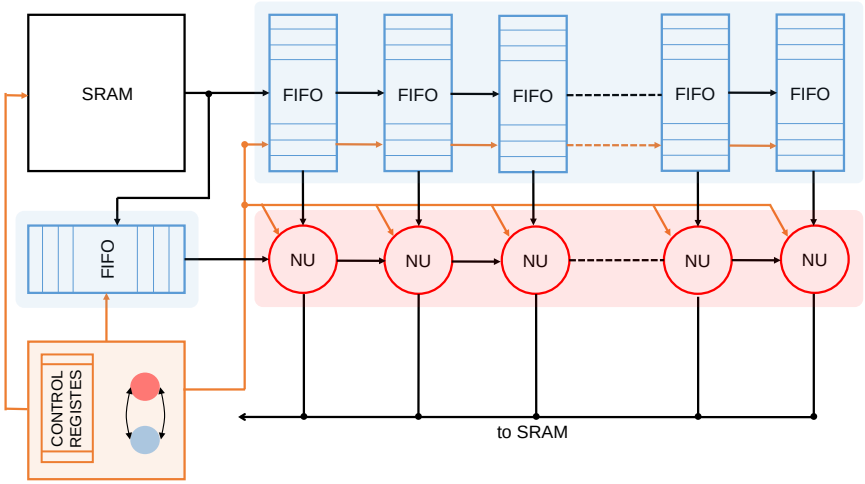

Fig. 5.13. Organization of CMOS architecture for SNNs (SNeuE). The SRAM weights are fetched and stored into the weight FIFOs present in the computation core. Each Neuron Unit (NU) receives its weights from a dedicated weight FIFO.

Application	*Dataset*	*Layers*	*Neurons*	*Synapses*
House Number Recognition	**SVHN**	**6**	**9226**	**16787456**
Object Classification	**CIFAR-10**	**5**	**6666**	**12063744**
Digit Recognition	**MNIST**	**3**	**1546**	**1187328**
Face Recognition	**Yale FR**	**3**	**1039**	**794112**
Census Data Analysis	**Adult**	**2**	**1026**	**8192**
Flower Species Recognition	**Iris Flower**	**3**	**195**	**8384**

Fig. 5.14. Multi-layer perceptron based Spiking Neural Network benchmarks used to compare the All-Spin neuromorphic architecture against the CMOS implementation.

of the dataflow pattern in any typical SNN as neurons in a layer share the inputs. The control unit stores the SNN topology and coordinates dataflow between different components in SNeuE.

Neurons in an SNN are time-multiplexed onto SNeuE to implement the SNN. Within a layer, neurons are scheduled temporally on the NU array. Subsequently, the corresponding weights and inputs are fetched and stored into weight and input FIFO respectively. Once all the computations for the neurons currently scheduled on the NU-array is finished, the next set of neurons from the same layer are scheduled on the NU-array. Eventually successive layers of the SNN are temporally scheduled on the SNeuE computation core to realize one time-step of SNN computation.

A hybrid device-circuit-architecture co-simulation framework was utilized for this work. Device simulations were performed in MuMax [124]. The device characteristics were subsequently used to construct circuit models of such All-Spin SNNs in SPICE for further system level evaluations. The peripheral circuit for the SCA consisting of buffers and control logic was implemented at the Register Transfer Level and mapped to IBM $45nm$ technology using Synopsys Design Compiler. The energy consumption was estimated using Synopsys Power Compiler. The same process was utilized to synthesize and evaluate the energy consumption of the CMOS baseline implementation. CACTI [172] was used to model the SRAM modules. The SCA crossbar size was taken to be 32 rows x 32 columns and the throughput was optimized for each benchmark application to minimize the impact of γ. The NU array used in our evaluations comprises of 16 units. Consequently, there are 16 weight FIFOs in the CMOS implementation (with a FIFO depth of 32). The CMOS baseline implementation was also aggressively optimized by constraining the neuron/synaptic bit discretization to the minimum necessary precision required for negligible accuracy degradation in each specific application. The details of the benchmark suite have been outlined in Fig. 5.14 and consists of the following applications: (i) Flower Species Recognition (IRIS dataset [169]), (ii) Census data analysis (ADULT dataset [169]), (iii) Face recognition (YALE dataset [170]), (iv) Digit recognition (MNIST dataset [156]), (v) Object

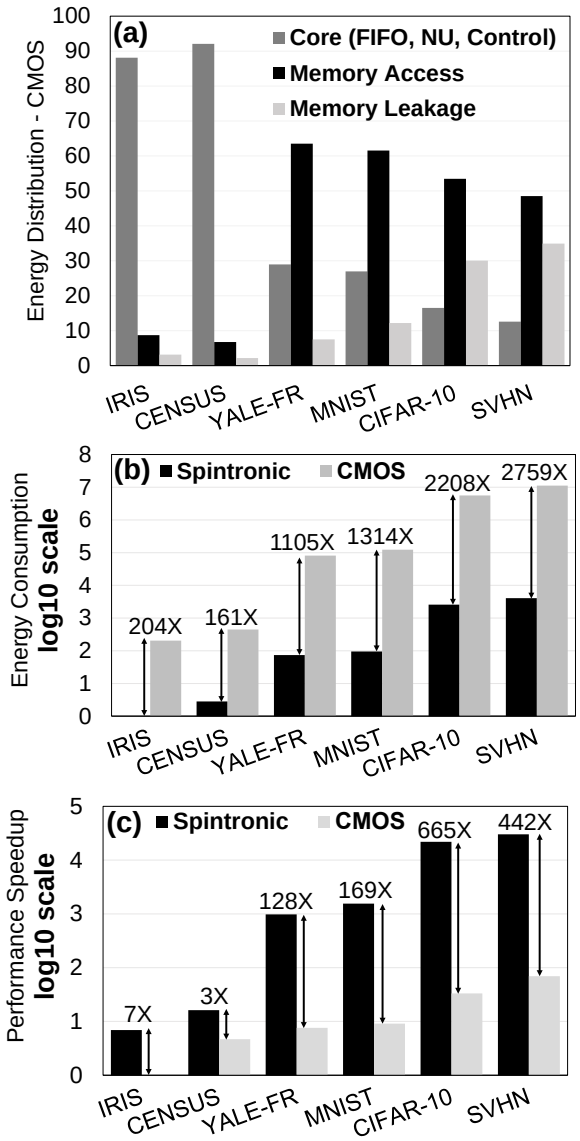

Fig. 5.15. (a) Energy distribution profile for the CMOS architecture, (b) Energy consumption comparison between Spintronic and CMOS architectures, (c) Performance speedup comparison between Spintronic and CMOS architectures [168]. The benchmark suite consists of the following applications: (i) Flower Species Recognition (IRIS dataset [169]), (ii) Census data analysis (ADULT dataset [169]), (iii) Face recognition (YALE dataset [170]), (iv) Digit recognition (MNIST dataset [156]), (v) Object classification (CIFAR-10 dataset [87]) and (vi) House Number Recognition (SVHN dataset [171]).

classification (CIFAR-10 dataset [87]) and (vi) House Number Recognition (SVHN dataset [171]). Note that the analysis performed in this article falls in the domain of offline learning and consequently does not consider the programming energy consumption involved in the learning process of synaptic weights.

Fig. 5.15(a) outlines the proportion of energy consumption involved in memory access and memory leakage in comparison to the core computation. As the problem complexity and hence the network size increases, the amount of energy consumed in memory accesses increases. Additionally, the access latency increases with increasing memory size, thereby causing a proportionate increase in the memory leakage energy. On the other hand, for spintronic crossbar arrays, better crossbar utilization occurs as the network size increases. Fig. 5.15(b) illustrates that the All-Spin SNN architecture can potentially achieve $204 - 2759\times$ improvement in energy consumption while achieving $3 - 665\times$ performance speedup in comparison to the CMOS baseline implementation (Fig. 5.15(c)). Note that the energy consumption (performance speedup) is normalized to the IRIS dataset on the spintronic (CMOS) implementation.

6. CONCLUSIONS AND OUTLOOK

Spin-based neuromorphic computing is currently a technologically evolving field. While preliminary experiments are being performed that provide proof-of-concepts for the various proposals mentioned in this thesis, a long and interesting path lies ahead for the realization of such All-Spin neuromorphic computing platforms. Experimental demonstration of full network-level synaptic learning and neural inference based on spintronic devices remains to be explored. Innovations are still required not only at the device level (for instance, achieving deterministic DW motion or fabricating scaled nanomagnets) but also at the algorithm level to exploit the underlying device physics of spin-devices. Nevertheless, such devices offer immense possibilities towards the realization of energy-efficient cognitive processors. As device dimensions start scaling, probabilistic neuromorphic computing platforms (that are inherently more "brain-like") leveraging the resultant device stochasticity will also start playing an important role. In conclusion, this thesis serves to propose various neural and synaptic functionalities that can be potentially implemented in spintronic devices. We believe that this thesis will stimulate efforts for the realization of All-Spin neuromorphic computing paradigms enabled with on-chip unsupervised cognitive learning capabilities.

REFERENCES

REFERENCES

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[2] S. Adee, "IBM unveils a new brain simulator," *IEEE Spectrum*, 2009.

[3] X. Fong, Y. Kim, R. Venkatesan, S. H. Choday, A. Raghunathan, and K. Roy, "Spin-transfer torque memories: Devices, circuits, and systems," *Proceedings of the IEEE*, vol. 104, no. 7, pp. 1449 – 1488, 2016.

[4] A. Sengupta and K. Roy, "A vision for all-spin neural networks: A device to system perspective," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 12, pp. 2267–2277, 2016.

[5] A. Sengupta, A. Banerjee, and K. Roy, "Hybrid spintronic-CMOS spiking neural network with on-chip learning: Devices, circuits and systems," *Physical Review Applied*, vol. 6, no. 6, p. 064003, 2016.

[6] A. Sengupta, P. Panda, P. Wijesinghe, Y. Kim, and K. Roy, "Magnetic tunnel junction mimics stochastic cortical spiking neurons," *Scientific reports*, vol. 6, 2016.

[7] G. Srinivasan, A. Sengupta, and K. Roy, "Magnetic tunnel junction based long-term short-term stochastic synapse for a spiking neural network with on-chip STDP learning," *Scientific Reports*, vol. 6, p. 29545, 2016.

[8] B. L. Jackson, B. Rajendran, G. S. Corrado, M. Breitwisch, G. W. Burr, R. Cheek, K. Gopalakrishnan, S. Raoux, C. T. Rettner, A. Padilla *et al.*, "Nanoscale electronic synapses using phase change devices," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 9, no. 2, p. 12, 2013.

[9] D. Kuzum, R. G. Jeyasingh, B. Lee, and H.-S. P. Wong, "Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing," *Nano letters*, vol. 12, no. 5, pp. 2179–2186, 2011.

[10] A. Sengupta and K. Roy, "Encoding neural and synaptic functionalities in electron spin: A pathway to efficient neuromorphic computing," *Applied Physics Reviews*, vol. 4, no. 4, p. 041105, 2017.

[11] ——, "Neuromorphic computing enabled by physics of electron spins: Prospects and perspectives," *Applied Physics Express*, vol. 11, no. 3, p. 030101, 2018.

[12] J. C. Slonczewski, "Current-driven excitation of magnetic multilayers," *Journal of Magnetism and Magnetic Materials*, vol. 159, no. 1, pp. L1–L7, 1996.

[13] L. Berger, "Emission of spin waves by a magnetic multilayer traversed by a current," *Physical Review B*, vol. 54, no. 13, p. 9353, 1996.

[14] E. Myers, D. Ralph, J. Katine, R. Louie, and R. Buhrman, "Current-induced switching of domains in magnetic multilayer devices," *Science*, vol. 285, no. 5429, pp. 867–870, 1999.

[15] J. Grollier, V. Cros, A. Hamzic, J.-M. George, H. Jaffrès, A. Fert, G. Faini, J. B. Youssef, and H. Legall, "Spin-polarized current induced switching in Co/Cu/Co pillars," *Applied Physics Letters*, vol. 78, no. 23, pp. 3663–3665, 2001.

[16] S. Yuasa, T. Nagahama, A. Fukushima, Y. Suzuki, and K. Ando, "Giant room-temperature magnetoresistance in single-crystal Fe/MgO/Fe magnetic tunnel junctions," *Nature materials*, vol. 3, no. 12, pp. 868–871, 2004.

[17] M. Julliere, "Tunneling between ferromagnetic films," *Physics letters A*, vol. 54, no. 3, pp. 225–226, 1975.

[18] L. Sun, Y. Hao, C.-L. Chien, and P. C. Searson, "Tuning the properties of magnetic nanowires," *IBM Journal of Research and Development*, vol. 49, no. 1, pp. 79–102, 2005.

[19] A. Driskill-Smith, D. Apalkov, V. Nikitin, X. Tang, S. Watts, D. Lottis, K. Moon, A. Khvalkovskiy, R. Kawakami, X. Luo *et al.*, "Latest advances and roadmap for in-plane and perpendicular STT-RAM," in *2011 3rd IEEE International Memory Workshop (IMW)*, 2011.

[20] G. Jeong, W. Cho, S. Ahn, H. Jeong, G. Koh, Y. Hwang, and K. Kim, "A 0.24-μm 2.0-V 1T1MTJ 16-kb nonvolatile magnetoresistance RAM with self-reference sensing scheme," *IEEE Journal of solid-state circuits*, vol. 38, no. 11, pp. 1906–1910, 2003.

[21] S. Ikeda, K. Miura, H. Yamamoto, K. Mizunuma, H. Gan, M. Endo, S. Kanai, J. Hayakawa, F. Matsukura, and H. Ohno, "A perpendicular-anisotropy CoFeB–MgO magnetic tunnel junction," *Nature materials*, vol. 9, no. 9, pp. 721–724, 2010.

[22] M. Gajek, J. Nowak, J. Sun, P. Trouilloud, E. Osullivan, D. Abraham, M. Gaidis, G. Hu, S. Brown, Y. Zhu *et al.*, "Spin torque switching of 20 nm magnetic tunnel junctions with perpendicular anisotropy," *Applied Physics Letters*, vol. 100, no. 13, p. 132408, 2012.

[23] S. S. Parkin, C. Kaiser, A. Panchula, P. M. Rice, B. Hughes, M. Samant, and S.-H. Yang, "Giant tunnelling magnetoresistance at room temperature with MgO (100) tunnel barriers," *Nature materials*, vol. 3, no. 12, pp. 862–867, 2004.

[24] J. Inoue and T. Shinjo, "GMR, TMR and BMR," *Nanomagnetism and spintronics. Elsevier, Oxford*, pp. 15–92, 2009.

[25] X. Fong, S. K. Gupta, N. N. Mojumder, S. H. Choday, C. Augustine, and K. Roy, "KNACK: A hybrid spin-charge mixed-mode simulator for evaluating different genres of spin-transfer torque MRAM bit-cells," in *Simulation of Semiconductor Processes and Devices (SISPAD), 2011 International Conference on*. IEEE, 2011, pp. 51–54.

[26] J. Z. Sun, "Spin-current interaction with a monodomain magnetic body: A model study," *Physical Review B*, vol. 62, no. 1, p. 570, 2000.

[27] J. C. Slonczewski, "Conductance and exchange coupling of two ferromagnets separated by a tunneling barrier," *Physical Review B*, vol. 39, no. 10, p. 6995, 1989.

[28] W. Scholz, T. Schrefl, and J. Fidler, "Micromagnetic simulation of thermally activated switching in fine particles," *Journal of Magnetism and Magnetic Materials*, vol. 233, no. 3, pp. 296–304, 2001.

[29] W. F. Brown Jr, "Thermal fluctuations of a single-domain particle," *Journal of Applied Physics*, vol. 34, no. 4, pp. 1319–1320, 1963.

[30] R. Matsumoto, A. Chanthbouala, J. Grollier, V. Cros, A. Fert, K. Nishimura, Y. Nagamine, H. Maehara, K. Tsunekawa, A. Fukushima *et al.*, "Spin-torque diode measurements of MgO-based magnetic tunnel junctions with asymmetric electrodes," *Applied physics express*, vol. 4, no. 6, p. 063001, 2011.

[31] R. D. McMichael and M. J. Donahue, "Head to head domain wall structures in thin magnetic strips," *IEEE Transactions on Magnetics*, vol. 33, no. 5, pp. 4167–4169, 1997.

[32] E. Torok, A. Olson, and H. Oredson, "Transition between Bloch and Néel walls," *Journal of Applied Physics*, vol. 36, no. 4, pp. 1394–1399, 1965.

[33] X. Fong, Y. Kim, K. Yogendra, D. Fan, A. Sengupta, A. Raghunathan, and K. Roy, "Spin-transfer torque devices for logic and memory: Prospects and perspectives," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 35, no. 1, pp. 1–22, 2016.

[34] L. Berger, "Low-field magnetoresistance and domain drag in ferromagnets," *Journal of Applied Physics*, vol. 49, no. 3, pp. 2156–2161, 1978.

[35] A. Yamaguchi, S. Nasu, H. Tanigawa, T. Ono, K. Miyake, K. Mibu, and T. Shinjo, "Effect of Joule heating in current-driven domain wall motion," *Applied Physics Letters*, vol. 86, no. 1, p. 012511, 2005.

[36] G. Beach, M. Tsoi, and J. Erskine, "Current-induced domain wall motion," *Journal of magnetism and magnetic materials*, vol. 320, no. 7, pp. 1272–1281, 2008.

[37] A. Brataas and K. M. Hals, "Spin-orbit torques in action," *Nature nanotechnology*, vol. 9, no. 2, pp. 86–88, 2014.

[38] I. M. Miron, T. Moore, H. Szambolics, L. D. Buda-Prejbeanu, S. Auffret, B. Rodmacq, S. Pizzini, J. Vogel, M. Bonfim, A. Schuhl *et al.*, "Fast current-induced domain-wall motion controlled by the Rashba effect," *Nature Materials*, vol. 10, no. 6, pp. 419–423, 2011.

[39] J. Hirsch, "Spin Hall effect," *Physical Review Letters*, vol. 83, no. 9, p. 1834, 1999.

[40] L. Liu, C.-F. Pai, Y. Li, H. Tseng, D. Ralph, and R. Buhrman, "Spin-torque switching with the giant spin Hall effect of tantalum," *Science*, vol. 336, no. 6081, pp. 555–558, 2012.

[41] G. Yu, P. Upadhyaya, Y. Fan, J. G. Alzate, W. Jiang, K. L. Wong, S. Takei, S. A. Bender, L.-T. Chang, Y. Jiang *et al.*, "Switching of perpendicular magnetization by spin-orbit torques in the absence of external magnetic fields," *Nature nanotechnology*, vol. 9, no. 7, pp. 548–554, 2014.

[42] L. Liu, O. Lee, T. Gudmundsen, D. Ralph, and R. Buhrman, "Current-induced switching of perpendicularly magnetized magnetic layers using spin torque from the spin Hall effect," *Physical review letters*, vol. 109, no. 9, p. 096602, 2012.

[43] S. Emori, U. Bauer, S.-M. Ahn, E. Martinez, and G. S. Beach, "Current-driven dynamics of chiral ferromagnetic domain walls," *Nature materials*, vol. 12, no. 7, pp. 611–616, 2013.

[44] G. Chen, J. Zhu, A. Quesada, J. Li, A. NDiaye, Y. Huo, T. Ma, Y. Chen, H. Kwon, C. Won *et al.*, "Novel chiral magnetic domain wall structure in Fe/Ni/Cu (001) films," *Physical review letters*, vol. 110, no. 17, p. 177204, 2013.

[45] K.-S. Ryu, L. Thomas, S.-H. Yang, and S. Parkin, "Chiral spin torque at magnetic domain walls," *Nature nanotechnology*, vol. 8, no. 7, pp. 527–533, 2013.

[46] D. Bhowmik, M. E. Nowakowski, L. You, O. Lee, D. Keating, M. Wong, J. Bokor, and S. Salahuddin, "Deterministic domain wall motion orthogonal to current flow due to spin orbit torque," *Scientific reports*, vol. 5, 2015.

[47] N. Perez, L. Torres, and E. Martinez-Vecino, "Micromagnetic modeling of Dzyaloshinskii–Moriya interaction in spin Hall effect switching," *Magnetics, IEEE Transactions on*, vol. 50, no. 11, pp. 1–4, 2014.

[48] E. Martinez, S. Emori, N. Perez, L. Torres, and G. S. Beach, "Current-driven dynamics of Dzyaloshinskii domain walls in the presence of in-plane fields: Full micromagnetic and one-dimensional analysis," *Journal of Applied Physics*, vol. 115, no. 21, p. 213909, 2014.

[49] S. Emori, E. Martinez, K.-J. Lee, H.-W. Lee, U. Bauer, S.-M. Ahn, P. Agrawal, D. C. Bono, and G. S. Beach, "Spin hall torque magnetometry of Dzyaloshinskii domain walls," *Physical Review B*, vol. 90, no. 18, p. 184427, 2014.

[50] Y. Ji, A. Hoffmann, J. Jiang, and S. Bader, "Spin injection, diffusion, and detection in lateral spin-valves," *Applied physics letters*, vol. 85, no. 25, pp. 6218–6220, 2004.

[51] Y. Fukuma, L. Wang, H. Idzuchi, S. Takahashi, S. Maekawa, and Y. Otani, "Giant enhancement of spin accumulation and long-distance spin precession in metallic lateral spin valves," *Nature materials*, vol. 10, no. 7, pp. 527–531, 2011.

[52] T. Yang, T. Kimura, and Y. Otani, "Giant spin-accumulation signal and pure spin-current-induced reversible magnetization switching," *Nature Physics*, vol. 4, no. 11, pp. 851–854, 2008.

[53] K.-S. Ryu, S.-H. Yang, L. Thomas, and S. S. Parkin, "Chiral spin torque arising from proximity-induced magnetization," *Nature communications*, vol. 5, 2014.

[54] P. K. Amiri and K. L. Wang, "Voltage-controlled magnetic anisotropy in spintronic devices," in *Spin*, vol. 2, no. 03. World Scientific, 2012, p. 1240002.

[55] J. Heron, M. Trassin, K. Ashraf, M. Gajek, Q. He, S. Yang, D. Nikonov, Y. Chu, S. Salahuddin, and R. Ramesh, "Electric-field-induced magnetization reversal in a ferromagnet-multiferroic heterostructure," *Physical review letters*, vol. 107, no. 21, p. 217202, 2011.

[56] K. J. Franke, B. Van de Wiele, Y. Shirahata, S. J. Hämäläinen, T. Taniyama, and S. van Dijken, "Reversible electric-field-driven magnetic domain-wall motion," *Physical Review X*, vol. 5, no. 1, p. 011010, 2015.

[57] A. Mellnik, J. Lee, A. Richardella, J. Grab, P. Mintun, M. H. Fischer, A. Vaezi, A. Manchon, E.-A. Kim, N. Samarth *et al.*, "Spin transfer torque generated by the topological insulator Bi_2Se_3," *arXiv preprint arXiv:1402.1124*, 2014.

[58] Y. Fan, P. Upadhyaya, X. Kou, M. Lang, S. Takei, Z. Wang, J. Tang, L. He, L.-T. Chang, M. Montazeri *et al.*, "Magnetization switching through giant spin–orbit torque in a magnetically doped topological insulator heterostructure," *Nature materials*, vol. 13, no. 7, pp. 699–704, 2014.

[59] A. Hirohata, J. Sagar, L. R. Fleet, and S. S. Parkin, "Heusler alloy films for spintronic devices," in *Heusler Alloys*. Springer, 2016, pp. 219–248.

[60] S.-H. Yang, K.-S. Ryu, and S. Parkin, "Domain-wall velocities of up to 750 m/s driven by exchange-coupling torque in synthetic antiferromagnets," *Nature nanotechnology*, vol. 10, no. 3, pp. 221–226, 2015.

[61] T. Shiino, S.-H. Oh, P. M. Haney, S.-W. Lee, G. Go, B.-G. Park, and K.-J. Lee, "Antiferromagnetic domain wall motion driven by spin-orbit torques," *arXiv preprint arXiv:1604.01473*, 2016.

[62] S. Woo, K. Litzius, B. Krüger, M.-Y. Im, L. Caretta, K. Richter, M. Mann, A. Krone, R. M. Reeve, M. Weigand *et al.*, "Observation of room-temperature magnetic skyrmions and their current-driven dynamics in ultrathin metallic ferromagnets," *Nature materials*, 2016.

[63] W. Kang, Y. Huang, X. Zhang, Y. Zhou, and W. Zhao, "Skyrmion-electronics: An overview and outlook," *Proceedings of the IEEE*, vol. 104, no. 10, pp. 2040–2061, 2016.

[64] A. Jaiswal, S. Roy, G. Srinivasan, and K. Roy, "Proposal for a leaky-integrate-fire spiking neuron based on magnetoelectric switching of ferromagnets," *IEEE Transactions on Electron Devices*, vol. 64, no. 4, pp. 1818–1824, 2017.

[65] Z. He and D. Fan, "A tunable magnetic skyrmion neuron cluster for energy efficient artificial neural network," in *2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2017, pp. 350–355.

[66] Y. Huang, W. Kang, X. Zhang, Y. Zhou, and W. Zhao, "Magnetic skyrmion-based synaptic devices," *Nanotechnology*, vol. 28, no. 8, p. 08LT02, 2017.

[67] S. Li, W. Kang, Y. Huang, X. Zhang, Y. Zhou, and W. Zhao, "Magnetic skyrmion-based artificial neuron device," *Nanotechnology*, vol. 28, no. 31, p. 31LT01, 2017.

[68] X.-G. Zhang and W. Butler, "Large magnetoresistance in BCC Co/MgO/Co and FeCo/MgO/FeCo tunnel junctions," *Physical Review B*, vol. 70, no. 17, p. 172407, 2004.

[69] S. Ikeda, J. Hayakawa, Y. Ashizawa, Y. Lee, K. Miura, H. Hasegawa, M. Tsunoda, F. Matsukura, and H. Ohno, "Tunnel magnetoresistance of 604% at 300 K by suppression of Ta diffusion in CoFeB/MgO/CoFeB pseudo-spin-valves annealed at high temperature," *Applied Physics Letters*, vol. 93, no. 8, p. 2508, 2008.

[70] A. Hirohata, H. Sukegawa, H. Yanagihara, I. Žutić, T. Seki, S. Mizukami, and R. Swaminathan, "Roadmap for emerging materials for spintronic device applications," *IEEE Transactions on Magnetics*, vol. 51, no. 10, pp. 1–11, 2015.

[71] M. de Kamps and F. van der Velde, "From artificial neural networks to spiking neuron populations and back again," *Neural Networks*, vol. 14, no. 6, pp. 941–953, 2001.

[72] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nature nanotechnology*, vol. 8, no. 1, pp. 13–24, 2013.

[73] M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, pp. 61–64, 2015.

[74] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[75] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G.-J. Nam *et al.*, "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.

[76] R. Hecht-Nielsen *et al.*, "Theory of the backpropagation neural network," *Neural Networks*, vol. 1, no. Supplement-1, pp. 445–448, 1988.

[77] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[78] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *International journal of neural systems*, vol. 19, no. 04, pp. 295–308, 2009.

[79] R. Brette, "Philosophy of the spike: Rate-based vs. spike-based theories of the brain," *Frontiers in systems neuroscience*, vol. 9, p. 151, 2015.

[80] V. Chan, S.-C. Liu, and A. van Schaik, "AER EAR: A matched silicon cochlea pair with address event representation interface," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 1, pp. 48–59, 2007.

[81] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity," *IEEE transactions on neural networks*, vol. 17, no. 1, pp. 211–221, 2006.

[82] B. Han, A. Sengupta, and K. Roy, "On the energy benefits of spiking deep neural networks: A case study," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 971–976.

[83] B. Han, A. Ankit, A. Sengupta, and K. Roy, "Cross-layer design exploration for energy-quality tradeoffs in spiking and non-spiking deep artificial neural networks," *IEEE Transactions on Multi-Scale Computing Systems*, 2017.

[84] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: VGG and residual architectures," *arXiv preprint arXiv:1802.02627*, 2018.

[85] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015, pp. 1–8.

[86] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *International Journal of Computer Vision*, vol. 113, no. 1, pp. 54–66, 2015.

[87] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[88] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[89] E. Wallace, M. Benayoun, W. Van Drongelen, and J. D. Cowan, "Emergent oscillations in networks of stochastic spiking neurons," *Plos one*, vol. 6, no. 5, p. e14804, 2011.

[90] M. Benayoun, J. D. Cowan, W. van Drongelen, and E. Wallace, "Avalanches in a stochastic model of spiking neurons," *PLoS Comput Biol*, vol. 6, no. 7, p. e1000846, 2010.

[91] B. Nessler, M. Pfeiffer, and W. Maass, "STDP enables spiking neurons to detect hidden causes of their inputs," in *Advances in neural information processing systems*, Vancouver, B.C., Canada, 2009, Dec, pp. 1357–1365.

[92] B. Nessler, M. Pfeiffer, L. Buesing, and W. Maass, "Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity," *PLoS Comput Biol*, vol. 9, no. 4, p. e1003037, 2013.

[93] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[94] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[95] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in Computational Neuroscience*, 2015.

[96] G.-q. Bi and M.-m. Poo, "Synaptic modification by correlated activity: Hebb's postulate revisited," *Annual review of neuroscience*, vol. 24, no. 1, pp. 139–166, 2001.

[97] D. Kuzum, R. G. D. Jeyasingh, S. Yu, and H.-S. P. Wong, "Low-energy robust neuromorphic computation using synaptic devices," *IEEE Transactions on Electron Devices*, vol. 59, no. 12, pp. 3489–3494, 2012.

[98] D. S. Modha and S. S. Parkin, "Stochastic synapse memory element with spike-timing dependent plasticity (STDP)," 2011, US Patent 7,978,510.

[99] A. F. Vincent, J. Larroque, N. Locatelli, N. B. Romdhane, O. Bichler, C. Gamrat, W. S. Zhao, J.-O. Klein, S. Galdin-Retailleau, and D. Querlioz, "Spin-transfer torque magnetic memory as a stochastic memristive synapse for neuromorphic systems," *IEEE transactions on biomedical circuits and systems*, vol. 9, no. 2, pp. 166–174, 2015.

[100] R. S. Zucker and W. G. Regehr, "Short-term synaptic plasticity," *Annual review of physiology*, vol. 64, no. 1, pp. 355–405, 2002.

[101] S. Martin, P. Grimwood, and R. Morris, "Synaptic plasticity and memory: An evaluation of the hypothesis," *Annual review of neuroscience*, vol. 23, no. 1, pp. 649–711, 2000.

[102] R. C. Atkinson and R. M. Shiffrin, "Human memory: A proposed system and its control processes," *Psychology of learning and motivation*, vol. 2, pp. 89–195, 1968.

[103] R. Lamprecht and J. LeDoux, "Structural plasticity and memory," *Nature Reviews Neuroscience*, vol. 5, no. 1, pp. 45–54, 2004.

[104] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[105] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[106] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Interspeech*, vol. 2, 2010, p. 3.

[107] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[108] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[109] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, 1990.

[110] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, "Neuromorphic electronic circuits for building autonomous cognitive systems," *Proceedings of the IEEE*, vol. 102, no. 9, pp. 1367–1388, 2014.

[111] P. Krzysteczko, J. Münchenberger, M. Schäfers, G. Reiss, and A. Thomas, "The memristive magnetic tunnel junction as a nanoscopic synapse-neuron system," *Advanced Materials*, vol. 24, no. 6, pp. 762–766, 2012.

[112] A. Sengupta and K. Roy, "Spin-transfer torque magnetic neuron for low power neuromorphic computing," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–7.

[113] A. Sengupta, S. H. Choday, Y. Kim, and K. Roy, "Spin orbit torque based electronic neuron," *Applied Physics Letters*, vol. 106, no. 14, p. 143701, 2015.

[114] A. Imre, G. Csaba, L. Ji, A. Orlov, G. Bernstein, and W. Porod, "Majority logic gate for magnetic quantum-dot cellular automata," *Science*, vol. 311, no. 5758, pp. 205–208, 2006.

[115] B. Behin-Aein, D. Datta, S. Salahuddin, and S. Datta, "Proposal for an all-spin logic device with built-in memory," *Nature nanotechnology*, vol. 5, no. 4, pp. 266–270, 2010.

[116] A. Van den Brink, S. Cosemans, S. Cornelissen, M. Manfrini, A. Vaysset, W. Van Roy, T. Min, H. Swagten, and B. Koopmans, "Spin-Hall-assisted magnetic random access memory," *Applied Physics Letters*, vol. 104, no. 1, p. 012403, 2014.

[117] Y. Acremann, X. Yu, A. Tulapurkar, A. Scherz, V. Chembrolu, J. Katine, M. Carey, H. Siegmann, and J. Stöhr, "An amplifier concept for spintronics," *Applied Physics Letters*, vol. 93, no. 10, p. 102513, 2008.

[118] C.-F. Pai, L. Liu, Y. Li, H. Tseng, D. Ralph, and R. Buhrman, "Spin transfer torque devices utilizing the giant spin Hall effect of tungsten," *Applied Physics Letters*, vol. 101, no. 12, p. 122404, 2012.

[119] A. Sengupta, Y. Shim, and K. Roy, "Proposal for an all-spin artificial neural network: Emulating neural and synaptic functionalities through domain wall motion in ferromagnets," *IEEE Transactions on Biomedical Circuits and Systems*, 2016.

[120] A. Sengupta, B. Han, and K. Roy, "Toward a spintronic deep learning spiking neural processor," in *Biomedical Circuits and Systems Conference (BioCAS), 2016 IEEE*. IEEE.

[121] A. Sengupta, Z. Al Azim, X. Fong, and K. Roy, "Spin-orbit torque induced spike-timing dependent plasticity," *Applied Physics Letters*, vol. 106, no. 9, p. 093704, 2015.

[122] S. Lequeux, J. Sampaio, V. Cros, K. Yakushiji, A. Fukushima, R. Matsumoto, H. Kubota, S. Yuasa, and J. Grollier, "A magnetic synapse: multilevel spin-torque memristor with perpendicular anisotropy," *Scientific Reports*, vol. 6, 2016.

[123] C. Lin, S. Kang, Y. Wang, K. Lee, X. Zhu, W. Chen, X. Li, W. Hsu, Y. Kao, M. Liu *et al.*, "45nm low power CMOS logic compatible embedded STT MRAM utilizing a reverse-connection 1T/1MTJ cell," in *2009 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2009, pp. 1–4.

[124] A. Vansteenkiste, J. Leliaert, M. Dvornik, M. Helsen, F. Garcia-Sanchez, and B. Van Waeyenberge, "The design and verification of MuMax3," *AIP Advances*, vol. 4, no. 10, p. 107133, 2014.

[125] R. Venkatesan, V. Kozhikkottu, C. Augustine, A. Raychowdhury, K. Roy, and A. Raghunathan, "TapeCache: a high density, energy efficient cache based on domain wall memory," in *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*. ACM, 2012, pp. 185–190.

[126] Z. Al Azim, A. Sengupta, S. S. Sarwar, and K. Roy, "Spin-torque sensors for energy efficient high-speed long interconnects," *IEEE Transactions on Electron Devices*, vol. 63, no. 2, pp. 800–808, 2016.

[127] A. Sengupta, M. Parsa, B. Han, and K. Roy, "Probabilistic deep spiking neural systems enabled by magnetic tunnel junction," *IEEE Transactions on Electron Devices*, vol. 63, no. 7, p. 2963, 2016.

[128] B. Rajendran, Y. Liu, J.-s. Seo, K. Gopalakrishnan, L. Chang, D. J. Friedman, and M. B. Ritter, "Specifications of nanoscale devices and circuits for neuromorphic computational systems," *IEEE Transactions on Electron Devices*, vol. 60, no. 1, pp. 246–253, 2013.

[129] G. Indiveri, "A low-power adaptive integrate-and-fire neuron circuit," in *ISCAS (4)*. Citeseer, 2003, pp. 820–823.

[130] A. Pouget, J. M. Beck, W. J. Ma, and P. E. Latham, "Probabilistic brains: Knowns and unknowns," *Nature neuroscience*, vol. 16, no. 9, pp. 1170–1178, 2013.

[131] J. Alspector, B. Gupta, and R. B. Allen, "Performance of a stochastic learning microchip," in *Advances in neural information processing systems*, 1989, pp. 748–760.

[132] B. Behin-Aein, V. Diep, and S. Datta, "A building block for hardware belief networks," *Sci. Rep.*, vol. 6, p. 29893, 2016.

[133] B. Sutton, K. Y. Camsari, B. Behin-Aein, and S. Datta, "Intrinsic optimization using stochastic nanomagnets," *Sci. Rep.*, vol. 7, no. 44370, 2017.

[134] Y. Shim, S. Chen, A. Sengupta, and K. Roy, "Stochastic spin-orbit torque devices as elements for bayesian inference," *Scientific Reports*, vol. 7, no. 1, p. 14101, 2017.

[135] Y. Shim, A. Jaiswal, and K. Roy, "Ising computation based combinatorial optimization using spin-Hall effect (SHE) induced stochastic magnetization reversal," *Journal of Applied Physics*, vol. 121, no. 19, p. 193902, 2017.

[136] D. Vodenicarevic, N. Locatelli, A. Mizrahi, J. S. Friedman, A. F. Vincent, M. Romera, A. Fukushima, K. Yakushiji, H. Kubota, S. Yuasa *et al.*, "Low-energy truly random number generation with superparamagnetic tunnel junctions for unconventional computing," *Physical Review Applied*, vol. 8, no. 5, p. 054045, 2017.

[137] Y. Shim, A. Sengupta, and K. Roy, "Biased random walk using stochastic switching of nanomagnets: Application to SAT solver," *IEEE Transactions on Electron Devices*, vol. 65, no. 4, pp. 1617–1624, 2018.

[138] A. Sengupta, C. M. Liyanagedera, B. Jung, and K. Roy, "Magnetic tunnel junction as an on-chip temperature sensor," *Scientific Reports*, vol. 7, no. 1, p. 11764, 2017.

[139] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, "Stochastic p-bits for invertible logic," *Physical Review X*, vol. 7, no. 3, p. 031014, 2017.

[140] R. Faria, K. Y. Camsari, and S. Datta, "Low-barrier nanomagnets as p-bits for spin logic," *IEEE Magnetics Letters*, vol. 8, pp. 1–5, 2017.

[141] P. Livi and G. Indiveri, "A current-mode conductance-based silicon neuron for address-event neuromorphic systems," in *Circuits And Systems (ISCAS), 2009 International Symposium On.* Taipei, Taiwan: IEEE, 2009, May 24, pp. 2898–2901.

[142] A. Joubert, B. Belhadj, O. Temam, and R. Héliot, "Hardware spiking neurons design: Analog or digital?" in *Neural Networks (IJCNN), The 2012 International Joint Conference on.* Brisbane, Australia: IEEE, 2012, June 10, pp. 1–5.

[143] C. M. Liyanagedera, A. Sengupta, A. Jaiswal, and K. Roy, "Stochastic spiking neural networks enabled by magnetic tunnel junctions: From nontelegraphic to telegraphic switching regimes," *Phys. Rev. Applied*, vol. 8, p. 064017, Dec 2017.

[144] W. Rippard, R. Heindl, M. Pufall, S. Russek, and A. Kos, "Thermal relaxation rates of magnetic nanoparticles in the presence of magnetic fields and spin-transfer effects," *Physical Review B*, vol. 84, no. 6, p. 064439, 2011.

[145] N. Locatelli, A. Mizrahi, A. Accioly, R. Matsumoto, A. Fukushima, H. Kubota, S. Yuasa, V. Cros, L. G. Pereira, D. Querlioz *et al.*, "Noise-enhanced synchronization of stochastic magnetic oscillators," *Physical Review Applied*, vol. 2, no. 3, p. 034009, 2014.

[146] M. Bapna and S. A. Majetich, "Current control of time-averaged magnetization in superparamagnetic tunnel junctions," *Applied Physics Letters*, vol. 111, no. 24, p. 243107, 2017.

[147] Y. Li, Y. Zhong, J. Zhang, L. Xu, Q. Wang, H. Sun, H. Tong, X. Cheng, and X. Miao, "Activity-dependent synaptic plasticity of a chalcogenide electronic synapse for neuromorphic systems," *Scientific reports*, vol. 4, 2014.

[148] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano letters*, vol. 10, no. 4, pp. 1297–1301, 2010.

[149] Y. Nishitani, Y. Kaneko, M. Ueda, E. Fujii, and A. Tsujimura, "Dynamic observation of brain-like learning in a ferroelectric synapse device," *Japanese Journal of Applied Physics*, vol. 52, no. 4S, p. 04CE06, 2013.

[150] S. Ramakrishnan, P. E. Hasler, and C. Gordon, "Floating gate synapses with spike-time-dependent plasticity," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 5, no. 3, pp. 244–252, 2011.

[151] A. Sengupta and K. Roy, "Short-term plasticity and long-term potentiation in magnetic tunnel junctions: Towards volatile synapses," *Physical Review Applied*, vol. 5, no. 2, p. 024012, 2016.

[152] K. Magleby, "The effect of repetitive stimulation on facilitation of transmitter release at the frog neuromuscular junction," *The Journal of physiology*, vol. 234, no. 2, p. 327, 1973.

[153] T. Chang, S.-H. Jo, and W. Lu, "Short-term memory to long-term memory transition in a nanoscale memristor," *ACS nano*, vol. 5, no. 9, pp. 7669–7676, 2011.

[154] T. Ohno, T. Hasegawa, T. Tsuruoka, K. Terabe, J. K. Gimzewski, and M. Aono, "Short-term plasticity and long-term potentiation mimicked in single inorganic synapses," *Nature materials*, vol. 10, no. 8, pp. 591–595, 2011.

[155] R. Yang, K. Terabe, Y. Yao, T. Tsuruoka, T. Hasegawa, J. K. Gimzewski, and M. Aono, "Synaptic plasticity and memory functions achieved in a WO_{3-x}-based nanoionics device by using the principle of atomic switch operation," *Nanotechnology*, vol. 24, no. 38, p. 384003, 2013.

[156] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[157] D. Lacour, J. Katine, L. Folks, T. Block, J. Childress, M. Carey, and B. Gurney, "Experimental evidence of multiple stable locations for a domain wall trapped by a submicron notch," *Applied physics letters*, vol. 84, no. 11, pp. 1910–1912, 2004.

[158] J. Lazzaro and J. Wawrzynek, *Low-power silicon neurons, axons and synapses.* Springer, 1994.

[159] C. Bartolozzi and G. Indiveri, "Synaptic dynamics in analog VLSI," *Neural computation*, vol. 19, no. 10, pp. 2581–2603, 2007.

[160] D. F. Goodman and R. Brette, "The Brian simulator," *Frontiers in neuroscience*, vol. 3, no. 2, p. 192, 2009.

[161] D. Querlioz, O. Bichler, P. Dollfus, and C. Gamrat, "Immunity to device variations in a spiking neural network with memristive nanodevices," *IEEE Transactions on Nanotechnology*, vol. 12, no. 3, pp. 288–295, 2013.

[162] H. Noguchi, K. Ikegami, K. Kushida, K. Abe, S. Itai, S. Takaya, N. Shimomura, J. Ito, A. Kawasumi, H. Hara *et al.*, "7.5 A 3.3 ns-access-time 71.2μW/MHz 1Mb embedded STT-MRAM using physically eliminated read-disturb scheme and normally-off memory architecture," in *2015 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers.* IEEE, 2015, pp. 1–3.

[163] A. Ankit, A. Sengupta, P. Panda, and K. Roy, "RESPARC: A reconfigurable and energy-efficient architecture with memristive crossbars for deep spiking neural networks," in *Proceedings of the 54th Annual Design Automation Conference 2017.* ACM, 2017, p. 27.

[164] R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," 2012.

[165] D. Heeger, "Poisson model of spike generation," *Handout, University of Standford*, vol. 5, 2000.

[166] M. Sharad, D. Fan, and K. Roy, "Spin-neurons: A possible path to energy-efficient neuromorphic computers," *Journal of Applied Physics*, vol. 114, no. 23, p. 234906, 2013.

[167] G. Srinivasan, A. Sengupta, and K. Roy, "Magnetic tunnel junction enabled all-spin stochastic spiking neural network," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*. IEEE, 2017.

[168] A. Sengupta, A. Ankit, and K. Roy, "Performance analysis and benchmarking of all-spin spiking neural networks," in *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, 2017.

[169] UCI Machine Learning Repository. [Online]. Available: http://archive.ics.uci.edu/ml/datasets.html

[170] Yale Face Database. [Online]. Available: http://cvc.yale.edu/projects/yalefaces/yalefaces.html

[171] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[172] N. Muralimanohar, R. Balasubramonian, and N. Jouppi, "Optimizing NUCA organizations and wiring alternatives for large caches with CACTI 6.0," in *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE Computer Society, 2007, pp. 3–14.

[173] C. Farabet, R. Paz, J. Pérez-Carrasco, C. Zamarreño-Ramos, A. Linares-Barranco, Y. LeCun, E. Culurciello, T. Serrano-Gotarredona, and B. Linares-Barranco, "Comparison between frame-constrained fix-pixel-value and frame-free spiking-dynamic-pixel ConvNets for visual processing," *Frontiers in neuroscience*, vol. 6, 2012.

[174] Z. Chen, M. Johnson, L. Wei, and K. Roy, "Estimation of standby leakage power in CMOS circuit considering accurate modeling of transistor stacks," in *Low Power Electronics and Design, 1998. Proceedings. 1998 International Symposium on*. IEEE, 1998, pp. 239–244.

[175] B. Zhao, R. Ding, S. Chen, B. Linares-Barranco, and H. Tang, "Feedforward categorization on AER motion events using cortex-like features in a spiking neural network," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 9, pp. 1963–1978, 2015.

[176] J. A. Pérez-Carrasco, B. Zhao, C. Serrano, B. Acha, T. Serrano-Gotarredona, S. Chen, and B. Linares-Barranco, "Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–Application to feedforward ConvNets," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2706–2719, 2013.

[177] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck, "Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output," *Proceedings of the IEEE*, vol. 102, no. 10, pp. 1470–1484, 2014.

[178] C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, 2011.

[179] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[180] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on neural networks*, vol. 14, no. 6, pp. 1569–1572, 2003.

[181] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[182] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting." *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[183] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[184] ——, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.

[185] *Link*. [Online]. Available: http://torch.ch/blog/2016/02/04/resnets.html

[186] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[187] *Link*. [Online]. Available: https://github.com/facebook/fb.resnet.torch

[188] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[189] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[190] M. Hardt and T. Ma, "Identity matters in deep learning," *arXiv preprint arXiv:1611.04231*, 2016.

[191] *Link*. [Online]. Available: https://github.com/szagoruyko/cifar.torch

[192] E. Hunsberger and C. Eliasmith, "Training spiking deep networks for neuromorphic hardware," *arXiv preprint arXiv:1611.05141*, 2016.

[193] B. Rueckauer, Y. Hu, I.-A. Lungu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in neuroscience*, vol. 11, p. 682, 2017.

[194] S. K. Esser, P. A. Merolla, J. V. Arthur, A. S. Cassidy, R. Appuswamy, A. Andreopoulos, D. J. Berg, J. L. McKinstry, T. Melano, D. R. Barch *et al.*, "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proceedings of the National Academy of Sciences*, p. 201604850, 2016.

[195] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in neural information processing systems*, 2015, pp. 1135–1143.

[196] C.-C. Chen and H.-W. Chen, "A low-cost CMOS smart temperature sensor using a thermal-sensing and pulse-shrinking delay line," *IEEE Sensors Journal*, vol. 14, no. 1, pp. 278–284, 2014.

[197] Y.-L. Lo and Y.-T. Chiu, "A high-accuracy, high-resolution, and low-cost all-digital temperature sensor using a voltage compensation ring oscillator," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 43–52, 2016.

[198] T.-H. Tran, H.-W. Peng, P. C.-P. Chao, and J.-W. Hsieh, "A low-ppm digitally controlled crystal oscillator compensated by a new $0.19 - mm^2$ time-domain temperature sensor," *IEEE Sensors Journal*, vol. 17, no. 1, pp. 51–62, 2017.

[199] C. Deng, Y. Sheng, S. Wang, W. Hu, S. Diao, and D. Qian, "A CMOS smart temperature sensor with single-point calibration method for clinical use," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 2, pp. 136–140, 2016.

APPENDICES

A. SCALING SPIKING NEURAL NETWORKS TO DEEP ARCHITECTURES FOR COMPLEX MACHINE LEARNING TASKS

Over the past few years, Spiking Neural Networks (SNNs) have become popular as a possible pathway to enable low-power event-driven neuromorphic hardware. However, their application in machine learning have largely been limited to very shallow neural network architectures for simple problems. In this appendix, we propose a novel algorithmic technique for generating an SNN with a deep architecture, and demonstrate its effectiveness on complex visual recognition problems such as CIFAR-10 and ImageNet. Our technique applies to both VGG and Residual network architectures, with significantly better accuracy than the state-of-the-art. Finally, we present analysis of the sparse event-driven computations to demonstrate reduced hardware overhead when operating in the spiking domain.

A.1 Introduction

Spiking Neural Networks (SNNs) are a significant shift from the standard way of operation of Artificial Neural Networks [173]. Most of the success of deep learning models of neural networks in complex pattern recognition tasks are based on neural units that receive, process and transmit analog information. Such Analog Neural Networks (ANNs) [85], however, disregard the fact that the biological neurons in the brain (the computing framework after which it is inspired) processes binary spike-based information. Driven by this observation, the past few years have witnessed significant progress in the modeling and formulation of training schemes for SNNs as a new computing paradigm that can potentially replace ANNs as the next generation of Neural Networks. In addition to the fact that SNNs are inherently more

biologically plausible, they offer the prospect of event-driven hardware operation. Spiking Neurons process input information only on the receipt of incoming binary spike signals. Given a sparsely-distributed input spike train, the hardware overhead (power consumption) for such a spike or event-based hardware would be significantly reduced since large sections of the network that are not driven by incoming spikes can be power-gated [174]. However, the vast majority of research on SNNs have been limited to very simple and shallow network architectures on relatively simple digit recognition datasets like MNIST [156] while only few works report their performance on more complex standard vision datasets like CIFAR-10 [87] and ImageNet [88]. The main reason behind their limited performance stems from the fact that SNNs are a significant shift from the operation of ANNs due to their temporal information processing capability. This has necessitated a rethinking of training mechanisms for SNNs.

Broadly, there are two main categories for training SNNs - supervised and unsupervised. Although unsupervised learning mechanisms like Spike-Timing Dependent Plasticity (STDP) are attractive for the implementation of low-power on-chip local learning, their performance is still outperformed by supervised networks on even simple digit recognition platforms like the MNIST dataset [95]. Driven by this fact, a particular category of supervised SNN learning algorithms attempts to train ANNs using standard training schemes like backpropagation (to leverage the superior performance of standard training techniques for ANNs) and subsequently convert to event-driven SNNs for network operation [85, 86, 175, 176]. This can be particularly appealing for NN implementations in low-power neuromorphic hardware specialized for SNNs [74, 75] or interfacing with silicon cochleas or event-driven sensors [177, 178]. Our work falls in this category and is based on the ANN-SNN conversion scheme proposed by authors in Ref. [85]. However, while prior work considers the ANN operation only during the conversion process, we show that considering the actual SNN operation during the conversion step is crucial for achieving minimal loss in classification accuracy. To that effect, we propose a novel weight-normalization technique that en-

sures that the actual SNN operation is in the loop during the conversion phase. Note that this work tries to exploit neural activation sparsity by converting networks to the spiking domain for power-efficient hardware implementation and are complementary to efforts aimed at exploring sparsity in synaptic connections [179].

The specific contributions of our work [84] are as follows:

(i) As will be explained in later sections, there are various architectural constraints involved for training ANNs that can be converted to SNNs in a near-lossless manner. Hence, it is unclear whether the proposed techniques would scale to larger and deeper architectures for more complicated tasks. *We provide proof of concept experiments that deep SNNs (extending from 16 to 34 layers) can provide competitive accuracies over complex datasets like CIFAR-10 and ImageNet.*

(ii) *We propose a new ANN-SNN conversion technique that statistically outperforms state-of-the-art techniques.* We report a classification error of **8.45%** on the CIFAR-10 dataset which is the best-performing result reported for any SNN network, till date. For the first time we report an SNN performance on the entire ImageNet 2012 validation set. We achieve a **30.04%** top-1 error rate and **10.99%** top-5 error rate for VGG-16 architectures.

(iii) We explore Residual Network (ResNet) architectures as a potential pathway to enable deeper SNNs. We present insights and design constraints that are required to ensure ANN-SNN conversion for ResNets. We report a classification error of **12.54%** on the CIFAR-10 dataset and a **34.53%** top-1 error rate and **13.67%** top-5 error rate on the ImageNet validation set. *This is the first work that attempts to explore SNNs with residual network architectures.*

(iv) *We demonstrate that SNN network sparsity significantly increases as the network depth increases.* This further motivates the exploration of converting ANNs to SNNs for event-driven operation to reduce compute overhead.

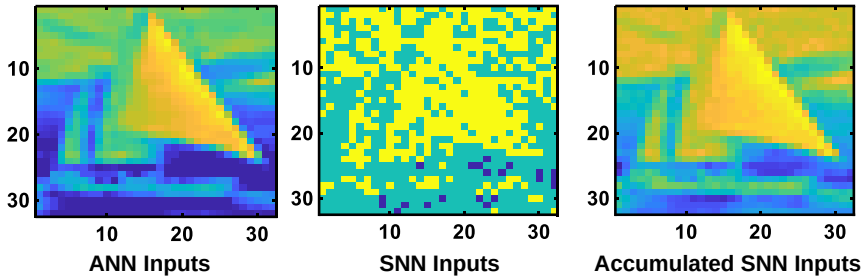

Fig. A.1. The extreme left panel depicts a particular input image from the CIFAR-10 dataset with per pixel mean subtracted that is provided as input to the original ANN. The middle panel represents a particular instance of the Poisson spike train generated from the analog input image. The accumulated events provided to the SNN over 1000 timesteps is depicted in the extreme right panel. This justifies the fact that the input image is being rate encoded over time for SNN operation.

A.2 Preliminaries

The main difference between ANN and SNN operation is the notion of time. While ANN inputs are static, SNNs operate based on dynamic binary spiking inputs as a function of time. The neural nodes also receive and transmit binary spike input signals in SNNs, unlike in ANNs, where the inputs and outputs of the neural nodes are analog values. In this work, we consider a rate-encoded network operation where the average number of spikes transmitted as input to the network over a large enough time window is approximately proportional to the magnitude of the original ANN inputs (pixel intensity in this case). The duration of the time window is dictated by the desired network performance (for instance, classification accuracy) at the output layer of the network. A Poisson event-generation process is used to produce the input spike train to the network. Every time-step of SNN operation is associated with the generation of a random number whose value is compared against the magnitude of the corresponding input. A spike event is triggered if the generated random number is less than the value of the corresponding pixel intensity. This process ensures that the average number of input spikes in the SNN is proportional to the magnitude of the

corresponding ANN inputs and is typically used to simulate an SNN for recognition tasks based on datasets for static images [85]. Fig. A.1 depicts a particular timed-snapshot of the input spikes transmitted to the SNN for a particular image from the CIFAR-10 dataset. SNN operation of such networks are pseudo-simultaneous, i.e. a particular layer operates immediately on the incoming spikes from the previous layer and does not have to wait for multiple time-steps for information from the previous layer neurons to get accumulated. Given a Poisson-generated spike train being fed to the network, spikes will be produced at the network outputs. Inference is based on the cumulative spike count of neurons at the output layer of the network over a given time-window.

ANN to SNN conversion schemes usually consider Rectified Linear Unit (ReLU) as the ANN neuron activation function. For a neuron receiving inputs x_i through synaptic weights w_i, the ReLU neuron output y is given by,

$$y = max \quad 0, \sum_i \left(w_i . x_i \right) \tag{A.1}$$

Although ReLU neurons are typically used in a large number of machine learning tasks at present, the main reason behind their usage for ANN-SNN conversion schemes is that they bear functional equivalence to an Integrate-Fire (IF) Spiking Neuron without any leak and refractory period [85, 86]. Note that this is a particular type of Spiking Neuron model [180]. Let us consider the ANN inputs x_i encoded in time as a spike train $\mathbb{X}_i(t)$, where $\mathbb{E}[\mathbb{X}_i(t)] \propto x_i$ (for the rate encoding network being considered in this work). The IF Spiking Neuron keeps track of its membrane potential, v_{mem}, which integrates incoming spikes and generates an output spike whenever the membrane potential cross a particular threshold v_{th}. The membrane potential is reset to zero at the generation of an output spike. All neurons are reset whenever a spike train corresponding to a new image/pattern in presented. The IF Spiking Neuron dynamics as a function of time-step, t, can be described by the following equation,

$$v_{mem}(t+1) = v_{mem}(t) + \sum_i \left(w_i . \mathbb{X}_i(t) \right. \tag{A.2}$$

Let us first consider the simple case of a neuron being driven by a single input $\mathbb{X}(t)$ and a positive synaptic weight w. Due to the absence of any leak term in the neural dynamics, it is intuitive to show that the corresponding output spiking rate of the neuron is given by $\mathbb{E}[\mathbb{Y}(t)] \propto \mathbb{E}[\mathbb{X}(t)]$, with the proportionality factor being dependent on the ratio of w and v_{th}. In the case when the synaptic weight is negative, the output spiking activity of the IF neuron is zero since the neuron is never able to cross the firing potential v_{th}, mirroring the functionality of a ReLU. The higher the ratio of the threshold with respect to the weight, the more time is required for the neuron to spike, thereby reducing the neuron spiking rate, $\mathbb{E}[\mathbb{Y}(t)]$, or equivalently increasing the time-delay for the neuron to generate a spike. A relatively high firing threshold can cause a huge delay for neurons to generate output spikes. For deep architectures, such a delay can quickly accumulate and cause the network to not produce any spiking outputs for relatively long periods of time. On the other hand, a relatively low threshold causes the SNN to lose any ability to distinguish between different magnitudes of the spike inputs being accumulated to the membrane potential (the term $\sum_i w_i.\mathbb{X}_i(t)$ in Eq. A.2) of the Spiking Neuron, causing it to lose evidence during the membrane potential integration process. This, in turn, results in accuracy degradation of the converted network. Hence, an appropriate choice of the ratio of the neuron threshold to the synaptic weights is essential to ensure minimal loss in classification accuracy during the ANN-SNN conversion process [85]. Consequently, most of the research work in this field has been concentrated on outlining appropriate algorithms for threshold-balancing, or equivalently, weight normalizing different layers of a network to achieve near-lossless ANN-SNN conversion.

Typically neural units used for ANN-SNN conversion schemes are trained without any bias term [85]. This is due to the fact that optimization of the bias term in addition to the spiking neuron threshold expands the parameter space exploration, thereby causing the ANN-SNN conversion process to be more difficult. Requirement of bias less neural units also entails that Batch Normalization technique [181] cannot be used as a regularizer during the training process since it biases the inputs to

each layer of the network to ensure each layer is provided with inputs having zero mean. Instead, we use dropout [182] as the regularization technique. This technique simply masks portions of the input to each layer by utilizing samples from a Bernoulli distribution where each input to the layer has a specified probability of being dropped.

Deep convolutional neural network architectures typically consist of intermediate pooling layers to reduce the size of the convolution output maps. While various choices exist for performing the pooling mechanism, the two popular choices are either max-pooling (maximum neuron output over the pooling window) or spatial-averaging (two-dimensional average pooling operation over the pooling window). Since the neuron activations are binary in SNNs instead of analog values, performing max-pooling would result in significant information loss for the next layer. Consequently, we consider spatial-averaging as the pooling mechanism in this work [85].

A.3 Deep Convolutional SNN Architectures: VGG

As mentioned previously, our work is based on the proposal outlined by authors in Ref. [85]. In order to ensure that a spiking neuron threshold is sufficiently high to distinguish different magnitude of the spike inputs, a worst case solution would be to set the threshold of a particular layer to the maximum of the summation of all the positive synaptic weights of neurons in that layer. However, such a "Model-Based Normalization" technique is highly pessimistic since all the fan-in neurons are not supposed to fire at every time-step [85]. In order to circumvent this issue, authors in Ref. [85] proposed a "Data-Based Normalization" Technique wherein the neuron threshold of a particular layer is set equal to the maximum activation of all ReLUs in the corresponding layer (by passing the entire training set through the trained ANN once after training is completed). Such a "Data-Based" technique performed significantly better than the "Model-Based" algorithm in terms of the final classification accuracy and latency of the converted SNN (three-layered fully connected and convolutional architectures) for a digit recognition problem on the

MNIST dataset [85]. Note that, this process is referred to as "weight-normalization" and "threshold-balancing" interchangeably in this text. As mentioned before, the goal of this work is to optimize the ratio of the synaptic weights with respect to the neuron firing threshold, v_{th}. Hence, either all the synaptic weights preceding a neural layer are scaled by a normalization factor w_{norm} equal to the maximum neural activation and the threshold is set equal to 1 ("weight-normalization"), or the threshold v_{th} is set equal to the maximum neuron activation for the corresponding layer with the synaptic weights remaining unchanged ("threshold-balancing"). Both operations are exactly equivalent mathematically.

However, the above algorithm leads us to the question: *Are ANN activations representative of SNN activations?* Let us consider a particular example for the case of maximum activation for a single ReLU. The neuron receives two inputs, namely 0.5 and 1. Let us consider unity synaptic weights in this scenario. Since the maximum ReLU activation is 1.5, the neuron threshold would be set equal to 1.5. However, when this network is converted to the SNN mode, both the inputs would be propagating binary spike signals. The ANN input, equal to 1, would be converted to spikes transmitting at every time-step while the other input would transmit spikes approximately 50% of the duration of a large enough time-window. Hence, the actual summation of spike inputs received by the neuron per time-step would be 2 for a large number of samples, which is higher than the spiking threshold (1.5). Clearly, some information loss would take place due to the lack of this evidence integration.

Driven by this observation, we propose a weight-normalization technique that adaptively balances the threshold of each layer by considering the actual operation of the SNN in the loop during the ANN-SNN conversion process. The algorithm normalizes the weights of the network sequentially for each layer. Given a particular trained ANN, the first step is to generate the input Poisson spike train for the network over the training set for a large enough time-window. The Poisson spike train allows us to record the maximum summation of weighted spike-input (the term $\sum_i w_i.\mathbb{X}_i(t)$ in Eq. A.2 and hereafter referred to maximum SNN activation in this text) that would

be received by the first neural layer of the network. In order to minimize the temporal delay of the neuron and simultaneously ensure that the neuron firing threshold is not too low, we weight-normalize the first layer depending on the maximum spike-based input received by the first layer. After the threshold of the first layer is set, we are provided with a representative spike train at the output of the first layer which enables us to generate the input spike-stream for the next layer. The process is continued sequentially for all the layers of the network. The main difference between our proposal and prior work [85] is the fact that the proposed weight-normalization scheme accounts for the actual SNN operation during the conversion process. As we will show in the Results section, this scheme is crucial to ensure near-lossless ANN-SNN conversion for significantly deep architectures and for complex recognition problems. The pseudo-code of the algorithm is given in the next page.

A.4 Extension to Residual Architectures

Residual network architectures were proposed as an attempt to scale convolutional neural networks to very deep layered stacks [183]. Although different variants of the basic functional unit have been explored, we will only consider identity shortcut connections in this text (shortcut type-A according to the paper [183]). Each unit consists of two parallel paths. The non-identity path consists of two spatial convolution layers with an intermediate ReLU layer. While the original ResNet formulation considers ReLUs at the junction of the parallel non-identity and identity paths [183], recent formulations do not consider junction ReLUs in the network architecture [184]. Absence of ReLUs at the junction point of the non-identity and identity paths was observed to produce a slight improvement in classification accuracy on the CIFAR-10 dataset [185]. Due to the presence of the shortcut connections, important design considerations need to be accounted for to ensure near-lossless ANN-SNN conversion. We start with the basic unit, as shown in Fig. A.2(a), and point-wise impose various architectural constraints with justifications. Note the discussion in this section

input : Input Poisson Spike Train $spikes$, Number of Time-Steps $\#timesteps$

output: Weight-normalization / Threshold-balancing factors $v_{th,norm}[i]$ for each neural layer ($net.layer[i]$) of the network net

initialization $v_{th,norm}[i] = 0 \; \forall \; i = 1, ..., \#net.layer$;

`// Set input of 1st layer equal to spike train`

$net.layer[1].input = spikes$;

for $i \leftarrow 1$ **to** $\#net.layer$ **do**

  for $t \leftarrow 1$ **to** $\#timesteps$ **do**

    `// Forward pass spike-train for neuron layer-i characterized by membrane potential` $net.layer[i].v_{mem}$ `and threshold` $net.layer[i].v_{th}$

    $net.layer[i] : forward(net.layer[i].input)$;

    `// Determine Threshold-balancing factor according to maximum SNN activation,` $net.layer[i].v_{mem}.input$

    $v_{th,norm}[i] = \max(v_{th,norm}[i], \max(net.layer[i].v_{mem}.input))$;

  end

  `// Threshold-balance layer-i`

  $net.layer[i].v_{th} = v_{th,norm}[i]$;

  `// Record input spike-train for next layer`

  $net.layer[i+1].input = net.layer[i] : forward(net.layer[i].input)$;

end

Algorithm 1: Spike-Norm

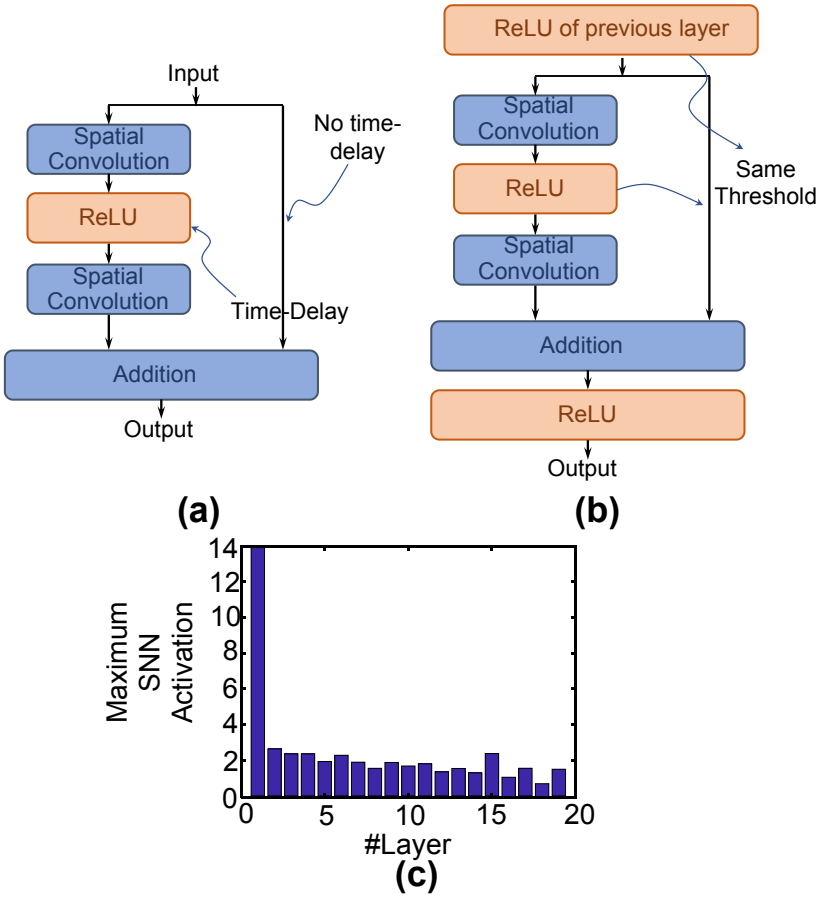

Fig. A.2. (a) The basic ResNet functional unit, (b) Design constraints introduced in the functional unit to ensure near-lossless ANN-SNN conversion, (c) Typical maximum SNN activations for a ResNet having junction ReLU layers but the non-identity and identity input paths not having the same spiking threshold. While this is not representative of the case with equal thresholds in the two paths, it does justify the claim that after a few initial layers, the maximum SNN activations decay to values close to unity due to the identity mapping.

is based on threshold-balancing (with synaptic weights remaining unscaled), i.e. the threshold of the neurons are adjusted to minimize ANN-SNN conversion loss.

As we will show in the Results section, application of our proposed Spike-Norm algorithm on such a residual architecture resulted in a converted SNN that exhibited accuracy degradation in comparison to the original trained ANN. We hypothesize that this degradation is attributed mainly to the absence of any ReLUs at the junction

points. Each ReLU when converted to an IF Spiking Neuron imposes a particular amount of characteristic temporal delay (time interval between an incoming spike and the outgoing spike due to evidence integration). Due to the shortcut connections, spike information from the initial layers gets instantaneously propagated to later layers. The unbalanced temporal delay in the two parallel paths of the network can result in distortion of the spike information being propagated through the network. Consequently, as shown in Fig. A.2(b), we include ReLUs at each junction point to provide a temporal balancing effect to the parallel paths (when converted to IF Spiking Neurons). An ideal solution would be to include a ReLU in the parallel path, but that would destroy the advantage of the identity mapping.

As shown in the next section, direct application of our proposed threshold-balancing scheme still resulted in some amount of accuracy loss in comparison to the baseline ANN accuracy. However, note that the junction neuron layer receives inputs from the previous junction neuron layer as well as the non-identity neuron path. Since the output spiking activity of a particular neuron is also dependent on the threshold-balancing factor, all the fan-in neuron layers should be threshold-balanced by the same amount to ensure that input spike information to the next layer is rate-encoded appropriately. However, the spiking threshold of the neuron layer in the non-identity path is dependent on the activity of the neuron layer at the previous junction. An observation of the typical threshold-balancing factors for the network without using this constraint (shown in Fig. A.2(c)) reveal that the threshold-balancing factors mostly lie around unity after a few initial layers. This occurs mainly due to the identity mapping. The maximum summation of spike inputs received by the neurons in the junction layers are dominated by the identity mapping (close to unity). From this observation, we heuristically choose both the thresholds of the non-identity ReLU layer and the identity-ReLU layer equal to 1. However, the accuracy is still unable to approach the baseline ANN accuracy, which leads us to the third design constraint.

An observation of Fig. A.2(c) reveals that the threshold-balancing factors of the initial junction neuron layers are significantly higher than unity. This can be a primary

reason for the degradation in classification accuracy of the converted SNN. We note that the residual architectures used by authors in Ref. [183] use an initial convolution layer with a very wide receptive field (7×7 with a stride of 2) on the ImageNet dataset. The main motive behind such an architecture was to show the impact of increasing depth in their residual architectures on the classification accuracy. Inspired by the VGG-architecture, we replace the first 7×7 convolutional layer by a series of three 3×3 convolutions where the first two layers do not exhibit any shortcut connections. Addition of such initial non-residual pre-processing layers allows us to apply our proposed threshold-balancing scheme in the initial layers while using a unity threshold-balancing factor for the later residual layers. As shown in the Results section, this scheme significantly assists in achieving classification accuracies close to the baseline ANN accuracy since after the initial layers, the maximum neuron activations decay to values close to unity because of the identity mapping.

A.5 Experiments

We evaluate our proposals on standard visual object recognition benchmarks, namely the CIFAR-10 and ImageNet datasets. Experiments performed on networks for the CIFAR-10 dataset are trained on the training set images with per-pixel mean subtracted and evaluated on the testing set. We also present results on the much more complex ImageNet 2012 dataset that contains 1.28 million training images and report evaluation (top-1 and top-5 error rates) on the $50,000$ validation set. 224×224 crops from the input images are used for this experiment.

We use VGG-16 architecture [186] for both the datasets. ResNet-20 configuration outlined in Ref. [183] is used for the CIFAR-10 dataset while ResNet-34 is used for experiments on the ImageNet dataset. As mentioned previously, we do not utilize any batch-normalization layers. For VGG networks, a dropout layer is used after every ReLU layer except for those layers which are followed by a pooling layer. For Residual

networks, we use dropout only for the ReLUs at the non-identity parallel paths but not at the junction layers. We found this crucial for achieving training convergence.

Our implementation is derived from the Facebook ResNet implementation code for CIFAR and ImageNet datasets available publicly [187]. We use similar image preprocessing steps and scale and aspect-ratio augmentation techniques as used in [188]. We report single-crop testing results while the error rates can be further reduced with 10-crop testing [189]. Networks used for the CIFAR-10 dataset are trained on 2 GPUs with a batchsize of 256 for 200 epochs, while ImageNet training is performed on 8 GPUs for 100 epochs with a similar batchsize. The initial learning rate is 0.05. The learning rate is divided by 10 twice, at 81 and 122 epochs for CIFAR-10 dataset and at 30 and 60 epochs for ImageNet dataset. A weight decay of 0.0001 and a momentum of 0.9 is used for all the experiments. Proper weight initialization is crucial to achieve convergence in such deep networks without batch-normalization. For a non-residual convolutional layer (for both VGG and ResNet architectures) having kernel size $k \times k$ with n output channels, the weights are initialized from a normal distribution and standard deviation $\sqrt{\frac{2}{k^2 n}}$. However, for residual convolutional layers, the standard deviation used for the normal distribution was $\frac{\sqrt{2}}{k^2 n}$. We observed this to be important for achieving training convergence and a similar observation was also outlined in Ref. [190] although their networks were trained without both dropout and batch-normalization.

A.5.1 Experiments for VGG Architectures

Our VGG-16 model architecture follows the implementation outlined in [191] except that we do not utilize the batch-normalization layers. We used a randomly chosen mini-batch of size 256 from the training set for the weight-normalization process on the CIFAR-10 dataset. While the entire training set can be used for the weight-normalization process, using a representative subset did not impact the results. We confirmed this by running multiple independent runs for both the CIFAR and Ima-

geNet datasets. The standard deviation of the final classification error rate after 2500 time-steps was ~ 0.01. All results reported in this section represent the average of 5 independent runs of the spiking network (since the input to the network is a random process). No notable difference in the classification error rate was observed at the end of 2500 time-steps and the network outputs converged to deterministic values despite being driven by stochastic inputs. For the SNN model based weight-normalization scheme (Spike-Norm algorithm) we used 2500 time-steps for each layer sequentially to normalize the weights.

Table A.1 summarizes our results for the CIFAR-10 dataset. The baseline ANN error rate on the testing set was 8.3%. Since the main contribution of this work is to minimize the loss in accuracy during conversion from ANN to SNN for deep-layered networks and not in pushing state-of-the-art results in ANN training, we did not perform any hyper-parameter optimization. However, note that despite several architectural constraints being present in our ANN architecture, we are able to train deep networks that provide competitive classification accuracies using the training mechanisms described in the previous subsection. Further reduction in the baseline ANN error rate is possible by appropriately tuning the learning parameters. For the VGG-16 architecture, our implementation of the ANN-model based weight-normalization technique, proposed by Ref. [85], yielded an average SNN error rate of 8.54% leading to an error increment of 0.24%. The error increment was minimized to 0.15% on applying our proposed Spike-Norm algorithm. Note that we consider a strict model-based weight-normalization scheme to isolate the impact of considering the effect of an ANN versus our SNN model for threshold-balancing. Further optimizations of considering the maximum synaptic weight during the weight-normalization process [85] is still possible.

Previous works have mainly focused on much shallower convolutional neural network architectures. Although Ref. [192] reports results with an accuracy loss of 0.18%, their baseline ANN suffers from some amount of accuracy degradation since their networks are trained with noise (in addition to architectural constraints men-

tioned before) to account for neuronal response variability due to incoming spike trains [192]. It is also unclear whether the training mechanism with noise would scale up to deeper layered networks. *Our work reports the best performance of a Spiking Neural Network on the CIFAR-10 dataset till date.*

The impact of our proposed algorithm is much more apparent on the more complex ImageNet dataset. The rates for the top-1 (top-5) error on the ImageNet validation set are summarized in Table A.2. Note that these are single-crop results. The accuracy loss during the ANN-SNN conversion process is minimized by a margin of 0.57% by considering SNN-model based weight-normalization scheme. It is therefore expected that our proposed Spike-Norm algorithm would significantly perform better than an ANN-model based conversion scheme as the pattern recognition problem becomes more complex since it accounts for the actual SNN operation during the conversion process. Note that Ref. [192] reports a performance of 48.2%(23.8%) on the first 3072-image test batch of the ImageNet 2012 dataset.

At the time we developed this work, we were unaware of a parallel effort to scale up the performance of SNNs to deeper networks and large-scale machine learning tasks. The work was recently published in Ref. [193]. However, their work differs from our approach in the following aspects:

(i) Their work improves on prior approach outlined in Ref. [85] by proposing conversion methods for removing the constraints involved in ANN training (discussed in Section A.2). We are improving on prior art by scaling up the methodology outlined in Ref. [85] for ANN-SNN conversion by including the constraints.

(ii) We are demonstrating that considering SNN operation in the conversion process helps to minimize the conversion loss. Ref. [193] uses ANN based normalization scheme used in Ref. [85].

While removing the constraints in ANN training allows authors in Ref. [193] to train ANNs with better accuracy, they suffer significant accuracy loss in the conversion process. This occurs due to a non-optimal ratio of biases/batch-normalization factors and weights [193]. This is the primary reason for our exploration of ANN-SNN

conversion without bias and batch-normalization. For instance, their best performing network on CIFAR-10 dataset incurs a conversion loss of 1.06% in contrast to 0.15% reported by our proposal for a much deeper network. The accuracy loss is much larger for their VGG-16 network on the ImageNet dataset - 14.28% in contrast to 0.56% for our proposal. Although Ref. [193] reports a top-1 SNN error rate 25.40% for a Inception-V3 network, their ANN is trained with an error rate of 23.88%. The resulting conversion loss is 1.52% and much higher than our proposals. The Inception-V3 network conversion was also optimized by a voltage clamping method, that was found to be specific for the Inception network and did not apply to the VGG network [193]. Note that the results reported on ImageNet in Ref. [193] are on a subset of image samples. Hence, the performance on the entire dataset is unclear. Our contribution lies in the fact that we are demonstrating ANNs can be trained with the above-mentioned constraints with competitive accuracies on large-scale tasks and converted to SNNs in a near-lossless manner.

This is the first work that reports competitive performance of a Spiking Neural Network on the entire $50,000$ *ImageNet 2012 validation set.*

A.5.2 Experiments for Residual Architectures

Our residual networks for CIFAR-10 and ImageNet datasets follow the implementation in Ref. [183]. We first attempt to explain our design choices for ResNets by sequentially imposing each constraint on the network and showing their corresponding impact on network performance in Fig. A.3. The "Basic Architecture" involves a residual network without any junction ReLUs. "Constraint 1" involves junction ReLUs without having equal spiking thresholds for all fan-in neural layers. "Constraint 2" imposes an equal threshold of unity for all the layers while "Constraint 3" performs best with two pre-processing plain convolutional layers (3×3) at the beginning of the network. The baseline ANN ResNet-20 was trained with an error of 10.9% on the CIFAR-10 dataset. Note that although we are using terminology con-

Table A.1.
Results for CIFAR-10 Dataset

Network Architecture	**ANN Error**	**SNN Error**	**Error Increment**
4-layered networks [86] (Input cropped to 24 x 24)	20.88%	22.57%	1.69%
3-layered networks [194]	–	10.68%	–
8-layered networks [192] (Input cropped to 24 x 24)	16.28%	16.46%	0.18%
6-layered networks [193]	8.09%	9.15%	1.06%
VGG-16 (ANN model based conversion)	8.3%	8.54%	0.24%
VGG-16 (SPIKE-NORM)	**8.3%**	**8.45%**	**0.15%**

sistent with Ref. [183] for the network architectures, our ResNets contain two extra plain pre-processing layers. The converted SNN according to our proposal yielded a classification error rate of 12.54%. Weight-normalizing the initial two layers using the ANN-model based weight-normalization scheme produced an average error of 12.87%, further validating the efficiency of our weight-normalization technique.

Table A.2.
Results for ImageNet Dataset

Network Architecture	**ANN Error**	**SNN Error**	**Error Increment**
8-layered networks [192] (Tested on subset of 3072 images)	–	48.20% (23.80%)	–
VGG-16 [193] (Tested on subset of 2570 images)	36.11% (15.14%)	50.39% (18.37%)	14.28% (3.23%)
VGG-16 (ANN model based conversion)	29.48% (10.61%)	30.61% (11.21%)	1.13% (0.6%)
VGG-16 (SPIKE-NORM)	**29.48% (10.61%)**	**30.04% (10.99%)**	**0.56% (0.38%)**

On the ImageNet dataset, we use the deeper ResNet-34 model outlined in Ref. [183]. The initial 7×7 convolutional layer is replaced by three 3×3 convolutional layers where the initial two layers are non-residual plain units. The baseline ANN is trained with an error of 29.31% while the converted SNN error is 34.53% at the end of 2500 timesteps. The results are summarized in Table. A.3 and convergence plots for all our networks are provided in Fig. A.4.

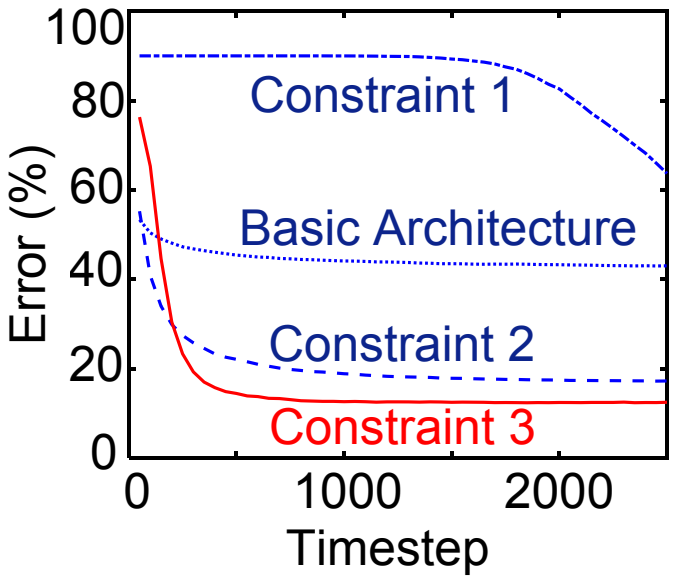

Fig. A.3. Impact of the architectural constraints for Residual Networks. "Basic Architecture" does not involve any junction ReLU layers. "Constraint 1" involves junction ReLUs while "Constraint 2" imposes equal unity threshold for all residual units. Network accuracy is significantly improved with the inclusion of "Constraint 3" that involves pre-processing weight-normalized plain convolutional layers at the network input stage.

It is worth noting here that the main motivation of exploring Residual Networks is to go deeper in Spiking Neural Networks. We explore relatively simple ResNet architectures, as the ones used in Ref. [183], which have an order of magnitude lower parameters than standard VGG-architectures. Further hyper-parameter optimizations or more complex architectures are still possible. While the accuracy loss in the ANN-SNN conversion process is more for ResNets than plain convolutional architectures, yet further optimizations like including more pre-processing initial layers or better threshold-balancing schemes for the residual units can still be explored. *This work serves as the first work to explore ANN-SNN conversion schemes for Residual Networks and attempts to highlight important design constraints required for minimal loss in the conversion process.*

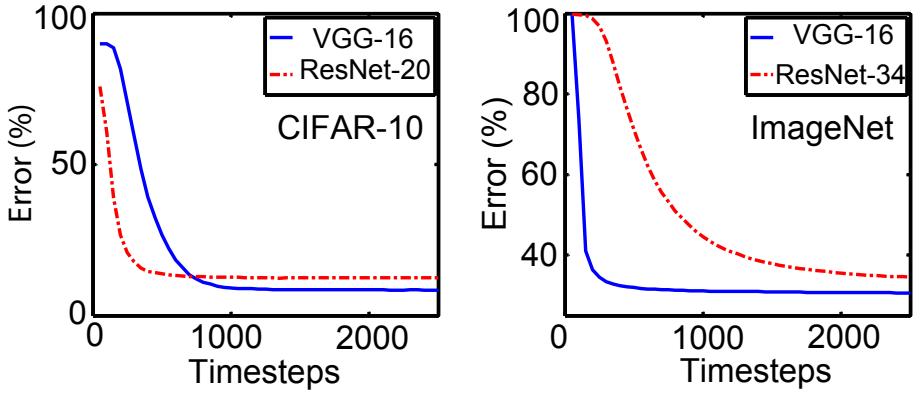

Fig. A.4. Convergence plots for the VGG and ResNet SNN architectures for CIFAR-10 and ImageNet datasets are shown above. The classification error reduces as more evidence is integrated in the Spiking Neurons with increasing time-steps. Note that although the network depths are similar for CIFAR-10 dataset, the ResNet-20 converges much faster than the VGG architecture. The delay for inferencing is higher for ResNet-34 on the ImageNet dataset due to twice the number of layers as the VGG network.

A.5.3 Computation Reduction Due to Sparse Neural Events

ANN operation for prediction of the output class of a particular input requires a single feed-forward pass per image. For SNN operation, the network has to be evaluated over a number of time-steps. However, specialized hardware that accounts for the event-driven neural operation and "computes only when required" can potentially exploit such alternative mechanisms of network operation. For instance, Fig. A.5 represents the average total number of output spikes produced by neurons in VGG and ResNet architectures as a function of the layer for ImageNet dataset. A randomly chosen minibatch was used for the averaging process. We used 500 timesteps for accumulating the spike-counts for VGG networks while 2000 time-steps were used for ResNet architectures. This is in accordance to the convergence plots shown in Fig. A.4. An important insight obtained from Fig. A.5 is the fact that neuron spiking activity becomes sparser as the network depth increases. Hence, benefits from event-driven hardware is expected to increase as the network depth increases. While an

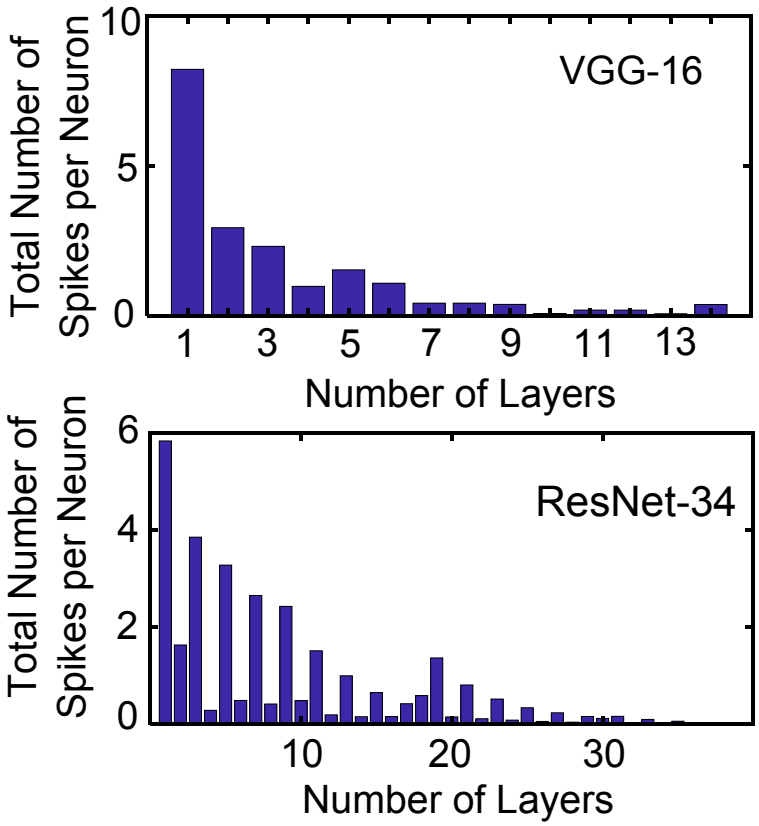

Fig. A.5. Average cumulative spike count generated by neurons in VGG and ResNet architectures on the ImageNet dataset as a function of the layer number. 500 timesteps were used for accumulating the spike-counts for VGG networks while 2000 time-steps were used for ResNet architectures. The neural spiking sparsity increases significantly as network depth increases.

estimate of the actual energy consumption reduction for SNN mode of operation is outside the scope of this current work, we provide an intuitive insight by providing the number of computations per synaptic operation being performed in the ANN versus the SNN.

The number of synaptic operations per layer of the network can be easily estimated for an ANN from the architecture for the convolutional and linear layers. For the ANN, a multiply-accumulate (MAC) computation takes place per synaptic operation. On the other hand, a specialized SNN hardware would perform an accumulate

Table A.3.
Results for Residual Networks

Dataset	**Network Architecture**	**ANN Error**	**SNN Error**
CIFAR-10	ResNet-20	10.9%	12.54%
ImageNet	ResNet-34	29.31% (10.31%)	34.53% (13.67%)

computation (AC) per synaptic operation only upon the receipt of an incoming spike. Hence, the total number of AC operations occurring in the SNN would be represented by the layerwise product and summation of the average cumulative neural spike count for a particular layer and the corresponding number of synaptic operations. Calculation of this metric reveal that for the VGG network, the ratio of SNN AC operations to ANN MAC operations is 1.975 while the ratio is 2.4 for the ResNet (the metric includes only ReLU/IF spiking neuron activations in the network). However, note the fact that a MAC operation involves an order of magnitude more energy consumption than an AC operation. For instance, Ref. [195] reports that the energy consumption in a 32-bit floating point MAC operation is $3.2pJ$ while the energy consumption is only $0.1pJ$ for an AC operation in 45nm technology. Hence, the energy consumption reduction for our SNN implementation is expected to be 16.2× for the VGG network and 13.3× for the ResNet in comparison to the original ANN implementation.

A.6 Conclusions and Future Work

This work serves to provide inspiration to the fact that SNNs exhibit similar computing power as their ANN counterparts. This can potentially pave the way for the usage of SNNs in large scale visual recognition tasks, which can be enabled by low-power neuromorphic hardware. However, there are still open areas of exploration

for improving SNN performance. A significant contribution to the present success of deep NNs is attributed to Batch-Normalization [181]. While using bias less neural units constrain us to train networks without Batch-Normalization, algorithmic techniques to implement Spiking Neurons with a bias term should be explored. Further, it is desirable to train ANNs and convert to SNNs without any accuracy loss. Although the proposed conversion technique attempts to minimize the conversion loss to a large extent, yet other variants of neural functionalities apart from ReLU-IF Spiking Neurons could be potentially explored to further reduce this gap. Additionally, further optimizations to minimize the accuracy loss in ANN-SNN conversion for ResNet architectures should be explored to scale SNN performance to even deeper architectures.

B. STOCHASTICITY OF SPINTRONIC DEVICES AS A FUNCTION OF TEMPERATURE: ON-CHIP TEMPERATURE SENSOR IMPLEMENTATION

This thesis has explored various neuromorphic computing paradigms that can be enabled by the stochastic switching of nanomagnets at non-zero temperatures. All these computing platforms are based on the stochastic switching response of the magnet as a function of the input current magnitude at a constant operating temperature. In this appendix, we explore an alternative approach for abstracting the stochastic switching response of the magnet as a function of temperature at fixed external current input and demonstrate its possible usage for on-chip temperature sensor applications.

B.1 Introduction

Due to continued device scaling and consequent addition of more components on-chip, which in-turn results in enhanced heat generation, chip temperature monitoring has become a critical issue for ensuring reliable operation. With advanced technology nodes, increased throughput is achieved at the expense of more heat generation. Hence, designing on-chip low-power, low-cost temperature sensors is becoming a crucial requirement [196–199]. The typical performance metrics for on-chip temperature sensors are the conversion rate and energy consumption per inference. The conversion rate is defined as the number of inference samples that can be produced by the sensor per unit sec which is the inverse of the time required by the sensor to make an inference. The energy consumption per inference is defined as the product of the power consumption of the sensor and the inverse of the conversion rate.

While most of the recent work in the domain of on-chip temperature sensors have been primarily based on CMOS sensors [196–199], it is interesting to note that

post-CMOS technologies like spintronic devices demonstrate temperature-dependent probabilistic switching due to thermal noise. Although, traditionally the stochastic switching behavior of spin-based devices have been primarily viewed as a disadvantage for on-chip memory applications, recently unconventional computing paradigms like neuromorphic computing [6, 7, 127], Ising computing [133, 135] and Bayesian inference networks [134] based on stochastic nanomagnets have been proposed that leverage the underlying stochastic device physics. The probabilistic switching of the spintronic device is a function of the input programming current and the operating temperature (assuming a fixed duration of the programming current). However, all these applications abstract the probabilistic switching characteristics of the spintronic device as a function of input current as the external stimulus, at a fixed temperature. This appendix section attempts to explore the stochastic magnet dynamics as a function of temperature and provides an estimation of its performance metrics as an on-chip temperature sensor in comparison to state-of-the-art CMOS based sensors. The potential advantages of such nanomagnetic temperature sensors are compactness, higher conversion rate and lower energy consumption per inference.

B.2 MTJ as Temperature-Biased Random Number Generator

The operation of the MTJ as a temperature-biased random number generator has been explained in Fig. B.1. A particular temperature inference takes place over a number of "write"-"read"-"reset" cycles. The timing waveform for a particular cycle has been shown in the figure. During the "write" cycle, the MTJ is driven by a current source which passes an input charge current through the heavy metal underlayer. Depending on the operating temperature, the MTJ switches with a given probability. Consecutively, during the "read" phase, the MTJ state is determined using the resistive divider circuit shown in Fig. B.1. The reference resistor, R_{REF}, is an MTJ whose state is fixed in the AP state. The read current is maintained to sufficiently low values such that the MTJ states are not disturbed. Note that the "write" and

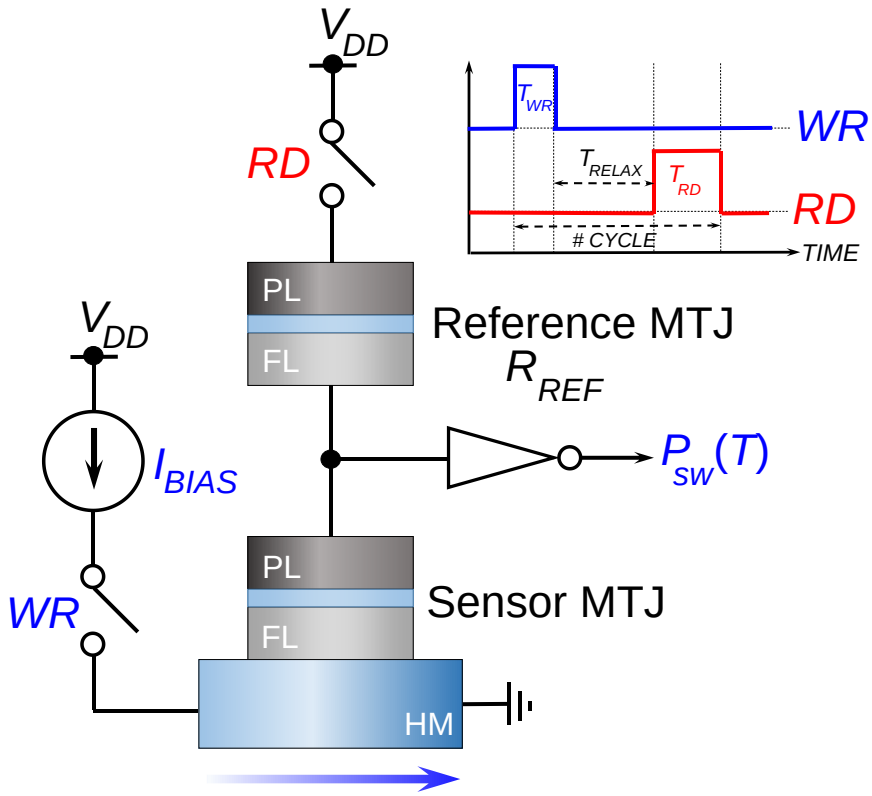

Fig. B.1. The Sensor MTJ is interfaced with a Reference MTJ (R_{REF}) to form a voltage divider circuit (driven by supply voltage V_{DD}) that drives an inverter at the output to determine the switching probability (P_{SW}) at an operating temperature T. WR and RD are control signals that activate the "write" and "read" current paths of the MTJ respectively. During the "write" phase (WR activated), a bias current (I_{BIAS}) probabilistically switches the magnet depending on the temperature. After a subsequent "relaxation" phase, T_{RELAX}, the "read" phase (RD activated) is used to determine the final state of the MTJ due to the corresponding "write" phase.

"read" phases are separated by a "relaxation" period, T_{RELAX}, in order to stabilize the magnetization directions to either of the two stable states after the "write" phase. The magnet is "reset" to the initial AP state for the next cycle in case a switching event takes place by passing a large enough magnitude of current through the heavy metal in the opposite direction to ensure approximately deterministic switching. The

switching probability is determined from multiple such measurement cycles and the operating temperature is determined from the measured switching probability.

The device parameters have been mentioned in Table. 4.1. The parameters are based on experimental measurements reported in Ref. [118]. The "Write", "Relaxation" and "Read" phase durations are $0.5ns$, $2ns$ and $1ns$ respectively. The design temperature is varied in the range $200 - 400K$.

B.3 Sensor Performance Metrics

Fig. B.2(a) represents the switching probability characteristics of the MTJ (as a function of "write" current through the HM) with varying temperature. The dispersion in switching probability characteristics between $200K$ and $400K$ is maximized at the central region of the switching probability characteristics (Fig. B.2(b)). Specifically, we note that for our design pulse width duration of $0.5ns$, the optimal design current is $\sim 70\mu A$ and the probability dispersion (absolute difference in the MTJ switching probabilities at $200K$ and $400K$) is $\sim 24\%$.

Fig. B.3 denotes the MTJ switching probability at the optimal bias current of $70\mu A$ as a function of temperature. Although the switching characteristic becomes non-linear and tends to saturate at very high temperatures, the characteristic is approximately linear in the range of $200K - 400K$. The resolution of the sensor linearity is $\sim 0.37\%/1°C$.

A single switching event of the MTJ can be considered to be a Poisson process with the probability of switching being determined by the temperature. Consequently, the precision of temperature sensing is expected to increase as the number of switching events ("write"-"read"-"reset" cycles) for the temperature inference process is increased. Fig. B.4 shows that the average sensing error in the range $200K - 400K$ is reduced to $\sim 1°C$ as the number of samples is increased to $100,000$. Considering each cycle to be of duration $4ns$ ($0.5ns$ for "write" phase, $2ns$ for "relaxation" phase, $1ns$ for "read" phase and $0.5ns$ for "reset" phase), the resultant time required for

one inference is $4 \times 10^{-4}s$ (with an error tolerance of $\sim 1°C$). The corresponding conversion rate is 2500 samples/s.

The energy consumption of the MTJ based sensor can be estimated by considering the energy consumed during the "write", "read" and "reset" phases of operation in one cycle. Considering the bias current of $70\mu A$ is provided by a $1V$ supply, the total "write" energy consumption is estimated to be $35fJ$ (VIT_{WR} energy consumption, where $V = 1V$, $I = 70\mu A$ and $T_{WR} = 0.5ns$). Assuming a design temperature sensing range of $200K - 400K$, the device exhibits a switching probability of $P_{RESET} = 46\%$ at the mean temperature of $300K$. Since, the MTJ needs to be reset for every switching event by passing a $140\mu A$ charge current in the opposite direction through the HM layer (to ensure deterministic switching: see Fig. B.2(a)), the "reset" energy consumption is estimated to be $\sim 32fJ$ ($P_{RESET}VIT_{RESET}$ energy consumption where, $V = 1V$, $I = 140\mu A$ and $T_{WR} = 0.5ns$). The "read" energy consumption was estimated by SPICE simulations of the MTJ based voltage divider driving an inverter stage (as shown in Fig. B.1). Non-Equilibrium Green's Function (NEGF) based transport simulation framework was used to model the MTJ resistance [25]. The total "read" energy consumption was estimated to be $\sim 21fJ$ (including the energy consumption of the latch being driven by the inverter stage). Considering the total number of cycles per inference to be $100,000$, the total energy consumption of the MTJ based temperature sensor per conversion is given by the product of the resultant energy consumption per cycle and the number of cycles required per inference, and is equivalent to $\sim 8.8nJ$. Comparison of the MTJ based temperature sensor in terms of conversion rate and energy/conversion with other recent proposals of CMOS based temperature sensors are summarized in Table B.1.

B.4 Scaling to the Super-Paramagnetic Regime

The discussion so far has been based on magnet dimensions exhibiting a barrier height of $\sim 20k_BT$ (at the nominal temperature $T = 300K$). However, as the magnet

Table B.1.
Comparison of MTJ With Other Proposed Temperature Sensors

Sensor Type	**Temperature Range** ($^{\circ}C$)	**Inaccuracy** ($^{\circ}C$)	**Conversion Rate (samples/s)**	**Energy /Conversion** (nJ)	**Technology**
CMOS [196]	$0 - 100$	$-0.8 - +1$	10	150	$0.35\mu m$
CMOS [197]	$0 - 100$	$-0.851 \sim 0.524$	$1.6K$	98.13	$0.18\mu m$
CMOS [198]	$-40 - 85$	± 1	$1K$	66.5	$0.18\mu m$
CMOS [199]	$20 - 50$	± 0.1	10	1600	$0.18\mu m$
MTJ (This Work)	$-73 - 127$	± 1	$2.5K$	8.8	$-$

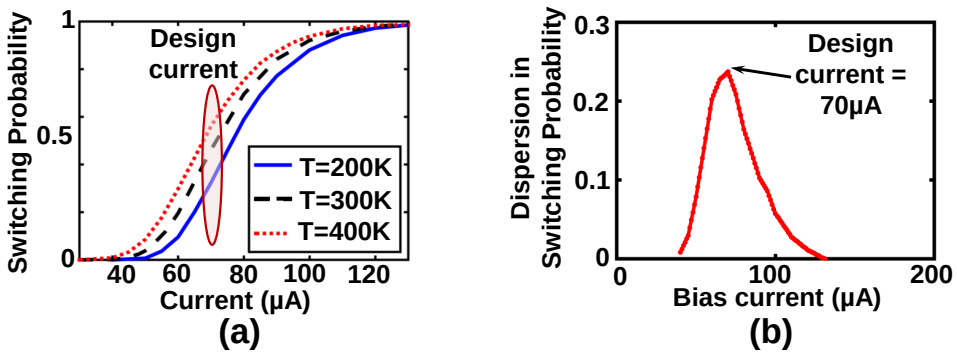

Fig. B.2. (a) MTJ switching probability characteristics with varying temperature in the range $200-400K$, (b) The dispersion in switching probability between $200K$ and $400K$ is maximized for a design bias current $70\mu A$ (central region of the switching probability characteristics).

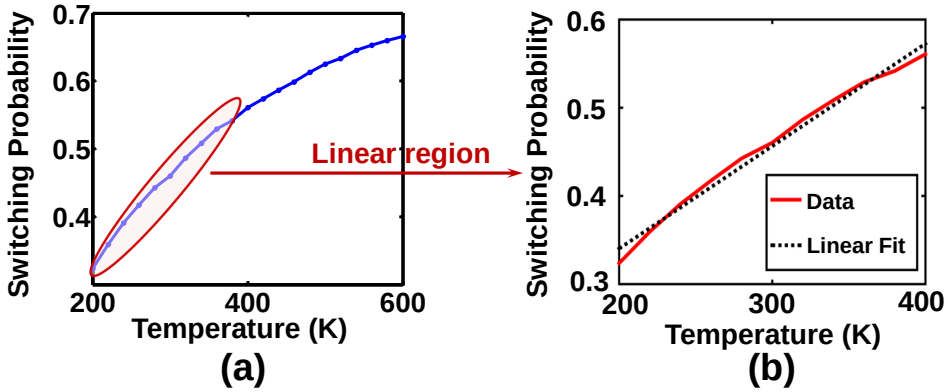

Fig. B.3. The switching probability of the MTJ subjected to a bias current of magnitude $70\mu A$ and duration $0.5ns$ as a function of temperature. Although the characteristics increase non-linearly, it is approximately linear in the design temperature range of $200-400K$.

dimensions are aggressively scaled down to the super-paramagnetic regime ($1k_BT$ barrier height), the magnet exhibits random telegraphic switching between the two extreme states. As discussed before, the average dwell time in each state is $\sim 50\%$, and the average in-plane magnetization over a duration of $500ns$ is approximately zero. The dwell time in either of the two extreme states can be biased by the magnitude of

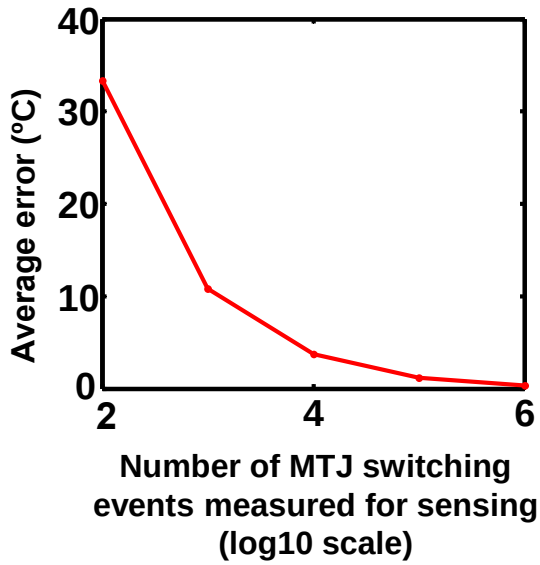

Fig. B.4. Inaccuracy of the MTJ based temperature sensor as a function of the number of switching events ("write"-"read"-"reset" cycles) used for inferring the switching probability and operating temperature. The average error reduces to $\sim 1°C$ as the number of samples is increased to $100,000$.

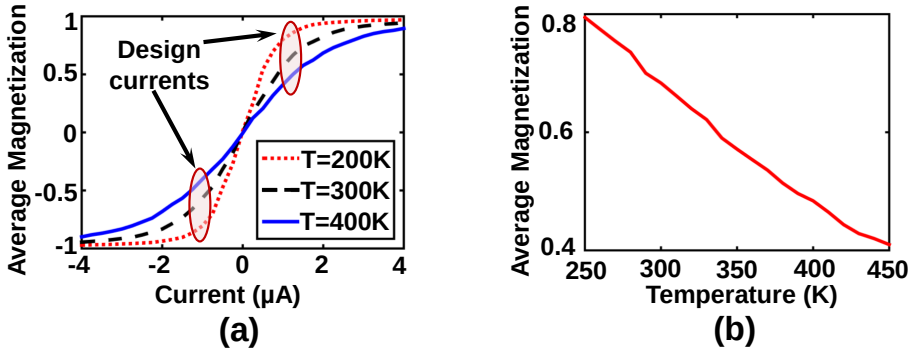

Fig. B.5. (a) Variation of the average in-plane magnetization with magnitude of the "write" current for $T = 200K - 400K$, (b) For a design bias current of $1\mu A$, the average magnetization varies approximately linearly with the operating temperature. The time-window used for the averaging operation is $100,000ns$.

the input current stimulus (flowing through the underlying HM layer) as well as the operating temperature. Fig. B.5(a) represents the average in-plane magnetization

as a function of the "write" current flowing through the HM layer at the nominal temperature $T = 300K$. For a design bias current of $1\mu A$, the MTJ exhibits linear variation of average magnetization profile with sensing temperature (Fig. B.5(b)).

Due to the low barrier height, the magnet essentially operates as a volatile device. Consequently, the circuit peripherals have to be operated in an asynchronous fashion (in contrast to the synchronous "write"-"read"-"reset" mode of operation discussed for high barrier height magnets). As mentioned before, the "write" and "read" current paths have to be activated simultaneously and the "read" circuit has to be optimized to ensure that the "read" current has minimal impact on the switching of the magnet. Circuit-level simulations indicate that the "read" current can be maintained to values below $100nA$, thereby having negligible influence on the switching probability characteristics of the magnet.

The potential benefits of such super-paramagnetic sensors lies in the conversion rate and energy consumption per inference. Since telegraphic switching occurs in the $\sim ps$ time scale, the time window per inference can be greatly reduced. Further, the "write" bias current magnitude is reduced by almost an order of magnitude, thereby reducing the "write" power consumption. Additionally, no "reset" operation is required (due to telegraphic magnet switching), leading to reduction in both the power consumption and the delay involved in the "reset" operation.

B.5 Conclusions

In conclusion, we proposed a compact nanoelectronic temperature sensor that is able to provide a higher throughput and lower energy consumption in comparison to state-of-the-art CMOS temperature sensors. A key point that enables the usage of stochastic switching behavior of MTJs for temperature sensing applications (in comparison to stochastic switching behavior of other resistive memory technologies) is that the causal element for the device stochasticity is thermal noise. Instead of considering the underlying device stochasticity to be disadvantageous, this work can

potentially pave the way for MTJ-enabled on-chip temperature sensors that exploit the probabilistic switching characteristics of nanomagnets at non-zero temperatures.

VITA

VITA

Abhronil Sengupta has been pursuing the PhD degree in the Department of Electrical and Computer Engineering at Purdue University, under the supervision of Prof. Kaushik Roy since Fall 2013. He received the B.E. degree from Jadavpur University, India in 2013. He worked as a DAAD (German Academic Exchange Service) Fellow at the University of Hamburg, Germany in 2012, and as a graduate research intern at Circuit Research Labs, Intel Labs in 2016 and Facebook Reality Labs in 2017. The ultimate goal of Abhronil's research is to bridge the gap between Nanoelectronics and Machine Learning. He is interested in pursuing an inter-disciplinary research agenda at the intersection of hardware and software across the stack of sensors, devices, circuits, systems and algorithms for enabling low-power event-driven cognitive intelligence. Abhronil has published over 45 articles in referred journals and conferences and holds 4 granted/pending US patents. He has been awarded the Bilsland Dissertation Fellowship (2017), CSPIN Student Presenter Award (2015), Birck Fellowship (2013), the DAAD WISE Fellowship (2012), and his publications have featured as APL Editor's Picks (2015) and top 5 popular articles in IEEE TCAS-I (2017). His work on spin-device based neuromorphic computing has been highlighted in media by MIT Technology Review, US Department of Defense, American Institute of Physics among others.