ENERGY EFFICIENT NEUROMORPHIC COMPUTING: CIRCUITS, INTERCONNECTS AND ARCHITECTURE

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Minsuk Koo

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2020

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. Shreyas Sen

School of Electrical and Computer Engineering

Approved by:

Dr. Dimitrios Peroulis

Head of the School Graduate Program

To my family and friends.

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Prof. Kaushik Roy for his encouragement and guidance. My research at Purdue has been guided with his warm-hearted and enthusiastic mentoring. I am truly fortunate to have him as my advisor. I am greatly indebted to him for guiding me not only in research but also in other facets of life at Purdue.

I am also highly thankful to my Ph.D. committee: Prof. Anand Raghunathan, Prof. Vijay Raghunathan and Prof. Shreyas Sen for kindly serving on the advisory committee and for their excellent advice and feedbacks.

I would like to thank all alumni of NRL who paved the way for current NRL members including me. I especially thank Dr. Woo-Shul Cho, Dr. Deliang Fan, Dr. Yusung Kim, Dr. Kon-woo Kwon, Dr. Parami Wijesinghe, Dr. Abhronil Sengupta, Dr. Yeongkyo Seo, Dr. Yong Shim, Dr. Gopalakrishnan Srinivasan, and Dr. Karthik Yogendra for helping me with many aspects.

I also want to thank all current NRL members with whom I had great time during my Ph.D. I especially thank to Amogh Agrawal, Mustafa Ali, Aayush Ankit, Indranil Chakraborty, Mei-Chin Chen, Chankyu Lee, and Chamika Liyanagedera for helpful discussions. I am also would like to thank Rwitti Roy for providing welcoming home, where we hung out playing games, having food, and etc.

Finally and most importantly, I would like to express my greatest thanks to family for their love and support, in particular, my mother Taihee Kim and fater Yunchon Koo who have shown endless love, devoted support and sacrifices. I would also like to recognize my aunt and uncle Yong-Bin and aunt Yong-sun.

This thesis has been possible due to the help and support of numerous people and I want to apologize for not being able to list everyone here.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

ABSTRACT

Koo, Minsuk Ph.D., Purdue University, May 2020. Energy Efficient Neuromorphic Computing: Circuits, Interconnects and Architecture. Major Professor: Roy K. Professor.

Neuromorphic computing has gained tremendous interest because of its ability to overcome the limitations of traditional signal processing algorithms in data intensive applications such as image recognition, video analytics, or language translation. The new computing paradigm is built with the goal of achieving high energy efficiency, comparable to biological systems. To achieve such energy efficiency, there is a need to explore new neuro-mimetic devices, circuits, and architecture, along with new learning algorithms. To that effect, we propose two main approaches:

First, we explore an energy-efficient hardware implementation of a bio-plausible Spiking Neural Network (SNN). The key highlights of our proposed system for SNNs are 1) addressing connectivity issues arising from Network On Chip (NOC)-based SNNs, and 2) proposing stochastic CMOS binary SNNs using biased random number generator (BRNG). On-chip Power Line Communication (PLC) is proposed to address the connectivity issues in NOC-based SNNs. PLC can use the on-chip power lines augmented with low-overhead receiver and transmitter to communicate data between neurons that are spatially far apart. We also propose a CMOS 'stochastic-bit' with on-chip stochastic Spike Timing Dependent Plasticity (sSTDP) based learning for memory-compressed binary SNNs. A chip was fabricated in 90 nm CMOS process to demonstrate memory-efficient reconfigurable on-chip learning using sSTDP training.

Second, we explored coupled oscillatory systems for distance computation and convolution operation. Recent research on nano-oscillators has shown the possibility of using coupled oscillator networks as a core computing primitive for analog/non-

Boolean computations. Spin-torque oscillator (STO) can be an attractive candidate for such oscillators because it is CMOS compatible, highly integratable, scalable, and frequency/phase tunable. Based on these promising features, we propose a new coupled-oscillator based architecture for hybrid spintronic/CMOS hardware that computes multi-dimensional norm. The hybrid system composed of an array of four injection-locked STOs and a CMOS detector is experimentally demonstrated. Energy and scaling analysis shows that the proposed STO-based coupled oscillatory system has higher energy efficiency compared to the CMOS-based system, and an order of magnitude faster computation speed in distance computation for high dimensional input vectors.

1. INTRODUCTION

Even with unprecedented success driven by device scaling, solutions to optimization, recognition, and classification problems based on Von-Neumann architecture turn out to be very inefficient. Moreover, the scaling-down of CMOS technology is approaching its fundamental limit. Hence, researchers are exploring new possibilities from novel devices to non von-Neumann like architectures to achieve performance beyond CMOS scaling as shown in Fig. 1.1.

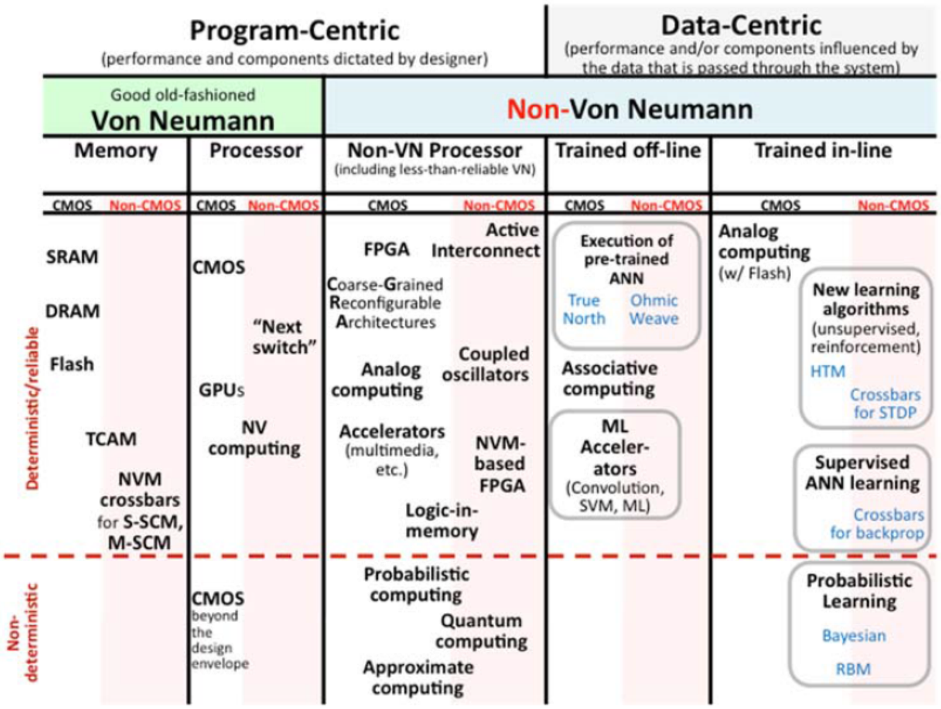


Fig. 1.1. Categories of emerging computing architectures (2016 IRDS) [1]

Neuromorphic computing that attempts to solve the problems in a "preferred way of nature" has acquired tremendous interest because of its ability to overcome the limitations of von-Neumann systems in data intensive applications. The key inspiration behind the development of such neuro-inspired computing systems comes from its high computing and energy efficiency to be comparable to biological systems. Hence, emerging devices that can mimic neurons and synapses, neuromorphic algorithms, and new architectures need to be explored individually and collectively to obtain such high energy efficiency.

Spiking Neural Networks (SNNs) offer a promising solution towards realizing energy-efficient neuromorphic systems. SNNs consider the presence and timing of spikes as the means of communication and neural computation. On account of spike-based event-driven computing capability and localized learning using Spike Timing Dependent Plasticity (STDP), SNNs are regarded as the third generation neural network [2]. However, the proper algorithm and architecture for SNNs still remain to be explored. This motivates us to investigate energy efficiency of the system for SNNs in terms of circuits, interconnects and architecture.

On the other hand, the paradigm of 'let physics do the computing' has also motivated researchers to look at alternative computing models that explore the use of emerging devices as functional units for better energy efficiency and speed. One such alternative model is based on the coupled oscillator network in which the oscillator array is used to compute (say) "similarity" between two multi-dimensional vectors. Such coupled oscillatory networks are widely found in nature such as pendulum clocks on a wall [3], flashing fireflies [4], animal flocking [5], coupled oscillations in the human heart and brain [6,7]. Spin-torque oscillator (STO) is an attractive candidate for such alternative computing models because it is CMOS compatible, highly integratable, scalable, and frequency/phase tunable. Based on these promising features, we are motivated to investigate a new coupled-oscillator based computing architecture that computes multi-dimensional norm.

The thesis also explores how emerging technologies like memristors [8–10] and memristive cross-bars (to do efficient dot-products, a core computing primitive for neuromorphic computing), and deeply scaled CMOS technologies can be used in novel computing architectures for efficient learning and inference.

The rest of the dissertation is organized as follows. In chapter 2, we focused on connectivity issues arising from Network On Chip (NOC)-based SNNs. The brain-like connectivity requires modification to typical NoC architectures. While NOCs can provide very high throughput, they suffer from high power and area requirements and lacks the connectivity required for neural computing. We propose augmenting local connectivity of NoC computing units with Power Line Communication (PLC) to communicate data between computing units that are spatially further apart. The intrinsic broadcast based communication in PLC not only brings in higher connectivity but also enables energy efficient communication, where data sent over power line can be transmitted to multiple neurons in a single cycle.

In chapter 3, we propose stochastic bit enabled binary SNN with on-chip STDP learning for memory-compressed neuromorphic computing. The binary SNN composed of stochastic neurons and binary synapses are programmed stochastically during training. We present an energy-efficient realization of the binary SNN using Biased Random Number Generator (BRNG) based ‘stochastic bits’ fabricated in 90nm CMOS process for on-chip pattern recognition. The proposed BRNG enabled binary SNN, with high power efficiency of 89.49 TOPS/Watt for two-layer fully-connected SNN of 400 neurons, offers a potential solution for energy-efficient edge computing with on-chip intelligence.

In chapter 4, we experimentally demonstrated a distance computing primitive based on a STO-based coupled oscillator array. We have shown that the combination of injection locking scheme and its interference with a reference signal can realize the Euclidean distance computation unit. The performance of the system as an $L2^2$ unit was measured by applying randomly generated test input vectors as bias current to the STOs. The characteristic curve from the experiment approximates an $L2^2$ norm which, in turn, is used as input to simulations that demonstrate the hybrid system as both a distance metric and a convolution computational primitive for image processing applications. Energy and scaling analyses show that the STO-based coupled oscillatory system has higher efficiency than the CMOS-based system with an order of magnitude faster computation speed in distance computation for high dimensional input vectors. Modest improvements in STO critical currents and magneto-resistance (through the use of magnetic tunnel junctions) can make oscillator-based systems even more attractive.

Finally, chapter 5 summarizes the thesis and discusses the future work.

2. POWERLINE COMMUNICATION FOR ENHANCED CONNECTIVITY IN NEUROMORPHIC SYSTEMS

Neuromorphic Computing (NC) has acquired tremendous interest because of its ability to overcome the limitations of von-Neumann systems in data intensive applications. NC systems are inspired from the human brain, which combines storage (synapse) and compute (neuron) to circumvent the memory bottlenecks in von-Neumann computing. Note, human brain consists of densely connected neurons, where each neuron can connect to 1000s of synapses [11]. Such dense connectivity is the key to obtaining high classification accuracies in NC systems such as the Spiking Neural Networks (SNNs), as connectivity enables hierarchical learning with large number of features in each hierarchy.

Past research has focused on many-core architectures which implement synapses with memristive crossbars to overcome the memory bottlenecks and enable efficient compute. However, mimicking brain-like connectivity poses significant challenges. This is because typical computation cores in a many-core architecture are connected with a Network On Chip (NOC). While NOCs can provide very high throughput, they suffer from high power and area requirements which decimates the benefits of efficient synapse implementation with memristive crossbars. In this chapter, we propose a Power Line Communication (PLC) based architecture built with memristive crossbars for SNNs.

PLC can use the on-chip power lines augmented with low-overhead receiver and transmitter to communicate data between neurons. This removes the high area overhead due to channels and routers present in an NOC. Further, the intrinsic broadcast based communication in PLC enables energy efficient communication, where data sent over power line can be transmitted to multiple neurons in a single cycle. Hence, PLC can enable dense connectivity required in SNNs, while preserving the efficiency of memristive crossbars. We perform evaluations of SNNs ranging in scale from 1M - 10M synapses to demonstrate the efficiency of PLC based system. Also, we propose a hybrid PLC - NOC based design which can achieve high throughput along with area and energy efficiency.

2.1 Introduction

Deep Neural Networks (DNN) are a class of machine learning algorithms and are extensively deployed in several learning tasks such as computer vision [12], speech and language processing [13], medical imaging [14], robotics [15] and gameplay [16]. DNNs are motivated from human brain and consist of densely connected neurons and synapses organized in a hierarchical fashion. This hierarchical nature and dense connectivity enable feature extraction from an input and its subsequent classification. The key feature that has enabled DNNs to achieve unprecedented performance in complex tasks is the ability to design and train large scale (in terms of neurons, synapses and layers) models. For example, in 1998, LeCun et al. used a model with less than ∼1M synapses for simple digit recognition tasks [17]. In 2012, Krizhevsky et al. proposed AlexNet with ∼60M synapses to recognize complex natural images [18]. Recently, Karpathy et al. proposed a DNN to convert image to natural language using ∼230M synapses [19].

Despite the success of DNN, their execution on von-Neumann systems suffer from extremely high energy consumption. This has motivated research in energy-efficient design of DNNs using various algorithmic and hardware techniques [20–22]. One of the promising pathways is design of biologically plausible algorithms namely Spiking Neural Networks (SNN) [2, 11, 23]. Unlike artificial neural networks (ANNs) which use real-valued inputs, SNNs communicate data in the from of spikes (0/1). The spike based inputs simplify the computations to simple add and accumulation instead of multiply-add and accumulate in ANN. Further, the resulting input sparsity in data enables event-driven computations which alleviate unnecessary memory accesses and computations to bolster the energy benefits.

To this end, past work have looked into deep SNN designs in different application domains and achieved near DNN classification accuracies [24–27]. However, their acceleration on von-Neumann machines built with CMOS technology suffers from memory bottlenecks. This is because SNNs are data-intensive applications with simpler compute requirements. Consequently, the frequent movement of data between memory and compute units on von-Neumann systems results in high data access energy and latency. Also, the ever growing model size increases the memory demands for storing the synapses. As a result, researchers have focused on designing application-specific accelerators based on CMOS technology to enable energy-efficient execution of SNNs [11, 23]. However, the inherent mismatch between CMOS technology and compute primitives required in SNNs (neurons and synapses) limits their benefits. For instance, a CMOS implementation of synapse requires more than a dozen of transistors [20].

To this effect, researchers have focused on post-CMOS technology based neuromorphic computing. Consequently, memristive crossbars have been proposed which can store synapses and perform dot-product computations in a single time step. A memristive crossbar is comprised of memristors at each cross-point that can encode a multi-bit value using one device. Upon applying an input voltage on the crossbar's row, the resulting output current on a column is equal to the weighted summation of input and synapse on the column. Subsequently, several past works [20, 28–32] have utilized the intrinsic suitability of memristive crossbars to design DNN accelerators.

Typical memristive crossbars based accelerators use array of Processing Engines (PE) connected together with a Network On Chip (NOC). The PEs are built with crossbars to enable synaptic storage and weighted summation computations. An NOC connects these PEs to realize the connectivity structure in DNNs. However, typical NOCs have significantly higher area and power consumption, thereby reducing the storage and compute benefits that can be harnessed from memristive crossbars.

Further, data transfer in typical NOC requires several hops between routers which increases the latency and energy consumption. Consequently, this either restricts the algorithm designers to use under-performing SNNs, or incur high energy-latency costs.

To address these limitations, we propose a Power Line Communication (PLC) based approach to enable efficient data transfer between PEs in a spatial architecture. PLC uses the on-chip power lines augmented with low overhead receiver and transmitter to enable communication between PEs. The minimal hardware overhead incurred, preserves the benefits from memristive crossbars. Further, the inherent broadcast based nature of PLC, where a data on power line is received by all PEs on the chip maximizes the input reuse pattern common to DNNs. However, the data transfer throughput obtained from PLC is lower than typical NOCs. Hence, we also propose a hybrid PLC-NOC based memristive crossbar architecture to maximize throughput and boost connectivity. A high throughput NOC enables fast communication between a cluster of PEs located over short-distance. On the other hand, the PLC network enables long-distance communication to boost the overall connectivity between PEs.

In summary, we make the following contributions:

- **powerline based communication** approach to enhance the connectivity for achieving higher classification accuracy in neuromorphic systems.

- **hybrid PLC-NOC based memristive architecture** for obtaining high throughput and energy efficient acceleration of SNNs.

- **evaluate** the design over a wide range of image recognition applications to study the energy and performance benefits.

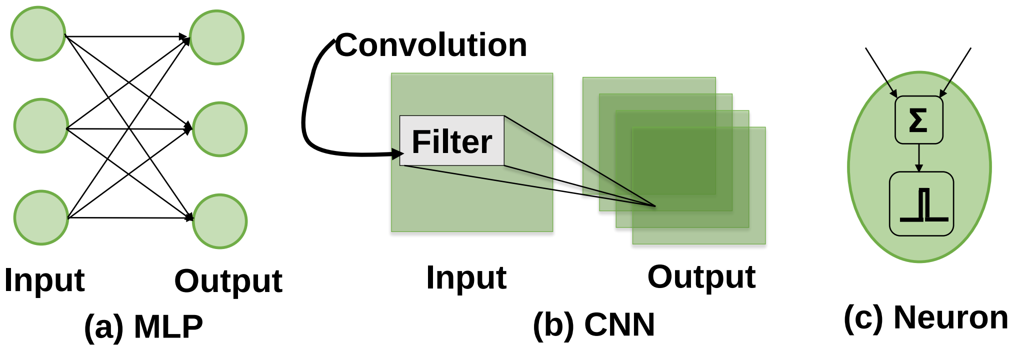


Fig. 2.1. (a) Multi layer perceptron SNN (b) Convolutional neural network SNN (c) Integrate-and-Fire neuron

2.2 Background

2.2.1 Spiking Neural Network

SNN is regarded as the third generation neural network [2]. It is a more bio-plausible version of classical neural networks and involves spike based communication between neurons. Each input to an SNN is encoded as a Poisson spike train, where the spike frequency represents pixel intensity. At a particular instant, each spike is propagated through the layers of the network while the neurons accumulate the spikes over time as its membrane potential. A neuron spikes when the membrane potential exceeds a threshold. Subsequently, the output spike is sent to the neurons in the next layer of SNN. The deep SNN topologies used in this work are Multi Layer Perceptrons (MLP) and Convolutional Neural Networks (CNN). An MLP, shown in Fig. 2.1(a), is a multi-layered SNN in which all neurons in a layer are connected to all neurons in the previous layer. A CNN, shown in Fig. 2.1(b), is also a multi-layered SNN composed of alternating convolution and sub-sampling layers. As shown in Fig. 2.1(c), a typical spiking neuron does an accumulation operation followed by thresholding operation. The spiking neuron model used in this work is the Integrate-and-Fire (IF) model. Note that, our work focuses on the testing/computation of the SNN and assumes that the memristive crossbars have already been programmed with

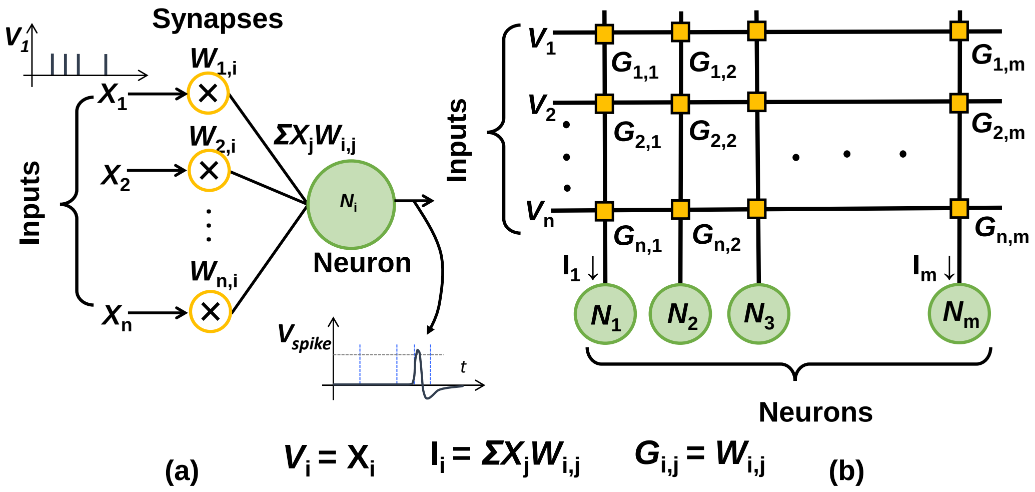


Fig. 2.2. (a) Typical spiking neural network (b) Mapping a spiking neural network on memristive crossbar

the trained weights. Hence, we do not consider energy or latency associated with writing the memristors.

2.2.2 Memristive Crossbar and Past Work

Fig. 2.2(a) shows a 2-layer fully connected SNN. Fig. 2.2(b) shows the connectivity structure/matrix (from Fig. 2(a)) mapped onto a memristive crossbar. The memristive devices at its cross-points encode the synaptic weights of the SNN. A memristive crossbar receives voltage inputs at its rows and the resulting current output at any column is the weighted summation of the encoded weights at that column and the input voltage. This is a direct consequence of the Kirchhoff's law as the current output into a column from any cross-point will be the product of the conductance at that cross-point and the voltage across it. Thus, a memristive crossbar is an analog "dot-product" computation unit. Further, each crossbar is interfaced with neurons that receive the input current and accumulate membrane voltage over time.

Several previous works have explored memristive crossbar based architectures for accelerating DNNs. ISAAC [29] designs digital computation units with ReRAM crossbars for accelerating CNNs. PRIME [30] proposes a crossbar based design for CNNs, where a crossbar is logically partitioned to be used for regular memory storage and computations. RESPARC [20] proposes a hierarchical architecture built with crossbars for executing SNNs. Further, TIME [31] and PipeLayer [32] focus on training DNNs using memristive crossbars. This chapter complements the previous works by enabling efficient communication between processing units to enhance the utility and efficiency of crossbar based architectures.

2.3 Power Line Communication

2.3.1 Motivation

Spatial many-core architectures built with emerging post-CMOS technologies have been extensively explored for DNNs, owing to their ability to exploit data parallel nature of multi-layered neural networks [20, 29–32]. However, the NOC size scales linearly with the number of cores, which leads to high energy consumption in communication between distantly located cores. As a result, researchers have explored techniques for efficient multi-hop communication to optimize the high latency and power consumption involved in moving data between two far apart cores in a many-core architecture.

For conventional NOCs, different ways to improve channel utilization such as virtual channels, bypass routers, and the novel topologies have been studied [33], [34]. On the other hand, using new materials rather than metal wires have been studied, such as three-dimensional integration, nanophotonic communication and on-chip wireless links [35–37]. We propose PLC as a new channel that uses metal wires not only for delivering power but also for carrying data to different nodes in a chip. Further, PLC enables single hop links, thanks to the inherent broadcast nature of on-chip powerlines. PLC has lower data rate compared to typical interconnects such as, conventional NOCs and wireless links. However, the low latency and low power consumption for long distance communication along with with small area penalty make PLC an attractive solution especially in many-core neuromorphic systems. Recent works have shown the use of PLC at integrated-circuit level for the purpose of design for testability (DFT) on CMOS microprocessors [38, 39].

2.3.2 Challenges

PLC mainly draws advantages from 1) dual usage, and 2) ubiquitous accessibility of Power Distribution Network (PDN). Since PLC uses the existing power supply line, it does not cost extra to obtain an additional channel for communication. Further, every circuit at any node connects to the power line and can receive data from it, thereby enabling data broadcast. Despite the apparent advantages of PLC, there are challenges that limit its applicability. First, loading data on top of power line, which works like a noise to PDN, is conflicting to the goal of a robust PDN design. Second, PLC channel suffers from simultaneous switching noise (SSN) resulting from the large current drawn by the switching devices. Third, ubiquitous accessibility to any internal nodes based on wide area network costs large power consumption to load data on top of the huge PDN.

2.3.3 PLC based neuromorphic system design

Neuromorphic algorithm such as SNNs are inherently error resilient and can harness the advantages from PLC with little or no accuracy loss [40]. Recall, typical neural network accuracy increases with network size, thereby increasing the connectivity requirements for their on-chip deployment [22]. Additionally, typical SNNs have high data sparsity and input sharing which enable low data rates and enhance the benefits of broadcast based interconnects. Thus, PLC based neuromorphic system design can enable the needed connectivity to enable large scale SNN acceleration in an efficient way. Next we discuss the two components of our PLC design namely 1) transceiver, and 2) PDN.

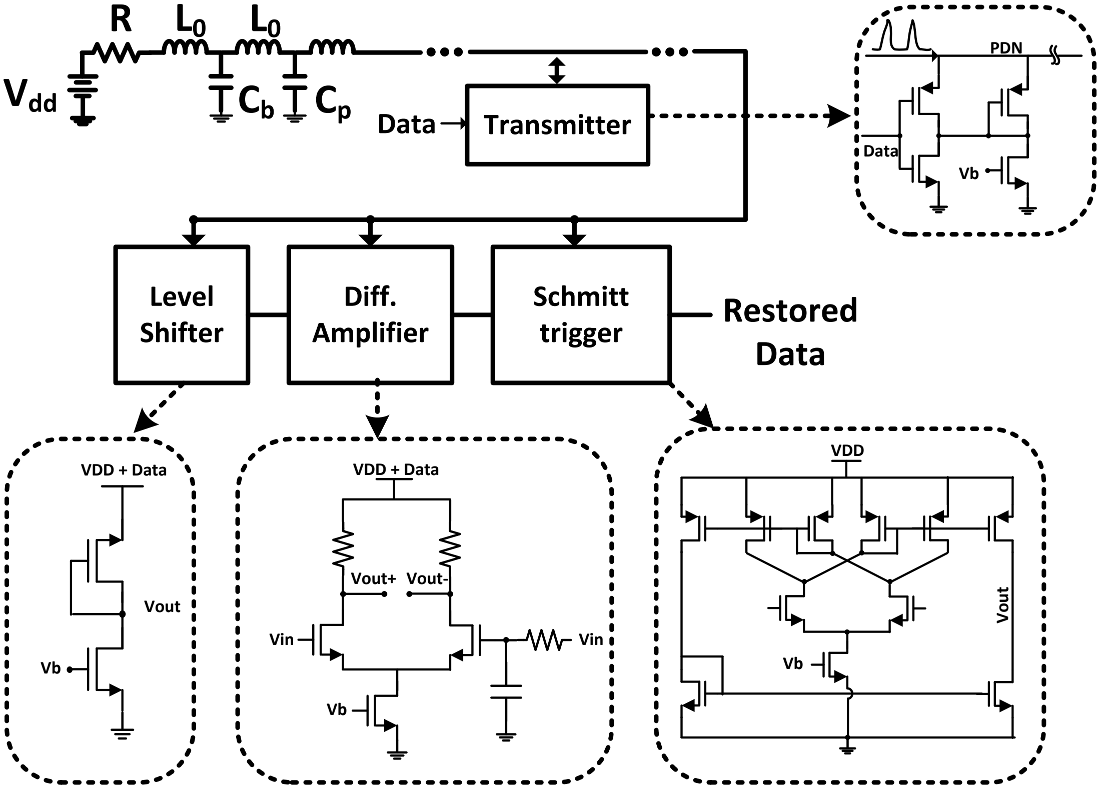


Fig. 2.3. Block diagram of PLC transceiver

Transceiver

Transceiver facilitates the send and receive of data between different processing engines (PEs) in a neuromorphic architecture (discussed in Section 2.4). Depending on the data to be sent, the transmitter (Tx) induces small glitches on the powerline by drawing current. This mechanism of loading data is similar to the way SSN gets added to a powerline. To distinguish the data from noise at the receiver side, glitches caused by data should be larger than the ones caused by SSN. In other words, a more noisy powerline would require larger amount of current (and power) for data transmission. Hence, keeping the powerline quiet is beneficial for PLC to achieve low-power communication. We achieve this by separating the powerline for noisy/clocked

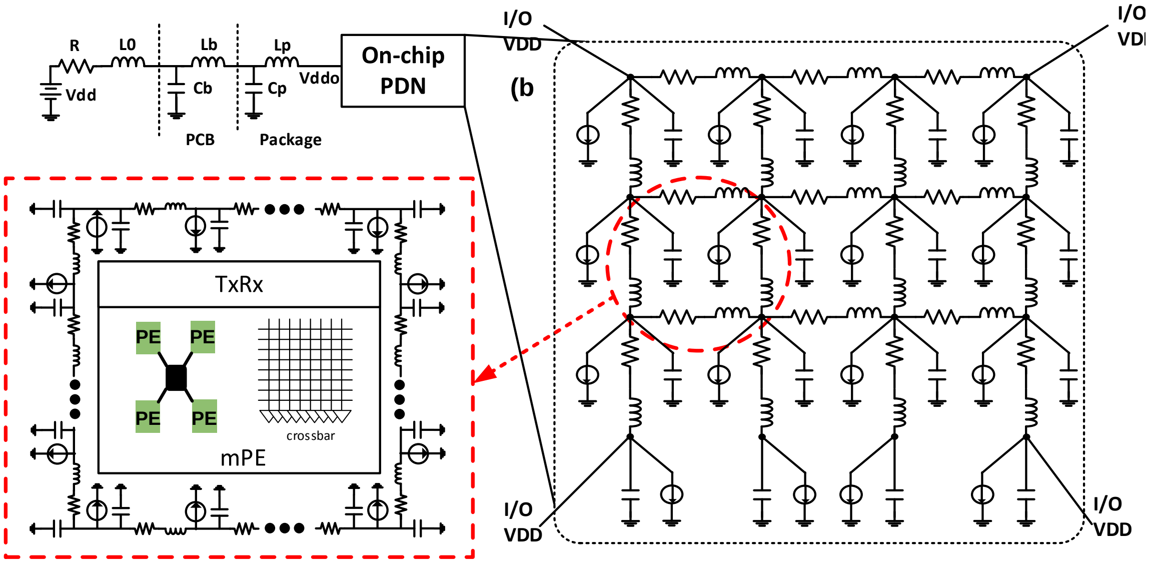


Fig. 2.4. Distributed RLC PDN model

blocks, for example, control unit, peripherals etc. from the non-noisy blocks like neurons and crossbars. In other words, the powerline used in communication connects to SSN free asynchronous components.

At the receiver side, a level shifter lowers the dc level of the signal to around half of the supply voltage (shown in Fig. 2.3). The circuit is a common source amplifier with a diode-connected load. In contrast to a typical amplifier, the power supply rejection ratio (PSRR) of level shifter is set to be small, in order to acquire the data from the signal on powerline [39]. Next, the level shifted signal is applied to a differential amplifier, in which one of the inputs is connected to an RC low pass filter for alleviating the dc offset voltage. Finally, the amplified signal goes into a differential Schmitt trigger for restoring the original data.

Power distribution network

We design a PDN comprised of power supply source, Printed Circuit Board (PCB), and on-chip power grid that accurately models the channel for PLC. The PCB and package parasitics are considered as a lumped model. However, a lumped model for on-chip power grid is not sufficient to estimate the desired characteristics of delay and attenuation at different locations of the PDN. Therefore, a distributed RLC model for on-chip power grid has been designed (shown in Fig. 2.4). Here, V_{DD}, R and L_0 are power supply source, the resistive impedance, and the inductance of Voltage Regulator Module (VRM) respectively. The parasitics of R,L and C are modeled for on chip power grid using Equations 2.1, 2.2, and 2.3 [41]. Rs, μ, and ϵ are sheet resistance, permeability, and permittivity respectively, where l, w, t, and h are length, width, thickness, and height from substrate.

$$R = R_s(\frac{l}{w}) \tag{2.1}$$

$$L = \frac{\mu}{2\pi} \ln{(\frac{8h}{w} + \frac{w}{4h})} \tag{2.2}$$

$$C = \epsilon[(\frac{w}{h}) + 0.77 + 1.06(\frac{w}{h})^{0.25} + 1.06(\frac{t}{h})^{0.5}] \tag{2.3}$$

2.4 Hybrid PLC-NOC based Architecture

We implement a hierarchical architecture using memristive crossbars as shown in Fig. 2.5. As shown in past works, a hierarchical architecture can efficiently exploit the available data-parallelism in SNNs [20]. Our novelty lies in proposing a hybrid PLC-NOC based technique to enable efficient communication between PEs in a tile.

Fig. 2.5(a) shows the organization of a Processing Engine (PE). A PE consists of multiple memristive crossbars, each interfaced with an input memory (InMem), neuron block (Neuron) and output memory (OutMem). Data received by a PE can

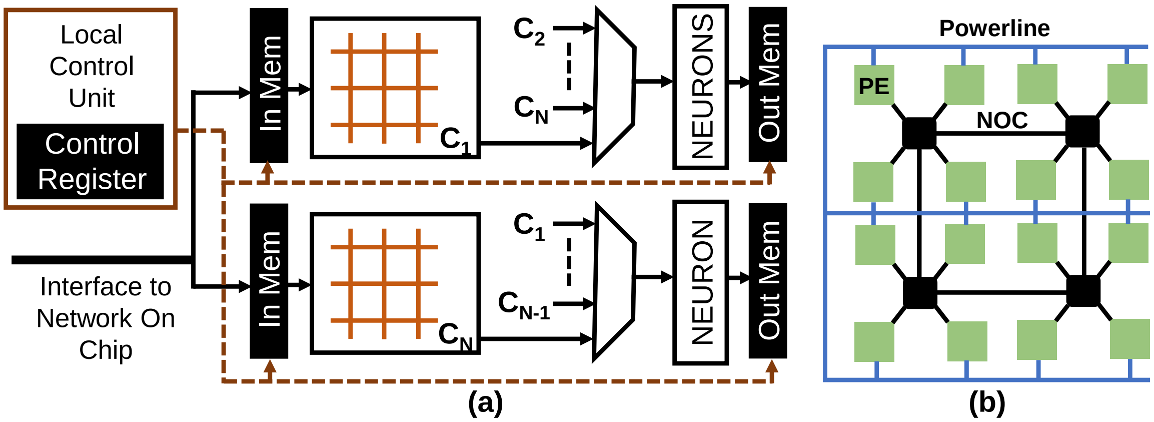


Fig. 2.5. (a) Processing Engine (PE) (b) Memrsitive crossbar architecture comprised of multiple PEs connected through hybrid PLC-NOC

be stored in one or multiple InMem depending on the input sharing between crossbars in a PE. For instance, crossbars on a PE mapped to different output neurons within an SNN layer will share inputs. Additionally, multiple crossbars in a PE can accumulate their outputs on different neuron blocks in a PE to produce final outputs. Lastly, outputs are sent to PEs mapped to the next layer of SNN.

Fig. 2.5(b) shows the architecture with multiple PEs connected through hybrid PLC-NOC. An SNN is partitioned to map the weights in different layers on different PEs. Further, within an SNN layer, weights are mapped across multiple crossbars on one or more PEs. This is because, typical crossbar sizes are an order of magnitude smaller than SNN layer sizes. Crossbar sizes can be limited by parasitic effects, sneak paths and peripheral overheads [20, 42]. Consequently, an SNN execution on memristive crossbar architecture can be subdivided into three operations 1) crossbar computation, 2) ***intra-layer data transfer***, and 3) ***inter-layer data transfer*** (illustrated in Fig. 2.6).

We propose a hybrid PLC-NOC based architecture that leverages PLC and NOC for inter-layer and intra-layer communication, respectively. An NOC-only design consumes significant energy consumption ($\geq \sim$60% of the total energy) due to inter-layer

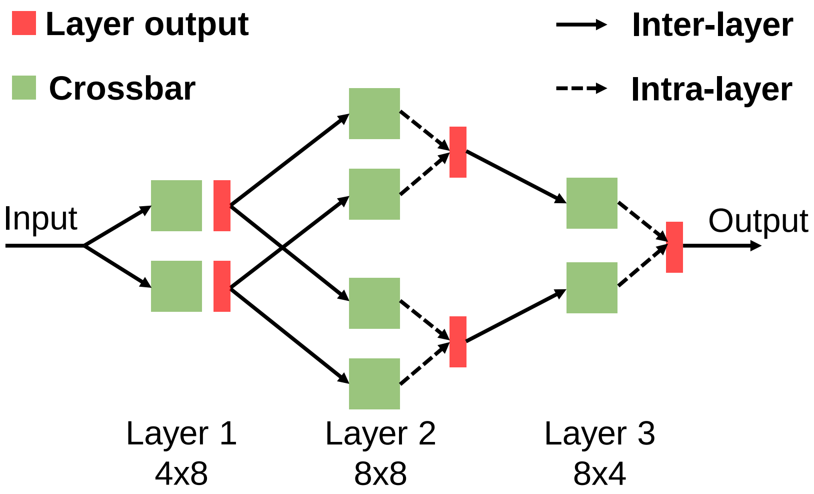


Fig. 2.6. SNN execution on multiple PEs showing intra-layer and inter-layer data transfers. Each PE can implement a 4×4 size layer.

transfers (shown in Sec. 2.6.4). To overcome this limitation of NOC-only design, PLC enables efficient inter-layer communication. PLC based broadcast provides a natural fit for one-to-many nature of *inter-layer data transfers*. Further, NOC provides a high throughput communication medium for *intra-layer data transfers*. Thus, the hybrid approach preserves inference latency while reducing the energy consumption.

Inter-layer data transfers implement SNN execution by propagating input through multiple layers to compute the classification output. Such transfers are one-to-many in nature where a PE mapped to previous layer sends its output data to multiple PEs mapping the next layer (shown in Fig. 2.6, Layer 2). This is because typical SNNs have input sharing where, neurons in the next layer share the outputs produced by the previous layer. PLC enables harnessing this input reuse in SNNs by broadcasting output data over power lines, which is received by next layer PEs in a single time step. On the contrary, an NOC based inter-layer data transfer will require multiple data transfers of same input data, where each transfer sends data to one PE in the next layer. This leads to increased energy consumption. Further, data transfer over NOC typically incurs multiple hops, owing to the large number of PEs (typically

100s) required to map any given layer. This leads to increased latency and energy consumption.

Intra-layer data transfers occur when neuron fan-in exceeds the crossbar size. Crossbars in multiple PEs compute partial products, which are aggregated through intra-layer data transfers to compute final outputs. These transfers are one-to-one, and occur between closely located PEs (typically 4-8). Further, multiple intra-layer transfers occur in parallel to compute multiple output neurons concurrently. Hence, PLC is not suitable for intra-layer communication as only one data can be transmitted over powerline at any given time. However, NOC based intra-layer transfers enable parallel communication (one-to-one) and are energy efficient owing to the closely located PEs.

2.5 Experimental Methodology

2.5.1 Neuromorphic architecture

The RLC parameters for modelling the on-chip PDN are taken from IBM 45nm process, which uses some fitting parameters in addition to the Equations 2.1, 2.2 and 2.3 in Section 2.3. A set of typical values for the PCB board and package parasitics is provided in [43]. A $1960\mu m \times 1960\mu m$ power grid is estimated to cover a 13×13 array of PEs.

We modeled PDNs using different metal layers and analyzed the channel loss and the phase response, which are represented by the magnitude and the phase of S21, respectively (Fig. 2.7). Higher metal layers have higher metal thickness and less parasitic resistance. Thus, powerline is typically designed with the highest metal layer in order to minimize the voltage drop between power nodes. It can be observed that PDN modeling using 9^{th} and 10^{th} metal layers (out of 11 layers) shows the lowest channel loss. However, the 9^{th} metal layer is 10× thicker than the 1^{st} metal layer, which makes it unsuitable for low-power requirements. Therefore, in this work, we use the 6^{th} and the 7^{th} metal layers, which are 2× thicker than the 1^{st} metal layer.

The PDN model is designed as rectangular power grid meshes, and it surrounds a PE (in the PE-array) of size 0.34 mm^2. Further, each PE has a transceiver (area ≤ 0.001 mm^2) located at the center of its power grid mesh.

The overall PE design was adopted from RESPARC [20]. Cycle-level NOC simulations were performed with Booksim2 [44] and Orion2 [45] was used for power measurements.

2.5.2 PLC channel quality

In contrast to digital data links in an NOC, PLC based communication is analog in nature. Therefore, channel's robustness needs to be guaranteed in terms of Bit Error Rate (BER), that depends on channel noise, signal attenuation between the communicating nodes, PDN modeling parameters, the size of decoupling capacitors etc. Effectively, the PLC channel should satisfy a level of BER ($\leq 10^{-5}$), which does not degrade output quality (SNN classification accuracy), irrespective of the distance between any two PEs considered for communication. BER primarily depends on the level of power supply noise (primarily SSN). Hence, the required transmitter power is determined by the amplitude of the noise on the powerline. Recall, our proposed PLC channel provides power to non-synchronous (non-clocked) components only, namely crossbar arrays and neurons. Thus, PLC will get affected when the neurons produce an output spike. However, for the worst case analysis, we assumed that the SSN noise has same frequency as the data bandwidth, because in-band noise degrades the channel quality significantly. Subsequently, for estimating the signal power required for a given noise level, we add an ideal current source to every line segment, such that its switching frequency matches the PLC bandwidth.

Typically, decoupling capacitors are added to counter the effect of switching noise. However, in our design, we purposely generate the switching noise to use it in PLC. Thus, the channel bandwidth is also affected by the decoupling capacitance. Each PE consumes an average power of 3mA and 20pF of decoupling capacitance is assumed

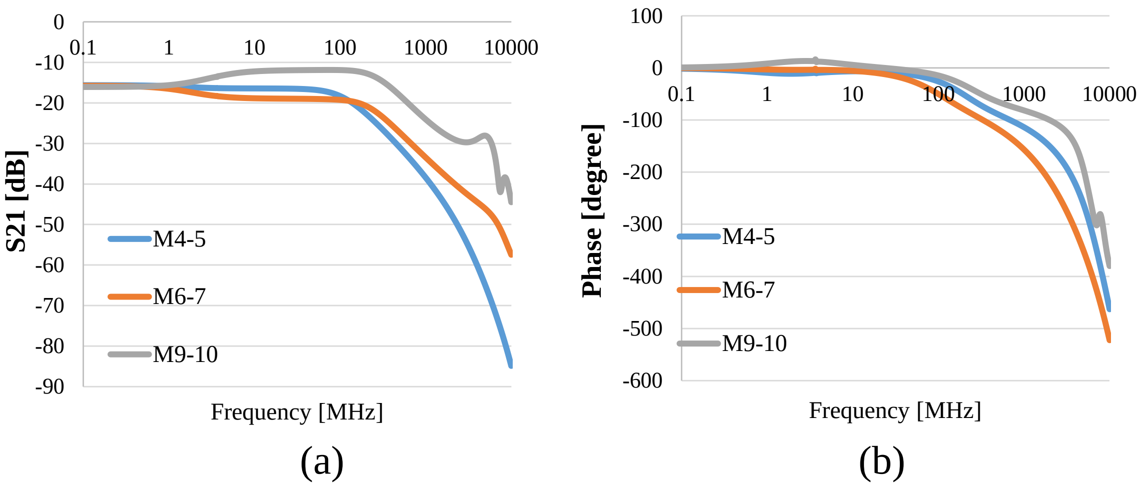


Fig. 2.7. (a) Channel loss, and (b) phase of the PDN channel for the longest path in 13x13 array of PEs using different metal layers

at the center of every PE. Considering the aforementioned requirements, we use a 1Gbps data rate for PLC for our simulations.

2.5.3 Benchmarks

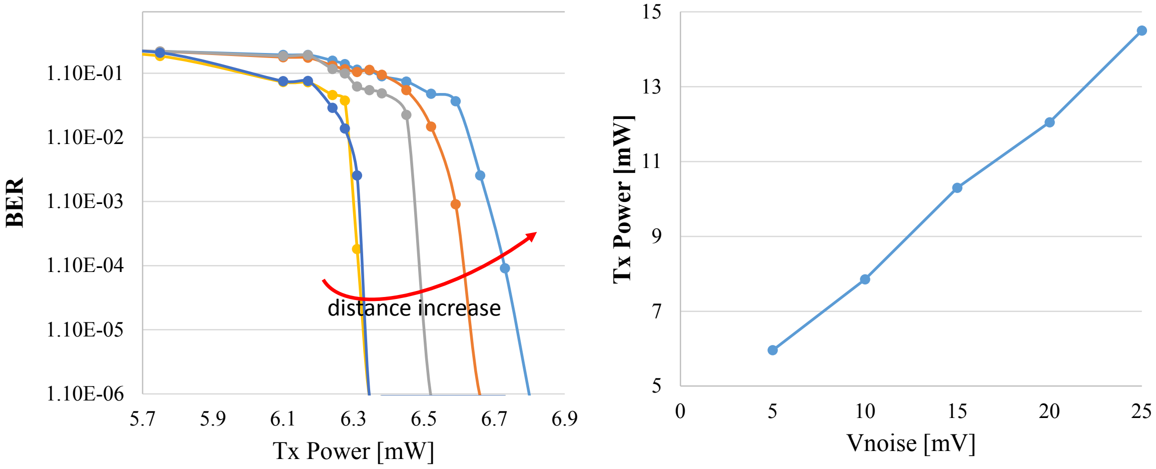
We construct MLP and CNN based SNNs for three popular image recognition datasets - MNIST [46], CIFAR-10 [47] and SVHN [48]. The SNNs were trained using supervised learning approach proposed in [24], wherein an ANN is trained using error back-propagation followed by its conversion to SNN [24]. Table 2.1 details the evaluated SNN benchmarks. We use 4-bit precision for inputs and weights, which obtains high classification accuracy on the SNN benchmarks [20].

2.6 Results

This section discusses the Tx power requirements for PLC. Further, the energy and latency of hybrid PLC-NOC based neuromorphic architecture is analyzed compared to PLC-only and NOC-only systems. Our analysis focuses on the inference or testing

Table 2.1.
SNN benchmarks

Application	Dataset	Layers	Neurons	Synapses
Digit recognition	MLP0	3	2378	1902400
	CNN0	3	66778	1484288
House recognition	MLP1	3	2778	2778000
	CNN1	3	124570	2941952
Object classification	MLP2	4	3778	3778000
	CNN2	3	231066	5524480

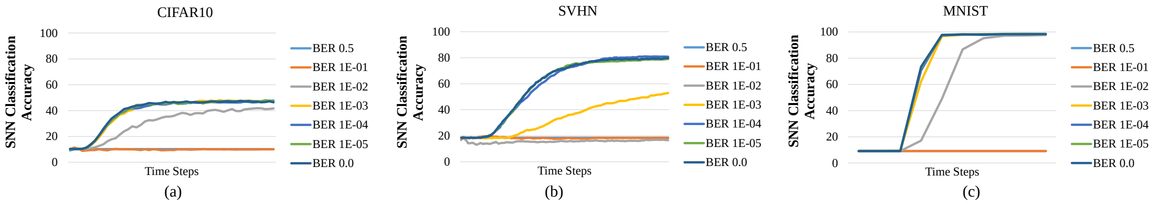


(a)

(b)

Fig. 2.8. (a) BER vs. Tx power @V$_{noise}$ = 5mV (b) The required Tx power vs. noise amplitude

phase as edge devices are typically used for deployment of inference applications whereas, training is performed in the cloud [22].



(a) (b) (c)

Fig. 2.9. Impact of BER incurred due to PLC on SNN classification accuracy for (a) CIFAR10 dataset (b) SVHN dataset and (c) MNIST dataset

2.6.1 Tx power requirements

Fig. 2.8(a) shows that the power required to maintain the same level of BER increases with increasing distance between the communicating PEs. The required BER level is decided based on the classification accuracy degradation for different BERs. Consequently, we use BER of $\leq 10^{-5}$ which suffices for SNN applications. Since signal attenuation increases with increasing distance between PEs, the largest attenuation occurs when the center PE transmits and the outermost PE receive. This enables us to estimate the minimum Tx power required for robust communication between PEs located at any distance. Subsequently, we analyzed the BER versus Tx power dependence for varying noise levels (V_{noise}) which is shown in Fig 2.8(b). The Tx power thus obtained are used in system-level simulations and are discussed in the following subsections.

2.6.2 Impact on classification accuracy

Fig.s 2.9(a), (b), and (c) show the impact of BER incurred in data transmission due to PLC on SNN classification accuracy. It is evident that BERs in range of $\leq 10^{-5}$ can ensure accurate SNN inference, which also justifies the choice of BERs used in Section 2.6.1. The applicability of relatively higher BER than typical requirements of (10^{-10}) is enabled by the error-resilient nature of neural network applications.

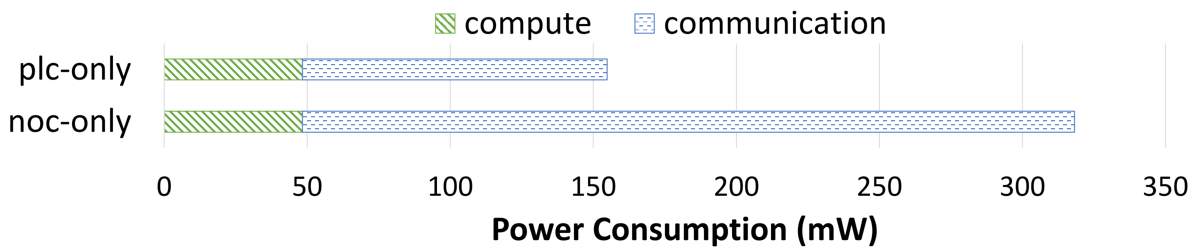


Fig. 2.10. Power consumption of chip

This error-resiliency comes from the use of non-linear functions (for instance sigmoid, clamped ReLU), which squash the output of matrix transformation layer to a small output range (typically 0 – 1) thereby suppressing errors resulting from small perturbations in input. It can also be seen that complex datasets such as CIFAR10 and SVHN have more constrained BER requirements than simpler datasets for achieving ideal classification accuracy (obtained for ideal PLC with no error i.e. BER 0.0).

2.6.3 Overall power consumption

Fig. 2.10 compares the power consumption of the neuromorphic architecture (discussed in Section 2.4) with PLC-only interconnect to a NOC-only architecture (baseline). Compute (synapse and neuron operations within PEs) constitutes only ∼15.17% of power consumption in the NOC-only architecture. This is because of the simple computation nature of SNNs, which consists of accumulations only, compared to multiplication and accumulation in ANNs. Efficient dot-product operation in memristive crossbars further reduce the PE power consumption.

A PLC-only architecture uses low-power on-chip interconnection network enabled by power lines and low overhead receiver and transmitter per PE. Thus, PLC-only system enables overall power reductions of ∼51.34% compared to NOC-only system for SNN acceleration. Low power is extremely valuable in power-constrained edge devices and battery-powered systems. Note that, while a PLC-only system has sig-

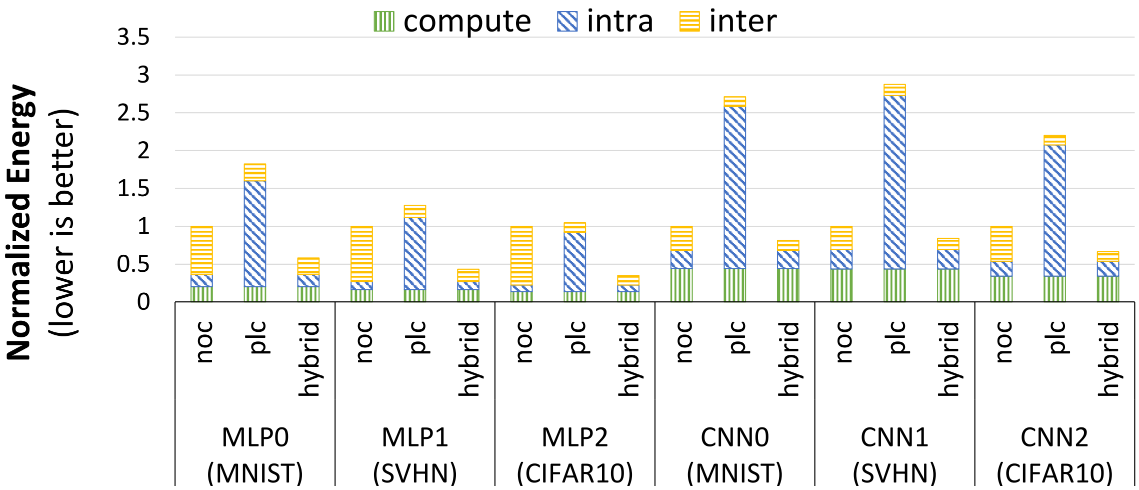


Fig. 2.11. Inference energy (batchsize = 1)

nificantly higher latency cost (discussed in Section 2.6.5), the power benefits can be useful in latency insensitive (or latency tolerant) applications.

2.6.4 Inference energy

Fig. 2.11 shows the inference energy consumption of six SNN applications on neuromorphic architectures built with three different interconnection namely (1) NOC-only, (2) PLC-only, and (3) hybrid PLC-NOC (discussed in Section 2.4). MLPs have inter-layer data transfers with high input sharing (within an inference) because of their fully-connected design. Recall, all outputs produced by the previous layer are used by all the next layer neurons. This results in large number of inter-layer communications for transferring input data to multiple PEs mapping the next layer, in an NOC-only system. Further, each data transfer requires multiple hops due to the large distance between PEs mapping the successive layers. A PLC-only system harnesses the high input-sharing in fully connected NNs (one-to-many) to reduce the energy consumed in inter-layer communication. However, intra-layer communication are more efficient in the NOC-only system compared to the PLC-only system. This is because of the one-to-one nature of intra-layer communication, wherein no input sharing exists between the data-transfers. Consequently, the hybrid PLC-NOC architecture enables efficient inter-layer communication (PLC) and intra-layer communication (NOC) to achieve $\sim$42.75%– $\sim$65.04% reductions in energy consumption for MLPs.

Typical CNNs have smaller receptive fields than MLPs, thereby reducing the number of receiver PEs per input data packet. Thus, CNNs have lesser input sharing compared to MLPs, which reduces the benefits obtained from PLC for inter-layer data transfer. Consequently, the hybrid PLC-NOC system achieves lower energy benefits for CNNs ($\sim$15.76%– $\sim$33.74%).

2.6.5 Inference latency

Fig. 2.12 compares inference latency of the three architectures for SNN applications. PLC achieves comparable latency with respect to NOC for inter-layer communication due to the one-to-many communication nature. Here, PLC benefits from its broadcast nature, while NOC benefits from the parallel one-to-one data transfers. However, PLC incurs significant latency costs for intra-layer communication. This is because of the sequential nature of data transfer in PLC, where only one data can reside on the powerline at any given time. Consequently, while a hybrid PLC-NOC system can achieve comparable latency to an NOC-only based system, the latency of PLC-only system can be $\sim$5.46× higher than NOC-only system.

2.7 Conclusions

Low power and energy-efficient inference has become extremely important as more and more machine learning applications are being deployed in the edge devices. Further, the number of edge devices used per person have continuously increased over the past decade with majority of the devices being battery powered. This has motivated the design of neuromorphic systems to enable data processing capabilities at the edge devices. Both low-power computation units and energy-efficient interconnect

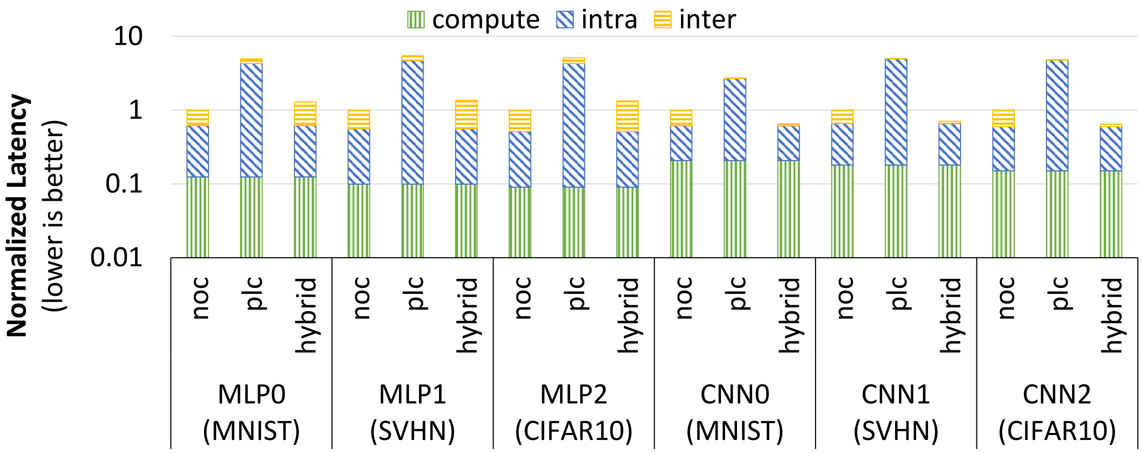


Fig. 2.12. Inference latency (batchsize = 1)

are fundamental to efficient neuromorphic system design. In this chapter, we propose a hybrid PLC-NOC based neuromorphic architecture built with memristive crossbars to enable efficient ML inference. Our hybrid interconnect harnesses the different data-transfer patterns in typical many-core architecture to optimize energy expended in data communication. Additionally, memristive crossbar based PEs achieve low energy consumption for neuromorphic computations. Our experiments over a wide range of spiking neural network benchmarks show average energy improvements of ∼39.32% at comparable latency.

3. SBSNN: STOCHASTIC-BITS ENABLED BINARY SPIKING NEURAL NETWORK WITH ON-CHIP LEARNING FOR ENERGY EFFICIENT NEUROMORPHIC COMPUTING AT THE EDGE

In this section, we propose stochastic Binary Spiking Neural Network (sBSNN) composed of stochastic spiking neurons and binary synapses (stochastic only during training) that computes probabilistically with one-bit precision for power-efficient and memory-compressed neuromorphic computing. We present an energy-efficient implementation of the proposed sBSNN using *'stochastic bit'* as the core computational primitive to realize the stochastic neurons and synapses, which are fabricated in 90nm CMOS process, to achieve efficient on-chip training and inference for image recognition tasks. The measured data shows that the *'stochastic bit'* can be programmed to mimic spiking neurons, and stochastic Spike Timing Dependent Plasticity (or sSTDP) rule for training the binary synaptic weights without expensive random number generators. Our results indicate that the proposed sBSNN realization offers possibility of up to 32× neuronal and synaptic memory compression compared to full precision (32-bit) SNN and energy efficiency of 89.49 TOPS/Watt for two-layer fully-connected SNN.

3.1 Introduction

In the current era of ubiquitous autonomous intelligence, there is a growing need for moving Artificial Intelligence (AI) to the edge to cope with the ever increasing demand for autonomous systems like drones, self-driving cars, and smart wearable devices. Energy-efficient neuromorphic systems are henceforth necessary to process the massive amount of data generated by the resource-constrained battery-powered edge devices. Furthermore, it is highly desirable to embed on-chip intelligence using low-complexity learning rules, which enable the edge devices to learn from real-time inputs. Real-time on-chip learning capability precludes the need for offline training in the cloud, which can otherwise lead to higher latency and security concerns for real-time applications.

Spiking Neural Networks (SNNs), on the account of event-driven computing capability and hardware-friendly local learning using Spike Timing Dependent Plasticity (STDP), offer a promising solution for realizing energy-efficient neuromorphic systems with on-chip intelligence. In fact, researchers in [49] demonstrated that SNN running on event-driven neuromorphic hardware like Intel *Loihi* [23] incurs the minimum energy per inference relative to similarly sized analog neural network executed on CPU/GPU while providing equivalent inference accuracy for a latency-critical keyword spotting task. Recent works on deep SNNs indicate that energy efficiency significantly increases with network depth due to exponential drop in the spiking activity across successive SNN layers [50, 51]. In this regard, prior works proposed energy-efficient implementations of SNN using CMOS [11,23,52] and emerging device technologies such as Resistive Random Access Memory (RRAM) [53,54], Conductive Bridge RAM (CBRAM) [55], and Magnetic Tunnel Junctions (MTJs) [56]. However, SNNs composed of deterministic neuronal and synaptic models require multi-bit precision to store the parameters governing their dynamics. As a result, the computational complexity and neuronal/synaptic memory requirements increase with network size, leading to reduction in the overall power- and area-efficiency.

We propose and implement *'stochastic bits'* enabled binary SNN (sBSNN) that computes probabilistically with one-bit precision for energy- and memory-efficient neuromorphic computing at the edge. The core building block of the sBSNN is a *'stochastic bit'*, which switches between its logic low and high states with a probability that varies in a sigmoidal manner based on the input. We realize the stochastic neurons, referred to as sNeurons, and synapses (stochastic only during training) using the proposed *'stochastic bit'* as explained below. The sNeuron receives the weighted sum of the input spikes with the synaptic weights, and spikes probabilistically depending on the weighted input sum. The firing probability of the sNeurons, similar to the switching dynamics of the *'stochastic bit'*, has sigmoidal relationship with the weighted input sum. The binary synapse interconnecting a pair of input (pre) and output (post) neurons is similarly emulated using the *'stochastic bit'* during training. The binary synaptic weight is trained using the stochastic-STDP (sSTDP) algorithm presented in [57], where the synaptic weight is potentiated/depressed with a probability that depends on the degree of correlation between the spike times of the pre- and post-neurons. The trained binary synaptic weights are then used deterministically during inference to predict the class of a test input. The proposed sBSNN, with event-driven computing capability enabled by state-less sNeurons and memory-efficient on-chip learning capability enabled by the hardware-friendly localized sSTDP rule, offers a promising solution for building the next generation of autonomous intelligent systems.

To that effect, we propose an energy-efficient realization of sBSNN, fabricated in 90nm CMOS technology, to achieve on-chip training and inference for visual image recognition tasks. The proposed *'stochastic bit'* is composed of a cross-coupled inverter with PMOS header and NMOS footer transistors for obtaining the sigmoidal switching probability characteristics. We interface the CMOS *'stochastic bit'* with the appropriate peripheral circuitry to realize the sNeurons and synapses. The energy and memory efficiency of the proposed implementation stems from three key factors. First, the power consumed by the sNeuron for generating a spike is comparable to that consumed in a single transition of a cross-coupled inverter, which is typically in the order of few μW. In addition, the *'stochastic bit'* design also leverages power gating technique [58] with header and footer transistors between the supply and ground rails for reducing the leakage power consumption. Second, the spiking dynamics of the sNeuron depend only on the current input and not on the integrated sum of the current and past inputs, which precludes the need for storing the neuron state (typically known as the membrane potential) as is common in deterministic spiking neurons like the leaky integrate-and-fire neuron. Further, the synapses need only one-bit storage to record the respective binary states. Last, the weighted sum of the inputs with the synaptic weights, which is typically a series of multiply and accumulate (MAC) operations in analog neural networks, is transformed to AND operations followed by pulse count in the proposed sBSNN, thereby reducing the computational energy significantly. Our analysis using a two-layer fully-connected SNN of 400 neurons indicates that the proposed realization offers high energy efficiency of 89.49 TOPS/Watt, which renders it a potential candidate for enabling the next generation of intelligent devices.

In summary, we make the following contributions:

- We proposed the *'stochastic bit'* as the core computational primitive to realize the stochastic neurons and binary synapses, which are implemented in 90nm CMOS process.

- We proposed and evaluated the *'stochastic bit'* enabled sBSNN that computes probabilistically with one-bit precision for power-efficient and memory-compressed neuromorphic computing.

- We proposed and demonstrated one of the first works on all-CMOS realization of stochastic SNNs. Our proposal provides reconfigurable on-chip learning that is suitable for the real-time and resource constrained edge devices.

3.2 Background

3.2.1 Stochastic Binary Spiking Neural Network (sBSNN)

The core building block of the proposed sBSNN is a set of input (pre) neurons connected to an output (post) neuron via binary weights. The input neurons, which represent the image pixels for a visual object recognition task, generate Poisson-distributed spikes at a rate proportional to the corresponding pixel intensities. At

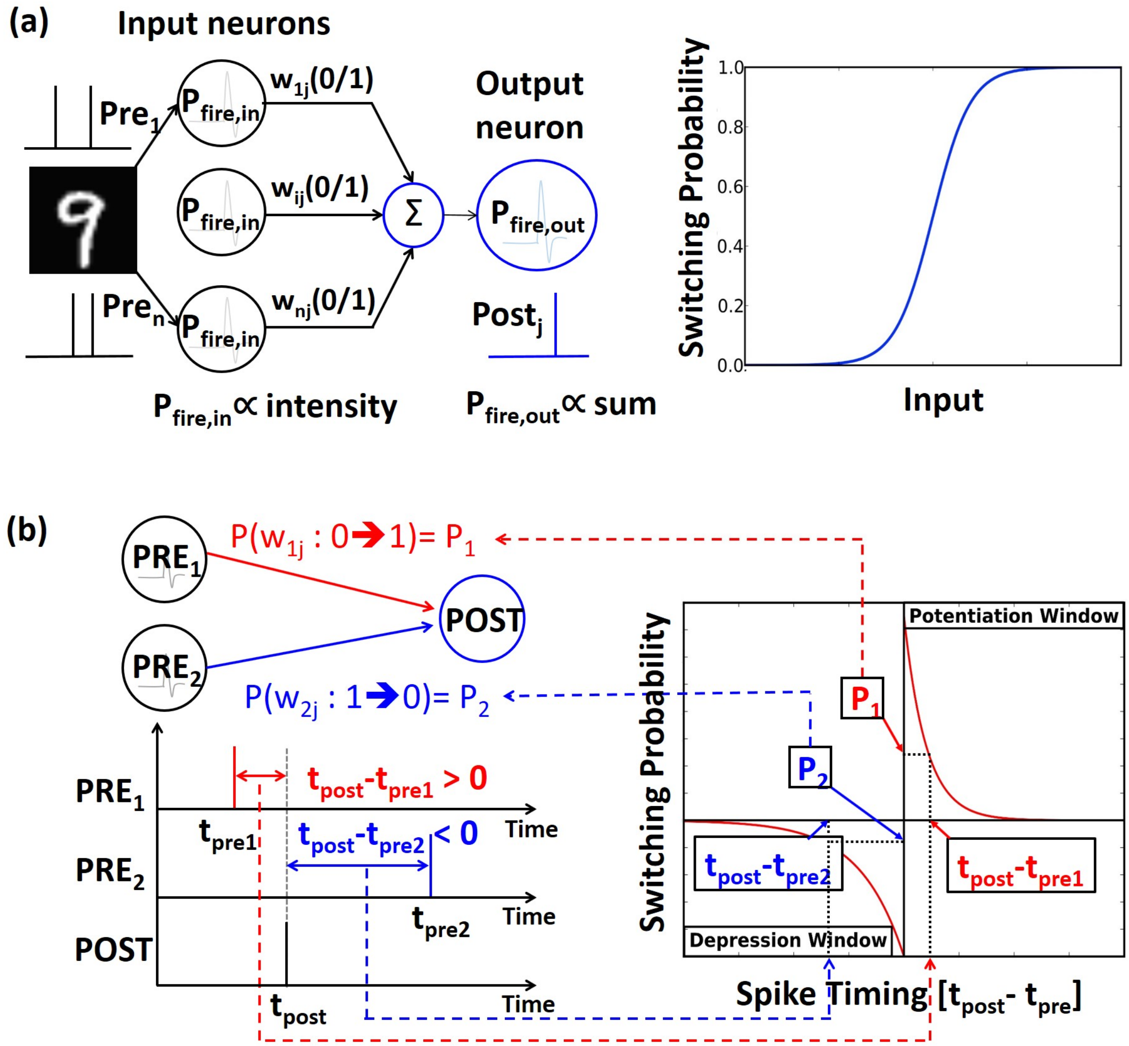


Fig. 3.1. (a) SNN composed of stochastic input and output neurons interconnected via binary synaptic weights. (b) Stochastic-STDP learning rule for binary synaptic weights

any given time, the input pre-spikes get modulated by the interconnecting synaptic weights to produce resultant current into the output neuron. Several previous works have explored the hardware implementations for these core building blocks of stochastic SNNs, using emerging technologies like CBRAMs and MTJs [57, 59] and built-in blocks in FPGA board [60]. We proposed a *'stochastic bit'* as the core building block for neuron and synapse (training) to achieve on-chip learning with compressed memory. We model the output neuron using the *'stochastic bit'*, which spikes probabilistically based on the input current (or weighted input sum) during both training and inference. The spiking probability of the output sNeuron has sigmoidal dependence on the input current as illustrated in Fig. 3.1(a). It is important to note that the sNeuron is state-less since the stochastic spiking dynamics depend only on the instantaneous input current and not on the integrated sum of current and past input currents as is typical in deterministic neuron models, thereby eliminating the multi-bit precision requirement for the neuron state (or membrane potential). The stochastic synapses (stochastic only during training) are similarly emulated using the *'stochastic bit'*, where the synaptic switching probability depends on the time difference between the pre- and post-spikes as explained in the following section 3.2.2.

3.2.2 Stochastic-STDP (sSTDP)

Spike Timing Dependent Plasticity (STDP) is a bio-inspired local learning mechansim, which has been experimentally observed in the rat hippocampus [61]. STDP postulates that the change in the weight of a multi-level synapse interconnecting a pair of pre- and post-neurons depends on the correlation between the respective spike times. If the pre-neuron spikes before the post-neuron, the synaptic weight increases (synaptic potentiation), while it decreases if the pre-neuron spikes after the post-neuron (synaptic depression). Binary synapses, on the contrary, require a probabilistic learning rule to prevent rapid switching of the weights between the high and low levels, which would otherwise render the synapses memory-less. We use the sSTDP learning algorithm proposed in [57] to train the binary synaptic weights, where the synaptic switching probability has exponential dependence on spike timing difference as illustrated in Fig. 3.1(b) and described by

$$P_{L \to H} = \gamma_{pot} \cdot e^{\frac{-\Delta t}{\tau_{pot}}} where\ \Delta t = t_{post} - t_{pre} > 0 \tag{3.1}$$

$$P_{H \to L} = \gamma_{dep} \cdot e^{\frac{\Delta t}{\tau_{dep}}} where\ \Delta t = t_{post} - t_{pre} < 0 \tag{3.2}$$

where $P_{L \to H}$ and $P_{H \to L}$ are the probability of potentiation and depression, respectively. In other words, the weight of a synapse changes based on the temporal correlation between the spike time of pre- and post-neurons. For example, if a pre- (post-) neuron fires before a post- (pre-) neuron does, it is positively (negatively) correlated with the input pattern [62]. Consequently, potentiation (depression) occurs probabilistically in the positive (negative) timing window of the sSTDP algorithm. The corresponding switching probability is determined by the spike timing difference between pre and post spikes as described in the above equations. The peak switching probability and time constant for potentiation (γ_{pot}, τ_{pot}) and depression (γ_{dep}, τ_{dep}) determine the synaptic learning efficacy. The sSTDP hyperparameters have to be chosen carefully to ensure right balance between the potentiation and depression weight updates, and achieve efficient learning. Once the training is complete, the learnt binary weights are used deterministically during inference. The presented sBSNN requires only one-bit precision for the neurons and synapses, leading to visual image recognition with compressed memory requirement.

3.3 sBSNN Design and Implementation

In this section, we first detail the design and implementation of the proposed *'stochastic bit'*, which is the core computing primitive of the sBSNN. We then present the design of sNeuron and synapse (stochastic only during training). Finally, we detail the system-level realization of two-layer fully-connected sBSNN for visual image recognition.

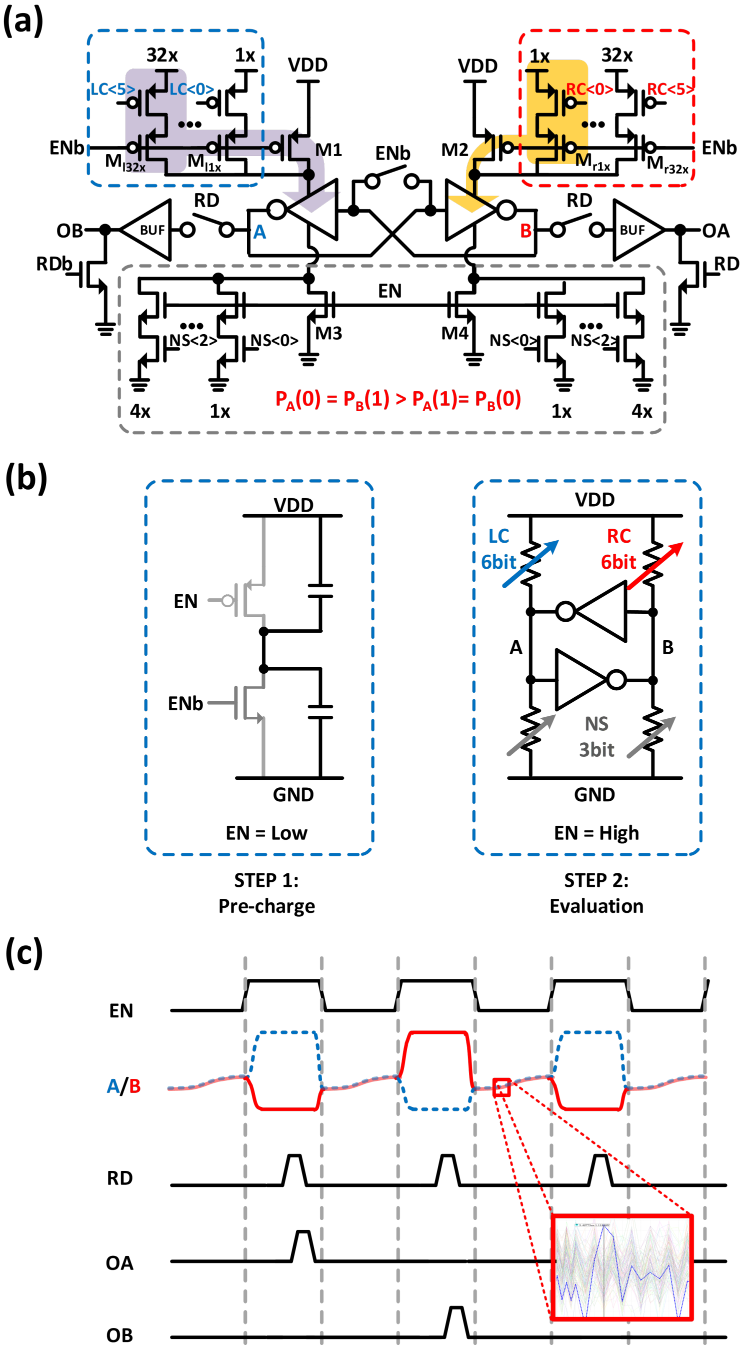


Fig. 3.2. (a) Schematic of 6-bit *'stochastic bit'* core. (b) Illustration of the pre-charge and evaluation modes of operation of the *'stochastic bit'*. (c) Timing diagram illustrating the operation of the *'stochastic bit'*.

3.3.1 CMOS 'Stochastic bit' Design

As mentioned in section 3.1, controllable stochastic behavior is the central characteristic of the *'stochastic bit'*. In CMOS-based designs, stochastic behavior is largely dependent on the characteristics of the random noise source. Thermal noise is one of the commonly used entropy sources in CMOS process, which stems from the channel fluctuations induced by random Brownian motion of electrons. The power spectral density of thermal noise across a resistor is given by $V^2 = 4kTR$, where k is the Boltzman constant, T is the temperature in Kelvin, and R is the resistance in ohms. Accordingly, thermal noise induced stochasticity is only affected by the device resistance and operating temperature. Thermal noise has been used as the source of randomness in many True Random Number Generator (TRNG) designs [63,64]. Also, metastability-based TRNG designs using cross-coupled inverters have been reported to achieve high operating frequency and power efficiency [65]. This motivated us to investigate the possibility of harnessing the metastable behavior of bi-stable circuits to implement the *'stochastic bit'*.

The proposed *'stochastic bit'* is realized using cross-coupled inverter with PMOS header transistors and NMOS footer transistors as depicted in Fig. 3.2(a). The operation of the *'stochastic bit'* is divided into two different modes, namely, pre-charge and evaluation, which are gated by the 'EN' (enable) signal as shown in Fig. 3.2(b). In the pre-charge mode (when 'EN' is low), the cross-coupled nodes A and B are pre-charged to the same voltage by leakage current, while the header and the footer transistors are turned off. Note that, the inherent power gating enabled by the PMOS header transistors and the NMOS footer transistors causes the leakage current of the proposed design to be lower than the gate leakage current of a 6T SRAM bitcell [58]. The switching probability depends on asymmetry in the effective strength of left- and right-wing PMOS transistors, which can be modulated using the input that is represented as 6-bit code in our implementation and activates different binary weighted PMOS switch transistors. The NMOS footer transistors connected to ground are controlled symmetrically in strength using the same input code, which is represented with 3-bit precision in our implementation, to modulate the shape of the probability curve. The shape of the switching probability versus the PMOS digital code is sigmoidal as will be shown in the results section 3.4. The shape and the covered range of probability is programmable and can be reconfigured on-chip. It is worth noting that, the *'stochastic bit'* consumes only leakage power during the pre-charge mode, and charging/discharging power for nodes A and B during the evaluation mode. In addition, the speed of operation is based on the speed of 'EN' signal. Therefore, the proposed design becomes more power efficient and faster as CMOS process scales. Also, the PMOS and NMOS sizing, and bit-precision for the respective codes can be tuned based on the application requirements.

3.3.2 Stochastic Neuron (sNeuron)

We now describe how the *'stochastic bit'* is used to realize stochastic input and output neurons forming the sBSNN. The input neurons map the image pixel intensities to spike trains, where each neuron fires probabilistically at a rate proportional to the corresponding pixel intensity. The *'stochastic bit'* can inherently realize an input sNeuron by mapping the pixel intensity to PMOS code that controls its switching probability. On the contrary, the *'stochastic bit'* is interfaced with counter and modulator circuit (shown in Fig. 3.3(a)), which generates and modulates the weighted input, for realizing the output sNeuron that spikes with the desired probability. Also, the spiking activity of the sNeuron can be suppressed by masking the 'EN' signal of the *'stochastic bit'*, which is used for implementing lateral inhibition that facilitates competitive learning as will be explained in section 3.3.4. The generated spikes from the input and the output sNeuron (PRE and POST) are applied to the stochastic binary synapses for synaptic updates as explained below.

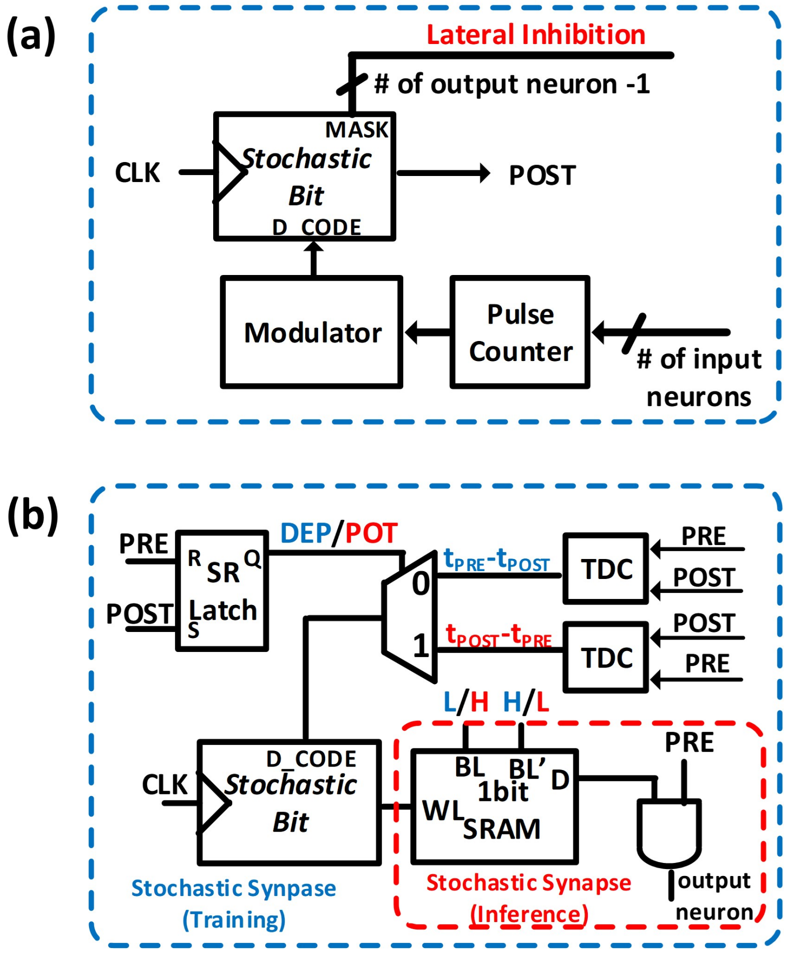


Fig. 3.3. Design of 'Stochastic bit' enabled (a) spiking neuron, and (b) binary synapse (stochastic during training and deterministic during inference).

3.3.3 Stochastic Binary Synapse

The stochastic binary synapse (during training) is realized by interfacing the *'stochastic bit'* with 6T SRAM as depicted in Fig. 3.3(b). Based on the sign of the spike timing difference, $t_{post} - t_{pre}$, the synaptic weight update event is determined as potentiation (depression) when the sign is positive (negative). Then, the spike timing difference, measured as the number of clock pulses using time to digital converter (TDC), feeds the *'stochastic bit'* to selectively turn on the PMOS header transistors, effectively causing it to produce an output pulse with the appropriate probability depending on spike timing. Note that TDC can be realized using a counter for potentiation (depression) that resets when PRE (POST) is high. TDC is shared by stochastic synapses that are activated by the same PRE/POST signal. The generated pulse activates the wordline of the 6T SRAM cell while the bitline is driven to VDD (ground) for synaptic potentiation (depression) update. Once the stochastic training process is complete, the *'stochastic bit'* is powered off and the learnt binary weight stored in the corresponding SRAM cell is deterministically used during inference as shown in Fig. 3.3(b). Note that, during both training and inference, the computation of the weighted input sum reduces to AND operations followed by pulse count since both the inputs and synaptic weights are binary. Hence, the sBSNN provides much higher computational energy efficiency relative to analog neural networks with real-valued (32-bit) inputs and synaptic weights, which require MAC (multiply-and-accumulate) units, and SNNs with real-valued weights and binary inputs that need accumulators for computing the weighted input sum.

3.3.4 sBSNN System-level Implementation

On-chip training: We demonstrate the efficacy of the proposed sNeuron and synapse using a two-layer fully-connected sBSNN depicted in Fig. 3.4. Fig. 3.5 illustrates the system-level implementation of the two-layer sBSNN. The input sNeurons representing the image pixels are fully-connected via binary weights to output (post) sNeurons. At every time-step, the weighted sum of the input spikes with the synaptic weights are modulated and fed to the *'stochastic bit'* in the respective post-neurons, causing them to fire probabilistically. The weighted sum received by each post-neuron is calculated by counting the number of pulses from the output of the AND gates in the corresponding column of synapses as depicted in Fig. 3.5. The pulses are only generated when both inputs of the AND gate are high. Accordingly, power is only dissipated when there are transitions in the AND gate. As a result, the weighted input sum computation in sBSNN consumes significantly lower power compared to full precision (32-bit) SNN. In the event of a post-spike, the spiking neuron inhibits the remaining post-neurons, as illustrated in Fig.3.4, by masking the respective enable (EN) inputs as explained in section 3.3.2 to uniquely learn the presented pattern. The synapses connecting the input to the spiking post-neuron are probabilistically potentiated based on spike timing. The spike timing difference, $t_{post} - t_{pre}$ $(t_{pre} - t_{post})$ in the number of clock pulses, is measured using the POT (DEP) counter shown in Fig. 3.5, which is reset at every pre-spike (post-spike) and decremented by unity at successive time-steps. The elapsed count of POT (DEP) counter is sampled upon a post-spike (pre-spike) for potentiation (depression) weight update. The spike timing difference is fed to the *'stochastic bit'* in the synapses (depicted in Fig. 3.3(b)), which in turn probabilistically programs the SRAM as detailed in section 3.3.3. The sSTDP-based probabilistic weight updates enable each excitatory neuron to learn a complete rep-

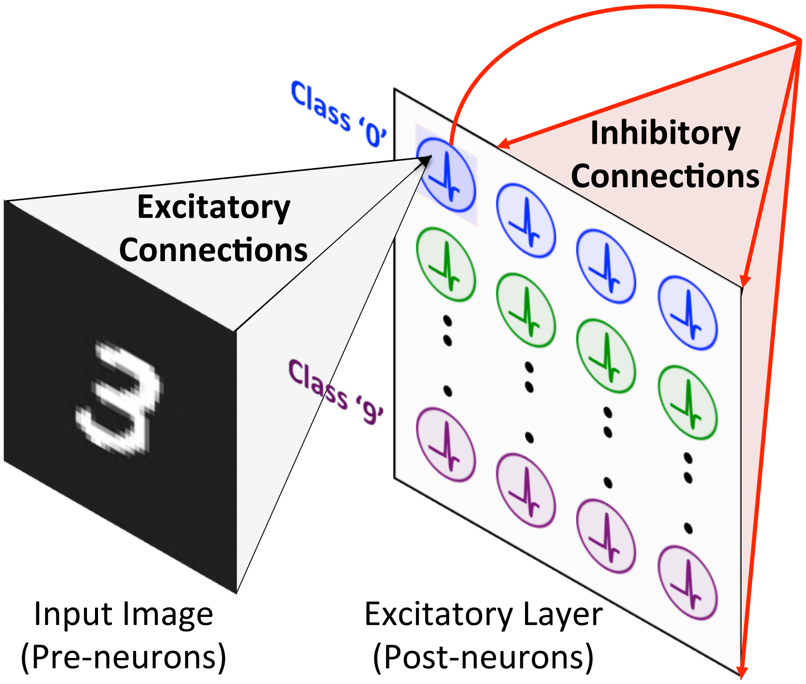


Fig. 3.4. Architecture of two-layer fully-connected sBSNN, with lateral inhibition, for object recognition.

resentation of the input pattern in the input to excitatory synaptic weights. In order to ensure that each excitatory neuron learns unique input representations, we divided the excitatory neurons into different clusters and trained each cluster of neurons on a distinct class of input patterns as proposed in [57]. Fig. 3.6 shows the MNIST

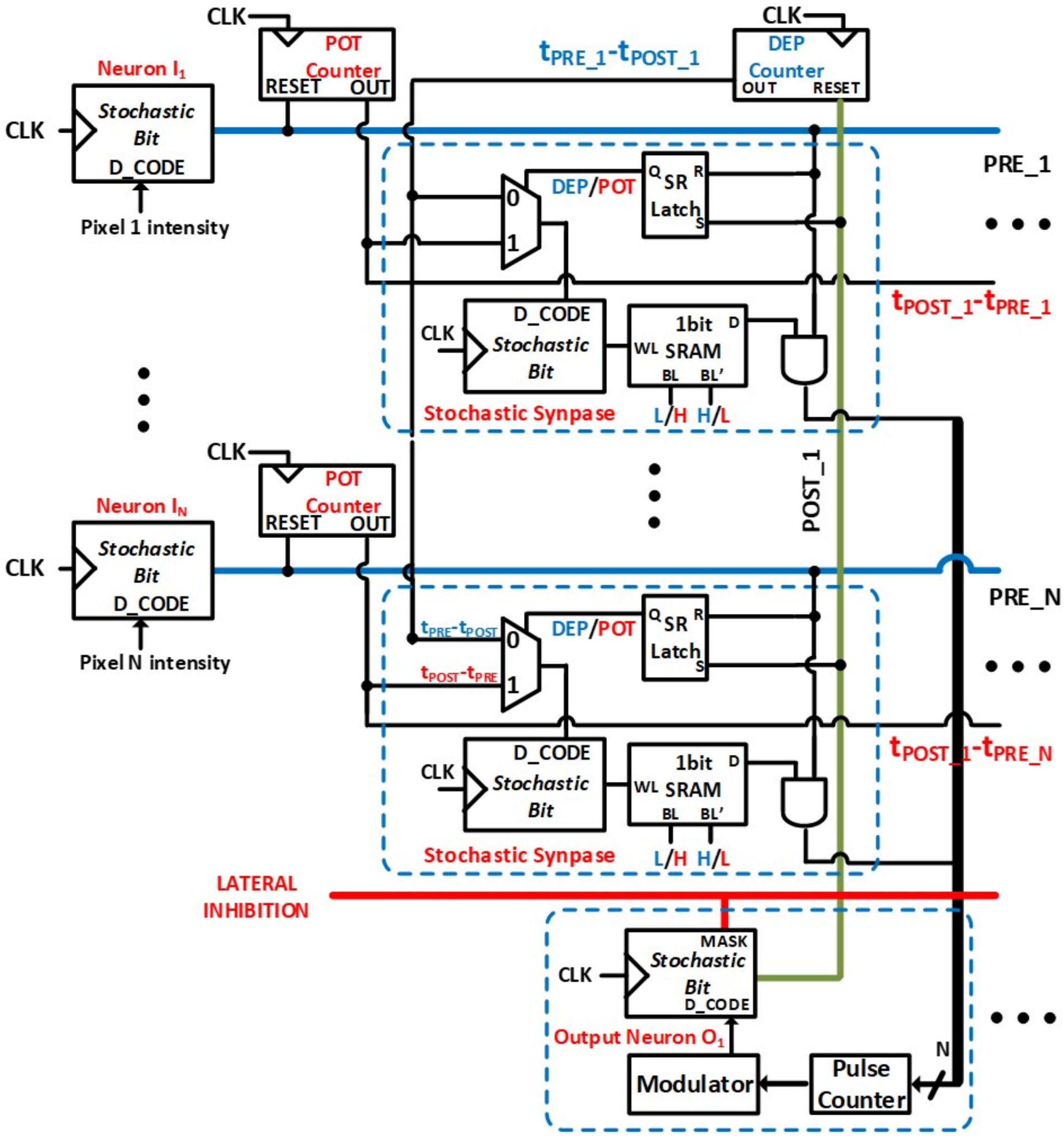


Fig. 3.5. System-level realization of two-layer fully-connected sBSNN with lateral inhibition.

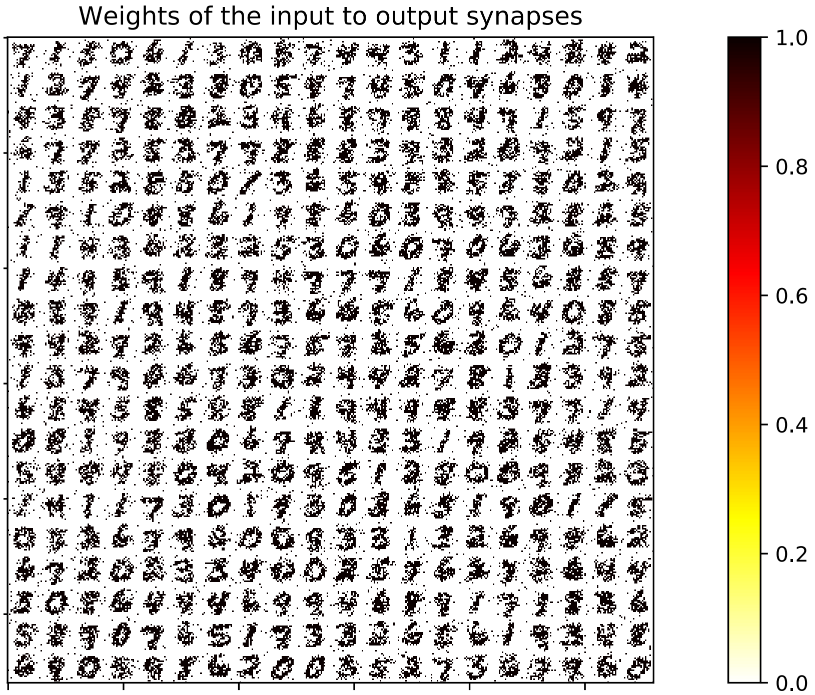


Fig. 3.6. MNIST digit representations (28×28 in dimension) learnt by a two-layer fully-connected sBSNN of 400 excitatory neurons (organized in 20×20 grid).

digit representations learnt by a two-layer fully-connected sBSNN of 400 excitatory neurons using the sSTDP-based training methodology.

On-chip inference: At the end of training, each post-neuron learns to spike for a unique input class by encoding a general input representation in the input to output synaptic weights as shown in Fig. 3.6. Once training is completed, we disable the clock signal of the *'stochastic bit'* in the synapses, thereby fixing the weights for the inference phase. The learnt binary weights, stored in the SRAM cells, are used deterministically during inference. A test pattern is predicted to belong to the class learnt by the group of neurons with the highest average spike count over the time period for which the test input is presented. The proposed sBSNN implementation, by virtue of using simpler weighted input sum computation and state-less stochastic neurons, can provide high energy efficiency during inference as will be shown in section 3.4.

3.4 Results

In this section, we first present the measured results of the sNeuron and synapse, which are fabricated in 90nm CMOS process. We subsequently show the simulation results of our sBSNN implementation (detailed in section 3.3.4) using the measured neuronal and synaptic dynamics on the MNIST dataset.

3.4.1 'Stochastic bit' Characterization

Fig. 3.7(a) illustrates the setup for characterizing the CMOS *'stochastic bit'* design (detailed in section 3.3). The on-chip timing controller generates sufficient number of enable (EN) pulses, which is set to 768 in our experiments, for obtaining reasonable estimate of the *'stochastic bit'* switching probability for a specific configuration of PMOS and NMOS codes. The number of resultant output pulses at OA (refer to Fig. 3.2(a)) is recorded by a 15-bit on-chip counter to determine the switching probability for the chosen PMOS and NMOS codes. For every set of input codes, we performed the switching probability measurement 1000 times. Fig. 3.7(b) shows that the switching probability of the *'stochastic bit'* varies roughly in a sigmoidal manner with the PMOS code. In each box, the central mark shows the median, the ends of the vertical blue boxes indicate the 25th and 75th percentiles, respectively, and the lines indicate the min and max values. The measured switching probability ranges from 11.6% to 90.1% with less than 5% standard deviation at a supply voltage of $1.4V$. In addition, we varied the NMOS code and found that it controls the shape of the switching probability curve as illustrated in Fig. 3.7(c). The variation in the switching probability dynamics with the NMOS code can be attributed to the change in the respective transistor sizes relative to the PMOS transistor sizes. Note that, the ratio of minimum to maximum switching probability is determined by the bit-precision of the PMOS code and the relative sizing (widths) of the PMOS and NMOS transistors, which need to be fixed at design-time based on the application requirements.

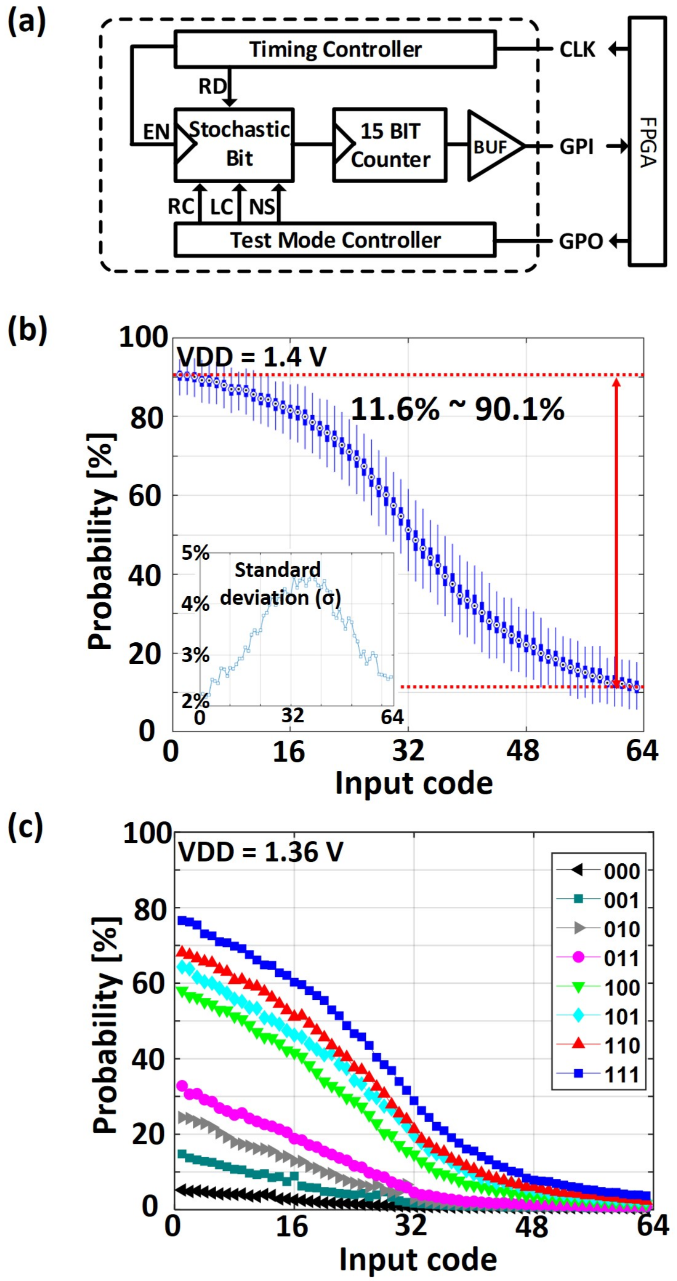


Fig. 3.7. (a) Measurement setup for the *'stochastic bit'* design, which is interfaced with a FPGA board (b) The measured box plots of switching probability versus the input (PMOS) digital code, and its standard deviation, σ (refer to the inset). (c) Switching probability dynamics for different 3-bit NMOS codes.

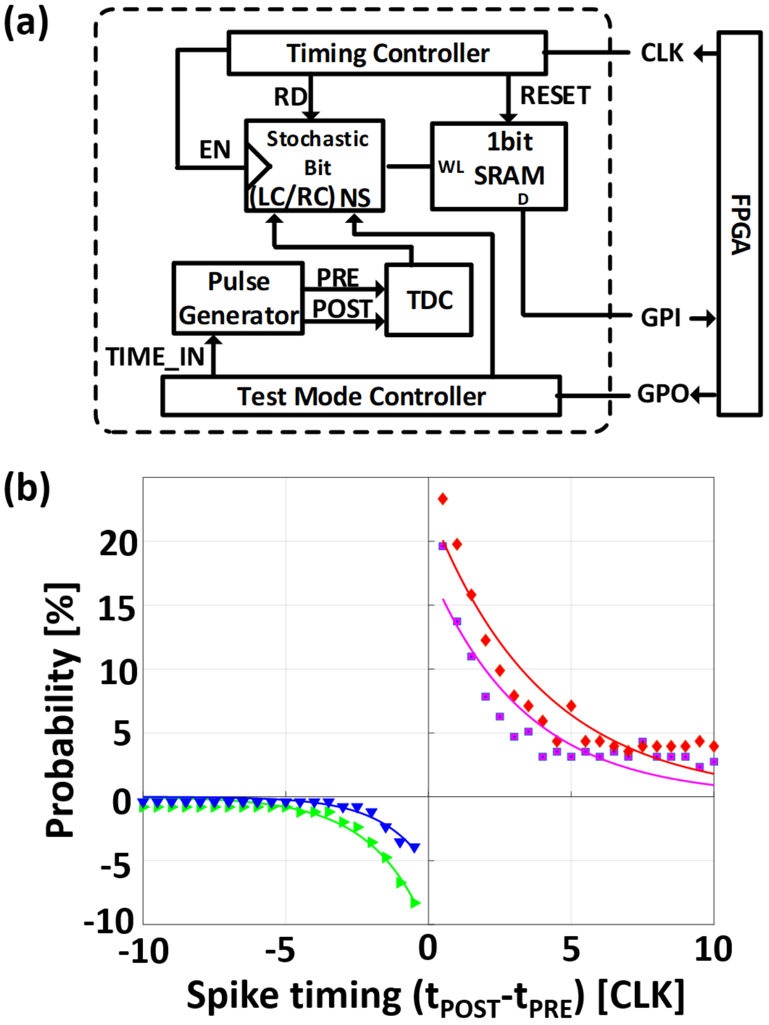


Fig. 3.8. (a) Measurement setup for the sSTDP dynamics needed to train a stochastic binary synapse, which is interfaced with an FPGA board for generating the clock and inputs (spike timing, TIME_IN), and monitoring the outputs (state of SRAM cell). (b) The measured sSTDP curve for different NMOS codes.

3.4.2 Stochastic Binary Synapse

The sSTDP dynamics required for training a binary synaptic weight are obtained by feeding the spike timing difference to the on-chip pulse generator, which generates the pre- and post-spikes as shown in Fig. 3.8(a). The Time-to-Digital Converter

(TDC) measures the spike timing difference and produces the PMOS code for the *'stochastic bit'*, which probabilistically activates the SRAM wordline. The SRAM cell is then probed for potentiation (depression) event to estimate the sSTDP characteristics for the positive (negative) timing window. We adopted a methodology similar to that used for the *'stochastic bit'* characterization for measuring the sSTDP dynamics as explained below. For every value of spike timing within the sSTDP window, TIME_IN in Fig. 3.8(a), we generated sufficient number of enable pulses (set to 768 as explained in section 3.4.1) for the *'stochastic bit'* constituting the binary synapse. We then probed the 6T SRAM for a change in the cell state to determine the corresponding switching probability. We repeated the switching probability measurement 1000 times for every value of spike timing. Fig. 3.8(b) shows the measured sSTDP dynamics, where the synaptic switching probability has roughly exponential dependence on spike timing, which conforms to the sSTDP rule depicted in Fig. 3.1(b). The sSTDP dynamics can be tuned on-chip by programming the NMOS code controlling the footer transistor sizes in the *'stochastic bit'* as explained in section 3.3.1. Note that the Time-to-Digital Converter (TDC in Fig. 3.8(a)) and pulse generators are used only for measurements. The binary synapse is composed of only the 6T-SRAM and the *'stochastic bit'* during training, where the pre- and post-spikes are generated by the input and output sNeurons, respectively, constituting the sBSNN. Also, the spike timing difference is estimated using a counter per pre-/post-neuron as described in section 3.3.4.

3.4.3 sBSNN for MNIST Digit Recognition

The sBSNN implementation was functionally trained and evaluated using the measured neuronal and synaptic dynamics shown in Figs. 3.7(b) and 3.8(b), respectively, on the MNIST digit recognition dataset. The accuracy on the test dataset is 65.88% for an SNN of 400 excitatory neurons trained on 900 MNIST digit patterns, which was sufficient for all the neurons to learn general input representations as depicted

Table 3.1.
Comparison with related works.

	This Work	**2016 VLSI [66]**	**2015 IEDM [67]**	**2017 VLSI Report [68]**	**2015 TCAS II [69]**
Learning Rules	Stochastic STDP	Stochastic STDP	STDP	Modulated STDP [70]	STDP
STDP Timing Window	267ns (10 time steps) @37.5MHz	10ms	100us	N/A	3.5us
Stochastic deviation	<5%	N/A	N/A	N/A	N/A
On-chip reconfigurable	YES	N/A	YES	YES	YES
Energy /spike/neuron	8.4pJ* /1.84pJ**	N/A	N/A	11.9 μW***	9.3pJ /3.6pJ***
System Configuration	Stochastic Neuron/Synapse	RRAM Synapse IF neuron	PCM Synapse LIF neuron	Stochastic Neuron/Synapse	RRAM Synapse IF neuron
Accuracy	92.30% ($784 \times 400 \times 10$) Trained on 60k MNIST digits	86% (784×10) Trained on 50k MNIST digits	N/A	$\approx$ 88% ($784 \times 500 \times 10$) Trained on 50k MNIST digits	N/A
Technology	90nm	Non-CMOS	Non-CMOS	Non-CMOS	180nm

* Measured power: ‘stochastic bit’ + 15b counter + etc. = 226μA*1.4V*26.7ns = 8.4pJ

** Estimated neuron power: 226μA * (33.3/153.2)*1.4V*26.7ns = 1.84pJ
(Post-layout simulated current: 153.2μ = ‘stochastic bit’[33.3μ]+ others[119.9μ])

*** Peak power with a single spike duration of $\approx$ 10 μs

**** Normalized power [71] from 180nm to 90nm: 9.3pJ *(90/180)*1.4/1.8=3.61pJ in Fig. 3.6. Any more increase in the number of training patterns could deteriorate the learnt representations, leading to further loss in accuracy. The accuracy can be improved by increasing the number of excitatory neurons and/or by incorporating an additional fully-connected classification layer trained on a larger fraction of the dataset. We augmented the SNN with a softmax readout layer of 10 neurons corresponding to the 10 classes in the MNIST handwritten digit recognition task, where each readout neuron is fully-connected to all the excitatory neurons. For a given input pattern, the spike count of the excitatory neurons are estimated using the sSTDP trained sBSNN, and subsequently fed to the softmax readout layer, which predicts the test pattern to belong to the category represented by the readout neuron with the highest activation. We trained the readout layer on the entire training dataset using the Adam optimizer [72], which is a popular gradient-based supervised training algorithm, and cross-entropy loss function with learning rate of 0.001 for 8 epochs. We obtained higher accuracy of 92.30% on the entire MNIST test dataset of 10,000 images.

sBSNN offers possibility of up to 32× neuronal and synaptic memory compression relative to similarly sized full precision (32-bit) SNN with accuracy loss that can be minimized for larger SNNs. The energy of the sNeuron with the measurement blocks (refer to the sNeuron measurement setup in Fig. 3.7) is measured to be 8.4pJ/spike. The standalone neuronal energy is estimated to be 1.84pJ/spike as detailed in Table 4.1. In addition, Table 4.1 also indicates that the proposed implementation offers lower neuronal energy consumption compared to related works in 90nm CMOS process.

3.4.4 Energy efficiency

Finally, we estimate the energy efficiency of the two-layer sBSNN implementation composed of 784 input and 400 output sNeurons in terms of Tera-operations (TOPS) per Watt. Our functional simulations indicated that the average number of transitions

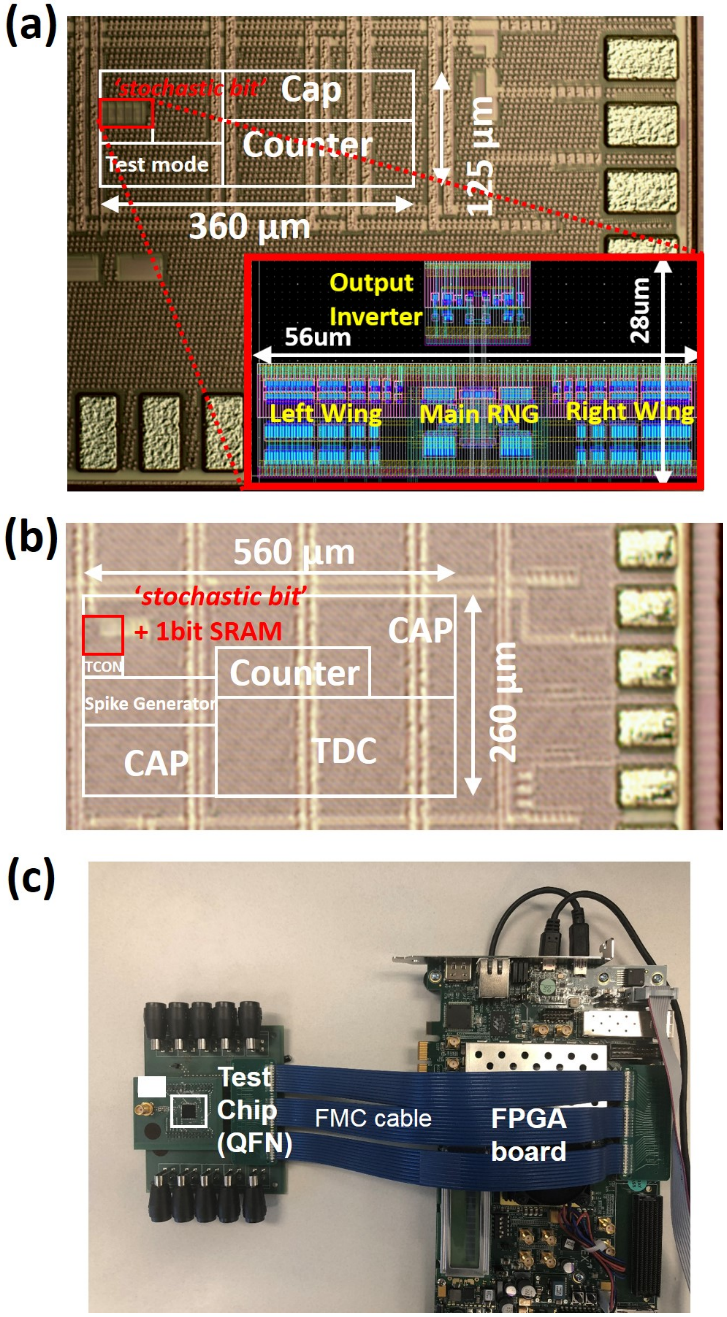


Fig. 3.9. (a) Die photo of the *'stochastic bit'* and its layout (refer to the inset). (b) Die photo of the stochastic binary synapse composed of the *'stochastic bit'* and 6T-SRAM bitcell. (c) Test chip measurement setup using FPGA.

in the AND gate of stochastic synapses is ∼700 out of 784×400 total possible transitions. The average power consumed by the AND gate per transition in 90nm CMOS process is estimated as $0.80\mu W$, which totals to 0.56mW per time step. Every output sNeuron requires a 10-bit ones counter for accumulating the maximum weighted input sum of 784, and the *'stochastic bit'* to spike probabilistically. The average weighted input sum received by the output sNeurons is functionally determined to be 21. The average power consumed by the 10-bit ones counter is estimated to be 0.558mW per sNeuron while that of the *'stochastic bit'* is measured to be 0.033mW per sNeuron. The total output neuronal power is 236mW (0.558mW+0.033mW ×400) while that of the input neurons is 25.87mW (0.033mW×784). The proposed implementation performs 23.52 TOPS (784×400×2×37.5MHz) while consuming 262.8mW, leading to energy efficiency of 89.49TOPS/Watt. The high energy efficiency can be attributed to binary dot product computations and the inherent sparsity in the neuronal spiking activity offered by SNNs. Figs. 3.9(a)-(b) show the die shot of the sNeuron, synapse, and the layout of the *'stochastic bit'* core (inset of Fig. 3.9(a)). For measurements, we interfaced an FPGA to the QFN packaged chip on a custom PCB as depicted in Fig. 3.9(c).

3.4.5 Process and temperature variation

Fig. 3.10(a) shows the simulated switching probability curves affected by process and temperature variations. The black solid line represents the baseline of our design and the other lines represent variations caused by the different combinations of process corners (FF, TT, SS, FS, SF) and temperatures (-55°C, 27°C, 125°C). The (SS, -55°C) corner shows less than 10% change in probability due to decreased temperature and current, decreasing noise or the source of the randomness. The variations can be easily compensated by having variable size of M1 and M2 transistors of Fig. 3.2(b) in the same way we size M3 and M4 transistors. The size ratio between M1/M2 transistors and M_{l1x}/M_{r1x} transistors determines the unit step change of probability

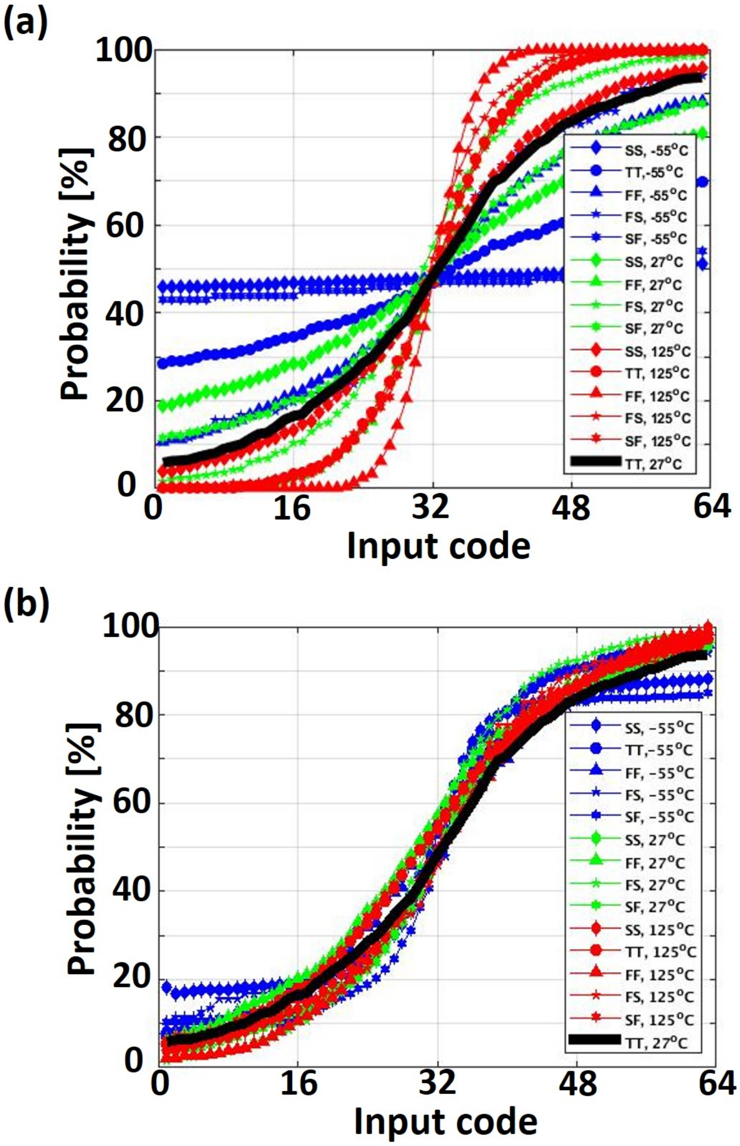


Fig. 3.10. (a) The switching probability curves with process (FF, TT, SS, FS, SF) and temperatures (-55 $°C$, 27 $°C$, 125 $°C$) variations. (b) The compensated switching probability curves for all corners presented in (a).

and thus, the slope of the probability curve. Fig. 3.10(b) shows the compensated switching probability curves from all corners presented in Fig. 3.10(a). In addition to the variation compensation, this approach also allows us to control the shape and slope of the probability curve at the cost of area required for sizing M1 and M2 transistors. Further, the probability range can also be controlled through the NMOS codes applied to M3 and M4 transistors as shown in Fig. 3.7(c).

3.5 Conclusions

In this chapter, we proposed stochastic binary SNN that requires only one-bit precision for the constituting neurons and synapses for memory-compressed neuromorphic computing. We presented an energy-efficient implementation of the binary SNN using Biased Random Number Generated (BRNG) as 'stochastic bit' to realize the stochastic neurons and synapses (during training) fabricated in 90nm CMOS process. We demonstrated high energy efficiency of 89.49 TOPS/Watt for two-layer SNN, which renders the proposed realization amenable for IoT/edge devices with on-chip intelligence.

4. COUPLED SPIN-TORQUE-OSCILLATOR BASED DISTANCE COMPUTATION: APPLICATION TO IMAGE PROCESSING

Recent research on nano-oscillators has shown the possibility of using coupled oscillator network as a core computing primitive for non-Boolean computation. The spin-torque oscillator (STO) is an attractive candidate because it is CMOS compatible, highly integrable, scalable, and frequency/phase tunable. Based on these promising features, we propose a new coupled-oscillator based architecture for hybrid spintronic/CMOS hardware that computes multi-dimensional norm. The hybrid system, composed of an array of four injection-locked STOs and a CMOS detector, is experimentally demonstrated. The measured performance is then used as input to simulations that demonstrate the hybrid system as both a distance metric and a convolution computational primitive for image processing applications. Energy and scaling analysis shows that the STO-based coupled oscillatory system has higher efficiency than the CMOS-based system with an order of magnitude faster computation speed in distance computation for high dimensional input vectors.

4.1 Introduction

Distance computation between multi-dimensional vectors is used in numerous applications, particularly for data and workload intensive problems such as combinatorial optimization, recognition, and classification. In order to process massive amount of data effectively in real time, it is desirable to realize an energy efficient hardware for the distance computation that utilizes parallelism. The computation of the Euclidean distance (L2 norm) requires expensive operations in hardware for squaring compared to that of the Manhattan distance (L1 norm) [73, 74]. Since it is inefficient to implement such expensive operators in digital CMOS circuits, analog computation based approaches that use device physics to directly compute complex functions such as squaring [75–79]. Such analog computations obtain better energy efficacy at the cost of tolerable errors, and is beneficial for applications where an approximate result is sufficient instead of exact result. The paradigm of 'let physics do the computing' has motivated researchers to look at "alternative computing models" that explore the use of non-CMOS devices as functional units for better energy efficiency and speed. One of those alternative models is based on the coupled oscillator network in which the oscillator array is used to compute (say) "similarity" between two multi-dimensional vectors. The similarity can be defined in terms of the distance between the two vectors. [80–87].

Such coupled oscillatory networks are widely found in nature such as pendulum clocks on a wall [3], flashing fireflies [4], animal flocking [5], coupled oscillations in the human heart and brain [6, 7]. Inspired by such systems, researchers have proposed coupled oscillator systems to solve image or pattern recognition problems [81, 84, 88–93] in a "preferred way of nature". With the development of emerging oscillators such as Spin-Torque Oscillator (STO) and vanadium dioxide (VO_2), recent research has demonstrated fabricated oscillators [94–97]. However, they use external capacitors or bonding wires for coupling, making it difficult to couple large number of oscillators or to build high-density networks.

Nanoscale oscillators such as spin-torque oscillators (STOs) STOs are attractive for implementing the large number of coupled oscillator network for computation because they provide potential scalability of the functional units to smaller dimensions, along with faster computation time and less energy consumption compared to standard digital or analog CMOS implementations [75–77, 79, 86, 94, 98, 99]. In this work, we focus our attention on STO-based coupled oscillatory system for approximate Euclidean distance (ED) computation. Such a hybrid nano-oscillatory system comes with multiple challenges that must be overcome to be practically implemented and adopted. First, the system has to be CMOS compatible (in terms of the fabrication processes, operating currents and voltages) and scalable. Second, the system should perform computations in parallel for better energy efficiency, taking advantage of the properties of nano-oscillators. STOs are back-end process compatible with CMOS, and their oscillating amplitudes are also CMOS compatible. Moreover, STOs offer tunable frequency and phase, and generate microwave RF oscillating signals that enable fast computation. These features motivated us to explore STOs with the injection locking scheme for the coupling as the primitive for a distance computing architecture. We experimentally demonstrate a system composed of giant magnetoresistance (GMR)-based STO devices [100–102] and a CMOS detector as the core computing primitive for distance computation. In addition, we theoretically show that phase dynamics of the system inherently introduces non-linearity that makes the system appropriate for measuring the L2 distance between two multi-dimensional vectors. The hybrid system is used to measure the distance between two input vectors whose output follows the $L2^2$ norm. Our experimental results on 4 coupled system along with CMOS peripherals for distance measurement is used to parameterize larger-scale simulations of coupled oscillatory system. The hardware for $L2^2$ distance calculation is also used to compute approximate convolution, which in turn has been used for an edge detection task.

4.2 Coupled Spin Torque Oscillator array

Before we explain how the proposed system can be used to compute the ED between two multi-dimensional input vectors, it is important to explore the device characteristics of STOs, the core computation unit of our injection-locked coupled oscillator array. STOs are compact RF oscillators based on magnetic spin valves that consist of a fixed and a free magnetic layer separated by an insulating spacer layer. The sustained magnetization precessions of the free layer is induced by injecting DC current through the device and the resultant oscillating resistance leads to an alternating voltage across the device. The precession frequency changes as the charge current through the oscillators changes. Therefore, STOs work as current controlled oscillators.

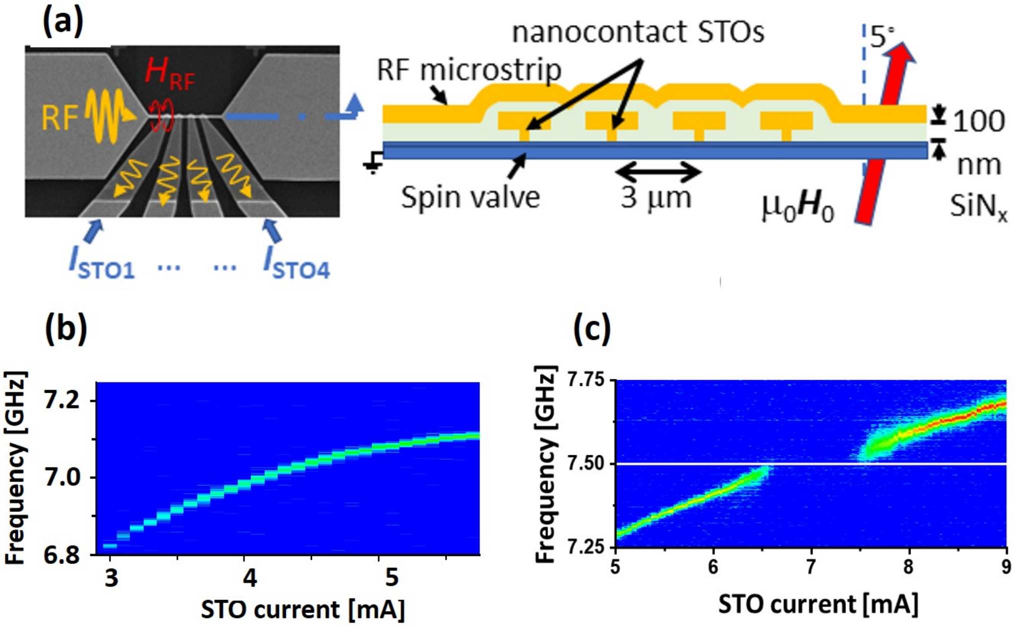


Fig. 4.1. (a) schematic of RF signal conditioning board for the spin torque oscillator array. (b) Free-running response from single device. (c) Injection locking response of similar device.

The device array is shown in Fig. 4.1(a). The proposed STO devices are giant magneto-resistive (GMR) nano-contact devices consisting of a CoFe reference layer and CoFe/Ni multilayered free layer (see Methods section for fabrication details), chosen for the simplicity of their magnetic structure that minimizes variations due to patterning. To operate in the target band of 5 GHz to 10 GHz, the STOs are contacted by tapered coplanar waveguide (CPW) lines connected to pads at the edge of the chip. To ensure that each STO receives an injection signal of similar amplitude and phase, the STOs are patterned 3μm apart. Scanning electron micrograph of RF microstrip and CPW device lines shown in upper left. Red lines are stitched grounded CPW lines (20 GHz bandwidths).

A DC current flowing through the nano-contact produces an anti-damping torque on the free layer, causing the free layer to precess about the net effective field. Increasing the current changes the net effective field (primarily by changing the demagnetizing field via the cone of precession), thereby changing the precession frequency [103]. A typical free-running individual device spectral response is shown in Fig. 4.1(b). As seen in Fig. 4.1(c), the device frequency pulls to the injected signal and phase locks. The degree of locking—that is, the fraction of the time the device is phase-stable relative to the injected signal—increases as the STO frequency approaches the injection frequency. By measuring the amount of power outside of the injection signal and comparing it to the free running power, we can estimate the degree of locking of the device. For the devices in the measured array, we restrict our operation to regions where the estimated degree of locking is greater than 95%.

The injection signal is delivered to the STOs via a 1 μm wide microstrip patterned over the devices (see Fig. 4.1(a)), producing an ac magnetic field at the devices of approximately 0.4 mT for the data presented here. This ac field is transverse to the 0.379 T field applied at 5° to the surface normal, which is to place the oscillation frequency near 7 GHz. Both the amplitude and the phase of an injection locked oscillator change across the locking range. For STOs, the phase has been shown to shift from -$\pi/2$ at the $f_{osc} < f_{inj}$ range, to $+\pi/2$ for $f_{osc} > f_{inj}$ range. [104] Thus, the shape of the response curve across the locking range is adjusted by changing the amplitude and phase of the net reference.

Fig. 4.2(a) shows the signal at $f = f_{inj}$ across the locking range for a single oscillator for different values of the reference phase. The bottom panel of Fig. 4.2(b) shows the calculated response for those phase angles, based on a STO model whose amplitude and phase at f_{inj} vary as shown in the top panel. We choose a reference phase and amplitude shift such that the curve has a maximum near the center of the locking range.

Based on these observed features, STOs can be used to encode input information as the frequency in the free-running mode or as the phase in the injection-locking

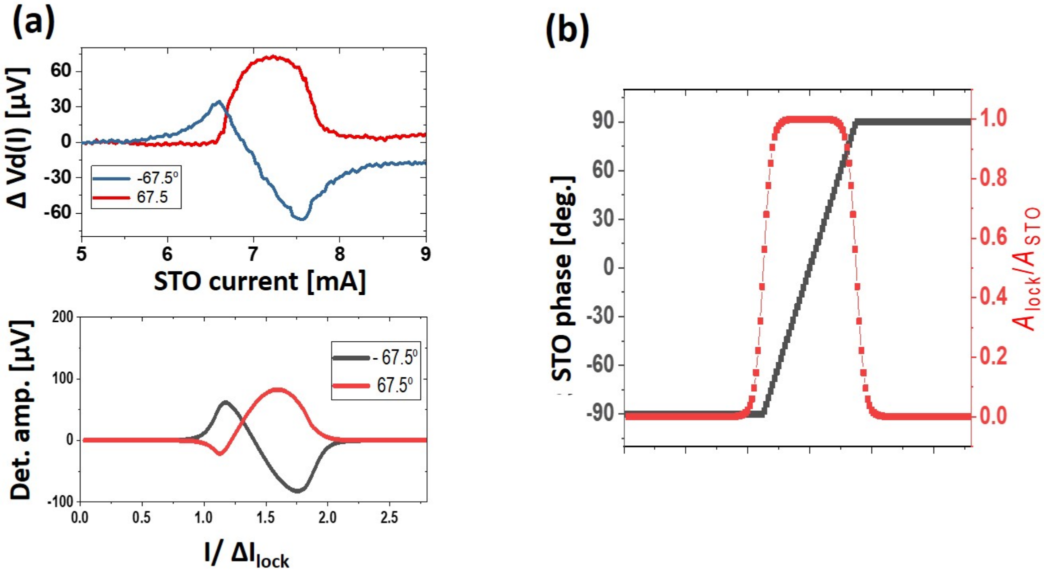


Fig. 4.2. (a) Voltage from microwave diode across locking range for two different values of microwave phase reference. (b) *Top panel:* Inputs for amplitude and phase of STO signal for Phasor model below. *Bottom panel:* Phasor model of expect signal at $f = f_{inj}$, for different values of reference phase.

mode. The latter is the case and STOs are used as current to phase converters. The relative phase of STOs to the injected signal represents input information mapped to the bias current of STOs. Note that the injected signal plays another significant role in addition to enabling the coupling of oscillators in our distance computing system. The injected signal (referred as the reference signal hereafter) also provides better approximation of the ED computing as explained in the next section.

4.3 Coupled Oscillator-based Distance Computation System

In this section, we first describe the functional configuration of the oscillator-based computing system, and explain how our system computes the distance between two input vectors by exploiting the device characteristics described above. We analyze the system with the derived equations to show the relationship between the input phase information and the corresponding output signal amplitude. Finally, the impact of noise from the device is considered for the operation of the system.

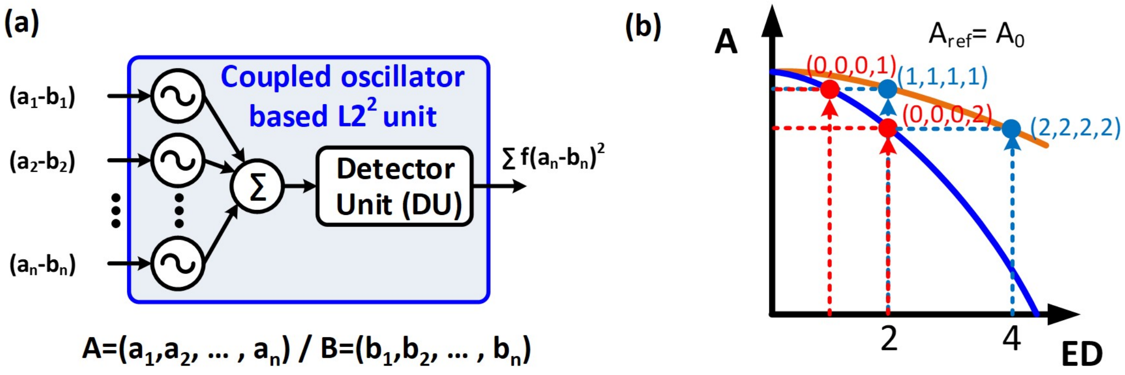


Fig. 4.3. (a) Block diagrams of coupled oscillator based L2^2 unit for a distance computing primitive (b) The cases showing the same A with different ED when the amplitude of reference signal is same as that of the STO signal.

In our system, the coupled STOs are injection-locked in frequency by an AC magnetic field. Locked STOs emit the same frequency as the injection signal but can have different phases depending on their input currents. We choose the input currents to be proportional to the difference between the two input vectors, such that the coupled spin-torque oscillator network maps input information into phases of the oscillatory signals produced by the GMR-STOs. As depicted in 4.3(a), each injection-locked oscillator is biased with a current corresponding to an element-wise difference between two vectors of an input. The output signals of the STOs are merged through a summing element (denoted as $\sum$ in Fig. 4.3(a)) before presenting to the Detector Unit. Different techniques such as resistive coupling [105] and capacitive coupling [84] have been used to sum incoming signals from the oscillators. In our implementation (described in the next section) we use Wilkinson coherent power combiners to avoid additional phase and amplitude offsets. The output signal from the summing unit exhibits different amplitude as a function of the relative phases to the reference signal.

The Detector Unit measures this amplitude, and returns a digitized code that is proportional to the squared ED between input vectors as described below.

We analyze how the oscillator network converts input information into the amplitude of the combined signal and how the non-linear distance metric can be obtained. Free-running STOs are frequency-tunable with current. When injection-locked, STOs emit a constant frequency as a function of current, but change phase monotonically across the locking range as explained in the previous section [104,106,107]. Here, the combined signal of the oscillator array can be expressed as $\sum_{i=1}^{n} A_i cos(\omega t + \delta_{ref} + \delta_i)$, where ω is the reference signal frequency, A_i is the amplitude of STO i at ω, n is the number of STOs (*i.e.*, the dimension of the input vector), δ_{ref} is the phase of the reference signal, and δ_i is the phase relative to that of the reference signal. Using harmonic addition theorem [108], this signal can be rewritten as $Acos(\omega t + \delta)$, where the amplitude and phase are defined as

$$A^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} A_i A_j cos\,(\delta_i - \delta_j) \tag{4.1}$$

$$\delta = tan^{-1} \frac{\sum_{i=1}^{n} A_i sin\delta_i}{\sum_{i=1}^{n} A_i cos\delta_i} \tag{4.2}$$

In addition to the signals of n oscillators, we have the injected reference signal that has the frequency of ω and provides the reference phase, δ_{ref}, to the oscillators. Therefore, we have the amplitude of the combined signal as

$$A^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} A_i A_j cos\,(\delta_i - \delta_j) + 2A_{ref} \sum_{j=1}^{n} A_j cos\,(\delta_j) + A_{ref}^2 \tag{4.3}$$

We can expand equation (4.3) with the approximation of $cos\theta \approx 1 - \theta^2/2$, where the approximation error is less than 5% when $|\theta|$ is less than $\pi/3$. The amplitude of the combined signal can be expressed as (see Supplementary Information for details)

$$A^2 = (A_0 n + A_{ref})^2 - ((n-1)A_0^2 + A_{ref}A_0) \sum_{i=1}^{n} \delta_i^2 + A_0^2((\sum_{i=1}^{n} \delta_i)^2 - \sum_{i=1}^{n} \delta_i^2) \tag{4.4}$$

where A_0 is the amplitude of n oscillators. The first term of the equation (4.4) is constant with respect to δ, and the second term is proportional to the ED square of n-dimensional vector $[\delta_1, \delta_2, ..., \delta_n]$. We call the third term as the error term since we can obtain the approximated ED if the error term is small enough. The maximum of the error can be derived by using Chebyshev's sum inequality, $(\sum_{i=1}^{n} \delta_i)^2 \leq n(\sum_{i=1}^{n} \delta_i^2)$. The error term can be expressed as

$$A_0^2((\sum_{i=1}^{n} \delta_i)^2 - \sum_{i=1}^{n} \delta_i^2) \leq A_0^2(n-1)(\sum_{i=1}^{n} \delta_i^2) \tag{4.5}$$

where the equality holds if all δ_is are identical. Therefore, the error is maximized when the phases of n oscillators are the same. Here, the role of the reference signal becomes important to reduce the contribution of the error term to A. For example, let us assume that our system has 4 STOs and 1 reference signal, where the amplitude of the reference signal is the same as that of STOs ($A_0 = A_{ref}$). Under this condition, the reference signal works as another single STO whose phase is δ_{ref} or '0' relative phase. To exhibit the combination of the input vectors into a single axis, ED has been used as an x-axis component. For instance, once the phase of the four input signals are expressed as [1,1,1,1], then the x-axis component becomes $\sqrt{(1^2+1^2+1^2+1^2)}$ = 2. Fig. 4.3(b) shows the relative phase information for 4 STOs, which can be represented as [0, 0, 0,1] and [1, 1, 1, 1], where the ED of the two cases are 1 and 2, respectively. The two cases can be considered as 5 STOs with phase information [0, 0, 0, 0, 1] and [0, 1, 1, 1, 1]. Therefore, from equation (4.3), we see that both of these cases result in the same amplitude of the combined signals, although their ED is different. A strong reference signal resolves this problem by increasing the contribution of the second term in equation (4.4), thereby reducing the error term.

Fig. 4.4 shows the normalized amplitude of the combined signal from n oscillators (n=4, 25, and 100) compared to the expected L2 output. The experiment considers an additional reference signal of phase equal to the phase of the injection signal. The output obtained when the n-oscillators have n random phases is shown in blue points,

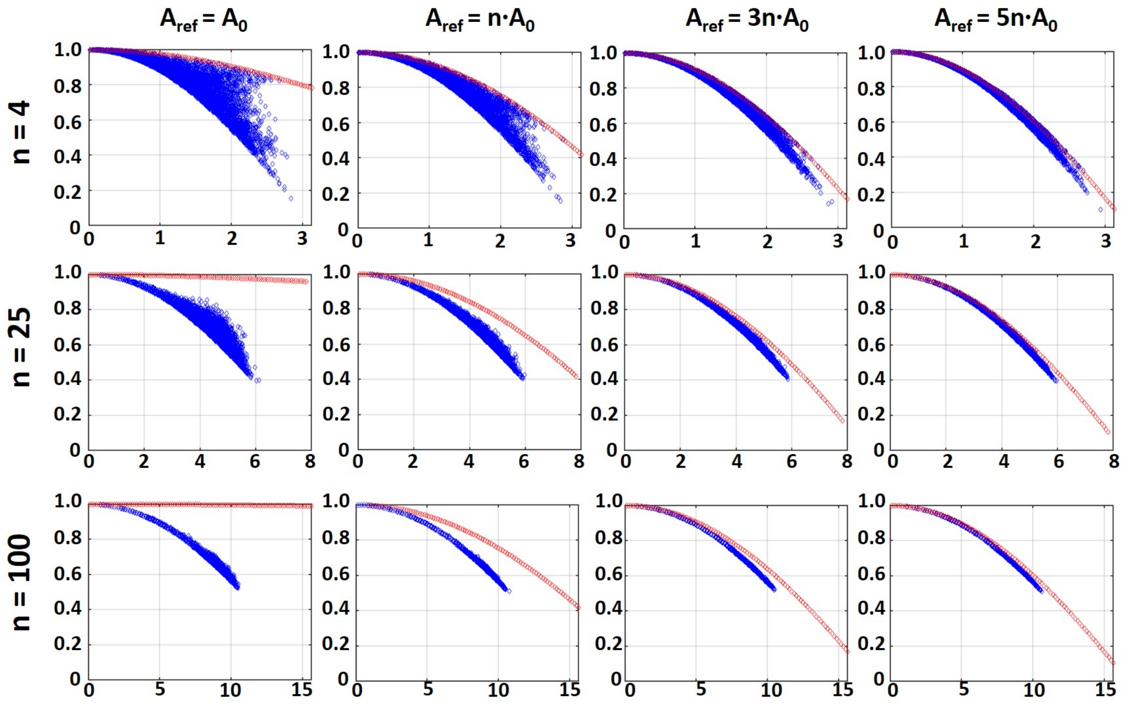


Fig. 4.4. The normalized amplitude, A, versus ED for n = 4, 25, and 100 with different $A_{ref} = A_0$, nA_0, $3nA_0$, and $5nA_0$.

whereas the red points show the output when the n-oscillators have equal phase. For a given L2, the amplitudes of all blue points are less than the amplitude of red point, showing the aforementioned claim that A has maximum error when the phases of n oscillators are the same. When the reference signal has the same amplitude as the STOs, the system shows a broad, noisy, quadratic dependence on L2. With the aid of a stronger reference signal, the output shows a clear quadratic dependence on L2 by reducing the contribution of the error term as described above. We can also observe that the quadratic dependence on L2 becomes less noisy as n increases for the same strength of the reference signal. This is because the variance of the error term for the case of n independent random variables (δ_i) linearly increases with increasing n, whereas that of the amplitude (A) increases as $\sqrt{n}$. In other words, a clearer quadratic dependence on L2 can be obtained for a fixed strength of the reference signal as the number of dimension increases if the input phases are randomly distributed in the locking range.

The signals from the STO devices have small amplitude (≈ -59 dBm, see section 4.4). Thus, the amplitude variations in the combined signal from the oscillator array, occurring due to the phase differences between STO signals, are even smaller. Consequently, the amplitude of the combined signal needs to be detected using CMOS circuitry for further processing. The key challenge for the CMOS detector circuit is to differentiate the small amplitude differences in the incoming signal. Rather than measuring the exact amplitude based on the complex analogue circuitry [109, 110], we propose a simple but adjustable integrator that is able to represent the relative amplitude difference of the inputs for a wide range of amplitudes. The integrator only tracks the region of our interest, which is the small amplitude change in the incoming sinusoidal signal around the peak value. Thresholding the signal around its peaks enables the integrator to easily detect the change in signal amplitude.

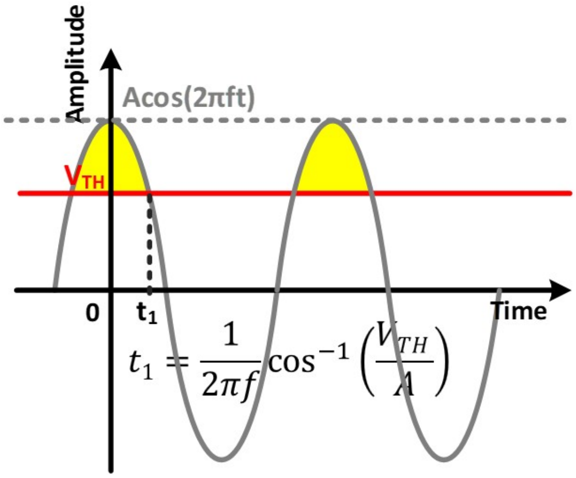

Amplitude
Acos(2πft)
V_{TH}
0
t_1
Time
$t_1 = \frac{1}{2\pi f}\cos^{-1}\left(\frac{V_{TH}}{A}\right)$

Fig. 4.5. The integration of a thresholded sinusoidal signal (The enclosed yellow area is proportional to the output voltage of the integrator).

The integrator outputs a voltage proportional to the definite integral of a thresholded sinusoidal signal, which represents the enclosed area between the threshold level

(V_{TH}) and the oscillatory signal as shown in Fig. 4.5. In a given integration time T_{INT}, the area is a function (f_{int}) of the amplitude of the sinusoidal signal A and V_{TH}, written as

$$f_{int}\left(A\right) \cong \frac{T_{INT}}{\pi} \cdot \sqrt{A^2 - {V_{TH}}^2} \tag{4.6}$$

where T_{INT} is long enough compared to the period of the sinusoidal signal $2\pi/\omega$. Thus, for a ≈7 GHz signal from the STOs, an integration time of a few nanoseconds is sufficient.

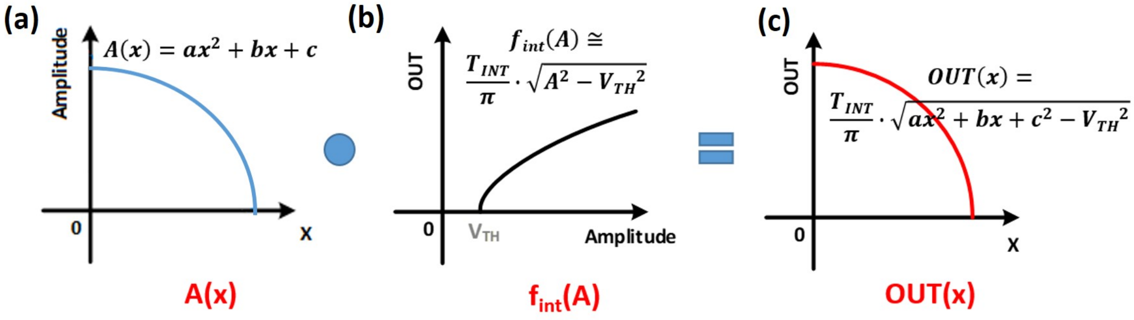


Fig. 4.6. The overall system equations. (a) The amplitude of the combined signal of the oscillator array, A(x). (b) The detector unit output voltage, f_{int}. (c) The system output, OUT(x) is the composition of the two functions (A(x) and f_{int}).

As shown in Fig. 4.6, the overall system can be expressed as equation (4.7) by replacing A in equation (4.6) with an arbitrary second order polynomial, $A = ax^2 + bx + c$, where x is set to the L2 norm:

$$OUT\left(x\right) = f_{int}\left(A\left(x\right)\right) = \frac{T_{INT}}{\pi} \cdot \sqrt{\left(ax^2 + bx + c\right)^2 - {V_{TH}}^2} \tag{4.7}$$

This representation of the integrated value, $OUT(x)$ ($= OUT(L2)$), is a quadratic function of L2, thereby illustrating that the proposed system can perform approximate distance computation (L2^2 norm).

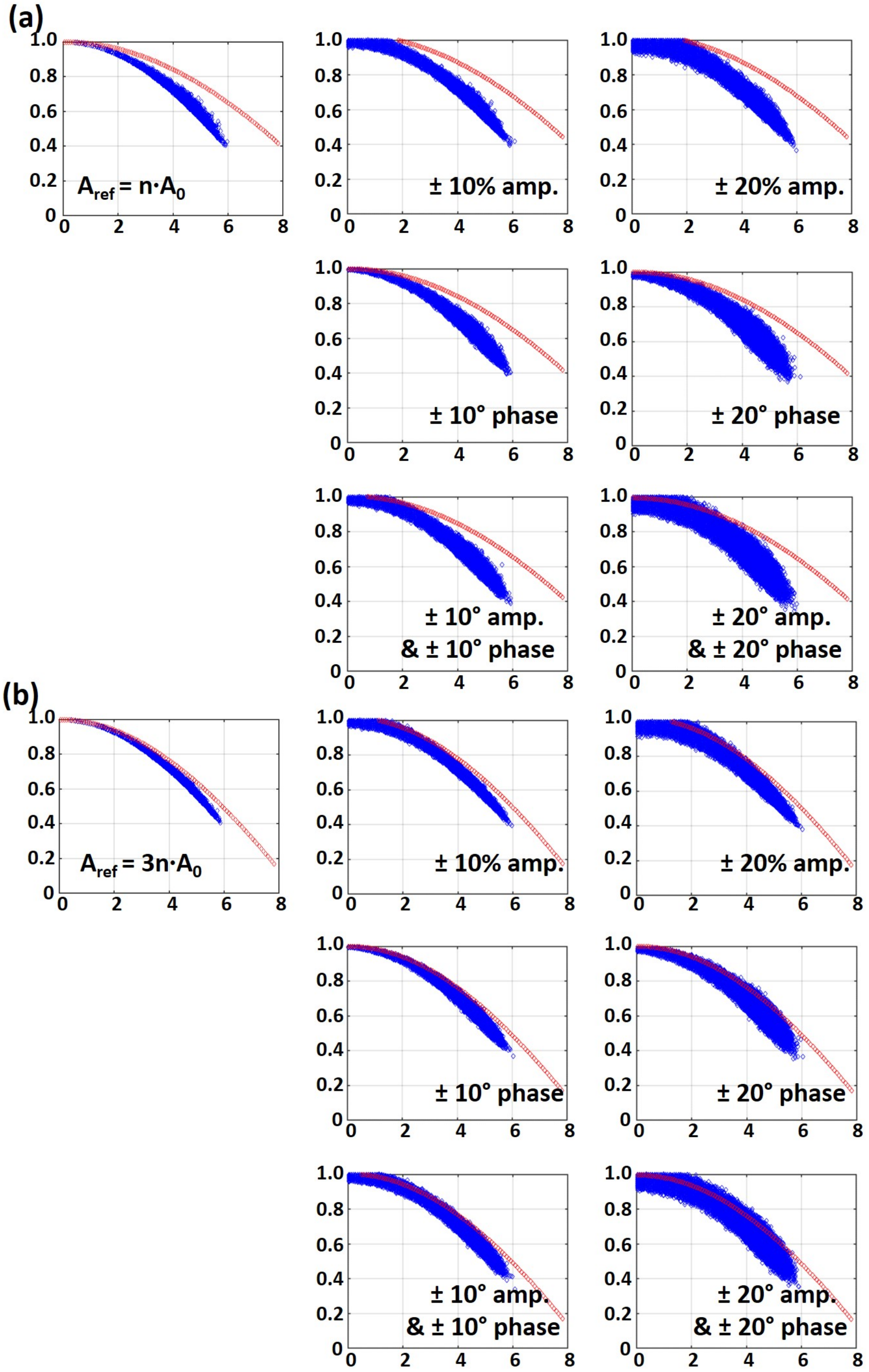


Fig. 4.7. The effect of phase (10°, 20°) and amplitude variations (10%,20 %). (a) $A_{ref} = nA_0$, and (b) $A_{ref} = 3nA_0$ (n = 25).

For real devices, the phase and amplitude of the oscillating signals from STOs deviate from the expected values due to process variations and noise. Based on the measured results [104, 111], the effect of phase (10°, 20°) and amplitude variations (10%, 20%) vs. L2 are shown in Fig. 4.7(a), and (b) for different amplitudes of the reference signal $A_{ref} = nA_0$, and $3nA_0$, respectively. Note that the quadratic dependence on L2 is still maintained even with process variations and device noise.

4.4 Implementation and results

In this section, the implementations of the coupled STO network and CMOS detector are discussed. Fig. 4.8(a) and (b) show a block diagram of the entire hybrid system and its hardware implementation, respectively. For the chosen computation architecture of Fig. 4.3(a), first the STOs must couple to the injection signal, which works as a phase reference. Subsequently, the individual oscillator output signals must be combined with equal phase and amplitude. The STOs are arranged in a bank of 4 oscillators excited by a common microstrip field line. Each STO is contacted by a coplanar waveguide enabling independent current biasing. The injected signal, referred as the reference signal, couples parasitically to the STO output lines ($\approx$ 35 dB isolation), resulting in a signal with amplitude (-31 dBm) much larger than the STO ($\approx$ -59 dBm). This parasitic is at a fixed phase relative to the locked STO signal, and thus coherently mixes with the STO signal at the power detector, providing an effective phase reference. The reference signal enables the detection of this locking curve directly without spectrally resolving the STO output. If the combined signal is sent to the detector, when the STO phase locks, the resulting homodyne signal proportional to $A_{STO} \cdot A_{ref}$ (see equation (4.1)) is much larger than the STO signal itself. For the measurements, in this research, A_{ref} is set to be about $660A_{STO}$, and locking is easily detectable as a change in the detector output voltage.

Once the signal is presented to the CMOS module, additional amplification stages are used inside the CMOS detector (Fig. 4.8(c)). The first two stages are a low-noise

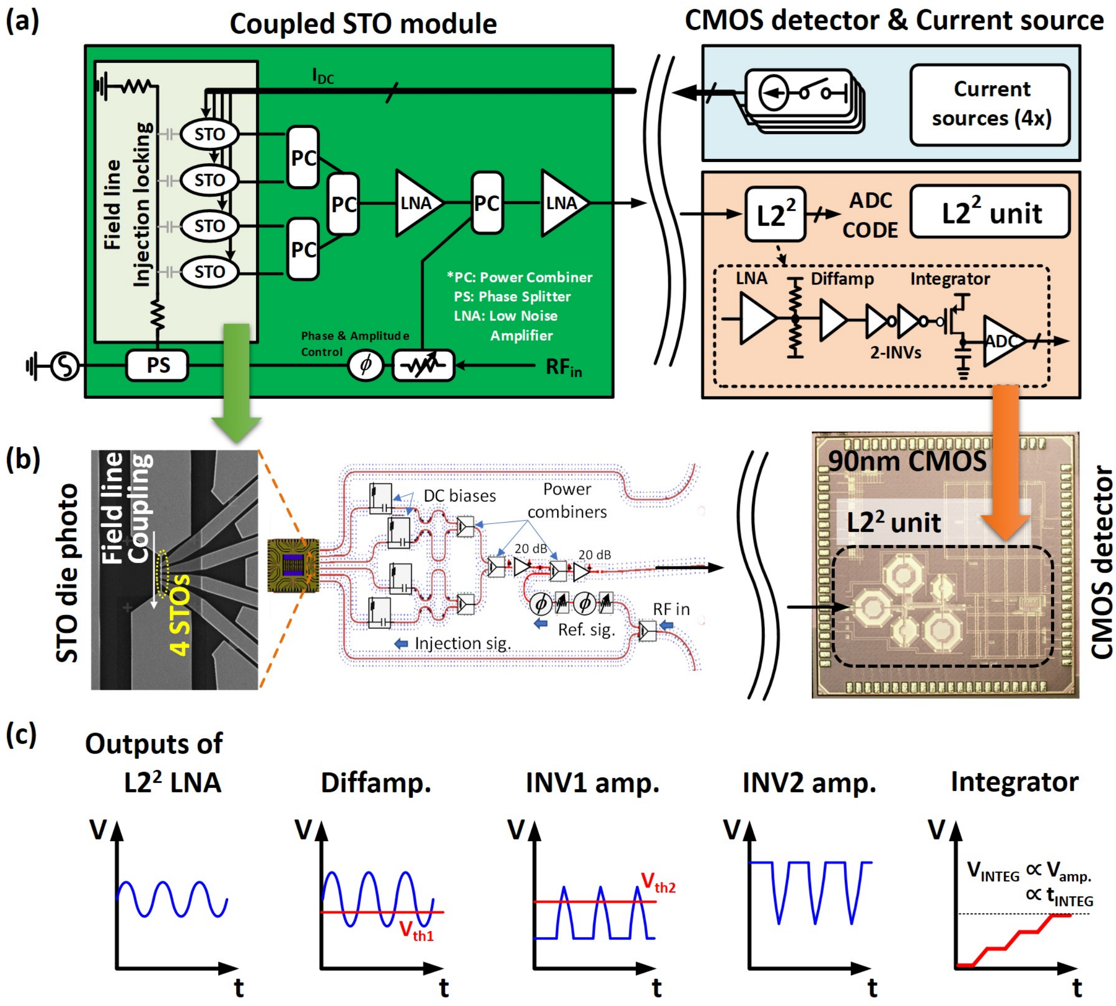


Fig. 4.8. The system implementation. (a) Block diagram for entire system, and (b) its implementation. (c) The expected signals at different nodes in the CMOS detector module.

amplifier (LNA) and a differential amplifier to make the signal compatible with CMOS circuits. Inverter-based amplifiers are used to amplify only the portion of interest—the peak of the incoming signal—based on the threshold of the inverter. Consequently, the signal at the input to the integrator behaves like the fourth waveform shown in Fig. 4.8(c) (INV2 amp.) where the signal normally stays at the VDD level, and only goes below when the INV1 amp signal exceeds the threshold level of the second inverter (V_{TH2}). The amplitude of the incoming signal V$_{amp}$ determines the depth of the dip in voltage from the VDD level, and thus the amount of current is stored onto the capacitor. Accordingly, the voltage across the capacitor rises during the integration time. Finally, the integrator output voltage is converted into a digital code at the analog to digital converter (ADC) stage for further image processing. The CMOS detector is fabricated in 90nm CMOS process.

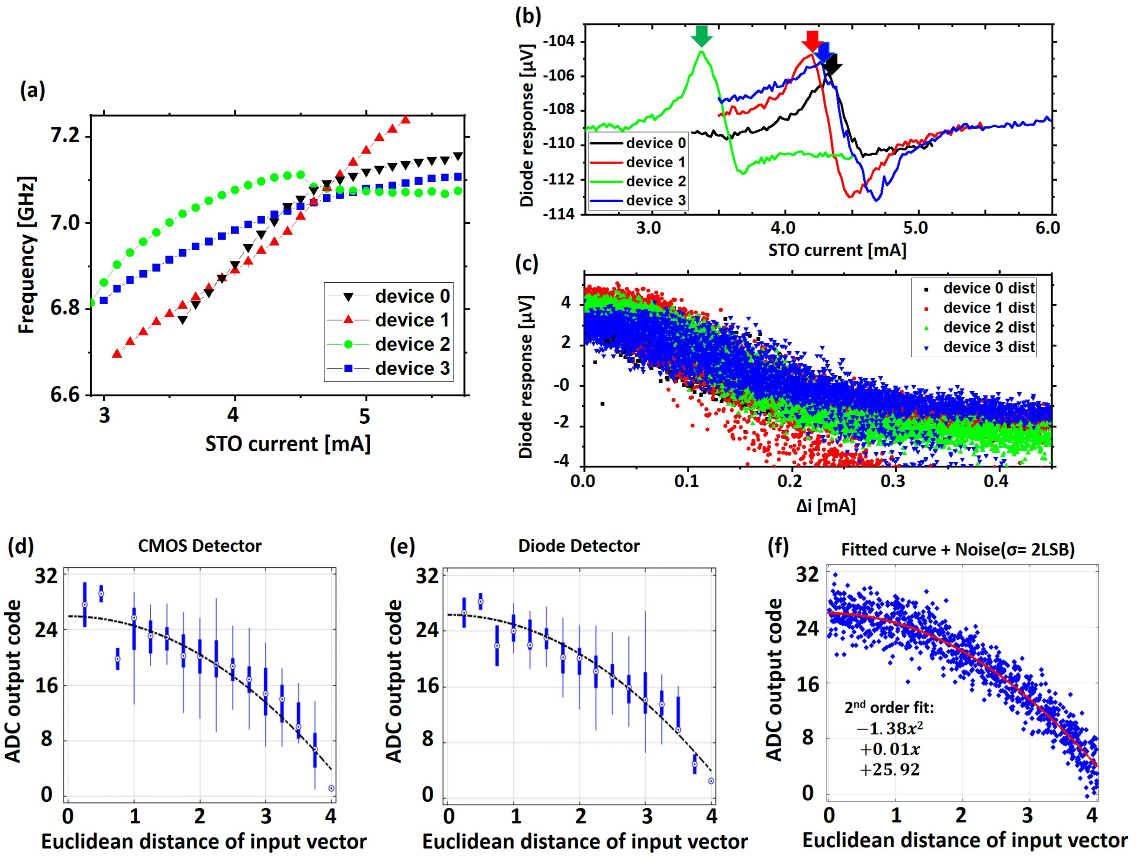


Fig. 4.9. (a) The fitted response of all devices, showing overlap at 7 GHz operating point. (b) Response curves for all devices. Arrows indicate the reference bias points. (c) Distribution of diode response for each device, 5000 test Δ I points. Box plots of response of 4 device array for test vectors within $\Delta I < 230\ \mu A$, (d) with CMOS detector and (e) with Diode detector plotted vs. ED (defined as mA/10^{-4} mA). (f) The fitted curve from (e) with normal distributed noise (σ of 2 LSB ADC code).

We first measured the response of each individual STO device in an array of four STOs to characterize the coupled oscillator network. The responses from the four different devices are shown in Fig. 4.9(a). The variations in the frequency and amplitude vs. current (i.e. the spectral response) from device-to-device are one of the most challenging aspects of creating arrays of oscillators with STOs. The devices in the array have sufficient overlap near 7 GHz to allow injection locking of all devices simultaneously. The locking response for each device in the array is shown in Fig. 4.9(b). The difference in shapes of the locking curves for a given reference signal amplitude and phase occurs due to a combination of extrinsic device differences (injection locking phase at a given STO, effective microwave path lengths) and intrinsic differences (STO locking dynamics). From the locking response, a reference current I_{0j} for each device is chosen, and a random set of test points mapped as current deviations from I_{0j} are applied. The resulting distribution is shown in Fig. 4.9(c) (see Methods section for details of this analysis). As the test current moves away from the reference point, the detected signal shown in Fig. 4.9(c) decreases monotonically, albeit with significant noise. The noise on the detected voltage at a given test current is a combination of electrical noise (primarily due to the large gain needed to detect the small GMR signals) and the asymmetry in the response curve for $I_j = I_{0j} \pm \Delta I$. Phase noise of the injection-locked STO can also add to noise. However, our measurements of the close-in power spectral density across the locking range do not show significant fluctuations out of the locking band.

Finally, the response curve of the four devices, which are simultaneously programmed based on ED of their currents ($[I_1, .., I_4]$) from the reference point I_{0j}, is shown in Fig. 4.9(d) and 4.9(e) using the CMOS detector and the diode detector, respectively. Approximately 400 randomly generated input vectors are applied to the STOs and the corresponding output voltage from the system is measured. On each box, the central mark indicates the median, the ends of the vertical blue boxes indicate the 25th and 75th percentiles, respectively, and the lines indicate the min and max values. The diode response remains monotonic with $|\Delta I|$ and it is similar to the curve calculated from the individual response curves in Fig. 4.9(c). This is typical for non-interacting devices with amplitudes much less than the reference signal. The response of the CMOS detector approximately follows that of the microwave diode detector, suggesting that the two detectors introduce similar non-linearities into the signal path. The output voltage of CMOS detector is sampled with 5-bit resolution ADC implemented in the detector for measurement. Note that the output analog voltage can be post-processed with ADC with different resolution. For the case of $A_{ref} = 5nA_0$ and $n = 4$ in Fig. 4.4, the standard deviations (σ) from the fitted curve are 0.32 and 1.98 LSB (5-bit ADC) for the case of no variation and 20° phase and 20% amplitude variation, respectively. The σ of the measured average CMOS detector response from the fitted curve is 2.67 LSB, which is larger than the σ obtained for 20° phase and 20% amplitude variation. Based on the power spectral density across the locking range, we believe that the measured noise comes mainly from the gain amplification stages which could be reduced or removed if the system is built on a single chip. Fig. 4.9(f) shows the second order fitting curve from Fig. 4.9(d) with normal distributed noise (σ for the 2 LSBs of 5-bit ADC code). We utilized the fitted curve from the measured results with normal distributed noise for approximate distance computation targeting image processing applications as described in the following section 4.5.

4.5 Applications using an STO-based L2^2 norm

To check the feasibility of using the proposed L2^2 unit for distance computation and convolution, we have parameterized the simulations using the experimentally-obtained response functions (Fig. 4.9(f)) to perform a facial recognition task and an edge detection task using Gabor filtering [113]. For facial recognition, 40 images from AT&T face database [112] are compared to each other and L2^2 calculated as shown in Fig. 4.10(a). The images with 92 x 112 pixels are converted to 5-bit grayscale. Each STO is biased to an amount of current corresponding the pixel-wise differences

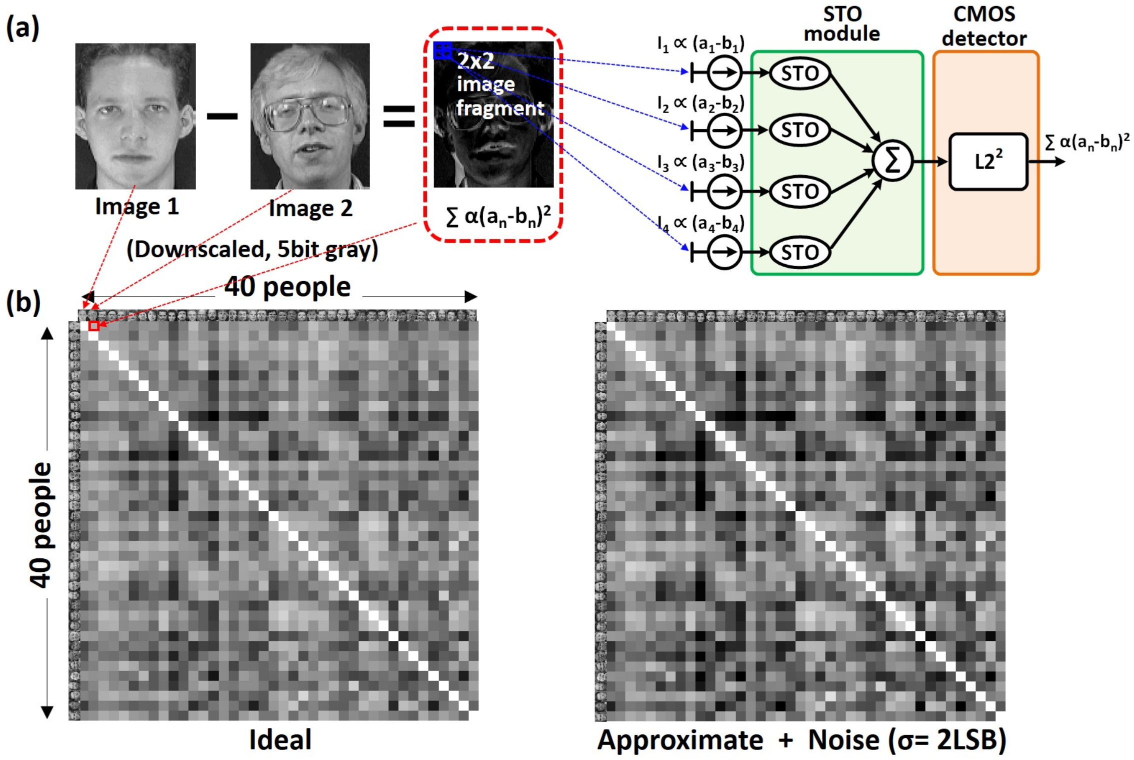


Fig. 4.10. (a) L2^2 distance computation with AT&T face database [112]. (b) The Ideal case (left) calculated mathematically and the approximate case (right) are shown using the fitting curve from the measured CMOS detector (Fig. 4.9(d)), with normal distributed noise (σ of 2 LSB ADC code) is shown.

between the reference image and the template image. As a result, squared ED (degree of match) between two images are calculated and shown for 40x40 cases. The distance is mapped on to a gray scale from white (similar) to dark (dissimilar) and plotted as a matrix in Fig. 4.10(b) for both the ideal and STO-based L2^2 calculation. It is possible to observe the similarity of response of the oscillator-based L2^2 to the ideal result.

In addition to computing the ED between $A = (a_1, a_2, ..., a_n)$ and $B = (b_1, b_2, ..., b_n)$, the L2^2 distance units (outlined in blue in Fig. 4.3(a)) can also be used to estimate convolution of two input vectors, A and B. Let us consider three L2^2 units having in-

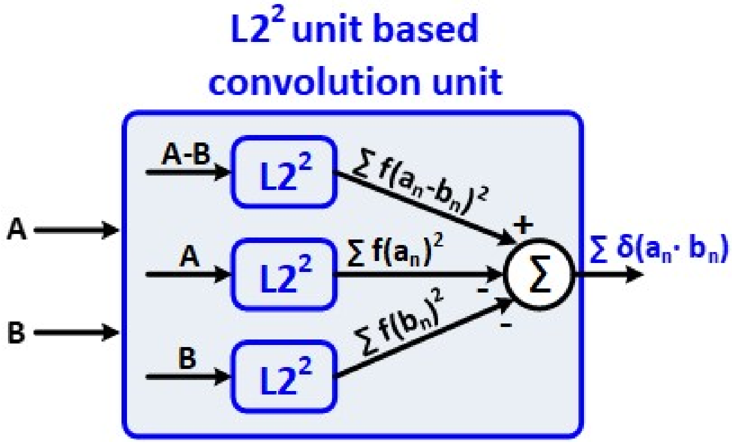


Fig. 4.11. Block diagrams for a convolution computing primitive based on ref. [105,114].

puts of $(A-B)$, A, and B as shown in Fig. 4.11. The output of $L2^2(A-B) - L2^2(A) - L2^2(B)$ can be represented as $\alpha \sum_{i=1}^{n}(a_n \cdot b_n)$, which is proportional to the dot product of A and B (or convolution) [105,114]. The computing block for convolution was used in edge detection of images to determine the efficacy of the proposed coupled oscillatory network.

Edge detection using a 2x2 kernel (using the Gabor filter [115] (Fig. 4.12(a))) was performed through the following process: First, the image was converted to 5-bit grayscale and convolved with Gabor filter kernel of size 2x2. The Gabor filter kernels have been generated based on the model in ref. [114]. For an image fragment I and the filter kernel F, the pixel-wise differences $I-F$, $I-0$, and $F-0$ were calculated, and the corresponding bias currents was applied to three L2^2 units, shown in Fig. 4.11. The system outputs a level proportional to convolution, thus generating a single pixel of the output edge map. The entire edge map is obtained by sequentially sliding the image fragment window across the image. Note that 5-bit quantization has been applied for a pixel intensity of the image and the Gabor filter kernel. Edge detection results from the ideal convolution and approximate convolution based on our system are shown in Fig. 4.12(b) for different levels of system noise. Despite the additional

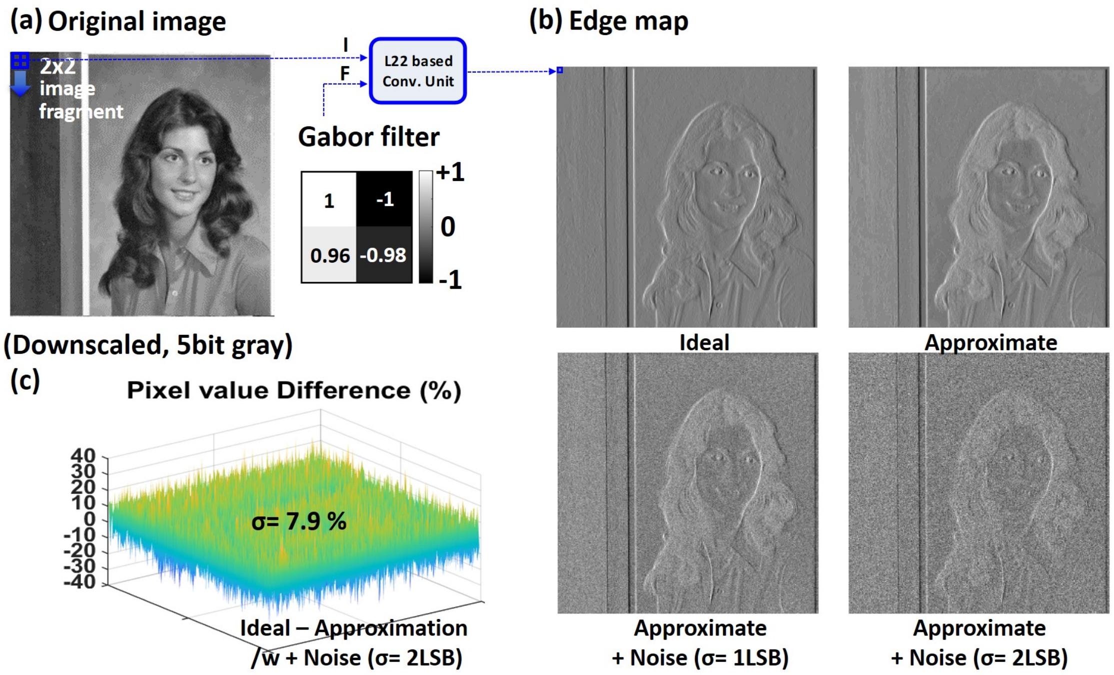


Fig. 4.12. (a) Edge detection on image (4.1.04 from SIPI database [115]) with the proposed computing primitive. (b) The edge map results from simulation (Ideal), experiment with the fitting curve (Approximate), and experiment with the fitting curve in addition to σ of 1 and 2 LSB noise. (c) The difference of the edge map between the ideal case and the approximation in addition to σ of 2 LSB noise.

noise, the images clearly show the edges present in the image, particularly when the noise is confined to the least-significant bit (LSB). The difference between the ideal edge map and approximate one with normal distributed noise ($\sigma = 2$ LSB) is plotted in Fig. 4.12(c), where the variance of the difference is 7.9%.

The energy consumption for distance computing has been estimated to project the efficiency of the proposed system. Specifically, we compare our system with CMOS based analog distance calculation circuits (DCCs) as a separate block. Note that the power consumption of our hybrid system is dominated by the LNA (See Fig. 4.13) needed to amplify the STO outputs to CMOS levels, a consequence of the GMR STOs used in the system. If higher magneto-resistance STOs are used, both

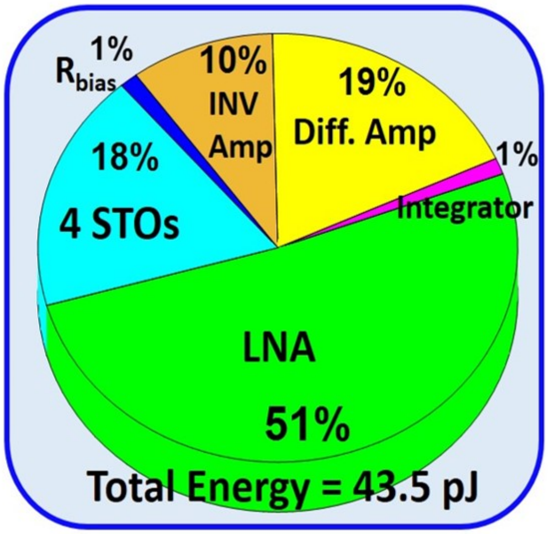


Fig. 4.13. The power breakdown of hybrid system for 4-dimensional case.

Table 4.1.
Comparison with CMOS distance computation circuits

Ref.	No. inputs	Power [mW]	fs [MHz]	Energy/dimension [pJ]
[76]	4x5	14.95	1	747.50
[79]	16x16	0.7	0.33	8.29
[116]	2x1	0.733	20	18.33
[77]	3x1	0.085	10	2.83
This work	4x1	10.87	250	10.87
	16x1	16.75	250	4.19
This work + projection [94]	4x1	8.92	357	6.25
	16x1	8.93	357	1.56

amplifier and STO power consumption (through lower operating currents) could be reduced. However, even without considering such devices, our current hybrid system with GMR-STOs becomes more power efficient as the number of dimension increases and shows comparable or better energy efficiency to CMOS analog DCCs with at least an order of magnitude faster computation speed (Table 1).

When n increases, the increase in total power consumption is only due to the power consumed by the STOs. In our system, each STO consumes 490 μW, which can be improved via the use of higher magneto-resistive materials, tunneling magneto-resistive (TMR) structures, or spin-orbit excitation (see Supplementary Information). The power consumed by STOs is only 18% of the total power in our system and is less than 1% when we assume the per-STO power consumption as 1μW [94, 117]. Therefore, power per dimension of our system decreases and converges to the power consumption of a single STO as n increases. As mentioned in section 4.3, we need a stronger reference signal as n increases to obtain more accurate distance measure. However, a stronger reference signal makes the contribution of the second term in equation (4.4) smaller and more vulnerable to noise. Hence, it is desirable to have higher resolution ADC or to have more amplification with low noise at the cost of using the stronger reference signal. We experimentally succeeded in the detection of the ED with 660 times stronger reference signal, which is sufficiently large enough to be $5n$ times stronger reference signal for more than 100 STOs.

4.6 Materials and Methods

4.6.1 STO Fabrication.

The nanocontact STOs used in this work have a pseudospin valve magnetic heterostructure of the form (thicknesses in nm):
Ta2/Cu(N)12/Ta3/Cu3/CoFe5/Cu4/[CoFe0.33/Ni0.37]x4/CoFe0.33/Cu2/Ta4 grown by sputter deposition. The Cu(N) layer is sputter-deposited in an Ar:N gas, which produces a smoother underlayer. The multilayer CoFe/Ni free layer has an effective in-plane magnetization $M_{eff} = 0.15$ T due to the surface anisotropy of the Ni interfaces. [118, 119]

The nanocontacts are formed via a self-aligned process. Electron-beam lithography is used to define the $\approx$ 70 nm diameter resist pillar on the film stack using a negative tone resist, and a 50 nm SiN$_x$ conformal layer then deposited. Mechanical polishing shears off the SiN_x/resist asperity, and after an O_2 plasma ash, results in a 70 nm nanocontact through the SiN_x. A Ta3/Au100 lead forming the center conductor of a coplanar waveguide contacts this nanocontact to bias and read out the STO. 50 nm of SiN_x is then deposited over this structure, upon which the microwave microstrip is patterned to deliver the RF injection magnetic field. To accomplish a high bandwidth, symmetric injection-locking scheme consisting of an STO device chip and RF signal conditioning board was fabricated as shown in Fig. 4.8(b) (see Supplementary Information for details).

4.6.2 RF conditioning board.

The RF delivery and signal combining board uses grounded CPW lines to deliver the injection signal and read out the STO dynamics (see Supplementary Information Fig. 4.15 for schematic). These ground-stitched CPWs have an effective bandwidth of $>$ 20 GHz. The different CPW path length to each STO are compensated on the signal conditioning board when combining the STO signals. The Wilkinson power combiners/splitters used have an insertion loss of 4 dB, and a typical imbalance in phase of 2.3° and amplitude of 0.1 dB. The two amplifiers used each have a gain of 20 dB, and a cutoff frequency of 8 GHz. To adjust the amplitude and phase of the coupled reference signal, the injection signal is first split on the signal conditioning board by a coherent power divider, adjusted in amplitude and phase by a digital attenuator (0.5 dB steps) and phase shifter (5.625° steps), and then coherently combined with the summed STO signals, producing a "net" reference signal. The CPWs terminate in Cu spring fingers at a chip pocket, which make contact to edge pads on the STO chip. These contacts have minimal insertion loss, and enable more efficient testing of multiple STO arrays and chips. The board was constructed of nonmagnetic elements, to allow usage in the gap of an electromagnet.

4.6.3 Distribution analysis.

The distributions shown in Fig. 4.9 are formed with 5000 trials, a measurement which took 20 min due to the laboratory equipment used. Due to the phase sensitive nature of the measurement, the zero point of the measurement drifted over this time, likely due to small temperature changes in the board. This drift was on the order of the device signal over the course of the measurement. A low-frequency smoothing procedure was applied to the data to "de-trend" the data, and is described in the Supplementary Information.

4.7 Supplementary Information

4.7.1 Distribution detrending

A series of bias currents are applied to the STO to determine the distribution of the detector response. A time trace of the output of a single device is shown in Fig. 4.14(a) for both the diode and CMOS detectors. Since these are applied at random, with a Gaussian distribution around zero (equivalently, a Gaussian distribution of currents around the center current were applied to the device). These traces should be flat with time, and the slow variations observed are due to small changes in the phase condition of the reference. These variations are slow, and are significant only due to the slow current sources used in this test (roughly 2 Hz data rate). These variations are a significant fraction of the device signal, and can thus mask the desired response. To remove these slow variations, a smoothing function was applied to the data (Savitsky-Golay second-order smoothing with 200 points) and subtracted. The diode detector is a standard microwave diode detector, sampled with a digital multimeter with ≈6 Hz bandwidth. The CMOS detector, on the other hand, is controlled via an field-programmable gate array (FPGA) board, which is queried once per current bias. Each query returns a value that is the average of 5000 samples of the detector itself (the FPGA code was also modified to measure the RMS error on these samples

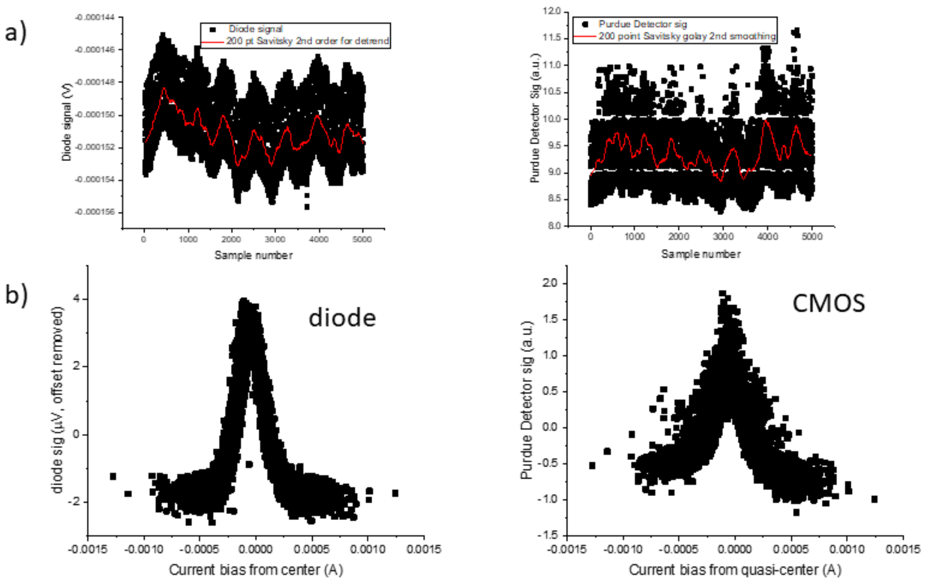


Fig. 4.14. Response of detectors to phase locking of a single STO device for both diode and CMOS detectors.(a) shows response as a function of time, for which random offsets are applied to the device. (b) shows the resultant response vs. current.

was measured separately.) The banding evident in the CMOS detector response in Fig. 4.9(b) is due to the bit depth of the analog-to-digital converter (ADC) on the detector. Note that the two detector traces are taken essentially at the same time, so that the slow time variations of the two signals are similar but not identical. There is a slow quasi-periodic oscillation (possibly due to temperature variations) visible in both, while an even slower variation is also evident in the diode detection. It is unknown what the source of this additional variation is, but could be due to the larger bandwidth of the diode detector. The resulting response functions are shown as a

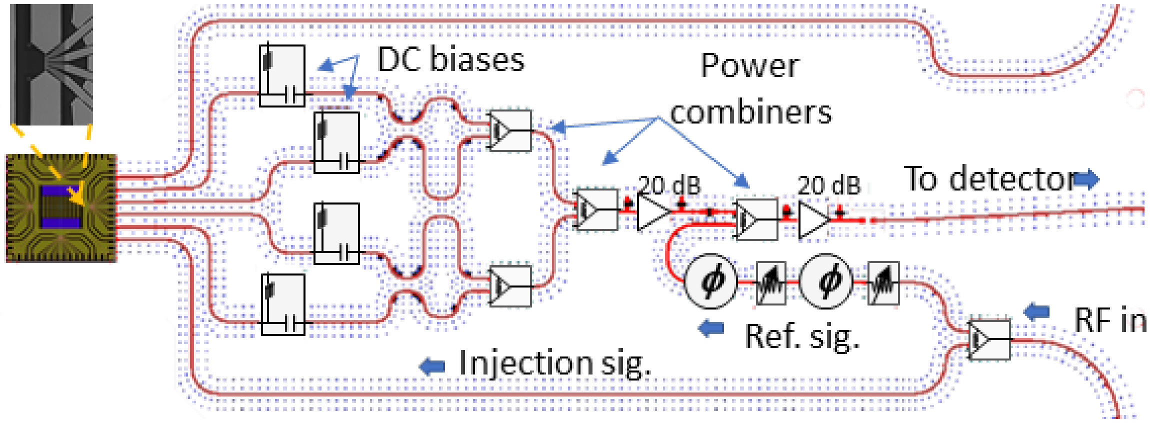


Fig. 4.15. Layout of microwave signal conditioning board.

function of bias current difference in Fig. 4.14(b), showing a proper peaked response of the device signal suitable for the distance computation. Use of faster measurement times, and possibly a self-referencing algorithm that dynamically accounts for phase drift, would remove the need for this additional step.

4.7.2 Signal conditioning board

The microwave signal conditioning board was constructed as shown in Fig. 4.15 to permit the introduction and measurement of microwave (5-10 GHz) signals to a separately fabricated STO chip. The STO chip has coplanar waveguide traces that are contacted by the board using spring-finger contacts with effective bandwidths > 10 GHz. The traces on the board are grounded, stitched, coplanar waveguides (bandwidths > 20 GHz), which connect microwave ICs for power combining, phase shifting, amplitude trimming, and amplification.

4.7.3 Addition of sinusoids at the same frequency but different phase

Addition of sinusoids at the same frequency but different phase can be expanded

$$\begin{aligned}\varphi &= \sum_{i=1}^{n} A_i cos(\omega t - \delta_i) = \sum_{i=1}^{n} A_i [cos(\omega t) cos(\delta_i) + sin(\omega t) sin(\delta_i)] \\ &= cos(\omega t) \cdot \sum_{i=1}^{n} A_i cos(\delta_i) + sin(\omega t) \cdot \sum_{i=1}^{n} A_i sin(\delta_i)\end{aligned} \tag{4.8}$$

Defining A$\cos\delta = \sum_{i=1}^{n} A_i cos(\delta_i)$ and A$\sin\delta = \sum_{i=1}^{n} A_i sin(\delta_i)$, (1) becomes

$$\varphi = A cos(\omega t) \cdot cos(\delta) + A sin(\omega t) \cdot sin(\delta) = A cos(\omega t - \delta), \tag{4.9}$$

where

$$\begin{aligned}A^2 &= \sum_{i=1}^{n} \sum_{j=1}^{n} A_i A_j cos(\delta_i - \delta_j) \\ \delta &= tan^{-1} \frac{\sum_{i=1}^{n} A_i cos(\delta_i)}{\sum_{i=1}^{n} A_i cos(\delta_i)}\end{aligned} \tag{4.10}$$

Note that the amplitude A is a function of δ_i, which are the variables from which we will calculate distance. If we define the last element of the sum as the reference signal $A_n = A_{ref}$, $\delta_n = \delta_{ref}$, then the amplitude A^2 can be written as

$$A^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} A_i A_j cos\,(\delta_i - \delta_j) + 2A_{ref} \sum_{j=1}^{n} A_j cos\,(\delta_j) + A_{ref}^2 \tag{4.11}$$

We expand equation (4.11) with the approximation of $cos\theta \approx 1 - \theta^2/2$ as below.

$$
\begin{aligned}
A^2 &= \sum_{i=1}^{n}\sum_{j=1}^{n} A_i A_j cos\,(\delta_i - \delta_j) + 2A_{ref}\sum_{j=1}^{n} A_j cos\,(\delta_j) + A_{ref}^2 \\
&\cong \sum_{i=1}^{n}\sum_{j=1}^{n} A_i A_j (1 - \frac{(\delta_i - \delta_j)^2}{2}) + 2A_{ref}\sum_{j=1}^{n} A_j (1 - \frac{\delta_j^2}{2}) + A_{ref}^2 \\
&= (A_0 n + A_{ref})^2 - \frac{A_0^2}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\delta_i - \delta_j)^2 - A_{ref}A_0\sum_{j=1}^{n}\delta_j^2 \\
&= (A_0 n + A_{ref})^2 - \frac{A_0^2}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\delta_i^2 + \delta_j^2) + A_0^2\sum_{i=1}^{n}\sum_{j=1}^{n}\delta_i\delta_j - A_{ref}A_0\sum_{j=1}^{n}\delta_j^2 \\
&= (A_0 n + A_{ref})^2 - (nA_0^2 + A_{ref}A_0)\sum_{j=1}^{n}\delta_j^2 + A_0^2\sum_{i=1}^{n}\sum_{j=1}^{n}\delta_i\delta_j \\
&= (A_0 n + A_{ref})^2 - ((n-1)A_0^2 + A_{ref}A_0)\sum_{i=1}^{n}\delta_i^2 + A_0^2((\sum_{i=1}^{n}\delta_i)^2 - \sum_{i=1}^{n}\delta_i^2)
\end{aligned}
\tag{4.12}
$$

, where A_0 is the amplitude of n oscillators.

The amplitude of the combined signal can be approximated to Euclidean distance square as shown above. The first term of the last equation in (4.12) is constant with respect to δ, and the second term is proportional to Euclidean distance square of n-dimensional vector $[\delta_1, \delta_2, ..., \delta_n]$. We call the third term as the error term since we can obtain the approximated Euclidean distance if the error term is small enough to be tolerable.

4.8 Summary

We have experimentally demonstrated a core distance computing primitive based on an STO-based coupled oscillator array. Starting from the theoretical background of obtaining an L2^2 norm from a coupled oscillator array, we have shown that the combination of injection locking of the oscillators and their interference with a reference signal can be efficiently used to realize the distance computation unit. The performance of the system as an L2^2 unit was examined by applying randomly generated test input vectors as bias current to the STOs and generating corresponding output digital codes from the CMOS detector. The characteristic curve from the experiment approximates an L2^2 norm which, in turn, is used to determine the feasibility of the STO-based coupled system for image processing applications. The approximate distance and convolution output based on our system shows reasonable accuracy as compared to the ideal results. Energy and scaling analysis shows that GMR-based STOs for distance computation have higher efficiency than CMOS-based DCCs for high dimensional input vectors. Modest improvements in STO critical currents and magneto-resistance (through the use of magnetic tunnel junctions) can make oscillator-based systems even more attractive.

5. CONCLUSIONS

Low power and energy-efficient computing have become extremely important as more and more data-centric applications are being deployed in the edge devices. Further, the number of edge devices used per person has continuously increased over the past decade with majority of the devices being battery powered. This has motivated the design of neuromorphic systems to enable data processing capabilities at the edge devices by looking into alternative computing models across the design stack: device, circuit, algorithm, and architecture.

In this research, we proposed energy efficient circuit, interconnect, and architecture for energy efficient neuromorphic computing. First, we propose a hybrid Power Line Communication (PLC)- Network On Chip (NOC) based neuromorphic architecture built with memristive crossbars to enable efficient multi-layer inference for for Spiking Neural Networks (SNNs). Both low-power computation units and energy-efficient interconnect are fundamental to efficient neuromorphic system design. Our hybrid interconnect harnesses the different data-transfer patterns in typical many-core architecture to optimize energy expended in data communication. Additionally, memristive crossbar based PEs achieve low energy consumption for neuromorphic computations. Our experiments over a wide range of spiking neural network benchmarks show average energy improvements of ∼39.32% at comparable latency.

In chapter 3, we proposed stochastic binary SNN that requires only one-bit precision for the constituting neurons and synapses for memory-compressed neuromorphic computing. We presented an energy-efficient implementation of the binary SNN using Biased Random Number Generated (BRNG) as ‘stochastic bit’ to realize the stochastic neurons and synapses (during training) fabricated in 90nm CMOS process. We demonstrated high energy efficiency of 89.49 TOPS/Watt for two-layer SNN, which renders the proposed realization amenable for IoT/edge devices with on-chip intelligence.

Finally, we proposed and demonstrated coupled oscillator based distance computation system using giant magnetoresistance (GMR)-based Spin Torque Oscillators (STOs). By exploiting inejction locking of the oscillators, we successfully realize an Euclidean distance computation unit ($L2^2$ unit). The $L2^2$ unit can act as a core distance computing primitive to perform a facial recognition and edge detection tasks. Energy and scaling analyses show that GMR-based STOs for distance computation have higher efficiency than CMOS-based distance computation circuits for high dimensional input vectors. We expect a TMR based device would further enhance the energy improvements.

REFERENCES

REFERENCES

[1] (2016) Beyond cmos, irds. [Online]. Available: http://irds.ieee.org/reports

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

[3] C. Huygens, *Oeuvres complètes de Christiaan Huygens.* M. Nijhoff, 1899, vol. 8.

[4] B. Ermentrout, "An adaptive model for synchrony in the firefly pteroptyx malaccae," *Journal of Mathematical Biology*, vol. 29, no. 6, pp. 571–585, 1991.

[5] S.-Y. Ha, E. Jeong, and M.-J. Kang, "Emergent behaviour of a generalized viscek-type flocking model," *Nonlinearity*, vol. 23, no. 12, p. 3139, 2010.

[6] N. Wiener, *Cybernetics or Control and Communication in the Animal and the Machine.* MIT press, 1965, vol. 25.

[7] G. Buzsáki and A. Draguhn, "Neuronal oscillations in cortical networks," *science*, vol. 304, no. 5679, pp. 1926–1929, 2004.

[8] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on circuit theory*, vol. 18, no. 5, pp. 507–519, 1971.

[9] Y. Chen, T. Luo, S. Liu, S. Zhang, L. He, J. Wang, L. Li, T. Chen, Z. Xu, N. Sun *et al.*, "Dadiannao: A machine-learning supercomputer," in *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture.* IEEE Computer Society, 2014, pp. 609–622.

[10] Y.-H. Chen, J. Emer, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," in *Computer Architecture (ISCA), 2016 ACM/IEEE 43rd Annual International Symposium on.* IEEE, 2016, pp. 367–379.

[11] 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.

[12] 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.

[13] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *arXiv preprint arXiv:1602.02410*, 2016.

[14] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, “Deep learning for identifying metastatic breast cancer,” *arXiv preprint arXiv:1606.05718*, 2016.

[15] 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.

[16] S. Shalev-Shwartz, S. Shammah, and A. Shashua, “Safe, multi-agent, reinforcement learning for autonomous driving,” *arXiv preprint arXiv:1610.03295*, 2016.

[17] Y. LeCun, C. Cortes, and C. J. Burges, “The mnist database of handwritten digits,” 1998.

[18] 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.

[19] A. Karpathy and L. Fei-Fei, “Deep visual-semantic alignments for generating image descriptions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.

[20] 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.

[21] A. Ankit, A. Sengupta, and K. Roy, “Trannsformer: Neural network transformation for memristive crossbar based neuromorphic system design,” *arXiv preprint arXiv:1708.07949*, 2017.

[22] V. Sze, Y.-H. Chen, T.-J. Yang, and J. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” *arXiv preprint arXiv:1703.09039*, 2017.

[23] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain *et al.*, “Loihi: A neuromorphic manycore processor with on-chip learning,” *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[24] 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.

[25] 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.

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

[27] 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.

[28] X. Liu, M. Mao, B. Liu, H. Li, Y. Chen, B. Li, Y. Wang, H. Jiang, M. Barnell, Q. Wu *et al.*, "Reno: A high-efficient reconfigurable neuromorphic computing accelerator design," in *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*. IEEE, 2015, pp. 1–6.

[29] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *Proceedings of the 43rd International Symposium on Computer Architecture*. IEEE Press, 2016, pp. 14–26.

[30] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, "Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory," in *Proceedings of the 43rd International Symposium on Computer Architecture*. IEEE Press, 2016, pp. 27–39.

[31] M. Cheng, L. Xia, Z. Zhu, Y. Cai, Y. Xie, Y. Wang, and H. Yang, "Time: A training-in-memory architecture for memristor-based deep neural networks," in *Proceedings of the 54th Annual Design Automation Conference 2017*. ACM, 2017, p. 26.

[32] L. Song, X. Qian, H. Li, and Y. Chen, "Pipelayer: A pipelined reram-based accelerator for deep learning," in *High Performance Computer Architecture (HPCA), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 541–552.

[33] J. Kim, J. Balfour, and W. Dally, "Flattened butterfly topology for on-chip networks," in *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE Computer Society, 2007, pp. 172–182.

[34] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha, "Toward ideal on-chip communication using express virtual channels," *IEEE micro*, vol. 28, no. 1, 2008.

[35] V. F. Pavlidis and E. G. Friedman, "3-d topologies for networks-on-chip," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 10, pp. 1081–1090, 2007.

[36] A. Shacham, K. Bergman, and L. P. Carloni, "Photonic networks-on-chip for future generations of chip multiprocessors," *IEEE Transactions on Computers*, vol. 57, no. 9, pp. 1246–1260, 2008.

[37] M. F. Chang, J. Cong, A. Kaplan, M. Naik, G. Reinman, E. Socher, and S.-W. Tam, "Cmp network-on-chip overlaid with multi-band rf-interconnect," in *High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on*. Ieee, 2008, pp. 191–202.

[38] V. Chawla and D. S. Ha, "Dual use of power lines for data communications in microprocessors," in *Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2011 IEEE 14th International Symposium on*. IEEE, 2011, pp. 23–28.

[39] J. M. Salem and D. S. Ha, "Dual use of power lines for design-for-testability—a cmos receiver design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 3, pp. 1118–1125, 2016.

[40] 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, pp. 2963–2970, 2016.

[41] N. H. Weste and K. Eshraghian, "Principles of cmos vlsi design: A systems perspective second edition," *Addision-Wesley Publishing, California, l994*, 1994.

[42] J. Liang and H.-S. P. Wong, "Cross-point memory array without cell selectors—device characteristics and data storage pattern dependencies," *IEEE Transactions on Electron Devices*, vol. 57, no. 10, pp. 2531–2538, 2010.

[43] Y. Shi and L. He, "Modeling and design for beyond-the-die power integrity," in *Computer-Aided Design (ICCAD), 2010 IEEE/ACM International Conference on.* IEEE, 2010, pp. 411–416.

[44] N. Jiang, J. Balfour, D. U. Becker, B. Towles, W. J. Dally, G. Michelogiannakis, and J. Kim, "A detailed and flexible cycle-accurate network-on-chip simulator," in *Performance Analysis of Systems and Software (ISPASS), 2013 IEEE International Symposium on.* IEEE, 2013, pp. 86–96.

[45] A. B. Kahng, B. Li, L.-S. Peh, and K. Samadi, "Orion 2.0: A power-area simulator for interconnection networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 1, pp. 191–196, 2012.

[46] 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.

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

[48] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop on deep learning and unsupervised feature learning*, vol. 2011, no. 2, 2011, p. 5.

[49] P. Blouw, X. Choo, E. Hunsberger, and C. Eliasmith, "Benchmarking keyword spotting efficiency on neuromorphic hardware," *arXiv preprint arXiv:1812.01739*, 2018.

[50] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: Vgg and residual architectures," *Frontiers in neuroscience*, vol. 13, 2019.

[51] C. Lee, S. S. Sarwar, and K. Roy, "Enabling spike-based backpropagation in state-of-the-art deep neural network architectures," *arXiv preprint arXiv:1903.06379*, 2019.

[52] K. Cheung, S. R. Schultz, and W. Luk, "Neuroflow: a general purpose spiking neural network simulation platform using customizable processors," *Frontiers in neuroscience*, vol. 9, p. 516, 2016.

[53] B. Linares-Barranco and T. Serrano-Gotarredona, "Memristance can explain spike-time-dependent-plasticity in neural synapses," 2009.

[54] V. Milo, G. Pedretti, R. Carboni, A. Calderoni, N. Ramaswamy, S. Ambrogio, and D. Ielmini, "Demonstration of hybrid cmos/rram neural networks with spike time/rate-dependent plasticity," in *2016 IEEE International Electron Devices Meeting (IEDM).* IEEE, 2016, pp. 16–8.

[55] Y. Shi, L. Nguyen, S. Oh, X. Liu, F. Koushan, J. R. Jameson, and D. Kuzum, "Neuroinspired unsupervised learning and pruning with subquantum cbram arrays," *Nature communications*, vol. 9, no. 1, p. 5312, 2018.

[56] 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.

[57] 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.

[58] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-v power supply high-speed digital circuit technology with multithreshold-voltage cmos," *IEEE Journal of Solid-state circuits*, vol. 30, no. 8, pp. 847–854, 1995.

[59] M. Suri, O. Bichler, D. Querlioz, G. Palma, E. Vianello, D. Vuillaume, C. Gamrat, and B. DeSalvo, "Cbram devices as binary synapses for low-power stochastic neuromorphic systems: auditory (cochlea) and visual (retina) cognitive processing applications," in *2012 International Electron Devices Meeting.* IEEE, 2012, pp. 10–3.

[60] A. Yousefzadeh, E. Stromatias, M. Soto, T. Serrano-Gotarredona, and B. Linares-Barranco, "On practical issues for stochastic stdp hardware with 1-bit synaptic weights," *Frontiers in neuroscience*, vol. 12, 2018.

[61] G.-q. Bi and M.-m. Poo, "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type," *Journal of neuroscience*, vol. 18, no. 24, pp. 10 464–10 472, 1998.

[62] S. Lowel and W. Singer, "Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity," *Science*, vol. 255, no. 5041, pp. 209–212, 1992.

[63] J. Holleman, S. Bridges, B. P. Otis, and C. Diorio, "A 3μw cmos true random number generator with adaptive floating-gate offset cancellation," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 5, pp. 1324–1336, 2008.

[64] C. S. Petrie and J. A. Connelly, "A noise-based ic random number generator for applications in cryptography," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 5, pp. 615–621, 2000.

[65] S. K. Mathew, S. Srinivasan, M. A. Anders, H. Kaul, S. K. Hsu, F. Sheikh, A. Agarwal, S. Satpathy, and R. K. Krishnamurthy, "2.4 gbps, 7 mw all-digital pvt-variation tolerant true random number generator for 45 nm cmos high-performance microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 11, pp. 2807–2821, 2012.

[66] S. Ambrogio, S. Balatti, V. Milo, R. Carboni, Z. Wang, A. Calderoni, N. Ramaswamy, and D. Ielmini, "Novel rram-enabled 1t1r synapse capable of low-power stdp via burst-mode communication and real-time unsupervised machine learning," in *VLSI Technology, 2016 IEEE Symposium on.* IEEE, 2016, pp. 1–2.

[67] S. Kim, M. Ishii, S. Lewis, T. Perri, M. BrightSky, W. Kim, R. Jordan, G. Burr, N. Sosa, A. Ray *et al.*, "Nvm neuromorphic core with 64k-cell (256-by-256) phase change memory synaptic array with on-chip neuron circuits for continuous in-situ learning," in *Electron Devices Meeting (IEDM), 2015 IEEE International.* IEEE, 2015, pp. 17–1.

[68] M. Jerry, A. Parihar, B. Grisafe, A. Raychowdhury, and S. Datta, "Ultra-low power probabilistic imt neurons for stochastic sampling machines," in *2017 Symposium on VLSI Technology.* IEEE, 2017, pp. T186–T187.

[69] X. Wu, V. Saxena, K. Zhu, and S. Balagopal, "A cmos spiking neuron for brain-inspired neural networks with resistive synapses and in situ learning," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 11, pp. 1088–1092, 2015.

[70] E. Neftci, S. Das, B. Pedroni, K. Kreutz-Delgado, and G. Cauwenberghs, "Event-driven contrastive divergence for spiking neuromorphic systems," *Frontiers in neuroscience*, vol. 7, p. 272, 2014.

[71] O.-C. Chen and R.-B. Sheen, "A power-efficient wide-range phase-locked loop," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 1, pp. 51–62, 2002.

[72] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[73] S.-R. Kuang, J.-P. Wang, and C.-Y. Guo, "Modified booth multipliers with a regular partial product array," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 5, pp. 404–408, 2009.

[74] E. Antelo, P. Montuschi, and A. Nannarelli, "Improved 64-bit radix-16 booth multiplier based on partial product array height reduction," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 2, pp. 409–418, 2016.

[75] A. Gopalan and A. H. Titus, "A new wide range euclidean distance circuit for neural network hardware implementations," *IEEE transactions on neural networks*, vol. 14, no. 5, pp. 1176–1186, 2003.

[76] B.-D. Liu, C.-Y. Chen, and J.-Y. Tsao, "A modular current-mode classifier circuit for template matching application," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 2, pp. 145–151, 2000.

[77] T. Talaśka, M. Kolasa, R. Długosz, and W. Pedrycz, "Analog programmable distance calculation circuit for winner takes all neural network realized in the cmos technology," *Ieee transactions on neural networks and learning systems*, vol. 27, no. 3, pp. 661–673, 2015.

[78] G. T. Tuttle, S. Fallahi, and A. A. Abidi, "An 8 b cmos vector a/d converter," in *1993 IEEE International Solid-State Circuits Conference Digest of Technical Papers.* IEEE, 1993, pp. 38–39.

[79] G. Cauwenberghs and V. Pedroni, "A low-power cmos analog vector quantizer," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 8, pp. 1278–1283, 1997.

[80] A. Raychowdhury, A. Parihar, G. H. Smith, V. Narayanan, G. Csaba, M. Jerry, W. Porod, and S. Datta, "Computing with networks of oscillatory dynamical systems," *Proceedings of the IEEE*, vol. 107, no. 1, pp. 73–89, 2019.

[81] D. E. Nikonov, G. Csaba, W. Porod, T. Shibata, D. Voils, D. Hammerstrom, I. A. Young, and G. I. Bourianoff, "Coupled-oscillator associative memory array operation for pattern recognition," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 1, pp. 85–93, 2015.

[82] S. P. Levitan, Y. Fang, D. H. Dash, T. Shibata, D. E. Nikonov, and G. I. Bourianoff, "Non-boolean associative architectures based on nano-oscillators," in *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, Aug 2012, pp. 1–6.

[83] V. Narayanan, S. Datta, G. Cauwenberghs, D. Chiarulli, S. Levitan, and P. Wong, "Video analytics using beyond cmos devices," in *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2014, pp. 1–5.

[84] D. Fan, S. Maji, K. Yogendra, M. Sharad, and K. Roy, "Injection-locked spin hall-induced coupled-oscillators for energy efficient associative computing," *IEEE Transactions on Nanotechnology*, vol. 14, no. 6, pp. 1083–1093, Nov 2015.

[85] J. A. Carpenter, Y. Fang, C. N. Gnegy, D. M. Chiarulli, and S. P. Levitan, "An image processing pipeline using coupled oscillators," in *2014 14th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA)*, July 2014, pp. 1–2.

[86] T. Shibata, R. Zhang, S. P. Levitan, D. E. Nikonov, and G. I. Bourianoff, "Cmos supporting circuitries for nano-oscillator-based associative memories," in *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications.* IEEE, 2012, pp. 1–5.

[87] G. Csaba, M. Pufall, D. E. Nikonov, G. I. Bourianoff, A. Horvath, T. Roska, and W. Porod, "Spin torque oscillator models for applications in associative memories," in *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, Aug 2012, pp. 1–2.

[88] P. Baldi and R. Meir, "Computing with arrays of coupled oscillators: An application to preattentive texture discrimination," *Neural Computation*, vol. 2, no. 4, pp. 458–471, 1990.

[89] F. C. Hoppensteadt and E. M. Izhikevich, "Synchronization of mems resonators and mechanical neurocomputing," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 2, pp. 133–138, 2001.

[90] ——, "Pattern recognition via synchronization in phase-locked loop neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 734–738, 2000.

[91] E. Vassilieva, G. Pinto, J. A. De Barros, and P. Suppes, "Learning pattern recognition through quasi-synchronization of phase oscillators," *IEEE Transactions on Neural Networks*, vol. 22, no. 1, pp. 84–95, 2010.

[92] D. Vodenicarevic, N. Locatelli, F. A. Araujo, J. Grollier, and D. Querlioz, "A nanotechnology-ready computing scheme based on a weakly coupled oscillator network," *Scientific reports*, vol. 7, p. 44772, 2017.

[93] H. Arai and H. Imamura, "Neural-network computation using spin-wave-coupled spin-torque oscillators," *Physical Review Applied*, vol. 10, no. 2, p. 024040, 2018.

[94] M. Romera, P. Talatchian, S. Tsunegi, F. A. Araujo, V. Cros, P. Bortolotti, J. Trastoy, K. Yakushiji, A. Fukushima, H. Kubota *et al.*, "Vowel recognition with four coupled spin-torque nano-oscillators," *Nature*, vol. 563, no. 7730, p. 230, 2018.

[95] N. Shukla, A. Parihar, M. Cotter, M. Barth, X. Li, N. Chandramoorthy, H. Paik, D. G. Schlom, V. Narayanan, A. Raychowdhury, and S. Datta, "Pairwise coupled hybrid vanadium dioxide-mosfet (hvfet) oscillators for non-boolean associative computing," in *2014 IEEE International Electron Devices Meeting*, Dec 2014, pp. 28.7.1–28.7.4.

[96] N. Shukla, W.-Y. Tsai, M. Jerry, M. Barth, V. Narayanan, and S. Datta, "Ultra low power coupled oscillator arrays for computer vision applications," in *2016 IEEE Symposium on VLSI Technology*. IEEE, 2016, pp. 1–2.

[97] S. Dutta, A. Parihar, A. Khanna, J. Gomez, W. Chakraborty, M. Jerry, B. Grisafe, A. Raychowdhury, and S. Datta, "Programmable coupled oscillators for synchronized locomotion," *Nature communications*, vol. 10, no. 1, p. 3299, 2019.

[98] N. Gala, S. Krithivasan, W.-Y. Tsai, X. Li, V. Narayanan, and V. Kamakoti, "An accuracy tunable non-boolean co-processor using coupled nano-oscillators," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 14, no. 1, p. 1, 2018.

[99] P. Talatchian, M. Romera, F. A. Araujo, P. Bortolotti, V. Cros, D. Vodenicarevic, N. Locatelli, D. Querlioz, and J. Grollier, "Designing large arrays of interacting spin-torque nano-oscillators for microwave information processing," *arXiv preprint arXiv:1908.09908*, 2019.

[100] S. Kaka, M. R. Pufall, W. H. Rippard, T. J. Silva, S. E. Russek, and J. A. Katine, "Mutual phase-locking of microwave spin torque nano-oscillators," *Nature*, vol. 437, no. 389, 2005.

[101] D. Houssameddine, U. Ebels, B. DelaÃt, B. Rodmacq, I. Firastrau, F. Ponthenier, M. Brunet, C. Thirion, J.-P. Michel, L. Prejbeanu-Buda, M.-C. Cyrille, O. Redon, and B. Dieny, "Spin-torque oscillator using a perpendicular polarizer and a planar free layer," *Nature Materials*, vol. 6, no. 447, 2007.

[102] P. M. Braganca, B. A. Gurney, B. A. Wilson, J. A. Katine, S. Maat, and J. R. Childress, "Nanoscale magnetic field detection using a spin torque oscillator," *Nanotechnology*, vol. 21, no. 23, p. 235202, 2010. [Online]. Available: http://stacks.iop.org/0957-4484/21/i=23/a=235202

[103] W. H. Rippard, M. R. Pufall, S. Kaka, S. E. Russek, and T. J. Silva, “Direct-current induced dynamics in $co_{90}fe_{10}/ni_{80}fe_{20}$ point contacts,” *Phys. Rev. Lett.*, vol. 92, p. 027201, Jan 2004. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.92.027201

[104] W. H. Rippard, M. R. Pufall, S. Kaka, T. J. Silva, S. E. Russek, and J. A. Katine, “Injection locking and phase control of spin transfer nano-oscillators,” *Phys. Rev. Lett.*, vol. 95, p. 067203, Aug 2005. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.95.067203

[105] D. M. Chiarulli, B. Jennings, Y. Fang, A. Seel, and S. P. Levitan, “A computational primitive for convolution based on coupled oscillator arrays,” in *2015 IEEE Computer Society Annual Symposium on VLSI*, July 2015, pp. 125–130.

[106] Y. Zhou, J. Persson, and J. Åkerman, “Intrinsic phase shift between a spin torque oscillator and an alternating current,” *Journal of Applied Physics*, vol. 101, no. 9, p. 09A510, 2007. [Online]. Available: https://doi.org/10.1063/1.2710740

[107] Y. Zhou, J. Persson, S. Bonetti, and J. Ã…kerman, “Tunable intrinsic phase of a spin torque oscillator,” *Applied Physics Letters*, vol. 92, no. 9, p. 092505, 2008. [Online]. Available: https://doi.org/10.1063/1.2891058

[108] N. Oo and W.-S. Gan, “On harmonic addition theorem,” *International Journal of Computer and Communication Engineering*, vol. 1, no. 3, p. 200, 2012.

[109] G. D. Geronimo, P. Oâ™Connor, and A. Kandasamy, “Analog cmos peak detect and hold circuits. part 1. analysis of the classical configuration,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 484, no. 1, pp. 533 – 543, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168900201020599

[110] ——, “Analog cmos peak detect and hold circuits. part 2. the two-phase offset-free and derandomizing configuration,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 484, no. 1, pp. 544 – 556, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168900201020605

[111] W. Rippard, M. Pufall, and A. Kos, “Time required to injection-lock spin torque nanoscale oscillators,” *Applied Physics Letters*, vol. 103, no. 18, p. 182403, 2013.

[112] F. S. Samaria and A. C. Harter, “Parameterisation of a stochastic model for human face identification,” in *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on.* IEEE, 1994, pp. 138–142.

[113] D. Gabor, “Theory of communication. part 1: The analysis of information,” *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, 1946.

[114] D. E. Nikonov, I. A. Young, and G. I. Bourianoff. (2014) Convolutional networks for image processing by coupled oscillator arrays. [Online]. Available: https://arxiv.org/abs/1409.4469

[115] U. Signal, "Image processing institute," *USC–SIPI image database," aviliable on http://sipi. usc. edu/database.*

[116] D. Fernandez, L. Martínez-Alvarado, and J. Madrenas, "A translinear, log-domain fpaa on standard cmos technology," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 2, pp. 490–503, 2011.

[117] M. Gajek, J. Nowak, J. Sun, P. Trouilloud, E. O'sullivan, 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.

[118] W. H. Rippard, A. M. Deac, M. R. Pufall, J. M. Shaw, M. W. Keller, S. E. Russek, G. E. Bauer, and C. Serpico, "Spin-transfer dynamics in spin valves with out-of-plane magnetized coni free layers," *Physical Review B*, vol. 81, no. 1, p. 014426, 2010.

[119] J. M. Shaw, H. T. Nembach, and T. J. Silva, "Damping phenomena in co90fe10/ni multilayers and alloys," *Applied Physics Letters*, vol. 99, no. 1, p. 012503, 2011.

VITA

VITA

Minsuk Kooreceived the B.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2007, the M.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 2009, and is currently working toward the Ph.D. degree in electrical and computer engineering at Purdue University, West Lafayette, IN, USA. From 2009 to 2012, he was with RadioPulse Inc., Seoul, Korea as a senior engineer where he had been involved with the development of ZigBee transceiver and SoC products. His research interests include circuits and system for neural networks and associative computing using CMOS and emerging devices.