Carnegie Mellon University

CARNEGIE INSTITUTE OF TECHNOLOGY

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF Doctor of Philosophy

TITLE Neurocomputing and Associative Memories Based on Emerging Technologies: Co-optimization of Technology and Architecture

PRESENTED BY Vehbi Calayir

ACCEPTED BY THE DEPARTMENT OF

Electrical and Computer Engineering

ADVISOR, MAJOR PROFESSOR — DATE 9/17/2014

DEPARTMENT HEAD — DATE 9/24/2014

APPROVED BY THE COLLEGE COUNCIL

DEAN — DATE

Neurocomputing and Associative Memories Based on Emerging Technologies:

Co-optimization of Technology and Architecture

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Vehbi Calayir

B.S., Electrical and Electronics Engineering, Bilkent University
M.S., Electrical and Computer Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

September, 2014

Acknowledgments

All the praises and thanks be to Allah, the Lord of the Worlds!

I would like to first express my sincerest gratitude to my advisor, Prof. Larry Pileggi, for his excellent guidance throughout my graduate studies. This research would not have been possible without his continuous support, inexhaustible funding and exceptional vision. He has taught me many crucial things, among which are how to think analytically and critically, see through what other people think impossible, judge everything from both positive and negative sides, and not lose hope, especially when it comes to this line of work. “There is no black or white when you are doing research. There is always gray somewhere. That is what research is,” he has always reminded me. His remarkable comments and suggestions constitute a big portion of this interdisciplinary study and have helped me complete my PhD much easier than what I had initially thought.

Next I would like to thank my other thesis committee members, Prof. James Bain (Carnegie Mellon University), Prof. Jeffrey Weldon (Carnegie Mellon University), and Dr. George Bourianoff (Intel Corporation) for their significant help and feedback as well as accepting my invitation in the first place.

I would like to also appreciate my lab members for their helpful discussions and brilliant advices in addition to always forming a motivating atmosphere in the workplace. In particular, I thank David Bromberg, Renzhi Liu, Ekin Sumbul, Daniel Morris, Soner Yaldiz, Thomas Jackson, Ozan Iskilibli, Kaushik Vaidyanathan, Gokce Keskin, Umut Arslan, Curtis Ratzlaff, Andrew Phelps, Vanessa Chen, Bishnu Prasad Das, Qiuling (Jolin) Zhu, Ying-Chih Wang, Jinglin Xu, and Cheng-Yuan Wen.

Last but not least, I would like to express my special and deepest thanks to my family for everything in my life: my mother, Nuray Calayir; my father, Yusuf Calayir; my brothers, Enes and Muhammed Uveys Calayir; my little sister, Zeynep Betul Calayir; and my beloved wife, Zehra Calayir. I would also like to specially thank my brother, Enes Calayir, for being a perfect and elusive comrade as a PhD student here with me at Carnegie Mellon University. I would not have survived during my PhD without his invaluable support and companionship as well as his unexcelled brotherhood.

This work was supported in part by the National Science Foundation under contract CCF1146799 and contract CCF1318160, and a grant from the Semiconductor Research Corporation Nanotechnology Research Initiative. This work was also supported in part by the Systems on Nanoscale Information fabriCs (SONIC), one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP), a Semiconductor Research Corporation program sponsored by MARCO and DARPA. Finally, this work was supported in part by the Intelligence Advanced Research Program Agency and Space and Naval Warfare Systems Center Pacific under Contract No. N66001-12-C-2008. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Intelligence Advanced Research Program Agency and Space and Naval Warfare Systems Center Pacific.

Abstract

Neurocomputers offer a massively parallel computing paradigm by mimicking the human brain. Their efficient use in statistical information processing has been proposed to overcome critical bottlenecks with traditional computing schemes for applications such as image and speech processing, and associative memory. In neural networks information is generally represented by phase (e.g., oscillatory neural networks) or amplitude (e.g., cellular neural networks). Phase-based neurocomputing is constructed as a network of coupled oscillatory neurons that are connected via programmable phase elements. Representing each neuron circuit with one oscillatory device and implementing programmable phases among neighboring neurons, however, are not clearly feasible from circuits perspective if not impossible. In contrast to nascent oscillatory neurocomputing circuits, mature amplitude-based neural networks offer more efficient circuit solutions using simpler resistive networks where information is carried via voltage- and current-mode signals. Yet, such circuits have not been efficiently realized by CMOS alone due to the needs for an efficient summing mechanism for weighted neural signals and a digitally-controlled weighting element for representing couplings among artificial neurons.

Large power consumption and high circuit complexity of such CMOS-based implementations have precluded adoption of amplitude-based neurocomputing circuits as well, and have led researchers to explore the use of emerging technologies for such circuits. Although

they provide intriguing properties, previously proposed neurocomputing components based on emerging technologies have not offered a complete and practical solution to efficiently construct an entire system. In this thesis we explore the generalized problem of co-optimization of technology and architecture for such systems, and develop a recipe for device requirements and target capabilities. We describe four plausible technologies, each of which could potentially enable the implementation of an efficient and fully-functional neurocomputing system.

We first investigate fully-digital neural network architectures that have been tried before using CMOS technology in which many large-size logic gates such as D flip-flops and look-up tables are required. Using a newly-proposed all-magnetic non-volatile logic family, *mLogic*, we demonstrate the efficacy of digitizing the oscillators and phase relationships for an oscillatory neural network by exploiting the inherent storage as well as enabling an all-digital cellular neural network hardware with simplified programmability. We perform system-level comparisons of mLogic and 32nm CMOS for both networks consisting of 60 neurons.

Although digital implementations based on mLogic offer improvements over CMOS in terms of power and area, analog neurocomputing architectures seem to be more compatible with the greatest portion of emerging technologies and devices. For this purpose in this dissertation we explore several emerging technologies with unique device configurations and features such as *mCell* devices, *ovenized aluminum nitride resonators*, and tunable *multi-gate graphene* devices to efficiently enable two key components required for such analog networks – that is, summing function and weighting with compact D/A (digital-to-analog) conversion capability. We demonstrate novel ways to implement these functions and elaborate on our building blocks for artificial neurons and synapses using each technology. We verify the functionality of each proposed implementation using various image processing applications based on compact circuit simulation models for such post-CMOS devices.

Finally, we design a proof-of-concept neurocomputing circuitry containing 20 neurons using 65nm CMOS technology that is based on the primitives that we define for our analog neurocomputing scheme. This allows us to fully recognize the inefficiencies of an all-CMOS implementation for such specific applications. We share our experimental results that are in agreement with circuit simulations for the same image processing applications based on proposed architectures using emerging technologies. Power and area comparisons demonstrate significant improvements for analog neurocomputing circuits when implemented using beyond-CMOS technologies, thereby promising huge opportunities for future energy-efficient computing.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

As we reach the end of the CMOS roadmap researchers have started to intensively focus on post-CMOS technologies for circuit and system solutions. While there are several emerging technologies that offer great promise as CMOS alternatives, there are none that completely replace CMOS for the systems and architectures that are implemented today. For this reason it is important to explore system opportunities for emerging technologies as they evolve, since small block- or circuit-level comparisons to CMOS technology are unlikely to demonstrate any significant benefit. The greatest potential for an emerging technology will only be evident in terms of complete systems that are configured and optimized to exploit their unique features.

1.1 Neurocomputing and Associative Memories

Neurocomputing is considered an intriguing alternative to computing based on traditional techniques due to its brain-inspired massive parallelism. A neurocomputer attempts to mimic the human brain via a network of coupled artificial neurons that process information in parallel. Each brain neuron corresponds to a computational unit in a neurocomputer, and a connection between two artificial neurons represents a synapse that is connecting two brain neurons. The

strength of this synapse is the synaptic weight in a neurocomputer that relates one artificial neuron to another. In this way, neurocomputers operate analogously to the human brain.

Traditional computing schemes (variants of the von Neumann architecture) run a software algorithm for a specific application by sequentially executing each line in the instruction code. Even though each execution might take a very short time the overall computation efficiency is not that high due to the serial execution of instructions when compared to neurocomputing [1]-[3]. Instead, a neurocomputer performs pattern recognition via associative memory in a massively-parallel manner. It maps a set of input patterns to a set of output patterns via synaptic weights, whereby an output pattern can be retrieved for a given initial pattern. It therefore represents a powerful component for non-algorithmic, nonlinear, complex problems, and statistical information processing applications such as pattern recognition, image processing, and associative memory. These applications would otherwise require numerous memory fetch operations and a processor that is executing a list of commands for optimization.

For example, traditional associative memory circuits use *Manhattan distance (MD)* technique to find the best match among stored patterns in the system for a given noisy input pattern [4]-[5]. Since directly implementing MD is costly due to the required numerous subtraction and absolute value operations, binary bits representing the grey-scale pixels are first converted into thermometer bits, and then *Hamming distance* is applied to find the number of unmatched bit locations between that input pixel and the corresponding pixel in the stored patterns. However, such an approach consumes significant amount of memory resources because each memorized pattern has to be stored in the system as opposed to the neural networks (NNs) where the stored patterns are represented by synaptic weights among artificial neurons. This technique does not also provide functional flexibility that associative memories based on NNs offer because MD finds a deterministic match while NNs offer a statistical

approach which is useful for various image processing applications other than pattern matching as well.

We have modeled both MD and NN using a hardware description language (VHDL). We have mapped these models to standard logic gates using a logic synthesis tool (Synopsys Design Compiler). Figure 1.1 shows a hardware comparison of both associative memories that are implemented in 32nm CMOS. Although a computer would require less power to perform these functions, and we would never consider an MD implementation of the scale that would require 10's of kilowatts, this comparison does show how these two custom implementations would compare to each other in terms of power and area when built to perform the same functions. Apart from aforementioned implementation challenges MD requires 200x more devices and dissipates 700x more power for large-scale systems. This comparison does not consider the potential benefits of localized highly-distributed memory and computation that the NNs offer as opposed to a separate memory and processor.

1.2 Neurocomputing Circuits Using CMOS Technology

While neurocomputers offer a promising architecture for energy-efficient, future computing systems, their CMOS implementations have largely been viewed as impractical and significantly inferior to implementations based on more traditional CMOS architectures and memories due to the required circuit complexity and corresponding power consumption, thereby limiting the wide use of such architectures.

Oscillatory neural networks (ONNs), a prevalent example of phase-based neurocomputing, are particularly interesting systems since they mimic nature and the brain. But each neuron requires a voltage or current controlled oscillator and a means of programming the phase relationship among all neurons to represent the stored information. Importantly, ONNs rely on having connections (artificial synapses) among *all* artificial neurons – that is, the ability to

program the n^2 phase relationships among n artificial neurons. CMOS voltage-controlled oscillators (VCOs) are theoretically viable to represent oscillations, but completely impractical from an energy standpoint. Moreover, implementing the phase relationships among the VCOs is even more costly. For example, proposed ONN architecture based on phase-locked loops (PLLs) in Figure 1.2 [3] requires *one PLL for each artificial neuron and one variable gain amplifier (VGA) for each programmable synapse*. It would be impractical to power n PLLs and n^2 VGAs on a single CMOS chip for even a modest-size associative memory. To provide an analog control voltage to VGAs this architecture also necessitates *the use of digital-to-analog converters (DACs) to convert digital control bits for synapse programming to an analog voltage level*, further exacerbating the overall system performance.

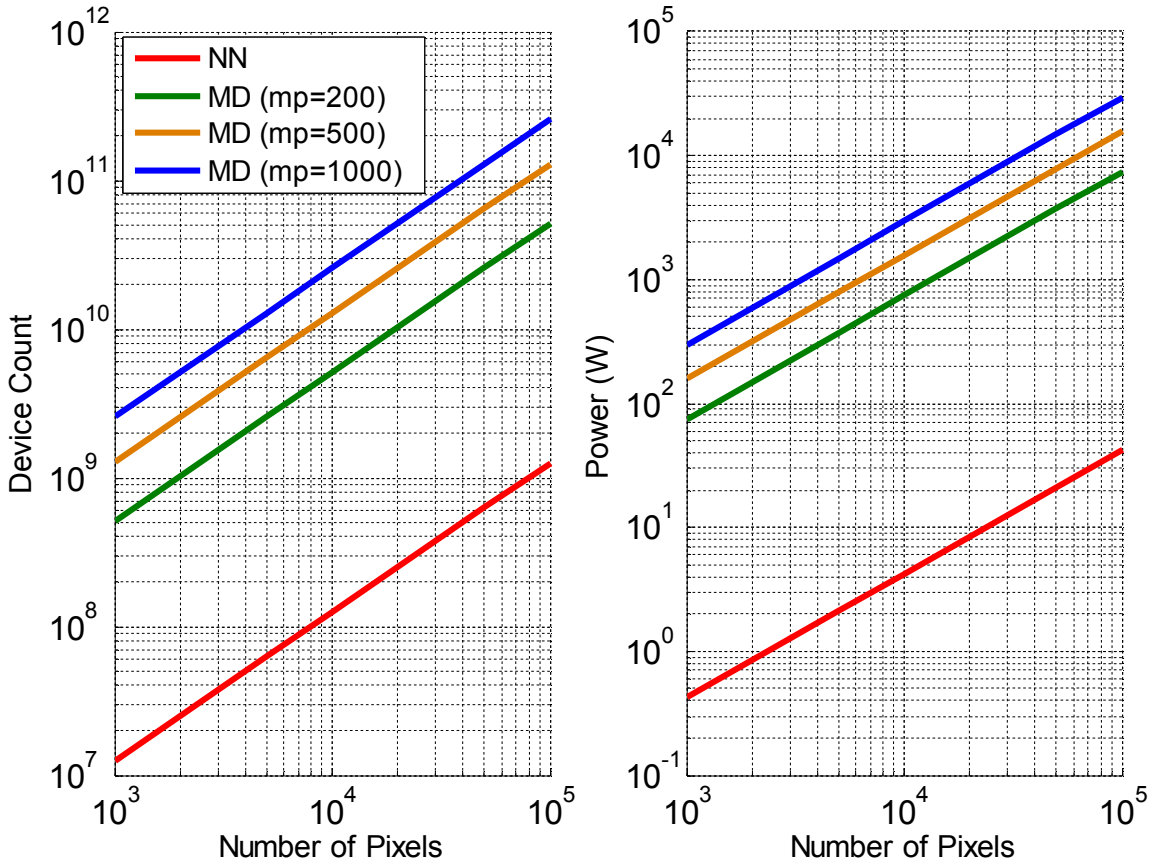

Figure 1.1. Hardware comparison of MD- and NN-based associative memory. *mp* denotes the number of memorized patterns.

While nascent ONN systems have severe implementation challenges, amplitude-based cellular neural networks (CNNs) offer more efficient circuits due to being a more mature technology. For CNNs there are training algorithms that have already been developed based on only *local connections* among just the nearest neighboring neurons. Independent of memory size or number of neurons, each artificial neuron is interconnected only to neighboring neurons within a specified radius. The radius can be extended for better accuracy such that all neurons share one-to-one connections like ONN architectures, but CNNs can also be implemented using the smallest possible radius that corresponds to only 9 connections to neighboring neurons. Even with so few connections though, CNN implementation of the neurons and synapses via CMOS is inefficient as compared to more traditional computing architectures.

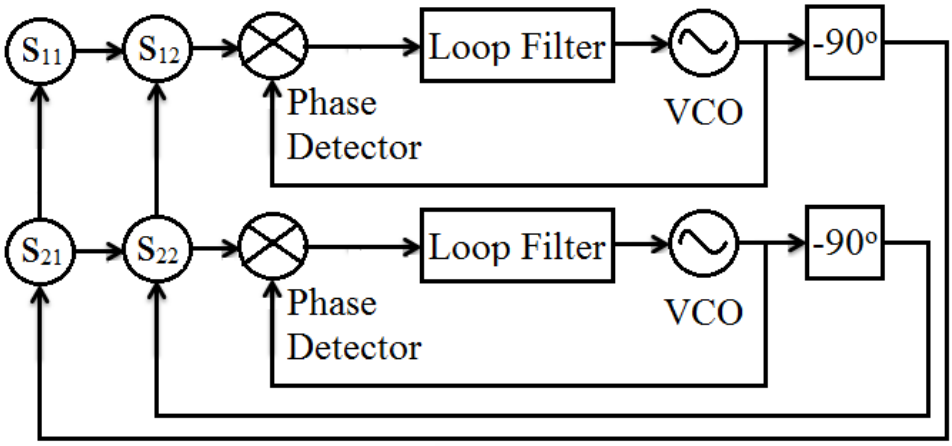

Figure 1.2. Conceptual architecture of PLL NNs for a 2-neuron system. VCO in this figure represents the voltage-controlled oscillator; and s_{11}, s_{12}, s_{21}, and s_{22} denote the corresponding programmable synaptic weights between artificial neurons that can be implemented with VGAs.

For example, analog implementations of CNNs have been attempted whereby a digital input for each programmable synapse is converted to a control voltage using a DAC and an analog variable-gain amplifier/multiplier [6]-[7]. Since each neuron in a CNN system has 9 artificial synapses, *nine DACs and nine analog amplifiers are needed for each artificial neuron*, which is

impractical for large-scale systems. Such implementations also lack the functional flexibility of neurocomputers, since artificial neuron circuits share the same synapse circuits using a cloning template for synaptic weights to compensate the high circuit complexity required for synapse circuits. Moreover, the best reported digital CNN implementation in CMOS [8] requires *32 D flip-flops and 155 mostly-large-size logic gates* (e.g., look-up tables, multiplexers, XORs, etc.) *for each neuron*, which is also not practical for large-scale systems.

1.3 Organization of Thesis

Although CMOS is inefficient for constructing neurocomputers, emerging technologies portend to offer new opportunities. However, the proposed implementations so far have either not addressed and solved all the performance issues that CMOS-based architectures already have, or posed new problems that require special circuit and/or manufacturing techniques. In this thesis we describe our novel approach to enable efficient and feasible neurocomputing circuits and associative memories. For this purpose we begin with a survey of previously proposed designs based on emerging technologies and show the issue(s) associated with each design in Chapter 2. Then we explain our design approach (i.e., co-optimization of technology and architecture) to deal with such issues, and provide the required device/technology specifications for building affordable neurocomputing systems. Next, using these specifications, we demonstrate our proposals for both digital and analog neurocomputing circuits and associative memories enabled by emerging technologies such as mLogic, mCell, ovenized aluminum nitride (AlN) resonator and graphene in Chapter 3, Chapter 4, Chapter 5 and Chapter 6, respectively. In these chapters we also provide circuit simulations and performance comparisons to CMOS technology using Verilog-A compact models for the corresponding devices all developed based on measurement data, and discuss device developments required for building practical systems based on such devices. We strongly believe our findings in these chapters will guide the beyond-CMOS device development for energy-efficient computing.

Chapter 7 presents the design and experimental results of a proof-of-concept CMOS chip in 65nm that emulates our proposed analog neurocomputing scheme. Chapter 8 concludes the thesis with brief summary and future considerations.

Chapter 2

Emerging Technologies for Neurocomputing

In this chapter we investigate the use of emerging technologies for a more efficient implementation of neurocomputing circuits. We start with an overview of recently proposed neurocomputing circuits and associative memories based on post-CMOS technologies, such as resistance change devices. We further evaluate the challenges and deficiencies associated with the deployment of these device technologies in actual neurocomputing systems. We then propose a generalized abstraction for a neurocomputing circuit that would be required to implement artificial neurons and synapses in a system context. We postulate how our system approach would be implemented, and outline the design features required for the device/technology for both digital and analog implementations.

2.1 Previous Work

Emerging technology devices offer new opportunities to efficiently construct *a tunable analog resistance* that could be used to implement programmable artificial synapses for

neurocomputers. Recently the use of specific memristor devices such as RRAM (resistive RAM) [9]-[10] and PCM (phase change memory) [11]-[13] (see Figure 2.1) has been shown to efficiently implement electronic synapses. Such voltage-controlled resistances are adjusted to have programmable synaptic weights. However, since these devices have only one control path for setting their resistances, using them as artificial synapses requires either *one DAC per synapse* to convert digital control bits into an analog control signal or *one digitally-controlled pulse generator for each synapse* to provide a pulse-based programming scheme for gradual resistance change [9]-[13]. In such implementations the neuron circuits would be implemented in traditional CMOS technology, which requires compatible monolithic integration with CMOS.

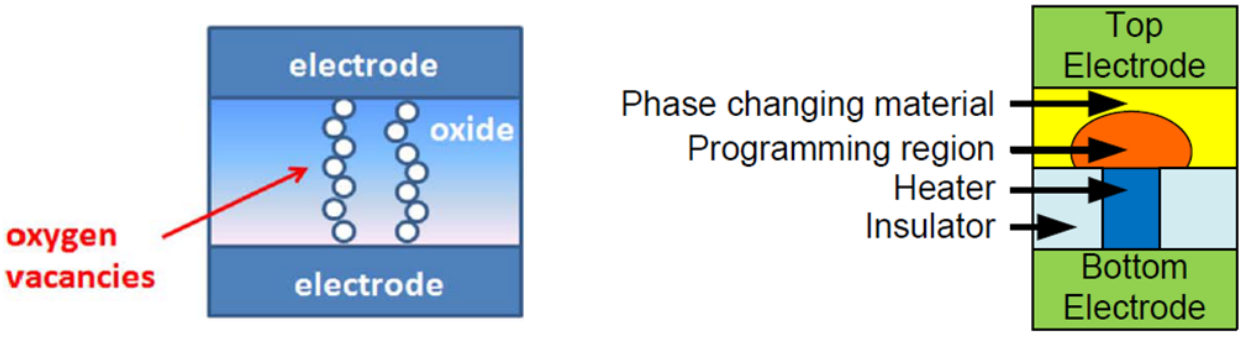

Figure 2.1. RRAM [9] (left) and PCM [13] (right).

Optically gated carbon nanotube field effect transistors (OG-CNTFETs) in Figure 2.2 have also been proposed to efficiently represent artificial synapses using a pulse-based programming for gradual resistance change [14]. Along with the aforementioned issues for two-terminal memristor devices, *OG-CNTFETs suffer from high resistance values ranging from MΩ to GΩ, thereby requiring high voltages* to create reliable/distinguishable currents through artificial synapses. For example, the proposed neurocomputing scheme in [14] uses a 7V supply voltage to generate only 10-40nA current, which would present design challenges with internal and external noise sources.

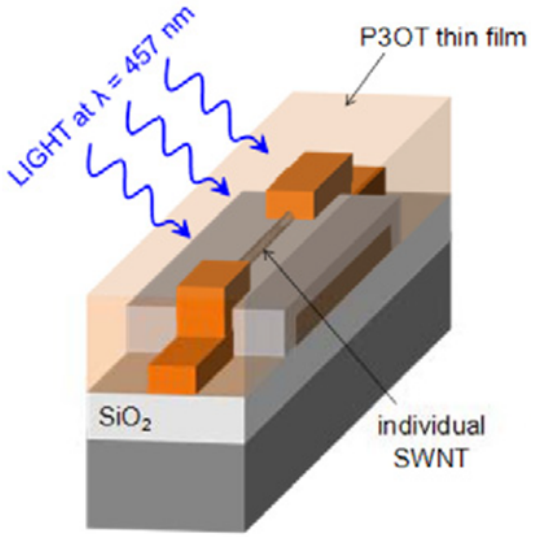

Figure 2.2. OG-CNTFET [14].

Another design based on memristors (e.g., titanium dioxide in Figure 2.3) [15]-[17] relies on a *memristor bridge synapse* consisting of 4 or 5 devices and a CMOS differential pair to convert voltage-mode signals to current-mode signals for an efficient summing of neural signals. The neuron circuit is implemented as a differential amplifier to sum these currents coming from neighboring neurons and generate an output voltage based on the summation. This implementation uses *wide and strong pulses to program memristors (synapse programming), but small and narrow pulses during neural evaluation* in order not to significantly alter resistance of memristors. This requires *complex pulse generation circuitry*, which is costly in terms of area and power. To prevent undesired resistance drift during neural evaluation, *a doublet generator* has been proposed [16] that results in further power and area consumption.

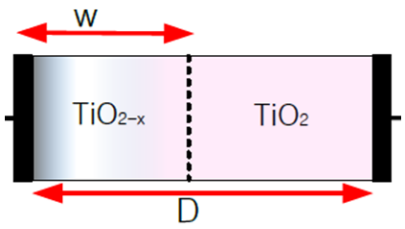

Figure 2.3. Titanium dioxide (TiO_2)-based memristor [17].

Spintronic devices have also been recently proposed to implement neurocomputers by exploiting spin properties of electrons to perform efficient computation [18]-[19]. Such circuits use domain wall magnets (DWMs) for constructing programmable synapses and magnetic tunnel junctions (MTJs) for implementing artificial neurons as shown in Figure 2.4. However, both neuron and synapse circuits still require *CMOS devices for converting spin-based signals into charge mode signals via latches and sending neural information through long-distance connections*. They also necessitate *one DAC for programming each artificial synapse and another DAC for applying initial inputs to each artificial neuron*. It is important to note, however, that domain wall based synapses would not be reliable for these types of applications since neural processing currents and/or thermal fluctuations and/or external magnetic fields can change the position of the domain wall inside the magnet during neural evaluation, thereby changing the corresponding synaptic weight and causing incorrect functioning.

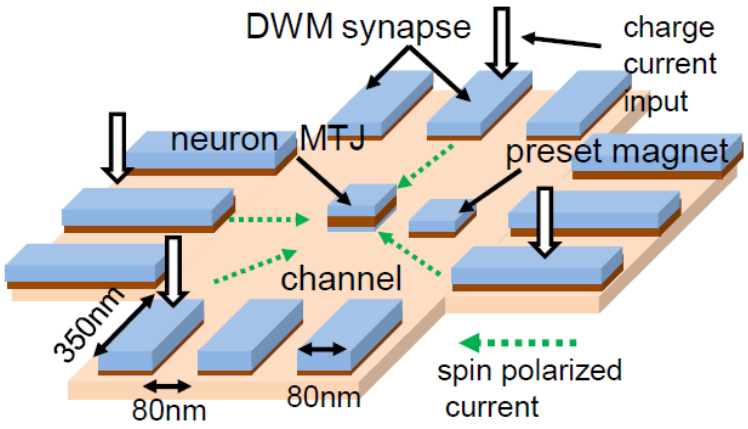

Figure 2.4. Spintronic neuron-synapse unit consisting of MTJ and DWM devices [18].

Nanoscale spin torque oscillators (STOs) depicted in Figure 2.5 offer single device oscillators to model the artificial neurons as well [20]-[21]. Such devices have been demonstrated to couple when implemented on a shared free layer for the magnetic spin [22]; however, efficiently implementing a programmable phase among all oscillators to represent the stored data is a

daunting challenge if not completely infeasible. *Local distances between neighboring neurons* have been claimed to represent such synaptic weights [21]. Yet, *an architecture based on using fixed distances as weights cannot be re-programmed once fabricated*, thereby eliminating the flexible functionality neurocomputers naturally enable. Moreover, a dc current is required to specify the STO oscillation frequency, thereby consuming significant stand-by power to store state. Several CMOS amplifying stages are also required to boost the tiny STO output power [23].

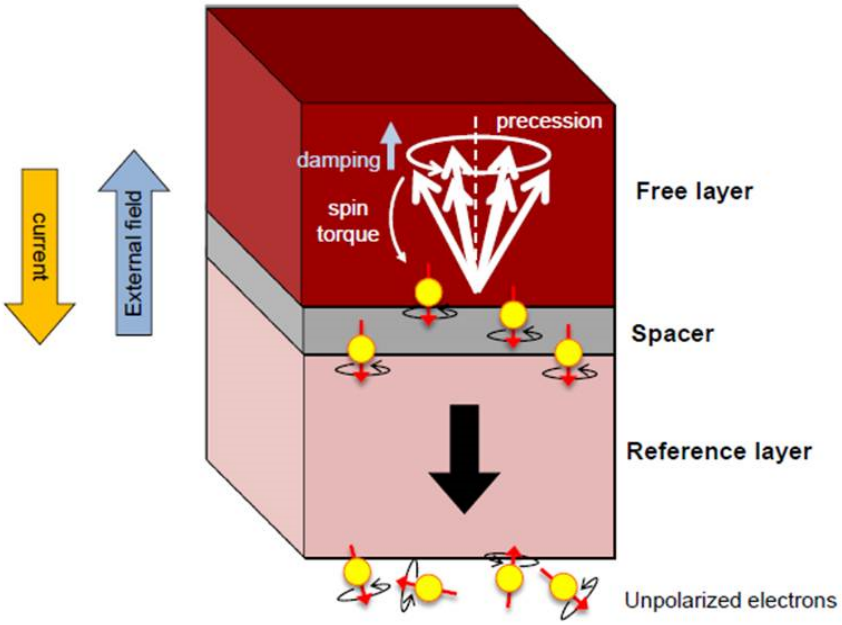

Figure 2.5. STO *(Courtesy of Jimmy Zhu's group from Carnegie Mellon University).*

Lastly, a neural network design has been demonstrated based on spin waves for information exchange [24]. Each artificial neuron is constructed as a magneto-electric cell (see Figure 2.6) that has only two magnetic polarization states. Therefore, *gray-scale image pixels are not possible with this neuron circuit*, thus limiting the functionality of overall neurocomputing system. *Couplings between the cells can be adjusted only by the direction and strength of a global magnetic field, so only a few functions can be performed once manufactured.* In

addition, extra circuitry is required to convert electrical signals to spin waves for the initialization of the neural network.

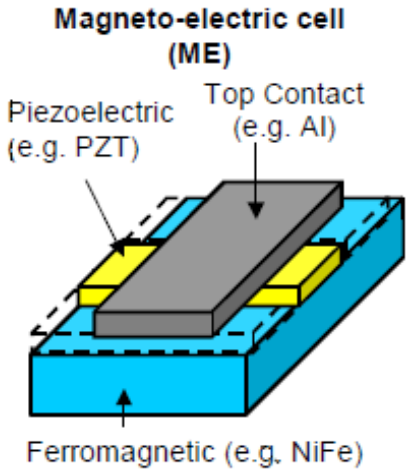

Figure 2.6. Magneto-electric cell [24].

2.2 Architecture Abstraction for Neurocomputing Based on Emerging Technologies

While proposed NN architectures based on emerging technologies that have been discussed in the previous section do not offer a complete, highly-efficient system solution for neurocomputing, co-design and co-optimization of future emerging devices and untried NN architectures could produce an efficient and practical neurocomputing system. Our co-optimization methodology consists of three key steps: i) determine device requirements for an efficient NN implementation, ii) explore emerging technologies and new device configurations based on these pre-determined requirements, and iii) develop novel neurocomputing systems using selected and/or proposed post-CMOS devices via technology-driven architecture optimization.

In this section we analyze a conceptual circuit diagram of a generalized neuron-synapse model that can be used for both feed-forward and recurrent (feedback) networks to explore what

device specifications are required to construct it efficiently and robustly. Figure 2.7 demonstrates a typical neuron-synapse model that has been used in both phase- and amplitude-based NNs [2]-[3], [6]-[8]. In this model the outputs of neighboring neurons are multiplied with corresponding synaptic weights. Then the results of these multiplications are accumulated in the neuron circuit to generate the neuron state. Finally, neuron circuit takes this summation and generates the output of the neuron based on an activation function that generally corresponds to a variant of the sigmoid function. Based on this model we now determine required device properties for both analog and digital implementations.

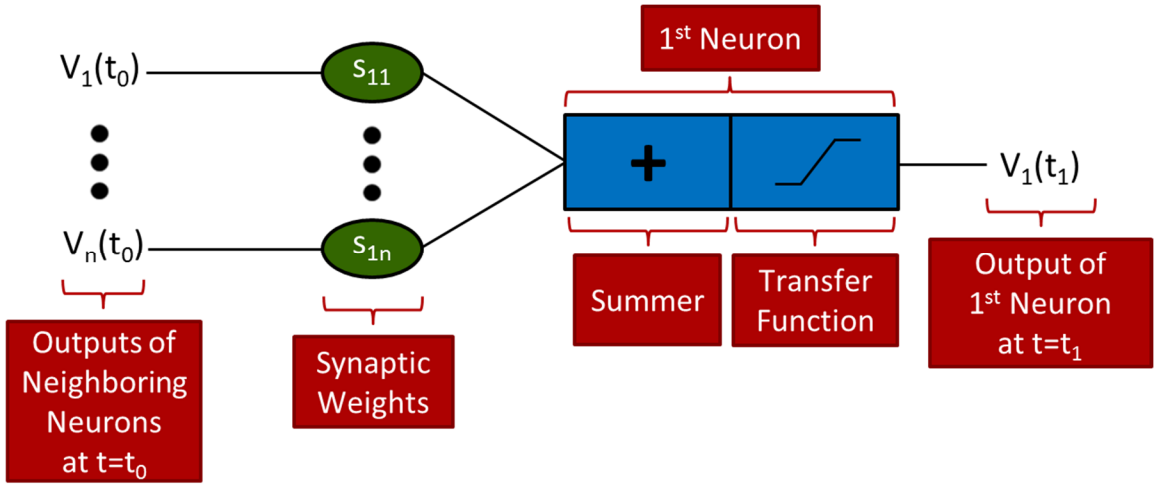

Figure 2.7. Conceptual architecture of a typical neuron-synapse model.

2.2.1 Digital Neural Networks

In a digital implementation both synaptic weights and neuron outputs are represented by binary bits (e.g., 4 or 5 bits [8], [25]). Multiplication of the neighboring neuron outputs by synaptic weights, summation of neural signals, and output generation based on this summation are done by standard digital logic gates. Therefore, a practical implementation of digital associative memories based on NNs requires *non-volatile* and *easily-programmable* logic families.

Non-volatile logic: Digital CMOS neurocomputing circuits consume a great deal of memory resources (e.g., D flip-flops, look-up tables and SRAM) for a sequential operation and storing data [8]. A non-volatile logic family provides storage at each logic stage, thereby eliminating the need for any extra memory element. By exploiting the inherent storage of logic state high throughput via deep pipelining can also be achieved [27].

Simplified programmability: Associative memories based on neurocomputing offer highly-distributed memory among small computational units in contrast to traditional computing schemes that require separate processor and memory. However, CMOS technology could not enable such a feature with today's technology. Therefore, the emerging technology device must provide easy, low-power re-configurability in order to enable highly-efficient digital associative memories.

2.2.2 Analog Neural Networks

In an analog implementation the neuron outputs are in voltage form, and the synaptic weights are represented as resistances that adjust the amount of current going into the neuron input from each neighboring neuron. The neuron circuit must efficiently sum these currents and then generate an output voltage based on that summation. Hence, an efficient implementation of analog associative memories based on neurocomputing requires all of the followings: i) an efficient tunable resistance element to represent artificial synapses; ii) an efficient means of converting between digital control data and analog computation engine; iii) an efficient summing mechanism for neural signals; iv) an efficient device with electrically-isolated input and output terminals; and v) an efficient conversion of the summed currents into an analog voltage level.

Tunable resistance element: A programmable analog resistance is required to represent artificial synapses. Artificial synapses are used to represent couplings between neighboring neurons based on the set of patterns that will be stored in the system. When implemented as a

tunable resistor the strength of these synapses can be gradually increased or reduced using a digital control data with 4-5 bits. Such resolution is sufficient enough for a hardware implementation of neurocomputing systems [2], [6], [8], [25]. Neurocomputers can also tolerate fuzziness on such resistances up to 15% [6], but still continue functioning correctly.

Efficient D/A (digital-to-analog) conversion: The control inputs for both synapse programming and neuron initialization are in digital form. Such digital data must be efficiently converted to a resistance value in a binary-weighted approach that is a typical feature of conventional DAC circuits. Hence, provided tunable analog resistance nature of an emerging technology device must be supported by such a property for an efficient NN implementation (i.e., digital control data applied to such a device must gradually alter its resistance in a binary-weighted fashion).

Efficient summing mechanism for neural information processing signals: As explained above, inputs to the neuron circuits coming from neighboring synapses are in current form. The device that is to be used for implementing artificial neurons must add such currents in a fast and compact manner without affecting the current level on artificial synapses.

Electrically-isolated input-output terminals: Since the input and output of the neuron circuits are in different forms (i.e., one is current, and the other voltage) they must not affect each other during neural evaluation. In other words, the output voltage must not impact the current level on artificial synapses, and currents coming from neighboring synapses must not leak into output path. This can be achieved by a device with electrically-isolated input-output terminals. This feature is also required to separate programming and reading paths of artificial synapses during neural evaluation (i.e., current passing through reading path of the device must not significantly affect its resistance). This would otherwise require special powering schemes and circuits, or tight integration with CMOS technology [9]-[19].

Efficient conversion mechanism between the summed neural signals and neuron output: Though an electrical isolation between the input and output paths of the device that will be used for efficiently building a neurocomputer is necessary, an efficient coupling mechanism between these two paths is required to convert the summation of neural signals into an analog output voltage using an inverter/buffer-like resistive divider (e.g., magnetic or thermal coupling). This provides a sigmoid-like transfer function for the neuron circuits.

An emerging technology that provides the aforementioned features in an efficient and compact way can enable a practical realization of an analog associative memory by representing each artificial neuron as a voltage divider with two tunable (pull-up and pull-down) resistors and each artificial synapse as a programmable resistor with 4 or 5 digital control bits.

2.3 Towards Highly-Efficient Associative Memories and Neurocomputing

Based on these requirements for both analog and digital architectures, we propose novel neurocomputing systems via a technology-driven architecture optimization in the following chapters that utilize various emerging technologies to create what might not be the definitive neural system, but could be a step toward a practical realization of a complete working system that could improve with future technology improvements and tuning.

It is important to note that both storage capacity and recognition accuracy of the proposed associative memories and neurocomputing architectures in the next chapters do not depend on the enabling devices and circuits, but on the selected training algorithms to calculate synaptic weights. Developing such algorithms, however, is not within the scope of this thesis. Moreover, the overall storage capacity can be further increased by hierarchical tree approach proposed in [20]. It is also possible to improve recognition capability of large neural systems by partitioning the network to process smaller parts of input patterns simultaneously (i.e., building smaller

associative memories in parallel) while still having the nearest neighbor coupling within each computing unit. In such an approach the final output pattern can be easily obtained by combining the pattern retrieval results from each partitioned unit.

Chapter 3

Fully-Digital Associative Memories and Neurocomputing

In this chapter we propose an approach for associative memories whereby the non-volatility offered by a newly-proposed magnetic logic family, *mLogic*, is exploited to construct fully-digital neurocomputing circuits with great efficiency [26]. This chapter first explains how to build mLogic circuits enabled by novel mCell devices, following a brief description for the structure and working mechanism of these devices. Second, it demonstrates our design approach for fully-digital NNs in application to CNN and ONN systems, along with related behavioral simulation results obtained from MATLAB. Last, it provides system-level comparison results with respect to 32nm digital CMOS implementations.

3.1 mCell: Non-Volatile Programmable Magnetic Device

The mCell device that has been recently proposed in [27]-[29] is based on modification of an existing STT-MTJ-based MRAM (magnetic RAM) device [30] to incorporate an electrically isolating but magnetically coupling layer [31] between the STT switching layer and coupled free

layer, as shown in Figure 3.1. With this added isolation layer it becomes a four terminal device comprised of a write-path ($w+$, $w-$) and an electrically-isolated read-path (R, R') (see Figure 3.1). The magnetic moments of the top magnetic electrodes are in the same direction and permanently fixed, acting as magnetic reference layer. The STT switching layer is composed of a magnetic metal connecting the bottom magnetic electrodes. For these electrodes, the magnetic moments at the opposite ends are oriented in the opposite direction by a pinning mechanism. A domain wall (i.e., a transition region of rotating magnetic moments) is formed due to this opposite magnetization in the bottom electrodes.

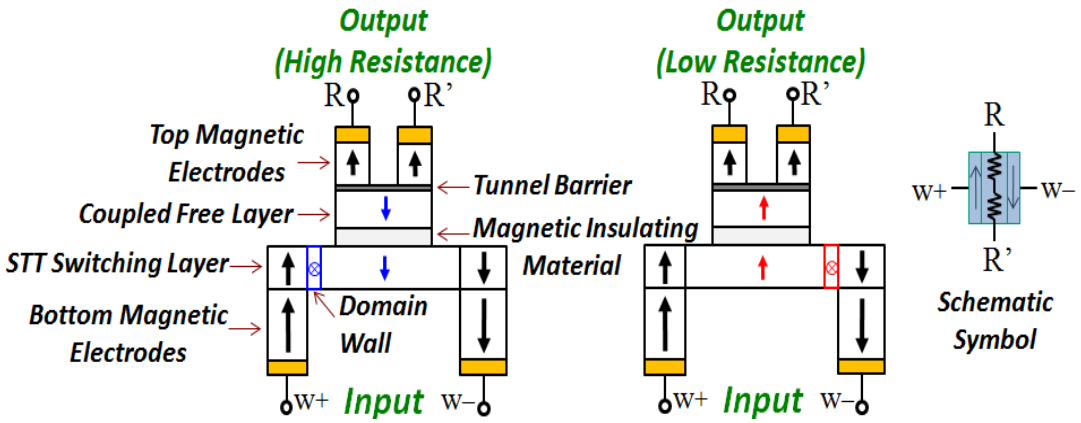

Figure 3.1. 2D cross section view of mCell for the write-path current direction from left to right (left) and right to left (center), and its schematic symbol (right).

The domain wall in the STT switching layer can be moved by sending a small current pulse through the write-path, utilizing an STT effect. This also programs the magnetization of coupled free layer via magnetic coupling between that layer and the STT switching layer. The domain wall has only two stable locations, as shown in Figure 3.1. As such, the magnetization of coupled free layer becomes only parallel or antiparallel to the magnetic moments of the top electrodes. When these two are in parallel, the read-path resistance is low and vice versa. This makes mCell work in two stable resistance states that are determined by both direction and pulse duration of the write-path current. These resistances are set by a non-volatile magnetic polarization and

therefore, remain the same when the device is powered off. With today's technology, the read-path resistance changes by only a factor of 2x [32] and has been best shown to change by 7x with special techniques [33]. It is expected that with improvements in magnetic devices and materials the high-to-low read-path resistance ratio will further increase, possibly surpassing what has been considered a 10x theoretical limit.

3.2 mLogic: Non-Volatile All-Magnetic Logic Technology

Although building non-volatile logic using mCells is challenging due to dynamic range for the read-path resistance as small as 2-3x with today's materials (e.g., 2.5kΩ/1.25kΩ), a complete mLogic family has been recently developed based on novel current-steering technique [27]-[29]. Pulsed supply voltages that synchronize and power the logic circuits provide current-based logic programming signals via resistor dividers. Hence, both input and output signals of mLogic gates are of currents, not voltages. The corresponding input and output logic levels are then determined by the direction of these currents (e.g., a positive current corresponds to logic '1' while a negative current corresponds to logic '0').

The mLogic NAND gate example implemented in a similar way to a CMOS NAND gate is illustrated in Figure 3.2 (left). A unique property of this logic technology is that inversion is free, since the read-path resistance of the mCell device is determined by the direction of its write-path current. For example, the *A* and *B* signals can be connected to the right terminal of the mCell devices in the pull-up network instead of connecting their inversions to the left terminal of those devices, as depicted in Figure 3.2 (center). Similarly, any other mLogic gate can be implemented without needing complementary inputs.

Moreover, a complementary mLogic gate can be obtained by simply interchanging the pull-up and pull-down networks, since the output logic level is determined by the resistance ratio of these two networks. In this way, for instance, the mLogic NAND gate in Figure 3.2 (center) can

be converted to an AND gate without requiring an inverter, as shown in Figure 3.2 (right). It is also important to note that inverters/buffers may still be used for signal buffering and restoration, or as delay elements in this logic family.

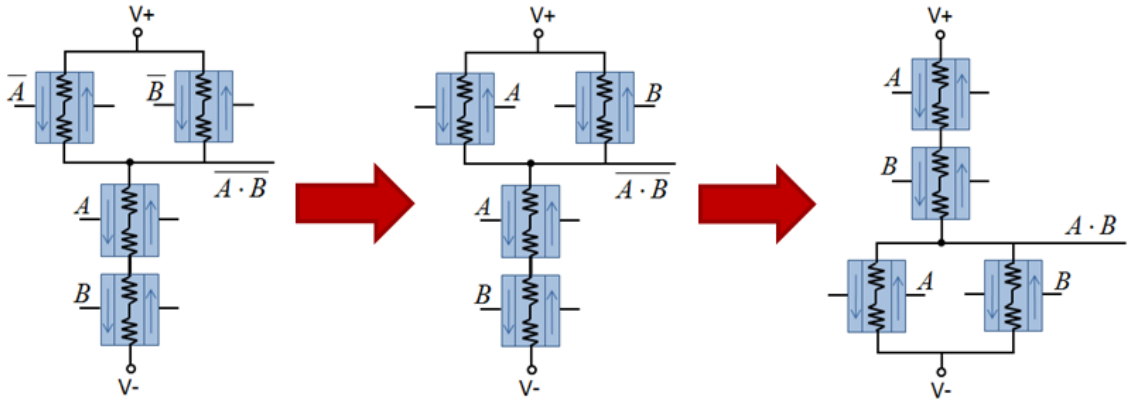

Figure 3.2. Traditional NAND (left), inversion-free NAND (center), and AND (right) gate examples using mLogic technology.

The vertical terminals of mCells (i.e., read-path) are characterized by two series MTJ resistances that are formed between the top electrodes and free layer via a tunnel barrier as shown in Figure 3.1. The horizontal path (i.e., write-path) for the programming current has small resistance as low as 120Ω [27]. This low input resistance represents the fanouts that are connected in series to form the current path that programs the states of mCells as shown in Figure 3.3. As such, each fanout mCell receives the same programming current, thereby preventing current shunting through unbalanced loads.

Even with only 2-3x read-path resistance change, with proper sizing and voltage levels the programming current signals can be properly steered to drive the logic states of the fanouts [27]-[29]. For example, referring to Figure 3.3, consider that the non-overlapping power clocks are positive ($pClk_{1,2}$+) and negative ($pClk_{1,2}$-) with respect to the ground shown at the end of the fanout chain. If the read-path resistance of mCell with the *A* input to the upward arrow is high resistance, and that to the downward arrow is low resistance, then the logic signal F would be a

current flowing from right to left in that figure. The direction of this current would program the states of mCells with the *F* input signal.

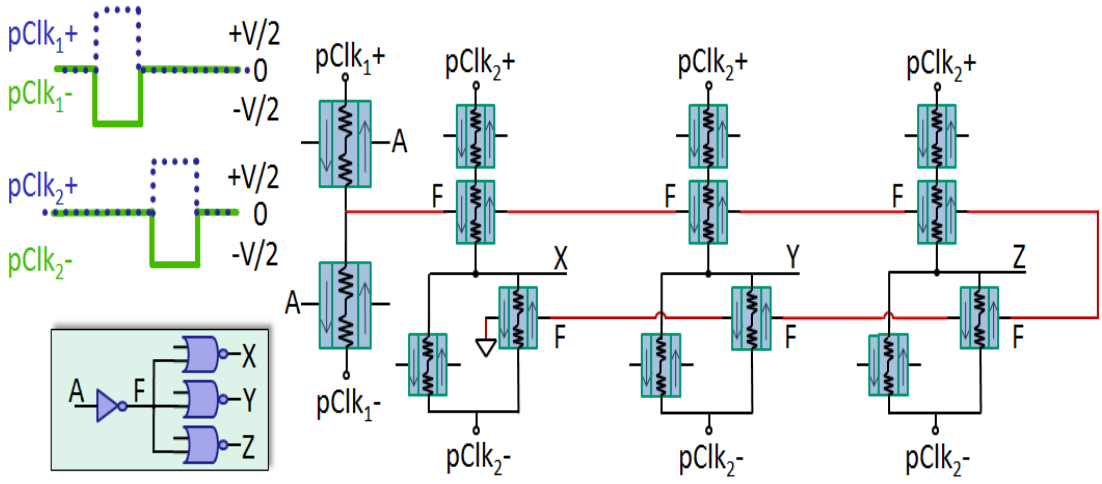

Figure 3.3. An inverter driving three two-input NOR gates. The fanout mCells are connected in series as highlighted by the red path. Two non-overlapping power clocks are applied to the driven and driving mLogic gates.

Moreover, efficient logic pipelining (see Figure 3.4) is enabled by the inherent storage of the mCell state and the non-overlapping power clocks [27]. In each pipeline stage the mLogic gates marked by *P1*, and those marked by *P2* are clocked with the non-overlapping $pClk_1$+/- and $pClk_2$+/- shown in Figure 3.4, respectively. This results in further power reduction for our proposed fully-digital associative memories in the next sections because only some of the mLogic gates will dissipate power at each time interval.

It is possible to integrate CMOS transistors with mCells to form a hybrid non-volatile logic family, but this eliminates some of the aforementioned energy-saving benefits. Moreover, there are integration issues that make this costly and impractical, since tightly integrating mCells and mLogic on top of CMOS would consume routing resources that are needed for all of the artificial synapses in our proposed fully-digital associative memory architectures. Instead, we envision a

fully-functional mLogic chip that can be layered or stacked (using through-silicon vias) with CMOS only for power, clocking and input/output (I/O).

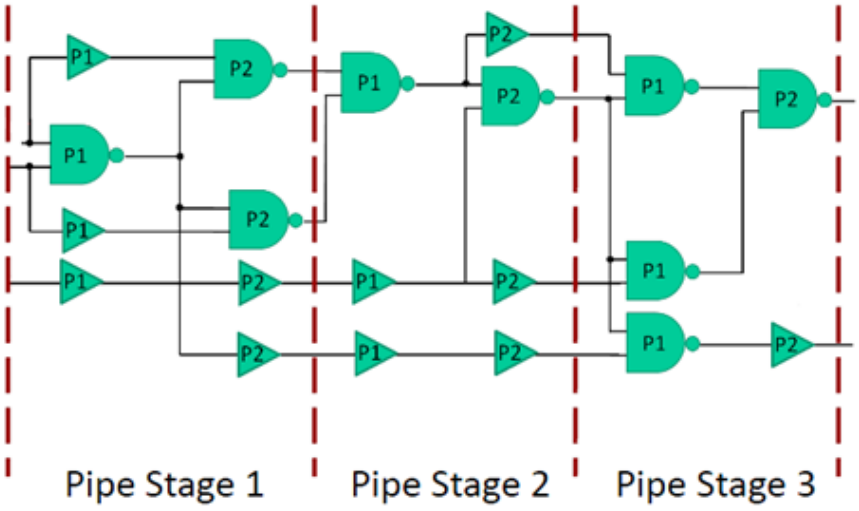

Figure 3.4. Pipelined mLogic gate stages [27]. Each pipeline stage is divided into two sub-stages. The mLogic gates in the first sub-stages and second sub-stages are clocked with two non-overlapping power clocks for power savings.

3.3 Proposed Fully-Digital ONN Architecture

ONNs are particularly interesting systems since they mimic nature and the human brain. In ONNs, memorized patterns are synchronized oscillatory states in which neurons periodically communicate with each other according to certain relations between their phases. This interaction between neurons is based on phase modulation (PM) encoding. The PM encoding neuron changes its firing pattern (i.e., at which time within the cycle the firing occurs) to represent its state. It is theoretically shown that ONNs can be built using PLLs [3], laser oscillators [35], and microelectromechanical system (MEMS) oscillators [36].

3.3.1 Derivation of Discrete-Time ONN Dynamics

Referring to the PLL-based NN shown in Figure 1.2, its dynamics is given by [3]:

$$\dot{\theta}_i = \Omega + V(\theta_i)\sum_{j=1}^{n} s_{ij} V(\theta_j - \pi/2) \tag{4.1}$$

where θ_i is the phase of the i[th] VCO, $\Omega >> 1$ is the natural frequency of the system, s_{ij}'s are synaptic weights, and V is the VCO output signal. Since $\Omega >> 1$, we can average Eq. 1 over time, which yields:

$$\dot{\theta}_i = \Omega + \sum_{j=1}^{n} s_{ij} H(\theta_j - \theta_i) \tag{4.2}$$

where the averaged transfer function H is:

$$H(\theta_j - \theta_i) = \lim_{T\to\infty} \frac{1}{T} \int_0^T V(\theta_i) V(\theta_j - \pi/2) dt \,. \tag{4.3}$$

Various types of transfer functions are given in [3] based on the different PLL output waveforms. Since we discretize the whole system, any of them could be selected; however, transfer function corresponding to the rectangular waveform is the easiest one to discretize (see Figure 3.5). This transfer function has only three stable points at θ = 0, π, and –π. Therefore, synchronization is achieved only if all oscillatory neurons are in-phase or anti-phase with respect to each other.

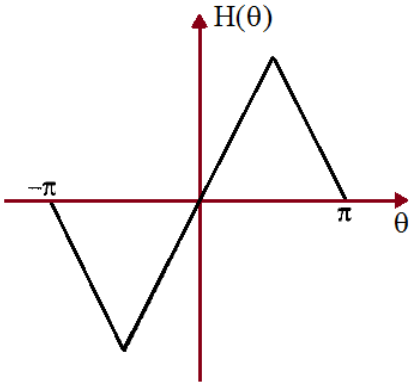

Figure 3.5. Transfer function for fully-digital ONN.

Next we define excess phase as:

$$\phi_i = \theta_i - \Omega t \,. \tag{4.4}$$

Plugging (4.4) into (4.2) results in:

$$\dot{\phi}_i = \sum_{j=1}^{n} s_{ij} H(\phi_j - \phi_i) \,. \tag{4.5}$$

Discretizing the continuous-time ONN system in (4.5) gives the discrete-time ONN dynamic equation as:

$$\phi_i[k+1] = \phi_i[k] + \sum_{j=1}^{n} s_{ij} H(\phi_j[k] - \phi_i[k]) \tag{4.6}$$

where k is the iteration step. The training algorithm is applied to specify synaptic weights that represent the patterns that are stored in this system. Solutions to this difference equation then correspond to these stored patterns.

3.3.2 Implementation of Fully-Digital ONN

The proposed fully-digital ONN architecture shown in Figure 3.6 is the representation of the discrete-time ONN dynamic system in (4.6). One adder without a reset signal is needed to accumulate excess phase. The transfer function block in Figure 3.6 is the realization of H function drawn in Figure 3.5. u_1 and u_2 are the initial inputs for the first and second neurons, respectively.

Most importantly, efficient implementation of this discretized ONN system is enabled by the inherent storage of the non-volatile logic family that allows us to *digitally* represent oscillations in phase domain instead of converting phase-domain computations into voltage domain. With local storage of state we can easily track the accumulation of excess phase, thus transforming the

oscillator's function into digital domain. Moreover, phase relationships among artificial neurons are discretized via non-volatile storage by digitally representing the transfer function shown in Figure 3.5.

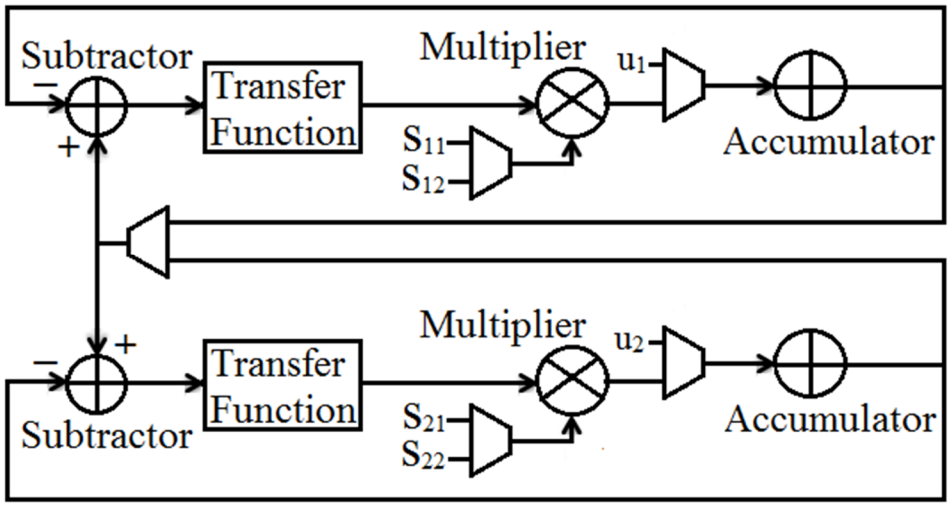

Figure 3.6. High-level circuit representation of the discrete-time ONN. The signal-selecting blocks are multiplexers.

The non-volatile logic with the inherent storage of state offers additional advantages: i) easy *programmability* without the need for extra memory; and ii) high throughput via a *fully-pipelined* architecture without the need for D flip-flops [27]. With inherent pipelining and simplified programmability, only a few hundred non-volatile logic cells are required to model each oscillatory neuron. With improvements in training algorithms for ONNs, even fewer logic gates would be needed. For example, the training algorithm proposed in [37] that can be potentially applied to any NNs used for associative memories is claimed to render implementation of sparsely-interconnected NNs possible. If it can be successfully applied to oscillatory associative memories as well, it will enable sparsely-interconnected ONNs, thus further simplifying the proposed ONN implementation.

3.3.3 Behavioral Simulation

To validate the functionality of our discretized ONN system we generate a high-level behavioral model with MATLAB based on our proposed architecture that is shown in Figure 3.6. This model is formed by direct implementation of each individual block (i.e., multiplier, multiplexer, accumulator, subtractor, and transfer function) in this proposed architecture. Using pattern recognition example from [3], the 60-pixel bit patterns shown in Figure 3.7 are stored in the system via synaptic weights. We use *Hebbian Rule* to calculate these weights as described in [3].

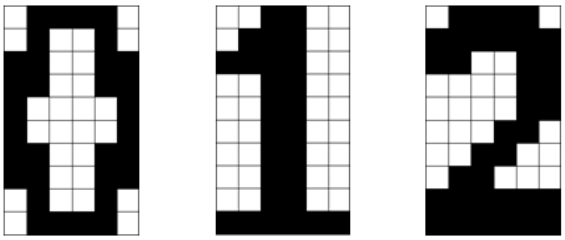

Figure 3.7. 60-pixel memorized bit patterns. They are stored in the neural network via synaptic weights.

In our model we represent inputs and synaptic weights as 5-bit signed binary numbers. Therefore, 31 possible phase levels can be defined in this system. Moreover, number of step size to recognize the pattern would alter with different choices of representing data and how higher order bits generated during neural computation are handled.

Figure 3.8 shows our first example with an input pattern that is generated as close to that used in [3] as possible. Eight steps are required to fully recognize the pattern. In this example, oscillatory associative memory gets locked to one of the stored patterns *as designated in the example in [3].*

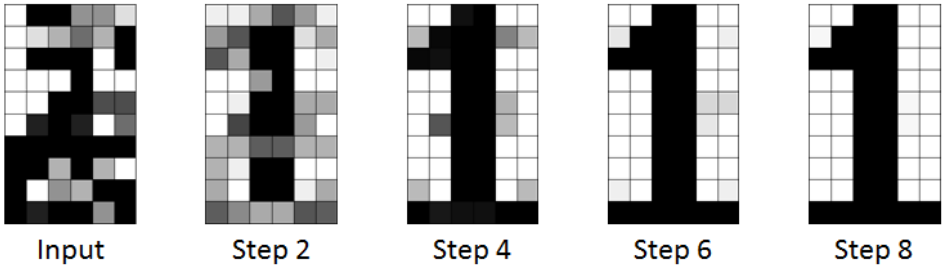

Figure 3.8. Pattern recognition process (Example 1). The initial input pattern is the distorted version of the bit pattern '1'.

Figure 3.9 demonstrates our second example using a different input pattern. Seven steps are required to fully retrieve one of the stored patterns. These examples verify that our proposed ONN system functions correctly, and does not lose any significant data due to discretization of system dynamics.

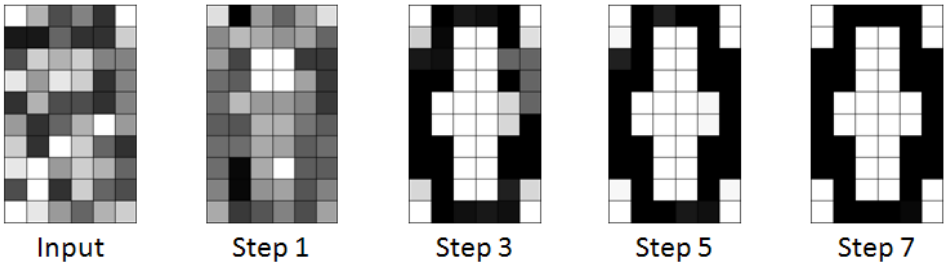

Figure 3.9. Pattern recognition process (Example 2). The initial input pattern is the distorted version of the bit pattern '0'.

3.3.4 System-Level Comparison to 32nm CMOS Technology

We constructed our proposed ONN architecture in Figure 3.6 with CMOS and mLogic based on circuit simulation modeling for both. We assumed a fully-interconnected 60-neuron system to comply with behavioral simulation presented in the previous sub-section. Similarly, 5-bit signed binary numbers were chosen for inputs and synaptic weights.

For the CMOS implementation we first generated a VHDL (VLSI hardware description language) based model for an oscillatory neuron and then compiled it using a logic synthesis tool (Synopsys Design Compiler) with 32nm CMOS logic library that completed the circuit optimization based on timing targets we provided. We verified the implementation of the same ONN architecture based on mLogic with +/-12.5mV supplies for buffers and multiplexers, and +/-50mV supplies for other logic gates using a circuit simulation with a physics-based device model for mCells [27]-[28].

In both implementations each neuron consists of two major blocks: i) a computation block (CB) that performs neural computation; and ii) an interconnection block (IB) that stores synaptic weights and inputs them to the computation block in an orderly manner. With our 60-neuron system example for each neuron the CMOS implementation with 0.7V supply requires 2168 transistors and 87.4μW power for CB, and 17208 transistors and 581μW power for IB. Nevertheless, for the same system the mLogic implementation per neuron requires 2238 mCell devices and 573μW power for CB, and 1200 mCell devices and 33μW power for IB. Hence, for a 60-neuron system the mLogic implementation represents a 5.6x improvement in the number of devices required while consuming comparable power as the CMOS implementation. With increase in the number of neurons, however, interconnections would dominate due to the fully-interconnected architecture. Hence, for large-scale oscillatory associative memories the improvement with mLogic would reach approximately 15x for the number of devices required and 18x for power as shown in Figure 3.10. As explained in Section 4.3.1 the device count comparison fairly represents the area comparison, since nanoscale mCell and CMOS transistor occupy approximately the same area. Moreover, these comparisons do not consider the potential benefits of layout and stacking of this all-magnetic logic technology.

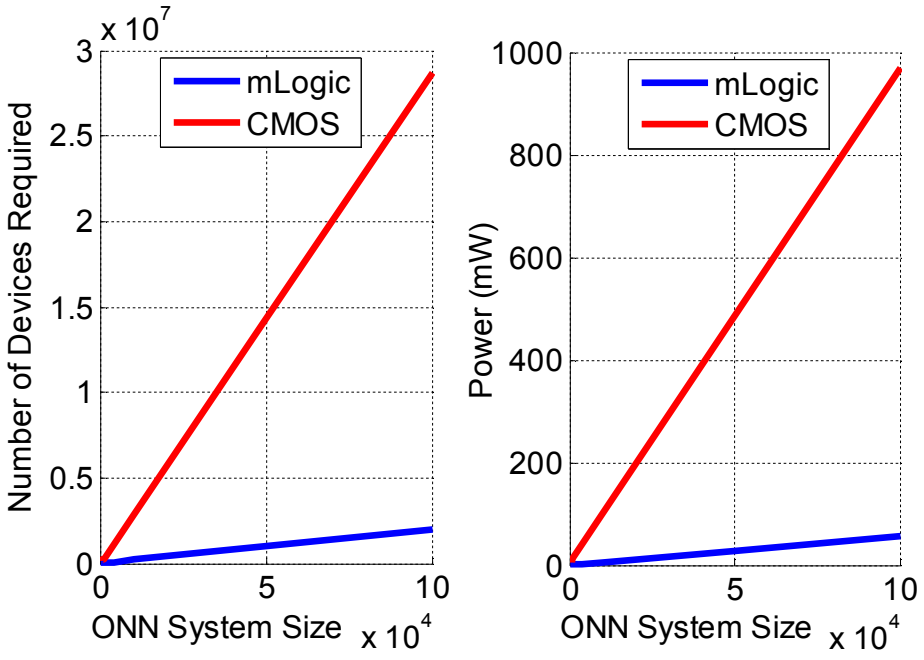

Figure 3.10. mLogic vs CMOS comparison per neuron for different ONN system sizes: Device count comparison (left) and power comparison (right).

3.4 Proposed Fully-Digital CNN Architecture

The CNN that was first proposed by L. O. Chua and L. Yang in 1988 [38] is a special class of NNs as it offers only local interconnections among artificial neurons. Regardless of the number of neurons in the CNN system, each neuron is connected to only neighboring neurons within a specified radius r and itself. For example, referring to a 2D CNN architecture shown in Figure 3.11, for r=1 the red cell in the center interacts only with the blue cells and itself, whereas it interacts with the white cells as well for r=2. This radius can be extended for better accuracy such that all neurons share one-to-one connections like other NN systems, but CNNs can be built for more efficient hardware implementations using the smallest possible radius that corresponds to only 9 connections for each neuron.

In CNNs each cell has an input, a state and an output, and interacts with the cells within its neighborhood r according to certain relations between their amplitudes. This interaction

between the CNN cells is based on amplitude modulation (AM) encoding. The AM encoding neuron changes its firing rate (i.e., the magnitude of its response over one cycle) to represent its state.

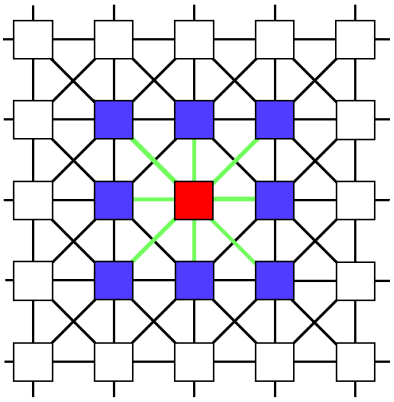

Figure 3.11. A 2D CNN illustrating neighboring cells for a CNN cell when r=1.

3.4.1 Derivation of Discrete-Time CNN Dynamics

The conventional dynamic equation of a continuous-time CNN is given by:

$$\dot{x}_i = -x_i + \sum_{j \in Neighbors\ of\ i} s_{ij} G(x_j) + \sum_{j \in Neighbors\ of\ i} b_{ij} u_j + I \tag{4.7}$$

where x_i is the state of the ith CNN cell, s_{ij}'s are feedback coefficients, b_{ij}'s are control coefficients, u_j is the input of the jth CNN cell, I is the bias term, and the activation function G is usually the saturation function defined as:

$$G(x) = \begin{cases} x & if\ -1 \le x \le 1 \\ 1 & if\ x > 1 \\ -1 & if\ x < -1 \end{cases} . \tag{4.8}$$

We convert this continuous-time CNN dynamic equation in (4.7) into the discrete-time CNN dynamic equation as follows:

$$x_i[k+1] = \sum_{j \in Neighbors\ of\ i}^{n} s_{ij} G(x_j[k]) + I_i \tag{4.9}$$

where k is the iteration step and I_i, the bias term corresponding to the ith CNN cell, is given by:

$$I_i = \sum_{j \in Neighbors\ of\ i} b_{ij} u_j + I\,. \tag{4.10}$$

The training algorithm is applied to specify both feedback and control coefficients that represent the patterns stored in this system. Solutions to the difference equation in (4.9) then correspond to the stored patterns.

3.4.2 Implementation of Fully-Digital CNN

Using the discrete-time CNN dynamics in (4.9) we construct our proposed fully-digital CNN system. Its conceptual diagram consisting of two neurons is shown in Figure 3.12. The signal-selecting blocks in that figure are multiplexers. Most importantly, the implementation of this discretized CNN system is enabled by the non-volatile logic. The inherent storage of logic state offers simplified *programmability* and highly-efficient *pipelined* stages, thereby eliminating the need for look-up tables and D flip-flops. As such, only a few hundred mLogic gates would be required to model each artificial neuron.

3.4.3 Behavioral Simulation

As an illustrative example we have generated a high-level behavioral model for the proposed CNN architecture in MATLAB, assuming zero bias terms for all artificial neurons. This model consists of individual implementations of each block shown in Figure 3.12. Similar to pattern recognition examples in Section 3.3.3 we use pattern recognition example demonstrated in [3]. The feedback coefficients are calculated based on the 60-pixel memorized bit patterns shown in Figure 3.13 using the synthesis procedure as described in [39].

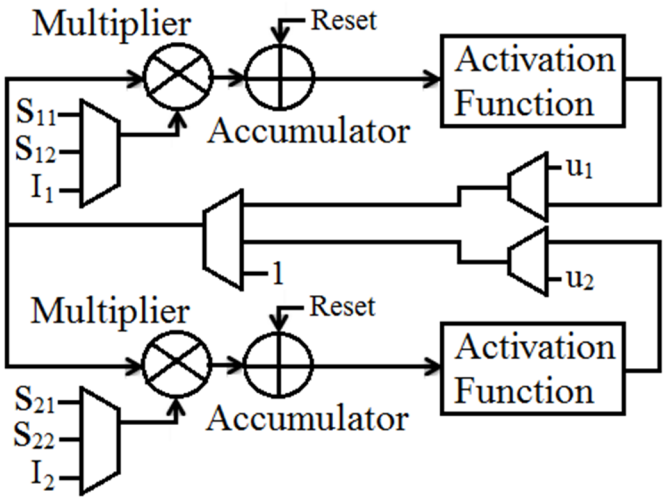

Figure 3.12. High-level circuit representation of the discrete-time CNN.

In this model 5-bit binary numbers are assumed to be used for inputs and feedback coefficients. As such, 31 possible gray-scale color levels can be defined in the model. Moreover, the number of steps to retrieve one of the stored patterns would alter with different choices of representing data and how higher order bits that are generated during neural computations are handled.

For our first pattern recognition example in Figure 3.13 we use the same input pattern given in the example in [3]. The generated CNN system is locked to the stored bit pattern '1' at three steps *as in the case in [3].* In our second pattern recognition example we use the distorted input pattern '0'. This time seven steps are required to recognize the pattern. All the results presented here comply with the results obtained for a fully-digital ONN in Section 3.3.3 with exception of time evolution of the retrieved output patterns, thereby confirming correct functionality of our proposed fully-digital CNN system.

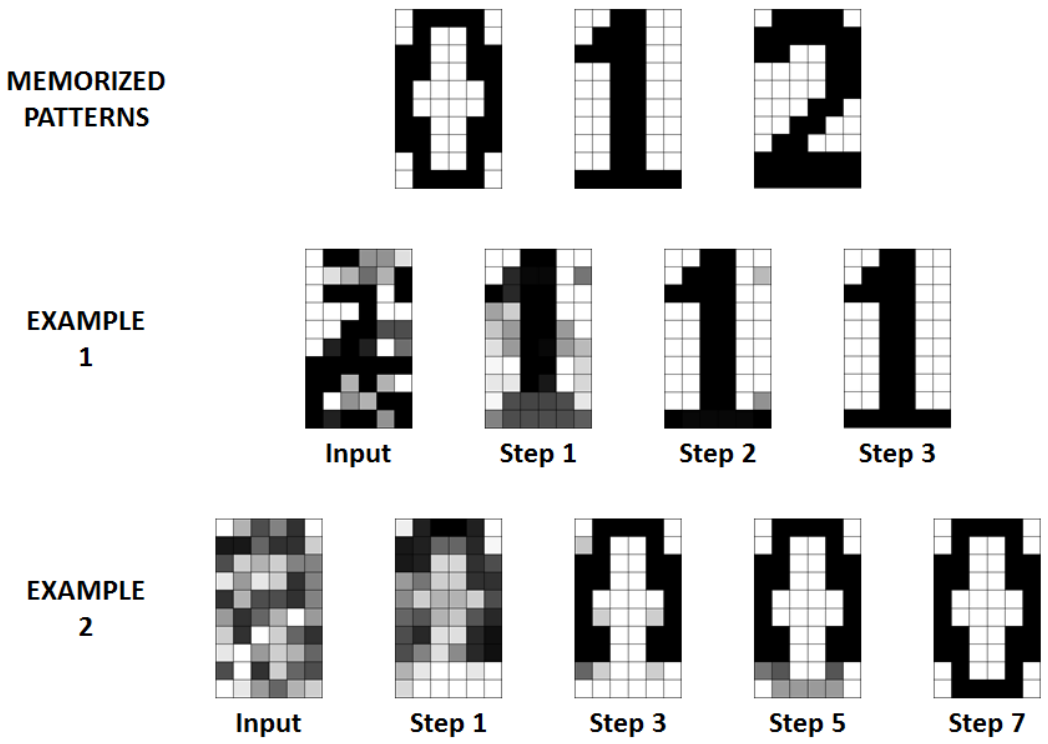

Figure 3.13. Memorized bit patterns and pattern recognition examples for our proposed fully-digital CNN system.

3.4.4 System-Level Comparison to 32nm CMOS Technology

We constructed our proposed fully-digital CNN architecture shown in Figure 3.12 using mLogic technology and compared it with the best-reported CMOS-based digital CNN topology [8] that is implemented here using logic gates for comparison purposes. This CNN system implemented in CMOS leverages a novel multiplication scheme that significantly reduces the number of CMOS transistors required. We used 5-bit binary numbers for inputs and feedback coefficients while assuming zero bias terms for all artificial neurons. The neural networks were built using the smallest possible interconnections among artificial neurons – that is, each neuron has only 9 connections.

For the CMOS implementation we first generated a behavioral model for the best-reported CNN hardware in [8] using a hardware description language (VHDL). Then we compiled it with

32nm CMOS standard logic library using a logic synthesis tool (Synopsys Design Compiler) that performed a circuit optimization based on our timing constraints. We used a 0.7V supply voltage, and a 500MHz clock for synchronization and pipelining.

For the mLogic implementation we used the Verilog-A model of the mCell device that was written in SPICE for circuit simulations using a first-order approximation of the underlying physics [27]-[28]. We applied +/-10mV power supplies for buffers and +/-25mV power supplies for the remaining mLogic gates. Again we used the same clock frequency of 500MHz as in the CMOS implementation.

Our system-level experiment demonstrates that each neuron requires 12606 transistors with a power dissipation of 423.5μW using 32nm CMOS technology; and 2269 mCell devices with a power dissipation of 271.3μW using mLogic technology. Since the complexity of each CNN cell is determined by the specified neighborhood, the CNN system size does not affect performance values we have provided above. With increased radius of neighborhood mLogic would provide about one order of magnitude improvement in both power and area when compared to digital CMOS implementations. Moreover, these comparison results again do not consider the potential benefits of layout and stacking of this all-magnetic logic family.

3.5 Summary

This chapter focuses on the design of fully-digital neurocomputing circuits (ONN and CNN) using non-volatile logic technologies such as mLogic. System-level comparisons to 32nm CMOS indicate one order of magnitude improvement in both area and power consumption for both systems. Although further development and scaling of such new non-volatile logic technologies can yield additional improvements as opposed to mature CMOS that has almost reached its scaling limits, analog neuromorphic circuits based on emerging devices and technologies might

offer much better performance by exploiting their unique features that CMOS transistors could not enable.

Chapter 4

All-Magnetic Analog Associative Memory and Neurocomputing

In this chapter we propose the use of newly-proposed mCell devices described in the previous chapter for our analog neurocomputing architecture that is suitable for efficiently building programmable artificial neurons and synapses [40]. To this end this chapter first describes why mCells could be efficient for such purposes, and then demonstrates our proposed neurocomputing architecture using mCells only. Finally, it evaluates the performance of this architecture by means of circuit simulations, and provides direction and targets for future device development to enable feasible neurocomputing systems based on such spin-based devices.

4.1 mCells for Neurocomputing

The use of the mCell devices for building an analog associative memory offers great promises. The easy-yet-efficient programming of such magnetic devices enables easy reconfigurability of artificial synapses and efficient initialization of artificial neurons without requiring DACs. Small write-path resistances and current-based programming allow efficient

summing of currents coming from neighboring synapses for the neuron circuits. The electrically-isolated programming and reading paths provide simplified conversion between current-mode and voltage-mode signals in the neuron circuits, thereby resulting in highly-efficient neural information processing. This feature of the mCell devices also prevents the strength of artificial synapses from altering during neural evaluation without requiring any special powering techniques. Local storage offered by these devices allows the non-volatile storage of the synaptic weights, thus eliminating the need for any memory elements to store data. The non-volatility allows neural information to remain stored even when mCells are powered off.

4.2 Proposed Neurocomputing Architecture Using mCells

The conceptual circuit model for an artificial neuron based on mCells is depicted in Figure 4.1. In this model, the neuron circuit that is implemented as a combination of magnetic buffers connected in parallel consists of two main components: i) *excitation* and ii) *inhibition*. For example, when the neuron circuit used for image processing represents an image pixel that is closer to a *black* pixel, the excitation and inhibition components generate positive and negative voltages (neuron outputs) with equal amplitudes, respectively. On the contrary, when the neuron circuit represents more like a *white* image pixel, the excitation component produces a negative voltage while the inhibition component produces a positive voltage. The amplitude of this neuron output signal then represents how close the corresponding image pixel is to a black or white pixel.

The synapse circuit is constructed as a number of binary-weighted mCells connected in parallel as shown in Figure 4.2. Each artificial synapse between neighboring neurons is also categorized into two components: i) *excitatory synapse* and ii) *inhibitory synapse*. The excitation component is activated by reducing the equivalent resistance of the corresponding synapse via digital control inputs when neighboring neurons are coupled to each other via a

positive synaptic weight (i.e., two neighboring neurons, each representing an image pixel, that are more likely to be of the same color, either white or black). The inhibition component is activated in the same way as the excitation component when the relationship between neighboring neurons is represented by a *negative* synaptic weight (i.e., two neighboring neurons, each representing an image pixel, that are more likely to be of the opposite colors).

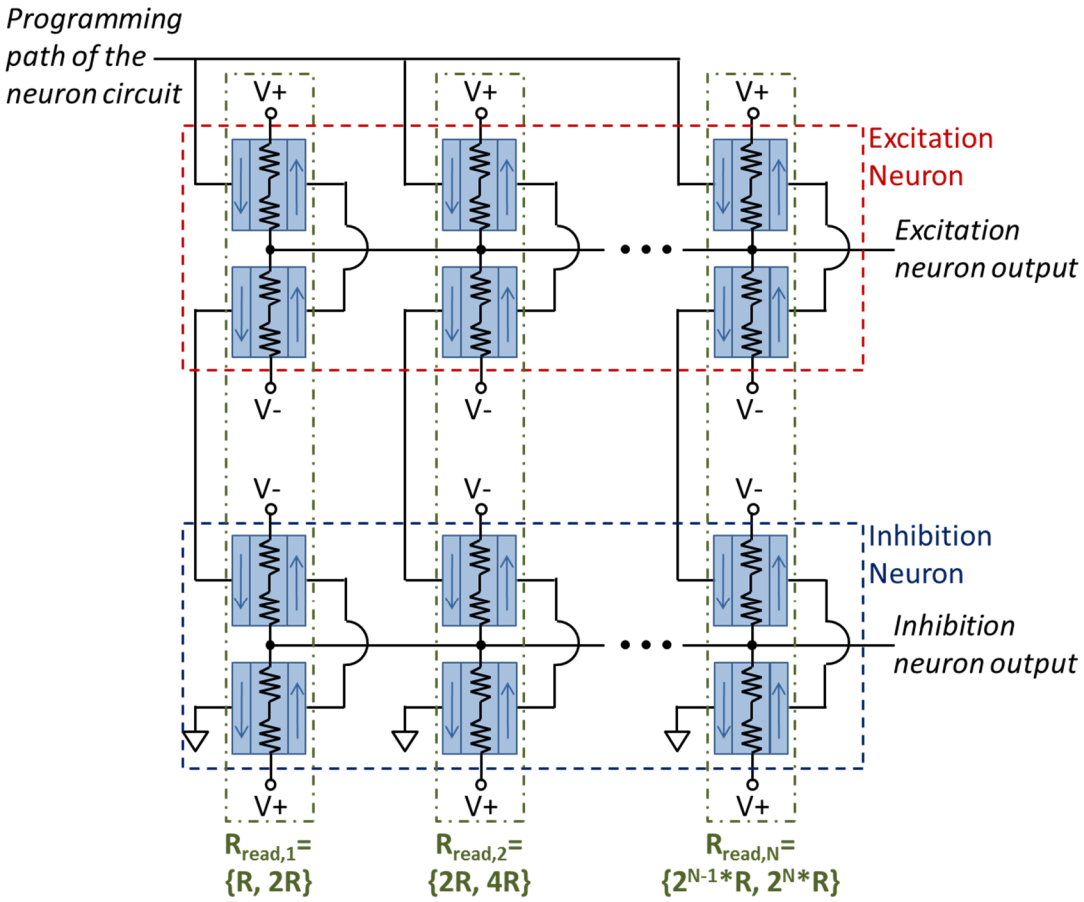

Figure 4.1. The proposed neuron circuit based on mCells. Different read-path resistances can be set by adjusting the cross-sectional area of tunnel barriers of the corresponding mCells. The write-path resistances of mCells are differentiated by changing their write-path lengths to enable different switching thresholds for magnetic buffers as follows: $R_{write,1}>R_{write,2}>\ldots>R_{write,N}$. R_{read} denotes the read-path resistances of the corresponding mCells. *V*+ and *V*- represent the positive and negative supply voltages with equal amplitudes, respectively.

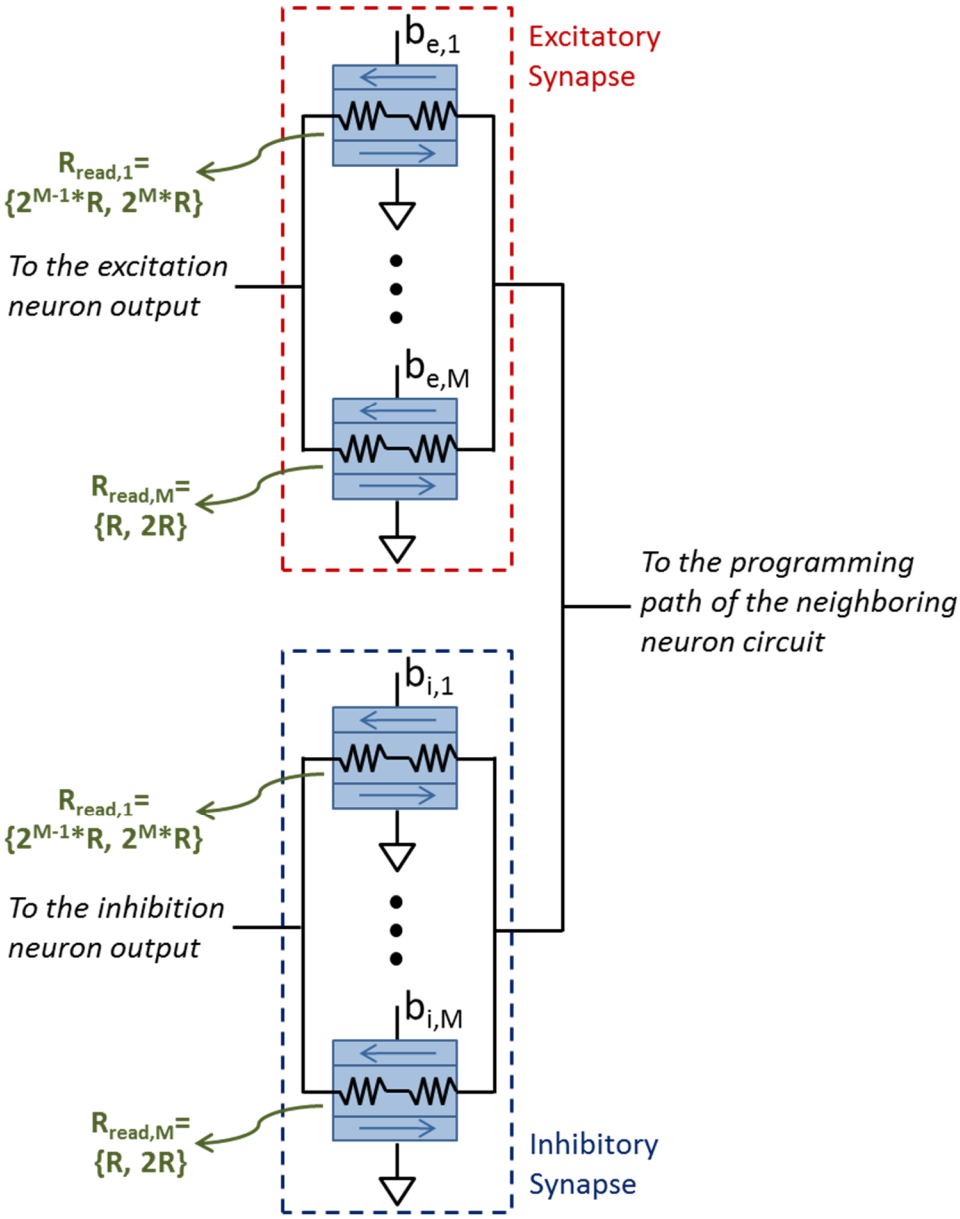

Figure 4.2. Excitatory and inhibitory synapse circuits based on mCells. Binary-weighted mCells can be obtained by changing the width of the device. $b_{e,1}$-$b_{e,M}$ and $b_{i,1}$-$b_{i,M}$ represent the M-bit excitation and inhibition control data, respectively. R_{read} denotes the read-path resistance of the corresponding mCell.

The *strength* of the relationships (i.e., synaptic weight) among artificial neurons can be adjusted by altering the equivalent resistance the corresponding synapse represents. A lower resistance corresponds to a stronger relationship (i.e., higher current), while a higher resistance

corresponds to a weaker relationship (i.e., lower current). The *subtraction* of these currents flowing through excitatory and inhibitory synapses then determines the final impact of the corresponding synapse on the neuron output, since excitatory and inhibitory synapses are connected to the oppositely-signed signals as explained earlier. As such, each weighted current generated via an artificial synapse takes one of the values from $I*\{-2^M+1, -2^M+2, \ldots, -1, 0, 1, \ldots, 2^M-2, 2^M-1\}$. For example, excitatory and inhibitory synapses consisting of four mCells in parallel can generate currents ranging from -15I to 15I, in steps of I.

4.3 Circuit Simulation Results

To verify the functionality of our proposed associative memory architecture we used a Verilog-A model of the mCell device that was written in SPICE for circuit simulations using a first-order approximation of the underlying physics [27]-[28]. Using this compact model we constructed a 5-neuron associative memory. We set the thickness of the write-paths for mCells to 6nm, which is a technology parameter that cannot differ from one device to another. We used three mCells for both excitatory and inhibitory synapses. This allowed 15 different synaptic weights to be defined in our model. To properly size the mCell devices in the synapse circuits, we adjusted the width of the devices as {10nm, 20nm, 40nm}, yielding the read-path resistances: $R_{read,1}$={2.5kΩ, 5kΩ}, $R_{read,2}$={1.25kΩ, 2.5kΩ}, and $R_{read,3}$={625Ω, 1.25kΩ} (see Figure 4.2). Moreover, we implemented the neuron circuits using four magnetic buffers in parallel for both excitation and inhibition neuron components. For proper configuration of the neuron circuits we set the length of tunnel barriers of the corresponding mCells along with the length of the write-paths to {80nm, 40nm, 20nm, 10nm} to get the read-path resistances: $R_{read,1}$={312.5Ω, 625Ω}, $R_{read,2}$={625Ω, 1.25kΩ}, $R_{read,3}$={1.25kΩ, 2.5kΩ}, and $R_{read,4}$={2.5kΩ, 5kΩ} (see Figure 4.1). Thus, the length of the write-paths of mCells in the neuron circuits became {190nm, 110nm, 70nm, 50nm}, resulting in the write-path resistances: $R_{write,1}$=326Ω, $R_{write,2}$=189Ω, $R_{write,3}$=120Ω, and $R_{write,4}$=85.7Ω (see Figure 4.1).

Using pattern examples [1 0 1 1 0] and [1 0 1 0 1] as memorized patterns by programming excitatory and inhibitory synapses accordingly, we evaluated convergence of this small network consisting of 5 neurons when other patterns were provided as inputs. Figure 4.3 shows our pattern recognition example using gray-scale pixels as well. The system correctly recognizes the pattern [1 0 1 0 1] as the closest memorized pattern to the initial gray-scale input pattern.

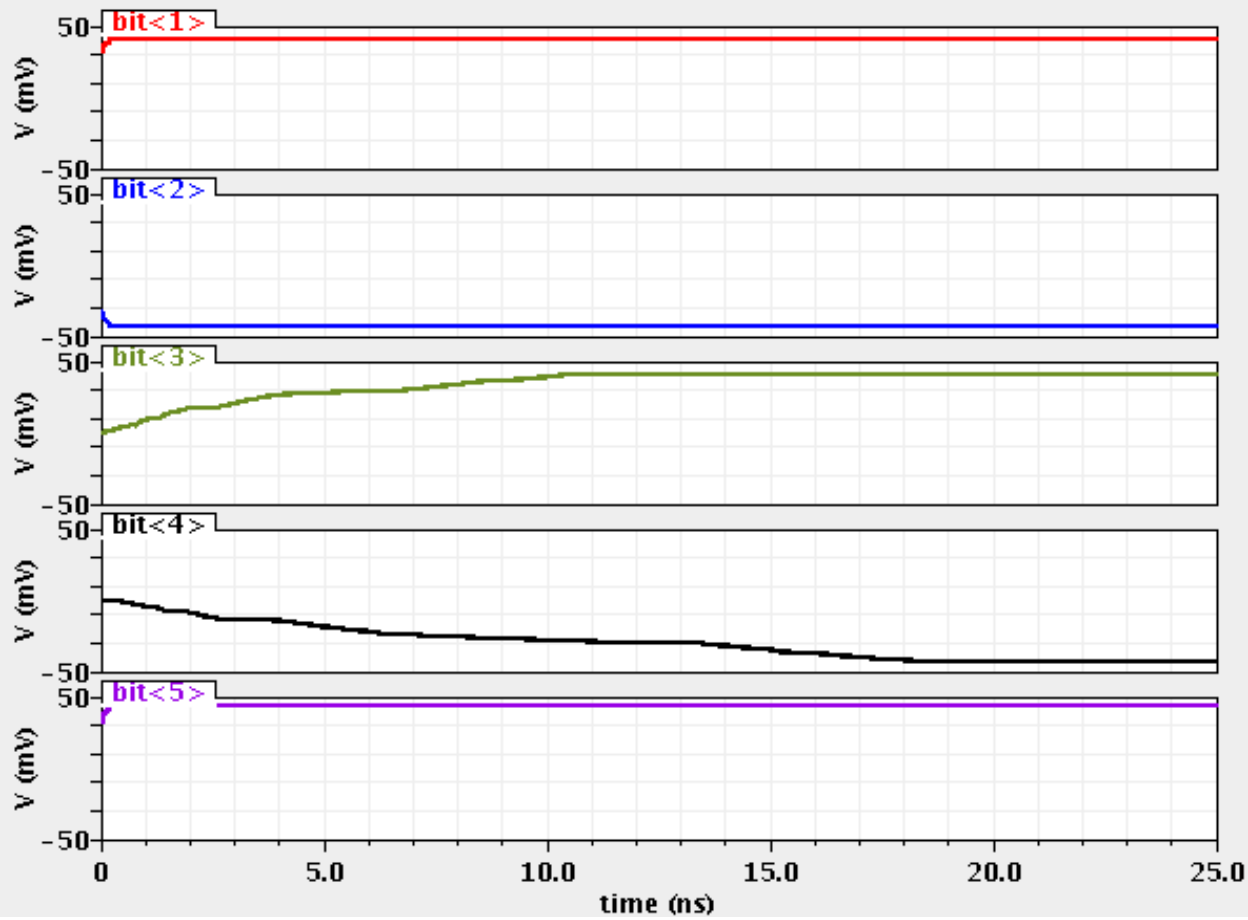

Figure 4.3. Pattern recognition example based on a 5-neuron system. Intermediate pixel values are possible as initial conditions by means of different switching thresholds of mCells in the neuron circuits. For this example the initial input pattern is [1 0 0.5 0.5 1] at time=0s and the recovered output pattern is [1 0 1 0 1] at time=25ns.

We also constructed larger associative memory circuits with local interconnections among artificial neurons (e.g., 9 connections for each neuron). For illustration we show here the same mCell simulation models to evaluate an associative memory consisting of 20 neurons. The bit patterns shown in Figure 4.4 are examples of memorized patterns that are programmed into the neural network circuit model via synaptic weights.

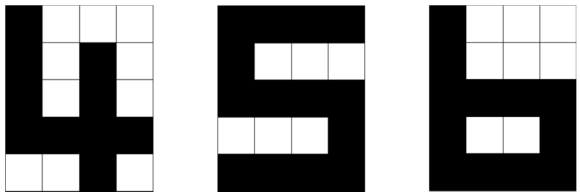

Figure 4.4. Memorized bit patterns for a 20-neuron based associative memory.

As an initial input pattern we applied a 27% distorted version of the bit pattern '4' with several gray-scale pixels as shown in Figure 4.5 (left), and our associative memory design successfully recovers the bit pattern '4' as shown in Figure 4.5 (right). Importantly, the results indicate that our proposed NN-based associative memory architecture scales well with the number of bits as long as the nearest neighbor coupling is sufficiently accurate.

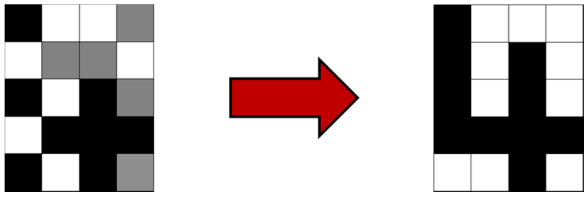

Figure 4.5. The initial input pattern for a 20-neuron based associative memory (left) and the output pattern produced by this associative memory (right).

Now we showcase the use of the same neural network system consisting of 20 neurons for different image processing applications such as edge and line detections. In order to program excitatory and inhibitory synapses, we used the cloning templates for edge and line detection

applications given in [8], [34]. Figure 4.6 presents our circuit simulation results for these applications. All the simulation results in this section validate the potential of our proposed neurocomputing architecture based on the integration of non-volatile magnetic devices to enable highly-efficient computing schemes.

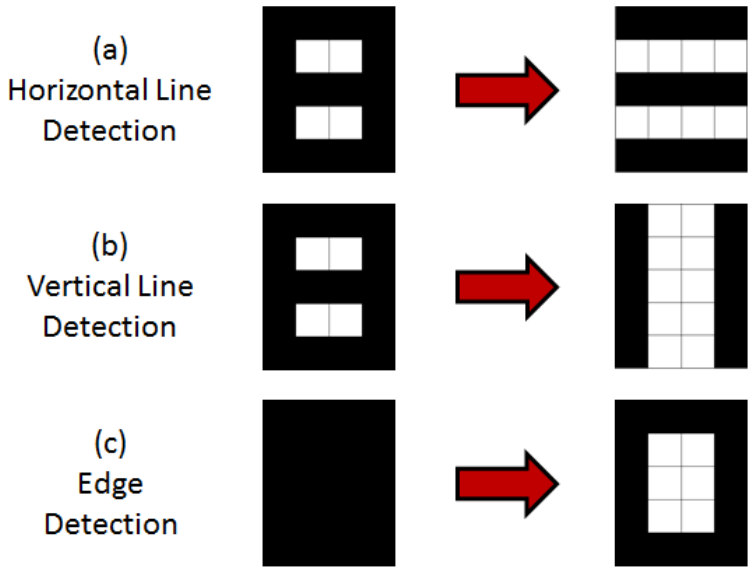

Figure 4.6. Circuit simulation results for neurocomputing circuits based on mCells. Input patterns (left) and output patterns (right). (a) Horizontal line detection. (b) Vertical line detection. (c) Edge detection.

4.3.1 Comparisons to Digital CMOS and mLogic Implementations

Using our neurocomputing circuit models developed in the previous section we here compare our proposed all-magnetic analog neural network with digital implementations in terms of power and device count. Though it is an immature technology, mCell-based analog implementation provides significant improvements over digital CMOS as shown in Figure 4.7. With device improvements described in the next section the use of such a programmable, non-volatile resistance element could enable a practical realization of neurocomputing systems and associative memories.

Figure 4.7 also highlights that analog computing could be much more efficient than digital one for certain applications that could shift the computing paradigm used in electronics and computer systems. A better system solution for future computing architectures might be based on the use of a mixed-signal scheme enabled by highly-efficient analog neurocomputing and a digital processor, inter-switched based on the application.

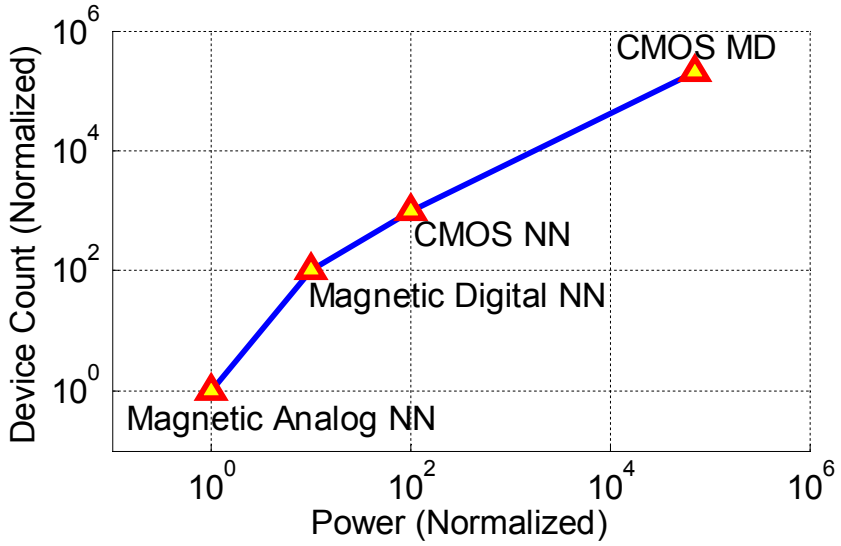

Figure 4.7. Comparison results for device count and power.

It is important to note that both simulated mCell and a 32nm CMOS transistor occupy roughly the same footprint area. Thus, device count comparison here represents area comparison for both implemented architectures. Since area comparison is also affected by how well each technology scales, a comparison based on device count better demonstrates corresponding circuit complexities for both technologies.

4.4 A Guideline for Future Device Development for mCells

This section provides initial targets for what can be improved for mCells to make proposed analog system viable. Firstly, threshold for switching current can be reduced to enable more efficient synapse programming and most importantly, ultra-low voltage and ultra-low power

operation of all-magnetic analog neurocomputing system. Increase in the switching speed (i.e., domain wall velocity per applied current density in the write-path) will decrease the settling time of overall system to a stable pattern (e.g., sub-ns), thereby allowing highly energy-efficient neurocomputing. In addition, increase in the high-to-low read-path resistance ratio of the mCell devices will significantly reduce power consumption of the neuron circuits.

Chapter 5

Thermal-Based Analog Associative Memory and Neurocomputing

In this chapter we propose a novel configuration of ovenized AlN resonators [41]-[43] as a tunable analog resistance for efficiently building artificial neurons and synapses [44]. This chapter begins with a brief introduction to ovenized AlN resonators, followed by our device and system proposals for neurocomputing. Next it elaborates on how to implement D/A programming feature required for synapse programming and neuron initialization based on our proposed device configuration while providing a *thermal DAC* design example. Finally it demonstrates our circuit simulation results for a thermal associative memory, following by a brief description of the Verilog-A compact model for the proposed device that is developed based on measurement data.

5.1 Ovenized AlN Resonator

As we reach the end of the CMOS roadmap it is apparent that not any one device will fully replace CMOS for general purpose integrated circuit applications. There are, however,

opportunities to explore the integration of emerging technologies with CMOS that will produce novel integrated systems in the near future. For example, MEMS devices [45]-[46] have already been demonstrated to enable high-Q filtering and analog/RF function that are otherwise impractical or impossible with CMOS alone. Such mechanical devices have also been explored for use as digital logic switches [47]-[49], but it is their *analog functionality* that could provide the greatest benefit for future computing systems.

AlN resonators have been recently demonstrated to provide a tunable RF resistance as a function of a controllable thermal input for ovenized oscillator applications [41]-[42]. The AlN resonator (see Figure 5.1) is formed by a piezoelectric AlN film sandwiched between two metal electrodes. The bottom electrode can be left floating or grounded. The top electrode is lithographically patterned into interdigitated electrodes in order to define the resonator center frequency and excites in-plane lateral vibrations.

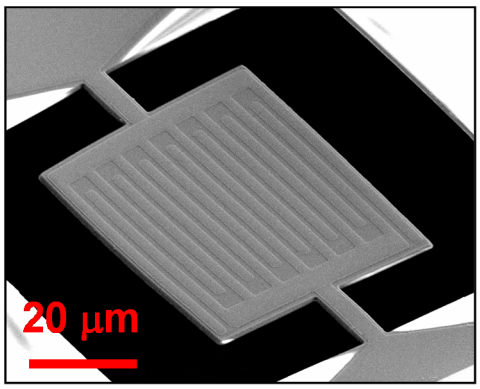

Figure 5.1. Micrograph of an example AlN RF resonator *(Courtesy of Gianluca Piazza's group from Carnegie Mellon University)*. The resonator size is relatively large (approximately 50µm x 100µm x 1µm) as it was designed for high power oscillators. The pitch of the interdigitated metal electrodes sets the resonance frequency.

When the piezoelectric material is excited by an external AC signal it converts applied RF voltage into a mechanical strain. As the frequency of the AC signal matches the resonance

frequency of the mechanical structure, the amplitude of motion increases. The generated displacement is sensed as an increase of charge, which effectively corresponds to a maximum value for electrical admittance given a fixed input voltage. By sweeping the AC signal over frequency the characteristic admittance of a mechanical resonator is attained (see Figure 5.2). The resonator center frequency is set by the physical dimensions and the acoustic velocity of the material, which is highly dependent on temperature (approximately -28ppm/K). Therefore, it is possible to rigidly shift the resonant admittance curve by heating the resonator.

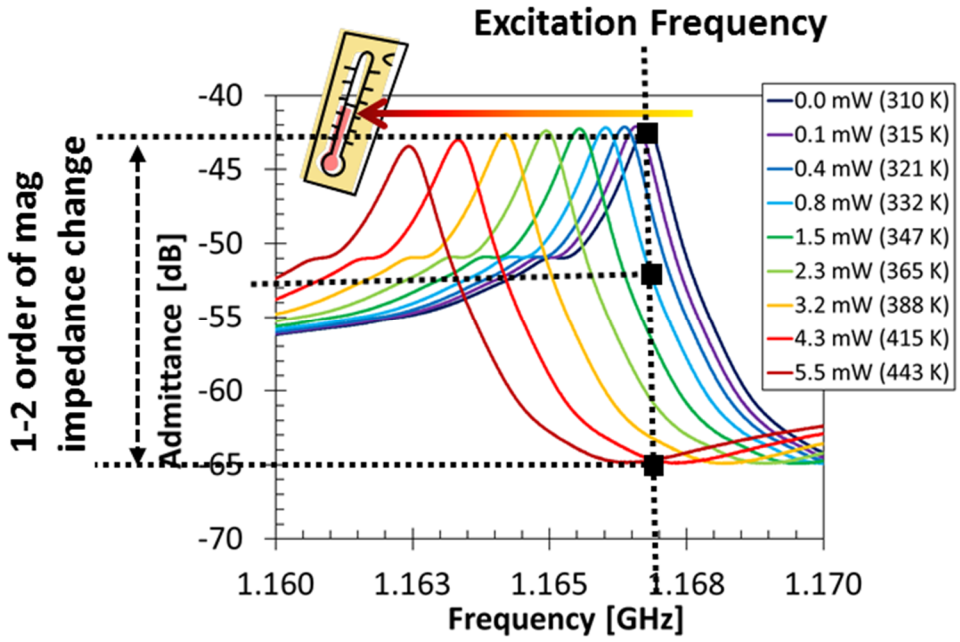

Figure 5.2. RF admittance of the resonator with different heater powers for an example resonator [41]. The thermal input applied via heater resistance shifts the admittance curve in frequency axis. One to two orders of magnitude change in RF impedance at a specific operation frequency enables a tunable analog resistance for artificial synapses and a building block for artificial neurons.

A method for in-situ and low power heating (ovenization) of the AlN mechanical resonator has been recently developed [41]-[43]. As the resonator is locally heated, its admittance versus frequency curve is rigidly translated towards lower frequencies as can be seen in Figure 5.2. The heater is formed by a simple resistor patterned in either the bottom or top electrode of the

resonator. By integrating the heater within the body of the resonator (see Figure 5.3) and taking advantage of small thermal mass of the AlN resonator, it is possible to attain one to two orders of magnitude change in impedance with just few mWs of power for these *large-scale resonators* that are used for RF circuit applications. For example, as shown in Figure 5.2, when operating at a fixed frequency it is possible to vary the equivalent impedance of the resonator by more than one order of magnitude with 5.5mW of heater power. As the dimensions of the resonator are scaled we calculate that merely few tens of μW power levels will be required to achieve similar response.

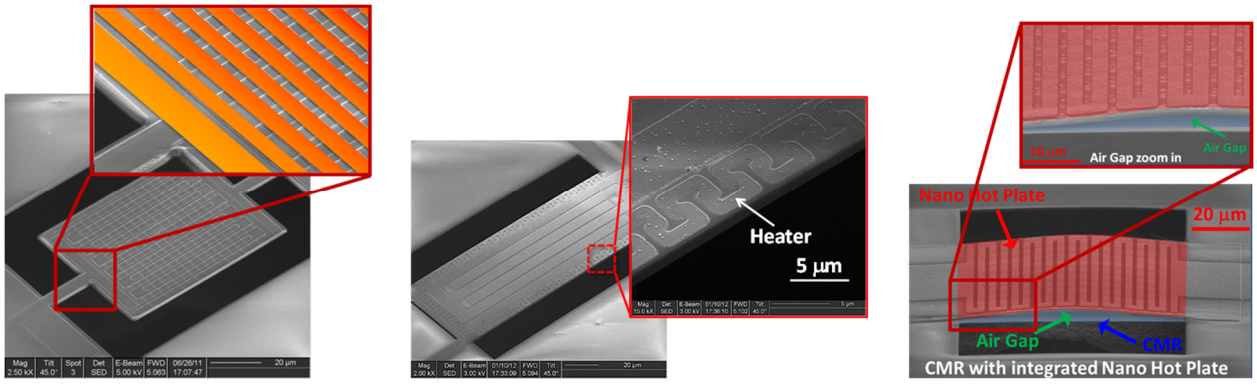

Figure 5.3. Example heater implementations as a serpentine on the bottom electrode [41] (left), around the top electrode [42] (center), and on top of the resonator [43] (right).

5.2 Proposed Architecture for Neurocomputing Based on Ovenized Resonators

The ovenized AlN resonator offers great opportunities for analog associative memories because its impedance can be set over a continuous range by a local heater when operated at a specific frequency (see Figure 5.2). However, it would be inefficient to require an analog control voltage for adjusting the impedance of each resonator, since this would correspond to requiring a DAC to convert digital control bits to an analog signal for each neuron and synapse.

Instead, we propose to use *multiple heaters* for each resonator to eliminate the need for a DAC and implement all-analog nanoscale associative memory based on a neuron-synapse model demonstrated in Figure 2.7. This approach is enabled by the additive feature of thermal power via multiple heaters that provides a natural summing and compact D/A conversion. Such a compact device can be implemented as one or a combination of the following ways to integrate multiple heaters into the device configuration: i) transforming the bottom floating electrode into a serpentine, ii) including a serpentine around the top RF electrode, and iii) adding a serpentine heater on top of the resonator (see Figure 5.3). Figure 5.4 demonstrates our proposed circuit schematic symbol for an ovenized resonator with multiple heaters.

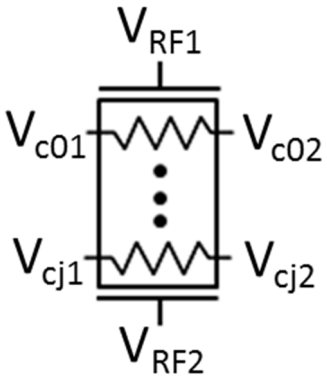

Figure 5.4. The proposed circuit schematic symbol for an ovenized resonator with *j+1* heaters. The thermal control inputs are applied between V_{co1}-V_{cj1} and V_{co2}-V_{cj2}. V_{RF1} and V_{RF2} represent the RF ports of the resonator.

The conceptual circuit model for our proposed architecture is shown in Figure 5.5. Each artificial neuron is constructed as one resonator pair that functions similar to a simple *inverter*, but produces intermediate values between digital logic levels '0' and '1' as well. This is enabled by tunable analog resistance of ovenized resonators as they are controlled by thermal inputs. This feature of ovenized AlN resonators provides efficient neuro-like computations using just one resonator pair for each artificial neuron.

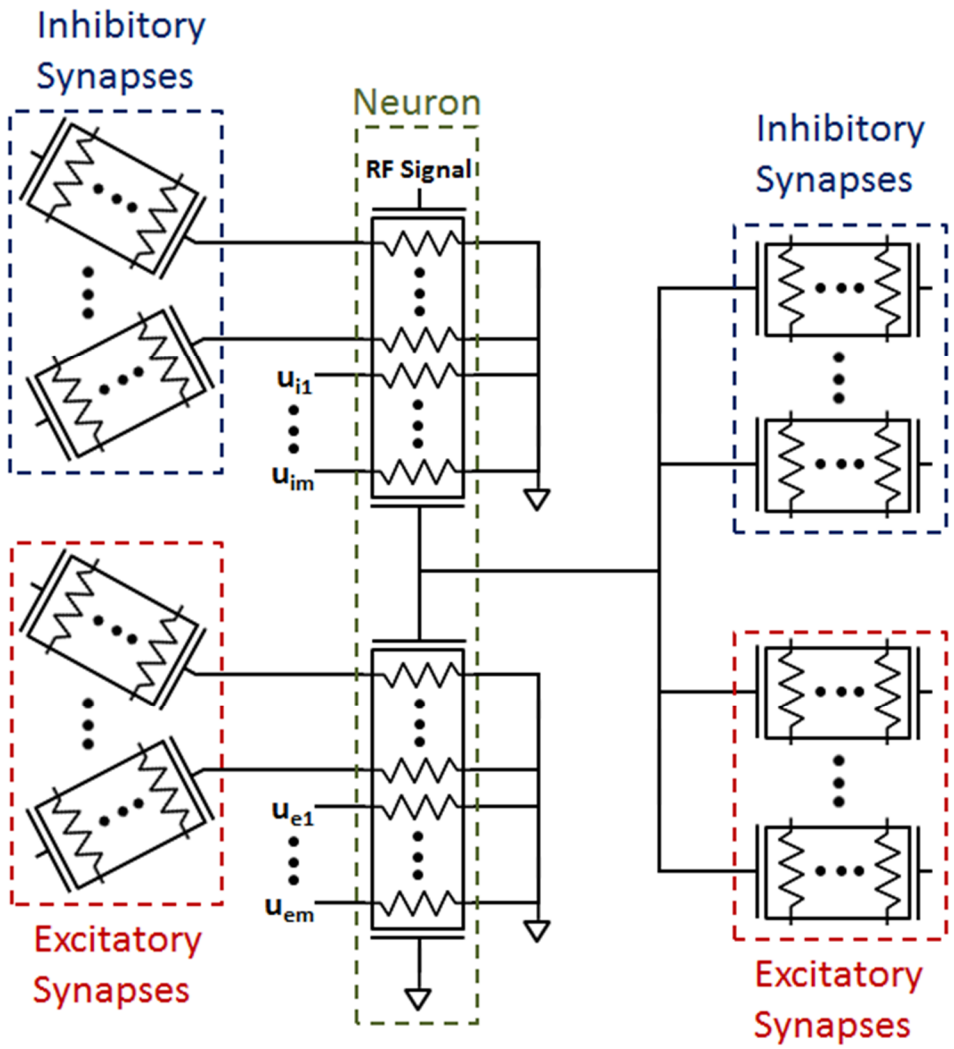

Figure 5.5. Conceptual circuit diagram of the proposed AlN resonator based associative memory. The heaters for synapse circuits and ones connected to the initial inputs in the neuron circuits need to be carefully designed to provide a proper D/A conversion from digital inputs to the resonator impedance while ones connected to neighboring synapses in the neuron circuits are equivalent to each other (e.g., 1kΩ).

Each synapse between neighboring neurons is categorized into two components: i) *excitatory synapse* and ii) *inhibitory synapse*, as shown in Figure 5.5. Each of these components is represented by one resonator. Excitation component is activated by reducing the impedance of the corresponding resonator via thermal control inputs when neighboring neurons

are coupled to each other via a *positive* synaptic weight (i.e., two neighboring neurons, each representing an image pixel, that are more likely to be of the same color, either white or black). The inhibition component is activated in the same way as the excitation component when the relationship between neighboring neurons is represented by a *negative* synaptic weight (i.e., two neighboring neurons, each representing an image pixel, that are more likely to be of the opposite colors). The *strength* of these relationships among artificial neurons can be adjusted by altering the impedance of the corresponding resonator. A lower impedance corresponds to a stronger relationship (i.e., higher current), while a higher impedance corresponds to a weaker relationship (i.e., lower current).

Similarly, the initial thermal inputs representing the initial input patterns consist of two components: i) *excitation control bits* connected to the pull-down resonator in the neuron circuit and ii) *inhibition control bits* connected to the pull-up resonator in the neuron circuit, as depicted in Figure 5.5. *When the initial thermal input corresponding to the initial input pixel is positive* (i.e., closer to a black image pixel), it is applied to excitation control bits (u_{e1}-u_{em}), thereby pulling the initial neuron output signal (i.e., resonator-based inverter output) up closer to RF supply signal. On the other hand, *when the initial thermal input is negative* (i.e., closer to a white image pixel), it is applied to inhibition control bits (u_{i1}-u_{im}), thus pulling the initial neuron output signal down closer to ground (i.e., 0V).

It is important to note that inhibitory synapses are connected to the pull-up resonator of the neuron circuit (pair of resonators representing an artificial neuron), while excitatory synapses are connected to the pull-down resonator of the neuron circuit. This is because the weighted currents through artificial synapses increase the heater power, which in turn increases the impedance of the corresponding resonator. Therefore, when excitation becomes more dominant as compared to inhibition, the neuron circuit generates an output level that is closer to the RF

signal level that is applied to the corresponding neuron. In contrast, when inhibition is more dominant, the neuron circuit produces an output that is closer to 0V.

For our proposed AlN resonator based neural network architecture the weighted outputs of neighboring neurons are summed in the neuron circuits by means of multiple heaters that are equivalent to each other. The additive feature of power generated by each heater offers a natural summing mechanism for the impact of artificial synapses. This could not be done by a single heater connected to all the artificial synapses since adding currents, instead of heat, would be problematic due to phase differences among the neighboring neuron output signals. These differences, when summed, would result in partial (or even complete) cancellation of signals and, therefore, incorrect outcomes. Using multiple heaters for each resonator eliminates such phase shift effects since total heat is now summed, and it depends only on the squares of the magnitude of the weighted RF signals. For proper operation, therefore, the synapse impedances have to be adjusted based on the squares of the current magnitudes to generate the required heat, rather than the current magnitudes themselves.

For any neurocomputing architecture, artificial neurons require a means of generating an output (i.e., an activation function) based on the sum of the incoming neural signals. The activation function in neural networks is generally a variant of the sigmoid function. The neuron transfer function based on measurement data from two exemplary AlN resonators (approximately 50μm x 100μm x 1μm) that are configured as a non-inverting "inverter" produces the simulated characteristic shown in Figure 5.6. This neuron characteristic provides the sigmoid-like functionality required for use by the architecture in Figure 5.5 to implement a NN-based associative memory.

It is important to note that both storage capacity and the size of basin of attraction for the proposed associative memory circuit depend on the selected training algorithm for calculating synaptic weights, but developing such algorithms for NNs is beyond the scope of this thesis.

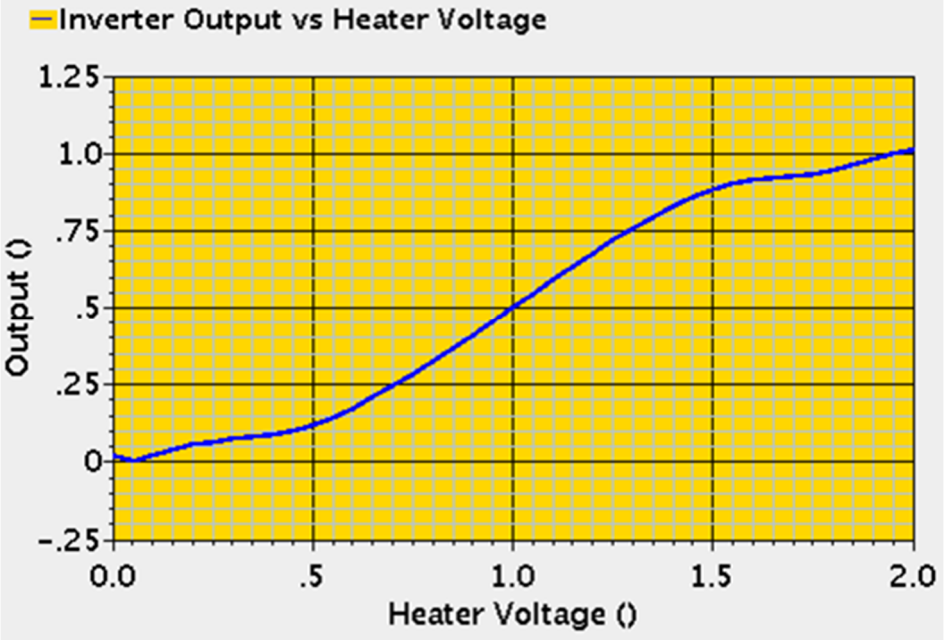

Figure 5.6. Transfer function of our "inverter-based" neuron using ovenized AlN resonators when the amplitude of RF supply voltage is 1V.

5.3 Implementing D/A Programming Feature

The careful tuning of artificial synapses and properly applying the initial input patterns are of great importance for neurocomputers to function properly. Despite being compact, the proposed AlN resonator-based architecture still necessitates the scrutinized design of DAC due to the non-linear relationship between the resonator impedance (Z_{res}) and heater power (P_{heater}) (see Figure 5.2). Figure 5.7 demonstrates our proposed artificial synapse device that is controlled by *j+1* digital control bits. The digital-to-analog conversion based on this device configuration is performed in accordance with the following equation:

$$\frac{\overline{b_0}^2}{R_0}+\frac{\overline{b_1}^2}{R_1}+\cdots+\frac{\overline{b_j}^2}{R_j}=P_{heater}\left(Z_{res}\right) \tag{5.1}$$

where the left-hand side of the equation represents total power generated via variable-size heaters, and R_0-R_j denote heater resistances that are connected to the corresponding control bits

($\overline{b_0}$ - $\overline{b_j}$). The complementary versions of the control bits are applied to the heaters because the increase in the heater power increases the resonator impedance, thereby decreasing the current flowing through the corresponding synapse (i.e., reducing the strength of artificial synapse).

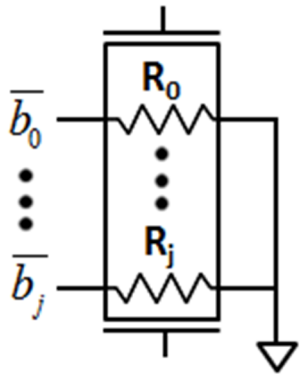

Figure 5.7. Proposed DAC device using multiple heaters for each resonator. b_o-b_j represent the corresponding bits for the digital input.

The heater power required for a specific output impedance (synaptic weight) can be found from the resonator impedance versus heater power curve. Using all possible combinations of the control bits (2^{j+1} combinations for a ($j+1$)-bit DAC) results in an over-determined system consisting of 2^{j+1} equations with $j+1$ unknowns (R_o-R_j). Since the resonator impedance changes non-linearly with the heater power, a regression (e.g., least-squares fitting) is required to determine variable-size heater resistances for such an equation system.

The variable-size heaters required for D/A conversion to apply the initial input patterns in the neuron circuits can be designed in a similar way to that required for artificial synapses. However, for this case the curves in Figure 5.8 should be used for fitting. These curves correspond to the transfer function shown in Figure 5.6, but are split into two parts: i) excitation and ii) inhibition.

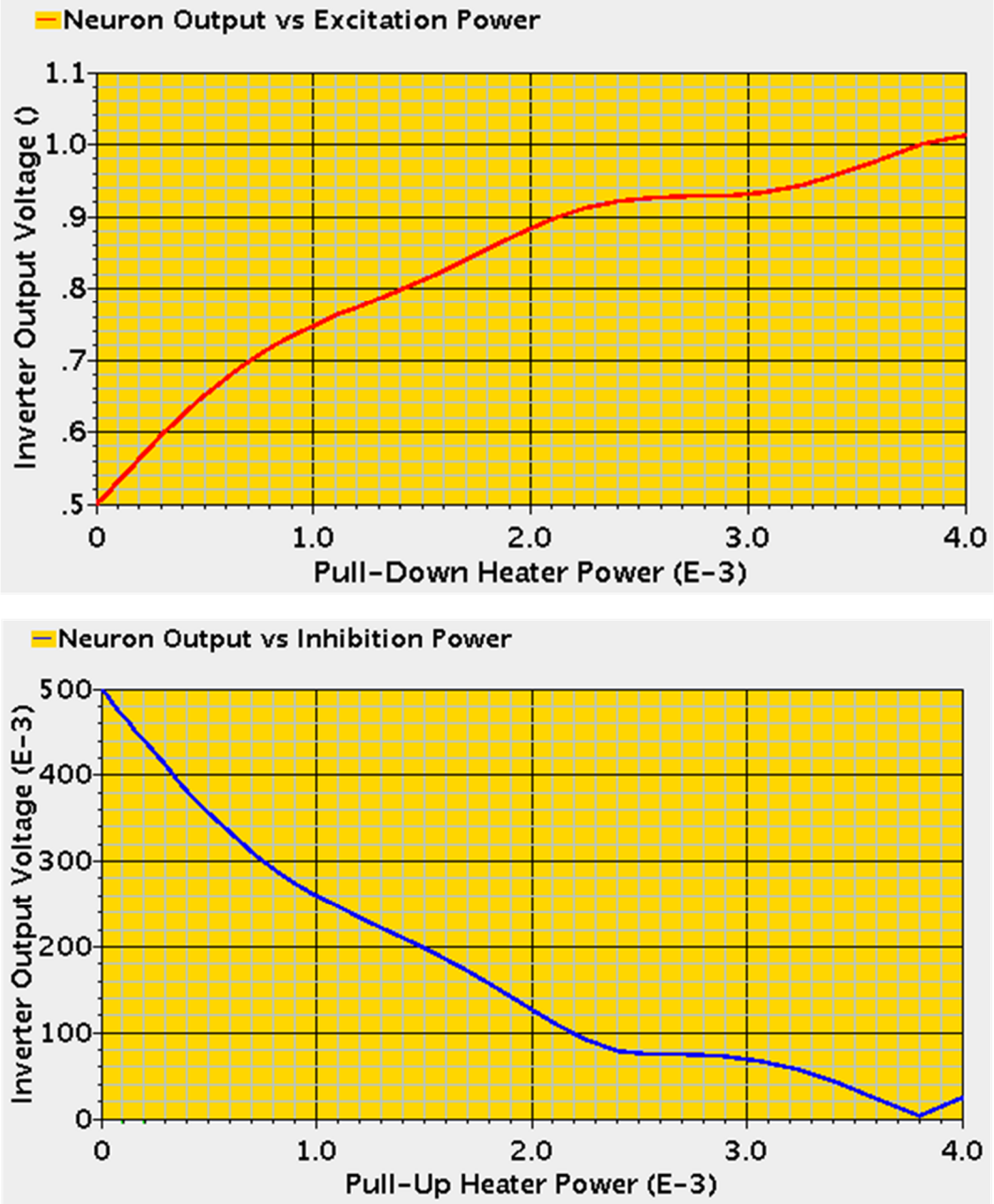

Figure 5.8. Neuron output signal level versus heater power applied via excitation control bits (top) and neuron output signal level versus heater power applied via inhibition control bits (bottom) when the amplitude of RF supply voltage is 1V. These two curves correspond to the transfer function shown in Figure 5.6.

5.3.1 Derivation of Variable-Size Heater Resistances

The equations used to determine variable-size heater resistances must consider the temperature-dependence of these resistances since heater resistances in AlN resonators increase linearly with increase in the resonator temperature [41]. This effect can be formulated as:

$$R_{heater}(\Delta T) = R_{nom}(1+\alpha_{th}\Delta T) \tag{5.2}$$

where R_{heater} is heater resistance changing with the resonator temperature, ΔT represents temperature increase in the resonator, R_{nom} is nominal heater resistance at ambient temperature, and α_{th} is the temperature coefficient of heater resistance. The relationship between temperature increase in the resonator and the heater power is linear [41], yielding:

$$R_{heater}(\Delta T) = R_{nom}(1+\alpha_{th}R_{th}P_{heater}) \tag{5.3}$$

where R_{th} is the slope of temperature increase in the resonator versus the heater power curve. Both α_{th} and R_{th} can be extracted from measurement data.

To derive the equation system for this design problem, we modify (5.1) by considering thermal effects on heater resistances, as follows:

$$V_{on}^2 \sum_{k=0}^{j} \frac{1}{R_k(\Delta T)}\left(\left\lfloor \frac{i}{k+1} \right\rfloor \bmod 2\right) = P_{heater}(Z_{res}) \tag{5.4}$$

where V_{on} is the on-voltage for the digital control bits, i is the number that the digital control bits represent, and $\lfloor \cdot \rfloor$ denotes the largest integer that is less than the number inside (*floor* function). Plugging (5.3) into (5.4) yields:

$$V_{on}^2 \sum_{k=0}^{j} \frac{1}{R_{nom,k}\left(1+\alpha_{th}R_{th}P_{heater}(Z_{res})\right)}\left(\left\lfloor \frac{i}{k+1} \right\rfloor \bmod 2\right) = P_{heater}(Z_{res}) \tag{5.5}$$

where $R_{nom,k}$ is nominal heater resistance for $R_k(\Delta T)$.

By taking all combinations of the thermal input bits into account we generate the system of equations that we seek:

$$V_{on}^2\left(I+\alpha_{th}R_{th}P_{diag}\right)^{-1}K\vec{G}_{nom}=\vec{P}_{heater}\left(\vec{Z}_{res}\right) \tag{5.6}$$

where the vector $\vec{P}_{heater}$ includes the target heater powers corresponding to the desired resonator impedances (determined using Z_{res} versus P_{heater} curve), I is the identity matrix, P_{diag} is a diagonal matrix with diagonal elements $\vec{P}_{heater}$, $\vec{G}_{nom}$ is the nominal heater conductances vector ($[1/R_{nom,0}\ 1/R_{nom,1}\ \ldots\ 1/R_{nom,j}]^T$), and K is a 2^{j+1}-by-$(j+1)$ matrix with elements $K(a, b)=\left\lfloor a/b\right\rfloor \bmod 2$.

To ensure monotonicity it can be easily proven that the following constraints must hold:

$$\begin{aligned} 1/R_{nom,0} &\geq \xi \\ 1/R_{nom,1}-1/R_{nom,0} &\geq \xi \\ &\vdots \\ 1/R_{nom,j}-\sum_{k=0}^{j-1}1/R_{nom,k} &\geq \xi \end{aligned} \tag{5.7}$$

where ξ sets the minimum step size between two consecutive heater conductances and is a parameter for our design problem. A regression method can be applied to this over-determined equation system with monotonicity constraints to determine heater resistances.

5.3.2 A "Thermal DAC" Example

To assess the potential of the proposed ovenized resonator configuration we created an optimization function in MATLAB that uses a "monotonic" least-squares fitting based on the equations described in the previous sub-section. As an example we designed a 4-bit DAC (i.e.,

j=3 in above equations) such that 16-level resistance can be obtained for an artificial synapse to set its synaptic weight. Figure 5.9 depicts the required curves based on measurement data for an approximately 50μm x 100μm x 1μm AlN resonator [41] to determine α_{th}, R_{th} and the target heater powers for the desired impedance values. The ambient temperature for this measurement data is 310K [41].

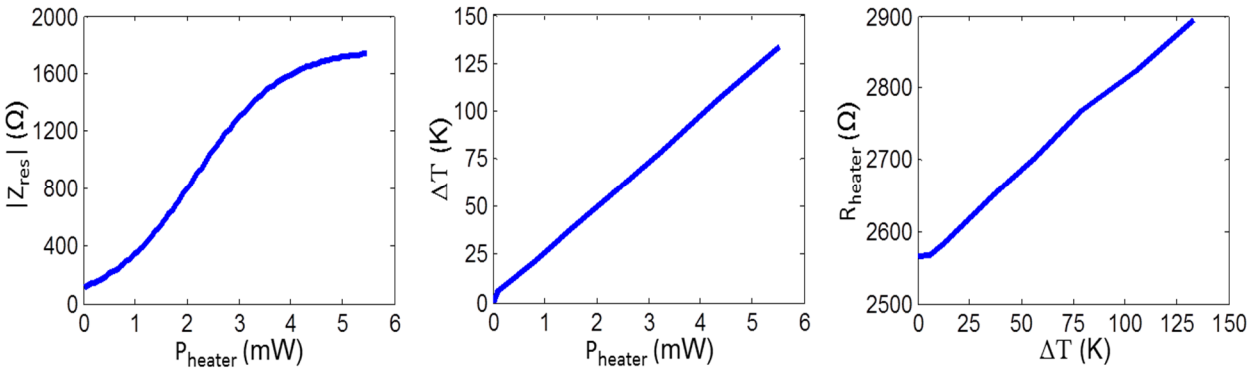

Figure 5.9. Resonator impedance versus heater power curve at 1.1667GHz operation frequency (left), temperature increase in the resonator versus heater power curve (center), and heater resistance versus temperature increase in the resonator curve (right). All curves are extracted from measured data for an example resonator [41].

Figure 5.10 shows three different cases where ξ is the lowest (i.e., 0) in Scheme 1 and the highest in Scheme 3. Scheme 1 provides the minimum least-squares fitting error (i.e., minimum integral non-linearity) whereas the step sizes are more uniform in Scheme 3 (i.e., minimum differential non-linearity). It is interesting to note that as ξ increases the proposed DAC architecture approaches the conventional binary-weighted (2^4) DAC design for this example.

As we change V_{on}, we only need to change the absolute values of nominal resistances while keeping their ratios constant. This is an important feature, especially in the presence of process variations, since the proposed architecture now depends on the ratios of heater resistances

rather than their absolute values. These results indicate that it is possible to represent each artificial synapse as a single resonator with multiple heaters in a compact and efficient way.

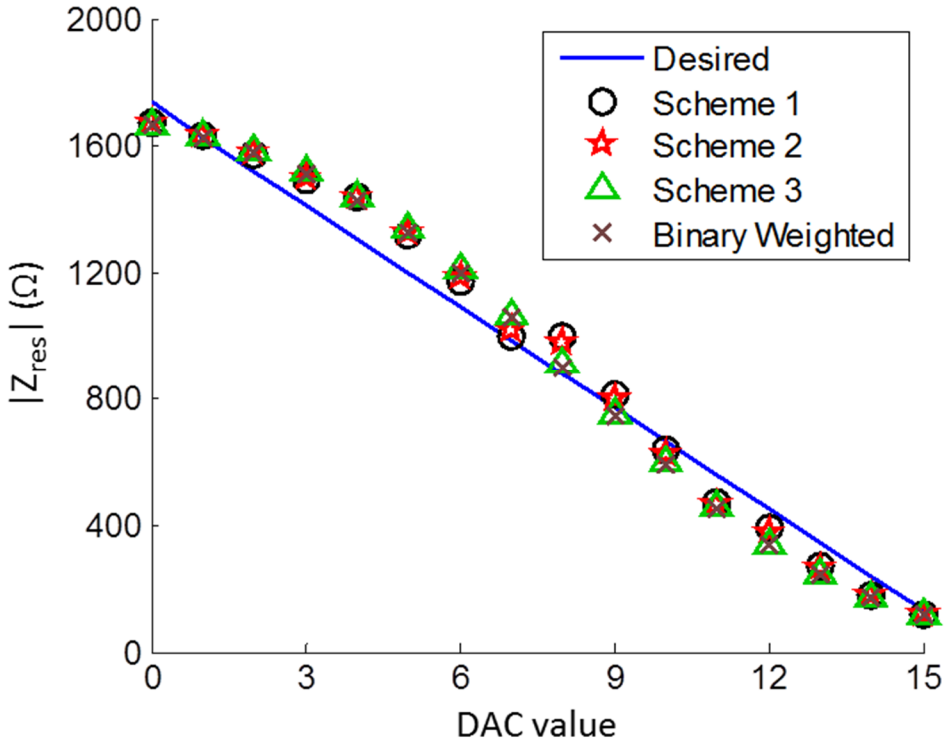

Figure 5.10. Designed DACs based on measurement data. The resonator impedance decreases as DAC value (i.e., synaptic weight) increases.

5.4 Circuit Simulation Results

To evaluate the potential of the proposed associative memory architecture in Figure 5.5 we have created a compact circuit simulation model in Verilog-A based on measurement data for approximately 50μm x 100μm x 1μm AlN resonator. While this resonator is much larger than what we propose to use in our associative memory implementation, it does provide some hardware data from which an accurate circuit-level simulation model can be built and characterized. Our model is comprised of an RLC circuit (see Figure 5.11) that is used for

modeling ovenized MEMS resonators [50]-[51]. The circuit parameters are extracted from measurement data using a least squares fitting to a Butterworth-Van Dyke model [52].

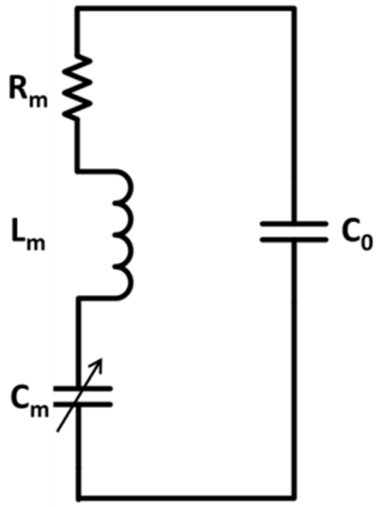

Figure 5.11. Circuit model for ovenized MEMS resonators. The variable capacitance (C_m) is thermally controlled by total applied power to the heaters. The circuit parameters are extracted from measurement data using a fitting algorithm specifically developed for such resonators.

Using this compact model we constructed a 5-neuron associative memory based on the architecture proposed in Figure 5.5. We used a resonator with four heaters to serve as a 4-bit DAC for synaptic weights. This allowed 31 gray-scale color levels to be defined in our model. Using pattern examples [1 0 1 1 0] and [1 0 1 0 1] as memorized patterns by programming excitatory and inhibitory synapses accordingly, we evaluated the convergence of this small network when other patterns were provided as inputs. Figure 5.12 shows a pattern recognition example. At the end of pattern recognition process, an RF signal amplitude of around 2V represents logic '1' while an RF signal amplitude of less than 1V represents logic '0'. Even though there is 40% distortion in the initial input pattern [0 1 1 1 0], the associative memory converges to the stored pattern that most closely resembles to this input pattern.

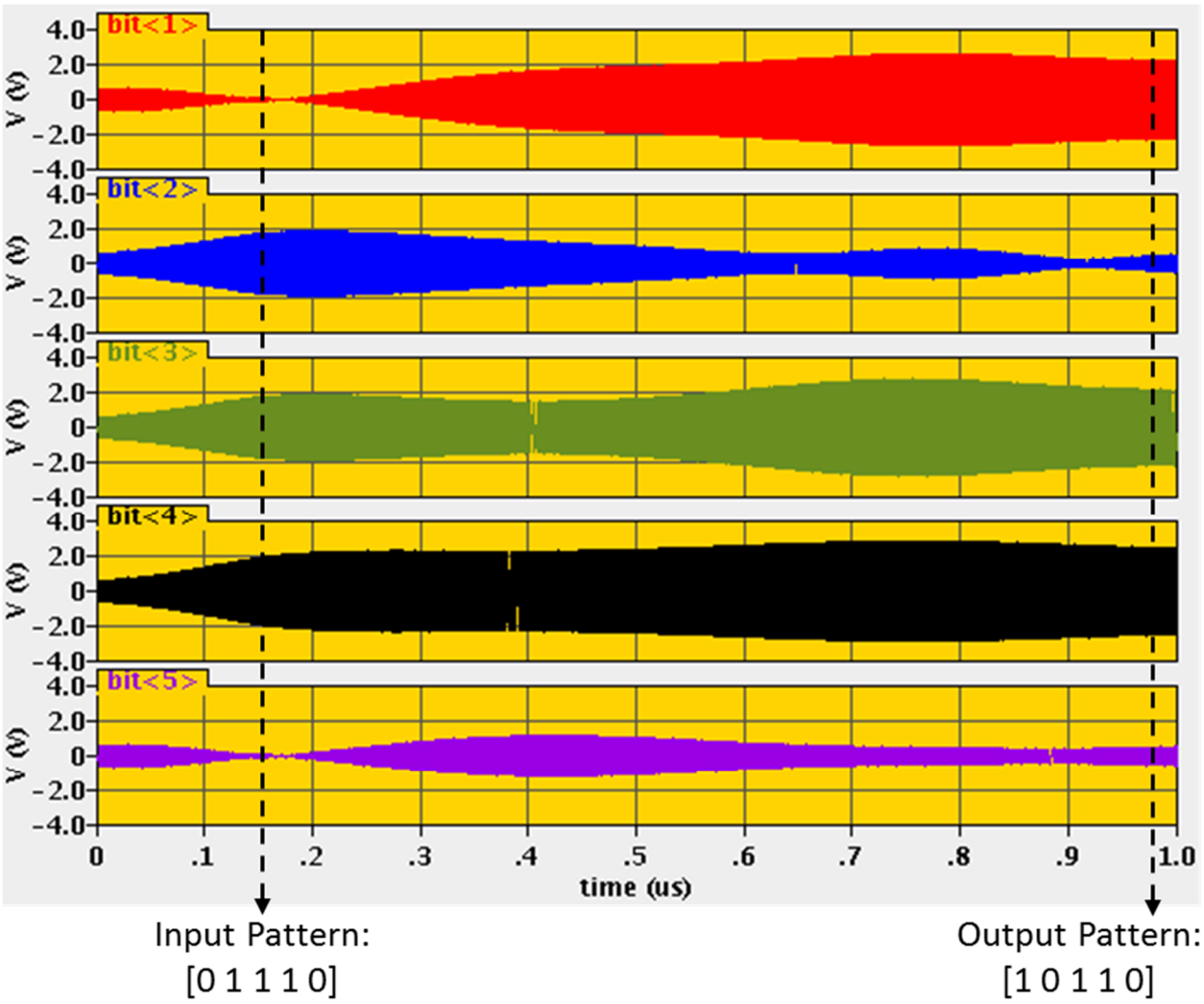

Figure 5.12. Pattern recognition example based on a 5-neuron system. Associative memory fully recognizes the pattern [1 0 1 1 0] despite 40% distortion in the initial input pattern.

Figure 5.13 demonstrates a second pattern recognition example for the same 5-neuron system, but now using gray-scale pixels represented by intermediate input signal values. The system correctly recognizes the pattern [1 0 1 0 1] as the closest memorized pattern to the initial gray-scale input pattern.

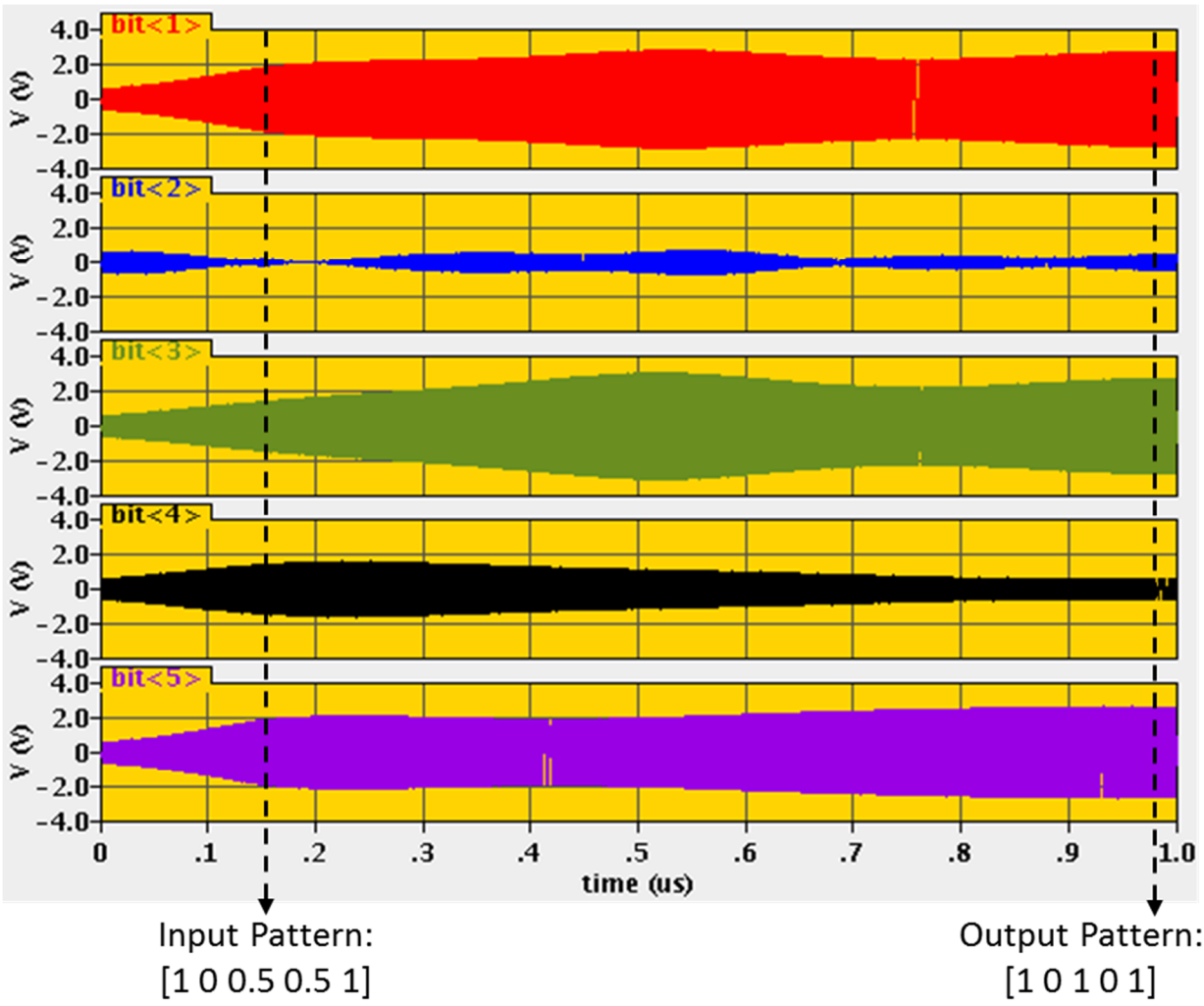

Figure 5.13. Pattern recognition example based on a 5-neuron system. Intermediate pixel values are possible as initial conditions by means of inherent tunability of analog resistance that ovenized resonators offer.

We also constructed larger associative memory circuits with local interconnections among artificial neurons (e.g., 9 connections for each neuron) based on the architecture in Figure 5.5. For illustration we show here the same AlN resonator simulation models to evaluate an associative memory consisting of 20 neurons. The bit patterns shown in Figure 4.4 are examples of memorized patterns that are programmed into neural network circuit model via synaptic weights (heater bits).

As an initial input pattern we applied a 28% distorted version of the bit pattern '5' with several gray-scale pixels, and our thermal associative memory successfully recovers the bit pattern '5' as illustrated in Figure 5.14. Importantly, the results here indicate that our proposed NN-based associative memory architecture scales well with the number of bits as long as the nearest neighbor coupling is sufficiently accurate.

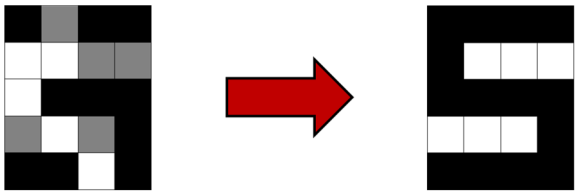

Figure 5.14. The initial input pattern for a 20-neuron based associative memory (left) and the output pattern produced by this associative memory (right).

Now we leverage the same thermal neurocomputing circuit consisting of 20 neurons for some other image processing applications such as edge detection, horizontal and vertical line detections. In order to program excitatory and inhibitory synapses in this system, we used the cloning templates for edge and line detection applications given in [8], [34]. We get exactly the same results as in Figure 4.6, proving correct functionality of the proposed thermal associative memory.

5.4.1 Energy and Area Comparisons

The Verilog-A compact model used for circuit simulations in the previous section was based on a large-scale resonator. Because such resonators are limited by the device area, thermal time constant [41], and total heater power required for maximum impedance swing, it is much more reasonable to analyze the performance of our proposed thermal neural network system in terms of energy and area. Figure 5.15 illustrates our comparisons to CMOS and biological neurons [1] based on these performance metrics. It is apparent that despite holding a huge promise for

future neurocomputing circuits, the proposed device configuration still needs significant improvements to enable practical thermal associative memories.

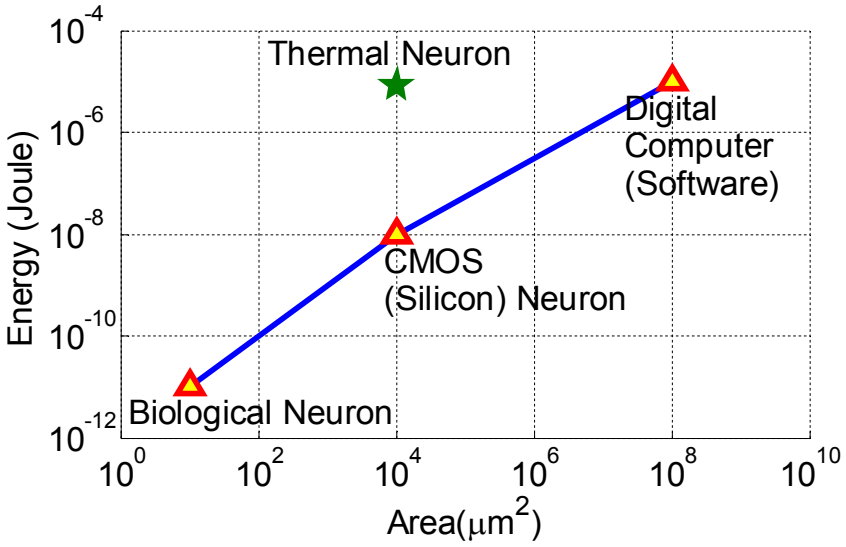

Figure 5.15. Energy and area comparison results for thermal neuron circuits.

5.5 Towards Viable Thermal Neurocomputing Circuits

This section is intended to provide direction and targets for future device development and research for AlN resonators that could enable highly-efficient neurocomputers. The most critical improvement required for ovenized AlN resonators is the device scaling. The primary challenge for this is the synthesis of nanoscale AlN films that preserve similar properties of their large-scale counterparts (e.g., piezoelectric coupling and intrinsic damping). Similar techniques that have been used for the deposition of highly oriented 10nm piezoelectric to make actuators and NEMS (nanoelectromechanical systems) relay [53]-[54] can be applied to enable the scaling of AlN resonators as well. As such, we anticipate that nanoscale devices could be formed by a piezoelectric plate of approximately 800nm x 500nm x 2nm.

To this effect, we have conducted a scaling analysis for such a miniaturized ovenized resonator. The main scaling equations are summarized in Table 5.1 where w, l and t denote the

width, length and thickness of the resonator. It is important to note that although the device dimensions are reduced, it is possible to keep a constant motional impedance and frequency for the resonator. This will enable the use of a lower power for generating the driving RF signal at few GHz that can easily be interfaced with the resonator (series resistance of about 100Ω despite scaling). More importantly, scaling is also advantageous in ensuring that a faster thermal time constant and lower voltages (i.e., lower energy) are required to operate each device. With these improvements we project operation with voltages lower than 200mV and operating speeds that are limited by a thermal time constant of approximately 10ns. This would provide significant performance improvements for an analog neurocomputing system as depicted in Figure 5.16. It should be noted that comparison with a biological neuron in that figure is not an apples-to-apples comparison as it offers much higher computational power with its extraordinarily dense connectivity and impeccable functionality than hardware implementations mimicking just its high-level abstraction.

Table 5.1. Scaling equations for ovenized AlN resonators. w, l and t denote the width, length and thickness of the device, respectively. Device dimensions used throughout this chapter are in the form of $\{w \text{ x } l \text{ x } t\}$.

Device Property	**Relation to Device Dimensions**
Resonator Area	$\propto wl$
Resonator Impedance	$\propto wl/t$
Operating Voltage	$\propto (wt/l)^{1/2}$
Thermal Time Constant	$\propto l^2$
Energy for Max. Impedance Swing	$\propto wlt$

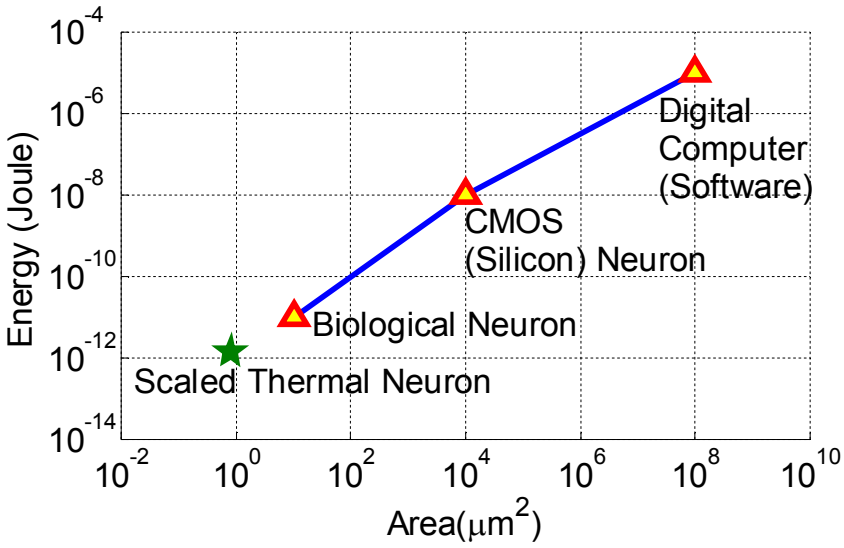

Figure 5.16. Modified energy and area comparison results for thermal neuron circuits, including predicted device improvements for ovenized AlN resonators.

Chapter 6

All-Graphene Analog Associative Memory and Neurocomputing

In this chapter we propose to construct a fully-functional analog neural network using multi-gate programmable graphene devices. Different from proposed designs in Chapter 4 and Chapter 5, this architecture represents couplings between neighboring neurons in voltage domain, and uses only electrical signals for both weighting and summing functions, thereby not requiring any special non-electrical properties for the enabling device. This chapter begins with a brief description of the multi-gate programmable graphene device that enables our proposed neuromorphic architecture. Next, we show our neurocomputing circuit design that is specifically configured for this kind of resistive devices, followed by our circuit simulation results to validate its correct operation.

6.1 Multi-Gate Programmable Resistive Device Using Graphene

Graphene is a two-dimensional material consisting of a single-atom thick layer of carbon that is arranged in a hexagonal lattice (see Figure 6.1). With its 2D planar structure it has been

shown to be compatible with traditional CMOS process [55]. Moreover, the charge carriers in graphene can move over great distances at a constant speed without scattering. This is similar to the behavior of photons travelling at the speed of light. In graphene the speed of charge carriers is slower than light by only a factor of 300 [56], resulting in high saturation velocity. Graphene also has high carrier mobility and zero energy band-gap. Thus, it offers unique opportunities for future nanoelectronics such as high frequency applications [57]. In addition, the linear dispersion relation in graphene gives rise to relativistic behavior of the charge carriers resulting in photon-like behavior and Klein tunneling [58].

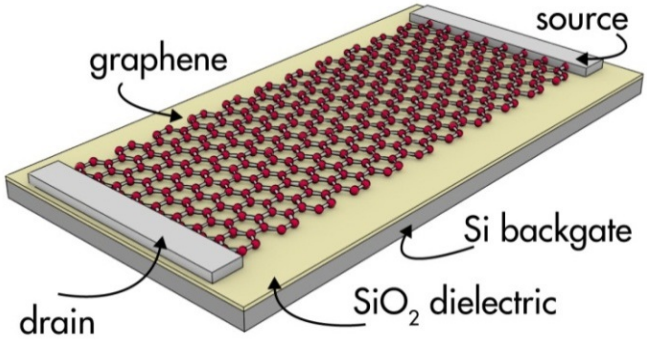

Figure 6.1. A graphene device [59].

Here we propose a novel graphene device configuration that could be used to represent artificial neurons and synapses. The number of charge carriers in a graphene ribbon can be controlled through local gating [60] that allows the gradual resistance change of the device. This is because the electric field between the gate and graphene ribbon due to applied gate voltage attracts electrons in graphene, thereby altering its conductivity. By means of this feature and zero energy band-gap offered by this particular material, a resistor string can be built using multiple gates over a single continuous graphene ribbon, as illustrated in Figure 6.2. Although the gates effectively dope the graphene creating multiple junctions, conductivity is maintained due to Klein tunneling at the junctions [58]. The contribution of each gate on the resistance level can be adjusted for specific purposes by simply altering the corresponding gate length (i.e., l_1, l_2

and l_3 in the figure). Such a multi-gate graphene device performs the DAC and summing functions that can be utilized for implementing artificial neurons and synapses without requiring any other data converters or CMOS amplifiers.

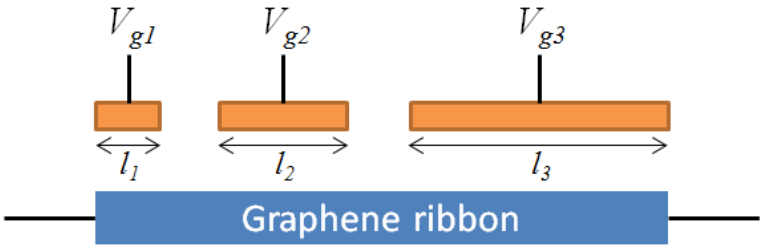

Figure 6.2. Cross-section drawing of the multi-gate graphene resistance. Each gate controls the conductivity of the corresponding area underneath it. V_{g1}, V_{g2} and V_{g3} represent voltage signals applied to these gates. The number of gates over graphene ribbon can be easily increased.

6.2 Proposed Neurocomputing Circuits Based on Graphene

The neuron-synapse building block for our proposed architecture is shown in Figure 6.3. Each artificial neuron is constructed as one pull-up (R_{PU}) and one pull-down (R_{PD}) graphene device with multiple gates. In combination they function similar to an *analog buffer* that produces intermediate values between high and low voltage levels. Each gate in these two devices has an equal impact on the total resistance (i.e., each having the same length, referring to Figure 6.2), and the number of gates per device in the neuron circuits is equal to the number of neighboring neurons. As such, these devices enable an efficient summing mechanism for weighted neural signals coming from neighboring synapses via equally-sized gate structure.

Each synapse between neighboring neurons is categorized into two components: i) *excitatory synapse* and ii) *inhibitory synapse*, as shown in Figure 6.3. Each of these components is represented by two multi-gate programmable resistances. Each gate in both devices controls a binary-weighted resistor based on the corresponding area on graphene ribbon (e.g., $l_1=x$, $l_2=2x$, and $l_3=4x$, referring to Figure 6.2). The number of control bits, i, can be

adjusted based on the requirements for the network function where the number of allowable resistance values is equal to $2^{i+1}-1$, because each synapse consists of two components. Generally 15-31 different synaptic values are enough for a wide range of neural network applications [8], [25]. This device configuration enables a compact D/A (digital-to-analog) conversion for digitally-controlled synaptic weights.

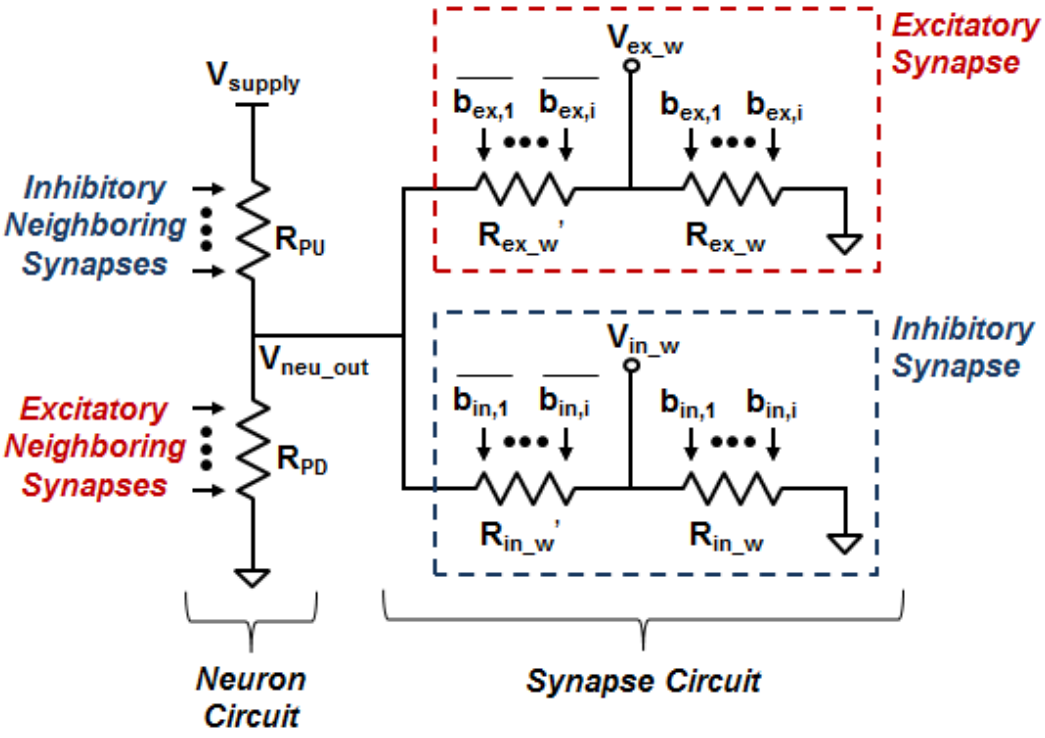

Figure 6.3. Proposed neuron and synapse circuits based on multi-gate graphene devices. $b_{ex,1}$-$b_{ex,i}$ and $b_{in,1}$-$b_{in,i}$ represent i-bits binary numbers for excitatory and inhibitory synaptic weights, respectively.

The total resistance of two devices in each synapse component is always constant since binary bits applied to them are complementary to each other (see Figure 6.3). Hence the couplings between neuron output and synaptic weights are linear as given by:

$$V_{ex\_w} = V_{neu\_out} \frac{R_{ex\_w}}{R_{total}} \quad for\ excitatory\ synapse \tag{6.1}$$

$$V_{in\_w} = V_{neu\_out} \frac{R_{in\_w}}{R_{total}} \quad for\ inhibitory\ synapse \tag{6.2}$$

where V_{neu_out} is the output of the corresponding neuron; R_{ex_w} and R_{in_w} denote excitatory and inhibitory synaptic weights, respectively; V_{ex_w} and V_{in_w} are weighted voltage signals corresponding to excitatory and inhibitory components, respectively; and

$$R_{total} = R_{ex\_w} + R_{ex\_w}{}' = R_{in\_w} + R_{in\_w}{}' \tag{6.3}$$

where $R_{ex\_w}{}'$ and $R_{in\_w}{}'$ are complementary versions of R_{ex_w} and R_{in_w}, respectively.

The excitation component of each synapse circuit is activated by increasing R_{ex_w} via digital control bits (i.e., $b_{ex,1}$-$b_{ex,i}$ in Figure 6.3) when neighboring neurons are coupled to each other via a *positive* synaptic weight (i.e., two neighboring neurons, each representing an image pixel, that are more likely to be of the same color, either white or black). The inhibition component is activated in the same way as the excitation component (i.e., via $b_{in,1}$-$b_{in,i}$ in Figure 6.3) when the relationship between neighboring neurons is represented by a *negative* synaptic weight (i.e., two neighboring neurons, each representing an image pixel, that are more likely to be of the opposite colors). A synaptic weight can be either positive or negative, so the corresponding synapse component is deactivated by setting R_{ex_w} or R_{in_w} to its lowest possible value (i.e., either $b_{ex,1}$-$b_{ex,i}$ or $b_{in,1}$-$b_{in,i}$ are set to 0).

An interesting property of this proposed architecture is that couplings between neighboring neurons are voltage-mode signals instead of current-mode signals, in contrast to other proposed resistive networks in Chapter 4 and Chapter 5. This is dictated by the gates being controlled by voltage-mode signals.

It is important to note that inhibitory synapses are connected to the gates of pull-up device in the neuron circuits, while excitatory synapses are connected to the gates of pull-down device, as shown in Figure 6.3. This is because weighted signals coming from artificial synapses increase

the resistance of the corresponding device. Therefore, when excitation becomes more dominant as compared to inhibition, the neuron circuit generates an output level that is closer to the supply voltage, V_{supply} in Figure 6.3. In contrast, when inhibition is more dominant, the neuron circuit produces an output that is closer to 0V (ground).

6.3 Circuit Simulations

To evaluate the potential of our proposed neurocomputing architecture in Figure 6.3 we have created a compact circuit simulation model in Verilog-A for the graphene device configurations based on device measurement data [60]. The relationship between graphene resistance and applied gate voltage is modeled as a saturation function with two cut-off voltages (i.e., V_{cut_min} and V_{cut_max}). This function outputs the lowest resistance value (R_{min}) when applied gate voltage is less than V_{cut_min}, and the highest resistance value (R_{max}) when applied gate voltage is higher than V_{cut_max}. When it is in-between these two voltage levels the device resistance increases linearly with applied gate voltage (from R_{min} to R_{max}).

In our circuit simulations we used $V_{supply}=1V$, $V_{cut\_min}=0.1V$, $V_{cut\_max}=0.9V$, and $R_{max}/R_{min}=100$ which is achievable with today's technology [60]. This R-V characteristic can be attained by means of a properly-selected reference voltage corresponding to the ground in our circuits (e.g., around -3V for the device example in [60]), or a carefully-tuned back-gate/substrate voltage, or both. We connected 1pF capacitor at the output of each neuron circuit to model device parasitics.

The devices used in the neuron circuits have nine equally-weighted gates (i.e., $l_1=l_2=...=l_9$) so that each neuron has nine local connections in the network. The devices used in the synapse circuits, however, have three binary-weighted gates (i.e., $l_1=2l_2=4l_3$) that allows 15 different synaptic weights that can be defined in the model (considering both excitatory and inhibitory synapse components). The device resistances in the synapse circuits are designed to be 100

times greater than those in the neuron circuits in order to prevent significant deviations in the neuron output voltages from their actual values due to fanout.

As a first example we constructed a 5-neuron associative memory based on the architecture proposed in Figure 6.3. Using pattern examples [1 0 1 1 0] and [1 0 1 0 1] as memorized patterns by programming excitatory and inhibitory synapses accordingly using *Hebbian Rule* [61], we evaluated the convergence of this small network when other patterns were provided as inputs. Figure 6.4 shows a pattern recognition example. Even though there is 40% distortion in the initial input pattern [0 1 1 1 0], associative memory successfully recalls stored pattern that most closely resembles to applied input pattern.

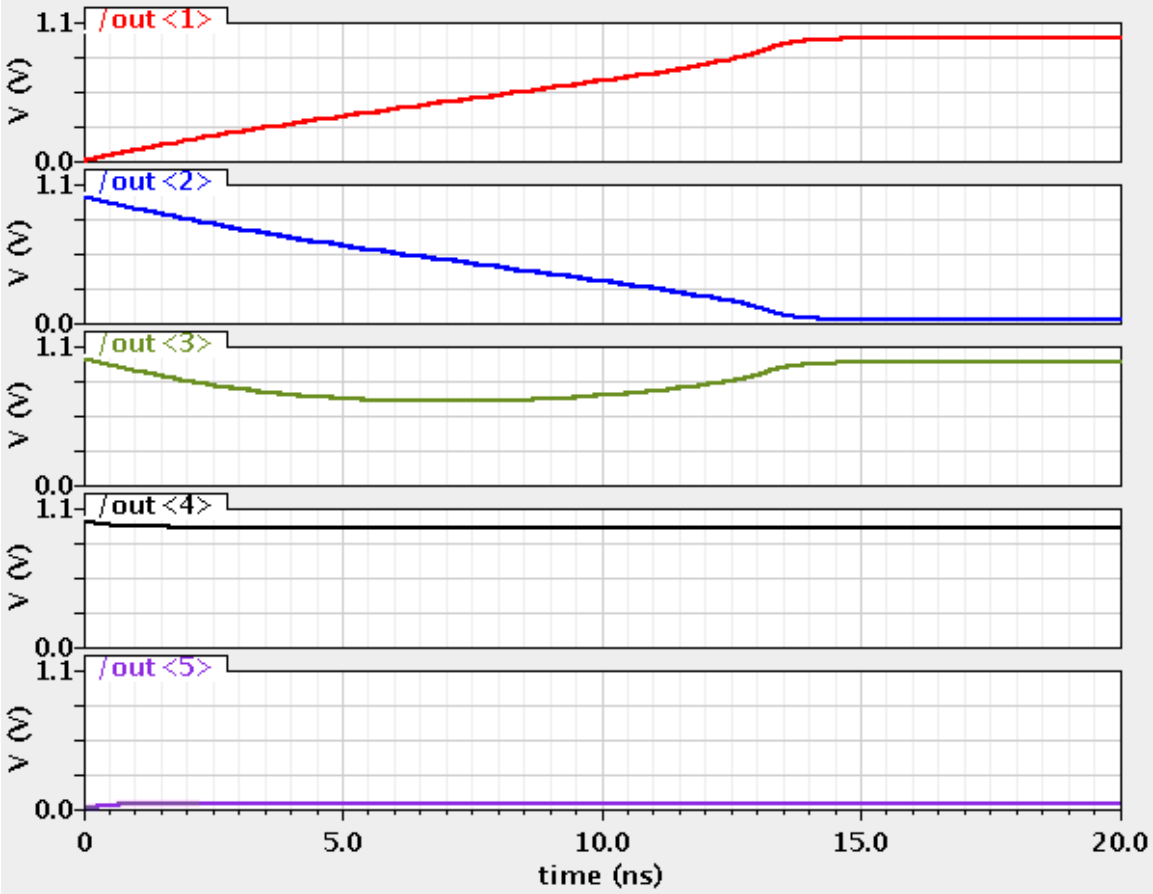

Figure 6.4. Pattern recognition example based on a 5-neuron system. Associative memory fully recognizes the pattern [1 0 1 1 0] despite 40% distortion in the initial input pattern.

Figure 6.5 demonstrates a second pattern recognition example for the same 5-neuron system, but now using gray-scale pixels represented by intermediate input signal values. The system correctly recognizes the pattern [1 0 1 0 1] as the closest memorized pattern to the initial gray-scale input pattern.

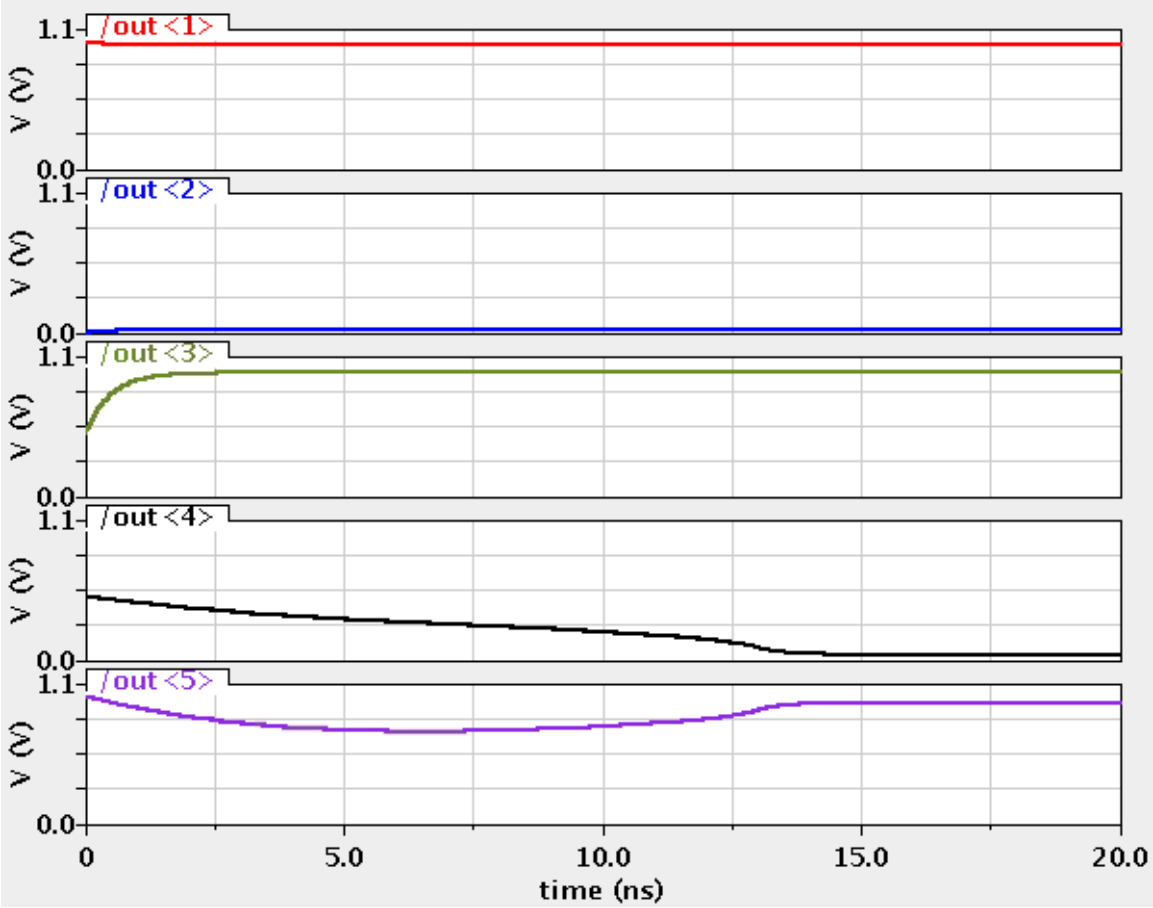

Figure 6.5. Pattern recognition example based on a 5-neuron system. Intermediate pixel values are possible as initial conditions by means of multi-gate structure of graphene devices. For this example the initial input pattern is [1 0 0.5 0.5 1] at time=0s and the recovered output pattern is [1 0 1 0 1] at time=20ns.

We also constructed 20-pixel gray-scale example for larger neuromorphic circuits with local interconnections among artificial neurons (e.g., 9 connections for each neuron) based on the architecture in Figure 6.3. For illustration we show here the same device simulation models to

evaluate an associative memory consisting of 20 neurons. The bit patterns shown in Figure 4.4 are examples of memorized patterns that are programmed into the neural network circuit model via synaptic weights.

As an initial input pattern we used a 28% distorted version of the bit pattern '5' with several gray-scale pixels, and our associative memory successfully recovers the bit pattern '5' as illustrated in Figure 5.14, similar to our thermal neural network proposed in Chapter 5. Importantly, the results here indicate that our proposed neuromorphic circuits scale well with the number of bits as long as the nearest neighbor coupling is sufficiently accurate.

Next we highlight the use of the same 20-neuron network for other image processing applications such as edge and line detections. In order to program excitatory and inhibitory synapses we used the cloning templates for edge and line detection applications given in [8], [34]. Circuit simulation results match the ones in Figure 4.6, thus confirming correct functionality of our proposed all-graphene analog neurocomputing circuitry.

6.4 Device Scaling and Improvements for Affordable All-Graphene Neurocomputing Circuits

With proper selection of back-gate voltage minimum device dimensions that are achievable with today's technology (i.e., nanoscale) could be used for proposed graphene devices while still attaining desired high-to-low graphene resistance ratio (e.g., 100). A sheet resistance of around 1kΩ could be achieved for a 0.1nm thick graphene device using today's technology [62]. A minimum size device can be used in the neuron circuits, and larger devices in the synapse circuits (e.g., 100x larger). Figure 6.6 shows tradeoff between power and area for our proposed all-graphene associative memory with the device width of 10nm. With smaller device dimensions this network would provide better performance for neurocomputing.

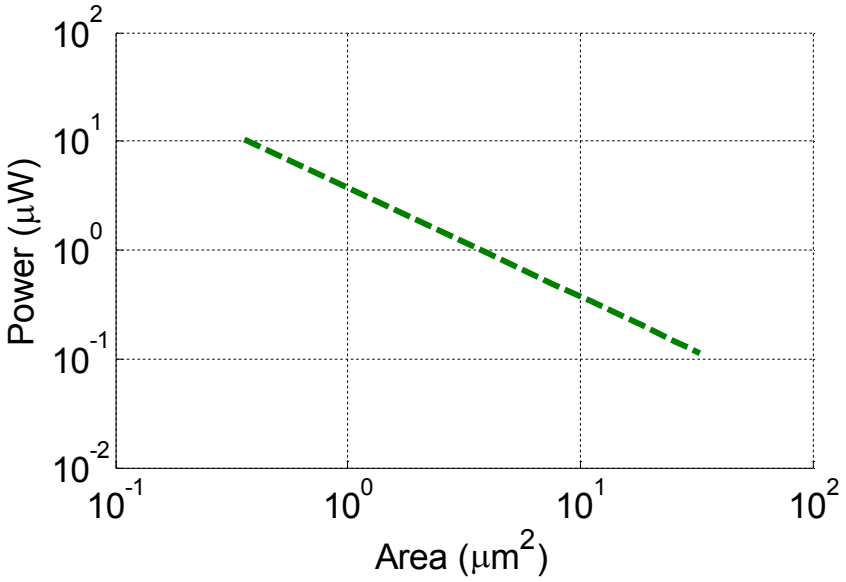

Figure 6.6. Power-area tradeoff per neuron with nine synapses for all-graphene associative memory with a minimum length ranging from 10nm to 1μm.

Furthermore, in order to reduce power consumption for this all-graphene associative memory the sensitivity of graphene resistance change for applied gate voltage can be increased (e.g., more than 100x change in graphene resistance for an applied gate voltage of 1V). This can be achieved by decreasing the thickness of gate oxide so that the electric field between the gate and graphene ribbon could increase for the same applied gate voltage. This would provide either larger high-to-low resistance ratio for the same amount of increase in applied gate voltage or lower operation voltages required for such networks (e.g., less than 1V), both resulting in less power dissipation for the entire system.

Chapter 7

Proof-of-Concept 65nm CMOS Analog Neurocomputing Chip

In this chapter we illustrate a CMOS emulation circuitry for our proposed analog neurocomputing scheme. This chapter first scrutinizes the design of this 65nm CMOS chip and then gives our experimental results for various image processing applications. The chapter ends with our concluding remarks regarding the chip performance.

7.1 Design of Analog CMOS Emulation Circuitry

In this section we examine a circuit topology for use as an analog neurocomputer that is cultivated based on the conceptual architecture depicted in Figure 2.7. The key characteristics this design has to possess are the efficient current summation for neural signals coming from artificial synapses, the input-output isolation at the neuron circuits, and the current-to-voltage conversion for output generation based on a sigmoid-like transfer function.

In the conceptual circuit diagram in Figure 2.7 both neuron outputs and synaptic weights can be either positive or negative. To represent all four possible neuron-synapse interactions,

the neuron circuits are split into two identical components: i) *excitation neuron* and ii) *inhibition neuron*. Each component is implemented as a two-stage summing amplifier (i.e., a two-stage operational amplifier with a resistive feedback) [63], followed by a large analog inverter, as drawn in Figure 7.1. The first stage is a single-ended differential amplifier consisting of five transistors, providing high gain. The second stage is built as a class-AB common source amplifier with two transistors, empowering high output swing and moderate gain. In addition, a large inverter is used at the output stages to achieve a sigmoid-like transfer function while producing sufficient current to artificial synapses.

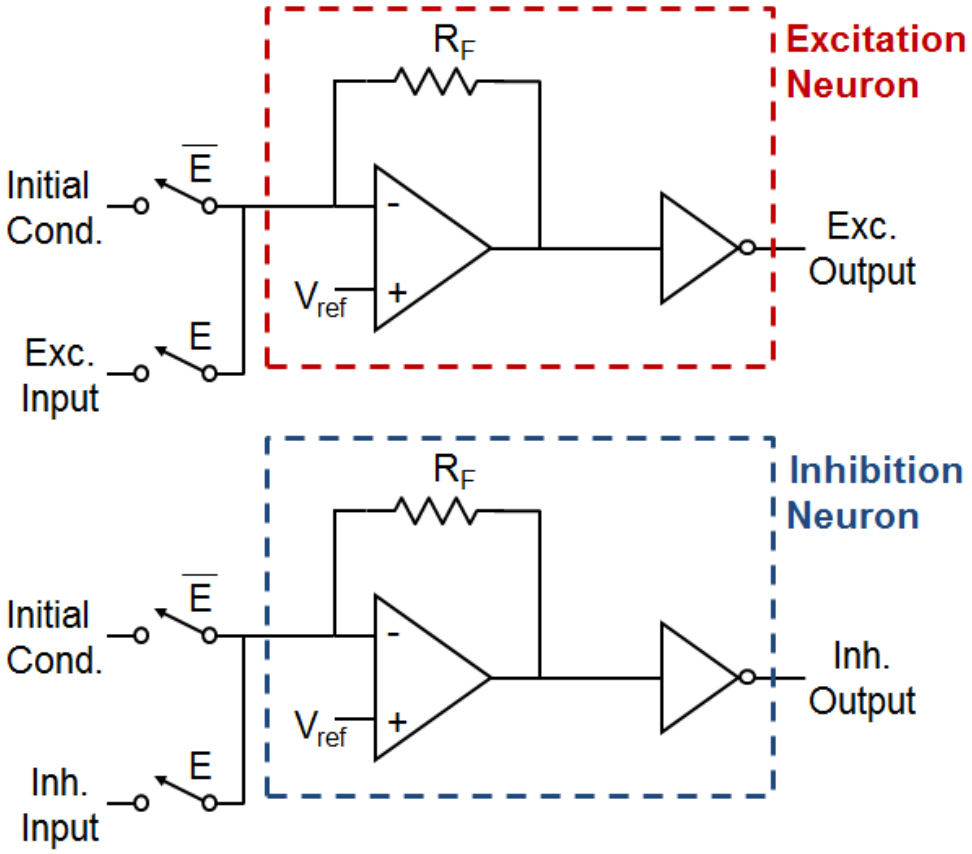

Figure 7.1. The CMOS neuron circuit consisting of two identical components.

The reference voltage, V_{ref}, in that figure corresponds to the average of maximum and minimum neuron output voltage levels as in:

$$V_{ref} = \left(V_{\max} + V_{\min}\right)/2 \tag{7.1}$$

where V_{max} and V_{min} denote maximum and minimum neuron output voltages. In our case V_{ref} is equal to 0.6V (i.e., {1.2V+0V}/2).

Moreover, the excitation and inhibition neuron outputs change in opposite directions with respect to the reference voltage as given by:

$$V_{ex} + V_{in} = 2V_{ref} \tag{7.2}$$

where V_{ex} is the excitation neuron output voltage and V_{in} is the inhibition neuron output voltage. For example, if the excitation neuron output is 0.9V in our hardware implementation, then the inhibition neuron will generate an output voltage of 0.3V (i.e., 1.2V-0.9V).

Since the differential amplifiers in Figure 7.1 are configured in a negative feedback loop, the negative input terminals serve as summing nodes for neural signals coming from neighboring synapses. These terminals also act as a virtual ground, thereby ensuring a fixed DC voltage at these nodes that is equal to V_{ref}. As such, artificial synapses can properly weight neural signals based on the voltage difference between the neighboring neuron output voltage and this reference voltage.

The designed CMOS neuron circuit in Figure 7.1 can operate at two modes. The first one is the *initialization mode* in which initial conditions are applied to the neuron circuits. The second mode is the *evaluation mode* that constitutes the neural operation to generate the corresponding output pattern based on the function of the neural network set by synaptic weights. The *E* signal in the figure determines at which mode the neural network operates.

Similar to the neuron circuits, the synapse circuits are also divided into two parts: i) *excitatory synapse* and ii) *inhibitory synapse*. Each part is formed by three binary-weighted

resistances in parallel controlled by digital data and switches as shown in Figure 7.2. This guarantees 15 different resistance levels that can be defined in this analog CMOS neurocomputing circuit. The last two switches in these circuits set the type of artificial synapse (i.e., the sign of synaptic weight) via the *T* signal. This signal controls the switches in such a way that two neuron outputs are connected to different inputs of neighboring neurons via artificial synapses. For example, when the synaptic weight is positive, the *T* signal becomes high, allowing the current flow between the excitatory neuron output and the excitatory input of neighboring neuron and between the inhibitory neuron output and the inhibitory input of neighboring neuron. Likewise, when the synaptic weight is negative, this signal is pulled down, thus swapping interconnections between neighboring neurons (i.e., the signs of synaptic weights).

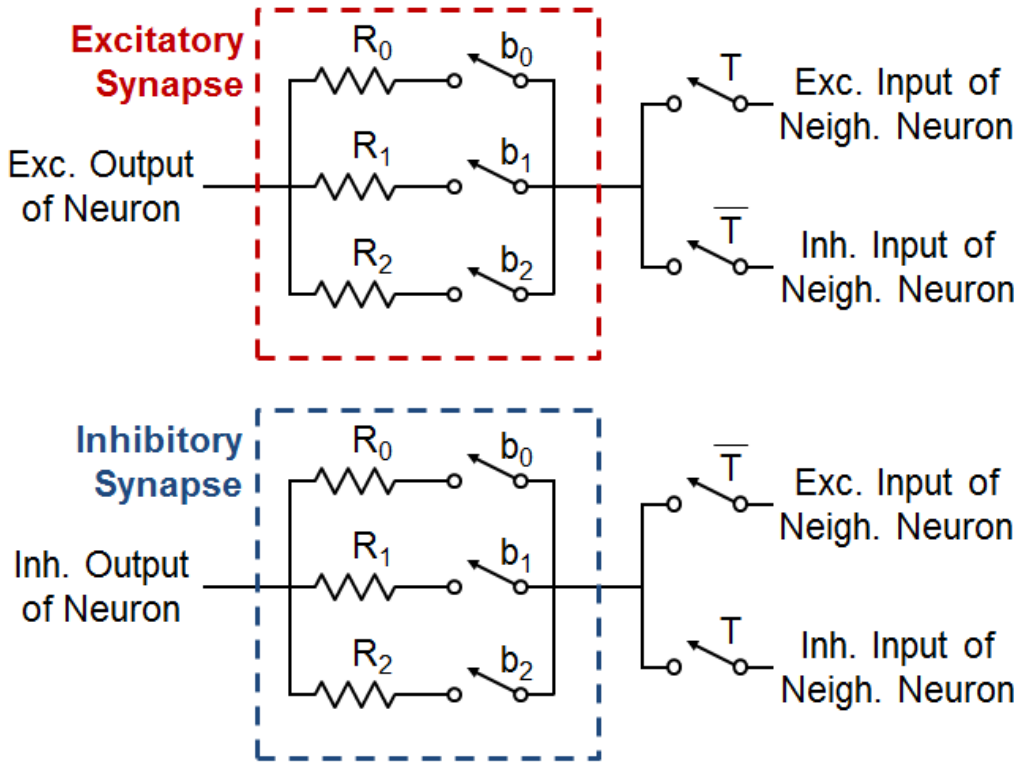

Figure 7.2. Excitatory and inhibitory synapse designs using CMOS technology.

The most important caveat to take during laying out these circuits is the proper distribution of the *E* signal among neuron circuits. In other words, all artificial neurons must get this signal at the same time for correct mode switches so as to prevent signal skew that could be caused by large mismatches in signal routing. *The h-tree circuit technique* that is used for clock distribution in digital circuits [64] is seemingly the most effective way to allow right circuit operation for such purposes.

7.2 Experimental Test Results

We have built an analog neurocomputing circuit described in the previous section using 65nm CMOS technology that contains 20 neuron circuits in a 5x4 grid, each having nine synapse circuits connected to the nearest neighbors (see Figure 7.3). In this section we share our test results, including the chip performance and several image processing applications.

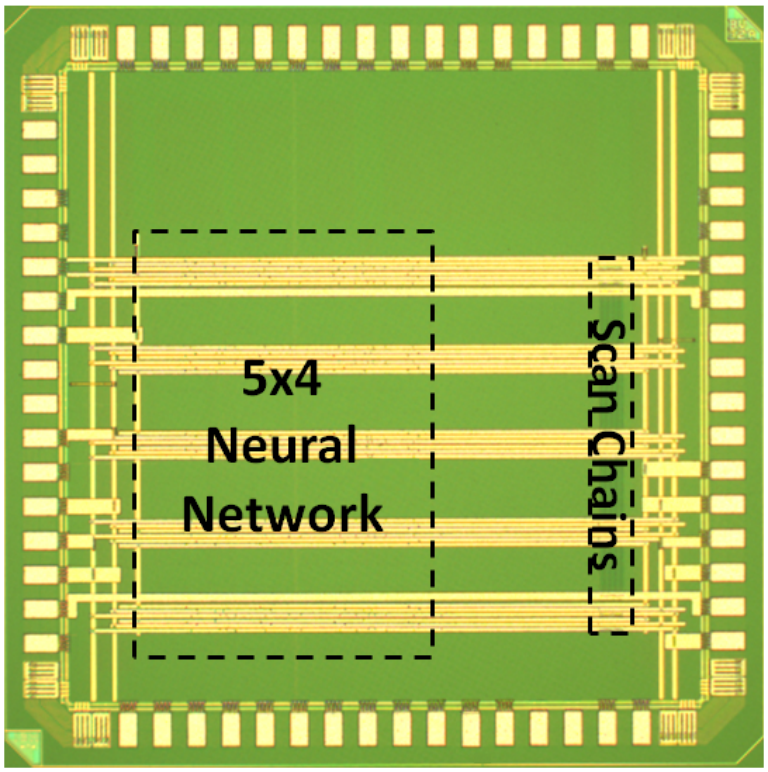

Figure 7.3. Micrograph of a 20-neuron network implemented in 65nm CMOS.

Table 7.1 tabulates the performance metrics for this 65nm CMOS chip. The digital part of the chip corresponds to the D flip-flop chains that scan control data for initial conditions and synaptic weights into the neural network while the rest of the circuit (i.e., the whole neural network design) constitutes the analog part. With about 10ns settling time this design requires more than 160x less energy and around 1.5x less area per neuron when compared to the best reported analog CMOS implementation [1]. Hence, these results set the upper limits for area and power consumption as well as serving as a reliable point of comparison in building feasible associative memories and neurocomputing circuits enabled by emerging technologies.

Table 7.1. Chip performance. The neuron-synapse module is composed of one neuron circuit with nine artificial synapses.

System Property	Value
Chip Area	4mm^2
Area of One Neuron-Synapse Module	0.0258mm^2
Area of One Neuron Circuit	0.0074mm^2
Average Power of Analog Part	122mW
Average Power of Digital Part	13.4μW
Average Power of One Neuron-Synapse Module	6.1mW

We first programmed artificial synapses in application to pattern recognition using Hebbian Rule described in [61] for training the network based on memorized patterns given in Figure 7.4 (a), since it achieves the best pattern retrieval performance in the presence of various noise sources when compared to other training algorithms used for such a locally-interconnected architecture [61]. The system successfully recalled stored patterns from distorted inputs as displayed in Figure 7.4 (b)-(c). These testing results match the ones in Figure 4.5 and Figure 5.14, validating correct operation of this CMOS emulation circuitry.

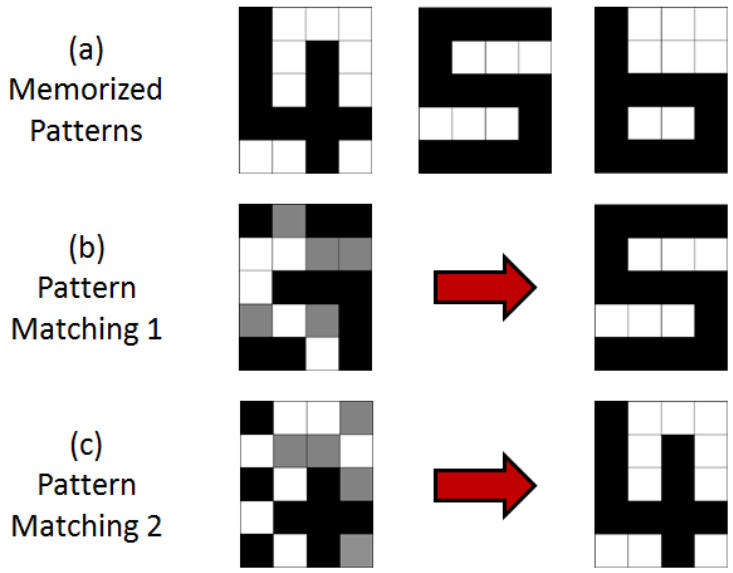

Figure 7.4. Pattern matching examples. (a) 20-pixel memorized bit patterns. (b) Example 1: input pattern (left) and output pattern (right). (c) Example 2: input pattern (left) and output pattern (right).

We then extended our pattern matching application to digit recognition by storing all ten digits (see Figure 7.5 (a)) in the network via artificial synapses. We used 25% distorted versions of bit patterns as initial inputs to the network as shown in Figure 7.5 (b)-(d). Our associative memory circuit successfully retrieves output patterns that most closely resemble to the initial input patterns, as can be seen in Table 7.2.

Table 7.2. Hamming distance as a percentage between stored and initial input patterns for three digit recognition examples in Figure 7.5 (b)-(d).

Input Patterns	**Hamming Distance to Stored Digits**									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Example #1	30%	30%	55%	45%	60%	55%	65%	**25%**	55%	40%
Example #2	70%	40%	35%	**25%**	60%	40%	45%	45%	40%	40%
Example #3	55%	70%	35%	30%	55%	35%	45%	55%	**25%**	45%

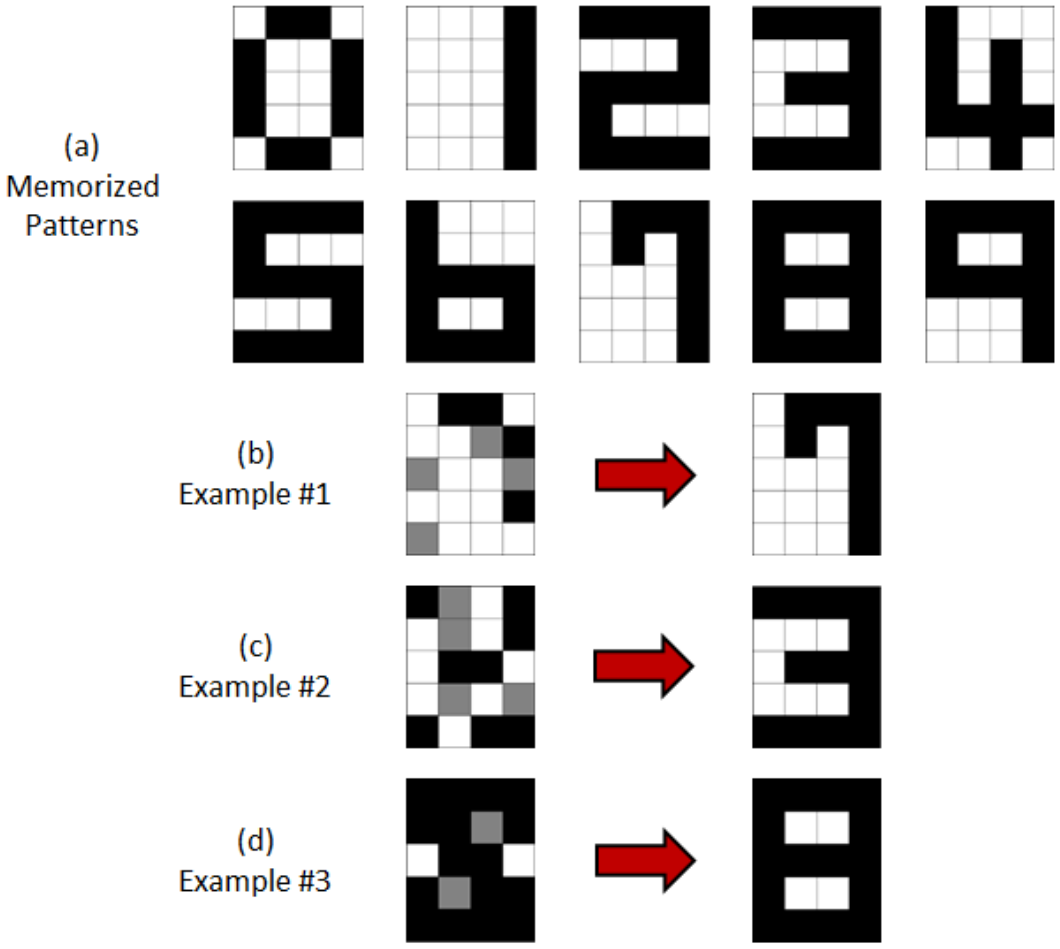

Figure 7.5. Digit recognition. (a) 20-pixel memorized bit patterns. (b) Example 1: input pattern (left) and output pattern (right). (c) Example 2: input pattern (left) and output pattern (right). (d) Example 3: input pattern (left) and output pattern (right).

Next, we adjusted synaptic weights for various image processing applications such as edge detection and hole filling. We leveraged the 3x3 cloning template for each application provided in [8], [34]. Figure 7.6 demonstrates our experimental results for five image processing applications. Recalled output patterns are the same as the ones in Figure 4.6, again verifying our analog neurocomputing system implemented in 65nm CMOS.

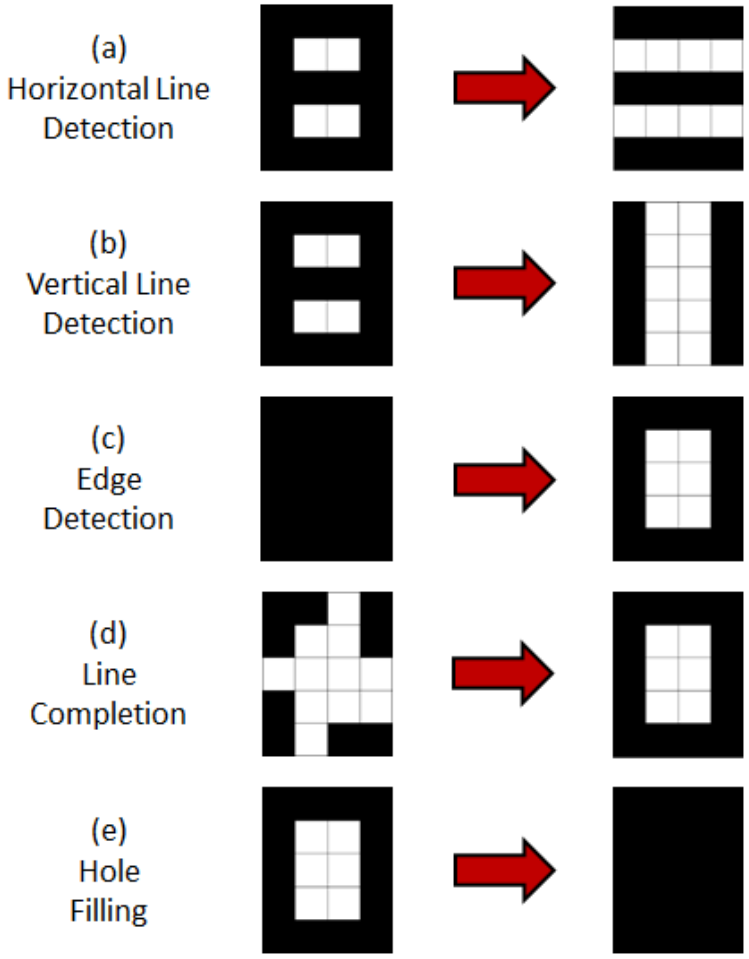

Figure 7.6. Testing results for different image processing applications. Input patterns (left) and output patterns (right). (a) Horizontal line detection. (b) Vertical line detection. (c) Edge detection. (d) Line completion. (e) Hole filling.

7.3 Comparisons to Our Proposed Analog Neural Networks Based on Emerging Technologies

Figure 7.7 depicts performance comparison results for the 65nm CMOS emulation circuitry relative to our proposed analog neurocomputing systems rigorously explained in Chapter 4, Chapter 5 and Chapter 6 in terms of power and area. This figure attests that unlike mature CMOS technology that has nearly reached its scaling limits emerging technologies and materials

offer unique properties that can be efficiently exploited in the practical realization of neurocomputing circuits and associative memories.

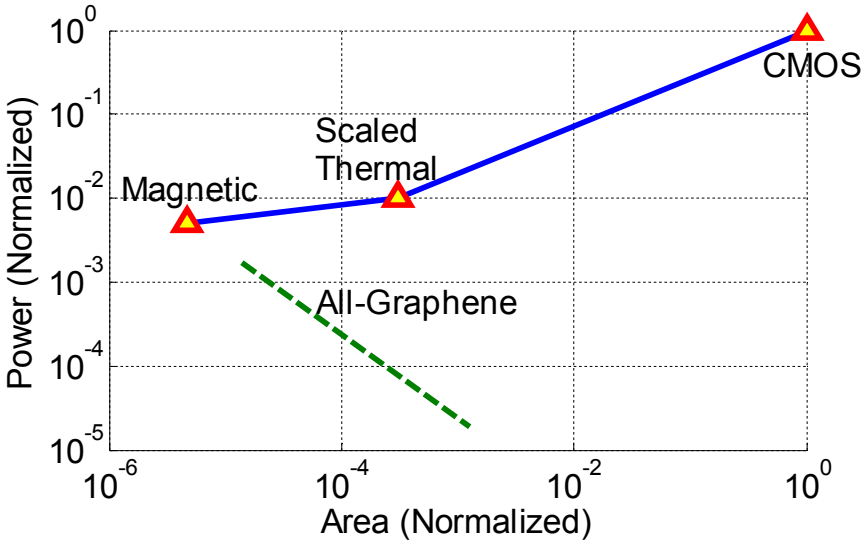

Figure 7.7. Performance comparison of CMOS, magnetic, all-graphene, and (scaled) thermal analog neural networks.

Chapter 8

Conclusion and Future Directions

The goal of this thesis has been to explore efficient and viable circuit and system solutions for associative memories and neurocomputing circuits based on emerging post-CMOS technologies such as magnetic and thermally-controlled devices by exploiting their distinctive features. All the proposed devices and technologies in this study offer great opportunities and could enable the practical realization of neurocomputing systems and associative memories.

In Chapter 2 we have proposed the co-optimization of technology and architecture for the practical realization of neurocomputers as previously-proposed approaches based on CMOS and/or emerging technologies do not offer a feasible solution for neurocomputing. In this respect we have developed required device properties to enable such systems in a compact and efficient way based on a conceptual circuit diagram.

In Chapter 3 we have proposed highly-efficient hardware implementations of fully-digital CNN and ONN using non-volatile, all-magnetic logic technology, mLogic. Proposed architectures offer *re-configurability* and highly-efficient *pipelining* by exploiting local storage of the mCell state, and require minimal connections with CMOS. We have validated true functionality of these systems using MATLAB-based behavioral simulations. Comparison results

to 32nm CMOS technology demonstrate that this new all-magnetic logic technology could enable the practical implementation of low-voltage, low-power, and nanoscale fully-digital associative memories.

In Chapter 4 we have proposed a design based on the use of a newly-proposed STT-MTJ device, mCell, to enable all-magnetic analog associative memories and neurocomputing systems. Simplified programmability and non-volatility of these devices preclude the need for DACs to convert digital control data into an analog signal and eliminate the need for additional memory elements to store such control data, thereby enabling efficient implementation of artificial synapses. Electrically-isolated read- and write-paths of the mCell devices allow efficient transformation of current-mode signals generated via artificial synapses into voltage-mode neuron outputs.

In Chapter 5 we have proposed a novel configuration for ovenized AlN resonators to efficiently implement artificial synapses and neurons, which are never efficiently built with CMOS exclusively. Thermal power due to multiple heaters on a single device is controlled by a digital input and naturally summed together to adjust the resonator impedance, and provide an extremely efficient D/A conversion for each synapse and an efficient building block for each neuron. We have shown the design of proposed DAC by considering possible design challenges caused by the non-linear dependency of the resonator impedance on the heater power and heater resistance change due to temperature.

In Chapter 6 we have proposed and designed a complete analog neurocomputing circuitry based on the multi-gate programmable graphene resistances that provide a natural summing and compact D/A conversion for the implementation of artificial neurons and synapses. The operation of proposed architecture has been confirmed using simple gray-scale pattern recognition and image processing examples.

In Chapter 7 we have demonstrated a proof-of-concept analog neurocomputing chip implemented in 65nm CMOS technology. It consists of 20 neuron circuits, each having nine synapse circuits as in CNN. Testing results coincide with image processing examples presented in Chapter 4, Chapter 5 and Chapter 6. Performance results constitute an upper bound for future neurocomputing circuits enabled by emerging technologies and materials.

8.1 Future Considerations

This research is intended to study novel system solutions for neurocomputing and associative memories using various types of emerging materials. Hence, the proposed implementations in this thesis might be extended to other cutting-edge materials and technologies that can be classified under the same categories that exhibit similar characteristics. For example, the use of all spin logic (ASL) [65] could be exploited to enable magnetic neural networks; and vanadium dioxide [66] or phase change materials [13] for thermal neural networks.

To expand the architectural diversity for neurocomputing circuits the next sub-sections explore high-level architectures based on novel device and circuit topologies using graphene and RRAM as emerging technologies. These sub-sections investigate what else could be accomplished, leading to new research opportunities for distinct kinds of neural networks and serving as an excellent starting point for designing viable neurocomputing systems based on such beyond-CMOS devices.

8.1.1 All-Graphene Neural Network Enabled by Evanescent Wave Coupling

In this sub-section we propose a novel graphene-based device configuration that could be used to represent artificial neurons. Two graphene ribbons in close proximity are coupled to each other via evanescent wave transport [67]. The current through one graphene ribbon becomes proportional to the current through other via evanescent wave coupling based on the

distance between them (see Figure 8.1). Such a device configuration, which we term *graphene coupler*, can be used to represent artificial neurons. This also enables the electrical isolation between the input and output of the neuron circuits as required.

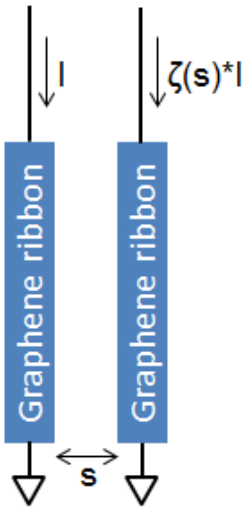

Figure 8.1. The graphene coupler. The current through the second graphene sheet is affected by the current through the first one and the distance between the two. ζ denotes the coupling coefficient between two graphene ribbons and is a function of s, the distance between these ribbons.

8.1.1.1 Proposed Architecture

Figure 8.2 demonstrates our proposed all-graphene associative memory that is based on the conceptual neural network architecture depicted in Figure 2.7. Variable resistors in that figure correspond to the graphene devices with binary-weighted gates construed in Section 6.1. In this circuit representation the currents coming from neighboring synapses are efficiently summed together via a small-resistance graphene ribbon. This summation is then converted to an analog output voltage via a simple pull-up/pull-down network, which is enabled by current coupling between two graphene sheets. We hope our findings here could open up new avenues and brings about great opportunities for leveraging graphene in neurocomputing circuits and associative memories.

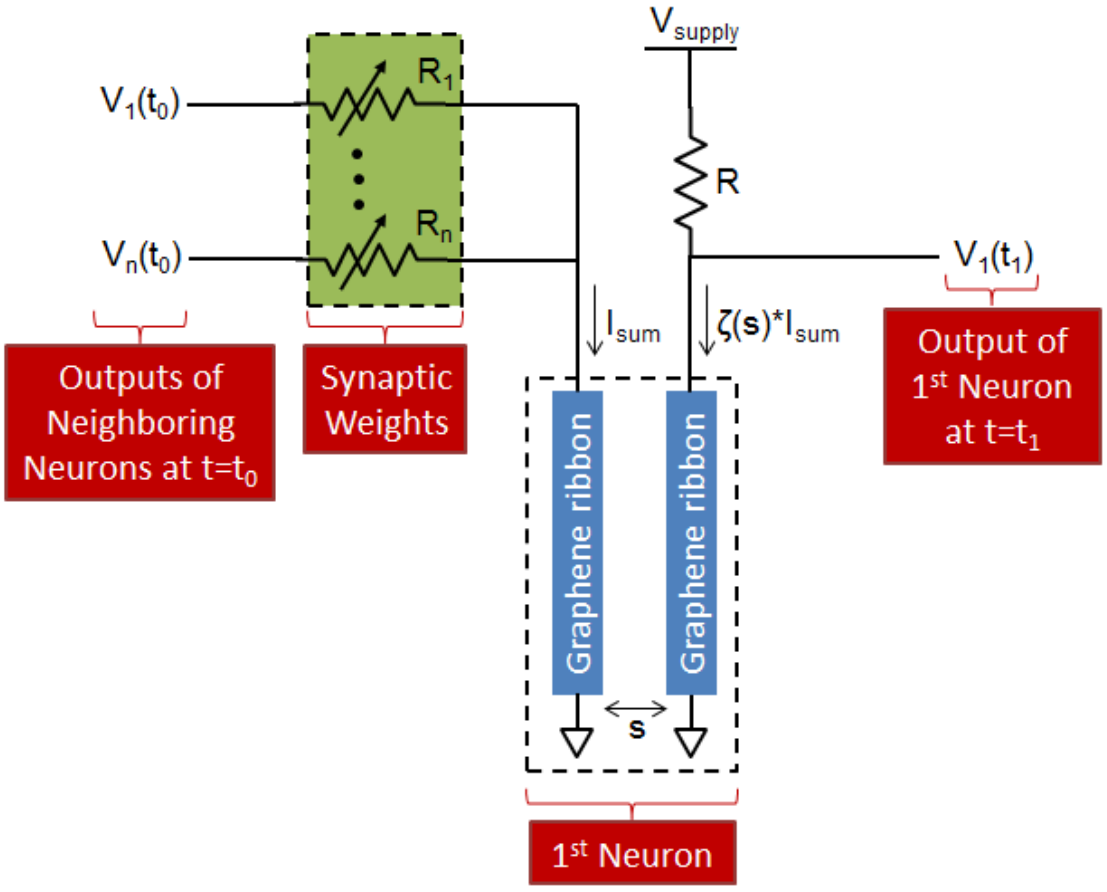

Figure 8.2. Conceptual circuit diagram for our proposed all-graphene analog neurocomputing circuit. I_{sum} represents the total current coming from neighboring synapses.

8.1.2 Oscillatory Neural Networks Using Low-Power RRAM Oscillators

RRAMs have already been demonstrated for use as artificial synapses in neurocomputing circuits as highlighted in Chapter 2. Such devices offer three orders of magnitude gradual resistance change via pulse-based programming [9]-[10]. In this sub-section we examine the use of a low-power RRAM oscillator, along with RRAM synapses, to build an oscillatory neural network with minimal CMOS. The main difference between ONN and our proposed analog neurocomputing topology in Chapter 2 is that information is carried by phase rather than amplitude. We believe such a system proposal can unleash new methods and paths for both device/material development and neurocomputing systems.

8.1.2.1 The RRAM Oscillator

State switching between high and low resistances in an RRAM device can be attained by changing the amount of current flowing through it (e.g., via Joule heating [68]). Using this feature and with proper biasing the continuous switching of an RRAM device could be sustained. For example, an RRAM oscillator can be built using a simple circuit setup shown in Figure 8.3 (left) (i.e., adding a resistance in series with the device). Figure 8.4 demonstrates the oscillation behavior of this circuit. Different oscillation frequencies can be obtained by altering the series resistance as illustrated in Figure 8.3 (right), since this affects the strength of the filamentation. Therefore, by exploiting more complex circuit topologies using RRAM devices a voltage-controlled oscillator could be built without requiring inductors and capacitors, thus allowing low-power operation. In spite of further need for exhaustive analysis and research to construct a complete oscillator circuitry, such an oscillator could offer crucial opportunities for enabling low-power oscillatory associative memories as mentioned in the next sub-section.

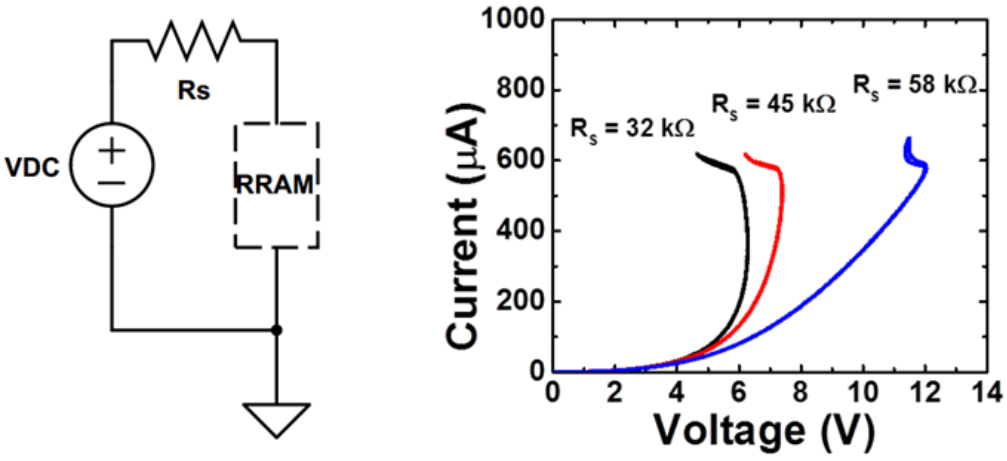

Figure 8.3. A simple RRAM oscillator setup (left) and state switching of RRAM with various series resistances (right) *(Courtesy of Jeffrey Weldon's group from Carnegie Mellon University)*. R_S denote the series resistance.

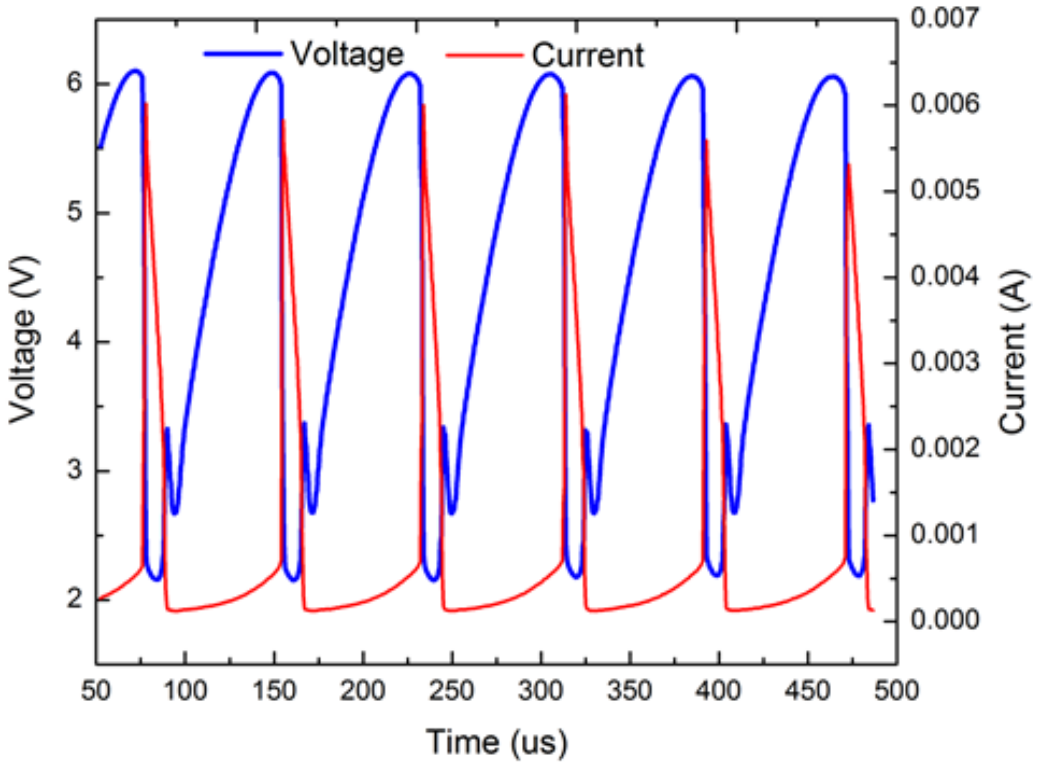

Figure 8.4. Oscillation behavior of the RRAM oscillator when R_s is 379Ω *(Courtesy of Jeffrey Weldon's group from Carnegie Mellon University)*. The oscillation frequency is 12.7kHz.

8.1.2.2 Proposed ONN System

Figure 8.5 demonstrates our proposed ONN system that is enabled by RRAM devices, based on the conceptual architecture in Figure 1.2. Each artificial synapse is represented by a single RRAM device while each neuron circuit is formed by an RRAM oscillator and a capacitor. The capacitor sums the currents coming from neighboring synapses, and forms a low pass filter with RRAM synapses for signal averaging as required (see Section 3.3.1). XOR logic gates calculate the phase differences between the neighboring neuron outputs, and are implemented using CMOS technology [69]. This architecture requires minimal connections to CMOS devices.

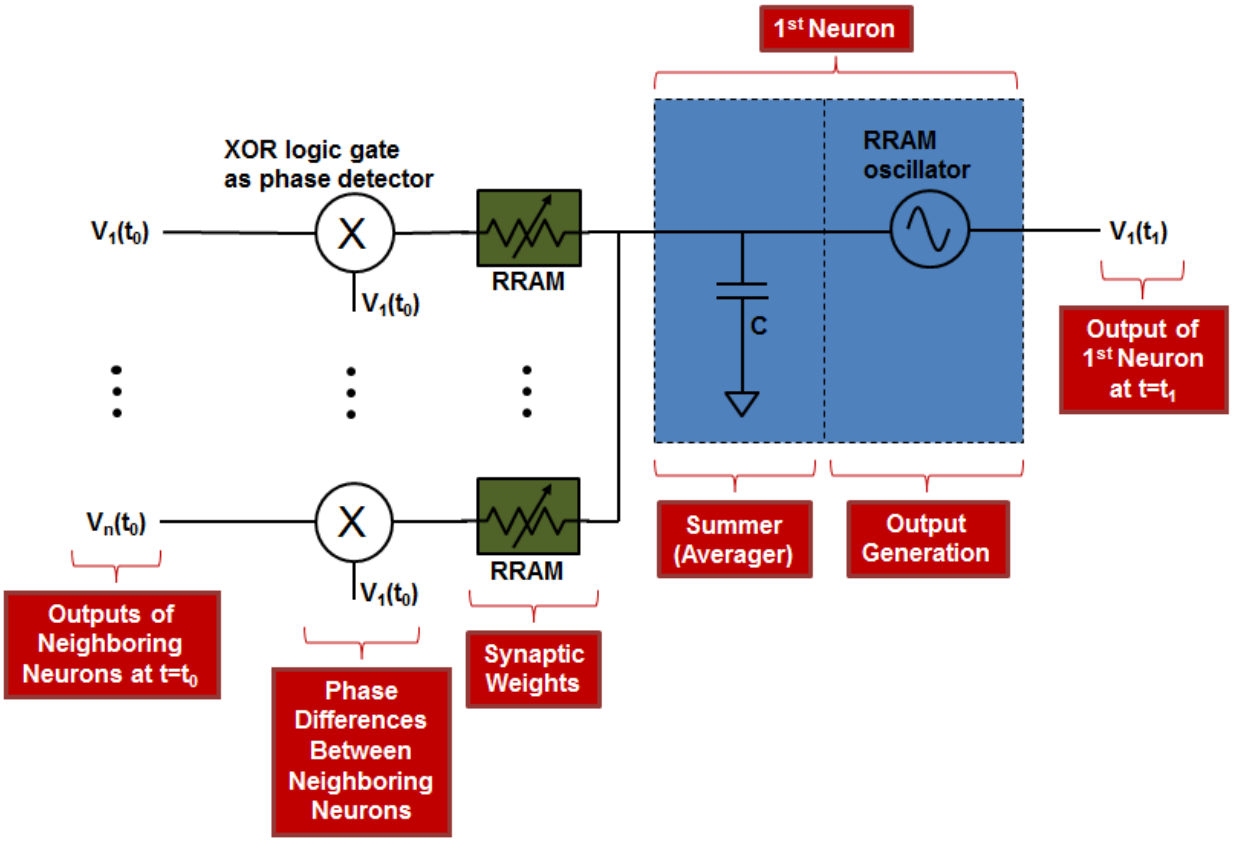

Figure 8.5. Proposed ONN architecture enabled by RRAM with minimal CMOS.

8.1.2.3 Preliminary Results

As a preliminary work to show the potential of proposed architecture depicted in Figure 8.5 we built a small network consisting of 5 neuron circuits and 25 synapses using Verilog-A models for the XOR gates and voltage-controlled RRAM oscillator with 10% tuning range around a center frequency of 1GHz, based on that figure. We used 3-input XOR gates with the last bit representing the sign of corresponding synapse (excitatory or inhibitory), thereby providing great flexibility for programming synaptic weights. The RRAM oscillator was modeled as a standard VCO with a gain of 100MHz/V. We configured the RRAM synapses based on two stored patterns, [1 0 1 0 1] and [1 0 1 1 0]. The circuits were powered with +/-0.5V supply voltages so as to prevent resistance shift in the RRAM devices [9]-[10].

When we apply stable states as initial inputs, the network stays in the same synchronized states as expected (see Figure 8.6 and Figure 8.7). We then apply noisy patterns as initial inputs. Figure 8.8 demonstrates the convergence of this network when there is 50% distortion in the second pixel. The pattern with two 50% noisy pixels (second and third) is successfully recovered via synchronization of this small network as depicted in Figure 8.9. Figure 8.10 shows associative recall of the memorized pattern, [1 0 1 0 1], when the initial input has 25% distortion in the fifth pixel and 50% distortion in the second and third pixels. Although these results seem to be promising, more analysis and research are still required for this implementation to pass the small prototype stage.

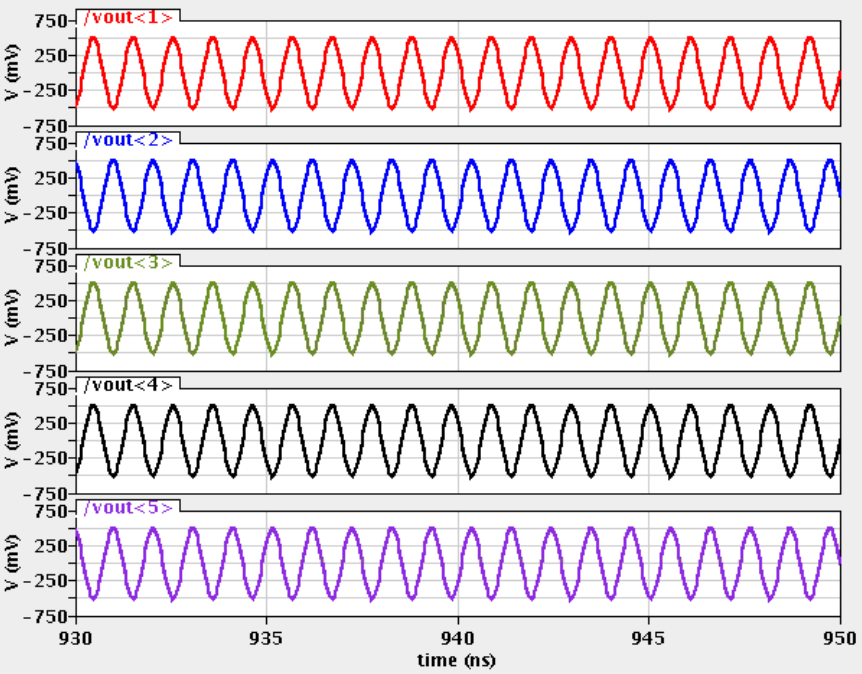

Figure 8.6. The system stays in the locked state with stored pattern, [1 0 1 1 0], as initial state.

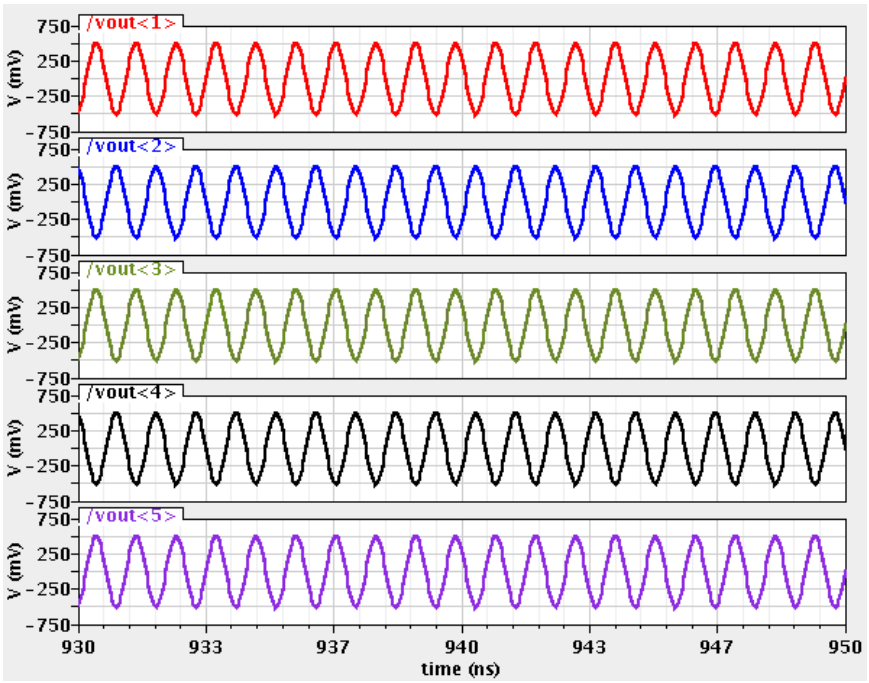

Figure 8.7. The system stays in the locked state with stored pattern, [1 0 1 0 1], as initial state.

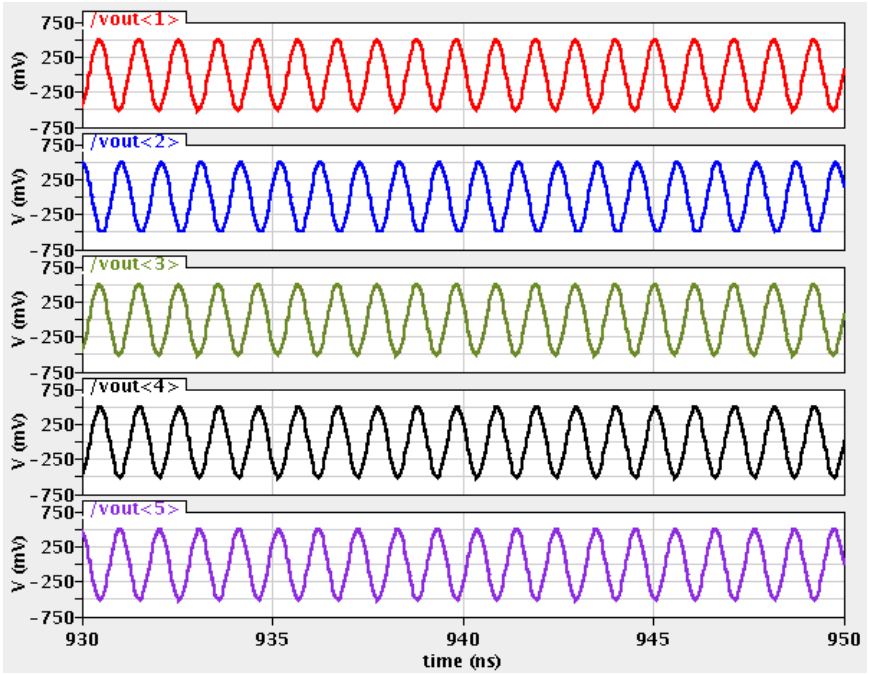

Figure 8.8. Synchronization of our proposed network with one 50% distorted pixel. Input pattern: [1 0.5 1 0 1] & Output pattern: [1 0 1 1 0].

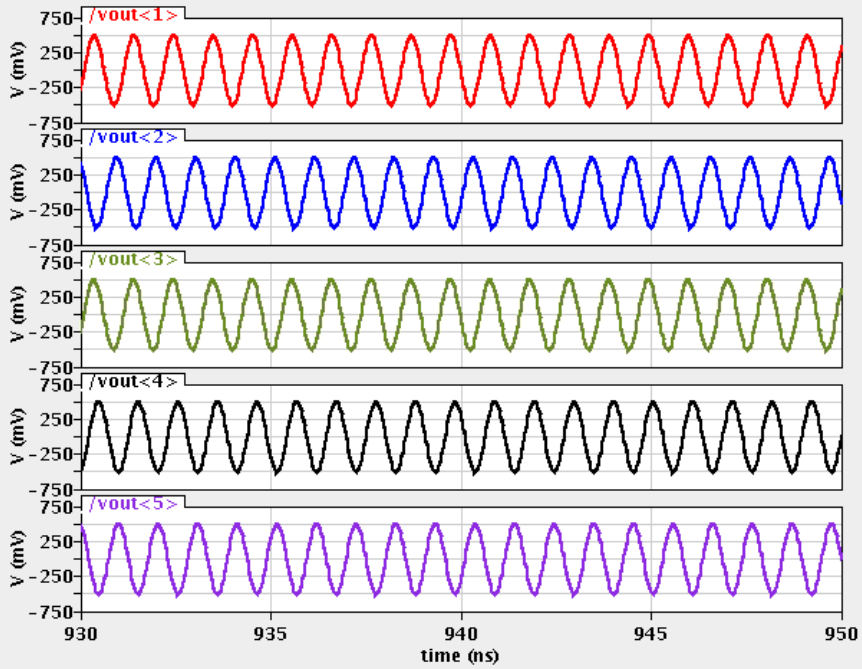

Figure 8.9. Synchronization of our proposed network with two 50% distorted pixels. Input pattern: [1 0.5 0.5 0 1] & Output pattern: [1 0 1 1 0].

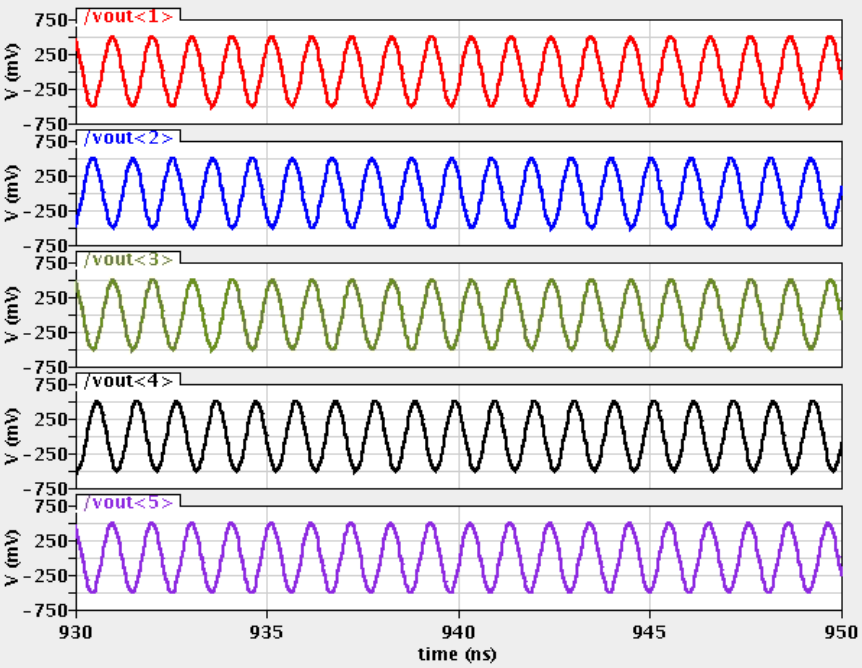

Figure 8.10. Synchronization of our proposed network with one 25% and two 50% distorted pixels. Input pattern: [1 0.5 0.5 0 0.75] & Output pattern: [1 0 1 0 1].

References

[1] C.-S. Poon and K. Zhou, "Neuromorphic silicon neurons and large-scale neural networks: challenges and opportunities," *Frontiers in Neuroscience*, vol. 5, no. 108, pp. 1-3, Sept. 2011.

[2] J.-S. Seo et al, "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," *IEEE Custom Integrated Circuits Conference*, pp. 1-4, Sept. 2011.

[3] F. C. Hoppensteadt and E. M. Izhikevich, "Pattern recognition via synchronization in phase-locked loop neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 734-738, May 2000.

[4] Y. Oike, M. Ikeda, and K. Asada, "A high-speed and low-voltage associative co-processor with exact Hamming/Manhattan-distance estimation using word-parallel and hierarchical search architecture," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 8, pp. 1383-1387, Aug. 2004.

[5] H. J. Mattausch, N. Omori, S. Fukae, T. Koide, and T. Gyoten, "Fully-parallel pattern-matching engine with dynamic adaptability to Hamming or Manhattan distance," *Symposium on VLSI Circuits Digest of Technical Papers*, pp. 252-255, June 2002.

[6] P. Kinget and M. S. J. Steyaert, "A programmable analog cellular neural network CMOS chip for high speed image processing," *IEEE Journal of Solid State Circuits*, vol. 30, no. 3, pp. 235-243, March 1995.

[7] J. M. Cruz and L. O. Chua, "A 16x16 cellular neural network universal chip: the first complete single-chip dynamic computer array with distributed memory and with gray-scale input-output," *Analog Integrated Circuits and Signal Processing*, vol. 15, no. 3, pp. 227-237, March 1998.

[8] E. Raschman, R. Zalusky, and D. Durackova, "New digital architecture of CNN for pattern recognition," *Journal of Electrical Engineering*, vol. 61, no. 4, pp. 222-228, July-Aug. 2010.

[9] S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, and H.-S. P. Wong, "An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation," *IEEE Transactions on Electron Devices*, vol. 58, no. 8, pp. 2729-2737, Aug. 2011.

[10] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, H.-S. P. Wong, "A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation," *Advanced Materials*, vol. 25, no. 12, pp. 1774-1779, March 2013.

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

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

[13] R. Jeyasingh, J. Liang, M. A. Caldwell, D. Kuzum, and H.-S. P. Wong, "Phase change memory: scaling and applications," *IEEE Custom Integrated Circuits Conference*, pp. 1-7, Sept. 2012.

[14] W. S. Zhao, G. Agnus, V. Derycke, A. Filoramo, J.-P. Bourgoin, and C. Gamrat, "Nanotube devices based crossbar architecture: toward neuromorphic computing," *Nanotechnology*, vol. 21, no. 17, pp. 175202, Apr. 2010.

[15] S. P. Adhikari, C. Yang, H. Kim, and L. O. Chua, "Memristor bridge synapse-based neural network and its learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 9, pp. 1426-1435, Sept. 2012.

[16] H. Kim, M. Pd. Sah, C. Yang, T. Roska, and L. O. Chua, "Neural synaptic weighting with a pulse-based memristor circuit," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, vol. 59, no. 1, pp. 148-158, Jan. 2012.

[17] M. Pd. Sah, C. Yang, H. Kim, and L. Chua, "A voltage mode memristor bridge synaptic circuit with memristor emulators," *Sensors*, vol. 12, no. 3, pp. 3587-3604, March 2012.

[18] M. Sharad, C. Augustine, G. Panagopoulos, and K. Roy, "Spin based neuron-synapse module for ultra low power programmable computational networks," *International Joint Conference on Neural Networks*, pp. 1-7, June 2012.

[19] M. Sharad, C. Augustine, G. Panagopoulos, and K. Roy, "Spin-based neuron model with domain-wall magnets as synapse," *IEEE Transactions on Nanotechnology*, vol. 11, no. 4, pp. 843-853, July 2012.

[20] 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," *International Workshop on Cellular Nanoscale Networks and their Applications*, pp. 1-6, Aug. 2012.

[21] T. Roska et al, "An associative memory with oscillatory CNN arrays using spin torque oscillator cells and spin-wave interactions architecture and end-to-end simulator," *International Workshop on Cellular Nanoscale Networks and their Applications*, pp. 1-3, Aug. 2012.

[22] S. Kaka et al, "Mutual phase-locking of microwave spin torque nano-oscillators," *Nature*, vol. 437, no. 7057, pp. 389-392, Sept. 2005.

[23] X. Zhu and J.-G. Zhu, "Bias-field-free microwave oscillator driven by perpendicularly polarized spin current," *IEEE Transactions on Magnetics*, vol. 42, no. 10, pp. 2670-2672, Oct. 2006.

[24] A. Khitun, M. Bao, and K. L. Wang, "Magnetic cellular neural network with spin wave bus," *International Workshop on Cellular Nanoscale Networks and their Applications*, pp. 1-5, Feb. 2010.

[25] T. Pfeil et al, "Is a 4-bit synaptic weight resolution enough? – constraints on enabling spike-timing dependent plasticity in neuromorphic hardware," *Frontiers in Neuroscience*, vol. 6, no. 90, pp. 1-19, July 2012.

[26] V. Calayir and L. Pileggi, "Fully-digital oscillatory associative memories enabled by non-volatile logic," *International Joint Conference on Neural Networks*, pp. 1-6, Aug. 2013.

[27] D. Morris, D. Bromberg, J.-G. Zhu, and L. Pileggi, "mLogic: ultra-low voltage non-volatile logic circuits using STT-MTJ devices," *ACM/EDAC/IEEE Design Automation Conference*, pp. 486-491, June 2012.

[28] D. M. Bromberg, D. H. Morris, L. Pileggi, and J.-G. Zhu, "Novel STT-MTJ device enabling all-metallic logic circuits," *IEEE Transactions on Magnetics*, vol. 48, no. 11, pp. 3215-3218, Nov. 2012.

[29] J.-G. Zhu et al, "mLogic: all spin logic device and circuits for future electronics," *IEEE International Magnetics Conference*, May 2014.

[30] S. Fukami et al, "Low-current perpendicular domain wall motion cell for scalable high-speed MRAM," *Symposium on VLSI Technology*, pp. 230-231, June 2009.

[31] V. Sokalski et al, "Naturally oxidized FeCo as a magnetic coupling layer for electrically isolated read/write paths in mLogic," *IEEE Transactions on Magnetics*, vol. 49, no. 7, pp. 4351-4354, July 2013.

[32] S. Ikeda et al, "A perpendicular-anisotropy CoFeB-MgO magnetic tunnel junction," *Nature Materials*, vol. 9, pp. 721-724, July 2010.

[33] S. Ikeda et al, "Tunnel magnetoresistance of 604% at 300K by suppression of Ta diffusion in CoFeB/MgO/CoFeB pseudo-spin-valves annealed at high temperature," *Applied Physics Letters*, vol. 93, no. 8, pp. 082508, Aug. 2008.

[34] L. O. Chua and L. Yang, "Cellular neural network: applications," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 10, pp. 1273-1290, Oct. 1988.

[35] F. C. Hoppensteadt and E. M. Izhikevich, "Synchronization of laser oscillators, associative memory, and optical neurocomputing," *Physical Review Letters*, vol. 62, no. 3, pp. 4010-4013, Sept. 2000.

[36] 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, Feb. 2001.

[37] D. Liu and A. N. Michel, “Sparsely-interconnected neural networks for associative memories with applications to cellular neural networks,” *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 41, no. 4, pp. 295-307, Apr. 1994.

[38] L. O. Chua and L. Yang, “Cellular neural networks: theory,” *IEEE Transactions on Circuits and Systems*, vol. 35, no. 10, pp. 1257-1272, Oct. 1988.

[39] M. Brucoli, L. Carnimeo, and G. Grassi, “Discrete-time cellular neural networks for associative memories with learning and forgetting capabilities,” *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, vol. 42, no. 7, pp. 396-399, July 1995.

[40] V. Calayir and L. Pileggi, “All-magnetic analog associative memory,” *IEEE International New Circuits and Systems Conference*, pp. 1-4, June 2013.

[41] A. Tazzoli, M. Rinaldi, and G. Piazza, “Ovenized high frequency oscillators based on aluminum nitride contour-mode MEMS resonators,” *IEEE International Electron Devices Meeting*, pp. 20.2.1-20.2.4, Dec. 2011.

[42] A. Tazzoli et al, “A 586 MHz microcontroller compensated MEMS oscillator based on ovenized aluminum nitride contour-mode resonators,” *IEEE International Ultrasonics Symposium*, pp. 1055-1058, Oct. 2012.

[43] M. Rinaldi, Y. Hui, C. Zuniga, A. Tazzoli, and G. Piazza, “High frequency AlN MEMS resonators with integrated nano hot plate for temperature controlled operation,” *IEEE International Frequency Control Symposium*, pp. 1-5, May 2012.

[44] V. Calayir, T. Jackson, A. Tazzoli, G. Piazza, and L. Pileggi, “Neurocomputing and associative memories based on ovenized aluminum nitride resonators,” *International Joint Conference on Neural Networks*, pp. 1-8, Aug. 2013.

[45] C. T.-C. Nguyen, "MEMS technology for timing and frequency control," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 54, no. 2, pp. 251-270, Feb. 2007.

[46] G. Piazza, "Integrated aluminum nitride piezoelectric microelectromechanical system for radio front ends," *Journal of Vacuum Science and Technology A*, vol. 27, no. 4, pp. 776-784, June 2009.

[47] N. Sinha, T. S. Jones, G. Zhijun, and G. Piazza, "Body-biased complementary logic implemented using AlN piezoelectric MEMS switches," *Journal of Microelectromechanical Systems*, vol. 21, no. 2, pp. 484-496, Apr. 2012.

[48] V. Pott et al, "Mechanical computing redux: Relays for integrated circuit applications," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2076-2094, Dec. 2010.

[49] S. C. Masmanidis et al, "Multifunctional nanomechanical systems via tunably coupled piezoelectric actuation," *Science*, vol. 317, no. 5839, pp. 780-783, Aug. 2007.

[50] J. Segovia-Fernandez, A. Tazzoli, M. Rinaldi, and G. Piazza, "Nonlinear lumped electrical model for contour mode AlN resonators," *IEEE International Ultrasonics Symposium*, pp. 1846-1849, Oct. 2011.

[51] A. Tazzoli, M. Rinaldi, and G. Piazza, "Experimental investigation of thermally induced nonlinearities in aluminum nitride contour-mode MEMS resonators," *IEEE Electron Device Letters*, vol. 33, no. 5, pp. 724-726, May 2012.

[52] J. D. Larson III, P. D. Bradley, S. Wartenberg, and R. C. Ruby, "Modified Butterworth-Van Dyke circuit for FBAR resonators and automated measurement system," *IEEE Ultrasonics Symposium*, pp. 862-868, Oct. 2000.

[53] N. Sinha et al, "Piezoelectric aluminum nitride nanoelectromechanical actuators," *Applied Physics Letters*, vol. 95, no. 5, pp. 053106, Aug. 2009.

[54] U. Zaghloul and G. Piazza, "10-25nm piezoelectric nano-actuators and NEMS switches for millivolt computational logic," *IEEE International Conference on Micro Electro Mechanical Systems*, pp. 233-236, Jan. 2013.

[55] M. C. Lemme, T. J. Echtermeyer, M. Baus, and H. Kurz, "A graphene field-effect device," *IEEE Electron Device Letters*, vol. 28, no. 4, pp. 282-284, March 2007.

[56] R. M. Westervelt, "Graphene nanoelectronics," *Science*, vol. 320, no. 5874, pp. 324-325, Apr. 2008.

[57] Y.-M. Lin, K. A. Jenkins, A. Valdes-Garcia, J. P. Small, D. B. Farmer, and P. Avouris, "Operation of graphene transistors at gigahertz frequencies," *Nano Letters*, vol. 9, no. 1, pp. 422-426, Jan. 2009.

[58] P. E. Allain and J. N. Fuchs, "Klein tunneling in graphene: optics with massless electrons," *The European Physical Journal B*, vol. 83, no. 3, pp. 301-317, Oct. 2011.

[59] J. Hedberg [Online]. Available: www.jameshedberg.com/img/samples/.

[60] B. Ozyilmaz, P. Jarillo-Herrero, D. Efetov, and P. Kim, "Electronic transport in locally gated graphene nanoconstrictions," *Applied Physics Letter*, vol. 91, no. 19, pp. 192107, Nov. 2007.

[61] A. D. B. Delbem, L. G. Correa, and L. Zhao, "Design of associative memories using cellular neural networks," *Neurocomputing*, vol. 72, no. 10-12, pp. 2180-2188, June 2009.

[62] R. Murali, K. Brenner, Y. Yang, T. Beck, and J. D. Meindl, "Resistivity of graphene nanoribbon interconnects," *IEEE Electron Device Letters*, vol. 30, no. 6, pp. 611-613, June 2009.

[63] B. Razavi, *Design of Analog CMOS Integrated Circuits*. Singapore: McGraw-Hill, 2001, ch. 9.

[64] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*. Upper Saddle River, NJ, USA: Prentice Hall, 2003, ch. 10.

[65] V. Calayir, D. E. Nikonov, S. Manipatruni, and I. A. Young, "Static and clocked spintronic circuit design and simulation with performance analysis relative to CMOS," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, vol. 61, no. 2, pp. 393-406, Feb. 2014.

[66] H. J. Schlag and W. Scherber, "New sputter process for VO_2 thin films and examination with MIS-elements and C-V-measurements," *Thin Solid Films*, vol. 366, no. 1-2, pp. 28-31, May 2000.

[67] R. Danneau et al, “Evanescent wave transport and shot noise in graphene: ballistic regime and effect of disorder,” *Journal of Low Temperature Physics*, vol. 153, no. 5-6, pp. 374-392, Dec. 2008.

[68] P. Gonon et al, “Resistance switching in HfO_2 metal-insulator-metal devices,” *Journal of Applied Physics*, vol. 107, no. 7, pp. 074507, Apr. 2010.

[69] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*. Boston, MA, USA: Pearson Education, 2005, ch. 10.