Development of Next Generation Computing Elements Fabricated with Emerging Technologies

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Shruti Patil

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

David J. Lilja, Adviser

December, 2011

Acknowledgements

Graduate school has been by far, the most educating, rewarding, and simply an amazing experience of my life. For this, I must thank many people who have helped make my years here wonderful and memorable, and because of whom I will always remember Minnesota as a very warm place.

I want to thank my adviser, for giving me this rare opportunity to work on an unconventional project that I had always wished for. Through his example, I learnt the important qualities of open-mindedness, patience, values and wisdom. I truly learnt the meaning of being an educated person. My graduate experience would have been incomplete without this training.

I also want to thank my lab-mates and colleagues, who made my work environment lively and fun. I am grateful for the excellent advice and support of my lab-mates, Chris Hescott, Sreekumar Kodakara, Peng-fei Chuang, Biplob Debnath, Nohhyun Park, Joe Myre, Weijun Xiao, Peng Li, Chengguang Ma, Kathy Grimes, and my colleagues, Hao Meng, Xiaofeng Yao, Andrew Lyle and Jon Harms.

I also want to thank my friends, Naman Saraf, Nikhil Kundargi, Salil Bapat, Neha Kulkarni, Rasik Phalak, Anup Holey, Vinit Padhye, Vineet Bhatawdekar, Pratap Tokekar, Vineeth Mekkat, Pulkit Jain, Namrata Sopory and Swati Agiwal for their support and the good times we have spent. I want to particularly thank Devdatta Kulkarni, for the long discussions about research and philosophy, which have been a major source of joy and enlightenment, which kept my research excitement strong day after day, and who kept me humble in the face of the ups and downs of the research program.

And last, but not the least, I want to thank the Minnesotan landscape, whose beauty seemed to increase with every change of season and which kept me wondering about the limitless possibilities of nature, physics and pure happiness.

Dedication

To my parents, for their unconditional love, support and guidance;
to my fiancé, for his encouragement, patience and boundless faith in me;
and to my adviser, whose constant motivation kept me going, and kept me believing.

Abstract

Revolutionary developments in the electronics industry have enabled rapid and unprecedented advances in modern systems. This has been achieved in part due to an aggressive push towards technological developments by the semiconductor industry. The electronics technology has sustained this steep trend, however, the International Technology Roadmap for Semiconductors (ITRS) that assesses future technology requirements, has identified several fundamental challenges of scalability, speed, energy and reliability that can severely limit the ability of CMOS devices to continue to maintain the sharp developmental curve. These challenges, and the discovery of new physics effects and materials, have ushered in several efforts dedicated to researching new technologies that can help support the aggressive technological roadmap. The emerging technologies bring novel capabilities for computing, however, there are large gaps in our understanding of these new technologies and how to build circuits with them, that must be filled before they can be integrated into computing systems.

This thesis focuses on evaluating the computing potential of two promising, emerging technologies: Nanoelectromechanical systems (NEMS) and Spintronics. Both technologies differ in their device physics and capabilities from electronic MOSFET devices, and pose novel challenges for integration into current computing systems. The devices from the NEMS technology are extremely power-efficient, however they have a high mechanical delay. To allow the NEMS devices to serve as effective digital switches, a novel logic design technique called 'weighted area logic' that addresses the fundamental delay challenge of the devices has been proposed. The new design technique also reduces power and area of implementation by reducing the number of devices in a circuit.

Devices from the Spintronics technology are based on magnetic effects and do not directly replace the switch-based electronic transistors. Their singular characteristics necessitate novel ideas to enable logic operations. Some of the differences of the devices from CMOS devices affect fundamental abilities that computing circuits are generally founded on. These include input-output signal compatibility, scalability of logic circuits and composability. Circuit designs and techniques that address these three challenges are proposed and studied using the spintronic devices of Magnetic Tunnel Junctions

(MTJs). A novel MTJ-based logic circuit that operates completely on spintronic principles and has spintronic input-output compatibility is designed and evaluated. An extension of this circuit into a scalable and programmable logic circuit is also proposed. The idea of combining the unique device capability of processing and storage is also presented through the design of a Spintronic Logic In Cache unit. Further, the design for an 8-function 1-bit spintronic arithmetic and logic unit has been proposed.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In his famous speech in 1959, Richard Feynmann envisaged a natural trajectory of miniaturization for computers, the elegance of manipulating atoms and information at a small scale, and designing circuits using novel effects found at the atomic scale such as quantized energy levels or quantized spins of electrons [1]. These may have been thought of as musings in 1959, however today radical transformations in the world have been made possible by the phenomenal development of the electronics technology. Advances in the semiconductor industry have facilitated the evolution of modern computers at an unprecedented scale. The continuous drive to push technology forward has resulted in great inventions in the science of computers, from hardware (cellphones, satellites, robots), to software (operating systems, programming languages, user interfaces) and applications (internet, natural language processing, AI). Several forward looking ideas came together to make all of this possible. Amongst them, one of the most significant hardware invention was that of the transistor device. The transformation of its 'switch'-like operation into complex computing algorithms, and the continual efforts to improve its basic switching ability have been the backbone of the remarkable progress that we have seen over the years.

Gordon Moore's vision of building more and more transistors on the same silicon substrate resulted in the advancement in computers and other electronic devices on many fronts – in their abilities to perform tasks, their speed, energy efficiency and size. The electronics technology in the last few decades has been based on the MOSFET transistor, an efficient electronic switch controlled by a voltage. One way to build more

devices on a chip is to diminish the size of each device. As we scale the MOSFET device into the nanoscale regimes, we approach fundamental limits and quantum-mechanical boundaries of transistor operation that result in degradation of device performance. New challenges such as sub-threshold leakage currents, variability and non-proportional interconnect scaling, have started to pose serious threats to the energy efficiency and speed of electronic chips when continued to be scaled on the same trajectory as before. Thus, the same techniques that pushed the technology to its current state of the art may be insufficient to provide the aggressive revolution that has marked the semiconductor industry for the past sixty years.

This presents an exciting opportunity for research and development into new ways of building our future computers. Poised to harbor in the era of exascale computing and beyond, in parallel to solving the existing MOSFET challenges, researchers are also exploring novel technologies to provide the necessary advantages. These explorations have been spurred by recent, novel discoveries in materials and device physics, and interesting ideas for employing them for computing. Our current expertise with the transistor technology has provided starting points for employing the new devices into computers. But further, the new technologies pose significantly different challenges that stimulate exciting new threads of research, and additional capabilities that require novel directions for computing. In this dissertation, we investigate development of computing elements with two emerging technologies - NanoelectroMechanical Systems (NEMS) and Spintronics. The two technologies are very different in nature. Switches based on the NEMS technology are similar to the MOSFET switches, albeit with near-ideal switching characteristics. This provides the potential of a low-power operation to the devices. Implemented with materials such as Carbon nanotubes (CNT), the devices have the potential for realization of low-Vdd switches, as low as 100mV, thus showing extremely promising characteristics for addressing the immediate, rising power issues in current computer chips.

On the other hand, the Spintronics technology is fundamentally different in operation from electronics technology. While the latter utilizes a single property of electrons, i.e. the *charge*, spintronics leverages two properties of electrons, the *spin* and the *charge*. In combination, charges and spins bring complementary properties to computing circuits in a low-cost, high-performance and elegant manner. For instance, the non-volatility

of spins in materials imparts an inherent memory property for computing operations, while charges naturally achieve transfer of information from one point to another.

The distinction of the NEMS and Spintronics technology from the MOSFET technology, when used for computation, can best be depicted as in Fig. 1.1[1] .

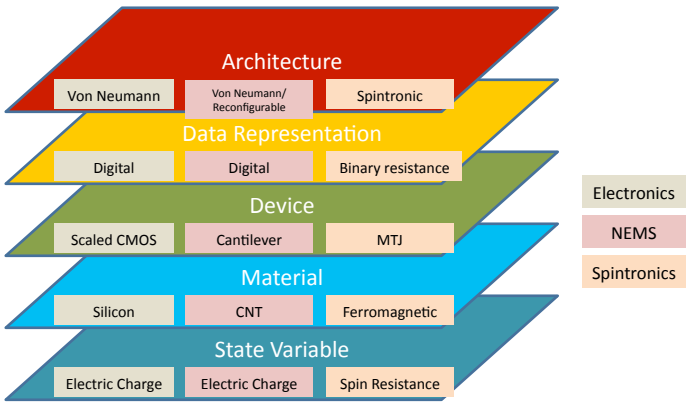


Figure 1.1: Technology differences between Electronics, NanoElectroMechanical Systems and Spintronics

Typically, a device that displays a switching behavior is candidate for logic and computation, while a device that displays a hysteresis or memory behavior is candidate for memory and storage. Both the technologies of NEMS and Spintronics display binary states that can enable digital logic, as well as a hysteresis behavior. Based on the NEMS technology, nanoelectromechanical memory (NEMM) has been developed as low-power memory technology [3], while with the Spintronics technology, magnetic non-volatile memories called Magnetic Random Access Memory (MRAM) are in production [4, 5, 6] using the device of *Magnetic Tunnel Junction* (MTJ). These memory technologies have been identified by ITRS as emerging technologies with the promise of satisfying the relentless memory goals within the next 5-10 years [2]. There is a lot of literature on the development of memories based on these emerging technologies. However, this dissertation deals with the logic potential of the two technologies.

Starting with the theoretical ability of logic computation of these technologies, the

[1] Modified from Figure ERD1 that appeared in [2]

natural questions to ask of emerging technologies are (a) Can they develop into full-fledged computer systems; (b) What are the theoretical and practical challenges they encounter in this process; (c) Can the technologies provide complete system solutions, or can they only solve problems of a specific nature; (d) If they do have the potential to be full-fledged systems, how do they perform in comparison to our current systems, or the expected performance of our current technology in the long term?

This thesis attempts to answer a number of pieces of these broad questions. In order to evaluate the potential of the technology for computing applications, the capabilities of the devices at the four abstraction levels of device, circuit, logic and architecture are explored simultaneously. The advantages and challenges presented by the novel and unique device operation are outlined at these levels of abstraction. This enabled novel logic gate circuits to be designed with these technologies along with a qualitative discussion of their advantages over existing approaches.

Narrowed down into specific problems, the specific contributions with the NEMS technology are:

- The high delay challenge of the NEMS devices is addressed at device structural level. The idea of incorporation of logic function within device structure is proposed using a logic design technique called as *'Weighted Area Logic'* [7].

The specific contributions with the Spintronics technology are:

- The device level issue of *input-output compatibility* is addressed by proposing a logic circuit with spintronic inputs and outputs [8].

- The circuit level issue of *composability* is addressed by proposing the concept of *Union with Neutralization*, and designing an MTJ-based circuit for an 8-function spintronic ALU [9].

- The logic and memory capability of MTJs are combined in a computing system by proposing the design of a *Spintronic Logic in Cache* unit [10].

- The scalability in spintronic circuits is addressed by proposing a *variable voltage bias* scheme [11].

The thesis is organized as follows: Chapter 1 introduces the motivation and the main objectives of this thesis. Chapter 2 briefly describes the NEMS technology, the NEMS device used in this thesis, and the related work published in literature. Chapter 3 presents the proposed logic design scheme of 'weighted area logic' technique. Chapter 4 briefly describes the Spintronics technology, the MTJ devices, and the related work published in literature. Chapter 5 describes spintronic logic gate circuit proposed in this thesis. Chapter 6 presents the concept of spintronic processing in memory. Chapter 7 discusses the design of a spintronic arithmetic and logic unit. Chapter 8 concludes this thesis with a final discussion of the main contributions, their impact and future work. This is one view of approaching the development of computing elements with novel devices in a holistic manner to enable rapid technology maturing, as well as evaluate the potential of technologies at the outset.

Chapter 2

The NanoElectroMechanical Systems (NEMS) Technology

2.1 Introduction

The emerging technology of Nanoelectromechanical Systems (NEMS) combines electrical and mechanical effects in devices. Its forerunner is the Microelectromechanical Systems (MEMS) technology, which is used extensively for sensors and moving parts integrated with digital components on a chip. The NEMS technology represents an advancement of the fabrication of electromechanical devices into nanoscale regimes.

Amongst the different types of NEMS devices, digital switches that are implemented using the NEMS technology show several desirable characteristics, such as zero leakage currents, high on-current capabilities and the potential for high-speed, low-Vt (threshold voltage of devices) operation. In the light of the growing power issues in digital designs that use CMOS transistors, the near-ideal OFF-state characteristics of the NEMS switches are particularly attractive. Currently, several research attempts are underway to develop the NEMS technology to enable fabrication of switches with high yield, low voltage requirements, fast operation and high reliability.

The NEMS-based switches rely on electrostatic forces of attraction for actuation (ON) and mechanical forces for restoration (OFF). This ON/OFF operation has been previously leveraged to propose varied device structures for digital switches[12, 13, 14,

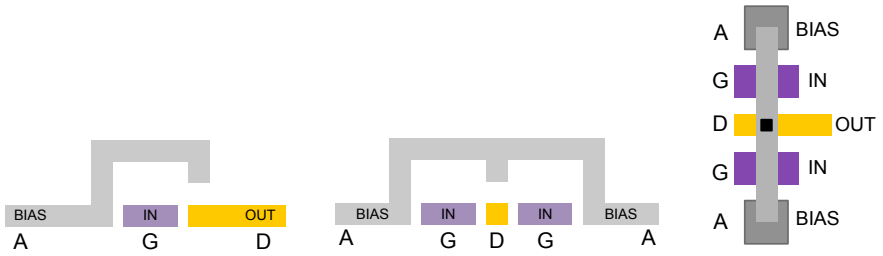
15]. However, the unique electromechanical operation of the device need not be restricted to a simple switch, but instead can be used to derive logic functions directly. It further allows for novel logic design strategies for NEMS devices.This idea is explored in this work.

The fundamental NEMS device used here is a cantilever based mechanical switch fabricated using carbon nanotubes (CNT). Carbon nanotubes provide the important material properties of high Young's modulus, low mass density, high mobility and low resistance properties for the cantilever design of the devices. The material properties contribute to the reliability, conductivity, speed and lifetime of the devices, while the structure of the device contributes to the ideal switching characteristics. To address the challenges associated with the device, the ideas proposed in this thesis merge the capabilities of the devices at the physical, circuit and gate level. By merging these levels of abstraction, the physical structure of the device is able to act as a logic gate directly. Thus, higher level of customization is possible when fabricating a structure for a certain logic function. This optimization eventually increases the performance benefits of using NEMS devices. Above these abstractions, the the architectures of the systems have been viewed as independent of the underlying technology, and similar to those developed for conventional CMOS devices.

2.2 Device Operation

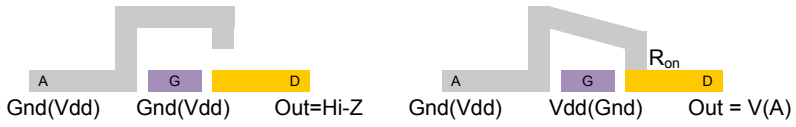
A cantilever-based nano-electro-mechanical (NEMS) switch is actuated by a combination of electrical and mechanical effects in the device. The NEMS-CNT switch that we use in this paper is a 3-terminal structure, with anchor (A), gate (G) and drain (D) terminals. A cantilever beam fabricated with a single wall carbon nanotube (SWCNT) thin film extends from the anchor to the drain terminal, forming a CNT-based conducting channel in the switch. Fig. 2.1(a) shows the device structure, where the cantilever beam stands freely above the drain. In order to operate the device, a gate voltage is applied to create a voltage difference between the gate and the anchor. This induces opposite charges on the gate and the cantilever, creating an electrostatic force of attraction between them. If the difference in the voltages on the terminals exceeds a certain threshold value called as pull-in voltage V_{pi}, the electrostatic forces exert a sufficient pull-in

force on the cantilever that forces it to make contact with the drain. This connects the anchor and drain terminals, thus closing the 'switch'. When the electrostatic force of attraction disappears, the pull-in force disappears, however, in the closed position, there exists a spring restoring force in the cantilever due to its mechanical properties. Therefore the cantilever springs back to its freely standing initial position, resulting in an 'open' switch. To avoid shorting the source to the gate, a thin insulator can be placed on top of the gate electrodes. This is achieved by using a high permittivity material such as HfO_2, to minimize any reduction in the electrostatic force [16].



(a) Freestanding Cantilever (b) Fixed-Fixed beam (c) Top View of Fixed-Fixed beam

Figure 2.1: NEMS-CNT Device Structures



(a) Open Switch (b) Closed Switch

Figure 2.2: Voltage biases on terminals for different logic states

The simple cantilever shown in Fig. 2.1(a) is very dense, but is quite sensitive to the effects of stress in the film, which can lead to significant changes in the zero-bias gap between the gate and the electrode. A more stable structure is shown in Fig. 2.1(b), where the cantilever beam is supported on both sides by the anchor terminals. This

structure, as seen in the top view is shown in Fig. 2.1(c). The behavior of the device is fundamentally similar to the first structure, but is less sensitive to process variables. Furthermore, it provides important new design capabilities by incorporating multiple gate electrodes into a single device. The biasing of the anchor and gate terminals in the open and closed states is shown in Fig. 2.2. Voltage Vdd is chosen to be $1-2$ times that of V_{PI} according to the required noise margin levels. NEMS-CNT switches that operate on this principle have been demonstrated in ([13, 17, 18]). With suitable materials, their potential to operate at pull-in voltages of 100mV has also been investigated [19], making them promising devices for use in low-power digital designs.

Previously, two terminal NEMS switches operating at about 600ps were demonstrated experimentally in [18, 20]. The promising characteristics that CNTs showed, especially, the high-speed measurements, strongly motivated the development of three-terminal devices that can be used as a controllable switch in digital designs. The 3-terminal devices are fabricated using a layer-by-layer (LbL) self-assembly process [16]. The fabrication challenges for these devices are attempting to address the device challenges of yield, scalability and low-voltage (1 V) operation.

The electromechanical operation of the device allows an important property of programmability. With the anchor biased at ground, the device acts as an NMOS device. It actuates only when the gate is biased to Vdd, and is OFF when the gate is biased to ground. On the other hand, when the anchor is biased at Vdd, the device acts as a PMOS device. This ability simplifies the fabrication process, keeps capacitances low and produces smaller layouts in computing circuits.

2.3 Related Work

A number of materials have been studied for NEMS devices such as tungsten, polysilicon [12], ultrananocrystalline diamond [21] and carbon nanotubes (CNT) [13, 17]. Additionally, a number of alternative structures have been proposed for NEMS-based switches [13, 12, 14]. These range from two-terminal [18, 14], three-terminal [13, 22, 16], four-terminal [12, 23, 14, 15] to seven-terminal devices. By pursuing various directions of research, the different structures strive to improve various aspects of the NEMS-switch technology, such as yield, fabrication processes and the performance (area, power,

delay) of the digital applications that are designed using these switches. The devices discussed in this thesis are unique in the material they use i.e. carbon nanotubes (CNTs). CNT films have promising material properties compared to the other materials, such as low mass density, high yield strength and high Young's modulus [17]. They have the potential to operate in Gigahertz ranges [24, 18]. These properties make them attractive devices for use in digital applications.

Various applications have been proposed for utilizing the low power and ideal OFF characteristics of NEMS devices. Dadgour et al. [25] have proposed using NEMS devices in hybrid NEMS-CMOS circuits, by replacing some CMOS transistors in conventional designs by NEMS devices. This reduces the leakage power in circuits by taking advantage of the ideal OFF state of the devices. Similar power savings were observed by replacing CMOS sleep transistors in digital circuits, and data storage inverters in SRAM cells. Chen et al.[12] have also proposed the use of NEMS switches for I/O applications by designing energy efficient Digital-to-Analog (DAC) and Analog-to-Digital (ADC) circuits.

Though the device-level area and delay of the NEMS devices are generally expected to be larger than that of a single MOSFET, the overall area and delay of complete circuits can be considerably lower than complete CMOS circuits. In their work, Dadgour et al. [15] have proposed using four-terminal laterally actuated NEMS devices that implement the XOR function directly. By using these devices to simulate adder circuits, large savings in area are expected. Chen et al. [12] used four terminal NEMS relays for designing a full adder combinational circuit that has a single unit of mechanical delay. These prior works demonstrated a number of properties of NEMS devices: integration with CMOS devices, reduction in circuit area and delay, and power advantages in conventional or custom circuit designs with NEMS devices. The literature so far supports the promise of NEMS devices in low-power computing circuits.

In this thesis, a general approach to designing custom NEMS circuits is proposed in order to further reduce the delay, area and power of the devices at the circuit-level. This reduction is obtained by customizing the device structure for the logic function, an optimization made possible by the properties of the NEMS technology. Due to this optimization, the delay disadvantage of an individual device does not propagate proportionally to the composed function, instead is much reduced at the functional unit

level. However, a greater control over the fabrication process is required to customize the geometry. The variability that exists in the fabrication process puts a limit on the structural control, and therefore, a limit on which functions can be accomplished using this method.

Chapter 3

Computing with NEMS

The NEMS switches are characterized by four strong properties that make them promising devices for use in low-power digital applications: (1) Controllable drain-source switching operation using three (or more) terminal devices; (2) Zero leakage current; (3) Potential for extremely low-Vt operation due to a sub-threshold slope much lower than that of MOSFET devices; and (4) Low device capacitances making high fan-outs possible.

The reduction in the dynamic power (due to low-Vt) and zero static power dissipation (due to zero leakage current) makes them low-power digital switches. The principal challenges, however, are their high mechanical delay and theoretical scalability limits. Even with the long-term issue of scalability, the low-power operation of NEMS switches provides a strong motivation for use in electronic chips to alleviate current power challenges in digital designs. Thus, the mechanical delay is the main challenge in utilizing NEMS switches in immediate electronic chips. To deal with this issue, this chapter proposes a logic design technique called as 'weighted area logic (WAL)'[7].

3.1 NEMS Logic Gates

With any new device that is candidate for digital designs, the feasibility of performing logic operations with the device must be established. To that end, logic circuits that can perform the universal logic operations of NAND, NOR or the set of functions, (AND, OR, NOT) must be realized. In order to design logic functions with the NEMS

switches, two techniques can be used. One is the standard complementary MOSFET design technique used for MOSFET devices, and the other is the proposed weighted area logic design technique.

3.1.1 Standard Design Technique with NEMS switches

3.1.2 Equivalence to a transistor

Digital designs using the MOSFET technology have traditionally utilized the ON/OFF switch-like behavior of the transistors to accomplish logic functions. In fact, the behavior of a NEMS-CNT device resembles that of a digital CMOS switch in that the voltage between the anchor and the gate acts as the controlling voltage, and determines the ON or OFF state, similar to a transistor operation. Therefore, an equivalent NEMS-CNT-based logic design can be derived by replacing the PMOS and NMOS devices in a CMOS-based design with appropriately biased NEMS-CNT devices. This design style will be referred to as the *standard design technique* for NEMS. In order to demonstrate that the standard design technique applies to real NEMS devices, a NAND logic gate was fabricated using a NEMS-CNT device with a fixed-fixed beam structure. The fabrication process, device details and characterization curves are presented in [16]. This experimentally verified two aspects of designing with NEMS-CNT switches: (1) the behavioral similarity between MOSFET switches and NEMS-CNT devices, and (2) the feasibility of deriving logic circuits by a simple replacement of CMOS devices by NEMS-CNT switches.

3.1.3 Model for circuit simulation

To assist in circuit simulation, the behavior of the device can be abstracted into a parameterized NEMS device model. Functionally, the NEMS cantilever device is a 3-terminal switch with a finite on-resistance R_{on}. Its switching mechanism has a mechanical delay T_{mech}. Switching occurs at pull-in voltage of V_{PI}. For any gate voltage less than V_{PI}, the device current is zero. While the gate-cantilever capacitance C_{GC} can be modeled as a dynamic capacitance that describes the electromechanical operation, we note that the internal capacitances of the device are extremely low. For the gate, this is due to the large gap, typically about 20nm, compared to a 1nm equivalent oxide thickness of a

fully scaled gate oxide. For the source and drain, the low capacitance is because there are no junctions in the device. It is built in a low dielectric contact material (SiO_2), instead of in a semiconductor. Thus, the capacitance of the device as experienced by a circuit stems mainly from the self capacitances of the wire associated with the three terminals of the anchor, gate and the drain, i.e. C_a, C_g and C_d respectively. Another significant behavior exhibited by the NEMS-CNT devices is their asymmetry in rise and fall times [17]. The rise time due to the device switching ON is about 80-90% higher than the fall time measured when the device switches OFF.

Based on this operation, a Verilog-A model was developed for the NEMS devices for circuit-level simulation of complex logic units that can allow a comparison between different designs. It models the NEMS-CNT device as a 3-terminal switch with five parameters R_{on}, T_{mech} $(= T_{ON})$, V_T, C_{node} and Asymmetry Ratio (T_{OFF}/T_{ON}). This simple behavioral model allows us to fit measured device values in the model, while giving the flexibility to observe the sensitivity of the device for different parameters along with preliminary performance evaluation and projection for different applications.

3.2 Cantilever-based Nanomechanical Logic Designs

While one feasible technique of designing with the NEMS-CNT devices is by using the classic complementary standard design technique, the devices also possess a number of unique properties. For example, the behaviors of the PMOS and NMOS devices in a CMOS implementation can be obtained by applying appropriate biases on the same device structure. This simplifies the fabrication process significantly and reduces the area. Secondly, the pull-in voltage of an individual device can be easily varied by varying the structure of a device. This in turn, enables the devices to be manufactured as more complex structures. On the flip side, the devices suffer from a large mechanical delay relative to an electrical delay. Therefore, it is desired to reduce the total delay of a logic circuit. Taking into account these strengths and challenges of the devices, we propose an electromechanical design strategy called as 'weighted area logic design' that relies on the electromechanical operation of the devices to implement NEMS-CNT based logic structures. By incorporating both electrical and mechanical effects to produce a logical result directly, the number of devices in the maximum combinational path are reduced,

resulting in faster logic computation. Also, by integrating the logic functionality into the device structure, the logic circuits that are produced result in smaller structures than a corresponding standard implementation.

3.2.1 Weighted Area Logic (WAL) Design Concept

The amount of electrostatic force formed between the cantilever (c) and the gate (g) depends upon many factors. It is given by:

$$F_{cg} = \frac{-\epsilon A V_{cg}^2}{2g^2} + \kappa(g_0 - g)$$

where V_{cg} is the voltage bias on the two terminals, A is the area of the cantilever and the gate resulting in the pull-in forces, g is the gap between the cantilever and the gate, g_0 is the gap at zero volts and zero spring extension, ϵ is the permittivity of the material between the gap, κ is the spring constant, and the negative sign indicates an attractive force [26]. The weighted area logic design strategy exploits the second factor, i.e. the area A. Clearly, the pull-in force experienced by the cantilever is proportional to the overlap area between the cantilever and the gate. By changing the cantilever-gate overlap area, the magnitude of effect of the gate on the cantilever can be controlled. The larger the overlap area, the higher is the effect, or the 'weight', of the gate. With this ability, instead of relying on switch-based electronic circuit to compute a logical output, the electromechanical properties of the devices can be leveraged to obtain the logical result directly.

When a larger impact of an input is required on the output, a device with a larger gate-cantilever overlap area is designed. For example, in the expression $Y = A + B.C$, the input A affects the logic-high output individually, while inputs B and C affect the logic-high result in combination. Thus, the contribution of input A in the function is larger, and necessitates a larger impact on the cantilever by input A alone as compared to that of inputs B and C alone. The contribution of an input to the logic-high and logic-low states in a truth table is evident when the logic design is expressed in the form of a sum-of-products (SOP) (or product-of-sum (POS)) expression. These expressions describe the desired output as a logical function of the inputs. For instance, when inputs are OR-ed together, all inputs affect the output equally and independently. On

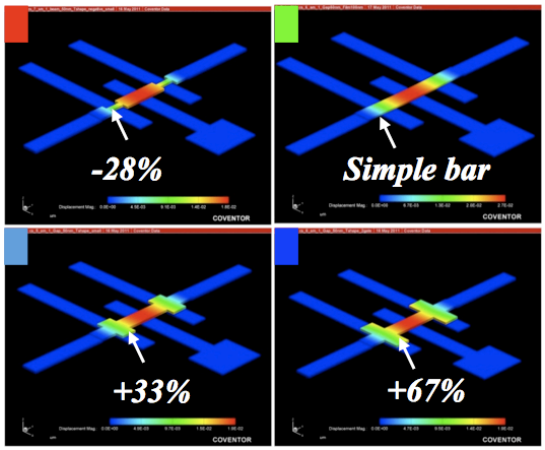
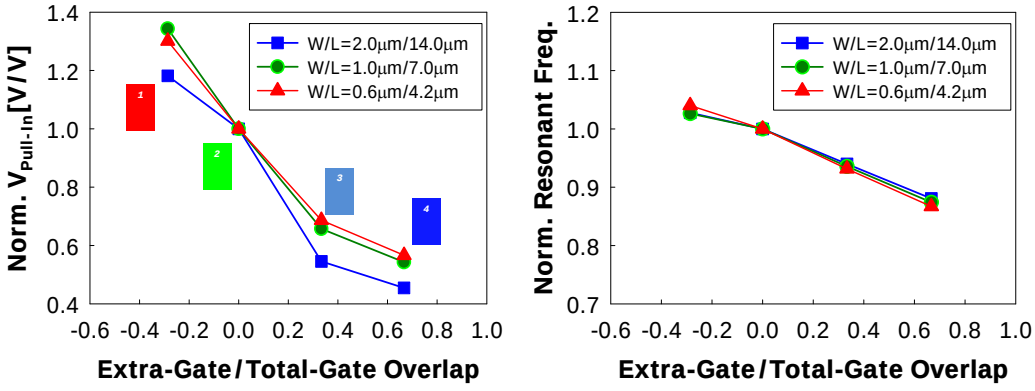


Figure 3.1: Weighted gate overlap structures simulated

the other hand, when inputs are AND-ed together, this indicates that the inputs affect the output in combination.

This observation can be converted into the weighted area logic design approach. An input that affects the output individually is designed so that it exerts sufficient mechanical pull-in on a cantilever by itself. On the other hand, inputs that affect the output in combination are designed and placed so that they are able to pull-in a cantilever only in combination. To increase the pull of a gate on the cantilever, its area of overlap with the cantilever is increased, while to decrease its pull on the cantilever, its area of overlap is reduced. In terms of weights, the inputs that are OR-ed are alloted equal weightage, each of which is the maximum weightage for the determination of the output. The inputs that are AND-ed are also given equal weightage, however the sum of the weights is the maximum weightage. Thus, a cantilever–based design can be directly derived from an SOP or a POS expression. Since the logical operation is partly incorporated into the physical structure, the resultant design occupies a lesser chip area than a design with devices that act as 3-terminal switches equivalent to a standard MOSFET operation.



(a) Normalized Pull-in Voltage (b) Normalized Resonant Frequency

Figure 3.2: Effects of weighted gate overlap on device characteristics

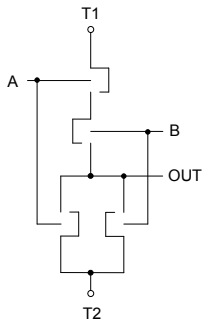
3.2.2 Device-level simulation of the WAL Concept

To first validate the weighted area theory, a MEMS device simulator called *Coventorware* was used [ref-Min-Woo's thesis]. The simulator enables a structural verification of the device layout and allows us to see the effects of the weighted gate overlap structures. Fig. 3.1 shows the four structures that were studied. With a normal structure as a baseline, the gate-cantilever overlap area was varied as -28%,$+33\%$ and $+67\%$ to observe the changes in pull-in voltages for a smaller area as well as larger areas. For the simulation, three differently sized devices were considered with W/L= 2um/14um, 1um/7um and 0.6um/4.2um respectively. Fig. 3.2 shows the change in pull-in voltage and resonant frequencies observed due to the change in the cantilever-gate overlap area. The results show that within +/-30% of overlapped gate area, the pull-in voltage changed within a range of +/-30%. However, its fundamental resonant frequencies that affect its switching delay did not change as much. This helps to strongly conclude that the weighted gate overlap structures could change their pull-in voltages effectively without much loss of switching speed.

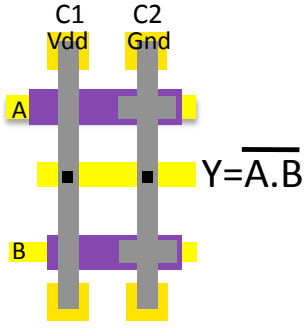
3.3 Weighted Area Logic (WAL) Design of Digital Elements

3.3.1 NAND/NOR Logic Gates

Using the weighted area concept, basic two-input logic functions of NAND and NOR can be directly implemented as electromechanical designs. Fig. 3.3(a) shows the NAND design employing NEMS-CNT devices using the standard MOSFET design technique. It is derived from CMOS-based implementation of a NAND gate using only the switch-like behavior of the devices. This design is expected to take advantage of the low static power dissipation due to near-ideal OFF characteristics of the devices, as well as the simplicity of fabrication due to same type of devices performing the pull-up and pull-down function (as opposed to p-type and n-type required in MOSFET designs). However, the maximum mechanical delay of the circuit for the worst-case input transition is two units of the mechanical delay of the cantilever beam.



(a) NAND/NOR circuit with Standard design technique (4 cantilevers). Each NEMS device corresponds to a MOSFET device in a CMOS-based NAND circuit.



(b) NAND/NOR circuit with NEMS-WAL design (2 cantilevers) - Top view. Cantilevers C1 and C2 have a weighted overlap area with gates A and B.

Figure 3.3: NEMS-NAND designs

By employing the weighted logic design methodology, a smaller design with a delay as small as a single unit of mechanical delay can be obtained. Fig. 3.3(b) shows a custom design based on the weighted logic design concept that electro-mechanical forces can combine logically to enable switching. The two cantilevers C_1 and C_2 operate in parallel biased by power and ground, and are controlled by common input lines A and B. With suitable overlap areas, forces from two gates can either pull in a cantilever individually or in combination. Cantilever C_1 experiences pull-in forces from the inputs and makes contact with the output when both inputs are at Vdd. Cantilever C_2 is constructed so that the overlap area between the cantilever and each input is much larger than the overlap area of cantilever C_1 with each input. Its overlap area with each gate is actually designed to ensure that each input is individually capable of exerting a pull-in force that is sufficient to enable the cantilever to make contact with the output. To increase the gate-cantilever overlap area, the cantilever C_2 can be designed as a 'winged' structure as shown in Fig. 3.3(b). However, even with a simply designed cantilever, the NAND function can be obtained by separating the individual gates into individual cantilevers, as shown in Fig. 3.9(a). With the cantilevers biased at Vdd and Ground respectively, the NAND logic function is obtained. On the other hand when the cantilevers are biased at Ground and Vdd respectively, the NOR logic operation is obtained.

3.3.2 Simulation of NAND-WAL circuit

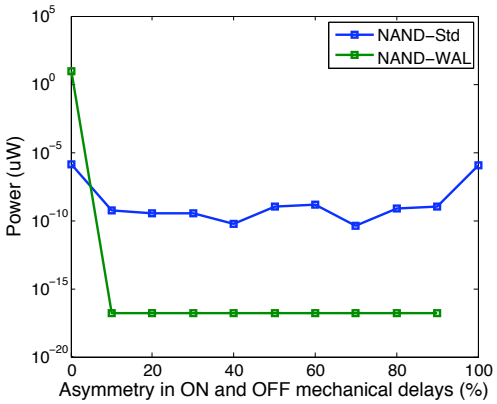
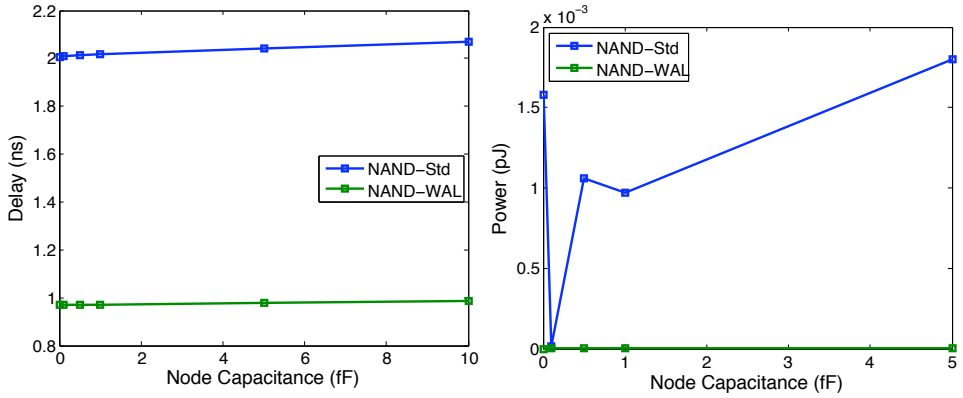


Figure 3.4: Power Sensitivity to Asymmetry in ON and OFF delay

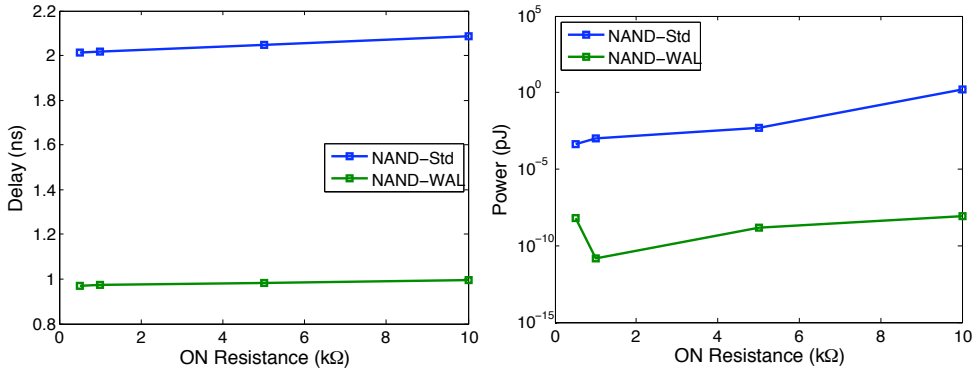
In order to verify the design, we simulated the NEMS-WAL circuit using the device model discussed in Section 3.1.3 with SPICE simulation tool. We extracted the sensitivity of the NAND-WAL design to varying values of the device parameters. Fig. 3.4 shows the power consumption of the devices with respect to the ON-OFF time asymmetry. With respect to ON-OFF time asymmetry, the NEMS-CNT devices consume lesser power in the WAL design, since the power dissipation in the short circuit mode when the devices are switching at once (P_{SC}) is reduced to almost zero. With respect to varying ON resistance and node capacitance, the WAL design did not show a lot of variation, however it always consumes lesser power and delay than the standard implementation.



(a) Delay (b) Power

Figure 3.5: Delay and Power of NAND gate wrt varying node capacitance

We also compared the NAND-WAL design with the standard NAND (NAND-Std) design, using the following device parameters: $C_{node} = 1fF$, $T_{mech} = 1ns$, $V_{PI} = 1V$, $R_{on} = 1k\Omega$ and $T_{off} = 0.5T_{on}$. These values were chosen as optimal or average parameters from the sensitivity curves. The load capacitance at the output nodes was assumed to be $1fF$. Fig. 3.7(a) shows the delay comparisons for the two designs, for six possible transitions in the NAND truth table. For all possible transitions in the NAND truth table, the WAL design reduces the maximum mechanical delay of two units to a single unit, demonstrating the speed advantage of incorporating logic capability within the device structure.



(a) Delay (b) Power

Figure 3.6: Delay and Power of NAND gate wrt varying ON Resistance

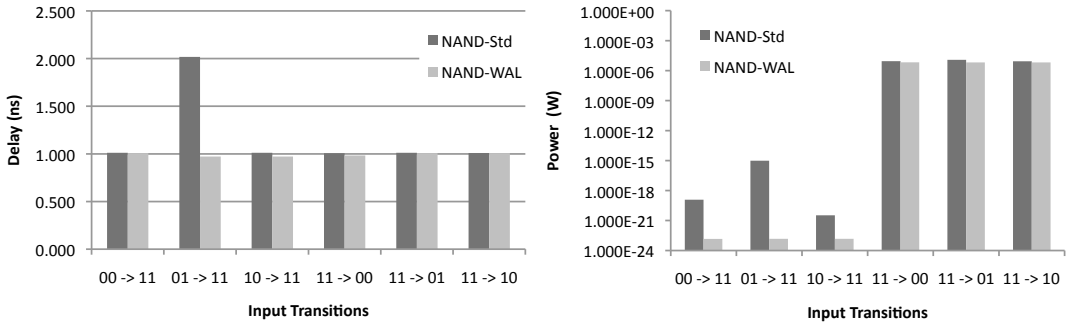
3.3.3 XOR gate

The XOR function is a key function that accomplishes the SUM operation in a half-adder. Several interesting device structures have been proposed previously to implement the XOR function effectively in order to reduce the area of an adder, for example, the 4-terminal device structure in [12] and the dual-gate structure in [15]. These structures were designed specially for the XOR operation, and therefore would prove more effective. Under the weighted area logic scheme, the XOR gate (Fig. 3.9(c)) can be designed as four cantilevers operating in parallel and enabled by different gate inputs. This demonstrates a regular pattern in which a NEMS-CNT circuit can be designed and fabricated. It reduces the maximum delay of the function to two mechanical units.

3.3.4 Three input functions Y=A+B.C

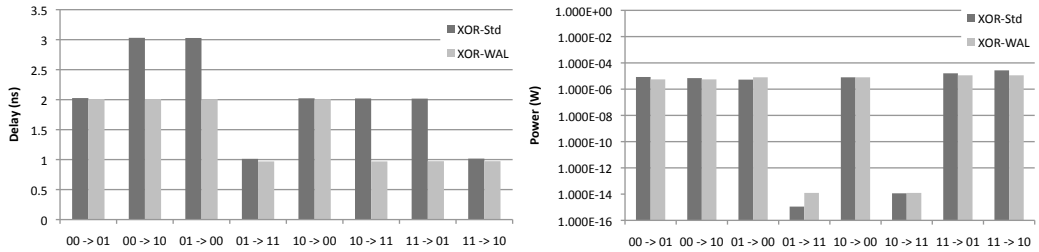
In theory, the weighted area logic design can extend to any number of inputs. However, the practical limitations of an integrated weighted logic design are determined by the number of gates that can be fabricated for a single cantilever. In practice, an OR expression of the form $Y = A + B + C + D + ...$ can be accomplished with multiple cantilevers in parallel, as in the alternate NAND design (Fig. 3.9(a)).

The AND expressions of the form $Y = A.B.C.D...$ are trickier. The logic function



(a) Delay (b) Power

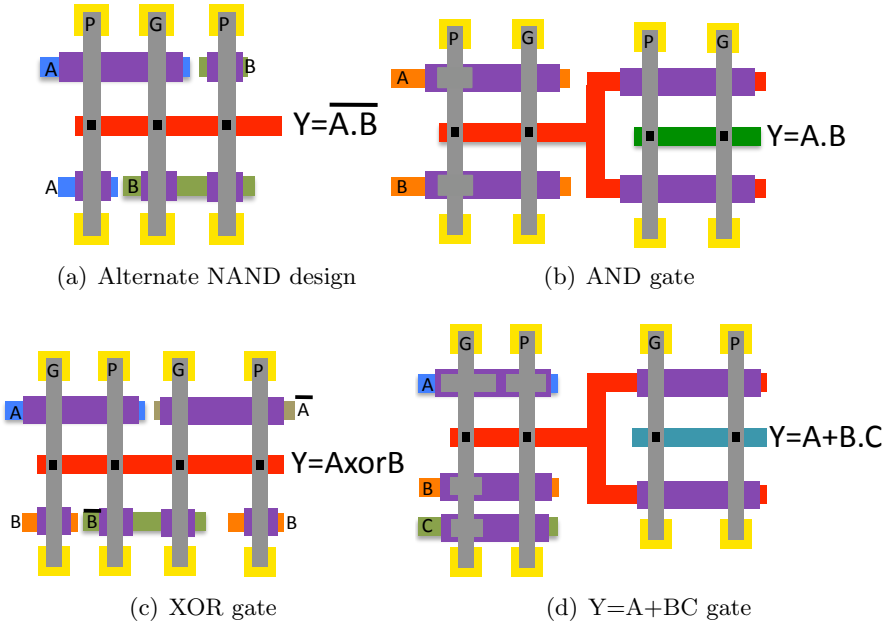
Figure 3.7: NAND-WAL effect on delay and power



(a) Delay (b) Power

Figure 3.8: XOR-WAL effect on delay and power

$Y = A + B.C$ offers good insights into implementing AND expressions. Fig. 3.9(d) shows a design that employs three gates driving a cantilever together with different weights. The truth table of the function $\overline{Y} = \overline{A + B.C}$ can be split into two parts, one part where maxterms M_0 to M_2 have the output 1 ($\overline{Y_1}$), and the other part where maxterms M_3 through M_7 have an output 0 ($\overline{Y_0}$). To implement $\overline{Y_0}$, two conditions must be satisfied: (1) the input A must be capable of closing the switch individually; and (2) inputs B and C must be capable of pulling in the cantilever together. Thus, as shown in Fig. 3.9(d), the overlap area of A with the cantilever biased at GND is designed to be large enough to establish the appropriate electrostatic attraction, while the overlap area is large and equally divided between the inputs B and C. In the case



(a) Alternate NAND design (b) AND gate

(c) XOR gate (d) Y=A+BC gate

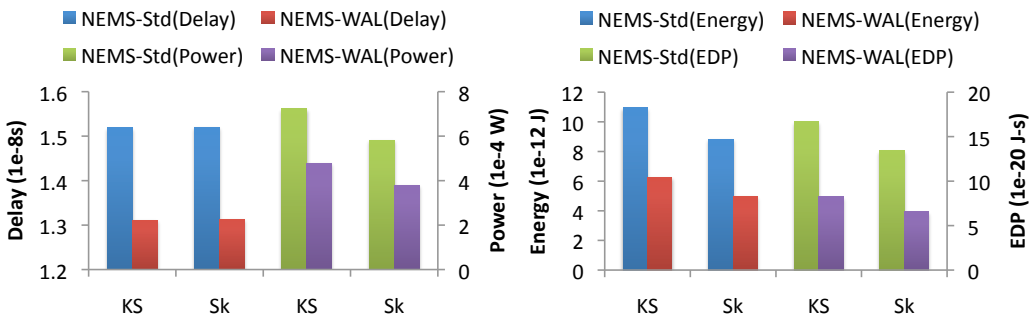
Figure 3.9: NEMS-WAL designs of adder components

of output of 1, the input A must not pull-in the cantilever by itself, but incorporate the force of B and/or C also. Therefore, the overlap area must be adjusted accordingly. If the overlap areas are specified in terms of weights, where a weight of 1.0 indicates that the input is capable of operating the cantilever by itself, then the weighted logic design of the function $\overline{Y} = \overline{A + B.C}$ can be represented as $\overline{Y_0} = \{A(1.0), B(0.5), C(0.5)\}$ for output of 0, and $\overline{Y_1} = \{A(0.75), B(0.25), C(0.25)\}$ for output of 1. The output of this design is then applied to a NOT gate to obtain the final output Y.

As the number of inputs in the AND term increases, the cantilever must be elongated for all inputs to be placed under the same cantilever. Design rules for a practical gate has limits of a minimum width, a minimum distance between two gates, and length of cantilever. These limits determine the number of inputs that can be implemented as an AND term. As fabrication challenges are addressed, it may become possible to have more than two gate inputs that actuate a single cantilever beam, thus forming more compact designs.

3.4 Putting it together - 32-bit Adder Design

The advantages derived from using a custom NEMS circuit, when compared to a non-custom circuit, increase as the logic function becomes complex. The three circuits discussed above form the basic circuits in an n-bit adder circuit. By composing them into a 32-bit adder architecture, we can now quantify the speed-up obtained by using the weighted area logic scheme over the standard technique of design. Fig. 3.9 shows the designs for the base circuits of the adder. The AND gate (Fig. 3.9(b)) is designed as the previously described NAND gate followed by an inverter. The mechanical delay of this gate is two units – one unit due to the NAND gate and the second due to the NOT gate.



(a) Delay, Power

(b) Energy, EDP

Figure 3.10: Comparing Std and WAL designs for 32-bit adders, KS: Kogge-Stone, Sk: Sklansky adder architecture

3.4.1 Speed-up and Energy

Amongst the various adder topologies proposed in the past, the Kogge-Stone adder has the lowest delay, and the Sklansky adder has the lowest energy-delay product [27]. By bringing logic functionality into the device structure, the WAL design strategy reduces the delay of a design, as well as power due to the reduction in number of devices. Fig. 3.10 shows the performance graphs. With the same simulation parameters as used for the NAND gate comparisons, the adder results show a 13% reduction in delay, 33%

reduction in power, 42% in energy and about 50% reduction in the energy-delay product. Thus, the delay and power advantages combine to result in a significant energy-delay efficiency.

3.4.2 Area Reduction

The weighted area logic designs reduce the number of cantilevers required to implement the logic function. For example, in the 32-bit Kogge-Stone adder, the WAL design uses 40% fewer cantilevers than the standard design. This directly reduces the area of the designs.

The NEMS-CNT based designs also have the potential to reduce the area of logic functions compared to an equivalent CMOS implementation due to a number of reasons. The elimination of the requirements of different types of wells (P-type and N-type) is the primary reason for the area reduction. Secondly, the layout of multiple cantilevers in parallel can form regular grids, leading to an effective use of the chip area. Finally, the programmability of the circuits as NAND and NOR circuits can be expected to add to this reduction.

3.5 Going forward

The weighted area logic design technique demonstrates that there can be effective structural solutions to handle the mechanical delay challenge of the NEMS technology. The technique is designed for a structure in which the gate and cantilever attract each other when a voltage threshold is exceeded. Since this is a general technique, it may become applicable to any of the NEMS based structures that have been proposed till now, and can be expected to reduce the area, delay and power of circuits using those other structures. Combining the weighted area logic capability with the strong low-power advantages, the devices hold potential to provide the improved power-efficiency that is vital in future systems.

Chapter 4

The Spintronics Technology

4.1 Introduction

Traditional computation has been achieved using *charge*, a property of electrons that has single-handedly carried technology to unimaginable boundaries. Until now, computing has little used the other property, the *spin*, or the angular momentum, present in electrons. The field of spin electronics, or spintronics, strives to exploit spins along with charges to develop a novel technology with more characteristics than simple electronics. Several years after the rise and fall of ferrite core memories, spintronics received a renewed interest with the discovery and understanding of the phenomenon of *Giant Magneto Resistance* (GMR). The revolutionary development of read heads, that made it possible to realize highly dense memories, motivated a whole new direction of research for spin-based applications.

Many efforts led by researchers in spintronics have revolved around finding answers to questions about spins in materials, such as (1) how can one create a spin-polarized system? (2) how can one detect the existence of spins? (3) how do spins propagate within and across material interfaces? (4) how long does a particle retain its spin orientation? By understanding the interaction between the spin of a particle and its solid-state environments, scientists hope to making useful devices based on spintronics physics. In the recent years, researchers have succeeded in developing many different types of spintronic devices. One such device is the magnetic tunnel junction (MTJ). It operates on the principle of tunneling magnetoresistance (TMR), an effect seen at

and below micro-scale dimensions. This makes them highly scalable, and hence high integration densities are possible with MTJs [4]. Built with ferromagnetic materials, they are also capable of retaining their states in the absence of power. The MTJ also exhibits a number of other properties such as noise-resistance, radiation hardness and programmability.

From a computer designer perspective, such a device is capable of solving a number of issues in current systems, one being a candidate for furthering device scaling to keep the Moore's law true. The property of non-volatility of MTJs has already been leveraged in developing cheaper, denser and low-power memory, called the Magnetic Random Access Memory (MRAM)[4, 5, 6].

Magnetic tunnel junctions are expected to be widely used in future applications to avoid data volatility in internal memory. But with their unique combination of abilities, it is believed that they may have more to offer than just memory applications. Early work that demonstrated basic logic functions with MTJ devices, has prompted exploration of MTJs as devices that in addition to memory, can complement the silicon technology in logic and computation. Since great achievements have been possible using charge alone, it appears that spintronics, which manipulates spins and charges together, should have the potential to provide more elegant solutions to computing problems. However the spintronics theory does not function in a switch-like manner as electronics does. The 'electronic' abstraction paradigm that has been applied successfully for computing and research at various stages do not translate to spintronics, and may require to be defined differently.

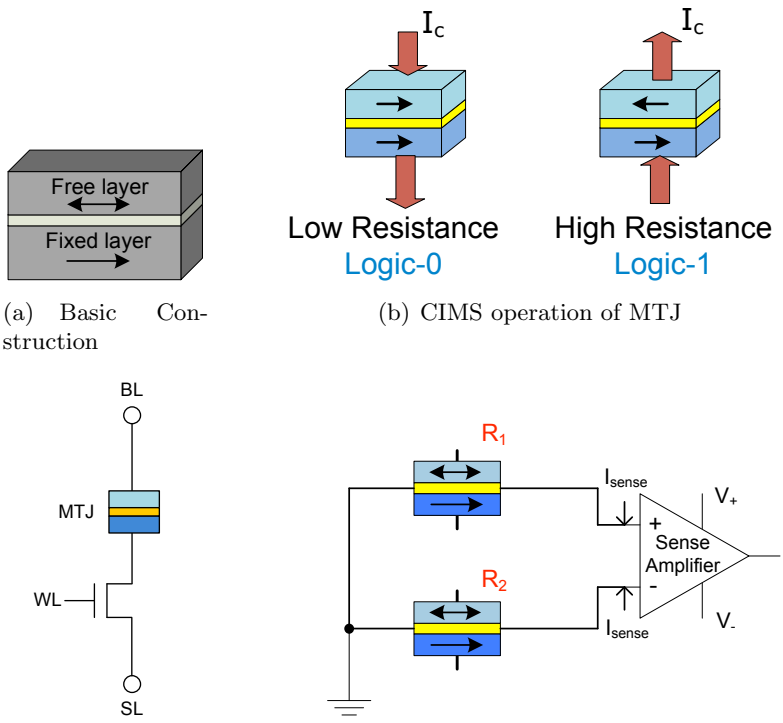
This is a rich field of research in that there is a continual need for fundamental studies, novel ways of engineering the devices, novel applications in computing circuits and novel ways of designing circuits. In this thesis, an approach is presented to begin thinking about taking the spintronic technology from what is known about it to the next level of what can be done with it. Since the bridge between application requirements and technology features is yet to be built, this discussion is also intended to provide a high-level perspective to the development of the technology in the direction of application in the computing area. Using our current knowledge of spintronics, the strengths, constraints and challenges of using MTJ devices in logic systems are examined in the next three chapters. The discussions in these chapters build towards problem

statements, proposed solutions and possible future directions.

4.2 Magnetic Tunnel Junction Device

The magnetic tunnel junction (Figure 4.1(a)) is a sandwich of two layers of ferromagnetic materials separated by a thin insulating barrier made of a metal oxide such as AlO or MgO [28]. The two ferromagnetic layers are called the free layer and the fixed layer. Each layer is magnetized when majority of the spins of the electrons in the layer are aligned in a single direction. When both the layers are magnetized in the same direction (parallel magnetization), the device exhibits a low resistance to current passing through it. When they are magnetized in the opposite direction (anti-parallel magnetization), the resistance of the device is higher. These two distinct resistance values, low (R_L) and high (R_H), can be used to denote the digital states of logic-0 and logic-1 for computing. The ratio $(R_H - R_L)/R_L$ is called the Tunneling Magnetoresistance (TMR) Ratio of the device. It quantifies the difference between the low and high values, and is indicative of the noise margin in a spintronic operation.

Usually, the fixed layer is magnetized in one direction and held constant by using additional layers. The magnetization of the free layer is controlled externally to set the resistance of the device as desired. One way of controlling the magnetization is by applying the magnetic field of a current carrying wire to the free layer. This is also called Field-Induced Magnetization Switching (FIMS). Another improved technique is due to a recently discovered mechanism called *spin torque transfer (STT)* [29, 30, 31]. With this effect, currents passing through an MTJ have been shown to magnetize the device. As shown in Figure 4.1(b), a current larger than threshold current (I_C) passing from top to bottom magnetizes the free layer in a direction parallel to the fixed layer, while a reverse current magnetizes it in the opposite direction. This method of controlling the magnetization of the free layer is known as *current induced magnetization switching (CIMS)*. For this technique, the input currents are $+I$ or $-I$ and can be denoted as logic-1 and logic-0 respectively. The resistance of the device can be measured by passing a small sense current (I_{sense}) through the device.



(a) Basic Construction

(b) CIMS operation of MTJ

(c) An MTJ-based memory circuit

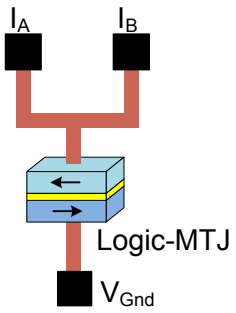
(d) An MTJ-based logic circuit

Figure 4.1: Magnetic Tunnel Junction - Device, Operation and Circuits

4.3 Related Work

Magnetic tunnel junctions have been demonstrated in 4Mb MRAMs using FIMS (first generation MRAM). Recent commercially developed 2Mb memory chips of Magnetic Random Access Memories (MRAMs) utilize the CIMS technique to write data to MTJ devices (second generation MRAM) [32]. A CMOS-based write circuitry converts data voltage into appropriate signals that generate directional currents through an MTJ. Reading of MTJ devices is accomplished using CMOS-based sense amplifiers, in a manner similar to reading any DRAM memory cell.

Apart from the memory applications of MTJs, logic operations of NAND, NOR, AND, OR and XOR have been demonstrated with the FIMS technique[33]. Designs for a full adder [34], a 1-bit ALU [9] and a 3-bit Gray counter [35] have also been proposed.



I_A	I_B	Net I through MTJ	Direction of I	Result State of MTJ-C (Preset H)
-I (0)	-I (0)	-2I	bottom→top	H (1)
-I (0)	+I (1)	0	-	H (1)
+I (1)	-I (0)	0	-	H (1)
+I (1)	+I (1)	+2I	top→bottom	L (0)

Figure 4.2: NAND operation using CIMS-based MTJs

The CIMS device operation has also been leveraged for performing logic operations and demonstrated by Wang et al in [36]. By constructing multiple current lines that act as inputs, the state of an MTJ is controlled by the net current that passes through it. With proper initial conditions and biasing, a single MTJ device performs logical operations of NAND, NOR, OR or AND on directional current inputs. Fig. 4.2 shows an example of an MTJ accomplishing the NAND operation. The MTJ is preset to the logic-1 state. Inputs are currents of magnitude $-I$ (logic-0) and $+I$ (logic-1), with $I = I_c/2$. Logical inputs (0,0),(0,1) and (1,0) do not affect the state of the MTJ. Only an input of (1,1) generates a net current of $+I_c$ through the MTJ that changes the magnetization of the device to a logic-0 state, giving the NAND truth table. This concept demonstrates the important characteristic of programmability of MTJ devices to accomplish four basic logic operations with a single device.

In another work, MTJs have been used in hybrid CMOS-MTJ circuits. The designs show a complete functional unit with the surrounding electronic circuitry that enables the read and write of the MTJs. The MTJs act as storage devices, connected to a suitable write circuitry and a sense amplifier circuitry, and are used to provide the function of a flip-flop [37, 38] and a non-volatile SRAM [39]. The inputs and output are electronic in nature, while data storage with the MTJ is non-volatile. Such use of MTJs shows the power advantages of hybrid CMOS-MTJ circuits without compromising on delay and area. Yet another work fabricated CMOS devices within MTJ memory to achieve the function of addition [40]. This adder implementation takes two electronic inputs and one spintronic input to perform the add operation with an electronic output. This work also proves the feasibility of integration of MTJ with CMOS as well as

demonstrates power advantages.

4.4 Computing with Magnetic Tunnel Junctions

Using MTJs in logic circuits adds a number of benefits. Firstly, the circuits themselves take the properties of the device: noise resistance, programmability, radiation-hardness, low-power operation, non-volatility. Secondly, the ability of memory devices to perform logic functions enables the possibiliies of accomplishing processing within memory, a task much desired by computer architects for various performance reasons.

The fundamental challenges in using magnetic tunnel junctions for logic applications can be identified at three levels of abstraction - the device, circuit and architecture level. At the device level: 1. MTJs are passive devices that store spintronic state reflected as a resistance (i.e. the output is resistance); and 2. their inputs are currents ($\pm I$). This input-output incompatibility manifests itself in a number of challenges in logic circuits. The main difficulty it presents is when connecting the output of one device directly to the input of another. To address this issue, the idea of using sense amplifiers to convert the resistance into a bidirectional current has been presented [41]. This is also used in memory circuits to 'read' the spintronic state of the MTJ device as an electrical signal, and has been used for other types of memories as well, such as DRAMs, SRAMs, memristors and other memory devices. This works very well in memory architectures, since an entire array of memory devices shares a sense amplifier, thus requiring far fewer sense amplifiers than the number of devices. However, if the same concept is used for a logic device, then the logic 'device' will not be a single MTJ device, but an MTJ device with its read and write circuitry (about 5-10 electronic transistors). This logic composition is shown in Fig. 5.1. Though this logic 'device' will solve the compatibility issues, it will be much larger, more power consuming and much more complex than a single transistor. However it will also be non-volatile, which makes it useful in an orthogonal application. Functionally, this circuit is similar to that of a flip-flop. Due to its non-volatility, it is more sensible to compare it to a 'non-volatile' transistor circuit, than with a single transistor. This comparison with an electronic flip-flop has been done in work by Hayakawa et al. in [37, 38].

Going back to our initial objective of building logic circuits with MTJs, this 'logic'

unit is not such a good device for building say, a NAND gate. Thus, aside from being able to replace memory devices like SRAMs, DRAMs, flash memories, etc, or building flip-flops (again, memory elements), it has been general opinion that MTJs are not very promising devices for logic. However, if there was a way to solve this challenge, if it were possible to build a NAND gate more simply, or to keep ourselves free from limiting our imagination, *any* logic gate, then the possibilities of MTJs for logic open up immensely. Thus, this first and foremost device-level challenge for logic poses a fundamental issue to the logic capability of MTJs. This chapter proposes a scheme to address this challenge, by describing a circuit called as the 3-MTJ logic circuit (also referred to as 'Voltage Logic' in ensuing publications [42]). This also comprises a spintronic logic gate, with spintronic inputs and spintronic outputs.

At the logic gate abstraction, the MTJs in the circuit present three further challenges:

1. The two 'input' MTJs require data to be stored in them prior to the logic operation.

2. The 'output' MTJ must be appropriately preset for logic operations.

3. The MTJs do not provide amplification. Therefore, the current flowing out from an MTJ is either the same or decreased current that is input through the device. Logic circuits designed using MTJs must work within this constraint, and not require signal restoration of any form.

4. Starting with a single device that performs the basic logic operations, there will be the need to *compose* them into more complex operations. The conventional design strategy (levels of basic operations enabled in multiple time steps) may not be acceptable on account of performance loss, thus requiring a fundamentally novel method of design. It is not yet clear how to implement spintronic circuits consisting of MTJs that can perform complex logic operations such as multiplication.

The last point describes the challenge of *composability* with spintronic devices. A scheme called Union with Neutralization is propsed in this chapter to address this issue.

For the data storage and preset requirement, write circuitry needs to be associated with each of the MTJs in the circuit. By adding such a write circuitry, it will become

possible to devise logic 'gates' that enable more complex functions. However, at this higher level of abstraction, it is important to note that MTJ elements are fundamentally different from CMOS devices and have special requirements for design:

1. The very requirement of stored data inputs and a preset data output device requires that there be an explicit data input and a preset step before a logic operation.

2. A logic state is a stored spintronic state, not a traveling charge signal. This implies that data movement is not automatic, and therefore the logic state does not propagate on its own through multiple levels of logic. Each stage will need to be enabled separately, giving rise to multiple time-steps in a cascade operation.

3. The spintronic logic gates proposed are resistive circuits, and hence consume power during a logic operation (dynamic power), even when there is no switching of a spin state. (Note that when no logic operation is being performed, there is no need to bias the circuit and therefore there is zero power consumed (static power)). Thus, there will be a need to reduce the power consumption in some circuit design or architecture design technique.

The dynamic power consumption of the circuit can be decreased by reducing the critical current required to write an MTJ. Research efforts focused on this issue are currently being devoted for this purpose.

Working with the challenges posed at various levels, and working within the constraints, the next three chapters propose ideas for addressing the challenges of compatibility, composability, reducing the number of electronic devices in MTJ-based circuits; and leveraging the advantages of the technology for computing through the concept of logic-within-memory.

Chapter 5

Spintronic Logic Gates

Logic gates form the basic building blocks in a digital design. They perform the smallest unit of information processing, by transforming inputs logically into a result that can be further processed upon. An important characteristic of logic gates is their input-output compatibility, i.e. the similar nature of signals that form the inputs and outputs (voltages in MOSFETs). This property, that has been taken for granted in electronic devices, is not characteristic of spintronic devices.

As discussed before, sense amplifiers allow different bias requirements to be met easily, and hence using sense amplifiers for reading spintronic memory is beneficial when MTJ-based storage devices are used within CMOS circuits that are performing computation. When the same concept is used for computation of logic operations using MTJs themselves, the circuitry consists of intermediate electronic circuitry to convert signals between the spintronic devices. The possible circuit shown in Fig. 5.1, where the sense amplifier senses the resistance difference between the MTJ to be read (MTJ-A) with a reference MTJ (MTJ-Ref) in a low state and outputs a voltage of 0V or +Vdd, which are then fed into a write circuitry consisting of current mirror circuits that generate the critical current for the logic-MTJ. Clearly, when it is desired to perform simple logic operations on data stored in MTJs using MTJs themselves, this extra level of electronic circuitry within the spintronic unit puts extra overheads on the circuit. This motivates the need for a simpler mechanism that performs the function of a sense amplifier, but entirely in the spintronics domain with as little extra circuitry as possible. To address this requirement, a logic circuit [8] is proposed in this chapter, that senses

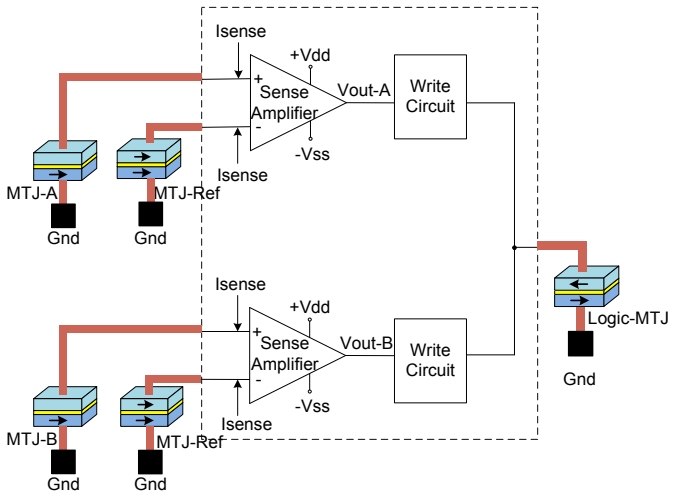
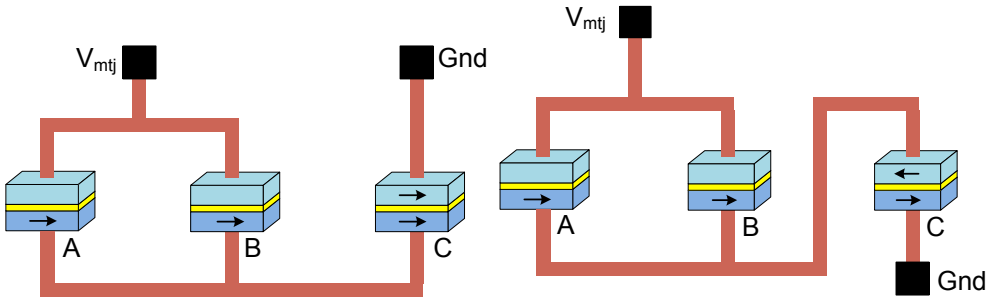


Figure 5.1: Logic operation using an MTJ for data stored in two MTJs with intermediate electronic circuitry for reading and writing data

or 'reads' the data in MTJs and computes the result of a logic operation in a third MTJ with no intermediate electronic circuitry.

5.1 Circuit for Logic Operations using MTJs

Based on the observation that the inherent spintronic state of an MTJ is a resistance, the proposed logic circuit senses the net resistance of a circuit to generate a current through it. The circuit consists of three MTJs A, B and C connected as shown in Fig. 5.2(a) or Fig. 5.2(b). The inputs are stored as a spintronic state in MTJs A and B. The MTJ that performs the logic operation and stores its result within itself is MTJ-C. In Fig. 5.2(a) MTJ-C performs the NAND or NOR operation on the data stored in input MTJs A and B. MTJ-C initially has a logic state of low, while input MTJs A and B have resistance-type data, i.e. a spintronic state corresponding to logic-0 or logic-1. Since the input devices are connected in parallel, the circuit current gets divided into the two branches without subjecting any input device to critical current values. This ensures that the states of the 'input' MTJs remain unchanged through the operation. On applying bias voltage V_{MTJ}, the total current that flows through the circuit experiences the resistance of all three devices. If this happens to be above the critical value, it changes the state of



(a) An entirely MTJ-based circuit for NAND/NOR operation

(b) An entirely MTJ-based circuit for AND/OR operation

Figure 5.2: Proposed MTJ-based circuit for logic operation

the result MTJ (MTJ-C). The circuit current then experiences the new total resistance of the circuit and settles into a steady state. If the initial current is not above the critical current value, the result MTJ retains its initial state.

The value of V_{MTJ} controls the operation performed by the circuit. For the NOR operation, this value is selected so that the current that flows through the circuit is greater than I_C only when MTJs A and B are low, while for the NAND operation, the current is greater than I_C for the first three rows of the truth table, when either or both the inputs are low. Thus, the same circuit performs both logic operations at different bias voltages.

Fig. 5.2(b) shows a variation of the circuit where the current passing through the logic-MTJ is in a direction from top to bottom. In essence, this can also be achieved by the circuit in Fig. 5.2(a) by simply interchanging the voltages on terminals V_{MTJ} and *Gnd*. For the OR operation, the value of V_{MTJ} is adjusted such that the current flowing through the circuit is greater than I_C only when MTJs A and B are low, while for the AND operation, bias voltage is such that the current is greater than I_C for the first three rows. This is summarized in Table 5.1.

Table 5.1: Selection of bias voltage for implementation of the four logic operations

Logic Operation	Implementation (Set V_{MTJ} such that)	Voltage requirement for delay of 1ns for MTJ devices [a]
NOR	$I > I_C$ when both inputs are low	1.7V
NAND	$I > I_C$ when either or both inputs are low	1.8V
OR	$I > I_C$ when both inputs are low	2.5V
AND	$I > I_C$ when either or both inputs are low	2.6V

[a] fabricated in [43]

5.2 Verification and Fabrication

The circuit in Fig. 5.2(a) can be simulated using a SPICE model of the MTJ [44] for functional verification of the NOR operation. For this simulation, MTJ device parameters reported for a fabricated device [43] were used: R_{low}=3472 Ω, R_{high}=5902 Ω, calculated I_C=320uA for a device of size $120nmx240nm$. The bias voltage (V_{MTJ}) calculated for the function of NOR, at a delay of 1ns, is 1.66V. Note that the NOR operation can be obtained even for a lower voltage, albeit with a longer delay, while it can be obtained with a smaller delay at a higher voltage.

After applying a bias voltage of 1.7V to the circuit in Fig. 5.2(a), Fig. 5.3 shows the resistance of the three devices for four combinations of inputs. During NOR operation, the resistance of MTJ-C changes to a high state when MTJs A and B are in the low state. This occurs at about 300ps. Fig. 5.4(a) shows the circuit current for MTJ-A=MTJ-B=R_{low} in the NOR circuit. The initial current is above the critical value I_C which changes the resistance of MTJ-C, and then settles into a steady state with a magnitude lesser than I_C. The waveforms also show that the current that flows through individual branches of input MTJs does not change their state.

The same circuit (Fig. 5.2(a)) can be used for NAND operation when a bias voltage (V_{MTJ}) of 1.8V is applied. This bias voltage is large enough to generate a current more than I_C for the first three rows of the truth table. The circuit current settles into a steady state after the NAND operation, as shown in Fig. 5.4(b).

For the AND and OR operations, MTJ-C is initially preset to a high resistance state. A bias voltage of 2.5V is required for accomplishing the OR function, while a

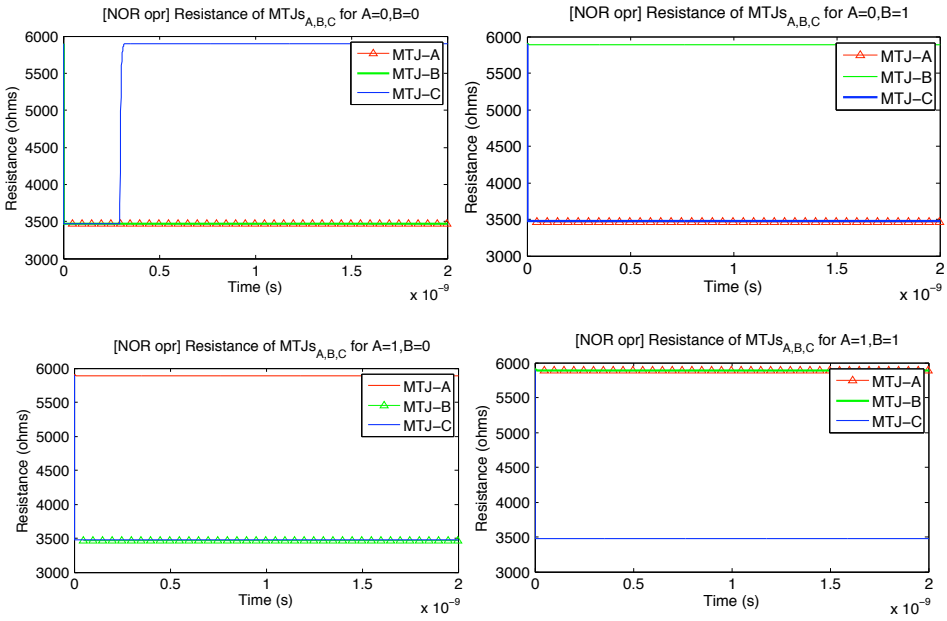
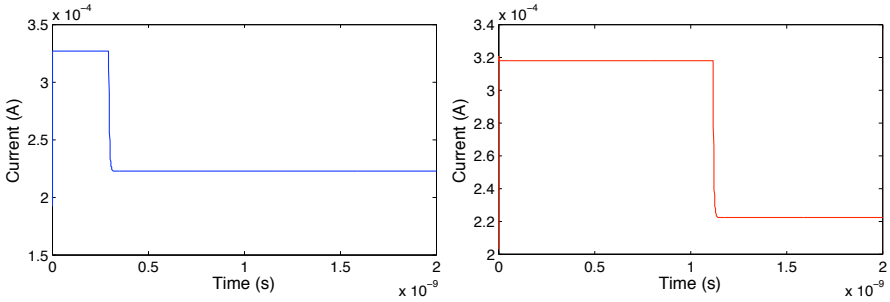


Figure 5.3: (Color online) SPICE simulation for functional verification for the NOR operation with bias voltage=1.7V. Resistance (ohms) of MTJs A, B and C shown vs Time (s).

bias voltage of 2.6V is required for the AND operation. This can either be accomplished by the circuit in Fig. 5.2(b) or by the circuit in Fig. 5.2(a) by interchanging the terminals of V_{MTJ} and Gnd.

Thus, in general, the circuit in Fig. 5.2(a) is programmable for the four logic operations by applying suitable voltages to the two terminals. The differences in bias voltages depend on the TMR ratio of the devices. This is the ratio $(R_H - R_L)/R_L$. The higher the ratio, the greater the difference between the voltages required for different logic operations, and hence the greater the noise or process variations that can be tolerated by the circuit.

To demonstrate the idea of the proposed circuit, this 3-MTJ circuit was fabricated and verified. Fig. 5.5 shows the optical image of the fabricated circuit on a chip. More details and characterization measurements are presented in [45].



(a) Circuit Current for A=0, B=0 in NOR circuit ($V_{MTJ} = 1.7V$)

(b) Circuit Current for A=0,B=1 in NAND circuit ($V_{MTJ} = 1.8V$)

Figure 5.4: Current in the MTJ-based logic circuit obtained from simulation

5.2.1 Expected trends in critical parameters

An important consideration for a logic circuit in terms of performance is its delay and power. The delay of the circuit is primarily governed by the magnitude of the switching current (I) flowing through the MTJ which is generated due to the bias voltage (V_{MTJ}), while the power consumption is based on the bias voltage and the current through the circuit (I). The bias voltage applied for the NOR and NAND operation is calculated as a direct function of the critical current density J_c of the fabricated devices. Thus, both the delay and the power consumption of the circuit are influenced by the bias voltage requirement and the critical current density. As the critical current density decreases, the power requirements for the logic operation decrease. A great amount of research aimed at reducing the critical current density of magnetic tunnel junctions is currently underway. The critical current density demonstrated in only a few years shows a drastic reduction, from $16MA/cm^2$ in 2005 to $1MA/cm^2$ in 2007[46, 47, 48, 43]. It is commonly believed that it is extremely difficult to lower the current density below $1MA/cm^2$. Thus, there is great opportunity to propose fundamental ideas to reduce the write energy of MTJs. The device reported in [43] with a critical current density of $1MA/cm^2$ is a dual barrier structure specially designed to reduce the critical switching current, while behaving as a standard MTJ. With the parameters of the devices fabricated and characterized in the three years, we calculated the bias voltage requirements for the NOR, NAND, OR and AND operations for the proposed circuit. These are shown in

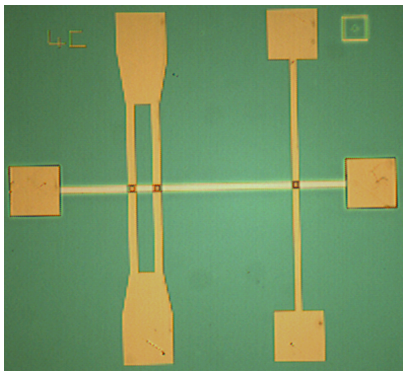


Figure 5.5: Optical Image of fabricated circuit

Fig. 5.6. With fabrication of smaller MTJs, these voltages decrease proportionally with the area reduction. The bias voltage requirements for the MTJ device fabricated in [43] ($120nm \times 240nm$ device with a critical current density of $1MA/cm^2$) are comparable to those of the 180nm and 130nm CMOS technologies. Thus, the proposed MTJ circuit operates on practical voltage values.

5.3 Impact on computing

The proposed circuit offers some interesting and novel possibilities for computation. Without the need for intermediate electronic circuitry, the circuit provides the potential to perform logic operations within a non-volatile memory unit, using the memory devices themselves. This capability is a critical advantage over the conventional von Neumann approach to computing, where the logic units and memory units are separated because the same circuits cannot achieve both functions simultaneously. Working around this constraint of CMOS-based devices, the concept of logic-in-memory with von Neumann architectures has been researched extensively, in which the processor is brought closer to memory in order to reduce the processor-memory communication bottleneck. Such processor-in-memory architectures have shown significant performance benefits for data-centric applications like image processing, signal processing, etc. ([49, 50]). With a device capable of simultaneous storage and logic operations, the memory element itself becomes a processor, and the overhead of communication further decreases.

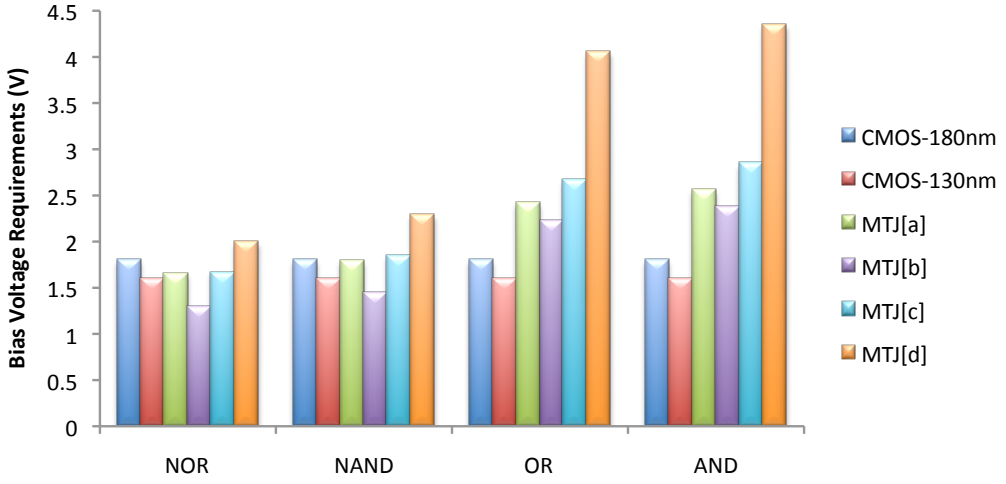


Figure 5.6: Bias voltage requirements for logic operations of NOR, NAND, OR, AND in fabricated devices

[a]MTJ device fabricated in [43]
[b]MTJ device fabricated in [46]
[c]MTJ device fabricated in [47]
[d]MTJ device fabricated in [48]

In a general-purpose computing system, data resides in non-volatile memory elements like hard drives, flash memory cells, etc. Since these are relatively slow compared to the fast computing circuits, data is brought closer to the processor using an intermediate hierarchy of volatile memory elements consisting of CMOS-based DRAM cells and SRAM cells in order to reduce the communication costs and increase performance. The complexity of such a setup could be potentially alleviated by using a logic-and-memory circuit for simple logic functions. To investigate the impact of this idea, the energy and delay measurements of a CMOS-based logic operation setup are compared with the MTJ-based logic circuit. Device parameters related to the recent technology status for both technologies are used in this comparison. The delay and power consumption of a single logic gate using the CMOS technology is lower than the MTJ-based logic circuit using devices fabricated in the laboratory so far. However, a complete data path for a logic operation with the CMOS technology on stored data includes the memory elements that contain input data and those that contain the result of the operation. To compare against the non-volatile MTJ memory cells performing a single logic operation

on stored data, the faster but volatile CMOS-based SRAM cells are chosen as memory elements that contain the input data and will store the result of the operation.

5.3.1 Experimental Methodology

A CMOS-based circuit performing the NAND function on data stored in SRAM devices is shown in Fig. 5.7. The 1-bit input operands are stored in two SRAM cells (shaded in the figure), which feed into a simple ALU circuit capable of performing the four logic operations of NAND, NOR, AND and OR. The result is stored into a third SRAM cell (shaded in the figure). In order to read and write the data in the SRAM cells, supporting circuitry is required. Thus, the two input SRAM cell arrays contain precharge and read circuitry, while the output SRAM cell array contains the precharge and write circuitry.

A 3-MTJ circuit that performs a logic operation and stores operands and the result of the logic operation is functionally equivalent to a 3-cell SRAM circuit performing a logic operation using a logic gate, except that the spintronic implementation is also non-volatile. However, for CMOS-based SRAM circuits, the read and write delay also depends on the number of cells connected to a single bit line. Performance of the SRAM configuration degrades as the number of cells increases. The number of cells in the cell array depends on the design of the memory unit and the requirements of the overall system in which it resides. In order to incorporate this effect of CMOS circuits, an n-bit SRAM cell array was built. The number of cells per bit line (or array size) are scaled to show the trade-off points of performance between MTJ and SRAM circuits.

This SRAM-based circuit was simulated in HSPICE using CMOS devices from 180nm and 130nm technologies, and the power and delay of a single-bit operation was measured. The bias voltage (Vdd) used for the 180nm and 130nm simulations was 1.8V and 1.6V, respectively. An assumption is made that data is already present in the input SRAM cells. Being volatile in nature, the SRAM cells consume power just for retention of the data before and after the logic operation. This power consumption that disadvantages the CMOS-based setup is ignored in this comparison.

The MTJ-based circuit in Fig. 5.2(a) is used to determine the current through the circuit for the time of a clock cycle. Since an MTJ circuit must be isolated from the rest of the circuitry in order to have a minimal connected path for current to flow through so that the power consumption is minimized, we simulated a single 3-MTJ circuit.

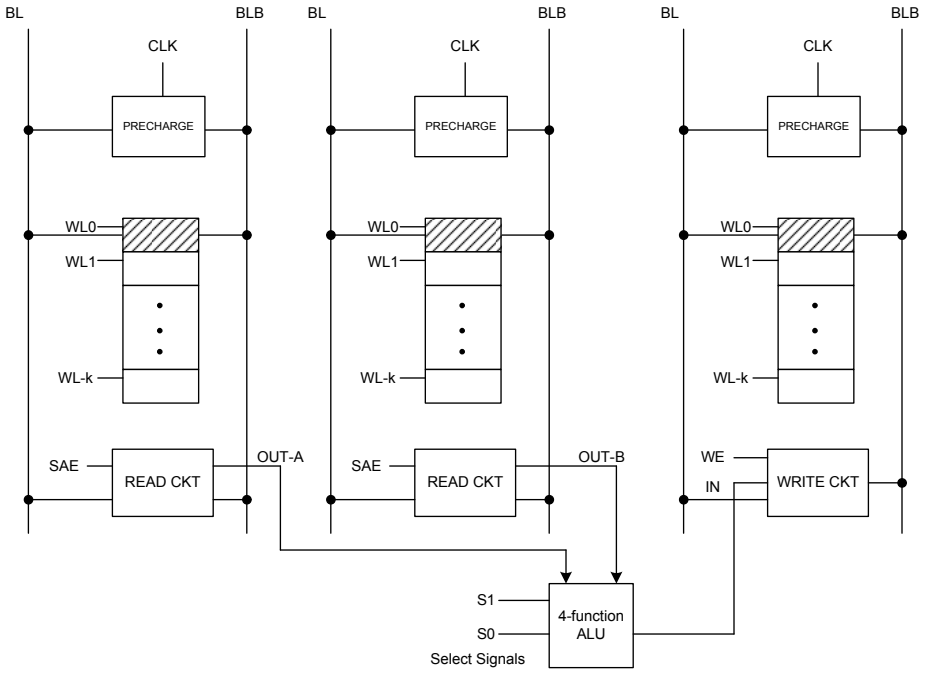


Figure 5.7: CMOS-based circuit for logic operations on data stored in volatile SRAM cells

It is assumed that the result MTJ is already preset to logic-0 before the operation commences. This assumption will be discussed in Section 5.3.3.

The CMOS-based circuit is simulated with minimal sized devices. Each SRAM cell is implemented as a 6T circuit. A single logic-and-memory MTJ-based circuit clearly provides area advantages over a separate logic and memory module using CMOS-based circuits. This area advantage is further enhanced by the scalability possible with the spintronic technology. This work does not quantify he area advantages, instead it focuses on the power and delay performance of the MTJ-based spintronic technology to get an insight into the potential of the technology to improve circuit performance.

5.3.2 Results: Power-Delay Product (PDP) and Energy-Delay Product (EDP)

The power consumption of the MTJ circuit is given by $V.I_{avg}$ for the length of a high clock cycle. Since the bias voltage for the spintronic device is computed for a delay of

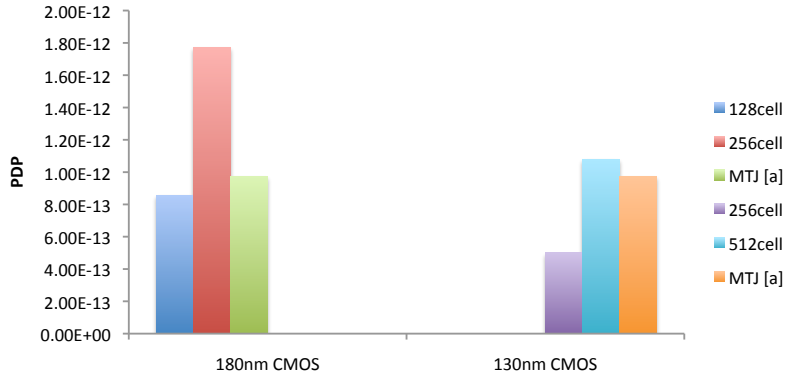


Figure 5.8: Comparison of the Power-Delay Product of 180nm CMOS-based SRAM circuit with that of MTJ-based circuit

[a]MTJ devices reported in [43]

1ns, we apply a clock high time of 2ns and compute the power consumption for that period. This ensures that the operation is reliably completed.

The CMOS-based operation consists of read, write, transfer and the logic operation. In this evaluation, the power and delay of the operation including the reading of operands, the logic function and the writing of the result is measured. The delay and power consumed during the transfer of data through communication lines between memory module and the ALU is ignored, since it is highly dependent on the layout of the actual chip. Fig. 5.8 shows the PDP comparisons between the CMOS and MTJ technologies with our simulation parameters. The data points show the trade-off points of performance of the MTJ-based circuit with respect to 180nm and 130nm CMOS technologies. The PDP of a logic operation using device parameters of $120nmx240nm$ MTJs falls between that of a 180nm CMOS-based logic setup with 128 cells and 256 cells connected to a single bitline in a memory array. With 130nm technology, the PDP of the MTJ-based logic operation falls between that of a CMOS-based memory array with 256 cells and 512 cells per bitline. The energy-delay product is an indicator of the energy-efficiency of the logic operation. The EDP for the CMOS and MTJ circuit configurations is shown in Fig. 5.9. For the 180nm technology, though the performance of the MTJ is between the CMOS circuit configurations, the EDP of the MTJ circuit is better than both the CMOS circuits.

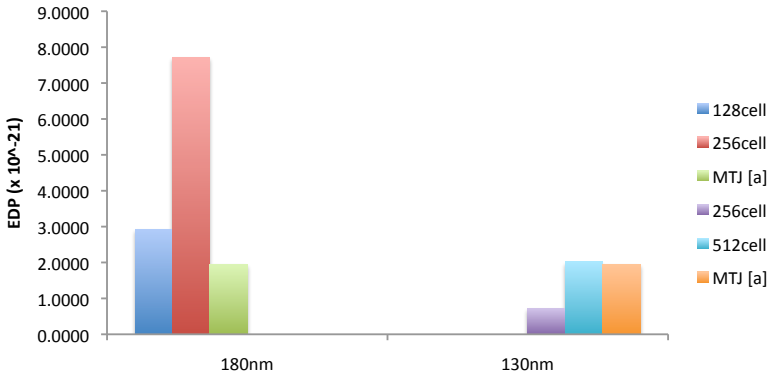


Figure 5.9: Comparison of the Energy-Delay Product of 180nm CMOS-based SRAM circuit with that of MTJ-based circuit

[a]MTJ devices reported in [43]

5.3.3 Note on additional supporting circuitry

In order to develop into a complete logic-and-memory module, there are three fundamental functions required of each MTJ in the unit: read, write and logic. When data is present in the devices, these three functions occur inherently in the proposed circuit during a logic operation (the circuit current is established after 'reading' the inputs – this current causes a 'write' in the result-MTJ). When data is not already present in the devices, an external read and write is necessary. To add a read/write capability to the present circuit, simple switches would be required to establish a read/write path. These switches would also allow the circuit to toggle between the logic mode and the memory mode.

The proposed circuit also requires a 'preset' mode for the result MTJ before applying inputs. In a CMOS analogy, this is similar to the requirement of precharging the bit lines of a memory module before applying read or write signals. In the proposed circuit, a preset may be achieved through external write circuitry in a separate and necessary time step before a logic step. Another way is by applying a suitable voltage to the present circuit that will set the state of the result MTJ to the desired value irrespective of the states of the input MTJs. However, to prevent the current due to this 'preset' voltage from affecting the states of the input MTJs, the circuit must include an additional MTJ ('preset' MTJ) in parallel with the input MTJs. Since the proposed circuitry inherently

writes to the result-MTJ, the preset state will be written to the result MTJ while the preset-MTJ protects the input data.

Alternatively, techniques to amortize the overhead of the preset may be used. Though beyond the scope of this thesis, it may be possible to preset an entire unit of MTJ-based logic-and-memory cells by using separate magnetic or spintronic means. For example, with a current plane above the array of result-MTJs, a current may be passed through the plane to magnetize the free layers of the entire row of result-MTJs at once.

5.4 MTJ-based programmable logic circuit

The circuit in Fig. 5.2(a) demonstrates a technique of employing MTJs in a circuit to obtain a spintronic logic output without intermediate electronic components. The actual logic computation on the inputs is brought about spintronically. While this circuit presents a feasible technique of achieving a completely spintronic computation, it runs into problems of scalability and composability. Specifically, the TMR of the devices puts a limit on the fan-in of the logic circuit. To alleviate these issues, a modification of the circuit is proposed that makes it programmable for any logic function, as well as more scalable.

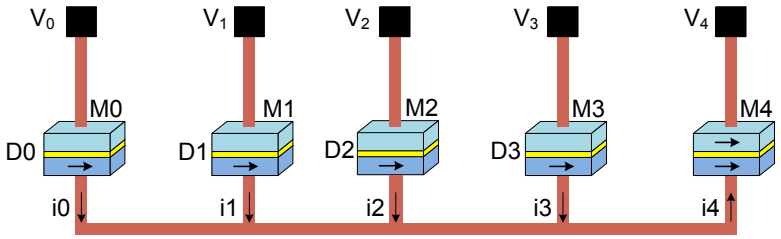


Figure 5.10: Programmable spintronic logic circuit

5.4.1 Design of the circuit

Fig. 5.10 shows the proposed circuit to implement a 2-input truth table. The biasing terminals are separated and the circuit has four MTJs (M0-M3) that store the complement of the truth table to be implemented as data D_0 through D_3. In this circuit, D_0

is the complement of the desired output when the inputs are (0,0), D_1 is the complement of the desired output when the inputs are (0,1) and so on. The logic operation is a selection operation. The terminal corresponding to the truth table row that has to be selected is biased with a voltage V_{select} and at the same time, the other terminals corresponding to the other inputs are biased with a voltage $V_{no-select}$. The magnitude of voltage $V_{no-select}$ is about half that of V_{select}. The terminal V_4 is grounded, so that most of the current flows into this terminal. MTJ M4 is the result MTJ. It is preset to a LOW state. When an input pulse of the bias voltages is applied, currents are established in the circuit as a function of the input voltages and the exisiting resistance states of the MTJs.

For example, if the data in the input MTJs is (0,1,1,1), and the data D_0 must be selected, the current in the circuit is $I_{0111} = (V_{select}/R_L + 3V_{no-select}/R_H)/(2 + 3(R_L/R_H))$. On the other hand, if the data is (1,0,0,0), then the current in the circuit is $I_{1000} = (V_{select}/R_H + 3V_{no-select}/R_L)/(4 + R_L/R_H)$. If this resulting current through M4 is greater than the critical current of the MTJ, it changes the state to logic HIGH. By carefully choosing the value of V_{select} and $V_{no-select}$, the relationship $I_{1000} > I_c > I_{0111}$ can be established, so that in the first example, I_{0111} exceeds the critical current, and the result MTJ is set to logic HIGH, while in the second case, I_{1000} is less than the critical current and the result MTJ does not change its preset LOW state. Therefore, at the end of the operation, the desired truth table output is available in MTJ M4.

The main idea is that the applied non-selecting voltages produce non-zero, but weak, currents in the circuit, while one stronger current is the deciding current that, when combined with the weak currents, either exceeds the critical current value or not. This is the total current that flows through the result MTJ and it either changes the resultant state or not, in order to produce the final output. Due to the presence of the weak currents in the circuit, the current flowing through the input MTJs are not individually strong enough to change the input state.

When voltage sources are thus added on the individual terminals, the circuit becomes scalable. With the same value of voltage biases, the circuit operates correctly for a 4-input-MTJ as well as a 64-input-MTJ circuit. The circuit in Fig. **??** is not largely scalable since the circuit operates on the difference between the combined resistance of the input MTJs. This difference is thus less than what is afforded by the TMR ratio of

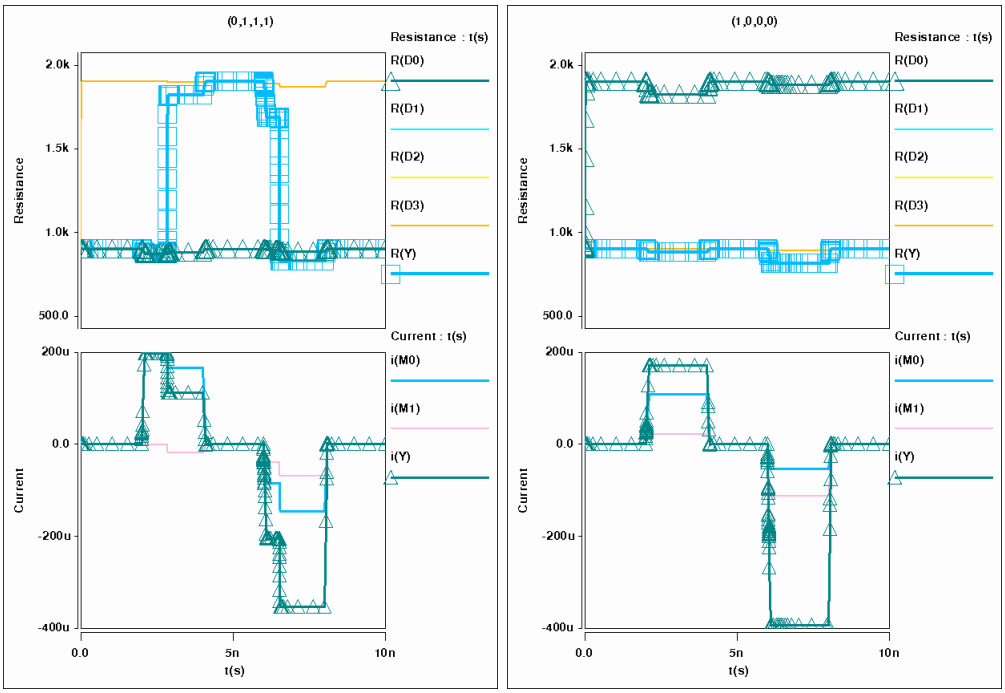
the devices. As the number of inputs increase, the distinction between the resistances of different maxterms vanish, until eventually the difference is too small to detect reliably in the presence of noise. The proposed modified circuit operates completely at the resistance difference specified by the TMR ratio, thus allowing for a higher margin than before to set the operating point of the circuit.

The circuit requires the result MTJ to be preset LOW before the selection operation. This is achieved by applying a preset voltage V_{PRE} to terminal V4, while the rest of the terminals are grounded. The four input MTJs combine into a parallel resistance that is lower than the total resistance of the circuit with only a single input MTJ. The total current also gets divided into the four branches. Therefore, though the magnitude of the current is strong enough to 'write' or preset the result MTJ M4, it is not strong enough to affect the input MTJs.

5.4.2 Simulation

In order to verify this operation, the 5-terminal, 5-MTJ circuit was simulated using HSPICE. The simulation model used for each MTJ is a SPICE model described in [51]. The specific MTJ device parameters used with this model were based on a device that was recently fabricated and characterized in [52]. The low and high resistance states of the device used for simulation were 900Ω and 1900Ω with a TMR of 111%. For simulations, a critical current density of $3MA/cm^2$ was used, resulting in a critical current of $195\mu A$ assuming a rectangular geometry for an MTJ device of size $130nm$x$50nm$. All logic operations were verified by selecting M0 each time and varying the data in M1 through M3. Due to the symmetry of the circuit, only eight unique operations were required to be verified. Denoting each experiment as $\{D_0, D_1, D_2, D_3\}$ where D_i is the data stored in M_i, the eight experiments were {0,0,0,0}, {0,0,0,1}, {0,0,1,1}, {0,1,1,1}, {1,0,0,0}, {1,0,0,1}, {1,0,1,1}, {1,1,1,1}. For each of these operations, an input voltage pulse of magnitude V_{sel} was applied at V_0, and a voltage pulse of magnitude V_{nosel} was applied at V_1 through V_3. At the end of this operation, the resulting state of M4 was the negation of D_0. To incorporate the preset energy in the energy measurements, a preset operation was performed after the logic operation (instead of *before*, for convenience of simulation). For this, the terminals V_0 through V_3 were grounded and a voltage pulse of magnitude V_{pre} was applied at V_4. This preset M4 to a LOW value. Each pulse applied

was 2ns wide. The voltages applied for the operation were: V_{sel}=350mV, V_{nosel}=170mV, V_{pre}=420mV. Fig. 5.11 shows the output of the operation and the circuit current for the cases {0,1,1,1,} and {1,0,0,0}, when input biases are applied for 2ns (from t=2ns to 4ns) and preset bias voltages applied for 2ns (from t=6ns to 8ns). Note that the circuit can also work correctly for slightly lower voltages than those applied in the simulation, however, it will take longer for the output MTJ to change its state. Therefore, a longer voltage pulse will be needed.



(a) $D_0 = 0, D_1 - D_3 = 1$ (b) $D_0 = 1, D_1 - D_3 = 0$

Figure 5.11: Simulation waveforms

5.4.3 Varying circuit parameters

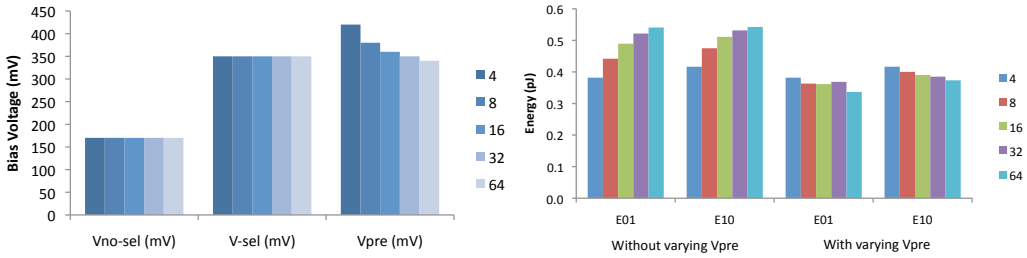
The maximum delay among the eight operations at these voltages was 949.5ps and the average energy for the simulation of a logic operation followed by a preset was 0.4pJ. The energy and delay of the logic and preset operation are affected by the bias voltages. In

this section, the effect of changing the circuit parameters on its performance is discussed.

Effect of scaling

In order to implement larger truth tables, the number of 'input' MTJs in the circuit will have to increase. This increase in the fan-in of the circuit over the original 4-input-MTJ circuit shown in Figure 2 affects circuit properties in two ways. First, in the logic operation mode, the non-selecting MTJs decrease the overall resistance of the circuit, which is a desirable effect. If the number of inputs is increased further, it may also allow for a lower V_{nosel} that can maintain the same magnitude for the total weak component of the circuit current. The effect of decreasing the bias voltages will be discussed in the next section. The decrease in the total resistance of the circuit allows for a decrease in the preset voltage for the output MTJ. The circuit was scaled from 4 input MTJs up to 64 input MTJs, which corresponds to a 6-input truth table, and the preset voltage was deceased to the minimum possible. Fig. 5.12(a) shows the decrease in preset voltage possible.

The second effect is that, at the same bias voltages, as the fan-in increases, the overall energy consumption of the circuit increases. This occurs because, with a lower total resistance in the circuit, the currents flowing in the circuit increase, and hence power dissipation increases. This current increase can be counteracted by decreasing the applied bias voltages to the minimum possible. With the scaled circuits, we found that they operated correctly at the same bias and preset voltages as those applied to the 4-input-MTJ circuit. Therefore, the energy consumption of the scaled circuits did not increase largely as compared to that of the original 4-input-MTJ circuit. Fig. 5.12(b) shows the energy consumption for two corner cases of inputs: $\{D_0$=0,D_1-D_{n-1}=1$\}$ denoted as E01 and $\{D_0$=1,D_1-D_{n-1}=0$\}$ denoted as E10. The total energy dissipation of the logic and preset operations in the 64-input circuit was 30-40% higher than that of the 4-input circuit with equal bias voltages applied to both circuits. However, when the decrease in preset voltages possible due to scaling was incorporated, it decreased the energy consumption slightly. Thus, choosing minimal bias voltages enables the circuit to scale efficiently.

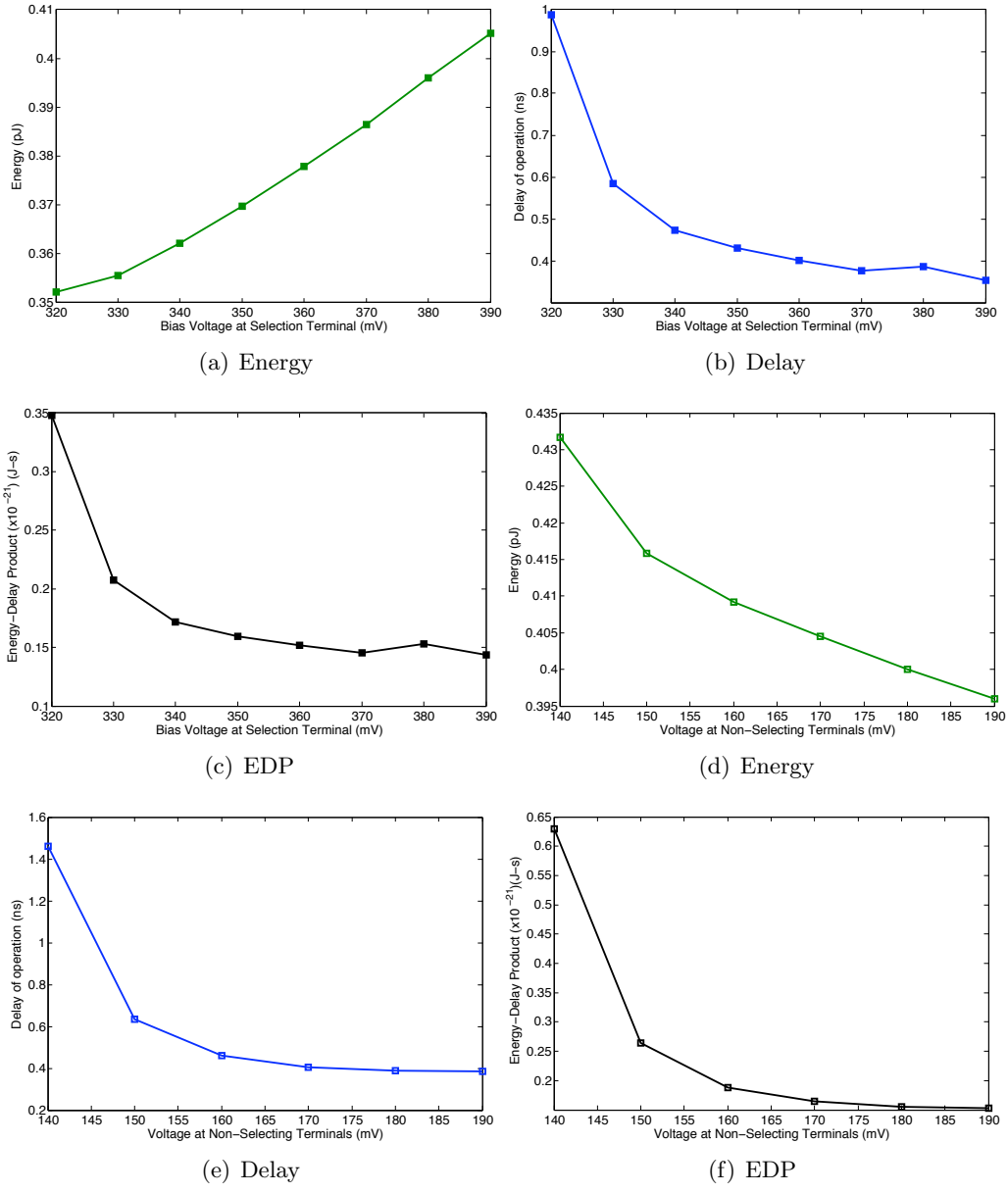


(a) Biasing requirements (b) Energy Consumption

Figure 5.12: Effect of scaling the circuit

Effect of changing bias voltages

In general, an increase in the applied bias voltages results in a decrease in the delay of an operation and an increase in the energy consumption. Therefore, the selection of bias voltages depends on the energy-delay trade-offs of the circuit. Figures 5.13(a) and 5.13(d) show the change in the energy of the logic operation (including the preset operation) for the case when D_0=0 and D_1-D_3=1, as the bias voltages are varied individually. Figures 5.13(b) and 5.13(e) show the delay of the logic operation for the same case. Figures 5.13(c) and 5.13(f) show the product of the energy and delay measured in the previous two curves, indicating the overall effect of varying a bias voltage on the circuit. With an increase in voltage V_{select} while keeping $V_{no-select}$ constant, the energy of operation increases as expected. On the other hand, increasing the voltage $V_{no-select}$ while keeping V_{select} constant decreases the energy consumption, as delay of the operation decreases more rapidly. However, in doing this, the difference between the voltages applied at the selecting and non-selecting terminals has decreased, resulting in a lower noise margin between the applied voltages. The TMR ratio of the devices themselves puts a limit on how much the applied voltages can be varied. For example, with $V_{no-select}$=200mV, the voltage V_{select} can only be varied from 300 to 330mV, since any increase beyond that results in an increase in the amount of the current in the circuit, so much that the result MTJ changes its state even when the input MTJ is logic high. In Fig. 5.13, when voltage V_{select} was varied, the voltage $V_{no-select}$ was kept constant at 190mV, while when the latter was varied, the voltage V_{select} was kept constant at



(a) Energy (b) Delay

(c) EDP (d) Energy

(e) Delay (f) EDP

Figure 5.13: Effect of varying the bias voltages on the selection and non-selection terminals on energy, delay and energy-delay product (EDP) of the circuit in Figure 2

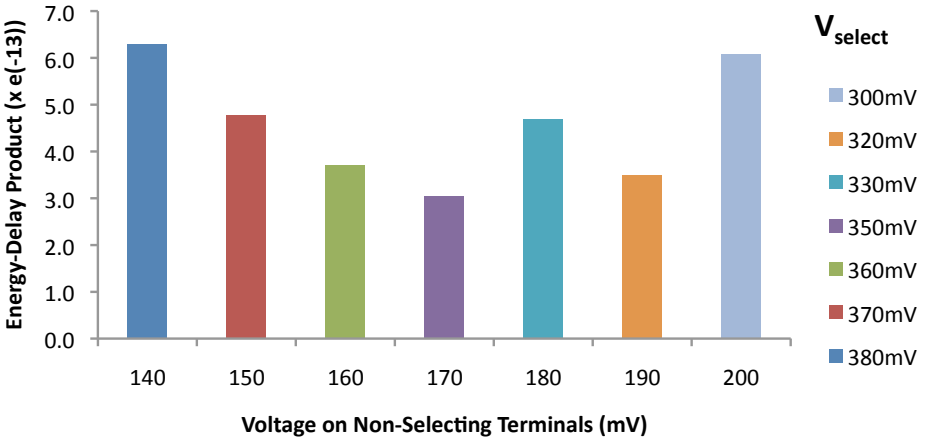


Figure 5.14: Combined effect of varying the voltage applied at the selecting and non-selecting terminals

380mV. These values were chosen to obtain the maximum range for varying the other entity in order to allow us the observe the trend. Fig. 5.14 shows the energy-delay product of the circuit when both the bias voltages are changed at the same time. The best operating point for the lowest energy-delay product in these simulations was found to be V_{select}=350mV and $V_{no-select}$=170mV.

5.4.4 Supplementary Functions - Preset/Data Input Functions

Supplementary functions necessary for the circuit to function correctly are preset operations on the result MTJ, and data input operations on the data MTJs. As discussed in Section 5.4.1, the preset LOW operation on the result MTJ can be accomplished by applying a preset voltage to terminal V_4 while grounding the other terminals. This establishes a current in the circuit that is larger than the critical current of the device, and hence it writes a logic-0 to MTJ M4 by flowing from the free layer to the fixed layer. If, for some other computation, the MTJ is required to be preset to a logic-HIGH, then the current must flow in the opposite direction, i.e. from the fixed layer to the free layer. This is accomplished by applying the preset voltage at the terminals V_0 through V_3 and grounding terminal V_4, which reverses the direction of current.

In addition to this output preset operation, the circuit must also provide a way to

preset the input MTJs to the desired truth table. The design of the circuit presents an interesting structure to accomplish this operation. The circuit is very symmetrical in nature. Thus, the preset operation on MTJ M_4 is no different than that on any other MTJ. For example, when it is desired to set MTJ M_0 to a logic-LOW, the terminal V_0 becomes the voltage source, while the other terminals become the sink for the current. In order to the reverse the current so as to set MTJ M_4 to a logic HIGH, the role of the terminals is reversed. The symmetry of the circuit thus brings about the preset and data input operations in a similar way. This is a major advantage of the circuit, since this is an important function for accomplishing logic functions.

5.5 Going forward

The two circuits presented in this chapter are purely spintronic, in the fact that the logic processing is performed not through CMOS switch-based logic, but through spintronic means. In both circuits, the current in circuit is a function of spintronic input data and it directly manipulates the resultant spintronic state. This demonstrates the concept of circuits with compatible inputs and outputs, both spintronic in nature, while requiring electronic voltages for control (biasing voltages, in the analogy of electronics).

Chapter 6

Spintronic Processing in Memory

Logic gates with compatible inputs and outputs were introduced in the earlier chapter. With this MTJ-switch architecture, the logic operation of NAND can be performed on the operands stored in memory in a single logic step. However, useful computing tasks require more than just the ability of NAND operation. They also require the ability to cascade the output of one operation to the input of another, operations on outputs of different logical functions, complex operations, etc. While many of these tasks are achieved by including appropriate wires between devices in electronics-based computers and by composing smaller logic functions into larger ones, it is not obvious how to accomplish this with spintronics devices such as MTJs. For example, it is not yet clear how to design a circuit with MTJ devices such that it is able to perform a chain of logic functions.

This chapter proposes a scheme for multiple logic operations on spintronic data. Based on the spintronic logic circuits presented previously, we introduce the idea of *switch-based inter-memory device connection* that provides connectivity to an array of memory devices, in order to enable it to perform bit-wise logic operations on data that is stored within it.

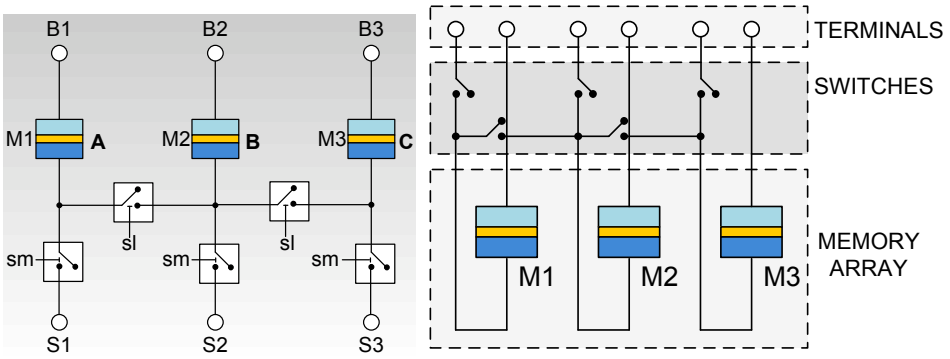
6.1 Spintronic Logic in Cache Architecture

With the capabilities of accomplishing both logic and memory, non-volatile devices like MTJs open up new opportunities to build logic-in-memory devices where both logic

and memory operations are performed by the same device. The area of logic-in-memory has been extensively researched before, by bringing logic and memory close to each other. Processor-in-memory systems [50, 49] consist of units of processor with a small amount of memory that each can access. A large data set is usually operated upon in a distributed way with computing units communicating with each other. However, the two functions of logic and memory are still provided by separate computing units, adhering to the von Neumann style of architecture. By adding interconnections to a spintronic memory unit such as a cache, a spintronic logic in cache (SLIC) unit performs both functions with the same devices. In the SLIC unit, the spintronic memory element itself is also the logic element, and the switch between the two operations is achieved by setting appropriate switches.

6.1.1 Spintronic Logic in Memory Circuit

With CMOS-based logic circuits, it is not advantageous to merge logic into memory circuits since the two circuits are dissimilar in nature and structure, thereby affecting the density of the memory array. Therefore, the electronics based logic-in-memory architectures bring the processor and memory physically close to each other, while the devices in each circuit perform their individual functions. However, with the MTJ-based logic and memory circuits, a closely integrated logic-and-memory unit can be designed due to the simplicity of the logic and memory circuits. With a triad of 1MTJ-1Switch memory circuit, we add two additional interconnections between the MTJ devices. These interconnections are provided with a 'logic' switch each, as shown in Fig. 6.1(a). The switches in the unit isolate the two functions of logic and memory from each other when any one function is being performed. The signals that control the function being performed as well as the voltages that enable the two functions are provided at the switch and at the biasing terminals respectively. The memory functions of read and write are enabled by using the switch memory control signal *sm*, while the logic functions of NAND, NOR, AND or OR are enabled using the switch logic control signal *sl*. When the switches controlled by *sm* are active, and those controlled by *sl* are inactive, the circuit becomes a three-bit 1MTJ-1Switch memory unit. On the other hand, when switches controlled by *sl* are active and those controlled by *sm* are inactive, the circuit becomes the 3-MTJ logic circuit described in the previous section. Thus, the unit is



(a) Simplified view (b) Modular view

Figure 6.1: Spintronic Logic In Memory circuit with two 'logic' switches and three 'memory' switches

capable of storing three bits and is programmable for upto four 1-bit logic operations. The interconnections necessary for this added logic functionality are uniform in their placement, and it is still possible for the circuit to retain its modular memory array structure as shown in Fig. 6.1(b).

6.1.2 Sensitivity to non-ideality of switches

The function of the switches in the circuit is to connect the devices in different ways so as to allow the circuit to take different configurations for the different functions. While ideal switches have zero on resistance, infinite off resistance and zero switching time, practical switches have a finite on and off resistance and switching time.

With the spintronic logic in memory circuit, we experimented with various finite switch resistance values to determine the sensitivity of the circuit to non-ideality of the switches. For this, the switch was modeled as a resistance that can take two values to indicate its closed or open condition. We used six switch settings of (closed,open) resistance, viz. (0,infinity), (50,100G), (100,10G), (500,1G), (1K,100M), and (5K,10M), and determined the bias voltages necessary at nodes B1,B2,B3,S1,S2,S3 (Figure 6.1(a)) in order to get the 3-unit MTJ circuit to perform four functions: 1. write A (logic-1); 2. write B (logic-0); 3. preset C (logic-0); 4. Nand(A,B). The switch conditions for the

four functions are shown in Table 6.1.

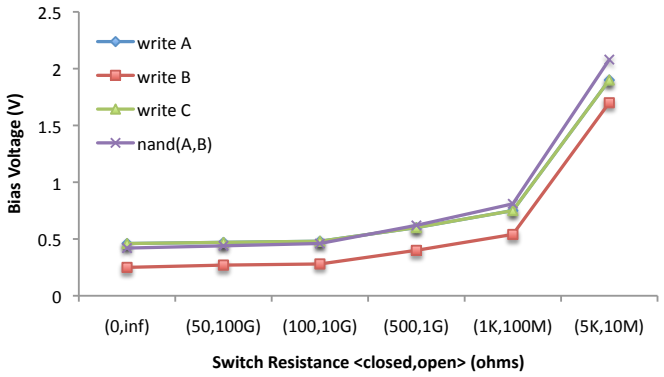


Figure 6.2: Bias voltages required for four operations using spintronic logic in memory circuit for different switch resistances

The spintronic circuit was simulated using a SPICE model of MTJs [51]. The device values used for the MTJ devices were based on the measurements of the MTJ device fabricated in [52]: R_{low}=900 Ω, R_{high} = 1900Ω, $J_C = 3MA/cm^2$, $I_C = 195uA$ assuming a rectangular device of size $130nm \times 50nm$. For different switch resistances, the resultant bias voltages required for the four operations are shown in Figure 6.2. These simulations show that the 3-MTJ unit is not very sensitive to variations in switch values from ideal up to the (1K,100M) switch setting. Beyond that, the rate of increase of bias voltages is high. This suggests that it is possible to use switches from existing technologies like CMOS, NEMS, etc. in order for the circuit to function correctly.

Table 6.1: Switch Conditions For Operations Using SLIC-L1 Circuit

operation	b1	b2	b3	s1	s2	s3	sm1	sm2	sm3	sl1	sl2
write A	0	0	0	Vm1	0	0	1	0	0	0	0
write B	0	Vm2	0	0	0	0	0	1	0	0	0
write C	0	0	Vm3	0	0	0	0	0	1	0	0
nand(A,B)	Vl	Vl	0	0	0	0	0	0	0	1	1

6.2 Spintronic Logic in Cache Unit Extensions for Bit-wise Logic Operations

With this MTJ-switch architecture, the logic operation of NAND can be performed on the operands stored in memory in a single logic step. However, useful computing tasks require more than just the ability of NAND operation. They also require the ability to cascade the output of one operation to the input of another, operations on outputs of different logical functions, complex operations, etc. While many of these tasks are achieved by including appropriate wires between devices in electronics-based computers and by composing smaller logic functions into larger ones, it is not obvious how to accomplish this with spintronics devices such as MTJs. With the proposed connectivity, we demonstrate these functions in an MTJ-based spintronic logic in cache (SLIC) unit. By extending the described spintronic logic in memory circuit to encompass more devices, we show that the logic operations of NAND, AND, OR and XOR can be performed on data present in the cache. This data is assumed to be brought into the cache from a larger dedicated memory unit and placed at locations suitable for a logic operation to be performed on it.

6.2.1 SLIC-L1 Unit: Demonstration of NAND operation

The logic operations of NAND can be performed in a single logic step by the unit in Fig. 6.1. This is called the SLIC-L1 unit ('L1' indicates that the number of logic operations = 1). MTJs M1 and M2 store the operands, and MTJ M3 stores the result of the NAND operation.

Fig. 6.3 shows the simulation of the NAND operation using the SLIC-L1 unit. In the first time step, operands and preset data is written to MTJs A, B and C. At this time, the memory switches *sm* are active. In the next time step, the SLIC-L1 unit switches to 'logic' mode and performs the NAND operation. The logic switches *sl* are active at this time. The states of MTJs for four input combinations of A and B are shown in the simulation.

The steps in which the NAND operation is accomplished can be converted into a 4-line program sequence for NAND:

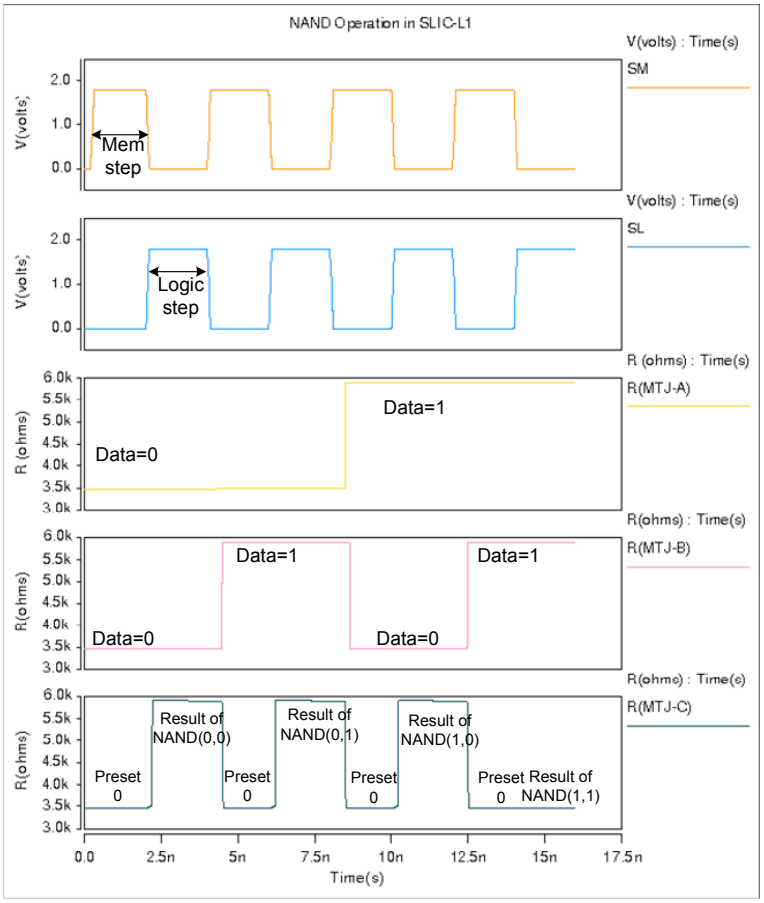


Figure 6.3: Four NAND operations with SLIC-L1

WRITE M1, A
WRITE M2, B
PRESET 0, M3
NAND M1, M2, M3

The first two lines indicate that the values of data operands A and B are first written to MTJs M1 and M2. The third line indicates that the MTJ M3 must be preset to logic-0. The last line indicates that a NAND operation must be performed on the data values as M3=M1 nand M2.

In order to qualify the nature of a logic operation, we can now define a 'footprint' for the algorithm of the operation. The footprint consists of five qualifers:

- The total number of steps (memory and logic) in the algorithm T;
- The number of 'writes' to the memory W;
- The number of presets required P;
- The number of logic steps in the logic operation L;
- Number of devices utilized N.

The footprint is thus described by a 5-tuple $\{T, W, P, L, N\}$. The NAND algorithm described above is characterized by the tuple {4,2,1,1,3}. By defining such a 'footprint', it becomes possible to quickly define an operation, understand its complexity and compare different algorithms that we will describe in the next sections. The significance of the footprint for insights about performance as well as design challenges will be discussed in Section 6.3.

6.2.2 SLIC-L2 Unit: Demonstration of AND operation

The AND operation is an example of cascading the output of one SLIC unit to another. The first unit performs a NAND function on its inputs, while the second one performs a NOT function on the output of the first unit. The NOT operation on an input can be performed as a NAND operation of the input with a logic-1. Since two logic operations (NAND and NOT) are executed in succession to complete the AND operation, this unit is called 'SLIC-L2'. In construction, it is just an extension of the SLIC-L1 unit. There are three control signals to this unit: the memory switch control sm, level-1 logic control $sl1$ and level-2 logic control $sl2$. At all times, only one of the three switches is active, and controls which operation is performed at a certain time.

Fig. 6.4 shows the mapping of the AND operation in memory. It requires five MTJs, with the result MTJ of SLIC-L1 unit as an input MTJ of the second SLIC-L1 unit, which is the cascading operation. Since the result of the first operation is not communicated instantly to the next logic gate as it is in an electronic circuit, it has to be enabled in a time-step manner. The switches $sl2$ isolate the second SLIC-L1 unit from the first when it is performing the NAND operation, while the switch $sl1$ between MTJs M2 and M3 isolate the second SLIC-L1 unit from the first when performing the NOT operation (i.e. NAND with logic '1').

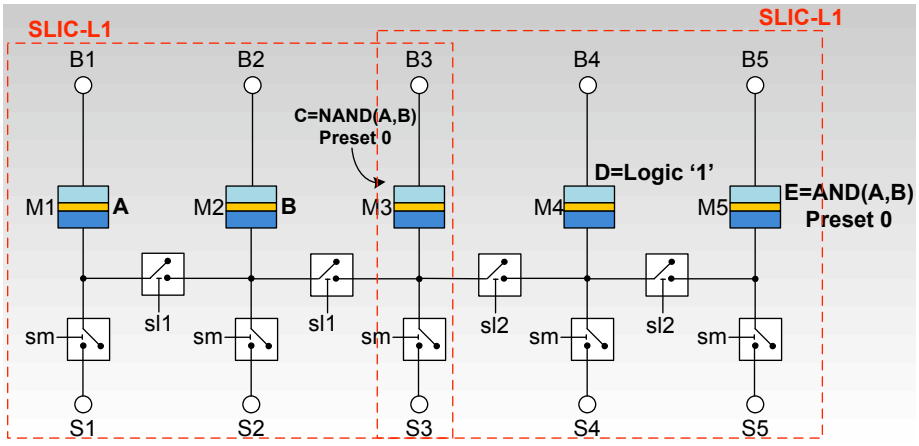


Figure 6.4: SLIC-L2 circuit performing AND operation in two logic steps

By activating the switches at different times, circuit isolation as well as cascading of the output is obtained. The switches used in this way also add a degree of control to the circuit, i.e. the propagation of the data to the next unit is a controlled operation. This results in an inherent synchronization in the operations of different logic units.

The program sequence for the AND operation is as follows:

WRITE M1, A
WRITE M2, B
PRESET 0, M3
PRESET 1, M4
PRESET 0, M5
NAND M1, M2, M3
NAND M3, M4, M5

From the above, the footprint of the AND algorithm is {7,2,3,2,5}.

6.2.3 SLIC-L3 Unit: Demonstration of OR operation

The OR operation is an example of a logic operation where the operation is performed on the outputs of two previous operations. In this, two different outputs must be connected

together to form the NAND circuit. This is different from a simple cascade as in the SLIC-L2 unit, but requires the same type of connectivity when data is placed in suitable locations. Fig. 6.5 shows the SLIC-L3 unit with the contents of each MTJ. It performs three NAND operations in total, hence it is called the *SLIC-L3* unit. However, the first two of the three logic operations can be performed simultaneously. Therefore, the final result is obtained in just two time-steps. In the first step, the complements $\bar{A}$ and $\bar{B}$ are obtained through two NOT operations, and in the second step, a NAND operation is performed on $\bar{A}$ and $\bar{B}$. The footprint of the OR operation is {9,2,5,2,7}.

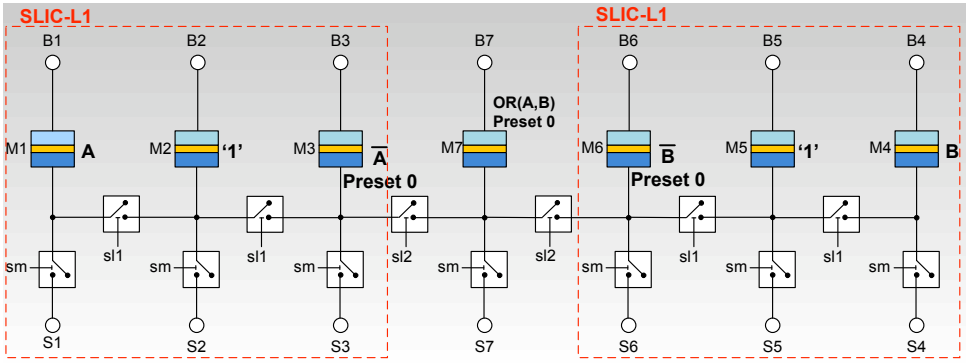


Figure 6.5: SLIC-L3 circuit performing OR operation with three NAND operations in two logic steps

6.2.4 SLIC-L2/L3 Unit: Demonstration of XOR operation

The SLIC L1, L2 and L3 are the basic units that address some of the primary requirements of logic functions, i.e. the NAND/NOR operation, cascading outputs and logic on multiple outputs. For complex functions, combinations of the three units may be required in some form. As an example, consider the XOR operation. Fig. 6.6 shows the switch locations required to accomplish the XOR function by combining the SLIC-L2 and SLIC-L3 units. The operation completes within three logic steps. With inputs A and B, the first step computes $\bar{A}$ and $\bar{B}$, the second computes $\overline{\bar{A}.B}$ and $\overline{A.\bar{B}}$, while the last logic step computes the final XOR result of A and B. Three switch controls $(sm, sl1, sl2)$ are necessary to make and break paths between the devices to provide the

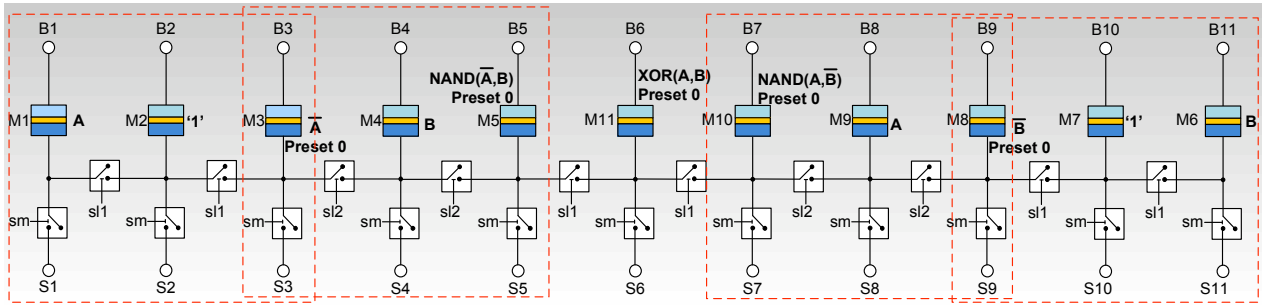


Figure 6.6: Combining SLIC L2 and L3 circuits to perform XOR operation in three logic steps

necessary connectivity and isolation.

In order to preserve the continuity of the logic operations until completion of the XOR operation, the operands A and B are each written into two different locations. This is a trade-off between redundancy of data and connectivity of MTJs within the unit. If the redundancy is undesirable, more connectivity between MTJs must be added. Fig. 6.7 shows an alternate mapping of the operands that does not make use of redundancy, thus reducing the number of devices necessary for the operation. The uniform placement of the switches in the circuit makes such a mapping possible. The footprint of the XOR algorithm in the first case is {14,4,7,3,11}, while in the second case, it is {12,2,7,3,9}.

6.2.5 The General SLIC unit

By combining the connectivity requirements of all the three units, a general Spintronic Logic in Cache unit can now be designed. This architecture shown in Fig. 6.8 consists of 11 switches for each five MTJs, i.e. an addition of an average of 2.2 switches per memory device. With this addition, an MTJ-based memory unit gets converted into a general-purpose spintronic logic-and-memory unit, capable of performing the basic logic operations as well as a cascade of logic operations in memory. To enhance the connectivity, more interconnections among the memory devices may be added. This design of the general unit shows the minimum interconnections required for the operations described till now.

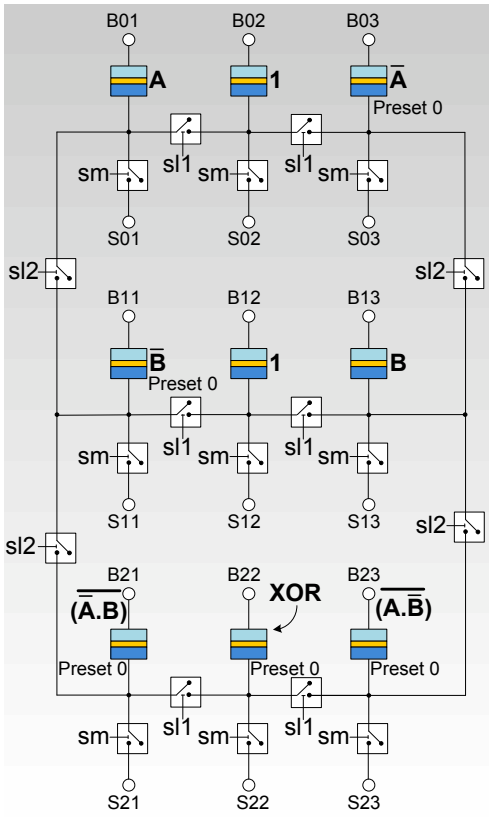


Figure 6.7: An alternate mapping for the XOR operation with reduced number of devices

6.3 Comparing the footprints of algorithms - Performance and Design Challenges

The footprint describes a logic algorithm in terms of its delay performance and device usage, and is able to give a quick insight into the 'complexity $O(n)$' of the algorithm. The first value T indicates the number of time steps in which the algorithm completes. This is an indication of the time delay of an operation and is desired to be as small as possible. The next three values in the tuple (W, P and L) represent the individual components of this delay.

The second value W is the number of data writes. If it is assumed that a memory

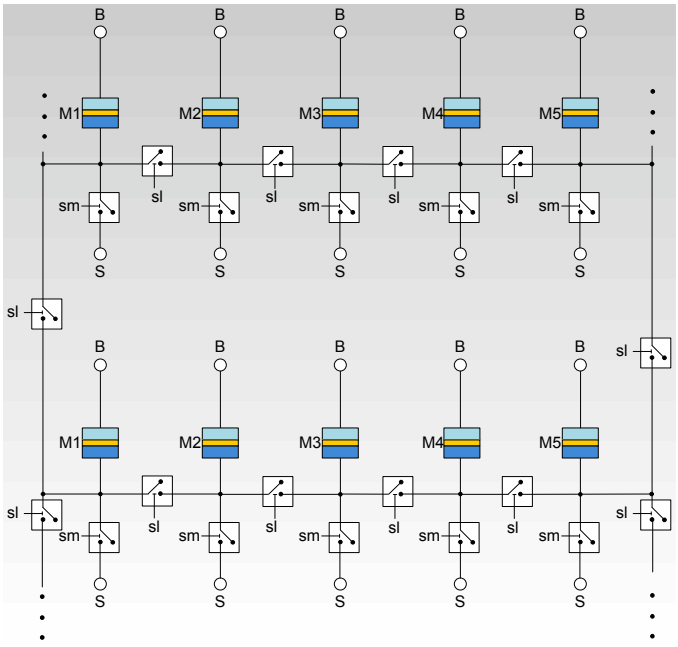


Figure 6.8: General SLIC architecture with about 2.2 switches per MTJ

transfer brings one operand into a cache in one transfer delay unit, then the total time taken for a logic operation on all the data in a cache is directly influenced by this number. The algorithms require an exact location mapping for the operands, and the final placement in the cache must be controlled by a controller. Due to this, the average delay of a data write in the spintronic unit may be higher than a data write in a normal cache.

The third value P is the number of preset operations. This required preset operation is a fundamental overhead incurred by the spintronic nature of computation in contrast to an electronic logic gate, and therefore must be minimized. The overhead of the delay of a preset operation may be addressed at the circuit level. For multi-bit, multi operand logic operations, the preset becomes a mass operation required to be performed on an array of MTJs. It may be possible to alleviate the time taken to perform this operation by spintronic methods. For example, it may be possible to use Field Induced Magnetization Switching (described in section **??**) on an array of MTJs by constructing

current planes above the array. This also allows the power of the preset operation to be shared by a number of devices.

The fourth value L describes the number of logic steps in the algorithm. This is the true delay of the logic operation itself. A complex logic operation is expected to consist of several basic logic steps.

The last value N is the number of MTJ devices that are necessary for a single bit logic operation. This directly influences the amount of parallelism that can be achieved for a certain operation for a fixed size of the spintronic cache. For example, when performing the NAND operation ($N = 3$) with a cache size of 128 bytes, the maximum amount of data that can be processed simultaneously is 42 bytes. However, for the AND operation ($N = 5$), only 25 bytes (maximum) can be processed in parallel. Therefore, for a logic algorithm, N must be minimized to gain the full advantage of the hardware parallelism. Fig. 6.9 shows the increase in the potential for parallelism available for increasing cache sizes for the four logic functions. With large cache sizes, this number can greatly exceed the data width of an SIMD-type of processor.

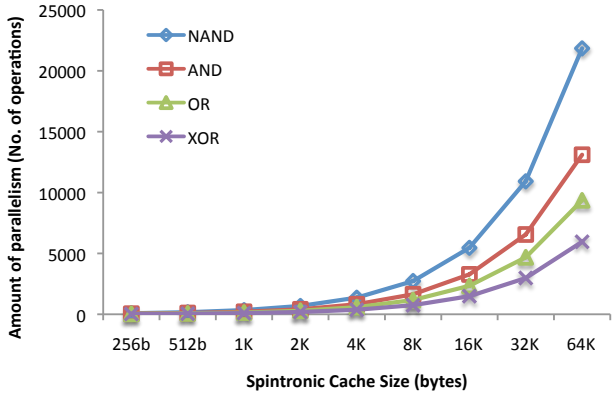


Figure 6.9: Number of logic operations that can be performed in parallel increases with the size of the cache

This value is also an indication of the temporary storage requirements for a logic operation, which is an overhead incurred due to the interconnections present in the unit as well as the spintronic nature of computation. For example, the NAND operation requires the minimum storage space in the cache, two devices for two operands and one

device for the result. In addition to this, the AND requires two temporary storage locations, one for the intermediate result of the first NAND operation, and another to store the data value '1' to perform the NOT operation. This is due to the spintronic nature of logic operation. On the other hand, the XOR operation requires more temporary or intermediate devices, which can be optimized by enhancing the connectivity of the cache as discussed in Section 6.2.4. Thus, improving the connectivity of the SLIC can decrease the temporary storage requirements of an algorithm.

6.4 SLIC simulation in the cache

As the logic algorithm from NAND to the XOR operation becomes more complex, the values of the terms in the footprints increase. For a fixed cache size, the delay of the operations increases from NAND to the XOR operation rapidly. In a practical machine, the delay is a function of the memory seek time and transfer time, cache line size, cache replacement algorithm, number of logic steps in algorithm itself etc. Further, the actual parallelism obtainable for a certain logic operation depends on the exact cache architecture and the mapping of the logic operation in the cache. Thus, a simple processor-cache-main memory system was designed and simulated to enable preliminary insights.

To implement the system at the granularity of logic operations, the Verilog hardware description language was used. The processor was implemented using the Very Small Processor Architecture (VeSPA) [53] Instruction Set Architecture (ISA), which is a simple and minimal ISA that allows a Verilog implementation for a processor in order to simulate its low-level operations so that the complexities of hardware implementation are exposed. The VeSPA ISA was enhanced with MUL and DIV instructions. To enable performing tasks with the SLIC cache, four new instructions were added to the SLIC processor: *slic-ld*, *slic-st*, *slic-preset* and *slic-nand*. At the abstraction of the architecture level, these SLIC extensions to the VeSPA ISA allowed the load, store, preset and NAND operations in the spintronic cache. The load and store instructions were designed to be location-specific instructions that load/store an entire array of MTJs with a continuous stream of operands during a memory transfer. The implementation details of these SLIC extension instructions are given in Appendix **??**.

6.4.1 Varying the SLIC cache size

In order to observe the delay of the four logic algorithms in a practical system, the architecture shown in Fig. 6.10 was simulated. It consists of a spintronic cache unit that conforms to the general SLIC unit architecture. The delay of the cache load-store instructions was modeled as a function of the number of bytes to be loaded(stored) from(into) the main memory, and the main memory delay. The main memory delay was modeled as the sum of the memory access time (that includes the seek and rotational latency) and the transfer time. Fig. 6.10 shows the values of the parameters used for the simulations. The *slic-preset* operation was assumed to preset the SLIC MTJs at the rate of 8 bytes per cycle, consistent with the main memory transfer rate.

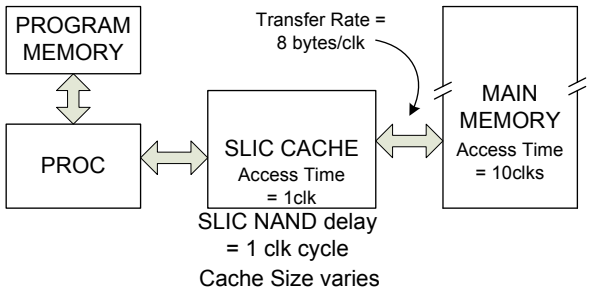


Figure 6.10: Simulated Architectural Model of SLIC System

The delay of the logic operations of NOT, AND, OR and XOR was measured with four values of SLIC cache size, from 1kB to 8kB. Fig. 6.11 shows the trend of increasing delay as the complexity of the logic operation increases. For a fixed cache size, the benefits of parallelism decrease as the logic operations become more complex, resulting in higher delays. From Fig. 6.11 there is an exponential increase in the delay from the NOT to the XOR operation. The results also show that as cache size increases, there is more performance improvement due to increased parallelism for the XOR operation than for the NOT operation. This is because though the parallelism increases, the delay of loading the larger cache with operands, as well as the delay of performing preset operations on the larger cache, increase in almost the same proportion. This may be addressed by a higher bandwidth between the cache and the memory (memory transfer

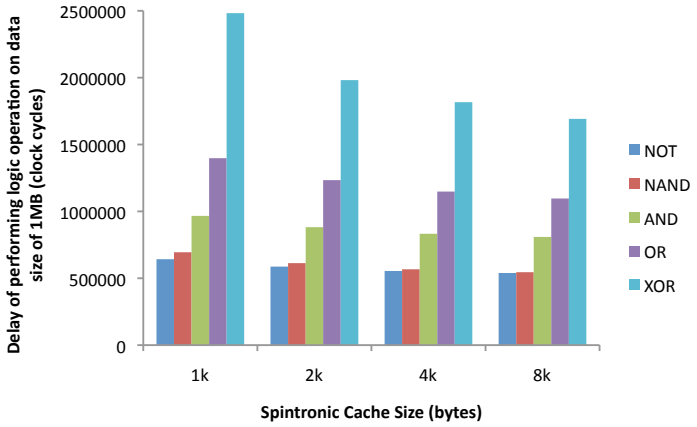


Figure 6.11: Delay of performing a logic operation on 1MB of data with varying SLIC cache sizes

rate) and a faster preset operation on the cache (increasing our simulation parameter of 8 bytes per clock).

6.4.2 Microbenchmarks

SLIC exhibits a parallel hardware configuration. As a starting point, the performance benefits of such a cache must be evaluated for algorithms that are parallel in nature, operate on stored data, and have no dependent tasks, i.e. the processor does not wait for the results of the operation. The third condition is imposed so that the current speed of the spintronic operation does not affect other instructions which can continue to execute in a superscalar processor. Ultimately, the processor may be able to delegate simple tasks entirely to the SLIC controller which will work on the data set, keeping the processor free for more complex tasks.

In order to evaluate the potential of the performance of SLIC-L1 and SLIC-L2 over a classic processor, two microbenchmarks were created:

1. Image Negation: This algorithm reads an image pixel by pixel, performs a bitwise NOT and stores it in a different location. The classic processor is provided with the base address of the original image and result location. It reads an image pixel, negates it and stores it back. The SLIC processor is also provided with the base address of the

original image and the result location. It reads a set of pixels into a SLIC-L1 cache, performs a bitwise NAND of image pixel with 1ś to generate the complement, and then stores it back.

2. AND Masking: This algorithm reads an image and a mask pixel-by-pixel, performs a bitwise AND on them and stores the result in a third location. The classic processor reads both operands, ANDs them together in the processor and stores the result back. The SLIC processor reads a set of image pixels and mask pixels. Since AND involves a NAND followed by a NOT, it utilizes the SLIC-L2 cache, with C, D and E locations preset to '1'. It reads the operands in locations A and B, and proceeds through two levels of NAND to obtain the AND result in location E. Once the results are generated, they are stored back in result image location.

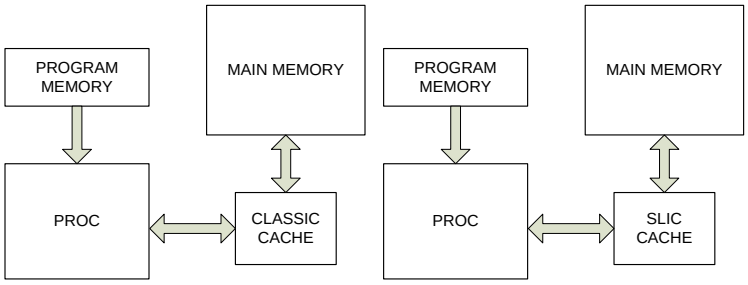
These microbenchmarks are inherently data-centric algorithms. The data is available in memory and needs to be brought into the cache first and then the computing unit for a basic logic operation. In the future, if the data is expected to be present in a large MTJ-based memory (enhanced with logic capability) itself, then transferring it to a CMOS-based computing unit will become unnecessary. However, for first experimentation, the goal in using these microbenchmarks is to determine the effect of performing operations in *small parallel* sets in an MTJ-based cache and determining the extent of performance benefits obtained in the presence of the overhead of switching delays and presetting of data.

6.4.3 Simulated system for the microbenchmarks

We compared the performance of the two microbenchmarks for the Verilog implementation of the VeSPA processor described earlier, with a two level memory hierarchy that consisted of one level of cache and a main memory. The block diagram of the classic and SLIC architectures is shown again for comparison in Fig. 6.12.

Processor

For these simulations, a non-pipelined VeSPA ISA, enhanced with MUL and DIV instructions was used to implement the classic processor. The same ISA with SLIC extensions was used to implement the SLIC processor.



(a) Classic System (b) SLIC system

Figure 6.12: Block diagram of architectures simulated

Memory

The memory hierarchy in both cases comprised of a main memory of size 8kB, and a cache. For the classic processor, a classic 4-way set-associative cache with LRU replacement policy was implemented. For SLIC processor, a SLIC cache was implemented. The first microbenchmark used SLIC-L1 cache, while the second microbenchmark used SLIC-L2 cache.

Evaluation Models and Results

For evaluation of a cache capable of performing simple logical operations, a parameterized area and delay model was used. The purpose of the models was to examine the potential of such a structure for use in complex processors. Analyzing accurately the performance metrics of area and delay due to SLIC can best be achieved by implementing a chip with SLIC. However given the current trend of research in MTJs, such a system would be inflexible to accommodate the rapid advances being made and yet possible in this progressive field. Hence, we will instead proceed to make an estimate of the area and delay required and draw conclusions as to the extent of performance advantages with the SLIC concept.

Delay Model

The delay of an operation comprises of the delay of the processor, cache and the main memory. Seven parameters were used to describe the delay of any instruction, as shown in Table 6.2. An arithmetic instruction would take time 'g' units, a memory instruction with a cache hit would take (g+cx) time units, while a memory instruction with a cache miss would take (g+cx+a+bx) time units. Additional simulation parameters used were size of the cache (in bytes) and size of image (in pixels, with one pixel=one word=4 bytes). In all simulations, one clock cycle forms one time unit.

The effect of these nine parameters on the SLIC architecture was observed. By varying the value of one parameter at a time, the sensitivity of execution time to the changes was desired to be determined. These experiments give the performance trends for a single parameter, but do not illustrate the factors that have the potential to form bottlenecks in a SLIC system. Hence, in addition to evaluating the performance benefits of such an architecture, to determine the most important parameters in the architecture, a Plackett and Burman analysis [54] was also performed on the SLIC architecture.

Table 6.2: Parameters of the Delay Model

Main Memory delay	a+bx	a=setup time, b=time taken for transfer per byte, x=num of bytes
Cache Memory delay	cx	c=transfer time per word, x=num of words
SLIC Switch Set Time	dx	d=Setting time per unit size of cache, x=Cache Size Unit, set to 1byte.
SLIC Operation Time	e	e=Time taken for a spintronic operation
SLIC Switch Unset Time	fx	f=Time taken to unset the switches; we assume d=f in this simulation; x=Cache size unit, set to 1byte.
Processor delay	gx	g=Time per instruction; x=num of instructions

Delay Performance

In every experiment, the execution time was measured by varying values of a parameter while keeping other parameters constant. The values used for every parameter are shown in Table 6.3, with the underlined values being the default values used for experiments with other parameters. The time values selected are relative to the processor clock. For all parameters, small values were used so as to gain a preliminary insight about the

impact of using SLIC, on execution time. The performance obtained after varying one parameter at a time is shown in Fig. 6.13.

Table 6.3: Simulation Values of Delay Parameters

Parameter	Variable	Values Taken (clk cycles)
Main Memory Setup Time	a	5, **10**, 15
Main Memory Transfer Time	b	0.125, **0.25**, 0.5
Cache Memory R/W Time	c	0, 1, **2**, 5, 10
SLIC Switch Set Time	d	2, **5**, 10
SLIC Operation Time	e	1, **2**, 5
SLIC Switch Unset Time	f	2, **5**, 10
Processor Time	g	0, **1**
Size of Cache (bytes)	h	32, 64, **128**, 512, 1K
Size of image (pixels)	i	32, **64**, 512, 1K, 2K

The slopes of the graphs in the figure indicate the trends of performance with respect to each parameter. The SLIC cache is relatively insensitive to changes in main memory setup and transfer time, processor delay and SLIC operation time. The changes in these four parameters have little impact on the execution time, due to the parallelism of the operation in the SLIC cache. On the other hand, the classic cache which does not have a high degree of parallelism suffers higher performance variation.

The sensitivity of the performance of SLIC cache to the memory R/W time and switch setting time can be observed from the graphs. These two parameters define the primary overheads in the SLIC concept, brought in due to the data storage and preset requirements. Hence they are desired to be kept as minimal as possible through use of fast switches and fast memory operations.

For increasing image size, one expects to get better performance regardless of underlying cache technology. The execution time for increasing image size shows that the performance returns are much better with the SLIC architecture (for both L1 and L2) than with a classic cache. The execution time does not grow as quickly with the SLIC cache with increase in image size as it does for the classical processor. Even for image sizes as small as 32 pixels, the SLIC cache performs faster than its classic counterpart, showing that the overheads of presetting and switching do not pose a problem with the selected simulation values of other parameters. Again, this gain of performance is attributed to the parallelism available in the SLIC cache. For increasing cache size also,

the SLIC cache executes the algorithms faster than classic cache.

Overall, these graphs show that when MRAM performance is demonstrated to be faster than current cache structures, and switches with fast switching time are identified, a cache with SLIC architecture would enhance performance of algorithms that perform logic operations on a large amount of data. Through the Plackett and Burman analysis, we were able to take into account the interactions of parameters, and rank them. The order obtained can be classified into five ranks based on their numerical effects as shown in Table 6.4. The order of parameters remains more or less the same for both types of SLIC caches. As expected for the algorithms, the image size dominates the performance, followed by the cache R/W time and main memory access time. As technology advances, it may be expected that the cache R/W time will decrease, making it less of a bottleneck. The SLIC switch and logic operation parameters affect the performance, yet are not the most significant factors. Therefore for large image sizes, the SLIC cache can perform better than the classical cache for these algorithms.

Table 6.4: Plackett & Burman ranking of simulation parameters of the delay model

Rank	SLIC-L1	SLIC-L2
1	i	i
2	a,c	a,c
3	d,e,f	d,e,f
4	b,h	h
5	g	b,g

6.5 More complex logic function - ADD

While the logic operations demonstrated have been simple functions, they demonstrate the feasibility aspect of logic in memory. In order to 'program' more complex functions in the memory, an appropriate cache-mapping algorithm that includes the switch control must be written and initial data must be moved suitably. As an example, we will demonstrate the logic function of half addition (Sum=A xor B, and C=A and B). Half adders form a critical component in n-bit adder algorithms like the Kogge-Stone, Sklansky and Brent-Kung adders, providing the first step of computation of n-bit addition. The functions of AND and XOR have been previously demonstrated and are performed

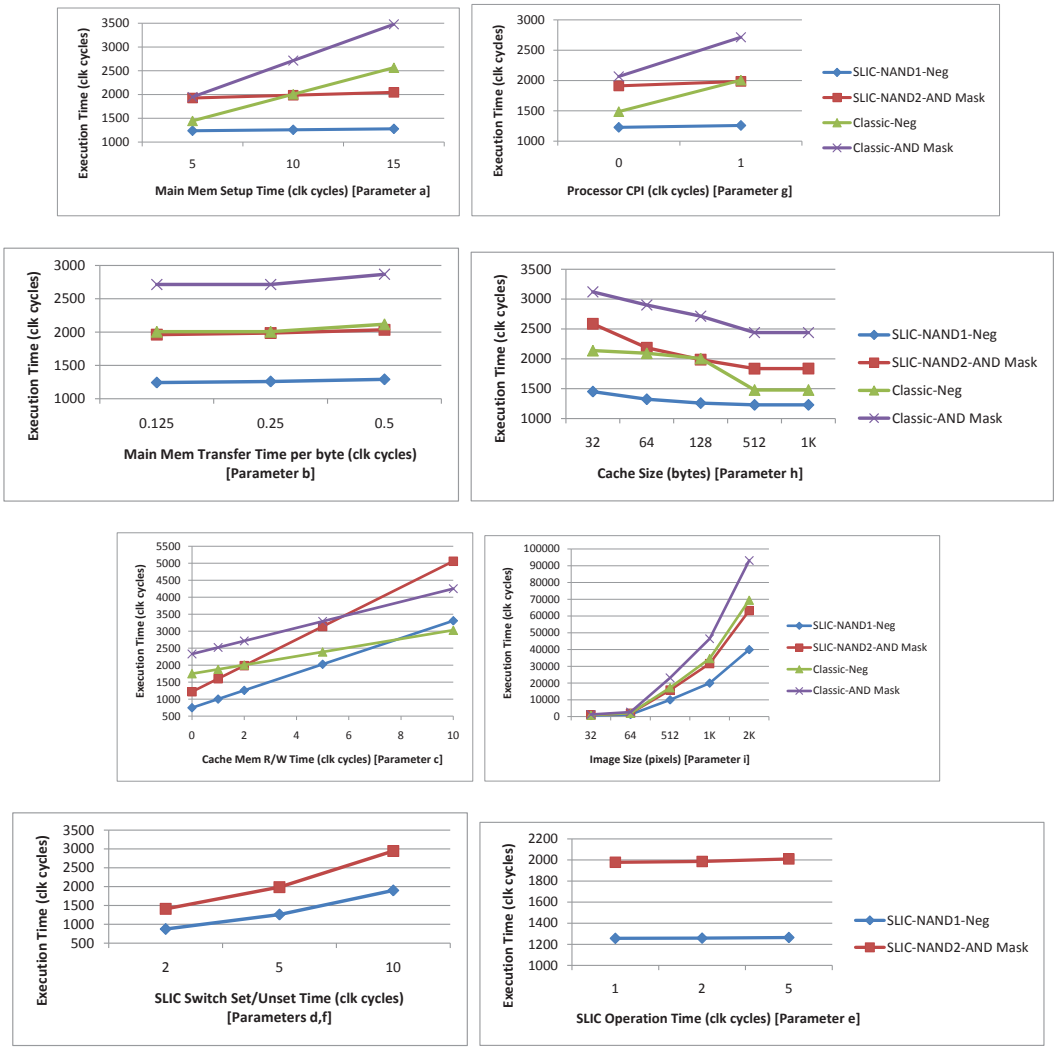


Figure 6.13: Execution Time of microbenchmarks after varying values of parameters, one at a time

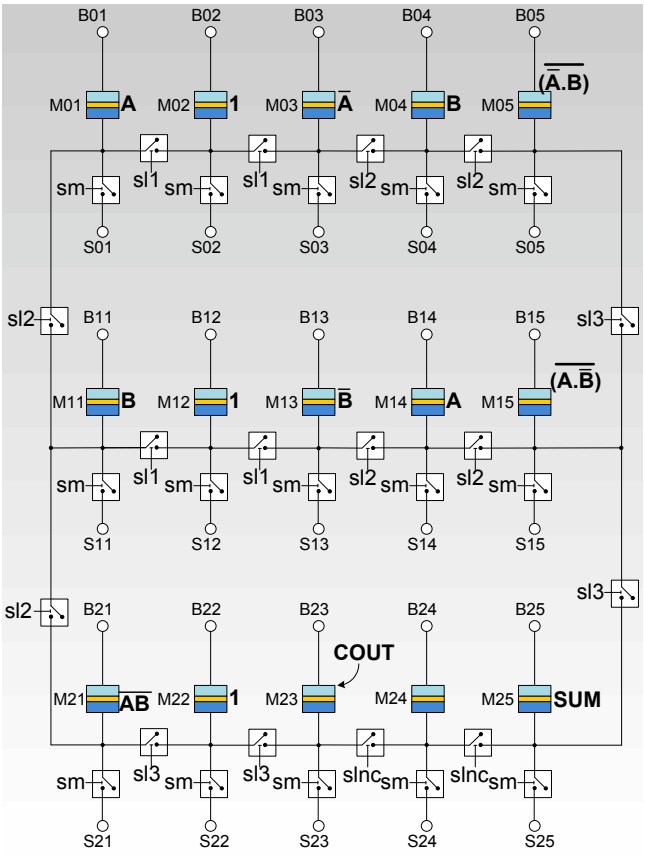


Figure 6.14: ADD function mapped onto general SLIC unit

together to accomplish the ADD operation. Fig. 6.14 shows the location mapping for the ADD function using the generic unit. The ADD algorithm requires 14 MTJ devices spanning three rows, with the operands A and B stored twice (A in MTJs M01 and M14, and B in MTJs M04 and M13). The operation completes within three logic steps. The program steps are described below.

WRITE A in M01,M14
WRITE B in M04,M11
PRESET M03,M05,M13,M15,M21,M23,M25 to 0
PRESET M02, M12, M22 to 1
NAND operation on (M01, M02, M03);

(M11,M12,M13) //$\bar{A}$ and $\bar{B}$ are obtained.

NAND operation on (M01, M11, M21); (M03,M04,M05);

(M13,M14,M15) //$\overline{AB}$, $\overline{\bar{A}.B}$ and $\overline{A.\bar{B}}$ are obtained.

NAND operation on (M21, M22, M23); (M05,M15,M25)

//COUT and SUM are obtained.

About area requirements

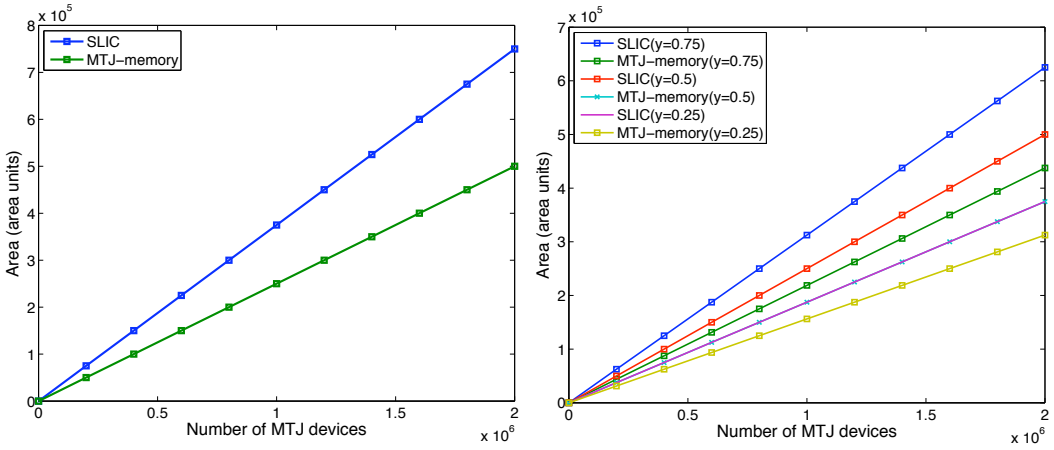
Additional area due to SLIC concept is due to switches in the memory, additional voltage and ground lines required for varying input voltages and added complexity of controller.

In a 3-device SLIC-L1 unit, there are three magnetic tunnel junction devices and five switches. Assuming that m=size of MTJ device, s=size of switch, w=area added due to additional routing requirements, a 3-device unit will roughly requires (3m+5s+w) area. If a memory array must be fabricated in powers of 2 and extended in two dimensions, then one can expect a 4-device unit in that array to take (4m+8s+w) area, with an MTJ having one memory and one logic switch each. To get an idea of the increase in total area due to switches, let switches require an area of factor y over the area of the device itself. i.e. s=my. Then, the 4-device memory unit takes 4m(1+2y)+w area. In contrast, a 4-MTJ classic cache would require (4m+4s)=4m(1+y) area.

The exact same calculations hold for a SLIC-L2 and a SLIC-L3 unit, except that the smallest unit must be at least 8-devices instead of four. Thus, a SLIC-L2 and a SLIC-L3 unit would require 8m(1+2y) area. For simplicity, assuming that the size of the switch is equal to the size of the MTJ device (i.e. y=1), as SLIC cache size increases, the area will increase as shown in Fig. 6.15(a). For a SLIC cache, if switches are chosen from a technology different from MTJs, then the device and switch sizes are not expected to be equal. As technology advances, it can be expected that m decreases and y becomes lesser than 1. As switch area decreases with respect to device area, the slope of growth of area decreases, shown at-a-glance in Fig.6.15(b).

6.6 Going forward

This section explains the concept of spintronic based logic in memory, a possible implementation and the potential advantages for accomplishing simple operations on large



(a) Switch area = Device Area (y=1)

(b) Varying y

Figure 6.15: Rate of growth of area of SLIC caches vs classic MTJ caches

data. It attempts to establish the feasibility of a true logic-and-memory scheme in a non-von Neumann architecture style. However, a number of studies and schemes will have to be designed to really benefit from the capability of memory and logic in a single device. For eg. general computation requires more than just basic operations. Electronic circuits compose naturally to perform complex operations with speed and efficiency. The spintronic method described here does not compose naturally, and a more elegant method, either at architecture-level or circuit-level must be designed. Then, general data requires interaction with other data that may not be stored physically closed to it. Also, the number of data items that a computation may require can be very high. Techniques to enable this computation in the logic-and-memory module must be devised.

Most previous work that has proposed processor in memory ideas have adhered to the von Neumann style of architecture. This means the processor and the memory units have been separated, and the distance between them varies. However, the ideas proposed in this chapter can largely benefit from insights from the extensive prior processor in memory work. For eg: the IRAM project [49] adds a vector processor in the DRAM memory unit. This takes advantage of the parallel processing and high bandwidth made

possible by adding a processor closer to the memory. Graphics accelerators have used the concept for a long time. Another related idea is that of SIMD processing, where a large amount of data is processed in a single step, much like vector processing. In general, huge potential performance benefits have been observed with such logic-in-memory architectures for data-centric applications [50]. Learning from this previous work, the concept of the SLIC must be further improved and evaluated for real-time data-intensive benchmarks.

Chapter 7

Spintronic Arithmetic and Logic Unit

One design style for MTJ devices is based on resistance difference between devices (called as *Resistive Design Technique* in the rest of this thesis) and is discussed briefly in chapter 5. It utilizes sense amplifiers to amplify the resistive difference in MTJ states into a voltage logic signal. The main challenge with this design technique is its *composability*. This is an important ability in circuits that enables the step from simple logic gates to complex logic blocks. Typical electronic circuits compose into larger blocks mainly due to their input-output compatibility. However, it is not yet clear how to compose circuits with resistive devices such as MTJs. This chapter addresses this in MTJ-based resistive circuits. To enable composability in these circuits, a novel design technique called 'union with neutralization' will be described, to combine individual component designs into multi-functional units. Further, individual functions of addition, subtraction and logical operations are first designed using the resistive design technique, and then combined into a single arithmetic and logical unit (ALU) using the proposed design technique. For the sake of simplicity, the MTJs are assumed to be operated by the FIMS technique. The extension of the proposed design technique for MTJs operated by CIMS-based MTJs is discussed later in the chapter.

7.1 Description of MTJ design techniques

7.1.1 More on the Resistive Design Technique

The use of Resistive Design Technique to create circuits using MTJs was discussed in [41]. In this technique, one or more MTJs are connected in series to positive and negative terminals of a sense amplifier as shown in Fig. 7.2. The circuit produces output in two time steps: logic step and output generation. In the logic step, all MTJ resistance states are set according to their respective inputs. During output generation, a small sense current is passed through the MTJs to generate a small voltage whose magnitude is based on the combined resistance of MTJs connected to a terminal. The sense amplifier then amplifies the voltage difference between the positive and negative terminals and outputs a voltage $0V$ if the resistances R_1 and R_2 are equal and $5V$ if R_1 is greater than R_2.

Following this technique, a design for a full adder using MTJs was given by Hao Meng et al [55]. This adder design used seven MTJs and two sense amplifiers. Another design used MTJs to construct a 3-bit gray counter [56]. This used ten MTJs and three sense amplifiers. Both designs take inputs in the form of currents $+I$ and -I, and produces output in the form of voltage, as $0V$ for logic 0 and $5V$ for logic 1.

7.1.2 Inter-device communication using Spin Torque Transfer (STT)

In order to transfer data between MTJs, Wang et al. [57] developed a novel nano-channel design that enables a fan-out function for MTJs using the principle of spin torque transfer. The set-up for performing spin torque transfer is shown in Fig. 7.1. Two MTJs are physically connected with a magnetic nano-channel that connects their free layers. The specially designed nano-channel isolates the MTJs and they can be operated individually during normal operation. When current is introduced in the device so that it flows in through terminal I_{talk} and out from I'_{talk}, it polarizes the free layer of MTJ B in the same direction as that of MTJ A. If the pinned layers of both MTJs are oriented in the same direction, then both MTJs will have the same resistance state after the spin torque transfer.

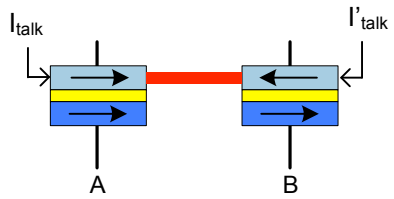


Figure 7.1: Direct communication between MTJs using spin torque transfer

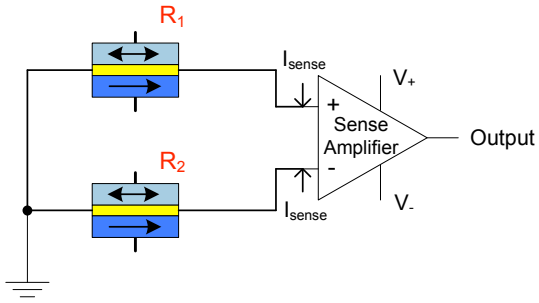


Figure 7.2: General MTJ set-up for a logic function

7.2 Design of ALU components

The design of the ALU will be shown in parts. All components are constructed so that there is no need to preset the resistance state of any MTJ before operation. Fig. 7.3 shows the design of a 1-bit subtractor with its truth table.

When inputs A, B and C (Borrow-in) are applied, resistance states R_1 through R_6 are set. To generate an output, sensing current I_{sense} is passed through the circuit as shown. A sense amplifier amplifies the difference $[(R_1+R_2+R_3)-(R_4+R_5+R_6)]$ and outputs $0V$, i.e. logic-0 when $R_1+R_2+R_3=R_4+R_5+R_6$. When $R_1+R_2+R_3>R_4+R_5+R_6$, output is V_+, i.e. logic-1. Another sense amplifier produces the logical output of the function $[(R_1+R_2)-(R_4+R_5)]$. This is equivalent to the function B_{out}. In the truth table for the subtractor, the resistance values are written as $R_1R_2R_3$ instead of their sum for clarity.

Figures 7.4, 7.5 and 7.6 shows the design of 1-bit logical units that perform the logical functions of AND, OR, NAND, NOR, XOR and XNOR separately. In Fig. 7.4 and Fig. 7.5, the *Ctrl* input controls the logic operation of circuit. NAND/NOR gates can be implemented in two different ways as shown.

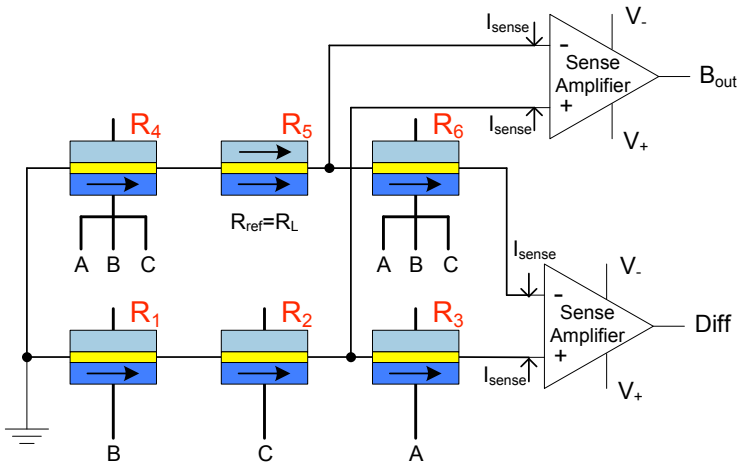


Figure 7.3: 1-bit spintronic subtractor using MTJs and its truth table

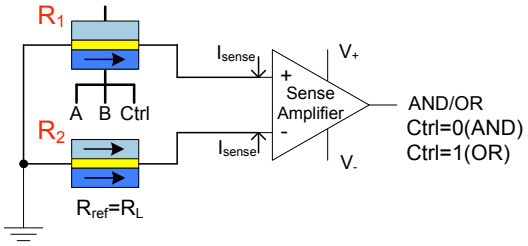
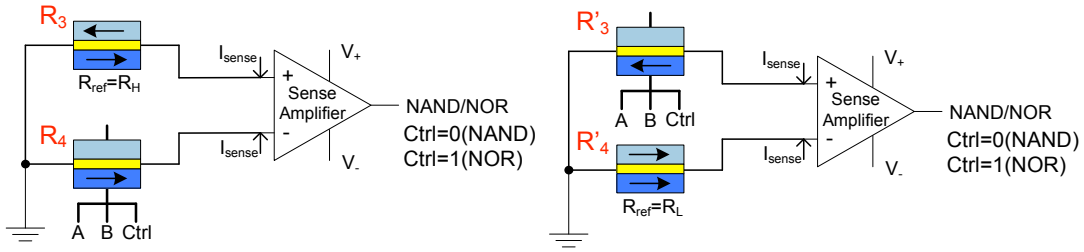


Figure 7.4: AND/OR gate using MTJs and truth table

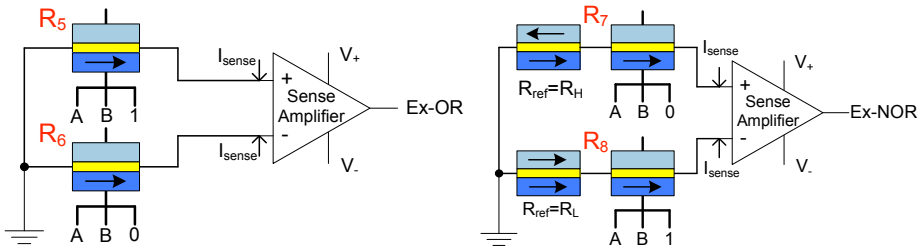
7.3 Combining logic functions into larger components

To follow a bottom-up approach to practical system design, it is desired that smaller units have the potential to combine into larger ones. In TMR-based circuits, since output depends on the state of every device connected in series, it is not always possible to combine two circuits directly. In cases where it *is* possible to construct meaningful circuits from direct combination, the sensing logic that produces output presents certain challenges for further system design. In this section, we discuss the issues that arise due to such direct combination and present the concept of our proposed solution to these problems.



(a) NAND/NOR gate using MTJs and truth table (b) NAND/NOR gate using MTJs in another configuration

Figure 7.5: NAND/OR gate using MTJs and truth table



(a) XOR gate using MTJs and truth table (b) XNOR gate using MTJs and truth table

Figure 7.6: XOR and XNOR logic functions using MTJs

7.3.1 Direct combination

An adder/subtractor may be designed by direct combination of a full adder and a full subtractor. Fig. 7.7 shows a 1-bit adder/subtractor unit constructed in this manner.

This design uses seven MTJs and three sense amplifiers. However, using this design in a practical circuit presents certain difficulties: The design should incorporate control inputs to ensure that addition is performed for a certain set of control inputs, and subtraction for another. Extending this design to an n-bit adder/subtractor will need switches that choose between C_{out} and B_{out} to connect to the C-input of next MTJ, according to desired operation of add or subtract. This shortcoming in design is overcome by the following approach.

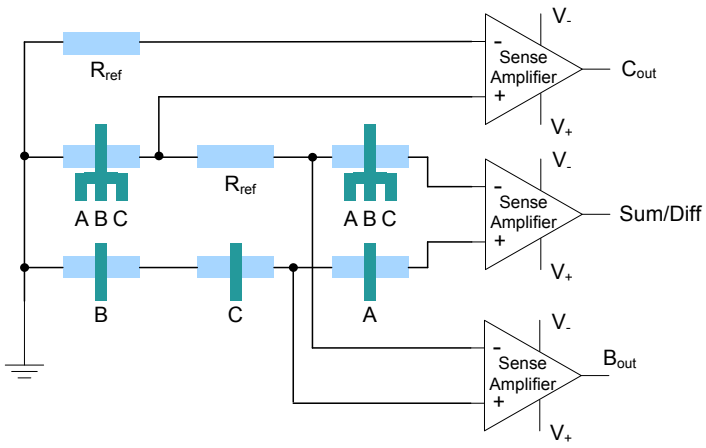


Figure 7.7: 1-bit spintronic adder/subtractor constructed from direct combination of full adder and full subtractor design

7.3.2 Concept of Union with neutralization

Fig. 7.8 shows the schematic of a modified design that incorporates control inputs, and generates the outputs C_{out} and B_{out} on a single output line to enable extension to n bits. Ideal sense amplifiers output a $0V$ or $5V$ as a result of the logic functions. However, by deliberately mismatching the transistors in the sense amplifier so that it has a small, positive V_{offset}, a difference of zero will be considered to be negative difference. Thus, when the two resistance branches are perfectly matched, the desired logic output is zero and the quantity $|V_+ - V_-|$ will be zero but the amplifier will saturate to V_-. When resistance branch connected to V_+ has higher resistance, the amplifier will saturate to V_+. These voltages are bias voltages and can be set so as to output the correct value of input current ($+I$ and -I) necessary for the operation of an MTJ. Thus, 1-bit adder/subtractor units can be connected together so that the output C_{out} is applied directly to the input C of the next adder, to get an n-bit adder/subtractor.

The 1-bit design uses twelve MTJ devices and two sense amplifiers. An n-bit design will use 12n MTJs and 2n sense amplifiers. Each unit requires three control steps in the sequence: logic step, spin torque transfer step and output generation. The third step in i-th unit overlaps with logic step in (i+1)-th unit. Thus the design operates in exactly 2n+1 time units.

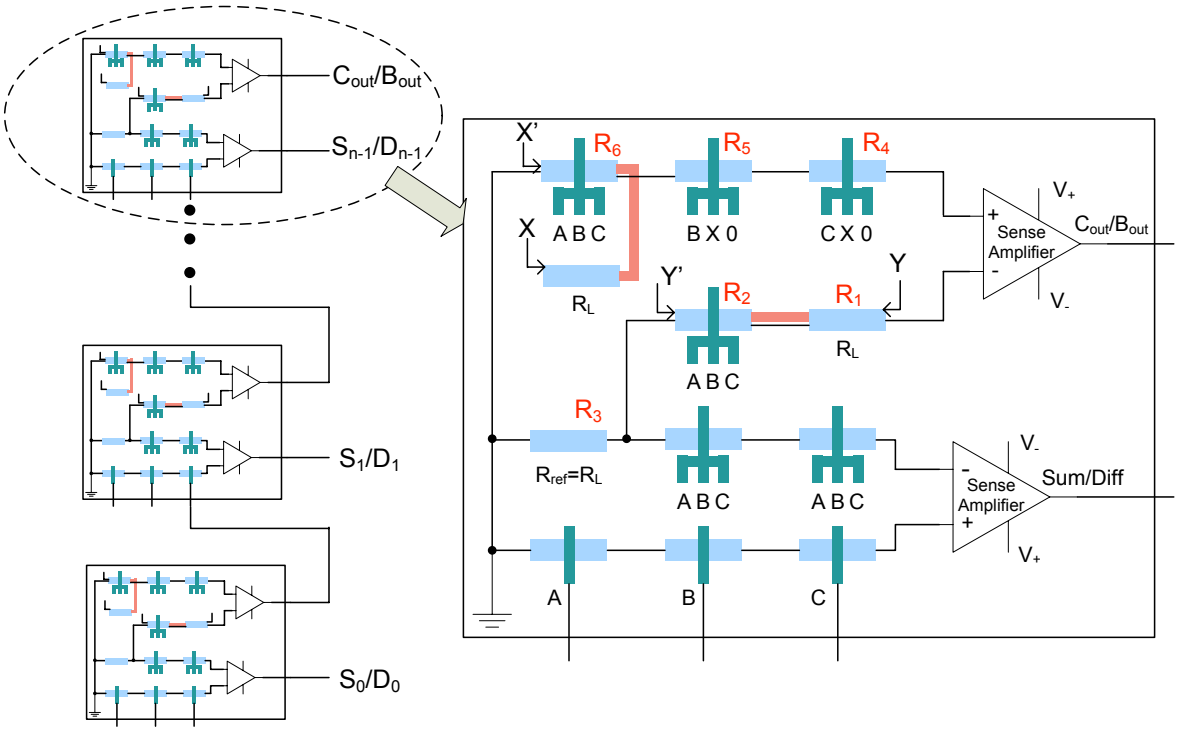
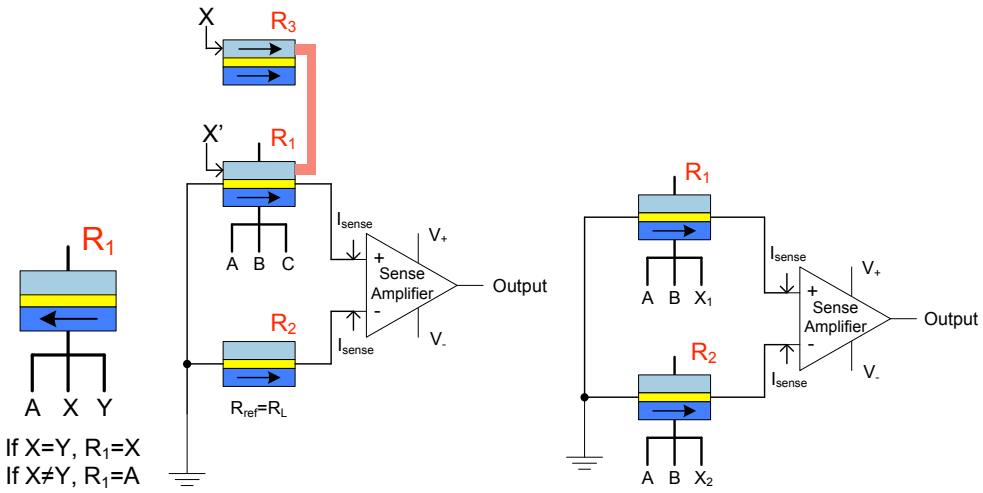


Figure 7.8: n-bit spintronic adder/subtractor using firld-driven MTJs

The modified design uses a technique called 'union with neutralization' for combining logic functions into a single unit. In general, two separate MTJ-based units when combined can give a truth table completely different from desired truth table or either of their truth tables. Consider the design for AND gate and NAND gate as shown in Fig. 7.4 and 7.5(b). If combined directly, i.e. R_1 and R_3' connected in series to positive terminal of sense amplifiers and R_2, R_4' connected in series to negative terminal, then the output is given by the expression $[(AandB) + (AnandB)] - [L + L]$ which produces an output of logic-1 for all input values, which is neither the truth table for AND nor NAND! However, the combination circuit will operate as an AND gate if the NAND part of the circuit behaves as if it were not present in the circuit, and it will operate as a NAND gate when AND part of the circuit seems to be switched off. This is called neutralization. Thus, neutralization is the method of eliminating the contribution of a part of a circuit to the output. It is done by equalizing the resistance states of two MTJs connected to opposite terminals of a sense amplifier, so that the logical



(a) Neutralization using control input(s)

(b) Neutralization using spin torque transfer

(c) Neutralization through logic

Figure 7.9: Neutralization techniques

difference between them is zero. Given two MTJ-oriented logic designs, combining them into a single unit can be achieved by introducing control inputs in the design, that neutralize all MTJs associated with one logic operation while the other is in effect. This technique is called as union through neutralization. We describe three ways of performing neutralization.

Neutralization using control input:

The state of an MTJ that uses a single input can be easily controlled by adding two control inputs to it (X and Y), or a control input and a static input to it (Y being a predefined input) . Fig. 7.9(a) shows an implementation of this method. If Y was always set to be logic-0, when the control input is low, the output of the MTJ is low. When control input is high, the output follows input A. Setting Y to logic-1 will make R_1 produce a high output on neutralization, when control input is low.

Neutralization using spin torque transfer:

MTJs that have more than one input cannot be easily neutralized using a control input. In such a case, additional MTJs can be used to perform the same function as MTJs connected to the opposite terminal of sense amplifier. Then using spin-torque transfer technique, this state can be transferred to MTJs, so that MTJs on both sides are neutralized. Fig. 7.9(b) shows an example using this technique. When X is 0, results of normal logic operation are output by the sense amplifier. When $X = 1$, state of R_3 is transferred to R_1 and the two resistances are equalized (both low). Thus, the output is zero.

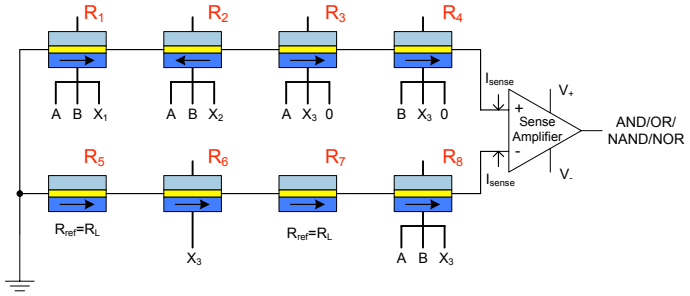
Neutralization through logic

This method works for specific logic functions only. It can be implemented through observation of logic function. Fig. 7.9(c) shows the use of this technique. When $X1 = 1$ and $X2 = 0$, the logical operation of XOR is performed by the circuit. When $X1 = X2$, the two resistances are equal and the output is zero.

A combination of the above methods may also be used where a single technique is not effective. In some cases more than one method may be applicable. In such scenarios, the choice of technique is based on design requirements of timing and area.

For example, to construct a logical unit that performs the operations of AND, OR, NAND and NOR, neutralization using control input and using logic may be used. Fig. 7.10 shows the use of such a technique. MTJ resistances R_1 and R_5 produce the AND/OR output. During this operation, sum of resistances R_2, R_3 and R_4 can be made equal to that of R_6, R_7 and R_8 using suitable control signals. The control signals are shown in the truth table. Similar method is used for NAND/NOR function.

This unit may also be constructed solely using neutralization with spin torque transfer. Fig. 7.11 shows the design of the same logical unit using neutralization with STT, along with the values of control inputs. The AND/OR part of the circuit is due to resistance R_1 and its reference R_3, while the NAND/NOR part is due to R_2 and its reference resistance R_4. R_5 and R_6 are additional MTJs in the circuit with a low resistance. The function of R_5 is to neutralize R_1, and that of R_6 is to neutralize R_2. Here, neutralization simply means making the output resistance, R_1 or R_2 low, as each



A	B	X1	X2	X3	R1	R2	R3	R4	R5	R6	R7	R8	R1R2R3R4	R5R6R7R8	OUT	Function
0	0	0	0	1	0	1	0	0	0	1	0	0	LLLH	LLLH	0	AND
0	1	0	0	1	0	1	0	1	0	1	0	1	LLHH	LLHH	0	
1	0	0	0	1	0	1	1	0	0	1	0	1	LLHH	LLHH	0	
1	1	0	0	1	1	0	1	1	0	1	0	1	LHHH	LLHH	1	
0	0	1	0	1	0	1	0	0	0	1	0	0	LLLH	LLLH	0	OR
0	1	1	0	1	1	1	0	1	0	1	0	1	LHHH	LLHH	1	
1	0	1	0	1	1	1	1	0	0	1	0	1	LHHH	LLHH	1	
1	1	1	0	1	1	0	1	1	0	1	0	1	LHHH	LLHH	1	
0	0	0	0	0	0	1	0	0	0	0	0	0	LLLH	LLLL	1	NAND
0	1	0	0	0	0	1	0	0	0	0	0	0	LLLH	LLLL	1	
1	0	0	0	0	0	1	0	0	0	0	0	0	LLLH	LLLL	1	
1	1	0	0	0	1	0	0	0	0	0	0	1	LLLH	LLLH	0	
0	0	0	1	0	0	1	0	0	0	0	0	0	LLLH	LLLL	1	NOR
0	1	0	1	0	0	0	0	0	0	0	0	0	LLLL	LLLL	0	
1	0	0	1	0	0	0	0	0	0	0	0	0	LLLL	LLLL	0	
1	1	0	1	0	1	0	0	0	0	0	0	1	LLLH	LLLH	0	

Figure 7.10: Logical unit performing AND, OR, NAND, NOR using neutralization by control input and neutralization through logic

of their references R_3 and R_4 are low.

The operation of the circuit may be described as follows: In the first time step, MTJs that constitute R_1 and R_2 are allowed to set their resistance (output) as a function of their inputs. Control inputs X_1 and X_2 are set according to desired function. In the second time step, for AND/OR function, MTJ R_2 is neutralized by transferring the state of R_6 to it. This requires that control input X_4 be high. R_2 and R_4 then have same resistance and hence the sense amplifier relies on the difference $(R_1 - R_3)$ only to generate its output. For NAND/NOR operation, MTJ R_1 is neutralized in a similar manner.

When neutralization with STT is used, the complete logic operation of the circuit needs at least three control steps: logic, spin torque transfer and output generation. When either or a combination of the other two methods is used, it may be possible to generate the output in just two control steps: logic and output generation.

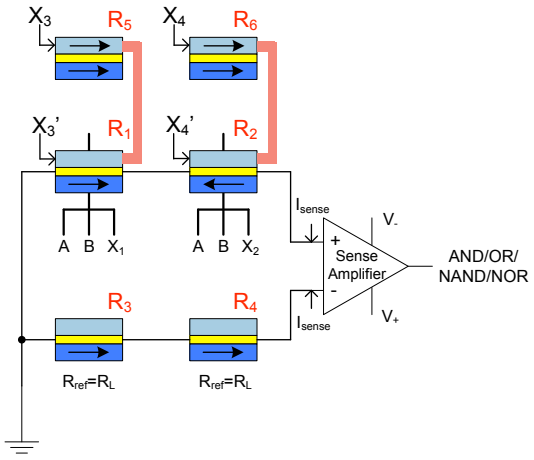


Figure 7.11: Logical unit performing AND, OR, NAND, NOR using neutralization by STT

Using the technique of neutralization, combining logic designs is simplified. An ALU can be constructed using all three techniques. Fig. 7.12 shows the schematic of a 1-bit ALU that is extendable upto n bits.

7.4 ALU operational details

Table 7.1 lists the MTJs responsible for each function in the ALU. The number of MTJs required if one or more functions are dropped from the ALU is also shown in the table.

Table 7.2 shows the logical values of control inputs required for each function of the ALU. X_1 through X_5 are current control inputs and have the value -I (logic-0) and +I(logic-1). X_7 through X_{10} are control inputs that have values of 0(logic-0) and +I'(logic 1). X_6 is generated as two different control signals suitably. The unit takes three control steps to produce output: logic step, spin torque transfer step, output generation step. When extended to n-bits, the last step overlaps with the first step of the next unit. Thus n-bit unit takes 2n+1 control steps to perform an n-bit operation.

A high-level timing graph of an n-bit ALU operation is shown by Fig. 7.13. The ALU operation completes within 2n+1 equal micro time spans. Each span is of the order of nanoseconds.

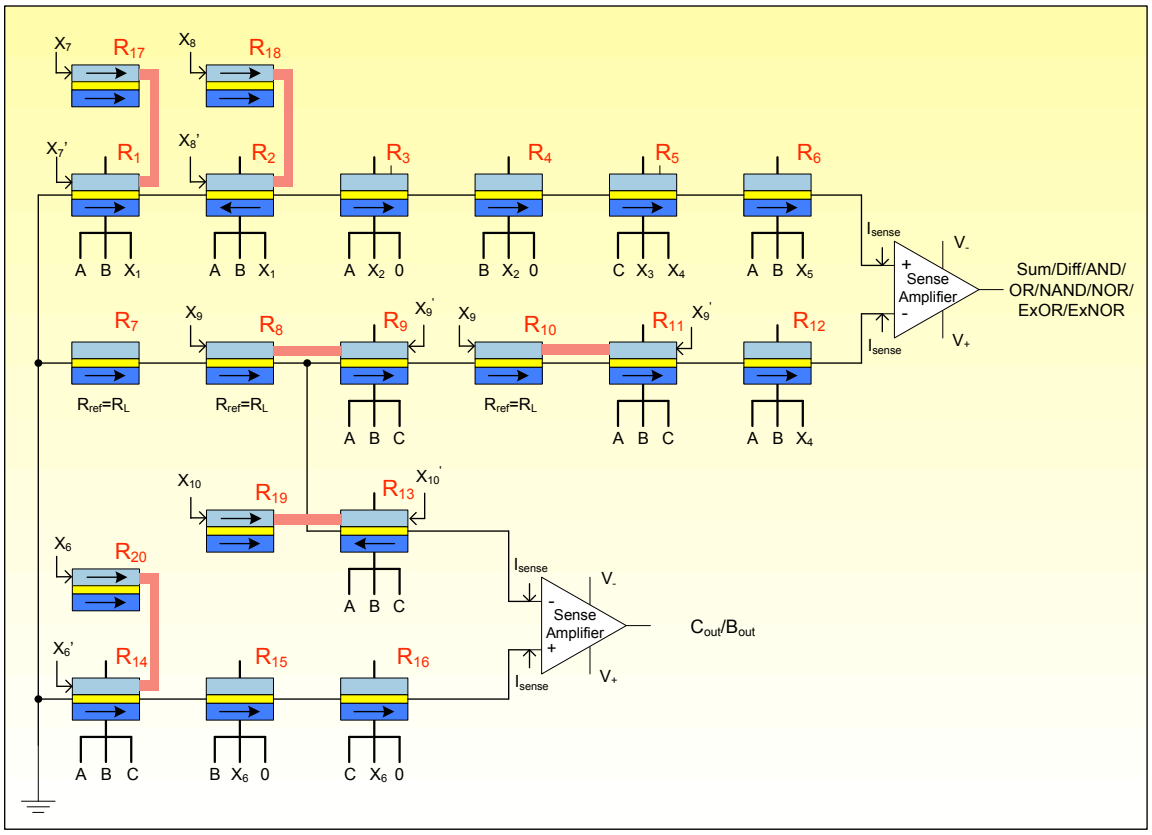


Figure 7.12: 1-bit spintronic ALU extendable upto n bits

7.4.1 Comparison with traditional ALU

The process technology for fabricating MTJs is still at the laboratory level. Intensive research is being done to improve characteristics of the device and better operating parameters are being investigated. Based on current research results, only a qualitative comparison is possible at this stage, between the infant spintronic technology and a mature CMOS technology.

A 1-bit ALU that performs the eight functions selected by 3-bit control inputs, constructed using current CMOS technology requires about fifty 45nm transistors depending on design and optimizations. MTJs can be fabricated to be rectangular devices with shorter length as low as 40nm in laboratory environment currently. However, the actual dimensions of the device depend upon a number of factors such as TMR desired,

Function	MTJ pair in ALU responsible for logic function	Number of MTJs required
AND, OR	R1,R7	3
NAND, NOR	R2,R8	3
ExOR	R6, R12	2
ExNOR	(R5,R6),(R7,R12)	4
Add	(R3,R4,R5), (R9,R10,R11) for Sum (R7,R8,R13), (R14,R15,R16) for C_{out}	7
Sub	(R3,R4,R5), (R9,R10,R11) for Sum (R7,R8,R13), (R14,R15,R16) for B_{out}	6
Adder-Subtractor		12
Logical Unit: Four logical Functions (AND, OR, NAND, NOR)		6
Logical Unit: Six logical Functions (AND, OR, NAND, NOR, ExOR, ExNOR)		10
ALU		20

Table 7.1: Description of ALU sub-units

Func	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
AND	0	0	0	0	0	x	0	1	1	x
OR	1	0	0	0	0	x	0	1	1	x
NAND	x	0	0	0	0	x	1	0	1	x
NOR	x	0	0	0	0	x	1	0	1	x
XOR	x	0	0	0	1	x	1	1	1	x
XNOR	x	0	1	1	0	x	1	1	1	x
Add	x	1	1	0	0	0	1	1	0	1
Sub	x	1	1	0	0	1	1	1	0	0

Table 7.2: Logical values of control inputs X1 through X10 for ALU

barrier layer thickness, barrier material choice, magnetic hardness of the layers, etc. and are expected to be smaller. An MTJ 1-bit ALU uses only 20 MTJ devices and two sense amplifiers (about four to seven CMOS transistors each), thus showing an area advantage over CMOS. The critical path in the CMOS ALU has a propagation delay of about 5 logic gates. The delay of the MTJ ALU is 3 control steps where each step has 1-MTJ device delay of the order of nanoseconds. MRAM research showed that write speeds of 3ns (and as low as 1.5ns) could be achieved with MTJs when combined with 180nm CMOS technology [58]. Thus an MTJ ALU can be expected to achieve speeds fairly comparable to CMOS. MTJs are low-power devices. They use bias voltages less than

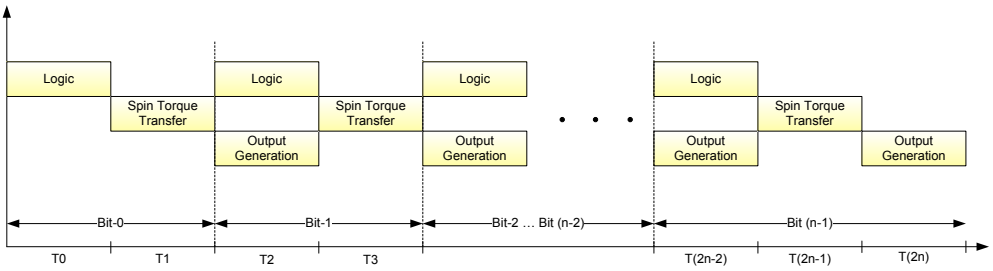


Figure 7.13: n bit ALU operation on a timeline

Parameter	CMOS-based 1-bit ALU	1-bit Magnetic ALU
Size	∼50 CMOS transistors	20 MTJs+14 CMOS
Delay	5-10 CMOS delay	3 MTJ delay + 2 CMOS delay
# of control signals	3	10
Speed of device	order of 0.1 ns	order of ns

Table 7.3: Comparison between 1-bit magnetic ALU and CMOS-based ALU

$1V$ for operation. This makes the 20-MTJ ALU power-efficient. A disadvantage of the MTJ ALU design is that it uses more control signals than a CMOS-based ALU. Also, some signals are directional currents while others take two values $(0, +I)$. Table 7.3 summarizes this comparison between a 1-bit magnetic ALU and a CMOS-based 1-bit ALU.

7.5 Going forward

The technique of Union with Neutralization can be used for MTJs operated using the CIMS technique. This has been shown in the ALU design presented in [59]. This design strategy allows for complex logic blocks to be designed without using single logic gates. That presents an interesting and a novel way of thinking for designing complete circuits using MTJs. The practical issues in such designs have not yet been examined.

Chapter 8

Conclusion

An emerging technology possesses an unexplored potential to provide useful functions. The historical evolution of early transistors into current computing systems is an excellent example of the potential and impact of a single technology. The emergence and wide usage of a technology depends on many factors: cost of production, availability of materials, its capability and performance, applications, need for those applications, etc. With the current trend, the eventual success of the technologies also depends on how long it can sustain a relentless growth in computing requirements.

The emerging technologies of NEMS and Spintronics are in their preliminary stages of development, with a number of questions about the extent, limit and impact of their potential. The initial research goals of this thesis were to recognize, predict and leverage the potential. With the NEMS and Spintronics technologies, several fundamental questions were asked about the synergy between their capabilities and computing needs: What are the basic abilities required to design computer systems, and how do the emerging technologies provide them? What are the new abilities these technologies bring that will allow us to perform computing better? How are the emerging technologies different from the conventional electronics technology? What aspects of computer design do these differences affect? The discussions of these questions form the basic answers to the feasibility question: *is it possible to compute with these emerging technologies?* With a switch-based technology such as NEMS, the feasibility issue is closely tied to performance perspectives, while for a non-switch based technology such as Spintronics, even the basic computing notions are needed to be examined and demonstrated.

8.1 Summary of Contributions

This thesis developed digital computing elements with devices from the two technologies of NEMS and Spintronics. The NEMS devices are behaviorally similar to CMOS devices, but differ in their internal device operation. Their mechanical switching abilities provide a near-ideal OFF state providing a significant reduction in power consumption of circuits. In addition, the same device structure acts as a p-type and an n-type device, and therefore the layout of a digital circuit can be expected to take up lesser area than a corresponding CMOS-based circuit. However, the mechanical switching has a significant disavantage of delay as compared to an electrical switching. This thesis took a unique approach to take advantage of the electromechanical device operation. The idea that a device can perform more than a simple ON/OFF operation was proposed, and a logic design scheme (*Weighted Area Logic*) to convert the simple switch into a logical switch was developed (Chp.3)). Theoretically, two surfaces that employ electromechanical forces for actuation and restoration can be converted into a logical switch irrespective of the device structure, however the Weighted Area Logic scheme relies on the ability to fabricate these devices with a good degree of control. Given the ability to fabricate these devices, the potential advantages of delay and energy were evaluated by taking the example of a 32-bit adder circuit. When compared to a circuit where the devices acted as simple switches, instead of logical switches, the metrics of delay and power were found to be better for the proposed scheme.

The spintronic devices provide a bistable operation based on spin transport and magnetic effects in structures. These effects can be leveraged for logic operations, however the next steps to complex computing are not straightforward. In the course of this work, it was found that there are several fundamental notions that current digital circuits are based on, which do not translate directly to a different technology such as Spintronics. For that reason, one cannot take advantage of the current abstractions of device, circuit, logic and architecture. Further, since a single spintronic device of Magentic Tunnel Junction (MTJ) can perform a logic function, the abstractions of device, circuit and logic are naturally blended. Therefore, the abstractions that apply to the spintronic technology must be developed anew, and appropriately.

An important ability necessary for designing complex computing systems is that of

input-output signal compatibility, which is further required for composing devices into a multi-device circuit. The *current* inputs of MTJs and *resistance* outputs do not allow a natural compatibility that could be leveraged for composing circuits. This is one of the major challenge of the technology for computation. This thesis proposed a circuit design technique to use the natural input and output states of the device while ensuring compatibility (Chp.5). The technique elegantly allows a simple biasing scheme to enable a programmable logic operation. This technique was demonstrated with a prototype circuit.

Extending this circuit into complex logic functions requires additional hardware. With additional switches, a non-volatile logic gate can be composed into a multi-gate function. The hardware requirements to enable this were studied, and presented in Chp.6 as the Spintronic Logic In Cache (SLIC) architecture. This architecture allows accomplishing a complex function such as an ADD function in a memory array without the need of electronic computing circuits.

The delay of a single spintronic operation is currently greater than that achieved by an electronic computation, however, a fair comparison must take into consideration the abilities of the technologies being compared. One way to compare the spintronic and electronic computation is by designing a data path that consists of a memory array and computing circuits, so that the two data paths are accomplishing equal work functionally. Electronic circuits have to rely on a non-volatile memory to store operands and electronic circuits to process them. Therefore, the non-volatile nature of spintronic computation has the potential to accomplish a non-volatile logic operation faster than electronic circuits. A preliminary comparison of this was shown in this thesis (Chp.5)

Taking another independent approach to designing logic circuits with MTJs, a tehnique called *union with neutralization* was also proposed (Chp.7). This technique extended the previously proposed idea of using the relative resistance difference between two arrays of MTJs connected serially to accomplish a logic function. The problem that the work in this thesis attempted to solve was that of combining multiple previously designed circuits. This is achieved through use of multiplexers for electronic circuits, however, it was not obvious how to combine spintronic circuits. Three methods of combination were proposed and were used to present the design of a 1-bit spintronic ALU that can perform 8-functions including full addition/subtraction and six logic operations.

8.2 Unanswered questions

The perspective taken during the entire work was that of evaluating the feasibility aspect of computing with the technologies, before measuring their performance. The goal was to perform the foundational work to evaluate the role of the technologies in computing, as well as explore the impact of computer design on technology development. Thus, the ideas presented here are aimed to guide device development efforts as well as application exploration by discussing the nature of the technologies and the ways of leveraging them.

This early work with the novel devices has also revealed multiple questions that must be answered in the near future: What performance benefits are possible with purely electromechanical or spintronic systems? How can one extract optimal power-performance designs? Where do the trade-offs lie in hybrid combinations with CMOS devices? What amount of variability can be tolerated by the logic designs proposed here and how can we increase the tolerance? Explorations along these lines would further aid application exploration for the technologies, along with a clearer picture of the role that they could play in future systems.

To answer these questions, extensive simulations with precise device models as well as prototyping efforts will be required. Due to the ongoing nature of the device work, accurate device models are not yet available, and must be developed. Also, there will be the need for close interaction between the devices, circuits and computer design efforts to take better advantage of the increased functionalities of these devices.

8.3 Speculating the timeline and future of the technologies

There are ongoing research efforts to characterize and improve device properties, reliability, yield, etc. at the device level, to build digital and analog circuits at the circuit level, and to design logic and memory systems at the logic level of system design. The NEMS technology has evolved from the now mature and into-production MEMS technology. MEMS devices are currently used in a variety of applications such as sensors, actuators and gyroscopes on silicon chips along with electronic devices. Going into nanoscale regimes, the digital switches that are being developed using the NEMS technology face challenges of scalability, variability, reliability, lifetime, etc. To address these, there is ongoing research into novel materials, fabrication processes and novel device structures.

New materials such as carbon nanotubes and graphene show promising characteristics to implement devices comparable in performance to existing MOSFETs. The NEMS switches can alleviate immediate power challenges in CMOS devices, and therefore, once fabrication challenges are resolved, the emergence and utilization of NEMS devices in computer technology can be expected in the near future, optimistically in the next 5-10 years. To enable this, along with device-level work, research must be undertaken to design and evaluate complete computing systems. More ideas at circuit, architecture and system levels must be proposed to leverage the device properties.

Spintronics, on the other hand, deals with quantum effects, that require an extremely sophisticated control for characterization and implementation. Spintronics-based memories (magnetic memories called MRAMs) are in near-production stage, however, spintronics-based logic units are in their preliminary stages of research. Reliable operation of the spintronics logic must be demonstrated along with prototypes with power advantages, and innovative ways to leverage their unique properties must be proposed before they can be expected to be used in modern computers for logic operations. One can anticipate that there will be high performance expectations from an elegant technology such as spintronics when devices such as FinFETS reach their power-performance tradeoff points. Therefore one can expect the spintronics logic technology to mature a little further in the future, perhaps within the next two decades.

Regardless of the actual time of emergence of these technologies in computer chips, the challenges with electronics technology itself and ongoing research with the emerging technologies suggest that VLSI would move towards heteregeneous technologies, instead of a single technology. The move to heterogeneous computing has already occurred at the system-level and now at the architecture-level. With unique advantages obtained from different technologies, as the expertise to fabricate them reliably increases, it will be a natural move to adopt heterogeneity at the technology level. This implies that much work must be performed on integration and co-existence of different physics, materials, devices, circuits and architectures, in order to gain the kind of performance expected from future systems.

With a legacy technology like CMOS poised to change in uncertain ways, this is an exciting time for research and development of computing-capable technologies. And

aside from the practical engineering motivation, research into nanoscale emerging technologies open up a rich and exciting knowledge base about the dynamics of forces at these dimensions. This provides us yet another opportunity to engineer the forces of nature into constructive end-products!

References

[1] Richard Feynman. There's plenty of room at the bottom. *Caltech Engineering and Science*, 23(5):22–36, February 1960.

[2] ITRS. Emerging research devices 2009 edition. *International Technology Roadmap for Semiconductors (ITRS)*, 2009.

[3] T. Rueckes, K. Kim, E. Joselevich, G.Y. Tseng, C.L. Cheung, and C.M. Lieber. Carbon nanotube-based nonvolatile random access memory for molecular computing. *science*, 289(5476):94, 2000.

[4] W. Gallagher and S. Parkin. Development of the magnetic tunnel junction mram at ibm: from first junctions to a 16-mb mram demonstrator chip. *IBM Journal of Research and Development*, 50(1):5–23, 2006.

[5] M. Hosomi, H. Yamagishi, T. Yamamoto, K. Bessho, Y. Higo, K. Yamane, H. Yamada, M. Shoji, H. Hachino, C. Fukumoto, H. Nagao, and H. Kano. A novel nonvolatile memory with spin torque transfer magnetization switching: spin-ram. *IEEE International Electron Devices Meeting, IEDM Technical Digest*, pages 459–462, 2005.

[6] S. Tehrani, J.M. Slaughter, E. Chen, M. Durlam, J. Shi, and M. DeHerrera. Progress and outlook for mram technology. *IEEE Transactions on Magnetics*, 35(5):2814–2819, 1999.

[7] S. Patil, M-W. Jang, C-L. Chen, D. Lee, Z. Ye, W. Partlo III, D. Lilja, S. Campbell, and T. Cui. Weighted area technique for electromechanically enabled logic

computation with cantilever-based nems switches. *Design, Automation and Test in Europe (DATE)*, February 2012.

[8] S. Patil, A. Lyle, J. Harms, D. Lilja, and J.-P. Wang. Spintronic logic gates for spintronic data using magnetic tunnel junctions. *IEEE International Conference on Computer Design (ICCD 2010)*, pages 125–131, 2010.

[9] S. Patil, X. Yao, H. Meng, J.-P. Wang, and D. Lilja. Design of a spintronic arithmetic and logic unit using magnetic tunnel junctions. *Proceedings of the 5th conference on Computing frontiers*, pages 171–178, 2008.

[10] Shruti Patil and David J. Lilja. Performing bitwise logic operations in cache using spintronics-based magnetic tunnel junctions. In *Proceedings of the 8th ACM International Conference on Computing Frontiers*, CF '11, pages 33:1–33:9, 2011.

[11] Shruti Patil and David J. Lilja. A programmable and scalable technique to design spintronic logic circuits based on magnetic tunnel junctions. In *Proceedings of the 21st edition of the great lakes symposium on Great lakes symposium on VLSI*, GLSVLSI '11, pages 7–12, 2011.

[12] F. Chen, Hei Kam, D. Markovic, Tsu-Jae King Liu, V. Stojanovic, and E. Alon. Integrated circuit design with nem relays. In *Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on*, pages 750 –757, nov. 2008.

[13] S.W. Lee, D.S. Lee, R.E. Morjan, S.H. Jhang, M. Sveningsson, OA Nerushev, Y.W. Park, and E.E.B. Campbell. A three-terminal carbon nanorelay. *Nano letters*, 4(10):2027–2030, 2004.

[14] T.-H. Lee, K.M. Speer, X.A. Fu, S. Bhunia, and M. Mehregany. Polycrystalline silicon carbide nems for high-temperature logic. In *Solid-State Sensors, Actuators and Microsystems Conference, 2009. TRANSDUCERS 2009. International*, pages 900 –903, june 2009.

[15] Hamed F. Dadgour, Muhammad M. Hussain, and Kaustav Banerjee. A new paradigm in the design of energy-efficient digital circuits using laterally-actuated double-gate nems. In *Low-Power Electronics and Design (ISLPED), 2010 ACM/IEEE International Symposium on*, pages 7 –12, aug. 2010.

[16] Min-Woo Jang, Chia-Ling Chen, Walter E. Partlo, Shruti R. Patil, Dongjin Lee, Zhijang Ye, David Lilja, T. Andrew Taton, Tianhong Cui, and Stephen A. Campbell. A pure single-walled carbon nanotube thin film based three-terminal microelectromechanical switch. *Applied Physics Letters*, 98(7):073502 –073502–3, feb 2011.

[17] JM Kinaret, T. Nord, and S. Viefers. A carbon-nanotube-based nanorelay. *Applied Physics Letters*, 82:1287, 2003.

[18] MW Jang, M. Lu, T. Cui, and SA Campbell. Functional 1.6 ghz mems switch using aligned composite cnt membrane by dielectrophoretic self-assembly. In *Solid-State Sensors, Actuators and Microsystems Conference, 2009. TRANSDUCERS 2009. International*, pages 912–915. IEEE, 2009.

[19] N.A. Poklonski, E.F. Kislyakov, S.A. Vyrko, O.N. Bubel, A.I. Siahlo, N.N. Hieu, I.V. Lebedeva, A.A. Knizhnik, A.M. Popov, and Y.E. Lozovik. A low-voltage magnetic nanorelay design. *SPIE*, nov. 2010.

[20] M. Lu, M.W. Jang, G. Haugstad, S.A. Campbell, and T. Cui. Well-aligned and suspended single-walled carbon nanotube film: Directed self-assembly, patterning, and characterization. *Applied Physics Letters*, 94(26):261903–261903, 2009.

[21] S. Srinivasan, J. Hiller, B. Kabius, and O. Auciello. Piezoelectric/ultrananocrystalline diamond heterostructures for high-performance multifunctional micro/nanoelectromechanical systems. *Applied Physics Letters*, 90(13):134101 –134101–3, mar 2007.

[22] S. Bhunia, M. Tabib-Azar, and D. Saab. Ultralow-power reconfigurable computing with complementary nano-electromechanical carbon nanotube switches. In *Design Automation Conference, 2007. ASP-DAC '07. Asia and South Pacific*, pages 86 –91, jan. 2007.

[23] K. Alzoubi, D.G. Saab, and M. Tabib-Azar. Complementary nano-electromechanical switches for ultra-low power embedded processors. In *Proceedings of the 19th ACM Great Lakes symposium on VLSI*, pages 309–314. ACM, 2009.

[24] LM Jonsson, S. Axelsson, T. Nord, S. Viefers, and JM Kinaret. High frequency properties of a cnt-based nanorelay. *Nanotechnology*, 15:1497, 2004.

[25] H.F. Dadgour and K. Banerjee. Design and analysis of hybrid nems-cmos circuits for ultra low-power applications. In *Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE*, pages 306 –311, june 2007.

[26] S. Senturia. *Microsystem Design*. Springer, 2000.

[27] D. Patil, O. Azizi, M. Horowitz, R. Ho, and R. Ananthraman. Robust energy-efficient adder topologies. In *Computer Arithmetic, 2007. ARITH '07. 18th IEEE Symposium on*, pages 16 –28, june 2007.

[28] J. S. Moodera, L. R. Kinder, T. M. Wong, and R. Meservey. Large magnetoresistance at room temperature in ferromagnetic thin film tunnel junctions. *Physical Review Letters*, 74(16):3273–3276, 1995.

[29] L. Berger. Emission of spin waves by a magnetic multilayer traversed by a current. *Physical Review B*, 54(13):9353–9358, 1996.

[30] E. B. Myers, D. C. Ralph, J. A. Katine, R. N. Louie, and R. A. Buhrman. Current-induced switching of domains in magnetic multilayer devices. *Science*, 285(5429):867–870, 1999.

[31] J. Slonczewski. Current-driven excitation of magnetic multilayers. *Journal of Magnetism and Magnetic Materials*, 159(1-2):L1–L7, 1996.

[32] T. Kawahara, R. Takemura, K. Miura, J. Hayakawa, S. Ikeda, Y. Lee, R. Sasaki, Y. Goto, K. Ito, I. Meguro, F. Matsukura, H. Takahashi, H. Matsuoka, and H. Ohno. 2mb spin-transfer torque ram (spram) with bit-by-bit bidirectional current write and parallelizing-direction current read. *IEEE International Solid-State Circuits Conference*, pages 480–617, 2007.

[33] J. Wang, H. Meng, and J.-P. Wang. Programmable spintronics logic device based on a magnetic tunnel junction element. *Journal of Applied Physics*, 97(10):10D509, 2005.

[34] H. Meng, J. Wang, and J.-P. Wang. A spintronics full adder for magnetic cpu. *IEEE Electron Device Letters*, 26(6):360–362, 2005.

[35] S. Lee, N. Kim, H. Yang, G. Lee, S. Lee, and H. Shin. The 3-bit gray counter based on magnetic-tunnel-junction elements. *IEEE Transactions on Magnetics*, 43(6):2677–2679, 2007.

[36] J-P. Wang and X. Yao. Programmable spintronic logic devices for reconfigurable computation and beyond – history and outlook. *Journal of Nanoelectronics and Optoelectronics*, 3:12–23, 2008.

[37] W. Zhao, E. Belhaire, and C. Chappert. Spin-mtj based non-volatile flip-flop. *7th IEEE Conference on Nanotechnology (IEEE-NANO 2007)*, pages 399–402, 2007.

[38] W. Zhao, E. Belhaire, V. Javerliac, C. Chappert, and B. Dieny. A non-volatile flip-flop in magnetic fpga chip. *International Conference on Design and Test of Integrated Systems in Nanoscale Technology (DTIS)*, pages 323–326, 2006.

[39] W. Zhao, E. Belhaire, C. Chappert, and P. Mazoyer. Spintronic device based non-volatile low standby power sram. *IEEE Computer Society Annual Symposium on VLSI*, pages 40–45, 2008.

[40] S. Matsunaga, J. Hayakawa, S. Ikeda, K. Miura, H. Hasegawa, T. Endoh, H. Ohno, and T. Hanyu. Fabrication of a nonvolatile full adder based on logic-in-memory architecture using magnetic tunnel junctions. *Applied Physics Express*, 1:091301–3, 2008.

[41] W. Black and B. Das. Programmable logic using giant-magnetoresistance and spin-dependent tunneling devices. *Journal of Applied Physics*, 87(9):6674–6679, 2000.

[42] Andrew Lyle, Shruti Patil, Jonathan Harms, Brian Glass, Xiaofeng Yao, David J. Lilja, and Jian-Ping Wang. Magnetic tunnel junction logic architecture for realization of simultaneous computation and communication. *Presented at IEEE International Magnetics Conference (InterMag), Taipei, Taiwan, April 2011. IEEE Transactions on Magnetics(In Press)*, 2011.

[43] Z. Diao, A. Panchula, Y. Ding, M. Pakala, S. Wang, Z. Li, D. Apalkov, H. Nagai, A. Driskill-Smith, L-C.Wang, E. Chen, and Y. Huai. Spin transfer switching in dual mgo magnetic tunnel junctions. *Applied Physics Letters*, 90(13):132508, 2007.

[44] J. Harms, F. Ebrahimi, X. Yao, and J.-P. Wang. Spice macromodel of spin-torque-transfer operated magnetic tunnel junctions. *IEEE Transactions on Electronic Devices*, 2010.

[45] A. Lyle, J. Harms, S. Patil, X. Yao, D. Lilja, and J.-P. Wang. Direct communication between magnetic tunnel junctions for non-volatile logic fan-out architecture. *Applied Physics Letters*, 97(15):152504–152506, 2010.

[46] H. Kubota, A. Fukushima, Y. Ootani, S. Yuasa, K. Ando, H. Maehara, K. Tsunekawa, D. Djayaprawira, N. Watanabe, and Y. Suzuki. Evaluation of spin-transfer switching in cofeb/mgo/cofeb magnetic tunnel junctions. *Japanese Journal of Applied Physics*, 44:L1237–L1240, 2005.

[47] J. Hayakawa, S. Ikeda, Y. M. Lee, R. Sasaki, T. Meguro, F. Matsukura, H. Takahashi, and H. Ohno. Current-induced magnetization switching in mgo barrier based magnetic tunnel junctions with cofeb/ru/cofeb synthetic ferrimagnetic free layer. *Japanese Journal of Applied Physics*, 45:L1057–L1060, 2006.

[48] Y. Huai, M. Pakalaa, Z. Diaoa, D. Apalkova, Y. Dinga, and A. Panchulaa. Spin-transfer switching in mgo magnetic tunnel junction nanostructures. *Journal of Magnetism and Magnetic Materials*, 304(1):88–92, 2006.

[49] B. R. Gaeke, P. Husbands, X. S. Li, L. Oliker, K. A. Yelick, and R. Biswas. Memory-intensive benchmarks: Iram vs. cache-based machines. *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS)*, 2002.

[50] R. Murphy and P. M. Kogge. The characterization of data intensive memory workloads on distributed pim systems. *Proceedings of Intelligent Memory Systems Workshop, ASPLOS-IX*, 2000.

[51] J. Harms, F. Ebrahimi, X. Yao, and J.-P. Wang. Spice macromodel of spin-torque-transfer operated magnetic tunnel junctions. *IEEE Transactions on Electronic Devices*, 2010.

[52] H. Zhao, A. Lyle, Y. Zhang, P. K. Amiri, G. Rowlands, Z. M. Zeng, J. Katine, H. W. Jiang, K. Galatsis, K. L. Wang, I. N. Krivorotov, and J.-P. Wang. Low writing energy and sub nano-second spin torque transfer switching of in-plane magnetic tunnel junction for stt-ram. *To be published in Journal of Applied Physics*, 2011.

[53] D. Lilja and S. Sapatnekar. *Designing Digital Computer Systems with Verilog*. Cambridge University Press, 2005.

[54] R. L. Plackett and J. P. Burman. The design of optimum multifactorial experiments. *Biometrika*, 33:305–325, 1946.

[55] H. Meng, J. Wang, and J.-P. Wang. A spintronics full adder for magnetic cpu. *IEEE Electron Device Letters*, 26(6):360–362, 2005.

[56] S. Lee, N. Kim, H. Yang, G. Lee, and H. Shin. The 3-bit gray counter based on magnetic-tunnel-junction elements. *IEEE Transactions on Magnetics*, 6:2677–2679, 2007.

[57] A. Lyle, Xiaofeng Yao, F. Ebrahimi, J. Harms, and Jian-Ping Wang. Communication between magnetic tunnel junctions using spin-polarized current for logic applications. *Magnetics, IEEE Transactions on*, 46(6):2216 –2219, june 2010.

[58] W. J. Gallagher and S. S. P. Parkin. Development of the magnetic tunnel junction mram at ibm: from first junctions to a 16-mb mram demonstrator chip. *IBM Journal of Research and Development*, 50(1):5–23, 2006.

[59] Xiaofeng Yao, Jonathan Harms, Andrew Lyle, Farbod Ebrahimi, Yisong Zhang, and Jian-Ping Wang. Magnetic tunnel junction-based spintronic logic units operated by spin transfer torque. *IEEE Transactions on Nanotechnology. In Press.*, 2011.