PERFORMANCE ANALYSIS OF FAULT-TOLERANT NANOELECTRONIC MEMORIES

A Dissertation

by

AYODEJI O. COKER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2008

Major Subject: Computer Engineering

PERFORMANCE ANALYSIS OF FAULT-TOLERANT NANOELECTRONIC MEMORIES

A Dissertation

by

AYODEJI O. COKER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Valerie E. Taylor
Committee Members,	Eun Jung Kim
	Hank Walker
	Lazlo Kish
Head of Department,	Valerie E. Taylor

May 2008

Major Subject: Computer Engineering

ABSTRACT

Performance Analysis of Fault-Tolerant Nanoelectronic Memories. (May 2008)

Ayodeji O. Coker, B.S, State University of New York at Albany; M.S, Northwestern University

Chair of Advisory Committee: Dr. Valerie E. Taylor

Performance growth in microelectronics, as described by Moore's law, is steadily approaching its limits. Nanoscale technologies are increasingly being explored as a practical solution to sustaining and possibly surpassing current performance trends of microelectronics. This work presents an in-depth analysis of the impact on performance, of incorporating reliability schemes into the architecture of a crossbar molecular switch nanomemory and demultiplexer. Nanoelectronics are currently in their early stages, and so fabrication and design methodologies are still in the process of being studied and developed. The building blocks of nanotechnology are fabricated using bottom-up processes, which leave them highly susceptible to defects. Hence, it is very important that defect and fault-tolerant schemes be incorporated into the design of nanotechnology related devices.

In this dissertation, we focus on the study of a novel and promising class of computer chip memories called crossbar molecular switch memories and their demultiplexer addressing units. A major part of this work was the design of a defect and fault tolerance scheme we called the Multi-Switch Junction (MSJ) scheme. The MSJ

scheme takes advantage of the regular array geometry of the crossbar nanomemory to create multiple switches in the fabric of the crossbar nanomemory for the storage of a single bit.

Implementing defect and fault tolerant schemes come at a performance cost to the crossbar nanomemory; the challenge becomes achieving a balance between device reliability and performance. We have studied the reliability induced performance penalties as they relate to the time (delay) it takes to access a bit, and the amount of power dissipated by the process. Also, MSJ was compared to the banking and error correction coding fault tolerant schemes. Studies were also conducted to ascertain the potential benefits of integrating our MSJ scheme with the banking scheme. Trade-off analysis between access time delay, power dissipation and reliability is outlined and presented in this work.

Results show the MSJ scheme increases the reliability of the crossbar nanomemory and demultiplexer. Simulation results also indicated that MSJ works very well for smaller nanomemory array sizes, with reliabilities of 100% for molecular switch failure rates in the 10% or less range.

DEDICATION

To my family!

ACKNOWLEDGEMENTS

This has been a long journey, and I did not get to this point purely on my own strength; there have been many people along the way that have encouraged and aided me along this arduous journey. Firstly, although this may sound cliché, it does not take away from the truth that accomplishing this feat would have been a virtual impossibility, if not for the grace of God. My parents, Ayo and Sade Coker, where do I start? They have always believed in me, even when I doubted myself. They have given me more than any son could ask for, and this degree is as much theirs as it is mine. Mum and Dad, I am eternally in your debt. I would also like to thank three of the greatest and most caring sisters that ever existed: Toyin, Tumi and Dami. You guys took it upon yourselves to bear the burden of my struggles with me. You have no idea how grateful I am. Thanks, Girls!

There are many people to thank in my academic life, first and foremost is my advisor and mentor, Dr. Valerie Taylor, who has stood by me all these years, all the way from Northwestern University to Texas A&M. Thanks for all the opportunities and for never letting me settle for less than my best. I would also like to thank Dr. Xingfu Wu for teaching me what it means to be hard working and dedicated. To my research group members, Charles Lively and Sameh Sharkawi, thanks a lot for putting up with all my antics, and for all the valuable research discussions and insights. Thanks are also in order for my past research group members Seung-Hye Jang, and Yue Zuo for all their help. To all my friends, too numerous to mention, you all know who you are; thanks for all the encouragements and prayers.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF FIGURES

Page

Page

Page

Page

Page

Page

Page

Page

Page

Page

Page

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION: NANOTECHNOLOGY MOLECULAR ELECTRONICS

Microelectronics is fast approaching its scaling limits due to physical and economic constraints. Molecular electronics is promising technology that is poised to continue the advances of microelectronics. The idea is to use single molecules as the building blocks to create logic circuits. These molecules will function as electronic switches and storage elements. These molecules have dimensions that are several orders of magnitude smaller than silicon based components. They are synthesized to carry out specific functions, and they can be coaxed via a self assembly process into forming regular two dimensional patterns as well as well defined three dimensional supramolecular objects. This makes them the ideal building blocks for future high density memory devices. With the concepts that underline the molecular self assembly process, one can envision entire computational processing units designed and grown from the bottom-up into practical working devices. Already molecular transistors with bistable switching properties have been developed and demonstrated for use in high-density non-volatile memories [1, 2].

This dissertation follows the style of *IEEE Transactions on Nanotechnology*.

Molecular electronics is still very much in its infancy, and as such many of the experiments, simulations and theories required to understand this subject this field are still being developed. This makes molecular electronics a difficult area to research, but albeit an interesting one. The primary contributions of this work are as follows;

- Studied and assessed the performance and reliability of crossbar or grid-like nanoelectronic memories.
- Developed and simulated a defect and fault-tolerant scheme called the Multi-Switch Junction scheme for achieving reliability in nanoelectronic memories.
- Trade-off analysis of the performance, reliability and area utility of crossbar or grid-like nanoelectronic memories.

The outline of this dissertation is as follows; in this chapter, the emerging nanotechnology field of molecular electronics is first discussed. In chapter II crossbar nanomemories are introduced along with their circuit model, architecture and operation; literature reviews on the most current research and development on crossbar nanomemories are also discussed. Also included in chapter II, is an overview and literature review of competing nanomemory technologies. In chapter III, crossbar nanomemory demultiplexers used to address the main crossbar nanomemory are presented along with accompanying circuit models and operational procedures. Chapter IV presents a detail study of the fundamental reliability techniques being researched and implemented in nanotechnology driven devices. The Multi-Switch Junction architecture

we developed is presented in chapter V, along with a detailed description of its design and implementation in crossbar nanomemories and demultiplexers.

The later chapters are focused on the simulation results. In chapter VI, the steps employed in modeling and simulating the performance of the nanomemory and demultiplexers access time delay and power dissipation; graphs and comparison tables of the simulation results, as well as result analysis, are also presented in chapter VI. The following chapter, VII, introduces and expands upon the reliability techniques and simulators—PRISM and Matlab—used in determining the defect and fault tolerance capabilities of the crossbar nanomemory and demultiplexers; simulation result graphs and comparison tables are also presented and analyzed. In conclusion, chapter VIII summarizes the research work presented in this dissertation, and concluding remarks on the simulation results and findings are discussed.

1.1 Challenges and Opportunities

Microelectronics is facing an array of challenges. They include developing lithographic patterning technology capable of producing sub-micron scale line widths, scaling power supply voltages, improving contact resistance, increasing gate capacitance and at the same time reducing gate dielectric tunneling leakages and sustaining device performance. These challenges are the driving force behind discovering new and better technologies. However, there are also obstacles that have to be scaled if molecular electronics is to be realized. They can be divided into the following categories [3]; Reliability, Interconnects and Parasitics, Charge Transportation, Power and Heat Dissipation and Molecular Electronic Devices. These categories are further expanded

upon in sections 1.1.1 through 1.1.5.

1.1.1 Reliability

Molecular electronic devices are fabricated using self assembly processes, As discussed earlier, this is a bottom-up process which means there will be a high occurrence of defects in the fabricated devices. As time progresses it is expected that better techniques will be developed to lessen the amounts of defect that are present in nano-electronic or molecular electronic devices. In the interim, defect-tolerant architecture will be a necessity if reliable molecular electronic devices are to be realized. The regularity and array nature of the self assembly process make grid or crossbar schemes [1], the architecture of choice for defect tolerance. Several fault tolerance schemes [4-6] are being considered as potential solutions to the quandary of reliably interconnecting a large number of molecular devices to each other. The Teramac computer [7] where a large number of faults were reliably tolerated is an example of such a scheme. A more in-depth look at reliability is presented in chapter VII.

1.1.2 Interconnects and Parasitics

Interconnects will also pose a major obstacle in molecular electronics. The difficulty lies in the fabrication and manipulation of nano wires. The nanometer dimensions of the wires limit the conducting cross-sectional area available for electron transport.

Scientists have been able to synthesize CNT wires nano scale width and lengths in the micrometer dimension [8]. On the other hand, controlling the electrical properties of

CNT, to synthesize conducting or semi-conducting CNTs is still a major obstacle for scientists. There are techniques [9, 10] for growing silicon and germanium nanowires which are presently being developed, and are already showing promising results. The electrical properties of the wires can be controlled by using dopants to yield semi-conducting nanowires [11].

The difficulties of interconnection in an ultra dense memory array are significant. Starting with the issue of isolating nanowires from each other to prevent such parasitic as cross-talk, RC signal decay during transmission between molecular devices in the chip as well as between chips. Another problem that arises is that of connecting Nanowires to individual molecules. Scientists are experiencing difficulties in reducing the high resistance encountered at the metal/molecule interface. This problem is a result of the conformational change that occurs to the molecules at the metal/molecule interface. This amounts to a change in shape of the molecules, which results in a distortion of its atomic orbital configurations thereby causing a change in the charge transfer characteristics at the metal-molecule interface [12].

1.1.3 Charge Transportation

Understanding charge transportation in nanoscale wires is an important step towards realizing molecular electronic devices. A fundamental requirement for molecular electronics is a mode by which they can be connected to other devices or circuitry. In order to drive current through individual molecules, it is necessary that we have an electrode pair with nanometer sized spacing to contact them [13]. Charge transportation can occur either by quantum transportation, where conduction is determined by the

tunneling of electrons through well defined energy levels, or by coulomb blockade where conduction is attained when potential is sufficient to overcome the energy barrier of charge correlation. Charge transportation in molecular electronics can be broken down conceptually into three discrete parts [14] – the molecular core and the electrode-molecule contacts on both sides of the core, an illustration (Figure 1) is provided below.

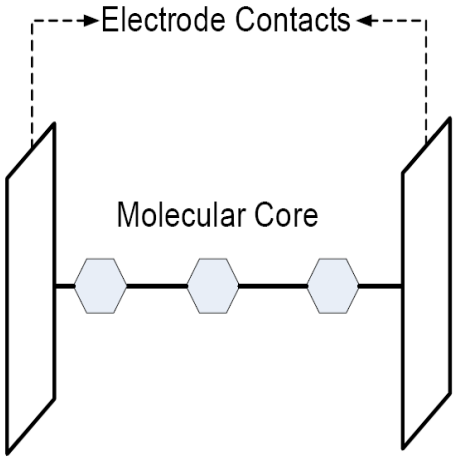

Figure 1. Electrode-Molecule-Electrode contact.

The nature of the electrode – molecule interface is determined by the choice of the metal electrode as well as the chemical functionality, or the "alligator clip" which connects the molecules to the metallic contacts [13]. The chemical structure and the electronic transport properties of integrated molecules are intertwined. Hence the advantage to scientists is that the electronic properties of devices may be adjusted by engineering the chemical structures of the molecules.

Electron transport between the electrode-molecule interfaces is dependent on the

coupling strength of the molecules to the contact electrode. The coupling is weak when the molecules are attached to the surface only by weak van der Waals forces [13]. In this case, electron transportation is manifested by a tunneling process in an electron travels from the electrode to the molecule, stays there for a while before making its way to the other electrode; thus yielding poor electron conduction. However, in the case where the coupling is strong, the molecular orbital hybridizes with the metallic state in the electrodes, thus yielding a broadening of the energy levels and a higher conductance [13].

1.1.4 Power and Heat Dissipation

Heat generation in a circuit is a function of the product of current and voltage. The steady state formula for the current needed to charge a capacitor one cycle time is given by equation (1-1), where f is the operating frequency of the device, C is the charge capacitance and V_{dd} is the applied voltage. As earlier stated, the power (P) required for a transistor or on/off device to switch, is given by the product of current flowing through the device and the voltage applied. The switching power is shown in equation (1-2).

$$I = f \bullet C \bullet V_{dd} \tag{1-1}$$

$$P = f \bullet C \bullet V_{dd}^{2} \tag{1-2}$$

These steady state equations assume that current does not flow during steady state operation, it corresponds only to the power lost during capacitive charging and

discharging. Molecular devices are projected to be very leaky devices, meaning that current will flow from them even during steady state operation; this presents a significant disadvantage for molecular device. In an ultra-dense molecular system with 10^{12} devices/cm^2, the power and heat dissipated are the integral of the switching power of each molecular device over the molecular system area. This will obviously yield a significant amount of heat since molecular devices are expected to operate at frequencies well into the gigahertz to terahertz regimes.

1.1.5 Molecular Electronic Devices

The molecular devices or switches referred to in this work, are those which consist of bistable molecules sandwiched between overlapping intersecting conducting or semi-conducting nanowires. The nature of the bistable molecules and nanowires dictate the functionality of the molecular device. Molecular switch junctions can also be implemented using intersecting nanowires and microwires, as is the case in nanoelectronic decoders.

The molecular switches could also be configurable, so that they can be independently activated or deactivated, as desired by the end user [15]. For example, configurable bistable molecules which function as resistors, when activated they behave as conventional resistors would, when deactivated their functionality resorts to that of a nanowire. In the crossbar configuration, this behavior is akin to that an open (deactivated) and closed (activated) switch. The molecules could be configured electrically, optically or mechanically. Reconfigurability may be performed repeatedly on these molecules. This is for the most part dependent on the application of these devices; when utilized in

nanomemories, repeat reconfiguration becomes a necessity. In the case of logic applications, reconfiguration may be few and far between.

1.2 Nanowires

Nanowires are gaining widespread acceptance as the building block for nano devices [16]. As a result we are seeing a transition from fabrication methodologies that depend primarily on top-down processes, to bottom-up processes. The top-down approach to fabrication has fueled the progress of microelectronics. It is an approach whereby bulk materials at the wafer level are patterned through lithography, etching and deposition processes, in order to achieve a desired micron or nanoscale device. As feature sizes in microelectronics get smaller, the limits to the resolution that can be achieved through lithographical processes is fast approaching. This not only poses an engineering impasse, but it also affects the ability of engineers to create economical microelectronic devices as the fabrication costs spirals at an equivalently faster pace.

Bottom-up processes are proving attractive, not only for the relative economic advantage they possess over top-down processes, but because of the latent potential of being able to molecularly self assemble devices. Using the bottom-up approach, molecules and atoms can be engineered through physical means to grow nanoscale structures. The main advantage of this method lies in the ability of chemists and material scientists to manipulate individual atoms or molecules such that they grow in a regular order. It is the short range orders to which these molecules can be grown that constitutes a problem; building long-range orders with molecules, are difficult to achieve because of breaks and disturbances in the formation process which results in higher defect rates. Due

to these facts, hybrid technology has been proposed as a potential solution; these hybrid devices will utilize the top-down approach in order to achieve a coarse pattern definition, while using the bottom-up approach to produce short range ordered nanoscale structures which align to the coarser but long range order [13] .

The bottom-up approach is primarily based on self-assembly processes. Self assembly can be defined as "a coordinated action of independent entities under local control of driving forces to produce larger, ordered structures or to achieve a desired group effect" [13].

1.2.1 Building Blocks

One-dimensional nanostructures are the smallest dimensional structures that can be utilized for efficiently transporting electrical carriers [16]. Another important attribute of one-dimensional structures, is their ability to function as interconnect wiring as well as devices in nanoelectronics systems. In particular, two structures have exhibited widespread promise; they are Carbon Nanotubes and Semiconductor Nanowires [16].

1.2.2 Carbon Nanotubes

Carbon Nanotubes (CNT), first discovered by Sumio Ijima of the NEC Corporation, has shown tremendous promise as an enabler of nanotechnology. A CNT can be visualized as a sheet of graphene that is rolled up into a hollow cylindrical structure. CNTs measure approximately 1 – 10 nm in diameter [13]. There are two types of CNTs.

1 Single Wall Nanotube: Single sheet of graphene wrapped up in a cylindrical

geometry.

2 Multi-wall Nanotube: As the name suggest, multiple graphene sheets in a cylindrical form, are tightly stuck to each other.

1.2.3 Geometrical Structure

Carbon Nanotubes are described by the full circumference of their tube known referred to as their chiral vector C_h, which is defined by equation (1-3):

$$C_h = na_1 + ma_2 \tag{1-3}$$

a_1 and a_2 represent the unit vector of the CNTs hexagonal lattice, while n and m are integers. The chiral vector also defines the periodicity of the tube parallel to its axis, this is also known as its propagation vector. The chiral angle of CNTs is denoted by the angle between C_h and a_1. In the case where n or m is zero, the chiral angle will be zero degrees and the structure is called *zig-zag*. When the n=m and, the chiral angle is 30° the structure is referred to as *arm chair*. The other structures that lay between 0° and 30° are called chiral Nanotubes.

1.2.4 Carbon Nanotubes as Interconnects

Scaling the width of interconnect wires, increase their resistance. This is as much a result of reduced cross sectional area as it is a result of scattering from the surface and the grain boundaries [13]. This scattering could be avoided by constructing wires that do not possess any inherent defects and have perfect surfaces.

Carbon Nanotubes are capable of meeting a large portion of this requirement. They possess unique translational symmetry in one direction with an innately flawless surface. Metallic Nanotubes have high electron density in addition conduction is also easy in its tubular axis. Also, electron transport has been shown [17] to be ballistic, within the electron-phonon scattering lengths, which have micron dimensions at room temperature [13]. Ballistic transport in this case, implies that no scattering occurs within electron-phonon scattering length. Power dissipated will be isolated to the nanotube contact region if the length of the wire does not exceed this scattering length. The elimination of scattering is advantageous as it allows a higher degree of current densities than that allowed in metals. Carbon Nanotubes have been demonstrated [18] to have current densities of up to 10^{10} A/cm^2. This is significantly better than in copper interconnects which have current densities of approximately 10^7 A/cm^2 [13].

1.2.5 Signal Propagation in Carbon Nanotubes

The time required for a signal to propagate through an ohmic wire is determined by the velocity at which an electromagnetic wave travels trough a dielectric. The signal rise time is dependent on such parameters as resistance, capacitance and the inductance of the wire.

The differences between an ohmic wire and a nanotube can be estimated if the nanotube is modeled as a wire with a length-independent resistance and a capacity which is altered due to the electrochemical capacity to account for the interference exerted on the nanotube by the electric field [19]. It is possible to infer the delay of ohmic and nanotube interconnects by approximating their capacitance in the coaxial cylinder

configuration, while neglecting inductance and the interaction between the drive and load transistors [20]

1.2.6 Memory Applications for Carbon Nanotubes

In the post Moore's law world, it would be very important for memory devices to have high storage densities, be capable of random data access, operate at high speeds, consume less power, inexpensive and practical, be easy to integrate into existing integrated circuit devices and be non-volatile.

CNTs could be used to build CNT-SRAMs. Already, NOR gates have been fabricated with CNTs and have been demonstrated [13]. Hence, it is possible to use them in combination with resistors to build an SRAM storage unit by cross coupling their inputs to their outputs.

Crossbar nanomemories using CNTs have also be demonstrated by Rueckes and Lieber et al [21]. The crossbar CNTs are constructed, as the name suggests, using a grid-like construct of CNTs, arranged in a regular periodic array. The intersections of two CNTs form the active storage devices of the memory. The upper CNT wires have two stable positions, one in which the they are in their minimum elastic energy positions, separate from the lower CNT wires, the other is when the upper and lower CNT wires are held in contact by Van der Waals forces, Figure 2 illustrates. The Crossed CNT wires alternate between states when a voltage is applied; that is they break their contact when driven by an applied voltage and they also reestablish their connection when they are in their high resistance separated state and a driving voltage is applied. In this manner they form the desired non-volatile on and off switch necessary for bit storage. Periphery

devices such as a sensing matrix, just as is used in CMOS devices can be used for reading and writing the bits. Lieber et al estimate that this device would have a minimum cell size of 25 nm^2 that is the minimum storage cell would have a square geometry of lengths 5 nm [13]. This dimension would yield a memory packing density of 10^{12} elements/cm^2. The switching time was intrinsically approximated to be 100 GHz.

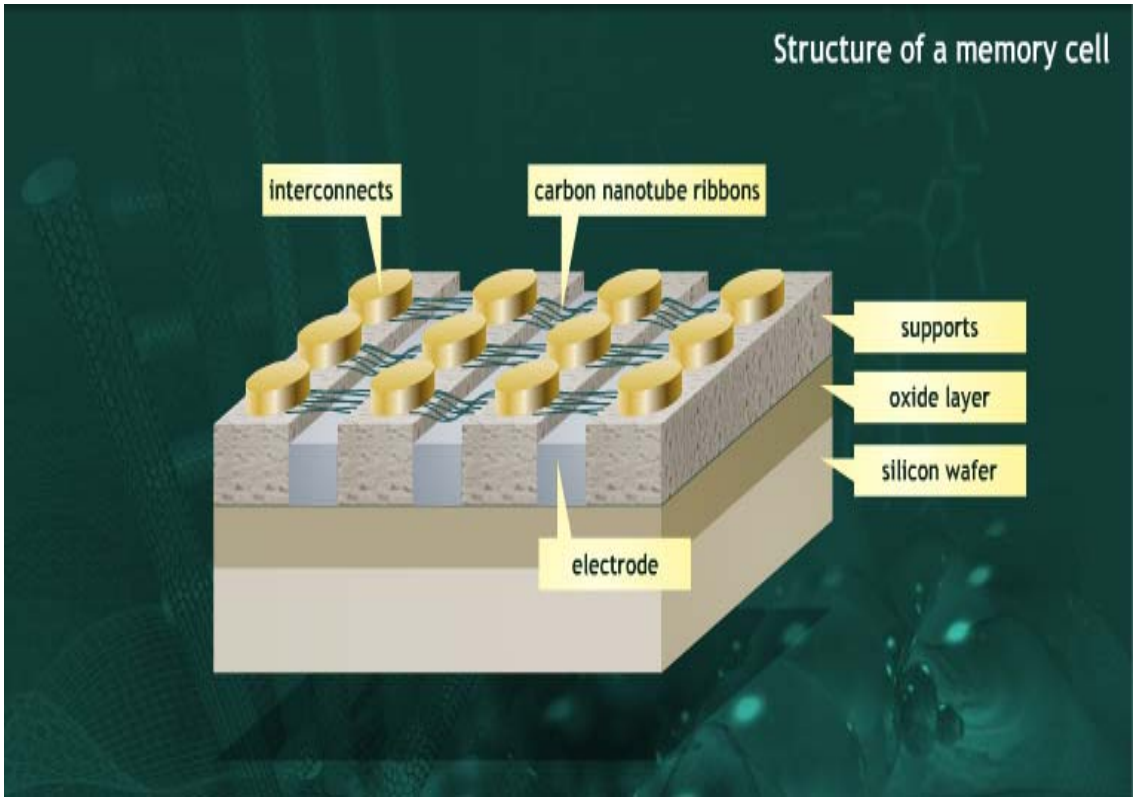

Figure 2. An NRAM (Nanotube-based/Non-volatile RAM) [22].

1.2.7 Semiconductor Nanowires

Semiconductor nanowires represent another class of wires with nanometer geometry that that have found applications in nanomemory devices. They differ from nanotubes in that they can be rationally and predictably synthesized into a single crystal [16] by controlling their chemical composition, length, diameter, and doping during growth. Size, predictability of growth, interfacial and electronic properties, are among the many properties of nanowires that make them appealing. These properties have made nanowires an important enabling material for a wide range of device integration. Nanowires are also the best controlled building blocks for devices; they can also be manipulated in ways foreign to conventional electronics, to make devices with new functions [23, 24]. Nanowires have already been fashioned into nanoscale FETs [21, 11], Light emitting diodes p-n diodes [25, 26] they have also been used to construct logic devices for computational circuits [27].

1.2.8 Synthesis of Semiconductor Nanowires

In the synthesis of nanowires, there is a requirement that two of its dimensions be in the nanometer regime, while the third dimension should be on a macro-scale. Growing nanowires to macroscopic length possess the most challenge to material engineers. An important concept used to spur the growth of one-dimensional nanowires uses a technique called Vapor-Liquid-Solid (VLS) growth mechanism to induce a so called catalytic growth [28, 29]. The catalyst defines the diameter of the nanowire and can be viewed as a nano-cluster site which serves to provide preferential direction to the addition of reactants to the end of the nanowires. This synthetic concept provides the knowledge needed for

the specification of the catalyst and the growth condition necessary to achieve predictable nanowire growth.

First, a catalyst material that that forms a liquid alloy with the nanowire material of choice is chosen with the aid of an equilibrium phase diagram. It also serves to determine the precise composition and growth temperature required, such that there is coexistence between the liquid alloy and nanowire phase. The liquid catalyst alloy cluster works as a favorable site for reactant absorption [16]. The preferred one-dimensional growth takes place in the presence of reactant only in the case where the catalyst nano-droplet stays in its liquid phase. With the previously mentioned framework, the synthesis of the nanowires with different diameter and composition, becomes a straight forward process as long as the appropriate nanometer scale diameter catalyst are available [16].

Other methods which utilize this framework include Laser assisted Catalytic Growth (LCG), which uses laser ablation to simultaneously generate nanoscale metal catalyst clusters and semiconductor reactant that produce nanowires using the VLS growth mechanism [16]; the advantage of this approach lies in the flexibility and generality provided by the laser ablation process. Another important derivative of the catalytic growth frame work is the Metal-catalyzed Chemical Vapor Deposition (CVD) process. This nano-cluster catalyzed (CVD) utilizes well defined gas sources [29, 16] and can be viewed as an alternative to the LCG implementation of the catalytic growth [16]. The advantage of this method is that it enables the nanowire size, composition and doping levels to be controlled in a precise way [30].

1.2.9 Electrical Transport in Semiconductor Nanowires

The electrical properties of Nanowires can be determined by fabricating a Field Effect Transistor (FET) out of the nanowire, Figure 3 provides an illustration of the NW-FET. The FET is constructed using a nanowire to connect two metal electrodes, which function as the source and the drain of the transistor, and are then supported on an oxidized silicon substrate with underlying conducting silicon to function as the global back gate electrode so that the electrostatic potential of the nanowire can be controlled. Current versus voltage (I-V curve) curves of the NW-FET are measured to characterize the electrical properties of the NW. The I-V curve is measured independently with respect to the source-drain and gate voltage [16].

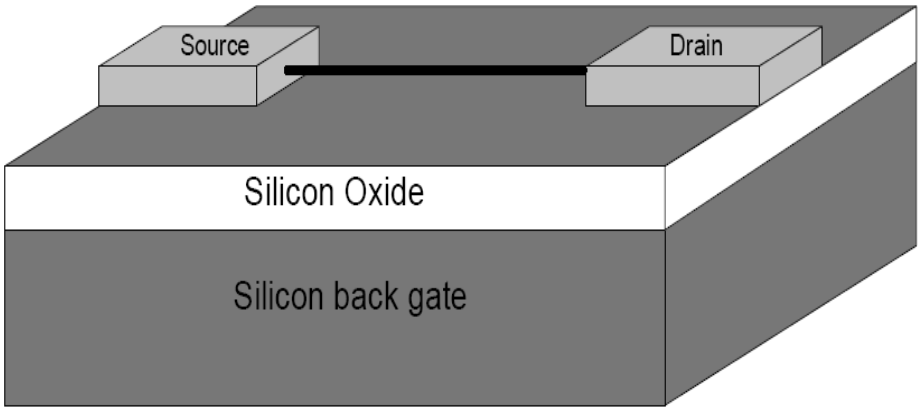

Figure 3. Nanowire Field Effect Transistor (NW-FET) schematic. A nanowire is placed between two electrodes which fuction as the source and drain.

The nanowire charge is given by $Q = C \cdot V_{th,}$ where C denotes the nanowire capacitance and V_{th} is the gate threshold voltage needed for complete nanowire depletion.

The capacitance C, of the nanowire can be expressed as $C \approx (2\pi\varepsilon\varepsilon_0 L)/\ln(2h/r)$, where ε is the effective gate oxide dielectric constant, h gives the thickness of the SiO_2 layer on the substrate, r denotes the nanowire radius and L the length of the nanowire.

1.3 Objectives and Contributions of This Dissertation

This dissertation is focused primarily on the defect and fault tolerance of nanoelectronics memories built on a regular array or crossbar geometry. The consensus in the field of nanotechnology is that the fundamental building blocks of nanoscale devices are highly defect prone in addition to being susceptible to transient ant operational faults. In this work we develop a defect and fault-tolerant scheme called the Multi-Switch Junction scheme, and we analyze its potential for improving the reliability of crossbar nanoelectronics memories. In this work, I also studied the effect of implementing fault-tolerant architecture in nanomemories, on the delay and power dissipation performance of the nanomemories. Other fault-tolerant schemes such as ECC and banking were also evaluated as well as their performance impacts on nanomemories. A trade-off analysis was also conducted to assess the cost of achieving reliability, on the access time and power dissipation performance.

CHAPTER II

CROSSBAR NANOMEMORIES

2.1 Introduction

Crossbar architecture has advantages that make it attractive for use in the fabrication of nanoelectronic memories. The crossbar geometry lends itself well to configurability and is among the easiest computational structures to fabricate at nanoscale dimensions [15]. The crossbar geometry is also considered to be the highest-density two-dimensional digital circuit topology for which every device can be independently addressed [31, 32]. The work presented in this thesis is centered on the study of these crossbar nanoelectronic memories. Conceptual designs of nanoelectronic based molecular computational platforms have been proposed based on the potential of nanotechnology.

The term "crossbar" signifies the geometry and architecture of the nanomemory device. The crossbar nanomemory consists of equal set of parallel nanowires which intersect each other at perpendicular angles, much like equally spaced grid of wires which intersect each other at 90˚. The overlapping nanowire grid is separated by a monolayer of bistable-molecules [33], the points at which the nanowires intersect are referred to as the junctions. The 'nanowire—bistable-molecules—nanowire' junction configuration constitutes a two terminal molecular switch, whose nature is governed by the composition of the nanowires and bistable-molecules. Specifically, the bistable-

molecules can be synthesized to function as two terminal reconfigurable active devices such as resistors; bistable-molecules can change their resistive state to a high or low value in response to an applied voltage field.

The transition from traditional computational platforms comprised of such elements as logic devices, multiplexers/demultiplexers etc, to practical non-traditional computational platforms designed from nanoelectronic elements. Current trends in the development of crossbar nanoelectronic memories indicate that they can potentially be used to fabricate much denser circuitry than current CMOS technology. This is in large part due to the steady progress being made in nanoscale fabrication techniques [34], for which the goal remains the development of nanoscale fabrication techniques that ultimately approach the feature sizes and densities characteristic of macromolecules [35]. The chronology of progress made in the increase in achievable bit densities in crossbar nanomemories has taken the path illustrated in Figure 4. Bit densities have increased from a single bit proof of concept in 2002 to the fabrication of a 160 kbit nanomemory with a bit density of 10^{11} bits/cm^2 in 2007 [31]. This improvement is a natural corollary of the progress realized from advancements in fabrication techniques as technology has moved from traditional patterning techniques like photolithography, to Electron Beam Lithography (EBL) to the more advanced Superlattice Nanowire Pattern transfer (SNAP). The 160 kbit crossbar nanomemory was fabricated using the SNAP technique.

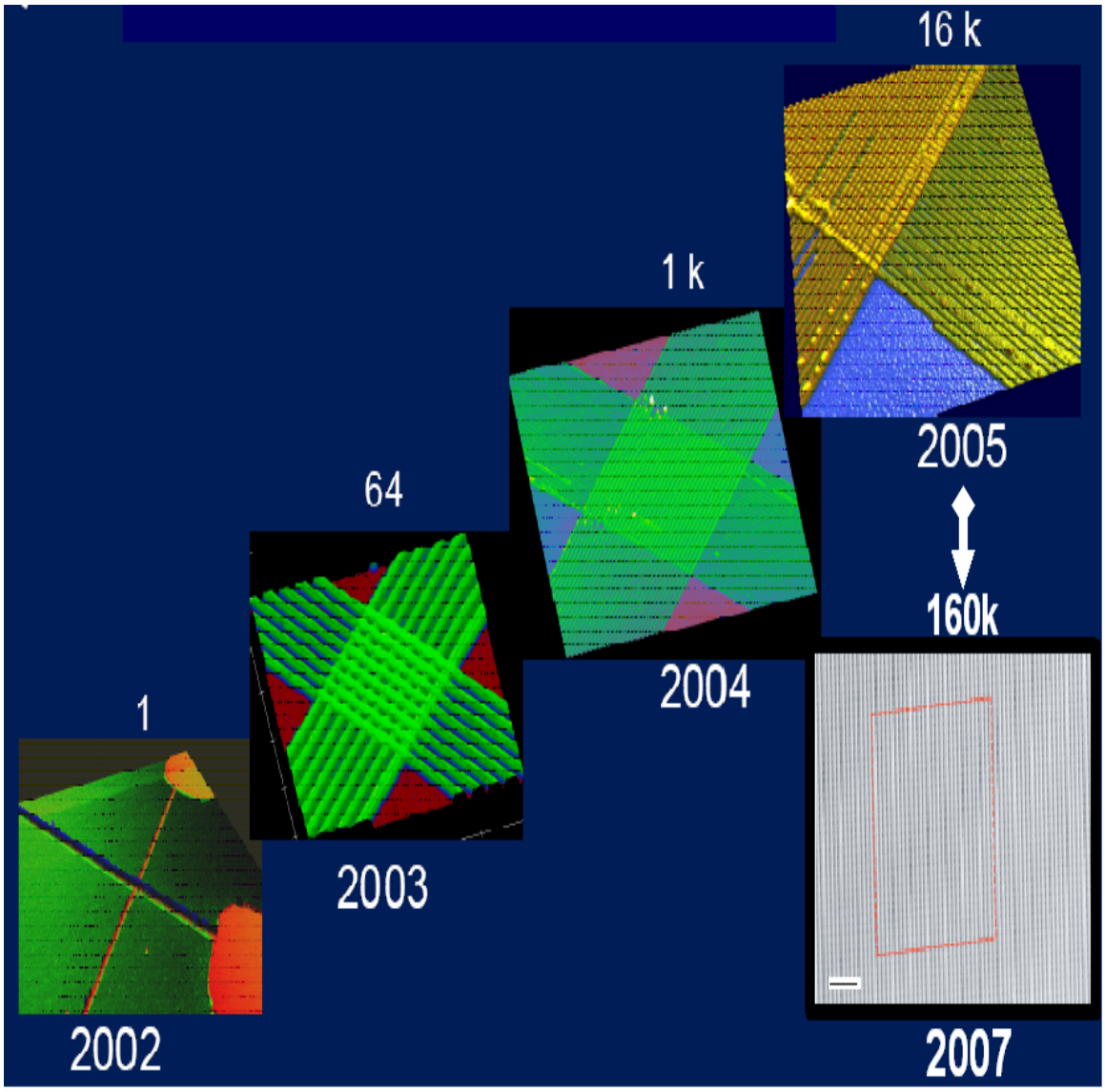

Figure 4. Development of crossbar nanomemory technology from 2002 – 2007. The image illustrated in the figure labeled 2002 image, is a proof of concept Atomic Force Microscope (AFM) image of intersecting nanowires. In 2003, nanotechnological fabrication processes improved, and scientists and engineers reported [1] the fabrication of a 64 bit crossbar molecular switch memory. In 2004 and 2005 more strides were made and a 1 Kilobit [46] and 16 Kilobit respectively, crossbar molecular switch nanomemory were fabricated. Recently, in 2007, the fabrication of the largest crossbar nanomemory to date, with a potential storage capacity of 160 Kilobit was demostrated and reported in [31]. Elements of this figure were taken from [1, 31].

2.2 Memory Architecture

Memories can be classified according their intended function. This function dictates the memory size, access time required to read or write data, the access pattern, and the requirements of the system being utilized.

In designing a computer memory there are certain performance metrics that help quantify its efficiency, access time or signal delay is an example of such a metric. Access time can be divided into to two functions, read and write access time. Read access time refers to the delay between the time a read request is initiated and the instance the data is available at the output. Likewise the write access time is the time delay between when a write is requested and the final time the data is written into the memory. The cycle time of these two processes, which refers to the minimum time required between successive read or writes [36], is also an important metric in delay analysis. The cycle time is usually greater than the access time.

2.2.1 Memory Organization

Memory systems just as in semiconductor memories can be classified based on their functionalities into Read Only Memory (ROM) or Read Write Memory (RWM). As the name imply, ROMs are memories that can only be written; data is hard wired into the memory circuitry and thus can only be read. RWMs on the other hand are memories that can be read and written; these memories are the most flexible kind and they offer comparable read and write access times. RWMs are the memories we are most concerned with implementing using nanotechnology methodologies. RWMs can be classified as either static or dynamic, depending on if they are stored in flip-flops or charge capacitors,

respectively. Static memories are able to store data as long as there is a constant supply of voltage. Dynamic memories on the other hand require only a periodic refreshment or cycle through of voltage in order to retain their data. Molecular switch nanomemories (MSNM) can be classified as dynamic memories since they have similar operating principles for data storage. They however defer from semiconductor dynamic memories in that they do not require frequent periodic supplies of voltage to hold their states; in addition they are also non-volatile memories possessing the ability to hold their states in the absence of a voltage supply just as in ROMs. We can thus view MSNMs as a hybrid of memory consisting of properties akin to ROMs and RWMs. The non-volatile RWM (NVRWM) is an example of such a hybrid memory.

Another important memory classification is the data retrieval pattern. MSNMs belong to a class of memories called Random Access Memories (RAM). As the name implies, memories of this class do not have to be written or read in any specific order, they can be accessed randomly. Majority of computer systems today make use of two kinds of RAMs; Static RAM (SRAM), and Dynamic RAM (DRAM). SRAMs are comprised of up to six transistors which are configured as cross coupled inverters; DRAMs as mentioned earlier is made up of a storage capacitor and up to two transistors. SRAMs are typically faster than DRAM because they require no refreshing but are more expensive because of their space requirements. They are usually used as on-chip caches because of their speed, but are limited in size because of the shortage of real estate on the processor. DRAMs are less expensive and have the distinct advantage over SRAM of being non-volatile. They are deployed as off chip caches with relatively larger storage

capacities.

Other retrieval patterns include FIFO (First In First Out), LIFO (Last In First Out), Content Addressable Memory (CAM) and Shift Registers. These patterns offer such advantages as faster access time, memory area management and other specific functionalities [36]. All these are among the many properties that are drawing computer architects and engineers into the vast array of possibilities of nanotechnology as an important enabling technology.

2.2.2 Memory Architecture

A major driver for the implementation of nanotechnology in memory devices is the relative simplicity of their structure. Memories are organized into grid-like storage cells; this is an important property as the most promising nanomemories are constructed using a crossbar architecture. CMOS memories have operating principles that will influence the way nanomemories are designed, not only because of their robustness but because of the large knowledge base that complements the technology.

In CMOS, memories are designed as N-word memories, with each word having M-bits. Each word can have single or multiple ports for reading and writing bits. The number of ports could be very large depending on the memory size. Memory capacities can typically possess over a million words; this poses a problem as it becomes somewhat impractical to implement a million signals and interconnects in a memory module due to packaging and wiring constraints. N-word memories are implemented by stacking subsequent memory words in a linear manner. The N cells are each M bit wide. Each of the N-word memory cells are selected for reading or writing by using a select bit,

assuming each memory cell has a single port; Figure (5a) illustrates a memory that uses a single input port for assessing each word.Having a single port translates into having a single signal be high at any time. This creates problems; if the memory is comprised of 1 million cells, then we would need to have 1 million select signals coming from off-chip locations, and this requires a great deal of wiring amongst other things. To alleviate this issue, a decoder circuit is implemented next to the memory structure to reduce the number of select signals. Memory words are chosen by providing an encoded address word address (represented as A_0 to A_{K-1} in Figure (5b) which the decoder translates into $N = 2^K$ ($K = \log_2 N$, and denotes the number of encoded address words sent to the decoder) select lines, and at each point in time only one select line is active. This drastically reduces the number of select signals from 1 million to $\log_2 2^{20} = 20$.

There are two types of decoders, the Row decoder and the Column decoder. The Row decoder is used to enable one row of the memory for Read/Write, while the column decoder picks a specific word from the selected row. In computer architecture, the select line is usually referred to as the word-line while the wire used in the connection of a single column to the input/output circuitry of the column decoder, is called the bit-line.

Designing the memory so that the rows and columns are of the same dimensions is also important. The word line height divided by bit line width is referred to as the aspect ratio [36] of the memory. The closer the aspect ratio is to unity, the more efficient the memory is expected to be. The reason being that delay increases at the very least, linearly with length. Hence the most optimal configuration for interconnects in a memory module would be to have equal word-line and bit line dimensions. Multiple words are

typically stored on the same word-line; each word is distinguished via a column decoder (illustrated in Figure 6). This aids in preserving a close to unity aspect ratio.

Transistor count is a factor in memory design. To reduce the transistor count of a single semiconductor memory cell, certain desired digital circuitry properties are traded-off; they include noise margin, logic swing, input/output isolation, fan-out or speed [36]. As a result of the tightly controlled confined domain of the memory core, degradation of these properties could be sacrificed. However, such sacrifices are not feasible when interfacing with external or surrounding circuitry. Therefore these digital signal properties must be recovered with the aid of peripheral circuitry. For example, interfacing with the external world requires the amplification of swings of internal signals to full rail to rail amplitude. The signals are low because the swing voltages of bit lines are set below supply voltage to reduce power delay and power consumption. Amplification is achieved with the aid of sense amplifiers. The diagram below illustrates the array structured memory organization [36]. The other peripheral circuitry includes the decoder, input/output buffers and the control/timing circuitry.

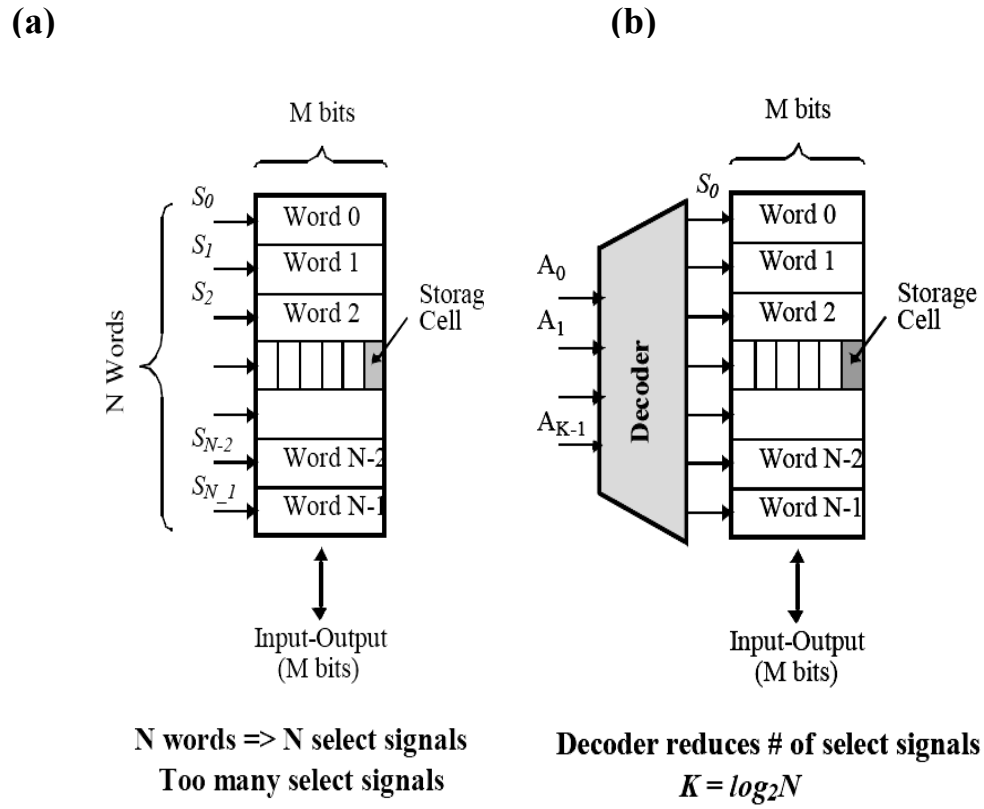

Figure 5. (a) Illustrates a memory that uses a single input port for assessing each word. Memory capacities can typically possess over a million words, hence, their configuration is optimized by implementing equal word-line and bit-line dimensions. (b) Multiple words are typically stored on the same word line; each word is distinguished via a column decoder. This aids in preserving a close to unity aspect ratio. Diagram was taken from [36].

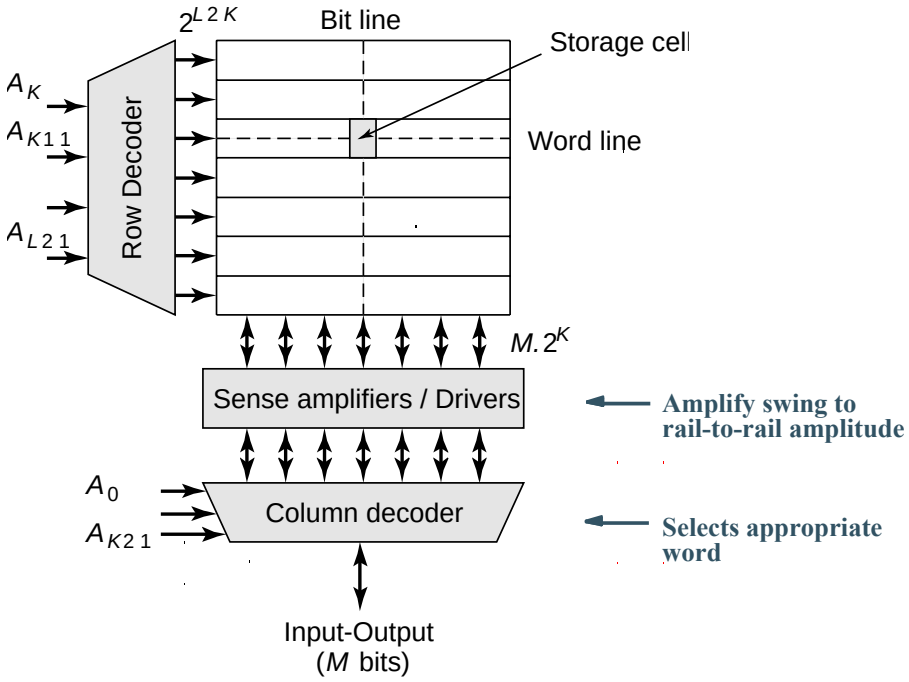

Figure 6. Array structured memory architecture, taken from [36].

2.2.3 Nanocomputer Architecture

The design of nanocomputers is presently focused on the design and fabrication of nano-circuit components, their optimization and the development of adequate architectures to realize them. The methods and algorithms required for various computer operations vary from operation to operation. As a result, speed, robustness, accuracy and other performance related metrics vary as well [37]. Hence, there is a direct correlation between the fabrication technology required to make integrated circuit, be it micro or nano in dimension, and the computational performance.

Drawing from computer architectures based on microelectronic devices, it is

possible to envisage the possible design flow of a nanocomputer. Basically, a nanocomputer architecture will consist of the integration of functional, interconnected hardware units and systems that perform the propagation, execution and processing of data. They will also be able to manipulate and utilize both digital and analog inputs to conduct various operations; by using a novel or pre-existing instruction set architecture.

In order to draw parallels between microelectronic and nanoelectronics based computers, we will first summarize the operational design flow of microelectronics computer architecture. In microelectronic computers, data is first input into digital computers via electromechanical devices such as mouses', keyboards, LCDs touch screens and so on. The data received is then stored in the computers cache memory after which it is manipulated and processed to perform specific operations by the Arithmetic Logic Unit (ALU). Their results are then output through specified output devices such as computer screens, printers, firewires etc. The input/output units of the system are usually referred to as I/O units. The Central Processing Unit (CPU) of the computer is tasked with coordinating the manipulation, processing and storage of data. The memory cells, which are among the core components of the computer, are comprised of billions of lithographically fabricated transistors. They are organized into groups denoting a specific bit storage capacity called *word.* The length of these *words* usually varies from 16 to 64 bits.

We can classify computer memory into two categories, as earlier mentioned. Memories can either be volatile or nonvolatile. The memory units that reside on the computer chip are volatile in nature, meaning they cannot retain information in the

absence of power supply or applied voltage; this is where advantages of nanotechnology become apparent. Computer performance suffers due to the volatility of on-chip memories. The performance lag lies in the inability of the on-chip memories to retain information in the absence of power. Nanomemories are expected to be nonvolatile. This coupled with the dimensional advantage nanoelectronics possesses over microelectronics results in larger storage capacities.

2.3 Crossbar Molecular Switch Nanomemories

The fabrication of molecular electronics devices using crossbar schemes appears to be the most investigated and auspicious for nanotechnologies [38]. Various crossbar nanoelectronics devices have been investigated [7, 14, 38, 39]. Crossbar structures for molecular self assembly are fabricated with bottom-up processes and have been studied at the architectural and the circuit level of abstraction. The nano-components are self assembled in a symmetrically ordered fashion that lends itself extensively to the crossbar architecture. The Crossbar scheme is an array based architecture built from molecular scaled wires, such as Carbon Nanotubes or Silicon Nanowires (SiNW) [40]. The molecular scaled wires are interconnected in cross arrays with switching devices which are non-volatile at their cross-points. Crossbar schemes garner their attraction from the fact that they are reconfigurable, which helps in developing defect tolerant devices; they also posses programmable logic functionalities much like FPGAs.

A nonvolatile random access memory (RAM), implemented with nanoscale molecular-switch crossbar arrays [1, 14] has shown very good potential as practical nanoscale memories. As a result, this implementation has been used as the basis for the

fault tolerance and performance analysis researched in this thesis. The crossbar molecular switch memory designed in [1] has a density of 6.4Gbits/cm^2 and was constructed using an 8×8 array of nanowires. The molecular switches of the crossbar memory array were fabricated using a monolayer of Rotaxane molecules sandwiched between metal nanowires, as illustrated in Figure 7.

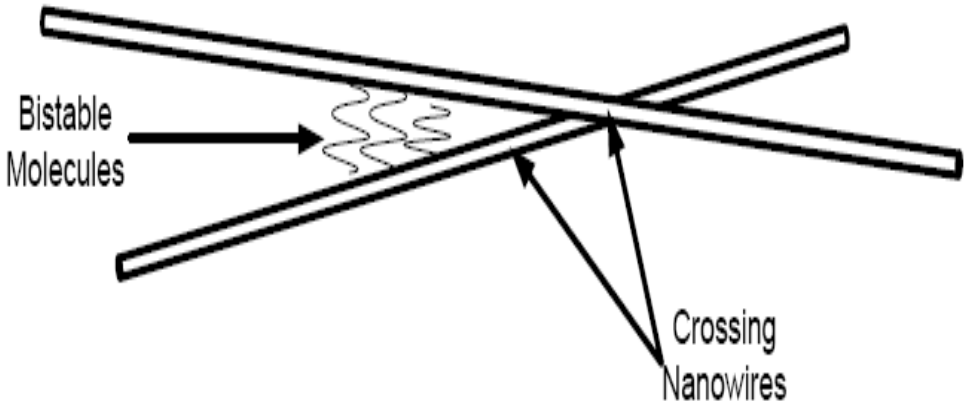

Figure 7. The intersection of two nanowires, with bistable molecules deposited at their intersection. This forms the building block moleciular switches of the nanoelectronic memory.

The high densities of nanoelectronics devices will mean a high rate of defects due to statistical variations. It is estimated that as much as 10% of computational resources in molecular electronics will be defective [41]. We then face a comparative problem to that encountered in amorphous computing [42], where the issue is how to create a coherent

and reliable system out of unreliable or defective parts.

The Teramac computer (Tera Multiple Computer Architecture) [7] embodies the definition of reconfigurable computing. The Teramac is a custom computer that is designed for architectural exploration and is based on FPGAs; it is also scalable and is able to run a million gate users designs at one megahertz. Despite the fact that three quarters (75%) of the FPGAs that make up the Teramac are defective, the system is still able to function correctly; Teramac uses complex interconnection networks to work around malfunctioning components, by so doing it is able to tolerate these defective components. Reconfiguration works in the following manner; A diagnostic test is first used to locate defective components such as wires and gates in the system, they are noted and the information stored in a database, so that the system can map and work around these defects.

The defective components anticipated in nano devices can also be circumvented by using other fault tolerance techniques, such as redundancy in the form of extra rows and columns in memory chips [43]. Redundancy is used to tolerate transient defects; this is accomplished by having two or more chips or systems operating in parallel so that their output can be compared and the majority gate taken as the right output [43]. The use of redundancy is a feasible choice because of the large number of nano devices that can be manufactured relatively cheaply through molecular self assembly.

2.3.1 Trends in Nanoscale Molecular-Switch Crossbar Circuit Fabrication

The earliest reports of a fabricated and tested crossbar molecular switch memory consisted of an 8×8 crossbar nanomemory comprised of a molecular monolayer of [44]rotaxanes sandwiched between metallic nanowires [1], within a 1 μm^2 area. The molecular switches formed at the junction of the nanowires formed the active memory cells, and the circuits of the crossbar operated as a rewritable non-volatile memory, with a 6.4 Gbits cm^{-2} density, Figure 8 illustrates. The crossbar nanomemory was fabricated using imprint lithography. Active molecular switch devices formed at the junctions of the crossbar consisted of amphiphilic bistable [44]rotaxane molecules which were demonstrated to function as reversible, electrically toggled switches [1, 45]. A monolayer of the rotaxane molecules were sandwiched between a top and bottom Pt/Ti nanowires. Thus forming the basic element of the circuit, the Pt/rotaxane/Ti junction located at each cross point of the nanomemory, which functions as a nonvolatile reversible switch. The width of the nanowires were measured to be 40nm and a half pitch of 65nm, thus yielding active device junction areas of approximately 1600 nm^2, which translates to approximately 1100 rotaxane molecules sandwiched between the nanowire electrodes.

The 8×8 crossbar nanomemory was also used to demonstrate a demultiplexer/multiplexing decoder functionality. This was achieved by partitioning the 8×8 crossbar nanomemory into a single 4×4 crossbar nanomemory and two 4×4 decoders for multiplexing/demultiplexing by setting the resistances at specific junctions to control the horizontal nanowires and the others to control the vertical nanowires.

Subsequently, in 2004 Jung et al [46] reported the fabrication of a 34×34 crossbar

nanomemory circuit; it was developed using a single-layer UV-nanoimprint process. The 34×34 crossbar nanomemory yielded a density of 10Gbits cm^{-2} at a 50 nm half-pitch. To track changes the progress being made in semiconductor technologies, the metrics used are the separation or pitch between tightly packed adjacent wires contained in a DRAM circuit. Two years after the report by Jung et al, Yu et al [47] reported the fabrication of 30 nm pitch nanowire arrays with an average line width of 17 nm.

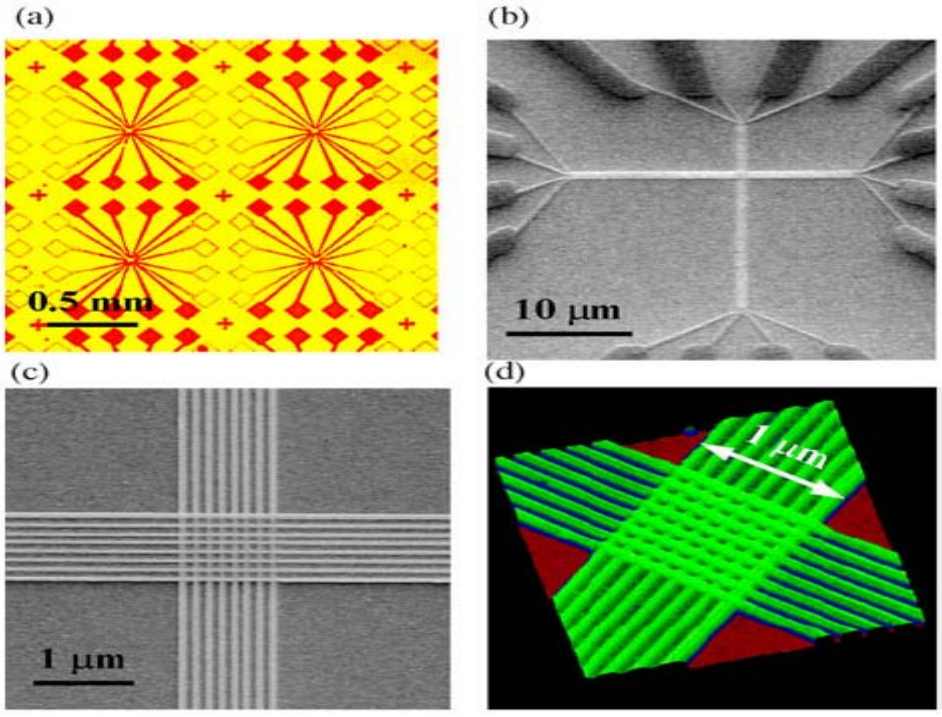

Figure 8. Detailed illustration of the crossbar molecular switch memory [1]. (a) shows an Optical microscope image of an array of 4 crossbar molecular switch circuits, each having 16 contact pads with micron-scaled scaled connections to nanoscale circuits in the center. (b) is a Scanning Electron Microscope (SEM) image showing two nanowire arrays, oriented perpendicularly to each other, and connected to their micron-scaled connections. (c) is an SEM image showing intersection nanowire arrays crossing each other at the central area and thus forming a crossbar. (d) is an Atomic Force Microscope (AFM) image of the cossbar molecular switch memory.

2.3.2 Design and Fabrication of a 160 kilobit Crossbar Nanomemory

Recently, a 160 kilobit molecular electronic memory patterned to yield 10^{11} bits/cm^2 has been reported [31]. The 160 kilobit nanomemory was fabricated at a pitch of 33 nm, and memory cell size 0.0011 μm^2; in contrast, the pitch of wires in modern DRAM circuits are 140 nm, and the memory cell are sized at 0.408 μm^2. The potential of crossbar nanomemories are further illuminated when consideration is given to the fact that the 160 kilobit crossbar nanomemory parameters are analogous to those expected for DRAM circuits in the year 2020 [31, 48].

The ultradense, highly aligned nanowire array of the 160 kilobit crossbar was fabricated using the Superlattice nanowire pattern transfer (SNAP) method. The crossbar nanomemory nanowires were comprised of 400 Si bottom nanowire electrodes with dimensions, 16 nm width, and 33 nm pitch. The Si nanowires were crossed by overlapping Ti top nanowire electrodes, with equivalent dimensions 16 nm width and 33 nm pitch. The storage elements were defined as molecular switch tunnel junctions (MSTJ), defined by a Si bottom nanowire and Ti top nanowire, sandwiched between them was a monolayer of approximately 100 [44]rotaxane molecules [31].

The 160 kilobit crossbar nanomemory was found to contain a large number of defects. The strategy for tolerating these defects consisted of identifying them through electronic testing and isolating them using software coding techniques. The working bits were then configured to form a fully functional working and operational read/write random access memory. When the 160 kilobit nanomemory was tested, several types of defects were discovered, one of which was “switch defects”; this defect was attributed to

sub-nanometer variations in the reactive ion etching process used to define the Ti top nanowire electrodes. Switch defects resulted in the proportional loss in the overall yield of operational bits. Another defect encountered was the "bad contact" or "shorted nanowire defect", this defect was more detrimental than the switch defect, as it led to the operational immobilization of entire row of bits.

The fabrication of the 160 kilobit crossbar nanomemory also brought to light the many scientific and engineering challenges that must be surmounted before these types of crossbar nanomemories can be practical. These challenges include device robustness, inventing better etching tools and achieving improved molecular device switching speeds among other things.

2.3.3 Crossbar Nanomemory Operation

In this section, the read/write operations of the crossbar nanomemory are explained. The write operation of a "1" bit to a molecular switch at the nanomemory junction is done by first selecting the row and column nanowires of the desired bit location, applying a voltage signal to the top (row) nanowire and grounding the bottom (column) nanowire. The row and column nanowires of the unselected bits are biased to a value which is half the write voltage to prevent the unintended writing of other bits. In order to read a bit, read voltage signal, which is lower than the write voltage signal, is applied to the row nanowire of the bit. All other rows and columns of unselected bits are grounded, and the resistance of the selected bit is read—which denotes its "1" or "0" value—is uniquely determined by measuring the current flow to ground from the bit's column nanowires. Figure 9 illustrates the write operation procedure.

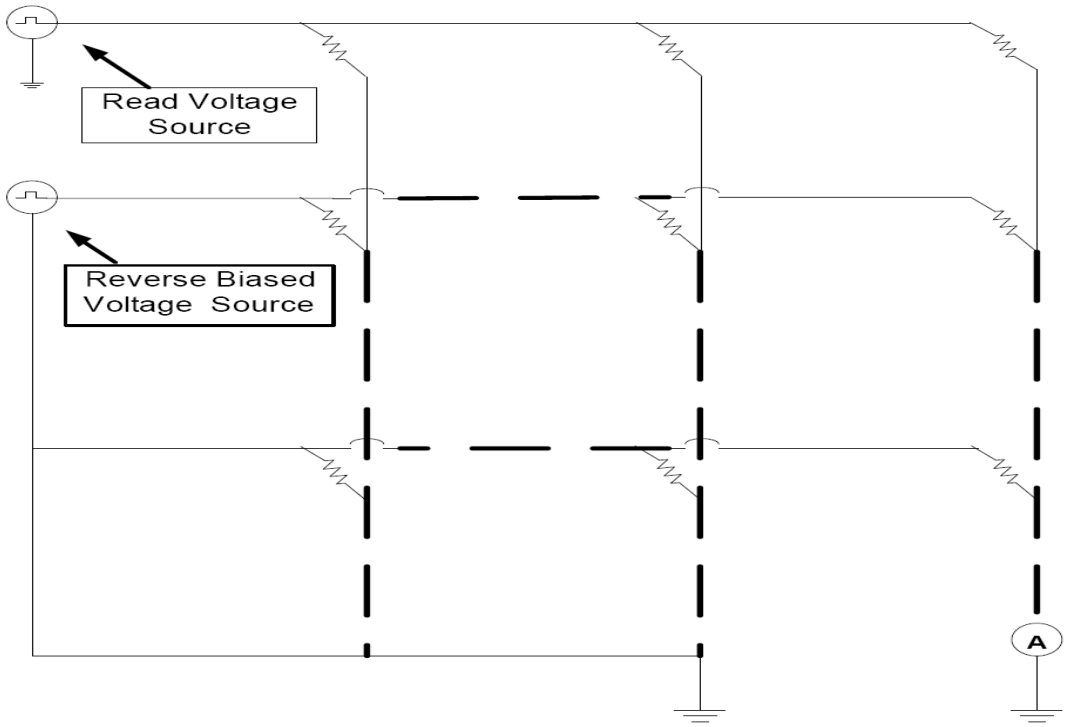

Figure 9. Write operation for a crossbar molecular switch nanomemory.

2.4 A Survey of Other Nanomemory Devices

Researchers have proposed the use of other nanoscale technologies for use as nanomemory RAMS. In this section a summary and literature review of the most widely studied potential nanomemory devices are provided.

2.4.1 Quantum Cellular Automata (QCA)

This device can be classified as a molecular electronic device. The QCA method of nano or molecular computing is dependent on the use of electrostatic field repulsion to transport information across the circuitry [49]. They can also be thought of as nanoelectronic circuits with computational circuit functionality which are realized through cooperative quantum mechanical and electrostatic interactions between electrons

confined in quantum dot arrays [50]. The basic block of the QCA is the quantum dot, which can be regarded as artificial atoms or "boxes for electrons" [49, 51] because they are similar to atomic systems that have discrete charge states and energy level structures.

The fundamental QCA cell is comprised of four quantum dots that are arranged in a square geometry as illustrated in Figure 9. Each individual QCA cell has two electrons which auto arrange themselves in a diagonal due to coulomb interaction [52]. Free electrons in the QCA cells can tunnel between dots but not between cells. It is this tunneling between the dots that enables the state of the cell to switch. This geometry can be suitably arranged into two configurations thus giving it bistable configuration functionality, Figure 10(A) illustrates. As a result they can thus be used to build logic gates as illustrated in Figure 10(B). Quantum dots can be conceptualized as electrically conducting regions that are small in dimensions such that their electron energies are quantized. They can also take different geometric and dimensional forms.

Quantum dots can be fabricated in a variety of ways such as in the depletion regions of multilayer semiconductor materials, tiny metallic islands connected through tunnel junctions or redox centers in particular molecules [50]. Under normal operation conditions the QCA cell will be close to, or in its ground-state configuration for the duration of a switching event. There are two factors which determine the QCA ground state configuration:

1. The coulomb interaction between the dots alters the electronic configurations of neighboring cells.
2. A clocking field alters the relative energy of the middle in-active and the

corner active dots [53].

Functional QCA devices have already been fabricated and reported. Orlov et al [54] fabricated a functional QCA cell using aluminum islands on the SiO_2 to construct the dots, which are coupled via aluminum oxide tunnel junctions and patterned capacitors. Other QCA devices constructed from tunnel junctions using shadow evaporation techniques, and function under cryogenic temperatures have also been demonstrated. Majority gates and clocked shift register circuits have also been fabricated using QCA cells [55-60]. These devices represent prototypes of molecular systems that will operate under room temperature conditions. Considerable progress has also been made in the construction of single-molecule QCA cell using mixed valence compounds [61, 62]. QCA half cells with two quantum dots [61] and four quantum dots [62] have also been synthesized.

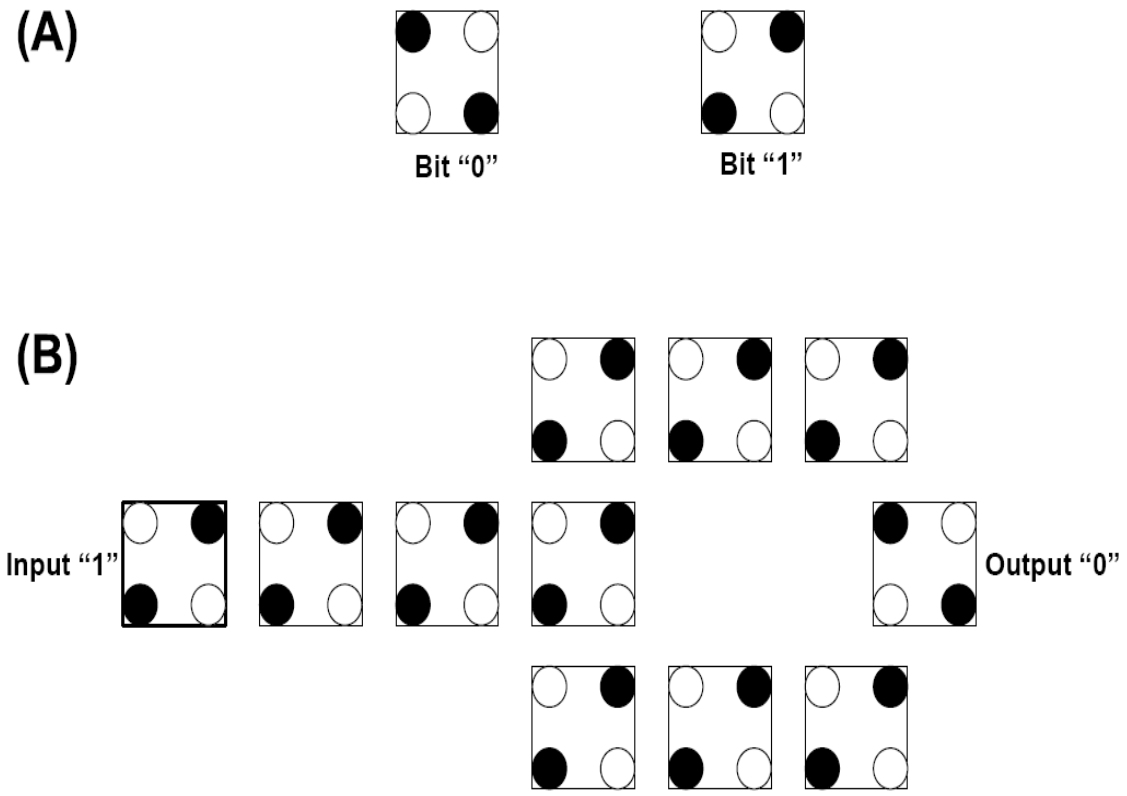

Figure 10. (A) Two QCA cells with four quantum dots in their two possible ground-state configurations representing binary "0" and "1" bits. (B) QCA cells used to demonstrate an inverter which takes an input logic "1" and yields an output logic "0".

2.4.2 Nanocells

In most proposed nanoelectronic architectures the precision arrangement and ordering of nanostructures, such as nanowires or molecules, are a core requirement for functionality. In addition, creating an adequate interface to microstructures is a necessity. The Nanocell approach to molecular and nanoelectronics electronics on the other hand does not dependent on the precise placement and orientation of nano elements [63]. A diagram illustrating the nanocell concept is provided in Figure 11.

A nanocell can be defined as a two or three-dimensional network of self assembled metallic particles connected by molecules that exhibit reprogrammable switching and or other memory properties such as negative differential resistance (NDR) [64, 65]. Also, the nanocell is conceptually based on the use of arrays of molecular switches to carry out logic functions, however there is no requirement for each switching molecule present in the nanocell to be individually powered or addressed [49]. Nanocells are fabricated by using the principle of chemical self assembly; this approach allows for the reduction in complexity and the expense of having a plethora of programming issues. The microelectronic interface to the nanocell is achieved through the use lithographically defined lead connected to the edges of the nanocell. The nanocell is internally comprised for the most part of disoriented switching elements. The nanocell is configured into desired logic devices by post-fabrication training, in a manner similar to the reconfiguration of a field programmable gate array (FPGA) [64].

Nanocells are potentially producible in high densities. They also have the potential for re-programmability throughout the computation process by altering their

ON/OFF switching states, as a result they can potentially function as real-time reconfigurable hard-wired logic. The properties of nanocells make them candidates for application in computer central processing units, where arrays of nanocells can potentially function as transistors working in tandem.

Several proof of concept of nanocells has already been reported. Tour et al [66] demonstrated the first nanocell device using disordered arrays of molecules and Au islands at room temperature; the nanocell device also exhibited reproducible switching properties and memory effects at room temperature [67]. Nanocell demonstrations using gold clusters and molecular self assembly have also been reported [68-73]. Seminario's group has also studied and reported the intrinsic characteristics of single molecules for application as programmable elements using high-level quantum chemistry methods; they have also theoretically demonstrated the possibility of transmitting signal through Molecular Electrostatic Potential (MEP) [67] of molecules.

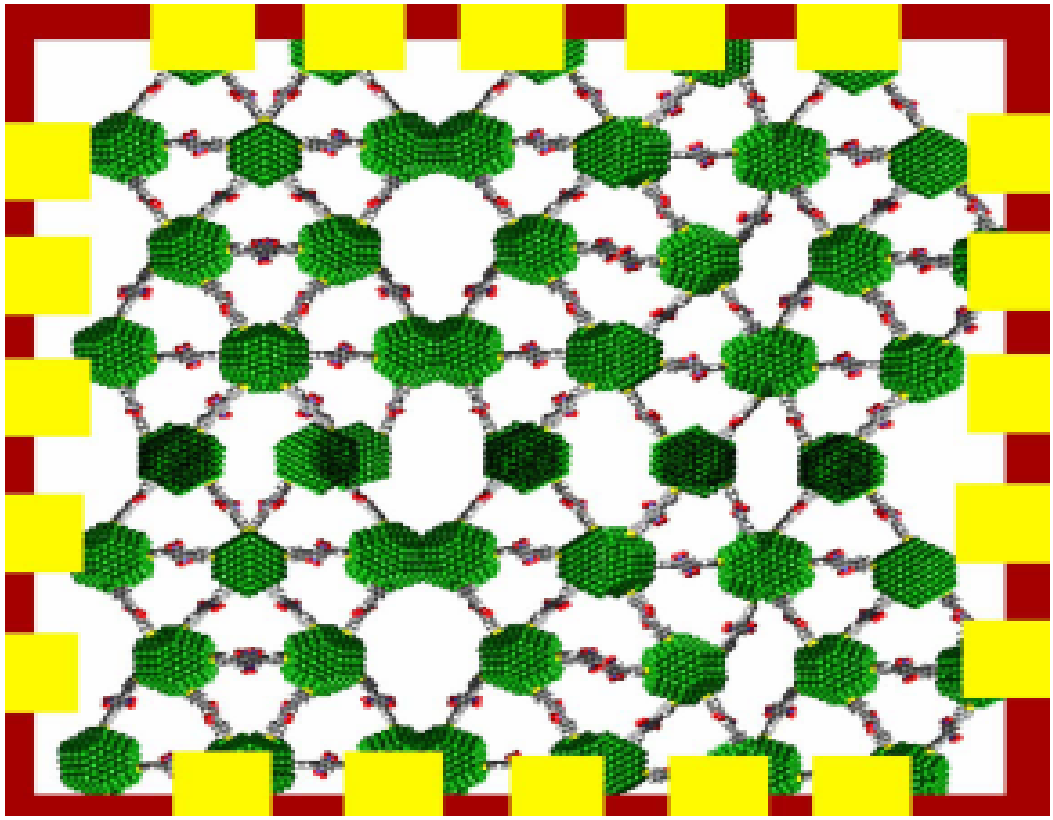

Figure 11. Image of a nanocell from [67] showing interconnected active molecules (green). The active molecules are accessed and programmed through the I/O leads (yellow squares).

2.5 Hybrid Microelectronic/Nanoelectronic Devices: Design and Architecture

Microelectronics has ushered in decades of technology growth and innovation. Microelectronics driven by CMOS silicon technology, which are highly organized inorganic structures designed for electronic charge and energy transduction [74-77]. CMOS technology allows for large device scaling densities and high operating speeds while dissipating low amounts of power. However, as the feature sizes of CMOS devices

are continuously scaled down, a variety of challenges are being encountered. It is becoming increasingly difficult to define smaller feature sizes using current top-down lithography fabrication techniques; sustaining a high degree of reliability in CMOS devices is another feat which is increasingly becoming more and more challenging.

A potential solution to the bottlenecks encountered in microelectronics is the use of Molecular/Nanoelectronics. Tour et al [78] defines molecular electronics as the use of single molecules or small groups of molecules such as wires, gain elements, switches etc, as the fundamental units for computing. Molecular electronics devices are fabricated using bottom-up processes, which means the incorporation of functionality into small features, such as molecules, in a manner that will potentially allow the molecules to further self assemble into higher ordered structural units such as transistors [78]. Bottom-up processes are less expensive to achieve, allow for the large density fabrication and diversity of molecular scale elements, and they are also expected to be more efficient that their silicon counter parts. On the other hand, more work is required for scientist to achieve better control over molecular level interactions.

As earlier mentioned, the objective is to build molecular electronics in an analogous manner to conventional silicon based electronics. A hybrid approach where microelectronics and molecular electronics are integrated is seen as the next logical route to achieving a paradigm shift from microelectronics to nanoelectronics. Crossbar molecular switch memories, as discussed in section 2.3**,** are an ideal example of a hybrid CMOS/Molecular electronic device. The overlapping crossbar nanowires are fabricated using the molecular self assembly Langmuir Blodgett technique [1]. The storage cells of

the crossbar molecular switch memory are comprised of molecular self assembled bistable Rotaxane molecules, which are sandwiched between the overlapping nanowires; other reported demonstrations have used bistable Catenane molecules [79]. Lithographically defined conventional CMOS circuitries are used as a bridge for interfacing and integrating the micro-scale and molecular electronic circuitry. In other words, external CMOS circuitry are used to address, program, and conduct read-out and erase operations on the crossbar molecular switch memory.

2.5.1 CMOL

CMOS/nanowires/MOLecular hybrid (CMOL) circuits have also been proposed [80-83] as potential hybrid CMOS/molecular electronic circuits. In the CMOL architecture, two-terminal nanodevices are formed at the junctions of overlapping intersecting nanowires, in a format similar to that of crossbar nanomemories. However, in contrast, the CMOS/molecular electronic interface of CMOL circuits is provided by perpendicular sharp-tip pins that are distributed all over the circuit area, on top of the CMOS stacks [83]. These sharp-tip pins connect with the intersecting crossbar nanowires above them, and when two of these perpendicular pins are activated, the two nanowires which make contact with them are connected to peripheral interfacing CMOS data lines. Each pin maintains a connection to exactly one nanowire, thus allowing CMOL to easily achieve junction configuration and optimize signal bandwidth between the CMOS and molecular electronic crossbar layer. The crossbar nanowires are oriented at an angle which is less than 90 degrees relative to the CMOS pin array. The crossbar nanowire array is oriented at an angle so that the nanowires do not need to be precisely

aligned with each other and the pins of the CMOS layer in order to be able to distinctly access a molecular or nano-device. Just as in crossbar molecular switch nanomemories, the molecular switches are turned ON/OFF by applying a switching voltage to the selected nanowires—which is greater than the threshold voltage of the other molecular switches in the crossbar nanowire array—so that only the intended molecular switch is accessed.

Each side of the angled CMOL crossbar nanowire grid, as illustrated in Figure 12(b), has a length of $2\beta\boldsymbol{F}_{cmos}$; where β is a dimensionless factor greater than 1 that depends on the CMOS cell complexity, and $\boldsymbol{F}_{cmos}$ is the half-pitch of the CMOS subsystem. The angle orientation of the crossbar nanowires relative to the CMOS pin array, is computed as angle $\alpha = \arcsin(2\beta\boldsymbol{F}_{nano}/\beta\boldsymbol{F}_{cmos})$, where $\boldsymbol{F}_{nano}$ denotes the half-pitch of the nanowires. Detailed analysis in which it is shown that this approach allows for the unique accessing of each molecular switch in CMOL—even in the case where $\boldsymbol{F}_{nano}$ is much less than $\boldsymbol{F}_{CMOS}$— is described in [81]. Placing the CMOS below the molecular electronic crossbar layer gives CMOL the added advantage of being able to distinctly separate configuration and data communication operations, as well as allow for large nano circuit densities.

Strukov and Likharev [83] report the use of CMOL field programmable gate arrays (FPGAs) as reconfigurable Boolean logic circuits, as the most important application of CMOL technology. Preliminary studies of CMOL FPGAs have demonstrated their potential to yield circuits that are two orders of magnitude denser than conventional FPGAs, and at comparable performance levels [83, 84]. In CMOL, CMOS

cells consisting of two pass transistors and an inverter are connected to the crossbar nanowires via two perpendicular pins, thus creating a configurable logic block equivalent to that found in conventional FPGAs. Programming the CMOS cell is achieved by disabling the inverter and selectively turning ON the molecular devices in the crossbar nanowire array. On completion of the configuration process, the pass transistors function as pull-down resistors while the molecular switches, which were previously programmed to be in the ON state, act as pull-up resistors. This allows for the formation of wire-NOR gates within the CMOS cell. The idea is for the CMOS cells to have a plethora of molecular or nano-devices; this allows for the creation of gates with both large fan-in and fan-out, or either of the two, with extra devices serving as "spares" for reconfiguration around defective devices. A Reconfigurable 32-bit adder and a 64-bit crossbar switch were simulated using Monte Carlo simulations, and it was shown that reconfiguration allowed for a 99% increase in circuit yields when the proportion of defective molecular or nano-devices were in the 20% range [85].

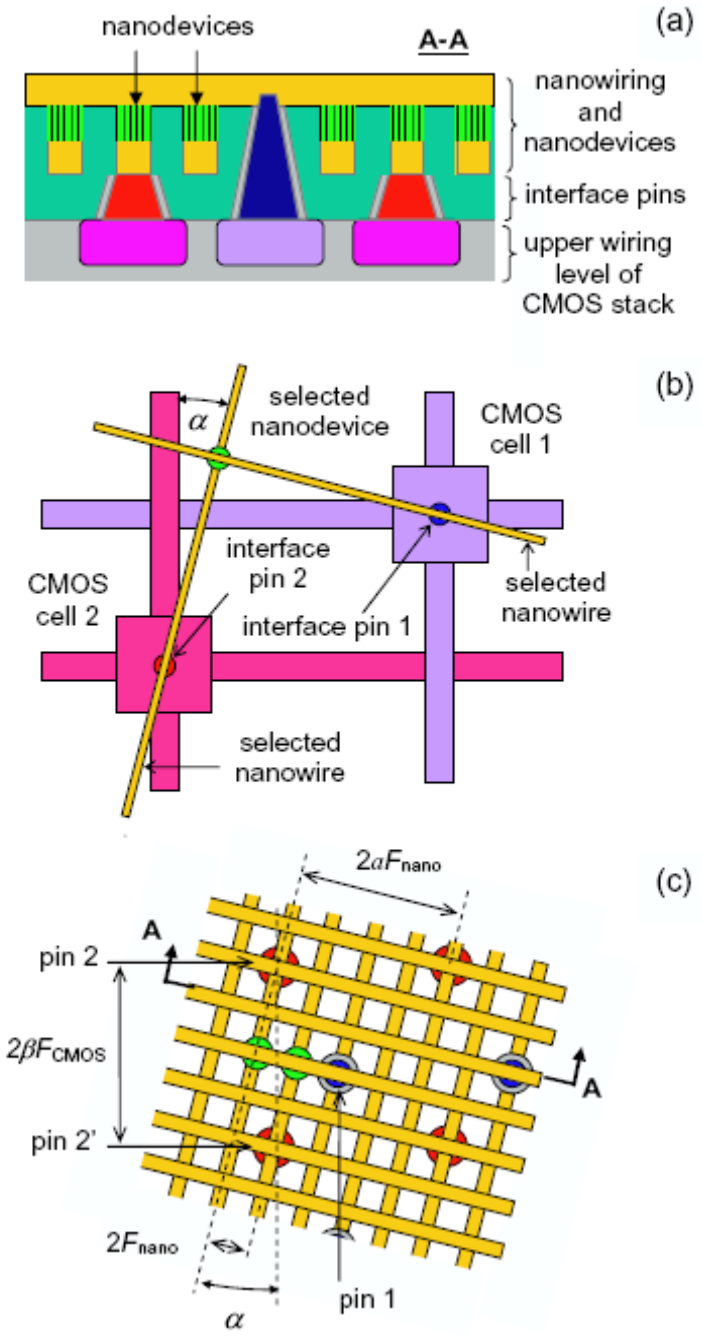

Figure 12. Illustration describing the lower level structure of the CMOL circuit architecture. (a) shows a cross sectional side view schematic. (b) illustrates two single pin contacts with two intersecting nanowires, used in addressing the molecular devices. (c) shows only two devices, as well as CMOS cells and wirings [83].

2.5.2 Field Programmable Nanowire Interconnect (FPNI)

FPNI is a generalized variation of the CMOS/molecular hybrid CMOL approach. FPNI, as reported by Snider et al, is a hybrid architecture that trades off a portion of the speed, areas density and defect-tolerance characteristic of CMOL in exchange for simpler fabrication, reduced power dissipation, and more latitude in the choices of nanoscale devices in the crossbar junctions [84]. In contrast with CMOL, the FPNI's improves upon the FPGA architecture by removing the configuration bits from the CMOS place, and implementing them in the nanowires plane as nonvolatile switches, which results in a reduction in area utility and power consumption. The major differences between FPNI and CMOL are illustrated in Figure 13, and can be described as follows [84];

The FPNI implements logic only at the CMOS level, and routing in the nanowires only. This allows for reduced static power dissipation, and use of approximate linear antifuses at for the nanowires junctions. The FPNI routing network is also made simpler by using a buffer based configuration instead of an inverter based configuration.

The crossbar nanowires array of the FPNI architecture requires alignment with the CMOS pin, but the accuracy of the alignment is on the order of the CMOS. The FPNI circuit can be fabricated using conventional CMOS processes and voltages. CMOL on the other hand requires reduced voltage supply and signaling swing, and nanoscale pins of different heights on a nonplanar silicon surface (FPNI provides a planar silicon surface for nanowires).

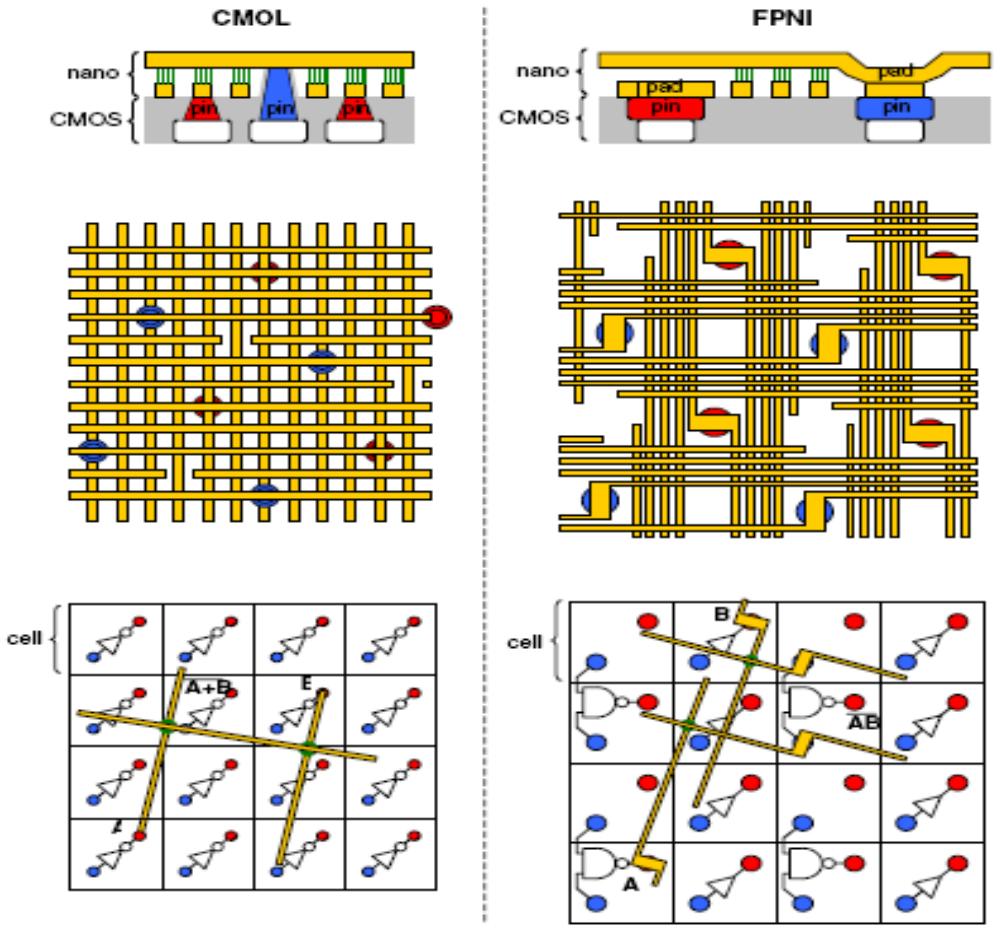

Figure 13. Description of the key differences between the CMOL and the FPNI architecture. The CMOL on the left shows the crossbar nanowires array placed above a layer of CMOS inverters. The crossbar has an angled orientation so that contact is made with only a single pin. The FPNI, shown on the right column, shows a sparse crossbar nanowires array placed over a layer of CMOS gates and buffers. The crossbar array also has an angled orientation for the same reasons as the CMOL case. Configured junctions in the FPNI are used for programmable interconnects only, with all logic implemented in CMOS [84].

Performance simulations of the FPNI chips were also reported by Snider et al [84]. Compilation of standard benchmark circuits were simulated onto models of the FPNI chip, results shows an 8 to 25 times reduction in area utility, lower power

consumption, and reduced clock speed, as compared with CMOL. A high degree of defect tolerance was also demonstrated by the FPNI; an FPNI chip with a 20% broken nanowire and 20% defective junction rate, produced a 75% effective yield with a non significant slowdown along the critical path, compared to an FPNI chip with no defects.

CHAPTER III

CROSSBAR NANOMEMORY DEMULTIPLEXERS

3.1 Introduction: Defect-Tolerant Crossbar Demultiplexers

Demultiplexers are electronic circuits which take combined input signals, separates them into their constituent signals, and yields an output with the desired signal. Demultiplexers are used in crossbar nanomemories as a means of bridging microelectronic scaled circuits to their nanoelectronic scaled counterparts. Nanoelectronics is thus encumbered with the challenge of developing reliable methodologies for addressing circuits that possess wire dimensions on a scale smaller than the resolution attainable through conventional lithographic processes.

In order to address individual nanowires fabricated via sublithograhic processes, the following architectural concepts should be satisfied [86].

1. The demultiplexer architecture must possess the capability of creating a connection or bridge between the microscale dimensions achievable through lithography, to the nanoscale dimensions achievable through bottom-up processes or alternative patterning techniques.
2. The architecture should have the ability to address many nanowires using only a few micron-scale wires.
3. The process used to manufacture the demultiplexer should be highly tolerant of defects which occur during fabrication.

A variety of demultiplexer architectures have already been proposed. They include concepts which combine binary tree demultiplexers [87] with the crossbar molecular switch architecture. The idea being to take advantage of the inherent order $2[\log_2(N)]$—where N is the number of nanowires, and $2[\log_2(N)]$ is the number of micro scaled demultiplexing wires needed to address the nanowires—scaling property of the binary tree scheme. Bottom-up processes used to assemble nanowires are typically error prone and susceptible to broken wire defects, lack of conducting properties in some of the nanowires, the fabrication process is also usually characterized by a certain degree of randomness in organization. Defects tolerant schemes which include the use of redundant nanowires [88]—which we have developed and analyzed in this thesis and is presented in chapter IV—and ECC [89] have been proposed as ways of achieving reliable demult-iplexers. Other defect tolerant techniques have been proposed; Kuekes and Williams [90] developed and patented a scheme based on a diode or resistor decoder that requires only $5[\log_2(N)]$ addressing microwires crossing an array of N nanowires. Another architecture based on the field-effect gating of nanowires by the demultiplexers, and which requires only a maximum of $2.2[\log_2(N)] + 11$ address microwires, has been proposed by Dehon et al [91]. This scheme utilizes a fabrication technique which is able to control either the doping profile or material composition along the axial dimension of the nanowires; controlling the doping profile allows for the effective control of the field-effect conduction threshold along the length of the nanowires, thus making some regions gateable and the other oblivious to the normal operating voltage of the intersecting crossbar nanowires. These nanowires have already been experimentally demonstrated

[92-94]. In other works, Beckman et al [86] proposed an electric field effect based demultiplexing scheme with defect tolerance capabilities, with the added advantage of not being seriously restricted in terms of the wire size and pitch of the demultiplexer structure. This scheme uses $2[\log_2(N)] + R$ microwires to address N nanowires, and R—which is zero or a small integer—represents the number of redundant address-lines required for defect tolerance. In contrast to [91], Beckman et al's scheme requires no control over the axial doping profile of the underlying nanowires, instead advantage can be taken of the readily attainable vertical doping profiles, and it is designed to bridge across length scales.

Recently, Stewart et al reported [95] a new direct-write lithography method for the construction of electrical connections to systems of nanoscale devices; in other words the implementation of a demultiplexer. More specifically, a demultiplexer micro-nano connection to a large array of 60 nm pitch nanowires was demonstrated, E-beam radiation was utilized to induce an insulator conductor transition in a thin film organic film separating the micro and nanoscale wires. This method requires only two high yield process steps, in addition to needing microscale overlay alignment between metal layers. It also has an added advantage in that it can be implemented using any standard scanning electron microscope. However, Stewart et al note the fact that high beam currents—in the tens of Pico Amperes—and long programming times in the order of tens of seconds, suggest that their research targeted process is likely to slow for volume integrated circuit manufacturing. Also, achieving demultiplexer programming using this method required extensive material exploration to identify a suitable metal/programmable/insulator/metal

material combination. An exploration which yielded organic monolayers that could not be controllably modified, because they displayed a high degree of sensitivity when exposed to electron beams. The oxides also reacted in an inverse manner, as they proved to be too insensitive.

3.2 Demultiplexer Architecture and Operation

The demultiplexer reliability scheme presented in this thesis is based upon the Kuekes et al proposed scheme [90, 96]. A proof of concept demultiplexer device has already been demonstrated [1], further lending credence to the practicality of this scheme. The fabricated device was built on an 8 × 8 crossbar structure, which was partitioned into 4 sections. One of the 4 × 4 crossbar grid functioned as the main memory, and two of the other 4 x 4 crossbar molecular switches were used to implement the row and column demultiplexers of the main nanomemory. The entire 8 × 8 crossbar nanomemory occupied 1 μm^2 chip area.

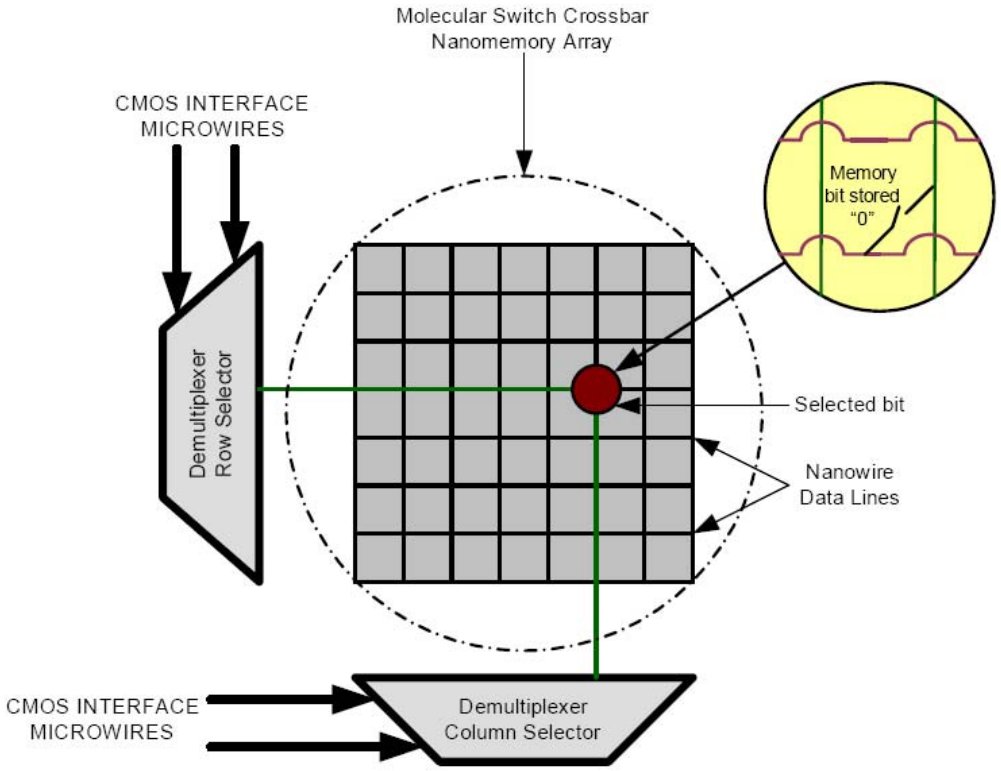

Figure 14. Crossbar nanomemory and demultiplexer organization. The important role played played by the demultiplexer can be inferred by its location at the interface of the crossbar nanomemory and CMOS interface addressing microwires.

The role and location of the demultiplexers in the nanomemory hierarchy gives it a very critical role in the structure of a nanomemory. Faults and defects that originate at the decoder can propagate through the entire memory system leading to paralyzing system failures, Figure 14 illustrates the demultiplexer location. In nanoelectronics, where defects are expected to be prevalent, to reiterate—it is imperative that a high degree of fault and defect tolerance is achieved. Illustrated in Figure 15 is a demultiplexer laid out in a crossbar configuration which controls a 4×4 crossbar nanomemory. Two signals, A0

and A1, which drive four microwire (vertical wires) signal lines, are interfaced with the demultiplexer nanowire address-lines via the placement of bistable molecules at their intersection; the molecular switches are represented by the resistor junctions. The output address-lines can be thought of as having an AND gate functionality. Hence, a nanowire address-line can only be selected if its two input signals are high or "1".

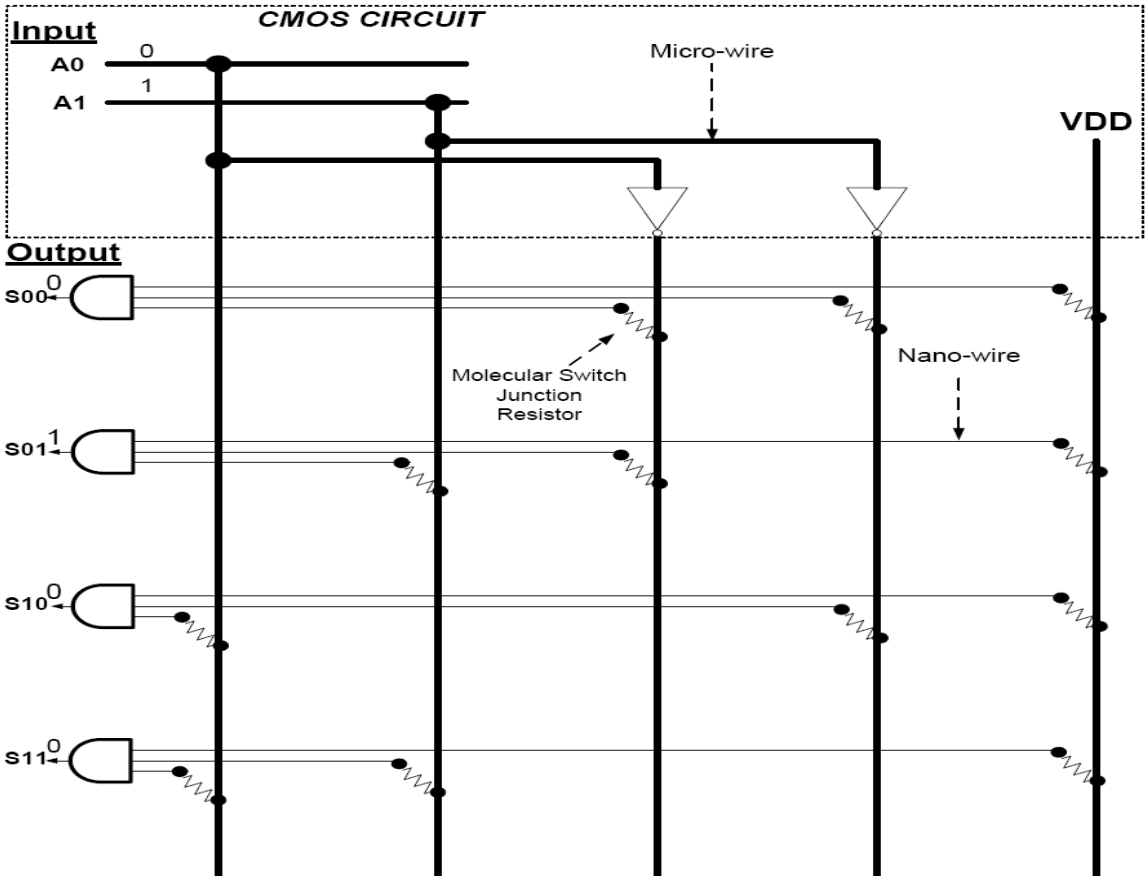

Figure 15. A molecular RAM demultiplexer that requires only m input signals to select 2^m nanowires.

CHAPTER IV

RELIABILITY ARCHITECTURES FOR NANO AND MOLECULAR ELECTRONIC MEMORY DEVICES

4.1 Introduction

Devices fabricated at nanoscale dimensions using top-down or bottom-up processes are intrinsically prone to permanent defects, in addition to being highly susceptible to operational transient faults. The performance driven progress of CMOS technology is made possible by the continuous miniaturization and scaling of microelectronic feature sizes. The dimension to which these devices are shrunk as well as the dimensions at which nano and molecular devices are synthesized leaves them highly susceptible to quantum effects, thermal variations, and cosmic particles; which manifest themselves as transient errors during device operation.

Nanoscale engineering involves the atomic scale manipulation of material to yield a desired device or effect. Fundamental theories of physics maintain that atoms, molecules and electrons can only be controlled or predicted within a statistical degree of certainty. In particular, quantum mechanics, Heisenberg's uncertainty principle and thermodynamics inform us that it is possible to discern the likely behavior of a device but not the absolute behavior of the individual elements of the device [98]. This uncertainty is what makes defect and fault tolerance and reliability in nanotechnology crucial.

4.2 Defects and Faults Classifications

Defects are generally classified as those groups of physical disruptions in a system which results in errors in that systems operation. Defects are usually attributed to the fabrication or manufacturing process. Faults on the other hand, can be described as an incorrect state of a system due to problems which occur during system fabrication, device element failures, unstable design, environmental conditions such as thermal variations, and so on [98]. Faults can be divided into the following sub categories:

1. Permanent Failure: - These occur due to physical ware and tear that result in permanent device failure. Defects which lead to these types of failures are currently being encountered in CMOS technology. For example, dust contamination which occurs at very coarse levels, during the semiconductor fabrication process, could potentially damage large number of transistors in a localized area of the chip, in effect rendering it useless [99]. Other examples include, junction failures caused by the statistical assembly of very few molecules at the switching junctions, which consequentially leads to non-programmable switches or permanent stuck-open junction faults.

2. Sporadic Failure: - These are the failures that intermittently occur in the devices and may lead to permanent device failure. These errors for the most part, are detectable and can ultimately be repaired.

3. Transient Failure: - These faults are attributed to external environmental

> factors, such as radiation, charge leakage, thermal noise, power supply noise and shot noise. The nano scale feature sizes of these devices make them even more susceptible to the fundamental effect of noise during operation. For example, the variation of voltage as well as the charging of the molecular switches increases the probability that random thermal noise will disrupt their operation. Nanoscale devices operating at high clock frequencies and low supply voltages are also very vulnerable to shot noise induced transient faults [100].

We can therefore deduce that defects, arguably, encompassed under faults. But for the most part, defect tolerance is understood to be the ability of a system to function correctly in the presence of fabrication or manufacturing defects. Fault tolerance can thus be defined as the ability to tolerate the permanent, sporadic and transient failures. Fault tolerance is primarily achieved through incorporating redundant or spare elements/devices in the design of the nano system. Redundancy may take the form of physical replication or the use of schemes such as N-modular redundancy or error-control coding, which utilizes redundancy in the code space for the data so that faults can be detected and corrected [98]. There are basically three main fault tolerant techniques being studied in this work for implementation in nanotechnology.

4.3 N-tuple Modular Redundancy Techniques

There are different types of modular redundancy techniques such as *R*-fold multiple redundancies, N-tuple modular redundancy (NMR), and the common triple modular redundancy technique (TRM) which represents the most general technique of NMR. These techniques work by replicating a unit multiple times, running them in parallel and comparing their output using a majority gate. They are used primarily to compensate for transient errors when the computers are in operation [4, 5]. There is however a trade-off with the implementation of modular redundancy; though the redundancy enhanced system as a whole will be more reliable, there will be an N factor trade-off in the processing time due to the depth of the redundant and voting circuitry. Power dissipation and circuit area overhead are additional penalties that result from the implementation of this technique.

The TMR technique involves the three fold replication of a device, all three devices operate the same way and are expected to provide the same output response when they are fed with the same input; this is true if all three devices are defect free. To protect defect or fault tolerance, all three of these devices are linked using a majority voting circuitry. The idea being that the most common output is selected, so that when one single device module is defective, the output from the voting circuitry can still be considered valid or reliable. TMR can thus be used to tolerate multiple faults that occur in one of the three device modules. The reliability of the voters is also important. If the voter is defective, then the function of the TMR is nullified. This problem can be countered by using three voters instead of one.

It should also be noted that the processors with which the majority gates are implemented from are also highly error prone. As a result modular redundancy is best implemented at the logic gate level of abstraction [101].

4.3.1 Von Neumann's Multiplexing Method

Von Neumann showed that by using a group of unreliable gates, all of which have an equivalent rate of failure, to represent a single gate, the probability of performing reliable computing was possible [6]. Von Neumann's proof was derived under the assumption that unreliable devices had statistically independent failures as well as a small probability of failure.

4.3.2 Reconfigurable Computer Technique

Reconfigurable computing is modeled on the Teramac computing concept [7]. This technique is used primarily to compensate for defects that occur to devices during manufacturing. The idea is that defects found after a nanochip has been fabricated are identified, and the chip reconfigured to work around these defects, thus making use of only the functional elements on the chip. Reconfigurable architectures have been found to be more successful in protecting against permanent defects, as in the case of the Teramac, and are less efficient in protecting against transient failures [102].

The question of future memory hierarchy architectures in the face of new nano devices incorporated into hybrid systems, still presents itself as a difficult problem to ponder. Recent studies [102] suggest that the ultimate reliability of nanochips could be as high as 0.9, assuming hundred of redundant components. The SIA roadmap estimate

nanochip densities of 10^{12} devices; it has been speculated by [102] that we can realize as much as between 10^{9} to 10^{10} effective devices after implementing redundancy; however this assumption is based on the utilization of the NAND multiplexing technique.

Studies [41, 103] have been conducted, comparing different fault tolerant techniques for use in nanotechnology. One study [59] compares the *R*-fold multiple redundancy (RMR); Cascaded triple modular redundancy; von Neumann's multiplexing method; and the reconfigurable computer technique. These studies were done using the following assumptions; chip density is 10^{12} devices with a 90% working device probability. The studies showed that RMR and von Neumann multiplexing do not work as well as reconfiguration. It is possible for RMR to be as effective as reconfiguration, but redundancy has to be increased which will correspondently reduce the total number of effective devices; the number of effective devices can be written as $\frac{N}{R}$, where N = # of devices (in this case10^{12}) and R = # of devices used to implement a function. Failure rates for CMOS based devices are less than 10^{-5}; to achieve comparative or better performance, nanochips will have to fail at rates less than 10^{-5}. Studies [5] show that using reconfiguration methods, we would need 10^{4} redundant devices to achieve a 10^{-3} fault rate per device. This translates to a chip with 10^{12} unreliable devices working as if it had 10^{8} perfectly reliable devices. It has also been theoretically demonstrated that by using TRM methods we can achieve failure rates of approximately 10^{-7} to 10^{-8} per device, which could be considered acceptable when compared with the current status quo. The draw back is that reconfiguration takes a long time to implement when dealing with transient errors. It is also implied that a high degree of device reliability will be essential in the

ultradense nanochips of the future.

4.4 Defect and Fault Tolerance in Molecular Electronic Devices

Fabricating reliable molecular electronic devices will depend on the development of efficient and reliable tools, in addition to the synthesis of novel reliable materials. The construction of large scale molecular electronics using self-assembly processes will require that devices and wires are first fabricated, and then assembled into circuits [104]. Ultimately, reliability in molecular electronics hinges on the ability of engineers and scientist to control these bottom-up processes, as well as manipulate atoms and molecules with a reasonable high level of precision. As iterated in the preceding sections, the laws of physics as dictated by the Heisenberg uncertainty principle makes the precise manipulation of atoms a difficult feat to achieve.

Great advances have been made in the research and development of molecular electronic devices. Progress has been made in the developments of molecular scale field effect transistors, single electron devices, non-volatile crossbar molecular switch memory devices, negative differential resistors and diodes. The use of innovative device design architectures and techniques presents the best interim solution for achieving reliability in nanoelectronics devices. Already established fault and defect tolerant schemes can be employed in nanoelectronics design; or at the very least their concepts can. Replication or redundancy, are concepts of choice in achieving defect and fault tolerance in devices.

4.5 Defect Tolerance

Defect tolerance can be implemented at various levels of the device abstraction [98]; the physical, architectural and application levels. This work is based on the application of fault tolerance at the physical level of abstraction. Stuck open faults are not the sole cause of unreliability; it, along with other known forms of defects that occur in crossbar nanomemories can be explained as follows [96];

1. Stuck-open defect: - In this case molecular switch junctions function exclusively as open circuits, maintaining no connection between intersecting nanowires regardless of intended functionality.
2. Stuck-closed defect: - Occurs when a molecular switch junction registers a connection between intersecting nanowires when one is not desired.
3. Broken wire shorts: - Occurs when a nanowire breaks in such a manner that it makes contact with an adjacent nanowire. This introduces an unwanted conduction path between both Nanowires.
4. Broken wire defect: - Signal propagation is permanently interrupted along that nanowire path due a structural break.

A major challenge facing reliability is finding that optimal point in which a balance between reliability and redundancy is achieved. Arbitrarily adding spare elements, which are also defect prone, to a device could ultimately prove to be detrimental to the system performance. Hence, we have researched and presented results on the impact of defects on the performance of nanoelectronics memories that utilize grid based architecture in the ensuing chapters.

CHAPTER V

MULTI-SWITCH JUNCTION RELIABILITY ARCHITECTURES

5.1 Introduction: Multi-Switch Junction Crossbar Architecture

Crossbar architectures are desirable because they are geometrically advantageous to the implementation of defect tolerant schemes; the grid like structure of the scheme allows for the straightforward implementation of redundancy. Additional benefits include; efficient signal propagation and optimized access time delay. The crossbar architecture signal propagation efficiency is achieved by keeping the aspect ratio (column length divided by the row length) of the memory at unity [36]. Therefore limiting the length of the column and row interconnects to the minimum required length necessary to reduce propagation delay. Furthermore, the crossbar is a good approach because signal delay increases, at the very least, linearly with interconnect length. In implementing the MSJ scheme, we take advantage of the fact that, only a single molecular switch is required to store a bit in a nanomemory. This stands in stack contrast to the six and four transistor requirement of the SRAM and DRAM, respectively. This geometric advantage is used as the foundation of our scheme.

We define redundancy (k) as the numbers of nanowires attached to the same row (column) electrode, which is used to represent a single wordline (bitline) in the crossbar molecular switch memory. The objective is to evaluate the effects of having k number of nanowires working in parallel and connected to the same electrode, as illustrated in

Figure 16. The primary cost of such a scheme becomes the multiplicative increase in total crossbar molecular switch memory area. The condition under which these switch junction sets are able to tolerate faults is equivalent to the ability for a mesh network graph of connected nodes to maintain connection between specified input and output node set, when a break in their natural connection path occurs. Mesh network with these characteristics have already been studied [105, 106], using similar approach we first define a network $k \times k$ of switch junction nodes, having equal number of rows and columns.

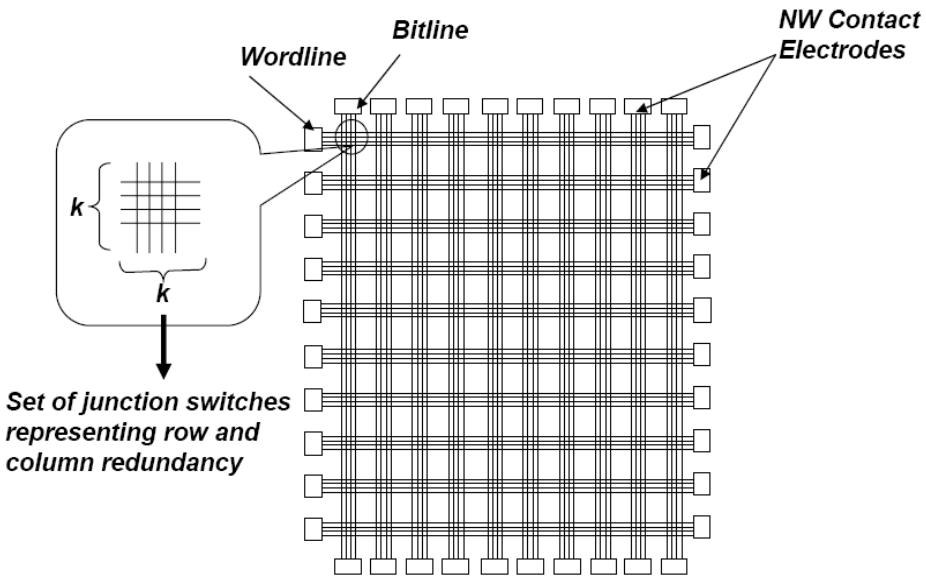

Figure 16. A crossbar molecular switch array, with a redundancy of $k = 4$.

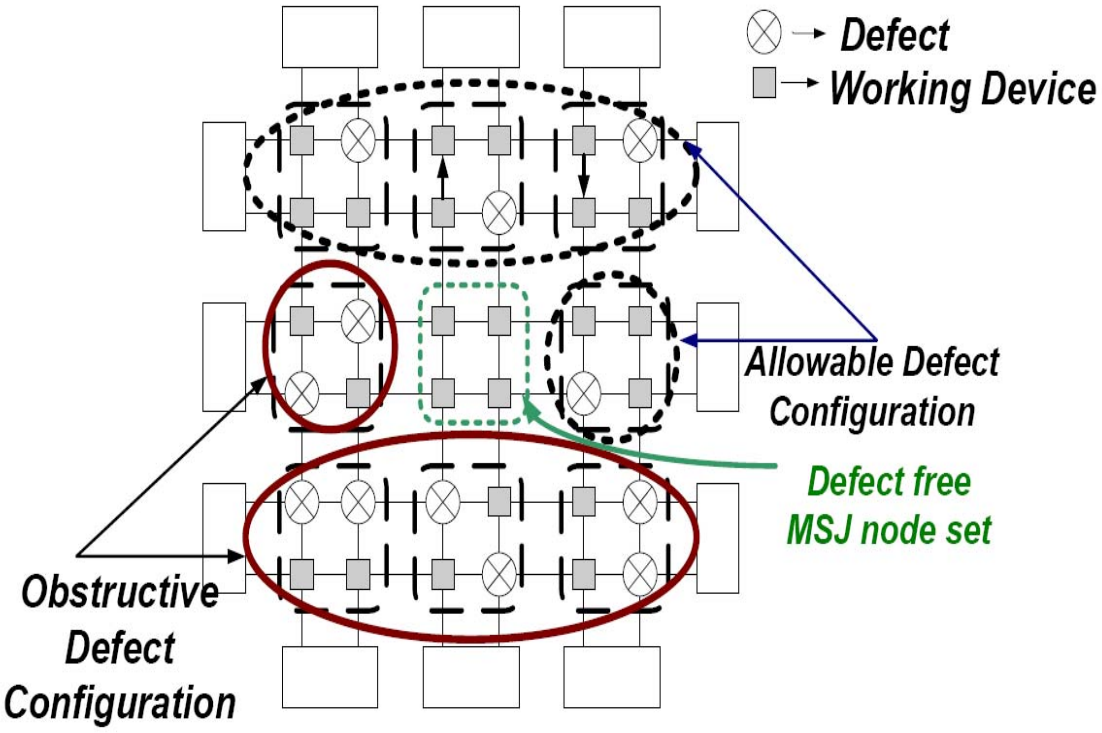

Figure 17. Illustration of some allowed defect configuration in a 3×3 memory crossbar array, having a row and column redundancy of 2.

The boundary nodes of the $k \times k$ subset network are considered to be the first nodes of each row and column in the network. In other words, for 1, 2, 3…, k rows and 1, 2, 3…, k columns, rows 1 and k, are the row input and output nodes respectively, likewise, columns 1 and k, are the column input and output nodes of the network. The defect tolerance in this network is defined as its ability to maintain a connection, between at least one of its row (column) input nodes and one of its row (column) output nodes. This suggests that only specific numbers and patterns of defective switch junction can be tolerated. Figure 17 illustrates this concept on a 3 × 3 crossbar molecular switch memory array with a redundancy of $k = 2$; the allowable and obstructive defect patterns are

indicated for better clarification.

5.1.1 Multi-Switch Junction Circuit Model

In this section, the parameterized circuit model analysis of a fault tolerant crossbar molecular memory [107] is expanded upon. Here, the performance benefits of having multiple nanowires represent an electrode word-line (bit-line) are analyzed. It has been shown that by taking advantage of the quantized nature of the separation between nanowires, which are assembled via the Langmuir-Blodgett method, the ratio of the number of nanowires per electrode can be defined [9]. This process is accomplished by setting the separation between nanowires to a value similar to the width of the contact electrode. In the MSJ scheme, the nanowires on each electrode will be closely packed together, with a pitch distance between the nanowires being on the order of the nanowire width.

5.1.2 Scaled Multi-Switch Junction Crossbar Circuit Model Approach

For a more accurate analysis, a scale model simulation was also designed and simulated; the scale model provided a reasonably more accurate picture of performance. The difficulty encountered in this approach resulted from the prevalence of leakage current in addition to the absence of gain in the circuit. The reason for this being, when you scale a circuit all the parasitics that appear minuscule in a parameterized circuit become amplified.

At the CMOS layer, micro scale devices fabricated with lithography, functions as a bridge for accessing and retrieving information from the nanomemory devices.

Conventional lithographically defined integrated circuits act as a substrate for the nanoelectronic devices, much like printed-circuit boards are used as a packaging hierarchy for integrated circuits [108]. The hybrid integration between the micro and nano layers are discussed in more depth in the decoder/demultiplexers chapters. Ultimately, researchers will have to achieve the capability of accurately modeling the performance of nanoelectronic elements, devices and systems. To accomplish this feat, scaled models will have to be properly understood and developed.

5.2 Nanomemory Decoders

5.2.1 Defect-Tolerant Crossbar Demultiplexers

As stated in previous sections, the demultiplexer is an integral part of the crossbar nanomemory architecture. Not only because it acts as an interface between CMOS and nano circuitry, but because failures at the demultiplexer essentially cripples the entire functionality of the crossbar nanomemory. As a result, in this work the MSJ scheme which improves the reliability of the demultiplexer, in combination with ECC has been simulated and shown to elevate the reliability of the demultiplexer and crossbar nanomemory as a whole.

We have already published work [107] on demultiplexers that implement both the MSJ and ECC fault tolerant schemes, jointly and separately. A one-to-one ECC parity scheme was implemented in this work. Our analysis also included a transmission line model implemented as bundles of carbon nanotubes.

5.2.2 Effects of Demultiplexer Defects

Defects and faults in crossbar nanomemories and demultiplexers may occur in a variety of forms. They may occur in the form of transient errors due to cosmic rays or tunneling electrons; fabrication defects such as nanowire breaks and insufficient molecules in the molecular switch junctions. These types of defects cause "stuck open" defects as previously explained, so that a defective address-line will always register logic "0" at its output, regardless of the input. When there is a stuck open fault, it becomes impossible to select a single address. For example, in the demultiplexer schematic illustrated in Figure 18, it can be observed that when there is a break o suck at zero defect in the ***S01*** address-line, an attempt to select address ***S00*** will result in the selection of both addresses ***S00*** and ***S01***. This is an undesirable effect that must be tolerated in order to achieve an effective demultiplexer decoder.

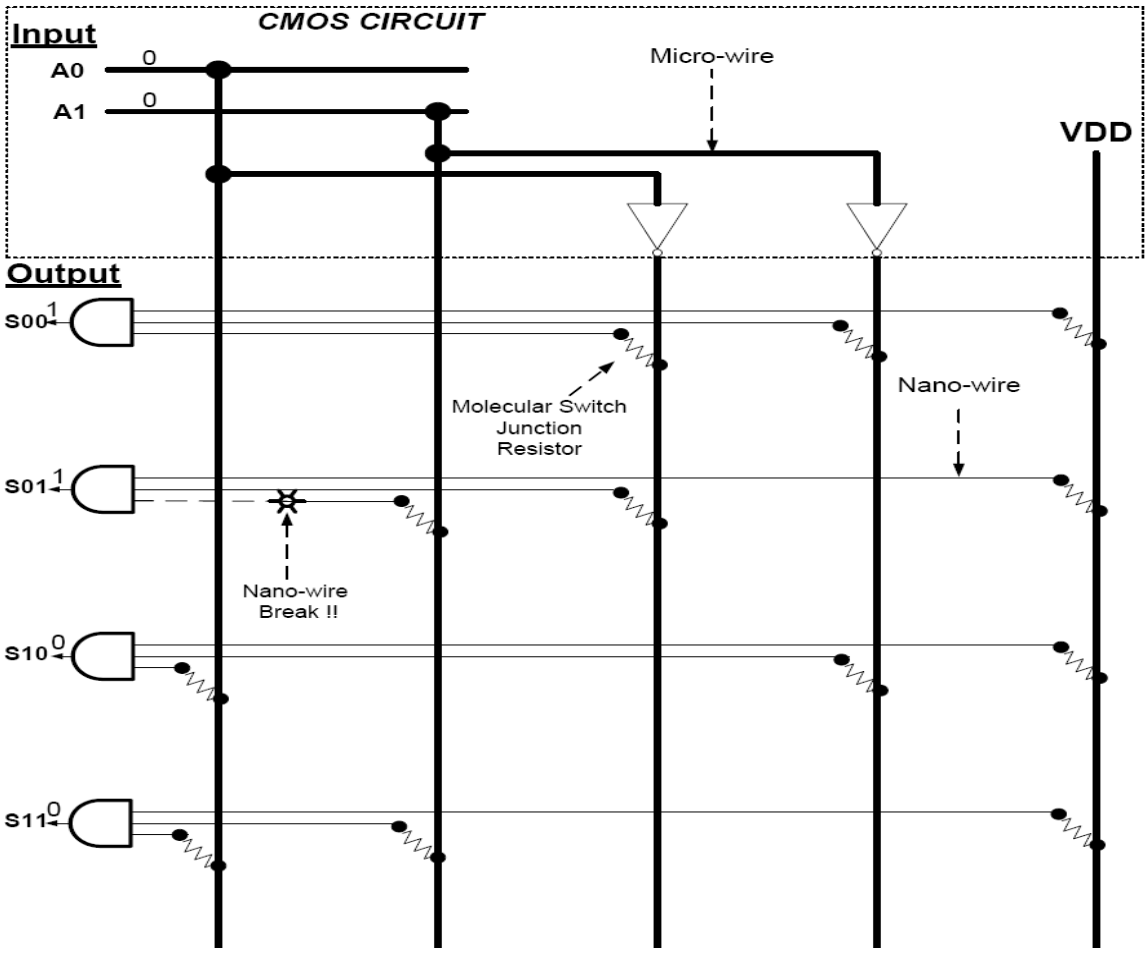

Figure 18. Resistor logic demultiplexer. When there is a break in the ***S01*** address-line, an attempt to select address ***S00*** will result in the selection of both addresses ***S00*** and ***S01***. This is an undesirable effect that must be tolerated in order to achieve an effective demultiplexer decoder.

5.2.3 Multi- Switch Junction Resistor Logic Demultiplexers

This scheme utilizes the relatively simpler strategy of introducing redundancy directly into the nanomemory fabric [107]. This approach requires no additional logic circuitry and can be easily implemented. As in earlier demultiplexer implementations, single or bundles of nanowires can be used, and as such, redundancy can be implemented

entirely in the horizontal nanowire dimension, or in both the horizontal and vertical dimensions thus creating nanowire bundle address-lines.

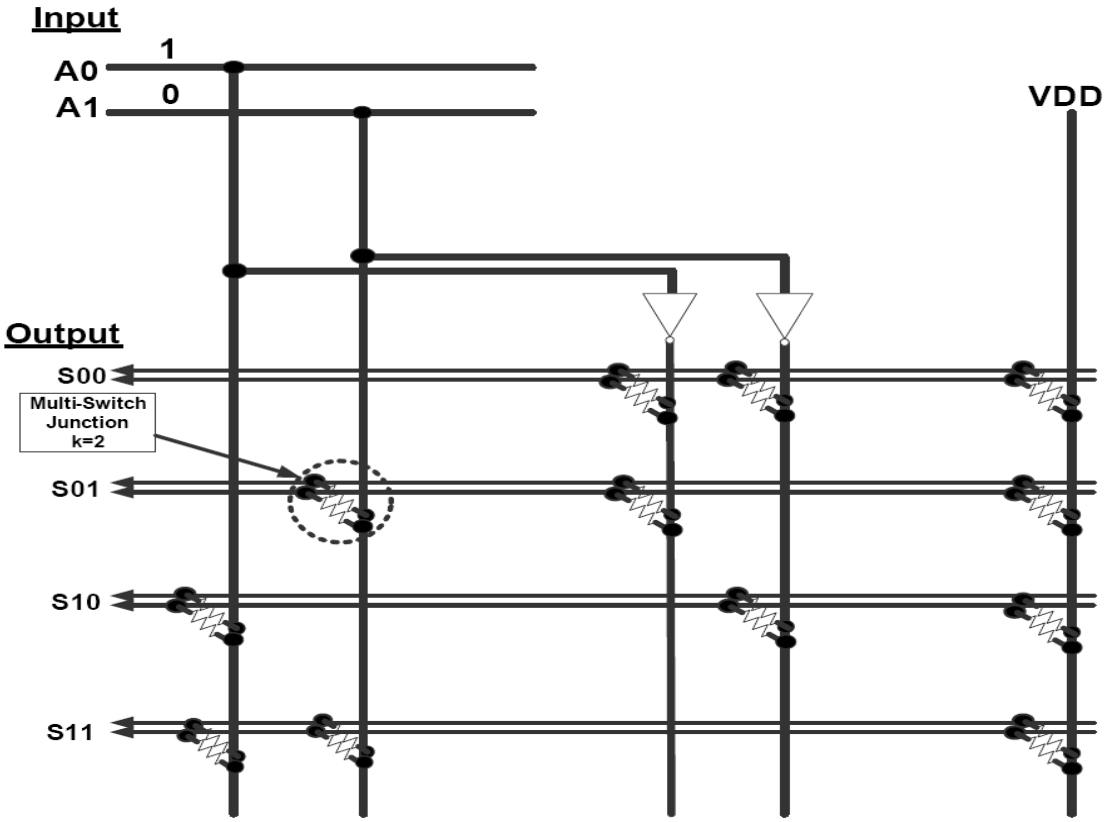

Figure 19. Demultiplexer circuit layout, illustrating the implementation the MSJ scheme using a redundancy of 2.

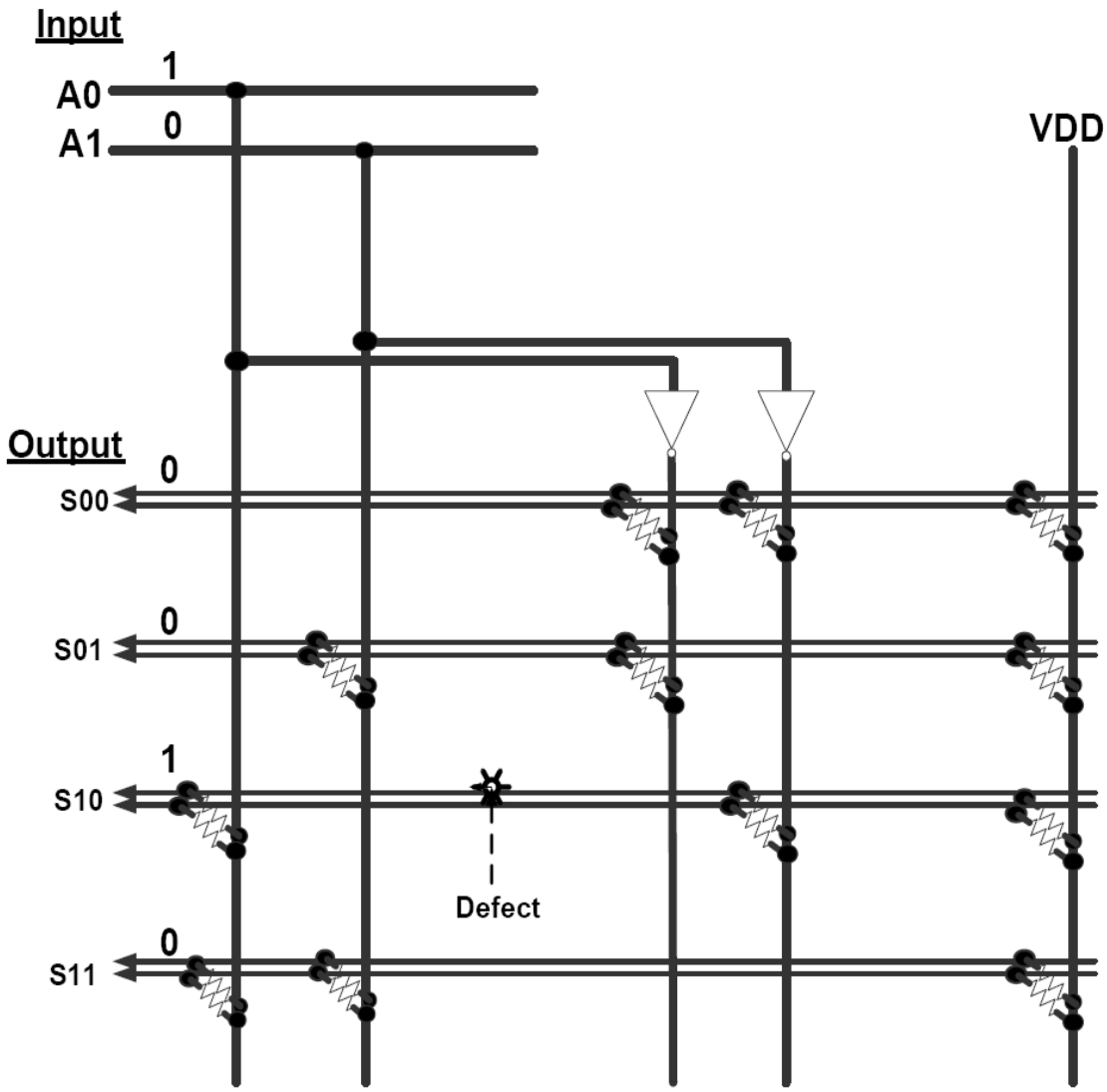

Figure 20. When a MSJ redundancy of 2 is implemented in the demultiplexer, A defect can be tolerated. In the illustration the S10 line contains a defect, but when the S10 address is input, the correct address-line is still selected.

5.2.4 ECC Resistor Logic Demultiplexers

The error control strategy used in this work is similar to that introduced by Kuekes et al [109]. Given the single direction signal flow from the input to output electrodes of the demultiplexer; the forward error correction strategy for one-way systems is used [110]. The address field matrix of the demultiplexer is represented as a linear block code. For any input signal length $m \geq 3$, there exists a Hamming code with parameters as follows [110]; Code length: $n = 2^m - 1$, input signal binary code length: $k = 2^m - m - 1$, and number of added parity bits: $m = n - k$.

Each address-line in the demultiplexer is comprised of a number of resistors in parallel, and as illustrated in Figure 18. These resistors coupled with their nanowire connections, provide the requisite AND gate functionality for each address-line. As in Figures 15, 18, 19 and 20, the input signals are represented as a binary input vector A[0]……A[n-1]. For two address-line vectors **a** = (a_0, a_1, a_2…….a_{n-1}) and b = (b_0, b_1, b_2…….b_{n-1}), the number of positions where a_i and b_i.

The important parameters of the code can be expressed by the syntax [***n, k, d***]. The linear code having length ***n*** and dimension ***k*** is called an (***n, k***) code. The minimum d_{dist} of the linear coding block will be denoted by d_{min}. The minimum distance parameter is essential in describing the error-correcting capabilities of the code. Codewords are generated from the input signals array vector by linear mapping [111]. The input signal vector $\boldsymbol{A = (A_0 A_1 A_2 \ldots A_{n-1})}$ is linearly mapped to a generator matrix ***G,*** which is a $k \times n$ matrix, yielding the codeword $\boldsymbol{Y = (Y_0 Y_1 Y_2 \ldots Y_{n-1})}$. Years of research in coding theory has yielded a plethora of efficient codes. Knowing the generator matrices of these codes

allows for the utilization of these codes in circuit theory.

When a break occurs without the implementation of ECC as described in Figure 18, a conflict is expected to occur between the ***S00*** and ***S10*** address-lines. However, because ECC is implemented, the circuit is pulled down by the ECC "0" bit, thus tolerating the fault; as is demonstrated in Figure 21.

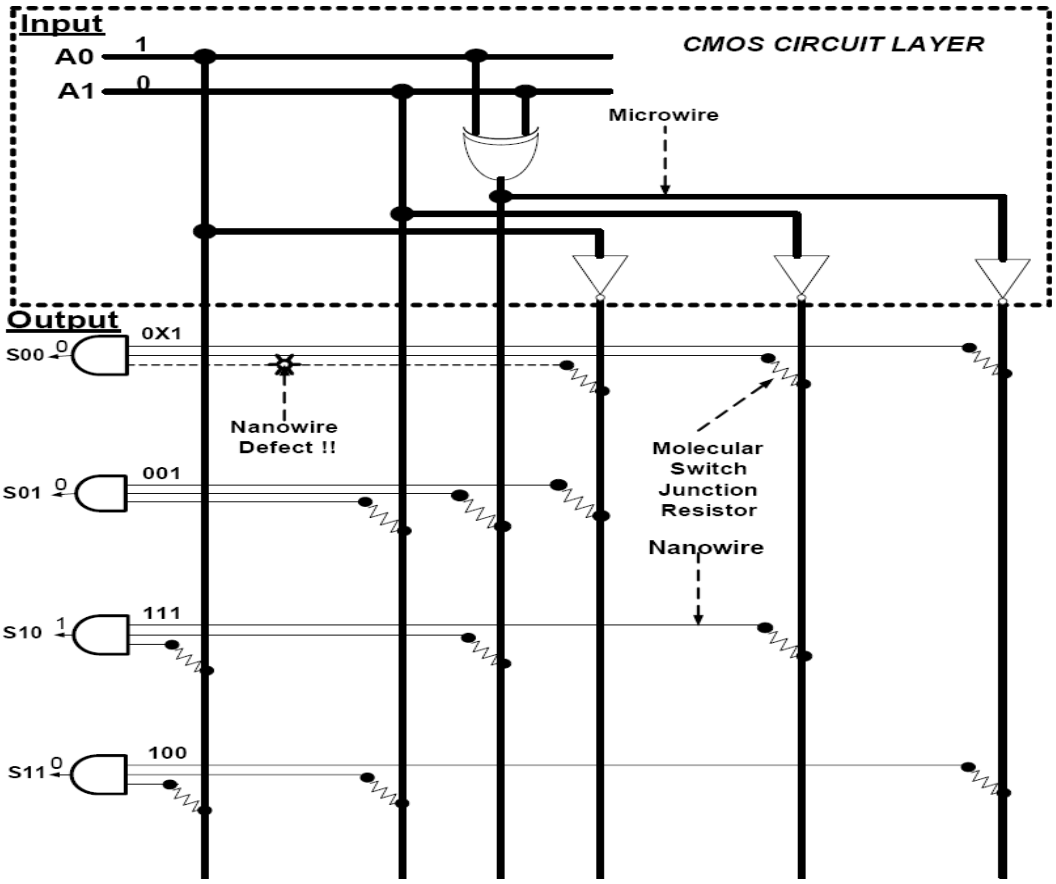

Figure 21. A molecular RAM demultiplexer with an additional EX-OR gate and inverter for ECC implementation. When a stuck open fault occurs in address line S11, a there is no conflict between addresses S11 and S10 because ECC helps to tolorate the fault.

5.2.5 Enhanced Multi-Switch Junction ECC Resistor Logic Demultiplexers

Error correction coding in combination with the MSJ scheme can be, and has been demonstrated in this work, to enhance the reliability of the demultiplexer. Error correction codes are limited in the number of defects they can tolerate—they are only able to tolerate $d_{\min} - 1$ defect. This is also true of the MSJ scheme which requires at list one of its redundant nanowire address-lines to be fully functional to achieve defect tolerance. Figure 22 illustrates.

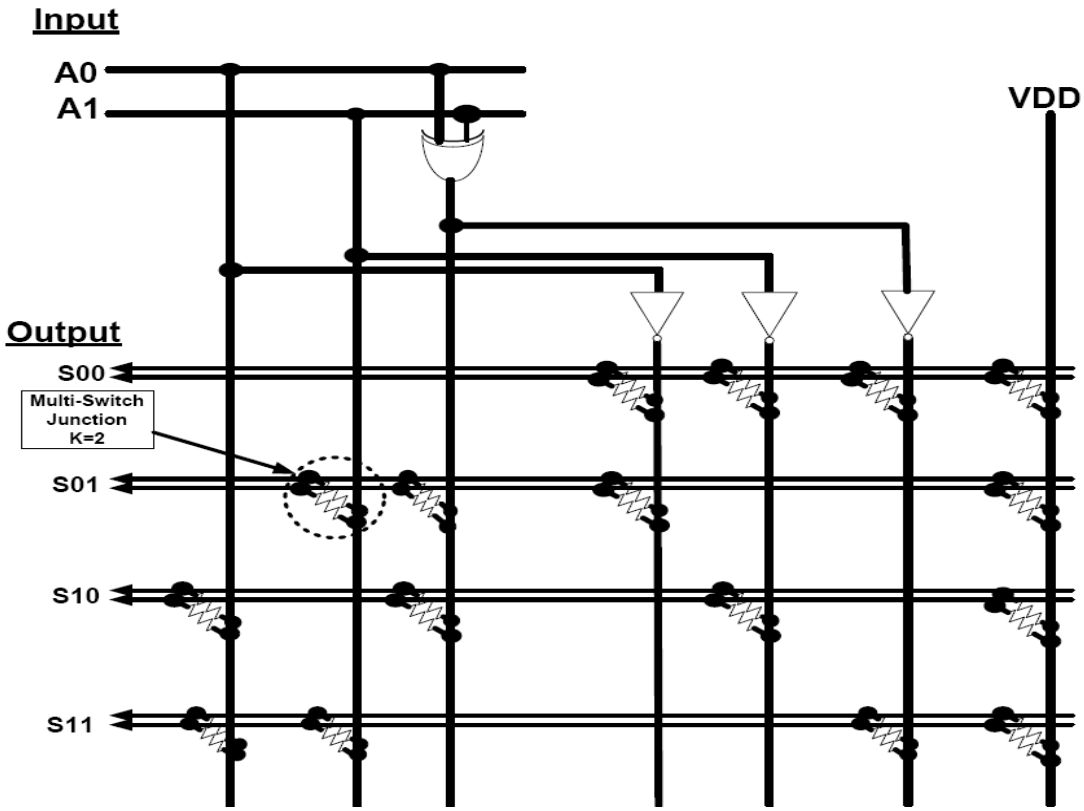

Figure 22. A molecular RAM demultiplexer with an additional EX-OR gate and inverter for ECC implementation. An additional nanowire is added to each row to implement the multi-switch junction scheme.

5.3 ECC Architecture Implementation

5.3.1 Crossbar Main Memory

Fabrication methodologies along with the configuration of the crossbar main memory, makes it difficult to implement ECC directly into the fabric of the main memory. To circumvent this limitation, R-modular redundancy logic may be implemented at the peripherals of the crossbar main memory block. It is implemented in CMOS and is located between the demultiplexer and the main crossbar nanomemory. Two problems arise from this implementation. (1) Area Cost (2) Performance (Delay and Power) Cost. The gains in reliability have been demonstrated in our simulation results to offset these costs.

As illustrated in Figure 23, a CMOS block implementing XOR gates for error detection, could be utilized. Performance could be optimized by assuming that the bit is correct, unless otherwise indicated by the error detection block, at which point the bits can be rolled back and the error correction mechanism initiated. The Nanomemory blocks could be increased to a triple modular redundant configuration, with all three block working in parallel. Their outputs can then be piped through CMOS level voting circuitry which decides the output based on the majority output. This would be similar to the triple modular redundancy technique discussed in [112]. Implementing the majority gates at the CMOS level will allow for increased reliability, as they are less error prone and less susceptible to noise than nanoscale devices.

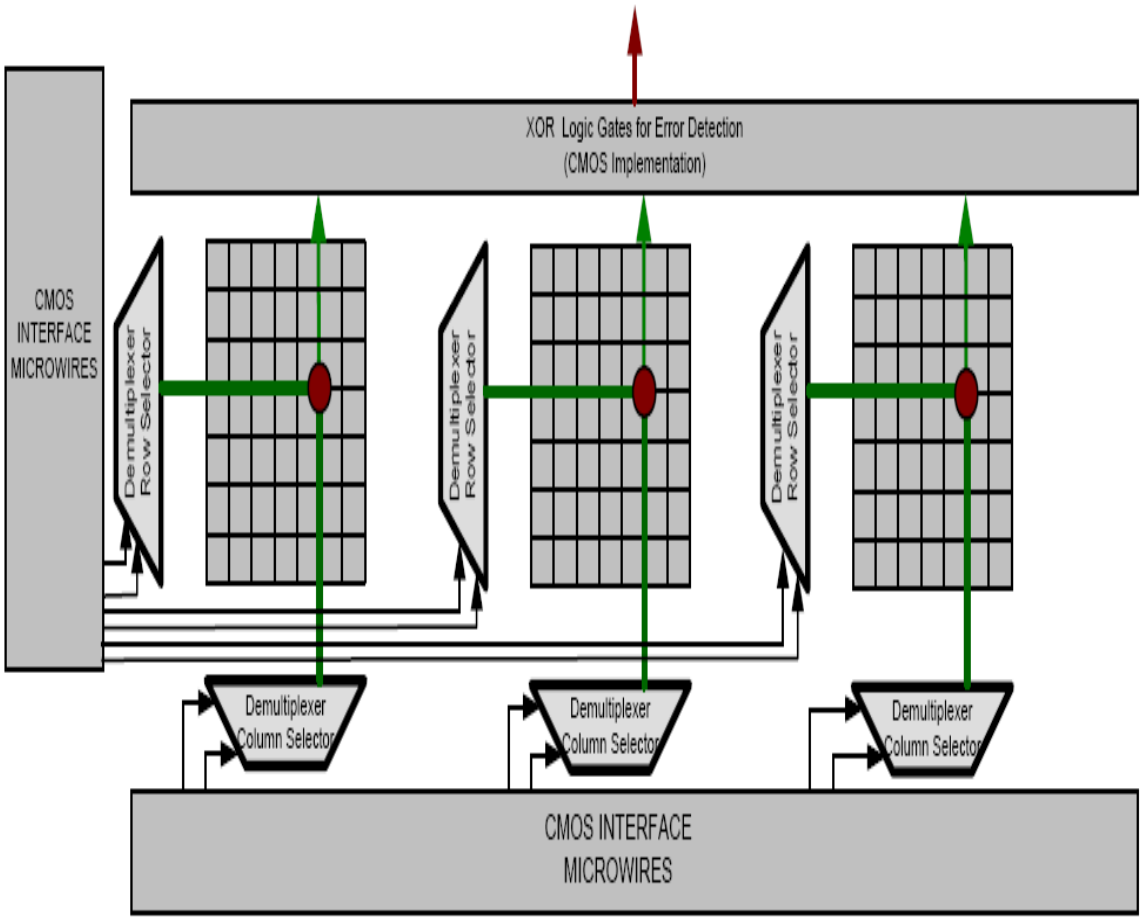

Figure 23. Error detection configuration of a triple modular redundant nanomemory block.

5.3.2 Decoder Probability Analysis Methodology

The process of adding spare nanowires as suggested in the MSJ scheme can be equated to ***N***-choose-***i*** sparing. Hence, the probability of yielding a desired number of non-defective molecular switch junctions from a cumulative yield of molecular switches available for a desired nanowire address-line can be determined using binomial distribution. Given an address consisting of ***i*** bits with (***k*** – 1) redundant bits and ***x*** ECC bits, the Binomial (***i, p***) distribution is the probability of yielding ***i*** useable bits from ***N*** total bits, given a specified defect probability ***p***. The probability of yielding ***i*** non-defective molecular switch junction from ***N*** total junctions is given by the following binomial distribution:

$$P_{yield}(N, i) = \left(\binom{N}{i} P^{i} (1 - P)^{N - i} \right) \tag{5-1}$$

Where

$$N = i + (k - 1) + x \tag{5-2}$$

This is another mode of computing the fabrication related defect probability. To use this approach, it must be assumed that defects are uniformly spread, independent and have a uniform probability of occurring.

5.3.3 Banking Architecture Scheme

Banking is a scheme that has also been suggested reliability in nanomemories [108, 113, 114]. It utilizes a partition of a nanomemory into smaller memory blocks in order to boost access time, power conservation and fault-tolerance. This is akin to schemes implemented in CMOS nonvolatile memories such as DRAM and SRAM. It involves the use of redundant memory modules to tolerate faults that occur in the main Nanomemory module. Choi et al proposed a variation of this scheme which was comprised of two main parts; the main molecular memory modules along with spare modules, and the CMOS based circuitry for providing power, address translation and the requisite logic for the molecular memory modules [114].

Given that n is the number of bits and m is the number of molecular memory modules; a $2n \times 2n$ is addressed using $2n$ bits. The number of molecular memory modules and spares are given by $2^{2(n-m)}$ + s, where is the number of spare modules required to achieve defect tolerance. The modules are selected by using a module table and decoder implemented in the CMOS layer. A module in deemed defective if the module cannot form the desired size memory array.

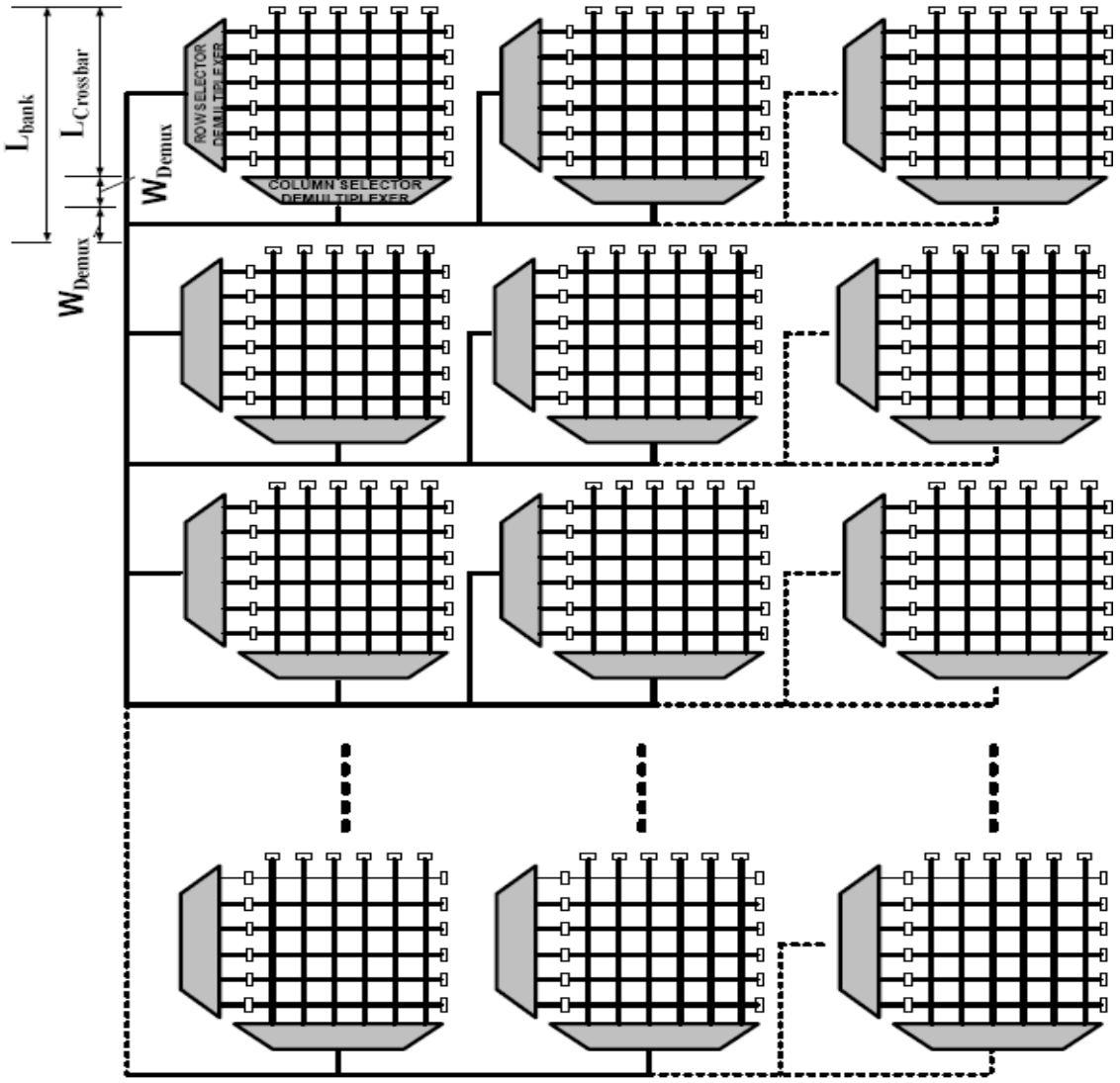

Figure 24. Area configuration of a crossbar nanomemory banking scheme used to implement fault tolerance.

5.3.4 Area Analysis Methodology

The area of the MSJ scheme can be modeled as the area of a single memory bank with redundancies at the nanomemory crossbar level, as well as the demultiplexer levels of abstraction. Demultiplexers serve as decoders for addressing and accessing stored information in the memory. Nanomemory banking implies using several small nanomemory modules to construct a large nanomemory as shown in Figure 24. Fault-tolerance at this level implies using redundant nanomemory modules as backup memories. The parameters used to evaluate the nanomemory bank area are N_{micro_row}, N_{nano_row}, W_{micro}, W_{NW}, W_{Demux}, L_{bank}, λ_{micro}, λ_{nano}, A_{bank} and A_{MSJ}. Where, N_{nano_row}, is the number of row nanowires in the crossbar nanomemory, for ease of analysis it is assumed that the number of row and column nanowires and micro scaled wires are equal. N_{micro_row}, is the number of micro scaled wires used to address the crossbar nanowires and is given by equations (5-3) and (5-4) for the banking and MSJ schemes.

$$N_{micro\_cols\_bank} = 2\,Log_2(N_{nano\_cols\_bank}) \tag{5-3}$$

And

$$N_{micro\_cols\_MSJ} = 2\,Log_2(N_{nano\_cols\_MSJ}) \tag{5-4}$$

L_{bank} is the length of the nanomemory bank defined as

$$L_{bank} = [2W_{Demux\_bank}] + [(W_{nano\_bank} + \lambda_{nano\_bank}) \times N_{nano\_rows\_bank}] \tag{5-5}$$

Where,

$$W_{Demux\_bank} = (W_{micro\_bank} + \lambda_{micro\_bank}) \times N_{micro\_cols\_bank} \quad (5\text{-}6)$$

W_{micro} is the width of the microwires in the demultiplexer. λ_{micro} is the pitch of the lithographically defined micro scaled address wires and λ_{nano} is the pitch of the nanowires. In the L_{bank} formula of equation (5-5), the parameter W_{Demux} (demultiplexer width) given by equation (5-6), is multiplied by a factor of two because the banks are conservatively estimated to be separated by two demultiplexer widths as illustrated in Figure 24. The parameters for the MSJ crossbar nanomemories are equivalent to those used for the bank nanomemory modules. The MSJ nanomemory is essentially one single nanomemory bank implementing a single nanomemory array with internal nanowire redundancy for fault tolerance. This point is reflected in the calculations by including the redundancy factor *k*, as is shown in equations (5-7) and (5-8).

$$L_{MSJ} = [W_{Demux\_MSJ}] + [(W_{nano\_MSJ} + \lambda_{nano\_MSJ}) \times N_{nano\_rows\_MSJ}] \quad (5\text{-}7)$$

Where,

$$W_{Demux\_MSJ} = (W_{micro\_MSJ} + \lambda_{micro\_MSJ}) \times N_{micro\_cols\_MSJ} \quad (5\text{-}8)$$

where L_{MSJ} is the length of the MSJ nanomemory bank. The area of a single nanomemory

bank and MSJ nanomemory respectively, are:

$$A_{bank} = (L_{bank})^2 \quad (5\text{-}9)$$

And

$$A_{MSJ} = (L_{MSJ})^2 \quad (5\text{-}10)$$

The following parameters were used to compare the areas of both schemes: a microwire pitch, λ_{micron} = 130 nm, based on the ITRS [48] specifications for DRAMs in 2007 and nanowire pitch, λ_{nano} = 40 nm, based on specifications in [59]. The analysis assumes that every memory bank has a dedicated backup module. The graphical analysis presented in Figure 25 quantifies these calculations and shows the advantage held by the MSJ over the banking scheme, in area utility or area overhead.

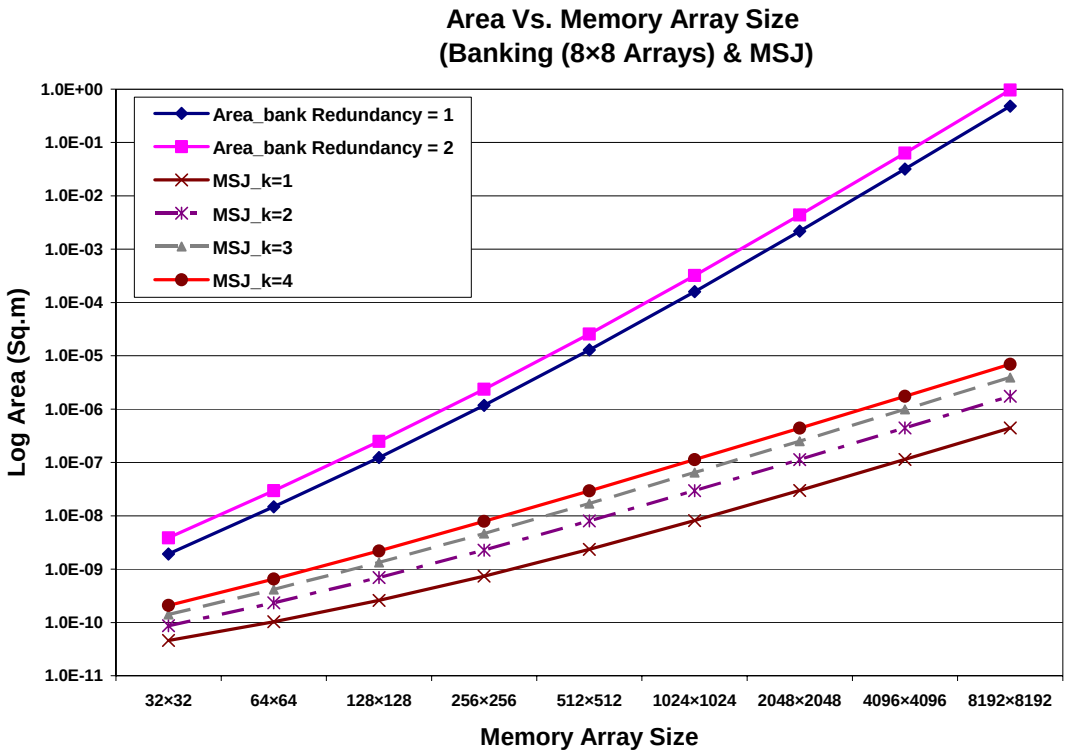

Figure 25. Comparison of area utilization of the MSJ scheme and the banking scheme. 8×8 array sized banks were utilized in this analysis.

Comparisons of the overhead incurred by using the MSJ scheme versus a banking scheme indicate an area utility advantage for the MSJ scheme over the banking scheme for memory array sizes in the range of 32×32 to 8192×8192. This advantage can be attributed to the absence of complex peripheral circuitry in the implementation of the MSJ architecture. When implementing the banking scheme, peripheral control circuitry that address the redundant banks such that when a fault occurs, the appropriate replicated bank is located, accessed, and the desired data retrieved. Furthermore, the banking scheme peripheral circuitry will more than likely be implemented using CMOS technology, thereby increasing area, delay and power penalties. From the plot of Figure

25, the observation can be made that the MSJ scheme has the advantage over the banking scheme for smaller nanomemory arrays when peripheral circuitry are not considered. The case is the reverse as the nanomemory array sizes are increased. It has been shown [114] that each nanomemory bank will require a backup bank in order to achieve a good degree of reliability. Hence, a comparison is made between a nanomemory array implemented with twice the required number of nanomemory modules, with an MSJ nanomemory implemented with k = 2, 3 & 4. The reliability analysis of the proceeding sections show the increase in reliability results achieved through the implementation of the MSJ scheme.

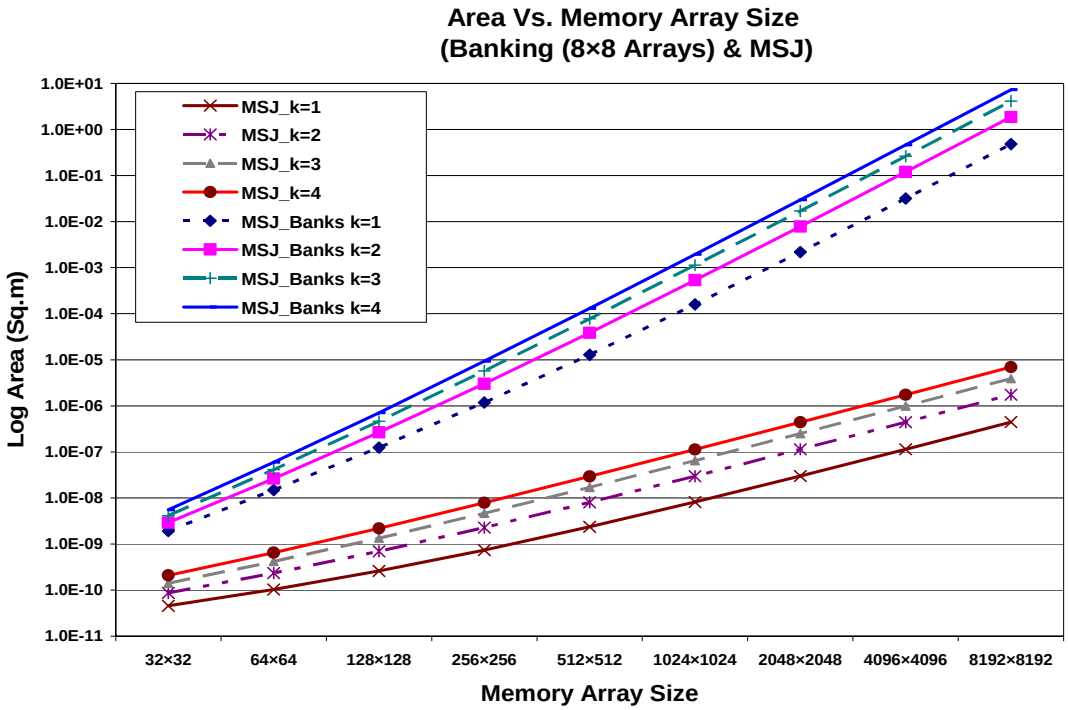

Figure 26. Comparison of area utilization of the MSJ scheme and an integrated MSJ-banking scheme. 8×8 array sized banks were utilized in this analysis.

The area analysis could also be extended to investigate the area overhead of enhancing the reliability of the nanomemory arrays in the banking scheme by implementing the MSJ scheme in them. The result of this analysis can be observed in the plot of Figure 26. The MSJ-only nanomemory configuration still offers better area utility than the integrated MSJ-banking scheme.

The MSJ area utility can be compared against the Banking scheme in the following way; when the probability of molecular junction failure is 1%, and no redundancy is implemented in the crossbar nanomemory, then the PRISM computed probability of reading or writing the correct bit in an (8×8) nanomemory array is 53%. By implementing an MSJ redundancy of $k = 3$, that probability of working is increased to a 100%. Under ideal conditions, when the probability of molecular junction failure is 0%, implementing the banking scheme for a (16×16) nanomemory array using (8×8) nanomemory banks, will require 4 such (8×8) banks to construct the (16×16) nanomemory array, requiring an area overhead of 0.256 nm^2. Constructing an (128×128) nanomemory array will require 256 (8×8) banks with an area overhead of approximately 123.9 nm^2.

As previously mentioned, when the probability of probability of molecular junction failure is 1%, it was determined, and has been elaborated on, in chapter VII (section 7.6), that 18 (8×8) nanomemory banks will be needed to construct an (16×16) nanomemory array, with an area overhead of approximately 1.15 nm^2. In the case of the (128×128) nanomemory array, 530 (8×8) nanomemory banks are required, and the area overhead was calculated to be approximately 256 nm^2. These area overhead are significant when you consider that just by implementing an MSJ redundancy of $k = 3$, without banking

(also provided in Figure 25), you achieve a 100% nanomemory reliability at comparatively lower cost to area overhead as indicated below:

- Implementing MSJ of $k = 3$.
  - 16×16 → Area overhead ≈ 0.052 nm^2
  - 128×128 → Area overhead ≈ 1.34 nm^2

By combining the MSJ and Banking schemes (Figure 26 plots this scheme), the (8×8) nanomemory banks can be enhanced to achieve a 100% reliability, thus requiring less banks for larger nanomemory array implementation. The area overhead is provided below:

- Implementing banks enhanced with MSJ of $k = 3$.
  - 16×16 = 4 (8×8) banks → (Area overhead ≈ 0.446 nm^2)
  - 128×128 = 256 (8×8) banks → (Area overhead ≈ 461.97 nm^2)

The MSJ-Banking approach only faired better in Area overhead cost for the (16×16) nanomemory array configuration but not the (128×128) array.

CHAPTER VI

CROSSBAR NANOMEMORY AND DEMULTIPLEXER DELAY AND POWER ANALYSIS RESULTS

6.1 Crossbar Nanomemory: Parameterized Circuit Model Analysis

The approach taken was to first develop a comprehensive understanding of the crossbar nanomemory circuit model requirements and operation. To this end, a parameterized model similar to previously studied circuit models [115], was modified and re-designed to perform transient and power analysis of the crossbar nanomemory. The objective here was to get a first-order analysis understanding of the performance of the crossbar nanomemory operations. Using a parameterized model analysis approach was of great value in understanding and developing a good circuit model of the crossbar nanomemory. It also served as a validation platform for the design and implementation of the scaled crossbar model presented in the proceeding sections.

The transient parameterized circuit model is illustrated in Figure 28. In the crossbar nanomemory parameterized circuit model, it was essential to incorporate a parasitic capacitance network witch mirrored the capacitive interactions of the nanowires in the crossbar nanomemory architecture. In this model, the nanowires on equivalent electrodes are assumed to be densely spaced, and driven by the same voltage source; in this way the nanowires switch in tandem, thus making the effective parasitic capacitance between them zero [116]; Figure 27 provides an illustration of the capacitor network. The

parasitic capacitance between separate nanowire sets can thus be modeled in a manner similar to [117] and as expressed in equation (6-1).

$$C_{eff} = C_0 + C_{ES} + C_Q \tag{6-1}$$

where C_0 is the capacitance between the driving electrode, the nanowire, the receiving electrode and the substrate. C_{ES} is the electrostatic capacitance and is given by

$$C_{ES} = \frac{2\pi L \varepsilon\varepsilon_0}{\ln(2h/r)} \tag{6-2}$$

C_Q is the quantum capacitance and is given by

$$C_Q = \frac{2e^2}{hv_f} \tag{6-3}$$

where v_f is the Fermi velocity of the nanowire.

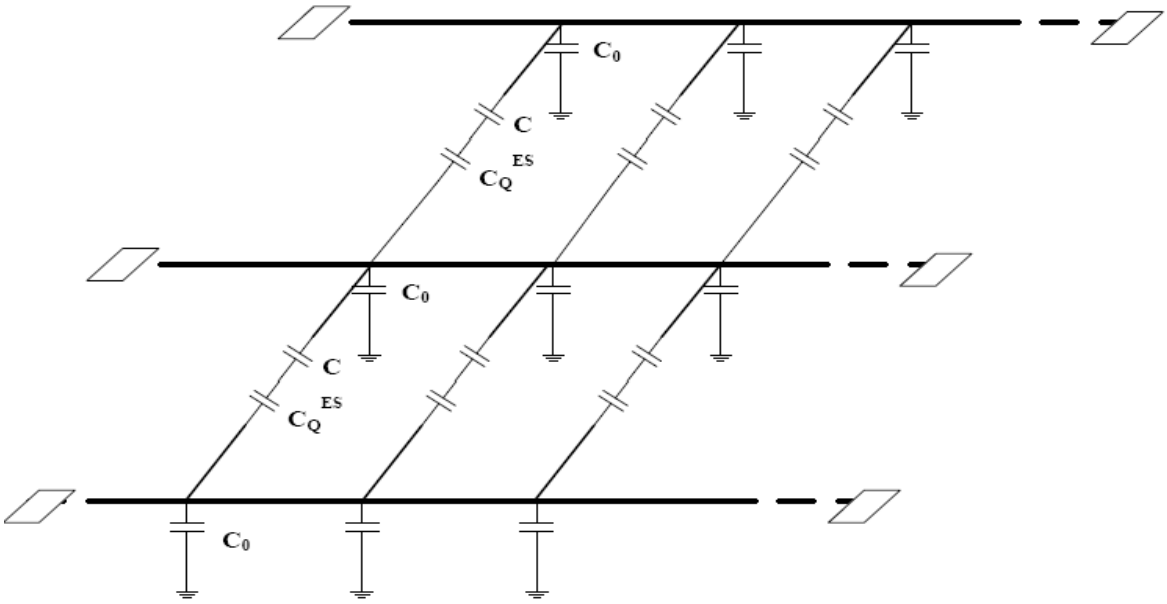

Figure 27. Capacitor network between two parallel nanowires. The number of nanowires at each electrodes scales with redundancy. For simplicity we show only a 3 × 3 network, with no redundancy at the electrode. The column network is not shown in this diagram for clarity.

Nanowires at each electrodes scale with redundancy. For simplicity, only a 3 × 3 network with no redundancy at the electrodes is shown. The column network is not shown in this diagram for clarity. Given the objective of this analysis is to investigate the relative delay of each redundancy scheme, in addition to among other things the lack of quantitative empirical data on the quantum mechanical capacitance of nanowires, C_Q is assumed to be a constant, and is consequently considered negligible in this model. C_{eff} is modeled as the series addition of the electrostatic and quantum capacitance, added in parallel to the ground capacitance. These facts are illustrated in Figure 27.

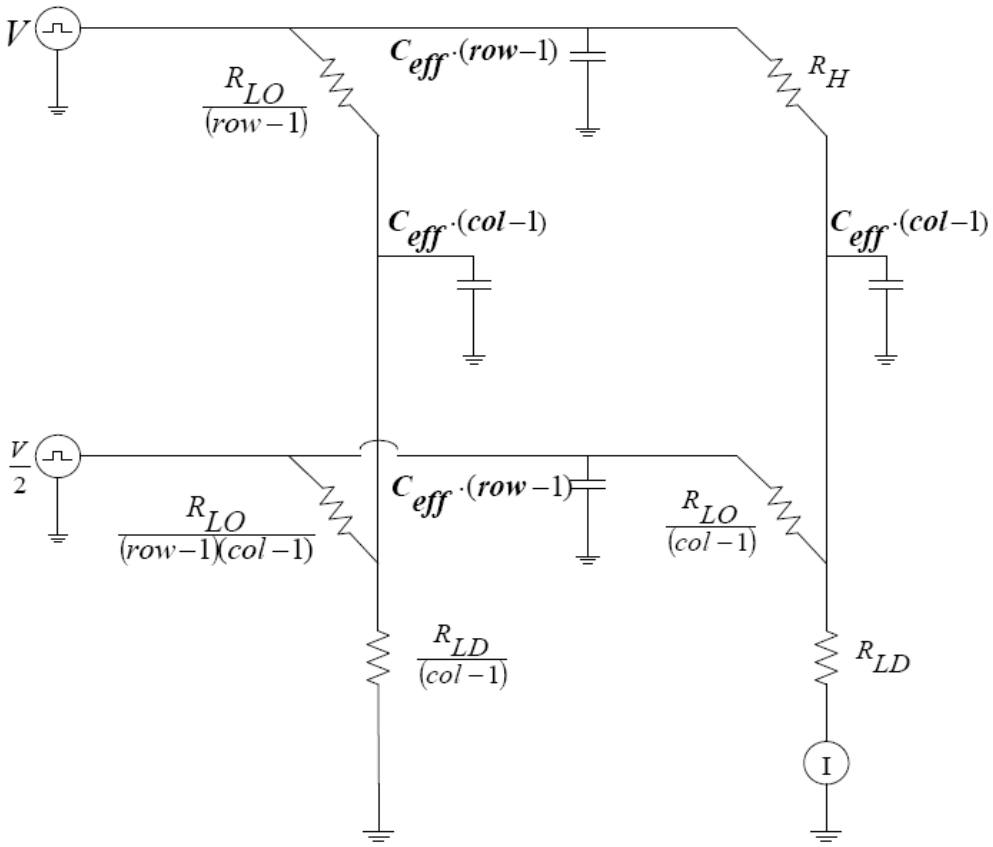

Figure 28. Capacitor parameterized circuit model used to simulate the MSJ crossbar.

Table 1
Parameters used in the parametric analysis SPICE model

Parameter	***Values***
R_H (On resistor value)	**0.5MΩ**
R_{LO} (Off resistor value)	**9MΩ**
R_{LD} (load resistor value)	**Matlab generated using Milliman circuit model**
C_{eff}	$C_0 + C_{ES} + C_Q$

The parameters utilized in Figure 28 are specified in Table 1. The difference between the input and output voltage of the circuit, also referred to as the readout margin, is also of great importance. A good read out margin is necessary to achieve a desirable signal-to-noise ratio. The resistances of Silicon nanowires are on the order of ≈ 106 Ω and can be considered negligible because the circuit will be dominated by the much larger resistance of the molecular switches. The input signals on the multiple nanowire electrode row (column) will switch at the same time, making it practical to consider the effects associated with destructive signal interference on each electrode row (column) as being negligible. This configuration could be considered advantageous since the constructive interference between the signals will help sustain the output signal integrity.

It should be noted that the effective storage capacity of the memory is derived by dividing the array size by the degree of redundancy being implemented; in other words, the array size is divided by *k*.

6.1.1 Circuit Analysis

HSPICE, as earlier discussed, was used to implement the parameterized circuit model of the MSJ crossbar nanomemory. The theory behind inductance in molecular electronic devices are still not well understood, and make no significant contribution in a parameterized analysis, as our main goal is to analyze the relative impact of redundancy in the proposed scheme. Hence, it has been excluded in the parameterized model.

The simulations performed are based on the signal delay when there is a single "1" bit or closed molecular junction residing in a single electrode row and column of the crossbar molecular switch memory array, while, the remainder of the crossbar array are assumed to all have junction devices in the "OFF" state [1].The circuit was modeled as a voltage divider with two voltage sources, just as stipulated in Millman's theorem [118]. In this way, the magnitude of variability of the readout voltage was computed as a function of the load resistance RLD. To obtain the appropriate resistance required to achieve an ON/OFF ratio or noise margin of 10, we reduced the circuit model shown in Figure 27 to a voltage divider circuit with two voltage sources using Milliman's theorem. The simultaneous equation yielded, was solved using Matlab and then sourced into Hspice. The on/off ratio of the molecular cross-point junctions is a very important metric, as it dictates the size of the memory array that can be fabricated. The farther the storage cell is from the voltage drivers, the worse the on/off ratio obtainable.

An AC pulse was used to drive the desired word-line in the circuit, the bit-line was assumed to be floating. The unused switches in the circuit were biased to one-half the reading voltage. In this way, they remain at a distinct voltage and are for the most part

unaffected by the reading process. Each word-line and bit-line was modeled as columns (rows) resistors in parallel (series), depending on the degree of redundancy implemented. The same was also the case for the model capacitors, Figures 27 and 28 illustrate.

A 7 v read signal was used on an 8×8 size array, in our analysis the same voltage value was implemented, however, it should be noted that in the absence of gain in the circuit a higher value voltage source would be required for larger array size. Increasing the voltage has its draw backs; power scales linearly with voltage thus, an increase in voltage adversely increases the power consumed by the nanomemory array. This translates to increased heat dissipation, which can be detrimental to the nanomemory array.

The simulation was done with respect to a worst case scenario. Hence, all analysis is carried on the farthest bit from the input voltage source and the readout current source, which is at cell (n, m); the n and m parameters reference the crossbar row and column, respectively; in addition to all non-selected bits being in their high resistive "OFF" state.

6.1.2 Performance Analysis Results

The addition of redundant nanowires, coupled with an associated increase in the number of resistors and capacitors make for a longer traversal path for the input signal. Analysis show that the MSJ scheme scales well for large memory sizes, both in terms of delay and reliability (described in section 6.1.1). The signal delay (Figure 29) follows an exponential trend, as the number of switching junctions increase exponentially with rising *k* as expected. Power dissipation, as observed in Figure 30, followed this trend for similar reasons.

Analyzing the delay penalties as a function of the percentage difference from the k = 1 case as k is increased, suggest the MSJ scheme would be most suitable for smaller memory configurations. The delta delay graph (Figure 31) shows the percentage change in delay with respect to increasing array size and redundancy k. It is also apparent that a significant penalty is paid as k increases. In the analysis performed, k was varied from k = 2 through k = 10, in order to demonstrate the power and delay trends. However, k cannot be arbitrarily increased because the larger k is, the larger the memory size, which leads to an increased signal path from voltage source to the sense amplifiers. The adverse effect of this is low signal-to-noise ratio at the sense amplifiers, in addition to a low bit per unit area storage capacity on the nanomemory chip.

Figure 31 presents a clearer picture of the difference in delay with varying k. The parameterization of the circuit model, in addition to the absence of gain prevents scalability of the model to larger array sizes. At k = 1, if the nanomemory is partitioned into banks [47], and peripheral circuitry introduced; signal delay would increase, along with hardware cost. However, the MSJ scheme can be used to enhance the banking scheme. This would be a necessary consideration when dealing with larger memory sizes, as nanowires are far more reliable at shorter lengths.

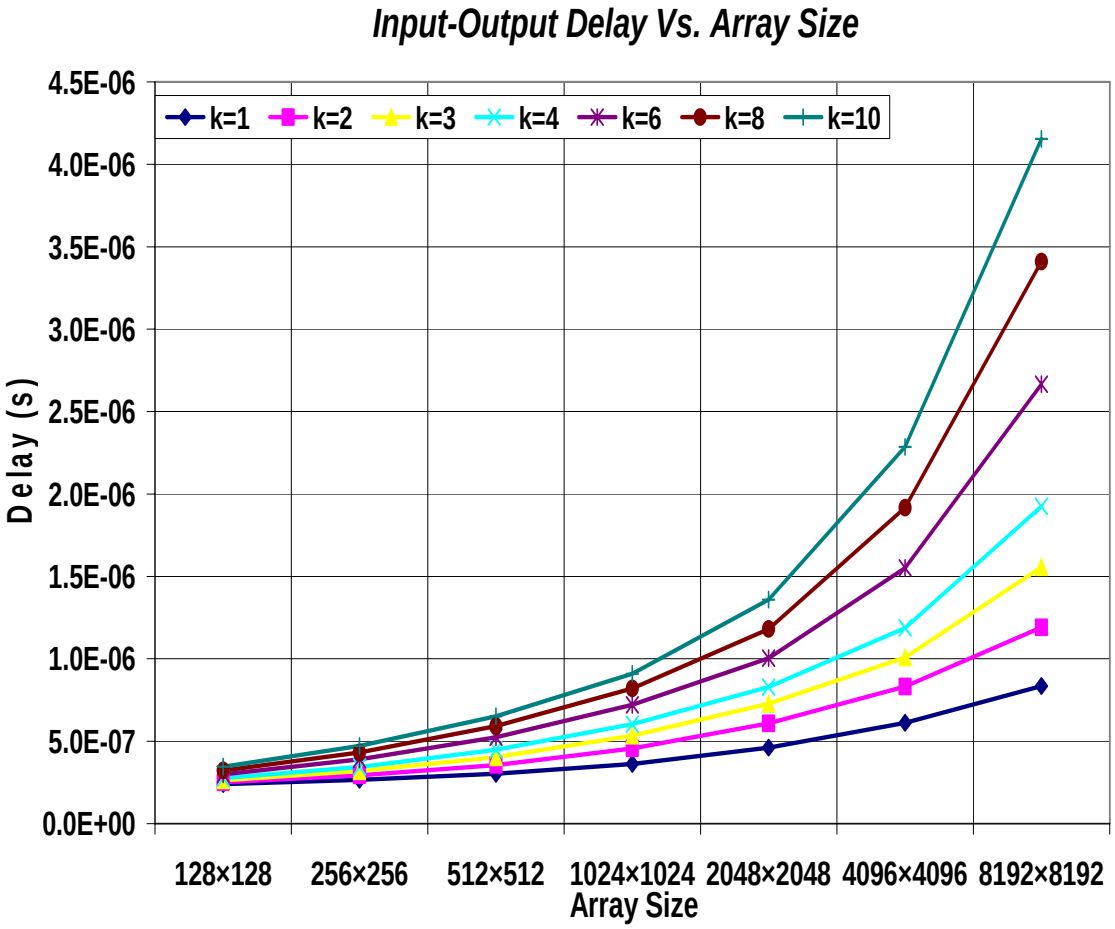

Figure 29. Delay realized when *k* is varied with memory array sizes.

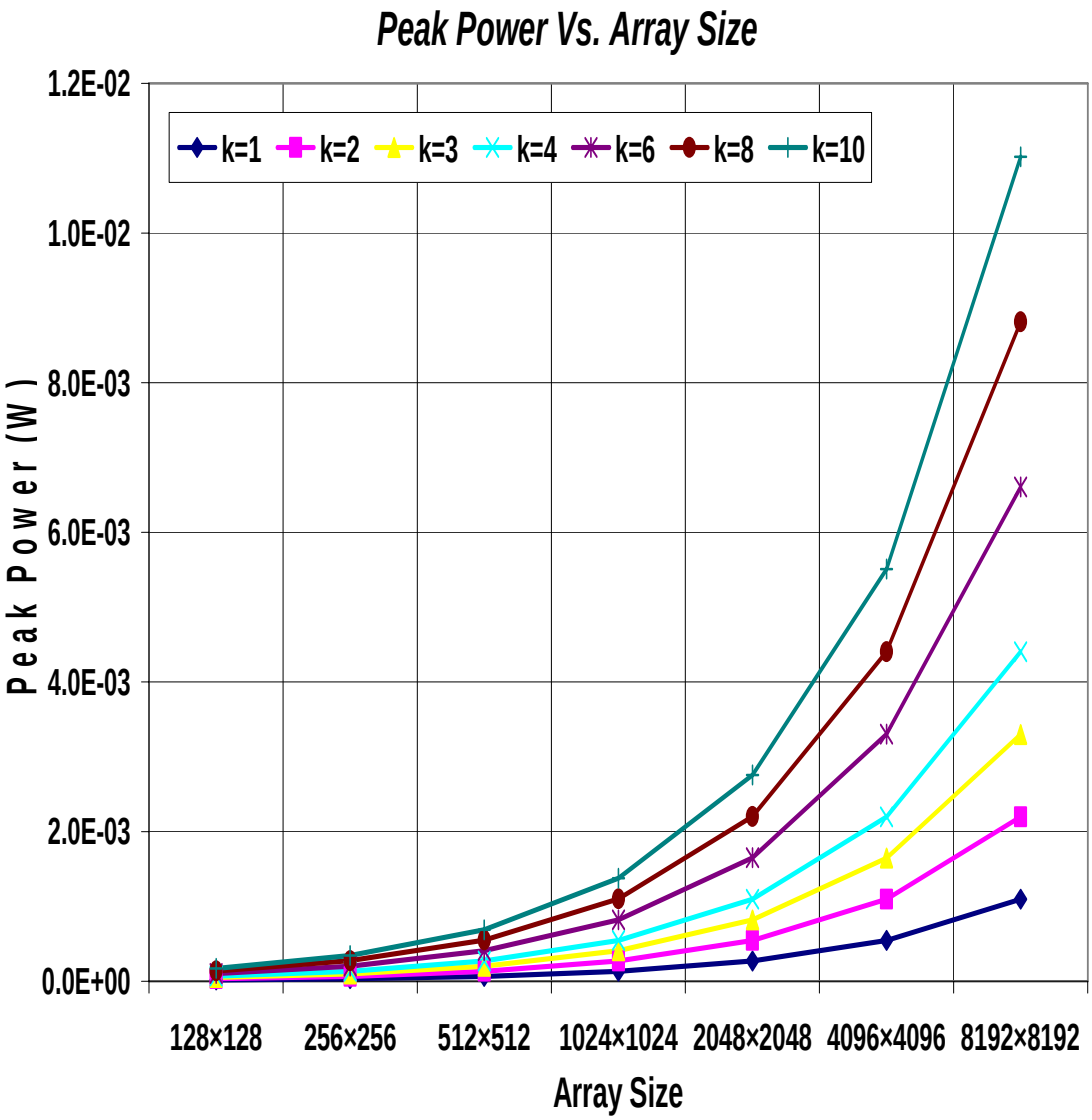

Figure 30. Peak power dissipated as a function of increasing *k*.

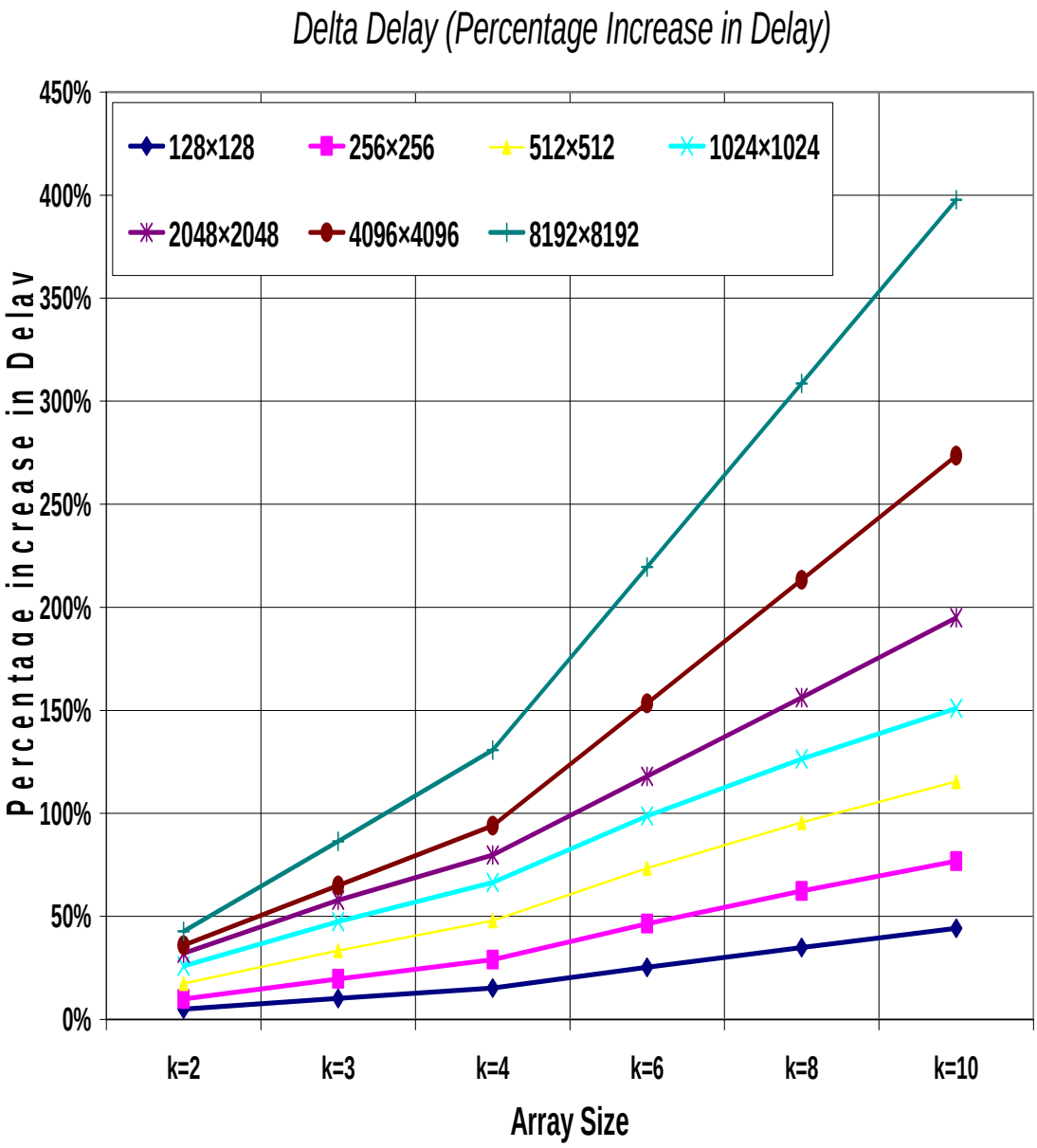

Figure 31. Analysis of delay change between two levels of k, e.g k4-k1 shows difference in delay between k=4 and k=1. All data is normalized with respect to k =1 and the maximum delta delay value.

6.2 Demultiplexer Model Implementation: Bundled Single-Wall Nanotubes Approach

In this section, the effects of using redundant layers of carbon nanotubes, where the carbon nanotubes are replicated in the **x** and **y** –dimension to create what is essentially a bundle of carbon nanotubes, is analyzed and results presented. The demultiplexer, as illustrated (Chapter V, Section 5.2.2 – 5.2.3), consists of the following; CMOS input circuitry, MW input signal lines, nanowire address-lines, Molecular switches modeled as resistors. On each nanowire address-line, the combination MW to nanowire resistor connections cumulatively carry-out an AND gate functionality for address selection. In [135], we demonstrated a small scale analysis of a fault tolerant demultiplexer. An analog demultiplexer circuit model was simulated using the SaberSketch [78] circuit tool. Delay measurements were conducted using the Synopsys CosmoScope waveform analysis tool [119].

Our previous work was limited by the simulation tool, which prevented efficient scaling up of the demultiplexer array sizes. As a continuation of this work, large demultiplexer arrays were studied and different architectural configurations researched. The penalty paid by incorporating the molecular switch junction defect tolerant scheme in the ECC scheme is manifested primarily through the additional surface area required for implementation. This fact will also be scrutinized to assess its merits based on the reliability benefits justifying the scalability cost.

6.2.1 Nanowires and Transmission Models for Demultiplexers

As interconnects shrink, they experience an increase in resistivity and become more susceptible to parasitic quantum effects. In this section, the nanowire interconnects are modeled as Single Wall Metallic carbon Nanotubes (SWNT). The primary reason being, SWNTs have been extensively studied, and as such their electrical properties are better understood, more so than most other types of nanowires. SWNTs also demonstrate ballistic electron flow with electron mean paths that measure in the order of several microns [8], and have the ability to conduct large current densities. Studies analyzing the potential use of SWNTs as transmission lines have been conducted [117, 120, 121]. Performance analysis comparing metallic SWNT and Cu for interconnect application has also been reported [121]. It has been shown that SWNT bundles have a better performance than individual SWNT interconnects [121, 122]. The electrical characteristics of SWNTs have been extensively studied by other groups and reported data used in this work [121, 122].

6.2.2 SWNT Transmission Line Model

As earlier stated, interconnects discussed here are modeled as bundle of ballistic SWNTs. The details of the various properties of SWNTs are given in the following subsections;

6.2.2.1 Inductance

The inductance of SWNTs has been computed in [97, 123]. For the case of a one-dimensional system, the kinetic inductance per unit length of a SWNT is given by the Equation (6-4):

$$L_k \quad \frac{h}{2e^2 v_f} \tag{6-4}$$

where ħ is Planks constant and v_f(≈ 8 × 105m/s) is the Fermi energy of graphene. When computed, the kinetic inductance yields L_k = 16 nH/μm, which is the value utilized in our SWNT transmission line model.

6.2.2.2 Capacitance

Parallel SWNT interconnects experience two distinct forms of capacitance [97, 117]. The first form of capacitance is a coupling electrostatic capacitance C_{ES}, given by Equation (6-5) below:

$$C_{ES} = \frac{2\pi\varepsilon}{\ln(h_g / d)} \tag{6-5}$$

where h_g is the height displacement of the SWNT above the ground plane substrate (illustrated in Figure 31) and d is the diameter of the SWNT. The second form of

capacitance is the quantum capacitance denoted by C_Q and expressed by Equation (6-6):

$$C_Q = \frac{2e^2}{hv_f} \tag{6-6}$$

The bundles are deemed to be densely packed, resulting in very weak quantum capacitance between the SWNTs in the bundle [121]. This presumption has also been verified by previous investigations [124, 125].

6.2.2.3 Resistance

The relationship between the resistance and the length of a SWNT has been theoretically analyzed [126, 127]. The calculations show that the resistance of a ballistic SWNT is dependent on its mean free path and interconnect length as given by the Equation (6-7):

$$\boldsymbol{R} = \boldsymbol{R}_b \left(1 + \frac{\boldsymbol{L}}{\boldsymbol{L}_\lambda} \right) \tag{6-7}$$

where L is the interconnect length, L_λ is the mean free path length, and R_b is the resistance of the ballistic SWNT bundle. The formula for R_b is given below in Equation (6-8);

$$R_b = \left(\frac{h}{4e^2 n} \right) \tag{6-8}$$

where n is the number of nanotubes in the bundle. At the 22 nm node year—which this model is predicated on—L_λ, as estimated will be about approximately 10 μm [89].

6.2.2.4 Conductivity

The conduction of a ballistic SWNT (G) is given by the Landauer formula as expressed in equation (6-9):

$$G = \left(\frac{4e^2}{h} \right) T \tag{6-9}$$

where T is the transmission coefficient of the SWNT, In this work ideal contacts are assumed, whereby $T = 1$, which yields $G = 155$ μS.

6.2.2.5 Copper (Cu) Transmission Line

The electrical parameters for Cu at the 22 nm node year are more concisely derived from ITRS specifications [128]. Inductance for Cu interconnects is predicted to be approximately 1 nH/mm [129]. The capacitance between adjacent Cu microwires can be modeled using the parallel plate capacitance formula given in Equation (6-10).

$$C_{Cu} = \left(\frac{\varepsilon h_{Cu} l}{d_{adj}} \right) \tag{6-10}$$

where h_{Cu} is the C_u wire height, l its length and d_{adj} is the separation distance between two adjacent Cu interconnects. Resistance per unit length for Cu at the 22 nm node is estimated to be approximately 2.2 μΩ/cm.

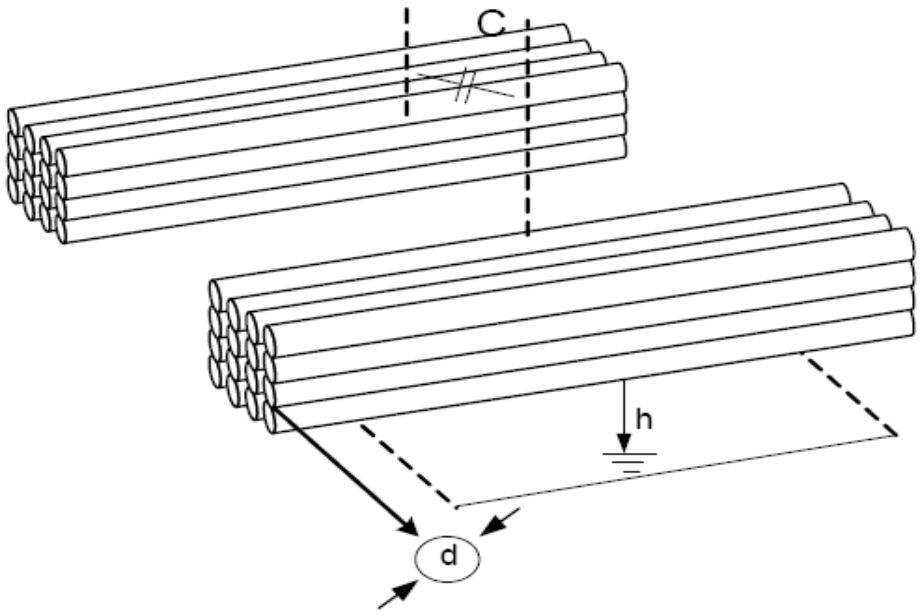

Figure 32. Schematic of adjacent nanotubes bundle showing their parameters.

The parameters presented in the section provide the basis for a workable RLC model for both the SWNT and Cu microwire used to implement the combined scheme for the demultiplexer.

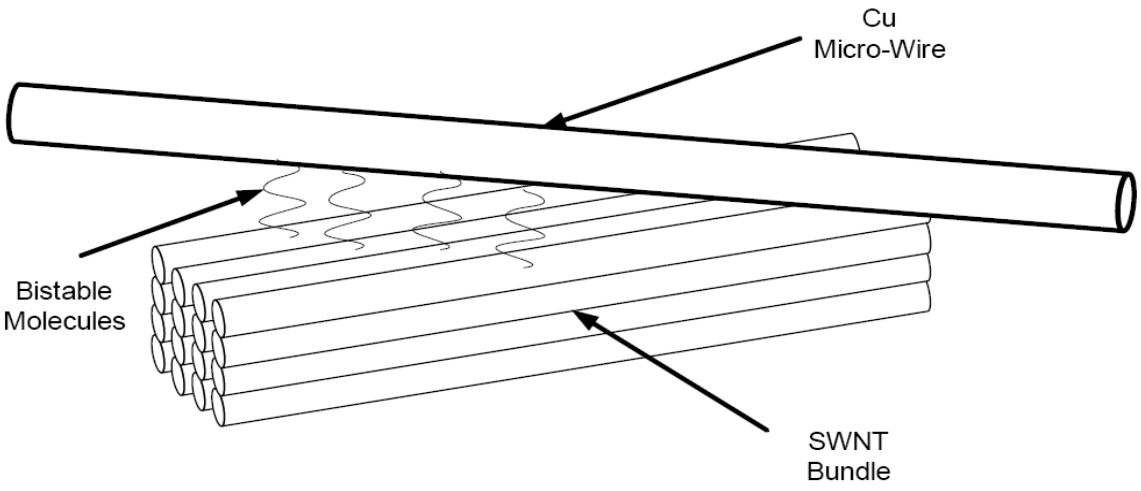

Figure 33. Diagram showing the SWNT-microwire junction with redundancy of $k = 4$. The redundancies in the bistable molecular junctions are also illustrated.

6.2.3 Demultiplexer Delay and Power Analysis Results

In accordance with earlier reported results in this thesis, it has been established that the introduction of fault-tolerance into the demultiplexer design affects its performance. Using an analog demultiplexer circuit model, which incorporates the nanowire and microwire RLC transmission line models described earlier, the fault-tolerance induced signal delay is investigated in this section.

The SaberSketch [130] circuit simulator was used to construct and simulate the demultiplexer circuits. Delay measurements were performed using the Synopsys CosmoScope waveform analysis tool [119]. All delay measurements were determined as a function of signal rise-time at the nanowire address-line output.

A 170 kΩ resistor was used as the "ON" resistance of the molecular junction; this

value is based on empirical data from [131], which indicates a maximum hysteresis current of (10×10^{-6}) Amps when approximately 1.7 V was applied to the molecular switch junction. The spacing between adjacent interconnect bundles was assumed to be 1 nm or the diameter of a single SWNT. Similar assumptions were made for the microwires. For clarity and first order estimations, ideal conditions were assumed to eliminate effects resulting from leakage current with unselected SWNTs address-line. Hence, all unselected SWNT interconnects were grounded. Each address-line consisted of densely packed SWNT bundles, depending on the degree of redundancy implemented. This mitigates effects resulting from parasitic coupling capacitance within the SWNT bundle. The only capacitance given in the model is due to interconnect bundles as discussed in the previous section. The resistance per unit length for each address-line SWNT bundle decreases with increasing redundancy; a result of each SWNT in the bundle contributes to conduction.

Redundancies of $k = 1$ (no redundancy), 2, and 4 were implemented here. Having $k = 2$, signifies an interconnect bundle that consists of four SWNTs and two molecular switches. The SWNT bundles are stacked vertically in a $k \times k$ geometric configuration.

The delay analysis was based on a worst case lower bound scenario. In other words, the signal delay accrued from selecting the address-line output farthest from the driving input signal. All simulations were made under the assumption that every molecular junction functioned correctly. Depending on the degree of redundancy, each junction comprised of k molecular switches in parallel, having a singular connection to the driving micro-scale column wire.

Results indicate that the inclusion of redundancy at the nanowire level improves signal delay. This is consistent with projections from recently published works [22, 120, 121]. The delay analysis graph of Figure 34 shows better delay improvement with increasing redundancy. The penalty paid by incorporating the molecular switch junction fault-tolerant scheme is manifested primarily through the additional surface area require for implementation. A scale model of the demultiplexer was used in this work; this made it virtually impractical to simulate larger array and redundancy sizes because of the extensive simulator memory requirements and the GUI interface of the simulator. In light of the presented result, it has been determined that the molecular switch junction scheme clearly enhances the performance advantages provided by the ECC scheme.

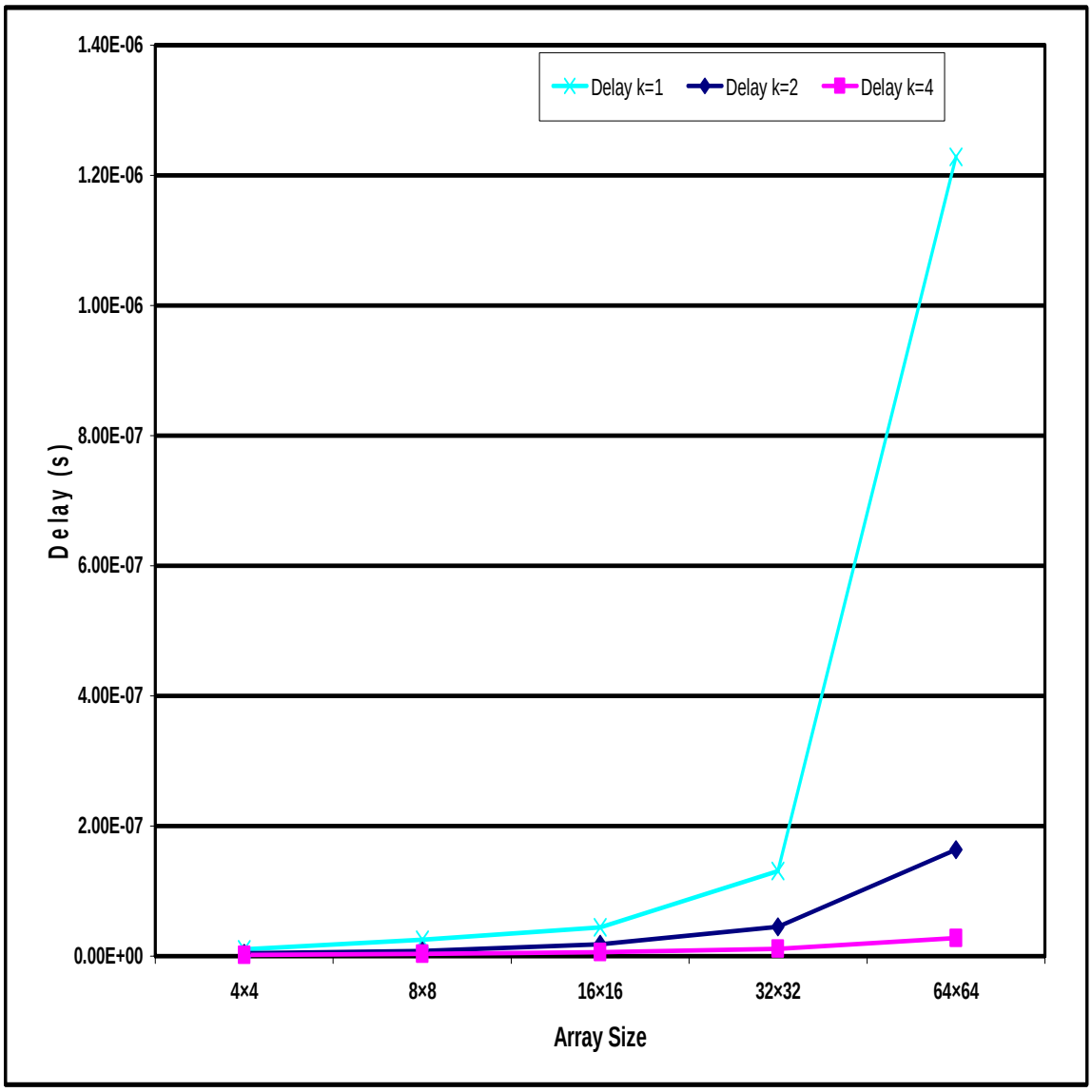

Figure 34. Delay incurred by using redundant SWNT bundles as interconnect address-lines.

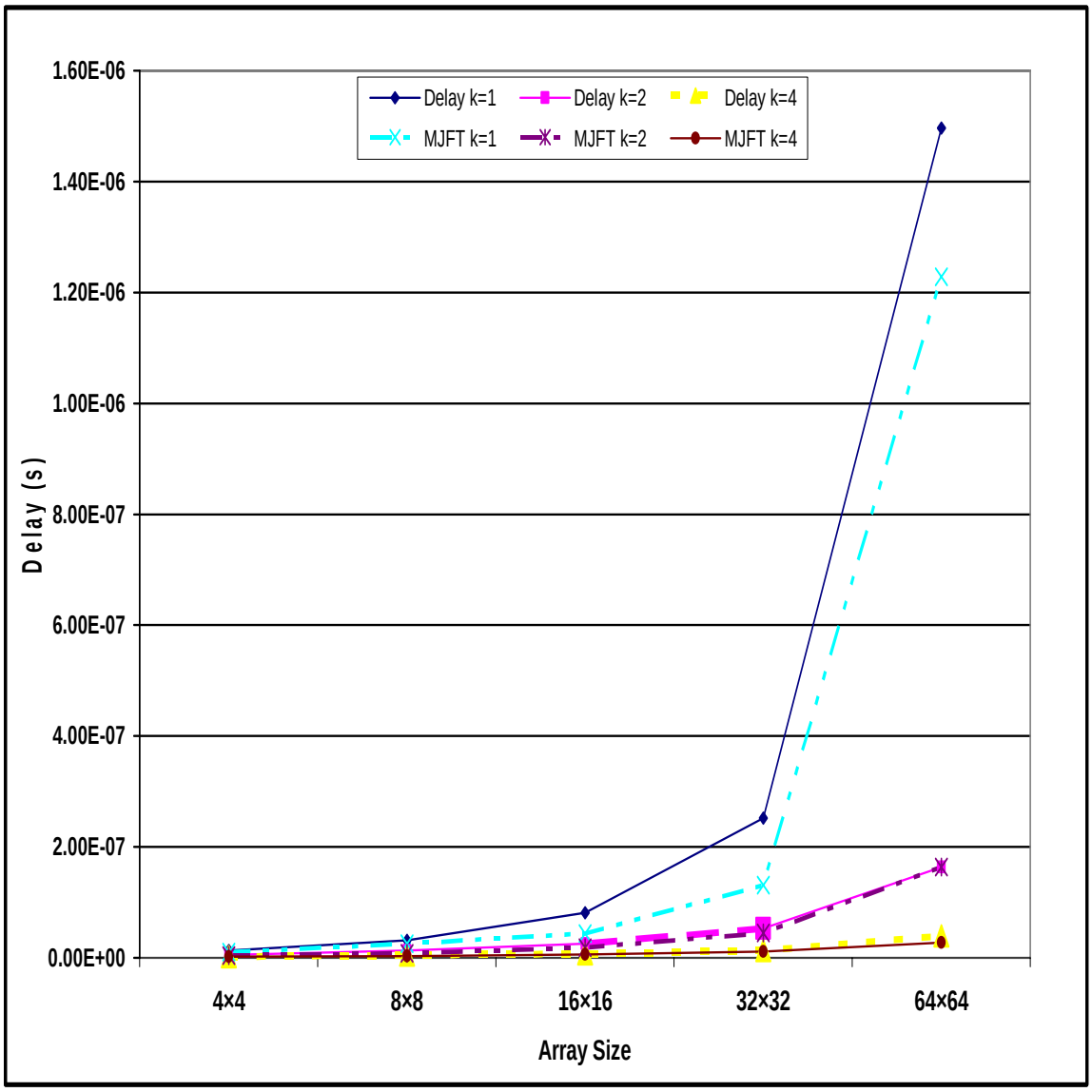

Figure 35. Delay comparison between the multi-junction fault-tolerance scheme and the multi-junction enhanced ECC fault-tolerance scheme.

6.3 Scaled Multi-Switch Junction Crossbar Nanomemory and Demultiplexer Circuit Model Analysis

In section 6.1, a parameterized circuit model was utilized to analyze the performance of the crossbar nanomemory [107]. In this section, a full scale model of the crossbar nanomemory and demultiplexer are implemented; the motivating factor being to develop a more detailed understanding of the fault-tolerance induced performance penalties incurred in the crossbar nanomemory and demultiplexer devices. The circuit models were implemented and simulated in HSPICE. The bistable molecular switch junctions were modeled as resistors and the redundant nanowires were modeled as interconnecting RLC transmission lines as described by Burke et al [117] and illustrated in Figure 37. A diagram illustrating the scaled MSJ crossbar and demultiplexer is also provided in Figure 36.

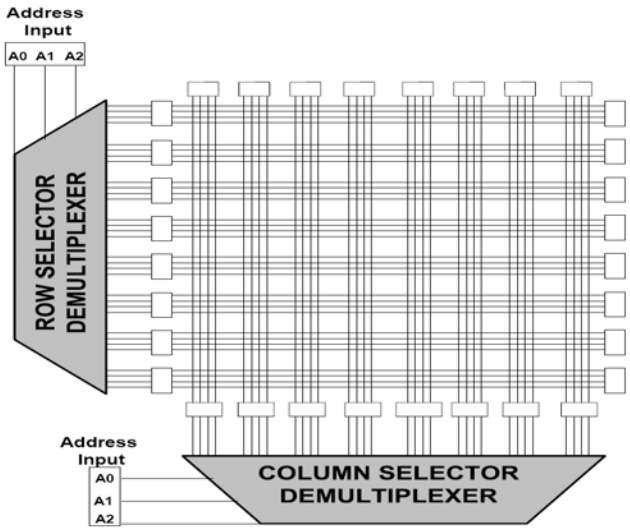

Figure 36. MSJ crossbar and demultiplexer nanomemory; this illustration shows, as an example, an MSJ implementation of $k = 4$.

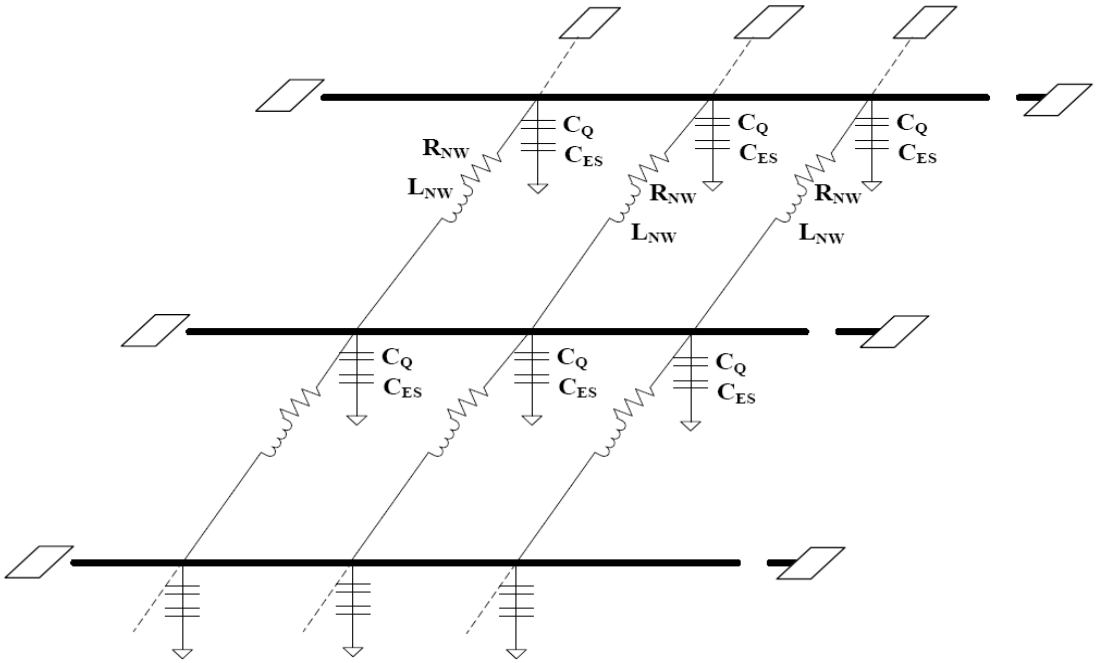

Figure 37. RLC network schematic for crossbar nanomemory.

C_{ES} and C_Q are the electrostatic and quantum capacitance, and are the same as were used in Equations (6-1) and (6-3). They are again provided in Equations (6-11) and (6-12) respectively, for easy reading. Where v_f (computed in *m/s*) is the Fermi velocity of the nanowire.

$$C_{ES} = \frac{L\, 4\, \varepsilon_o \varepsilon_r}{\pi\ \ln(\ 4\, d\ /\ w\ )} \qquad (6\text{-}11)$$

$$C_Q = \frac{2\, e^2}{h v_f} \qquad (6\text{-}12)$$

The capacitance of the MW was modeled as the parallel plate capacitance [Equation (6-13)] between the MW and the chip substrate and is calculated as a function of the parameter d_{dist_sub} , as observed in Equation (6-13) below:

$$C_{MW} = \left(\frac{\varepsilon h \ L}{d_{dist\_sub}} \right) \tag{6-13}$$

The nanowires were modeled as copper (Cu) nanowires; the resistance per unit length of the Cu nanowires was computed using the resistivity of bulk copper material [84]—according ITRS specifications—and is provided in Equation (6-14). The resistances of the demultiplexer MW address-lines were also computed using Equation (6-14), and were modified to take into account the MW dimension parameters.

$$\frac{R}{L} = \frac{\rho}{A} \tag{6-14}$$

Only the kinetic inductance L_{k_NW} is used in our nanowire interconnect model because it has been reported that in one-dimensional systems, kinetic inductance always dominates magnetic inductance [129]. Typical inductance values in CMOS on-chip environments have also been reported to be in the region of approximately 1nH/mm or less [132]. The inductance of the demultiplexer address-line MWs were modeled using Equation (6-15). The circuit model of the demultiplexer is shown in Figure 38.

$$L_{k\_NW} = \frac{h}{2e^2 v_f} \tag{6-15}$$

The redundant nanowires described in Figure 16 are connected to the same contact electrode, they are assumed to be densely spaced and driven by the same voltage source. Further, because the redundant nanowires switch in tandem, the effective parasitic capacitance between them is negligible [116].

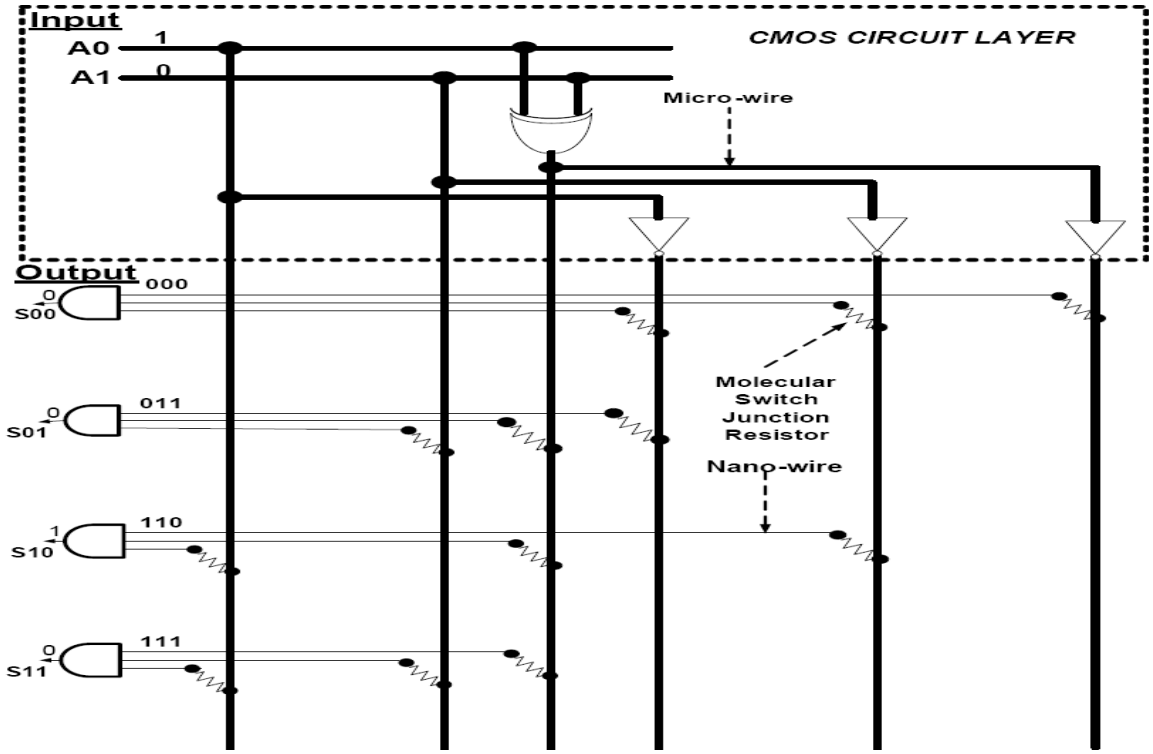

Figure 38. Circuit layout of a crossbar nanomemory demultiplexer with error correction coding. In this demultiplexer, the (S10) address-line is selected as indicated by the high (1) output on the S10 addressline. The extra bit on the address-line outputs are derived by the "1" parity bit added by implementing error correction coding into the circuit.

6.3.1 Simulation Engine Set-up

Due to the relative infancy of nanotechnology, there are no known simulation tools for addressing the performance metrics we intend to research. As a result a substantial part of this work was dedicated to creating developing a simulation strategy using available circuit simulators to meet our intended research goals. The structure of this simulator platform is illustrated in Figure 39. Perl scripts were used to generate a series bit or connection matrices. These matrices represent a map of connections or lack thereof, of the nanomemory:

- ***Nanowires Connection***: - A "0" will represent a connection between adjacent NWs and vice versa for "1".
- ***Storage Bit***: - A high or low state molecular junction will be represented by a "1" or "0" respectively, of the bit matrix.

Once the nanomemory layouts are created, they are converted into an equivalent electrical circuit in the form of an Hspice netlists. Figure 39 below shows the process chart of the simulator modules.

1 ***Bit Generator*****:** - Generates a bit address each NW address-line. The molecular switch junction connections are determined from a 2^n addressing bit matrix.

2 ***Bit Matrix Converter:*** - Converts the address bits into matrix form**.**

3 ***HSPICE Netlist Generator:*** - Reads matrix and generates an equivalent nanomemory circuit.

Figure 39. Simulator module process chart.

Delay is defined in this work as the time required to access and read stored information from a selected molecular switch bit. A worst case or upper bound performance analysis was performed (i.e. the target bit assessed was the bit farthest from the voltage driver source and the sense amplifiers as indicated in Figure 40.

Table 2
Parameters used for the scaled SPICE model analysis

Parameters	***Values***
Nanowire width [73]	**15nm**
Nanowire pitch	**33nm**
Microwire width	**90nm**
Microwire pitch	**180nm**
"ON" Resistance (R_{ON}) [43]	**0.48MΩ**
"OFF" Resistance (R_{OFF})	**9.2MΩ**
Load Resistance (R_{load}) [44]	$\sqrt{(R_{ON})(R_{OFF})}$

All bits on the assessed word-line and bit-line were set to their low resistive "ON" state, while the rest of the non-accessed bits were reverse biased. The nanowire contact electrodes were modeled as voltage sources and the sense amplifiers as load resistors, Figure 40 illustrates. Model parameters are provided in Table 2.

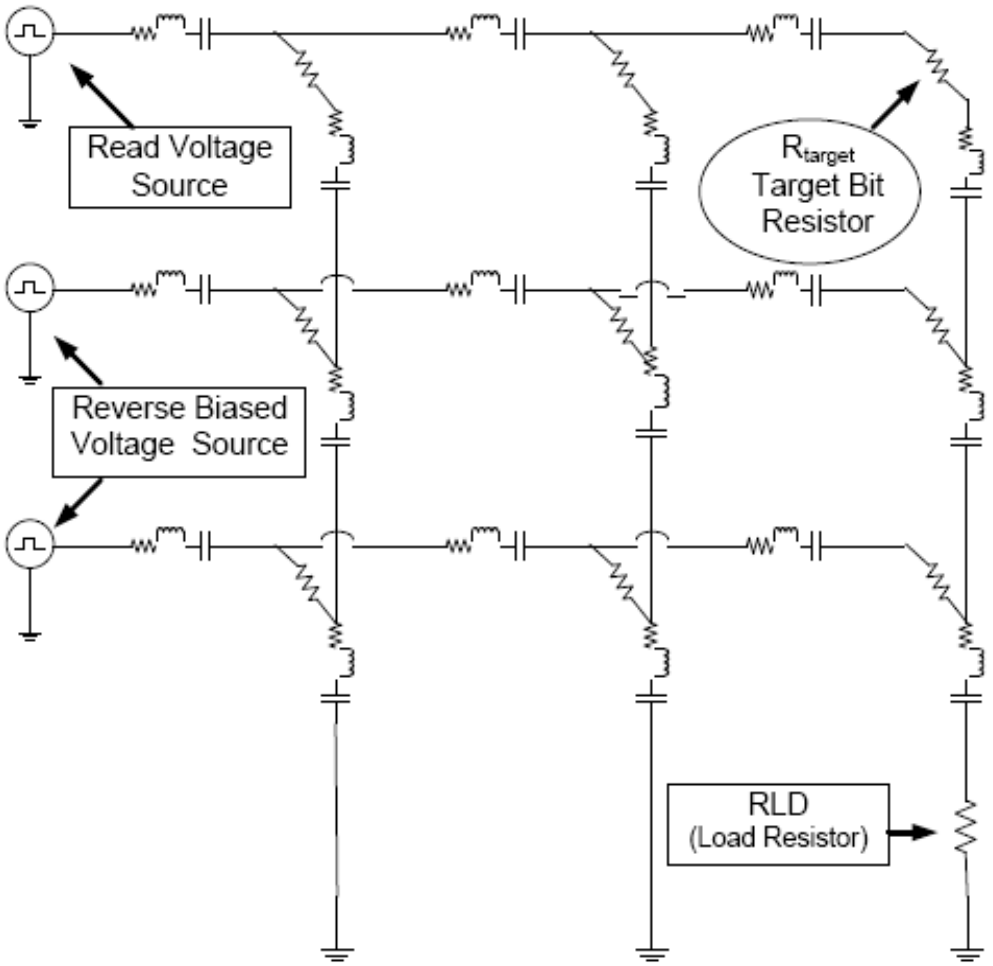

Figure 40. Scaled 3×3 model of the crossbar nanomemory.

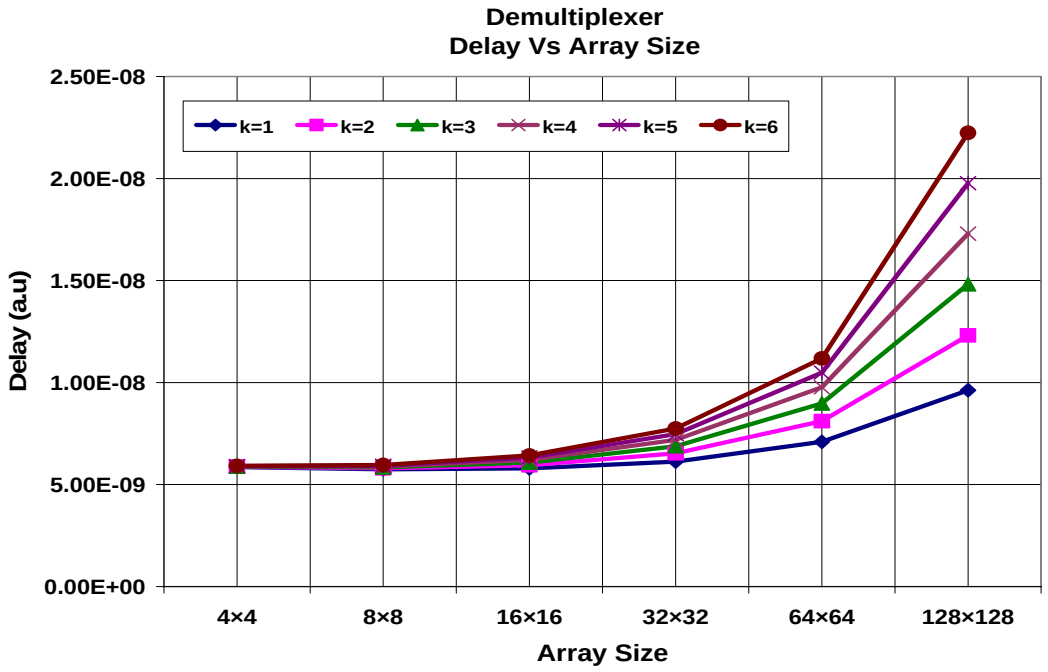

Figure 41. Access time delay penalty incurred by increasing the redundancy *(k)* in the demultiplexer.

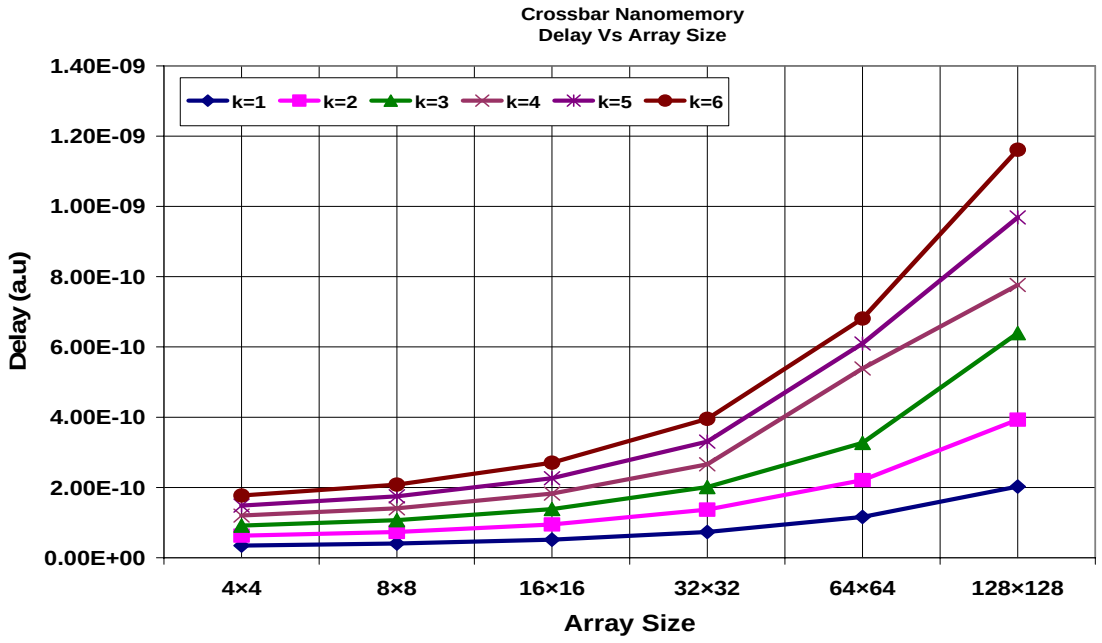

Figure 42. Access time delay penalty incurred by increasing the redundancy *(k)* in the crossbar nanomemory.

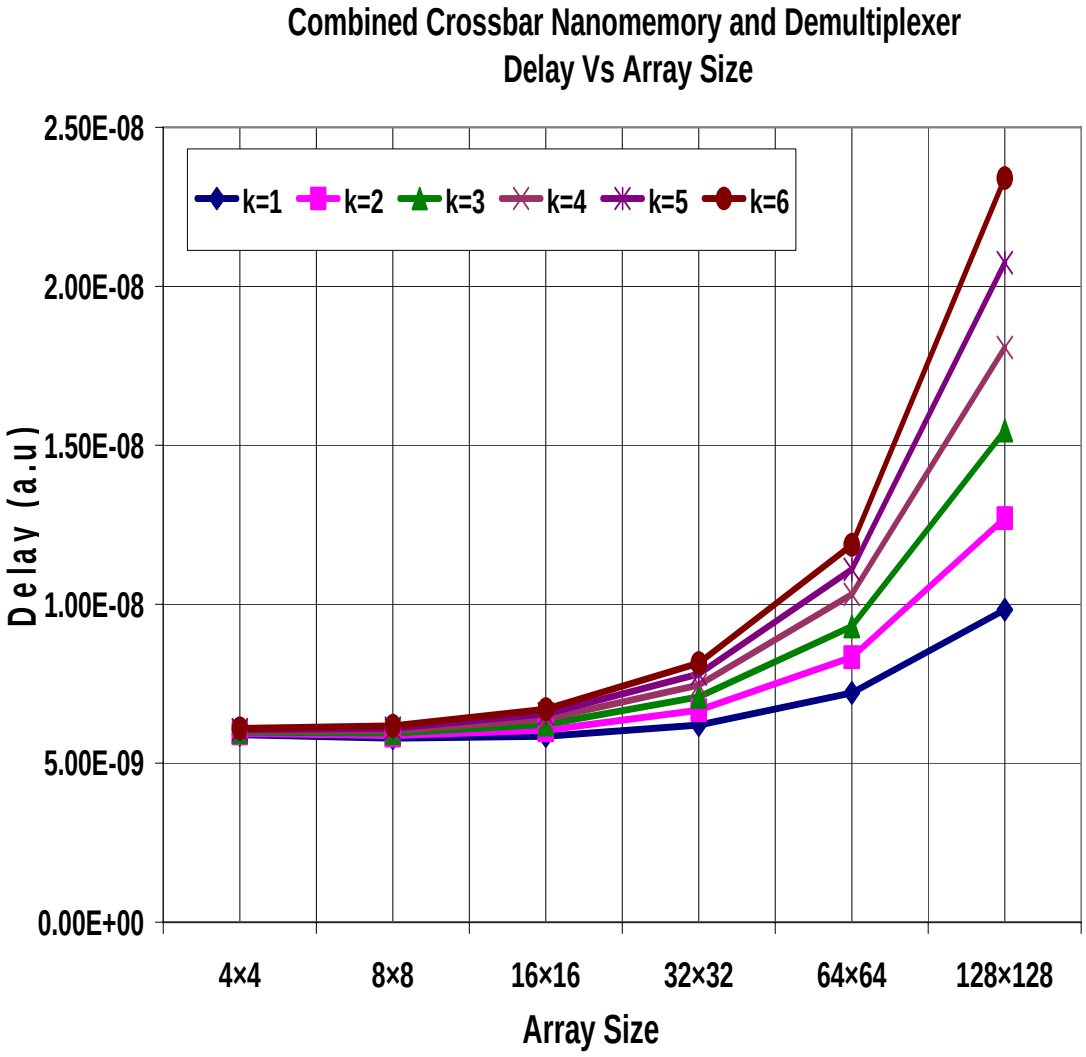

Figure 43. Signal delay measurement of selecting the memory address with the demultiplexer and reading a bit with the crossbar nanomemory. Specifically, graph describes the access time delay of the nanomemory and demultiplexer when increasing levels of *k* are implemented.

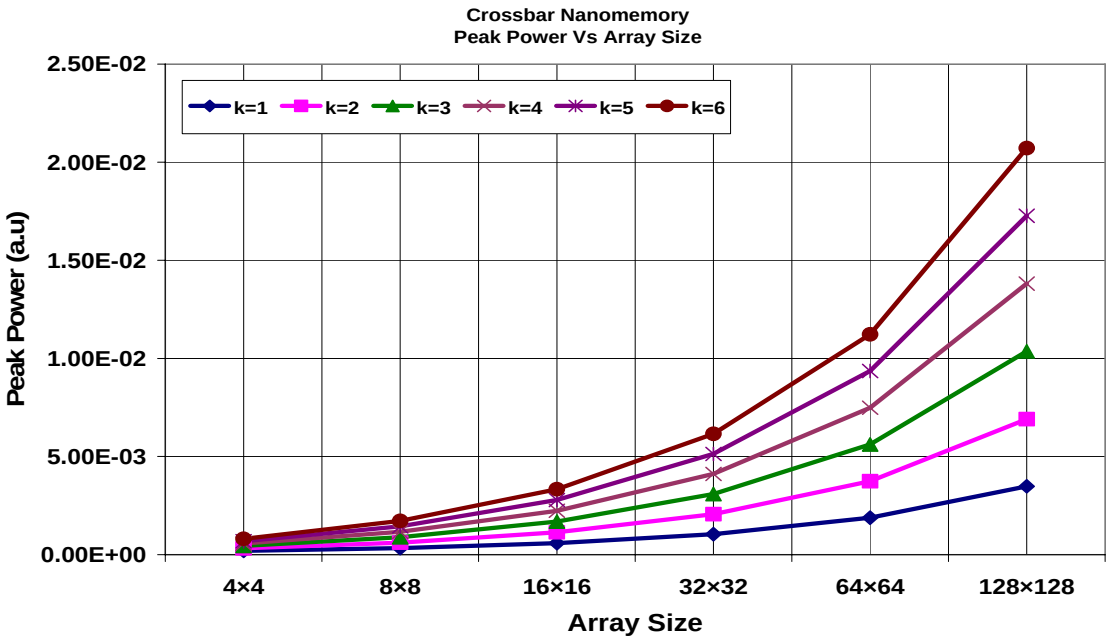

Figure 44. Graph showing the peak power dissipation penalty incurred by increasing the redundancy *(k)* in the crossbar nanomemory.

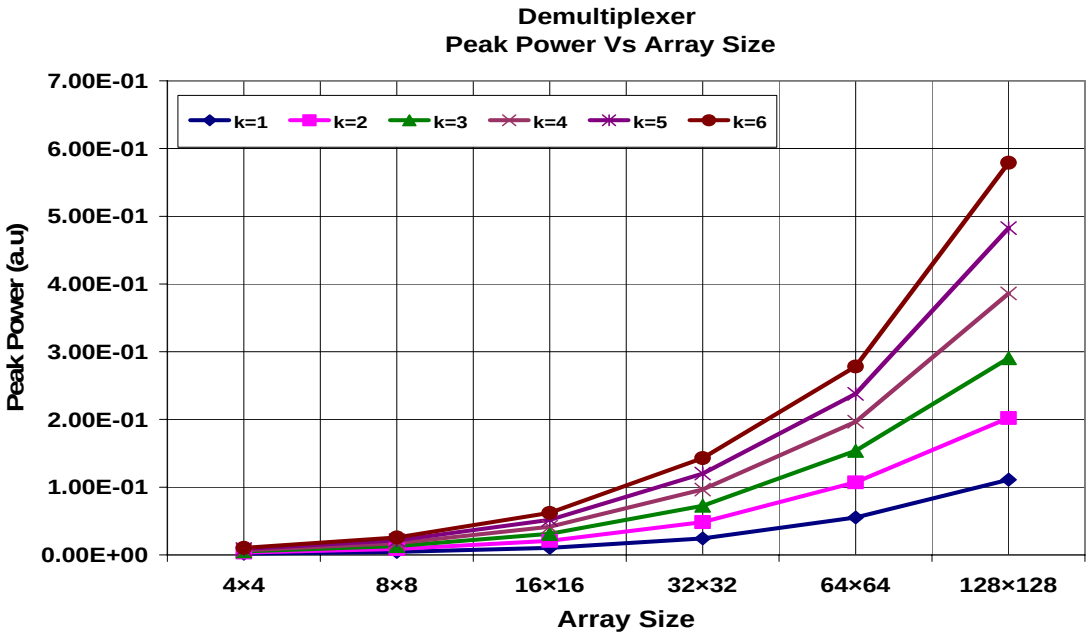

Figure 45. Graph showing the peak power dissipation penalty incurred by increasing the redundancy *(k)* in the demultiplexer.

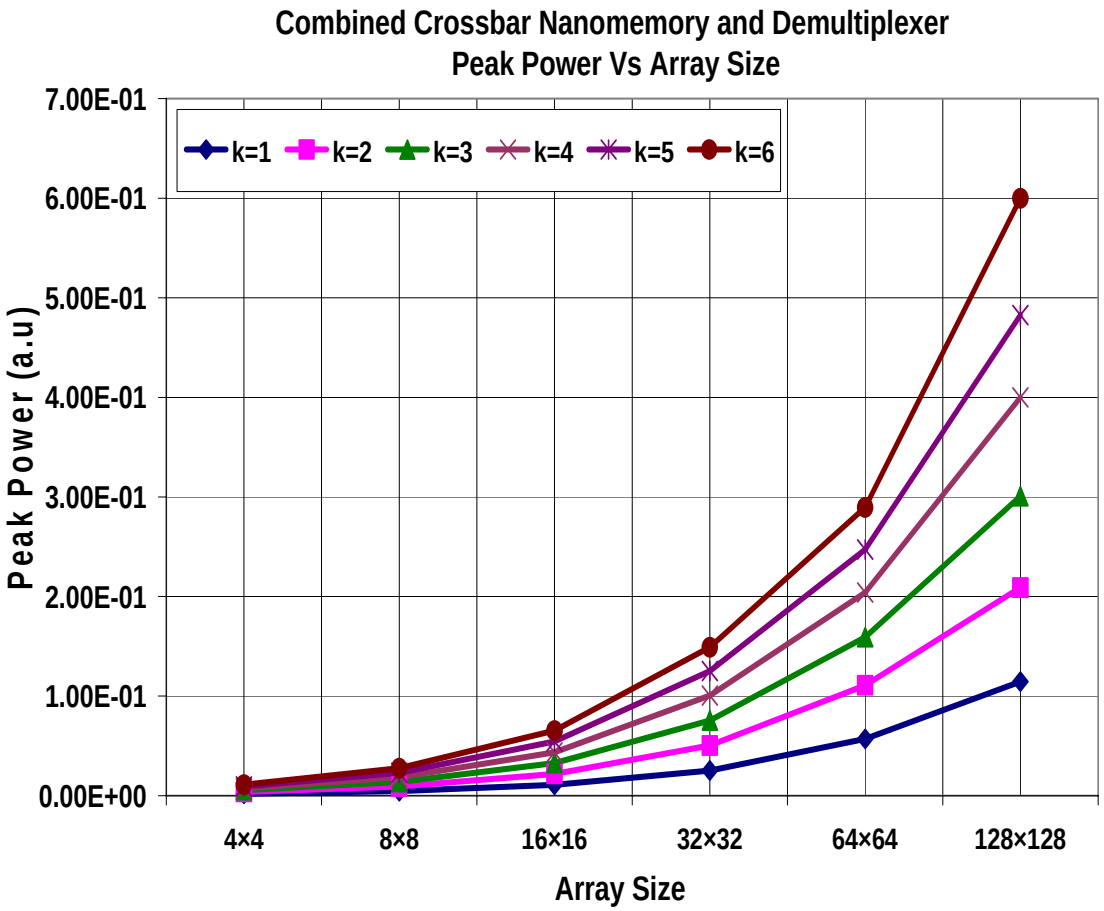

Figure 46. Signal delay measurement of selecting the memory address with the demultiplexer and reading a bit with the crossbar nanomemory. Specifically, graph describes the peak power dissipation trend with increasing degrees of *k*.

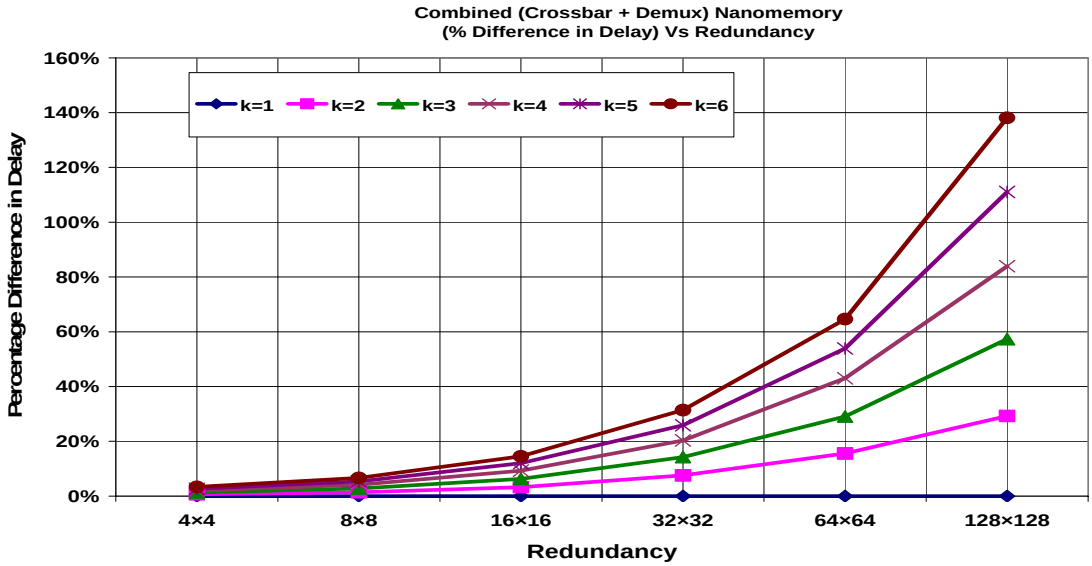

Figure 47. Graph showing the percentage increase in access time delay as a function of increasing redundancy *(k)* in the combined demultiplexer and crossbar nanomemory device. Percentage increase refers to the additional penalty paid by adding one extra level of redundancy. Example, the delay at (k = 3) – delay at (k = 2), computes the penalty of going from a redundancy of two to three.

Table 3

Results from the plot of Figure 47, showing the percentage difference in the increase in access time delay with respect to the baseline $k = 1$ case for a combined crossbar and demultiplexer nanomemory

Redundancy→	**(*k* = 1)**	**(*k* = 2)**	**(*k* = 3)**	**(*k* = 4)**	**(*k* = 5)**	**(*k* = 6)**
Array Size	***% increase Delay***	***% increase Delay***	***% increase Delay***	***% increase Delay***	***% increase Delay***	***% increase Delay***
4×4	Baseline	0.66%	1.33%	2.00%	2.67%	3.35%
8×8	Baseline	1.42%	2.78%	4.10%	5.38%	6.63%
16×16	Baseline	3.26%	6.28%	9.29%	12.01%	14.53%
32×32	Baseline	7.58%	14.24%	20.33%	25.87%	31.40%
64×64	Baseline	15.59%	29.06%	42.99%	53.91%	64.61%
128×128	Baseline	29.27%	57.31%	83.92%	111.04%	138.11%

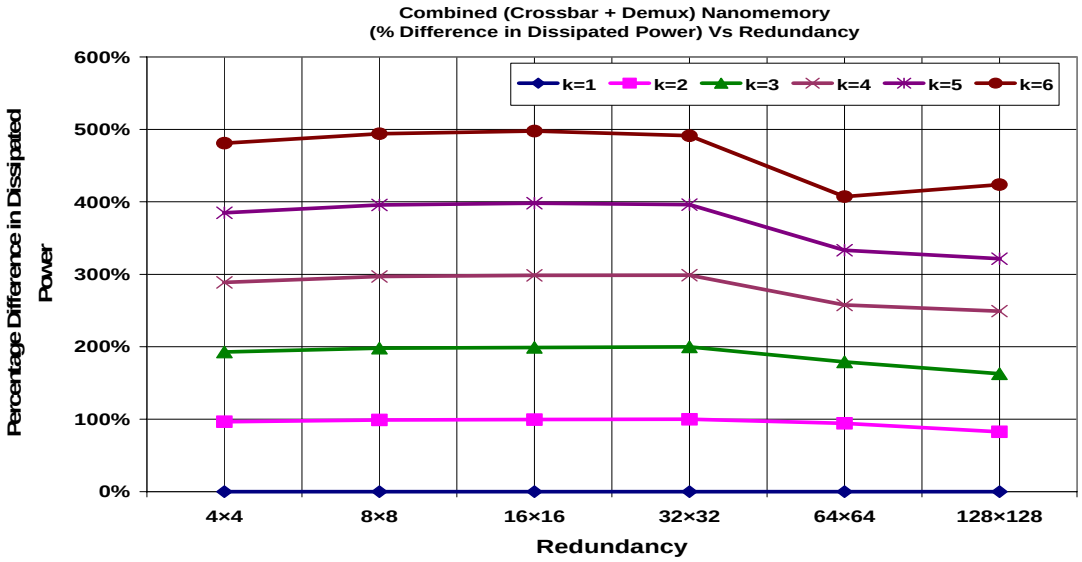

Figure 48. Graph showing the percentage increase in power dissipated as a function of increasing redundancy *(k)* in the combined demultiplexer and crossbar nanomemory device. Percentage increase refers to the additional penalty paid by adding one extra level of redundancy. For example, the power dissipated at (k = 3) – power dissipated at (k = 2), computes the penalty of going from a redundancy of two to three.

Table 4
Results from the plot of Figure 48, showing the percentage difference in to the increase in power dissipated with respect to the baseline *k = 1* case for a combined crossbar and demultiplexer nanomemory

Redundancy→	**(*k* = 1)**	**(*k* = 2)**	**(*k* = 3)**	**(*k* = 4)**	**(*k* = 5)**	**(*k* = 6)**
Array Size	***% increase Power***	***% increase Power***	***% increase Power***	***% increase Power***	***% increase Power***	***% increase Power***
4×4	Baseline	96.53%	192.66%	288.78%	384.86%	480.97%
8×8	Baseline	98.97%	197.97%	296.95%	395.72%	494.02%
16×16	Baseline	99.46%	199.02%	298.59%	398.16%	497.72%
32×32	Baseline	99.92%	199.72%	298.75%	396.15%	491.28%
64×64	Baseline	94.24%	179.05%	257.58%	333.16%	407.19%
128×128	Baseline	82.54%	162.69%	249.21%	321.44%	423.82%

6.3.2 Results

Simulation results demonstrate that access time delay and power dissipation scale with increasing redundancy. Given the exponential increase in the size of the nanomemory array, in addition to the fact that increasing number of redundant elements are being added to the nanomemory array, the linear-like scaling observed in the graphs presented in Figures 41 to 48, can be considered a relatively small penalty to pay for improved reliability. Results also show that delay penalties are higher in the crossbar nanomemory as compared to the demultiplexer. The reason being that redundancy is only implemented in the nanowire dimensions of the demultiplexer, while the CMOS addressing microwires are assumed to be far more reliable than the nanowires due to the matured state of CMOS technology. On the other hand, peak power dissipation penalty is highest in the demultiplexer, which is a direct result of the larger power dissipation observed in the demultiplexer microwires.

When the crossbar nanomemory and demultiplexer results are analyzed as a single device unit, quite a few things become apparent. In the delay analysis of the combined nanomemory device, it can be seen that the penalty paid by adding an extra level of redundancy increases slightly with increasing array size. For example, Figure 46 and Table 3 shows that there is an approximately 29% increase in access time delay going from the redundancy state of (k = 1) to the added redundancy of state of (k = 2), as opposed to the 57.31% increase in delay of going from (k = 1) to (k = 3) and 83.92% going from (k = 1) to (k = 3). The point to highlight here is the less than 100% increase in delay penalty when redundancy is added, in addition to the array sizes being increased by

the redundancy factor. This is because constructive interference between signals and improved conduction due to the additional electron paths created by the addition of redundant nanowires help to reduce delay penalties. These results further reinforce the practical and reliable attributes of the MSJ scheme.

The case is quite the opposite when it comes to the amount of power dissipated. While the constructive interference and improved conduction is advantageous in access time delay considerations, they work adversely in the power domain. In the delay analysis plot of Figure 46 and the data tabulated in Table 3, it can be observed that the delay penalty of implementing a redundancy of (k = 6) is approximately 138% increase over the (k = 1), while that increase comes out to be 423.82% in the power penalty case, as can be seen in Figure 48 and Table 4.

6.4 Demultiplexer Model Implementation: ECC Demultiplexer Approach

In this section, the performance of the crossbar nanomemory demultiplexer with combined ECC and MSJ implementation is presented. In this section we take a two phased approach to the access time delay performance analysis, by analyzing the performance penalty incurred by not only adding redundancy, but by also including varying degrees of error correction coding. This analysis was done solely with respect to a single demultiplexer array size.

The computed results, as presented in Figure 31, show linear scaling in the delay penalty for increasing redundancy in the demultiplexer; the reasons are explained in the previous sections (Section 6.3). The main point of this analysis is that there is no delay

penalty from the inclusion of ECC; in fact the addition off ECC improves the delay performance of the demultiplexer. This counter intuitive result is due to the fact that ECC is implemented by the inclusion of additional better conduction (CMOS microwires relative to nanowires) paths to the demultiplexer circuitry; In other words, the higher the number of microwires, the smaller the penalty paid in access time delay.

The improvement in the delay performance between the implemented (ECC-MSJ) demultiplexers and the (No ECC-MSJ) demultiplexer is relatively higher when ECC hamming distance 2 codes (i.e. **d** = 2) are implemented as compared to the hamming distance, **d** = 3 and **d** = 4 codes. This occurs because of the number of bits required to implement the hamming codes; eleven and twelve bits are needed to implement the ECC (**d** = 3) and (**d** = 4) codes respectively, while the hamming distance (**d** = 2) code is implemented with only eight bits. Take the (k = 3) and (k = 4) cases in Table 5, we see the relative closeness in their delay improvement differences; all calculations were made with respect to the base (k = 1) and No (ECC-MSJ) case. The properties of the ECC codes implemented in this work are presented in-depth in the reliability analysis of chapter VII.

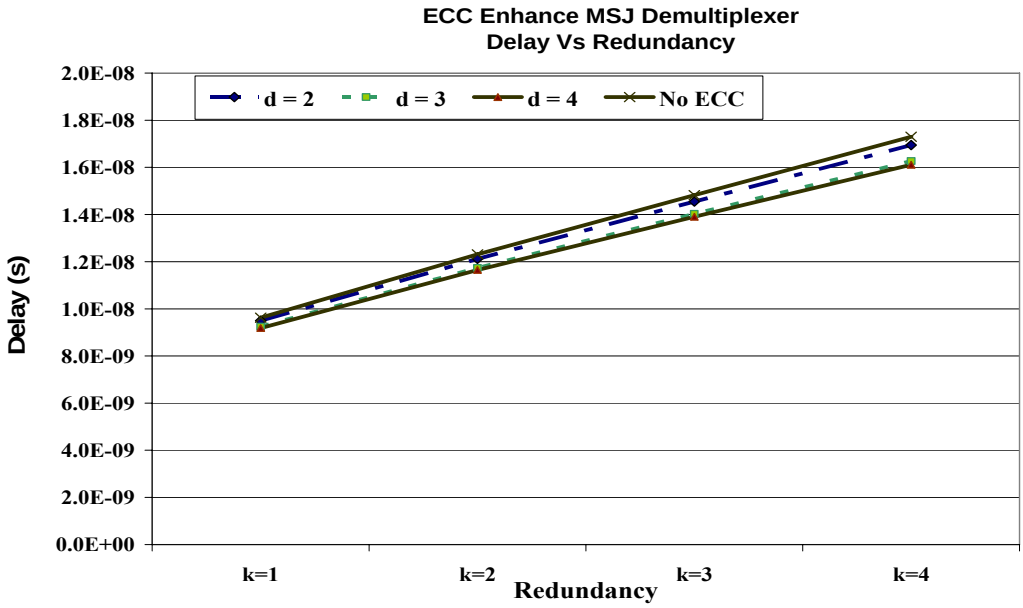

Figure 49. Relative increase in access time delay with increasing redundancy (*k*) and increasing error correction code parity bits, in the crossbar demultiplexer.

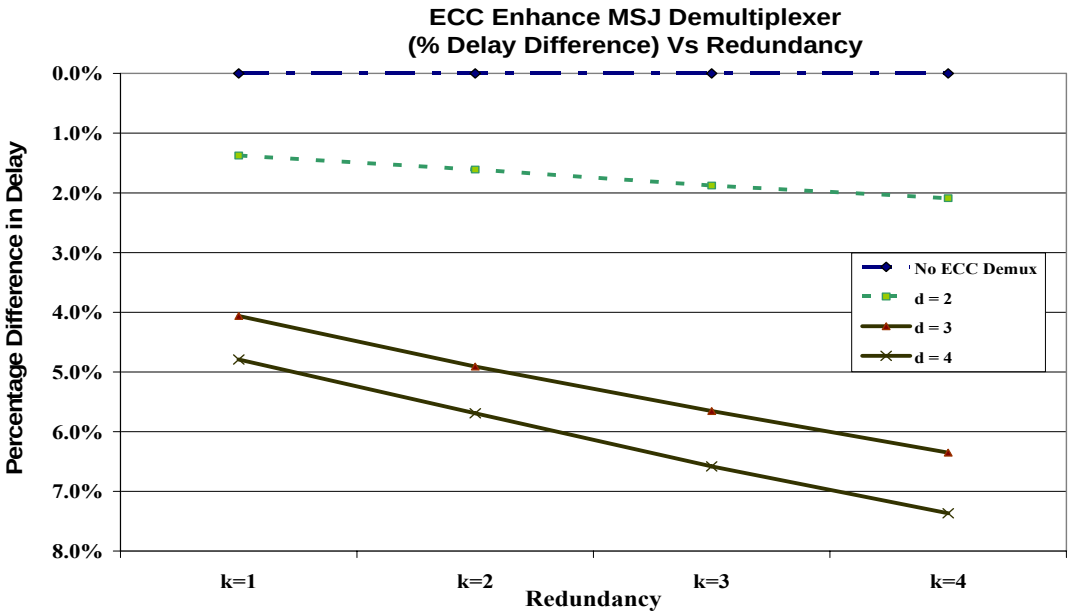

Figure 50. Percentage difference increase in access time delay (as measured from the No ECC baseline case) with increasing redundancy (*k*) and increasing error correction code parity bits, in the crossbar demultiplexer.

Table 5
Results from the plot of Figure 50, showing the percentage difference in the increase in access time delay with respect to the base line $k = 1$ case for a 128×128 demultiplexer implemented with both ECC and MSJ schemes

Array Size →	128×128			
Redundancy	*% Delay Difference* No ECC	*% Delay Difference* d = 2	*% Delay Difference* d = 3	*% Delay Difference* d = 4
$k = 1$	baseline	-1.37%	-4.06%	-4.79%
$k = 2$	baseline	-1.61%	-4.91%	-5.69%
$k = 3$	baseline	-1.88%	-5.65%	-6.58%
$k = 4$	baseline	-2.09%	-6.35%	-7.37%

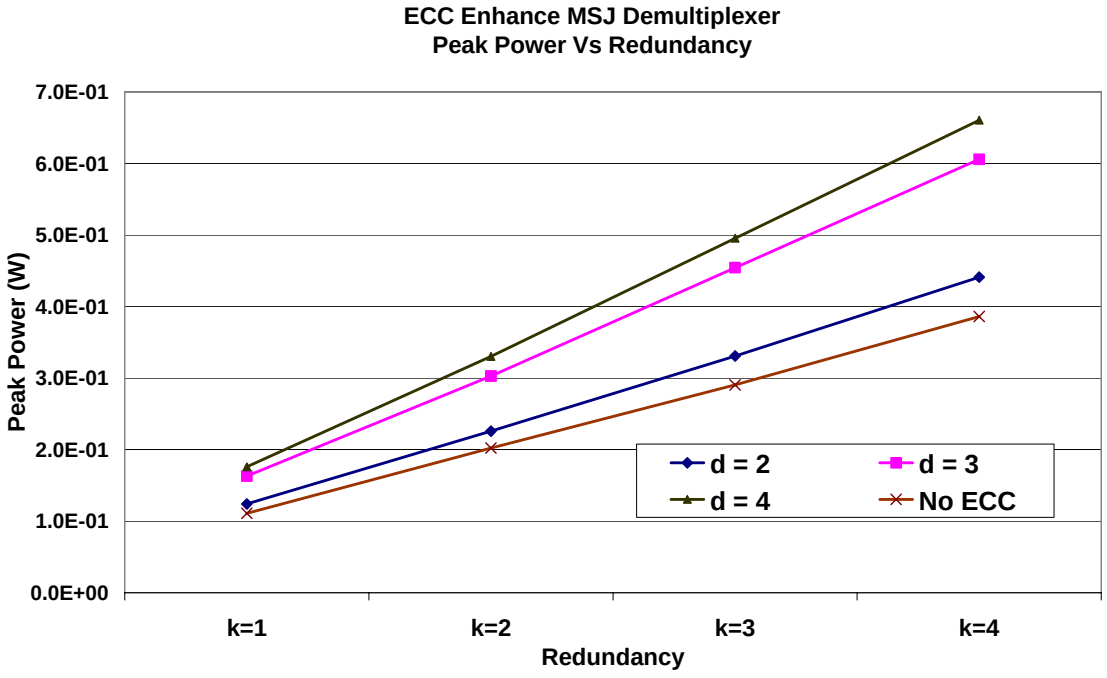

Figure 51. Relative increase in peak power dissipated with increasing redundancy (k) and increasing error correction code parity bits, in the crossbar demultiplexer.

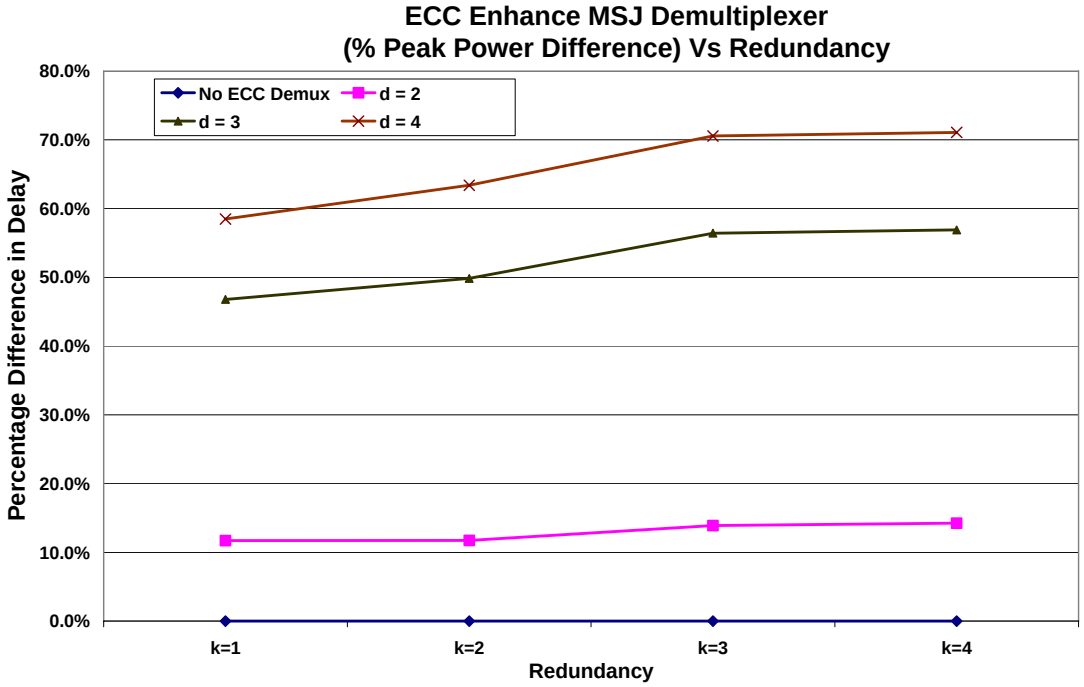

Figure 52. Relative increase in peak power dissipated with increasing redundancy (*k*) and increasing error correction code parity bits, in the crossbar demultiplexer.

Table 6
Results from the plot of Figure 52, showing the percentage difference in the increase in power dissipated with respect to the baseline *k = 1* case for a 128×128 demultiplexer implemented with both ECC and MSJ schemes.

Array Size →	**128×128**			
Redundancy	***% Power Difference* No ECC**	***% Power Difference* d = 2**	***% Power Difference* d = 3**	***% Power Difference* d = 4**
***k* = 1**	**Baseline**	**11.70%**	**46.79%**	**58.48%**
***k* = 2**	**Baseline**	**11.73%**	**49.86%**	**63.40%**
***k* = 3**	**Baseline**	**13.90%**	**56.42%**	**70.57%**
***k* = 4**	**Baseline**	**14.25%**	**56.91%**	**71.09%**

The results of the access time delay analysis simulation of Figure 49 can be interpreted as showing the cost with respect to access time delay, of increasing redundancy and hamming distance. In this case, delay improves with increasing hamming distances due to the addition condunction paths provided by the CMOS ECC circuitry. This fact is reflected in the decreasing values of Table 5. In contrast, the addition of peripheral CMOS circuitry as illustrated in Figure 37, increases the reliabilty induced power dissipation penalty. This can be observed in the plot of Figure 51. Also, the difference in the power dissipation penalties paid, due to the implemmention of ECC codes with **d** = 3 and **d** = 4, are marginaly small as a result of the single bit difference required for their implementation. For example, in the (k = 4) case presented in Table 6 and plotted in Figure 52, there is a 56.91% and 71.09% difference, as calculated from the baseline case (No (ECC-MSJ) implemented), when the hamming distance 3 and 4 codes, respectively, are implemented. As compared with the 14.25% difference in peak power dissipated when the (**d** = 2) ECC code is implemented. In the next section, the reliability of the crossbar nanomemory and demultiplexers are evaluated and results provided.

CHAPTER VII

CROSSBAR NANOMEMORY AND DEMULTIPLEXER RELIABILITY ANALYSIS RESULTS

7.1 Parameterized Circuit Reliability Analysis

In this section, the ability of the MSJ architecture to tolerate faults is determined by using the probabilistic model checker PRISM [133, 134]. Probabilistic model checking is an algorithmic methodology for determining the ability of a given probabilistic system to adhere to specified probabilistic parameters. For example, it can be used to answer the following question; given the requirement that a cross-point junction has a 0.001 probability of failure, what is the probability of reading a bit correctly from the memory? The system is generally regarded as a state transition system with probability different Markovian probabilistic models: Discrete-time Markov chains (DTMCs), Continuous-time Markov values attached to the transitions. PRISM supports the analysis of three chains (CTMCs) and Markov decision processes (MDPs). In particular, MDPs are used to model fault tolerant memory architectures. MDPs can be defined as a model of computation that is able to express the non-determinism of choosing the probability distributions from any given state. It is worth noting that each of these probability distributions indicates the outgoing transition from a specific state and sums up to one.

Modeling the flow of current through nanowires is a non-deterministic process,

and as such, MDPs is ideal for modeling redundant crossbar architectures. The attributes of crossbar based nanomemories that establish their non-deterministic and probabilistic behavior, can be surmised in two parts;

1. First, the geometry of a crossbar is such that, there may exist multiple paths by which current may flow from input to output. The distribution of defects along these varying paths must be accounted for via a non-deterministic model, which considers this varying probability distribution.
2. Secondly, the crossbar memory structure and the peripheral circuitry of the device, have different failure probabilities, which reinforce the probabilistic nature of the architecture.

The PRISM model used, assumes all junctions have an equal likelihood of failing, which is user defined. In addition, the failure distributions of the peripheral interconnects model may affect the system. In the analysis of the developed model, specific fault patterns are not considered. PRISM is used in the context of this work, to evaluate stuck-open faults by modeling them as the probability of failure to program the molecular junctions to the appropriate logic values.

The MSJ architecture is modeled as a generic MDP, which takes memory size ($n \times m$) and redundancy (k) as its input parameters. The model also assumes an independent, identical and unclustered fault rate for all the molecular junctions. The probability of correctness of the signal at each cell output is propagated until the desired output cross-point of the memory configuration is reached. In order to assert control over the number of state variables generated by the model, space is folded into time by

replacing the multi-junction set cells working in parallel, with cells working in sequence. In addition, the probability of obtaining a correct signal output from each cell is monitored, and the result propagated to the next cell sequentially. In this manner one is able to ensure that the state space does not become uncontrollably large. It is worth noting that the model that has been developed here, computes both the maximum and minimum probability of correctness, in view of the fact that the probability of having an input to output node connectivity in each cell depends on the number of intermediate defective junctions on a specific route.

7.1.1 Reliability Results

Results reflecting the probability of obtaining the correct output from a molecular switch junction given that it possesses a certain probability of failing are presented in the plots of Figures 53, 54 & 55. As the degree of redundancy is increased in these plots, a systematic improvement in the ability of the MSJ crossbar nanomemory to tolerate faults can be observed. The graphs also show a decline in fault-tolerance levels with increased array size. This is a result of the relative rise in the number of defects as a function of increased memory size. Figure 54, the $k = 2$ state shows a lower bound of approximately 65% for the memory range analyzed; this bound improves by 20% to approximately 85% when $k = 4$, as shown in Figure 55. This increase in the lower bound is an indication of the favorable scale-up potential of the MSJ scheme. Hence, it can be deduced that reliability in the crossbar nanomemory is improved by implementing the MSJ scheme.

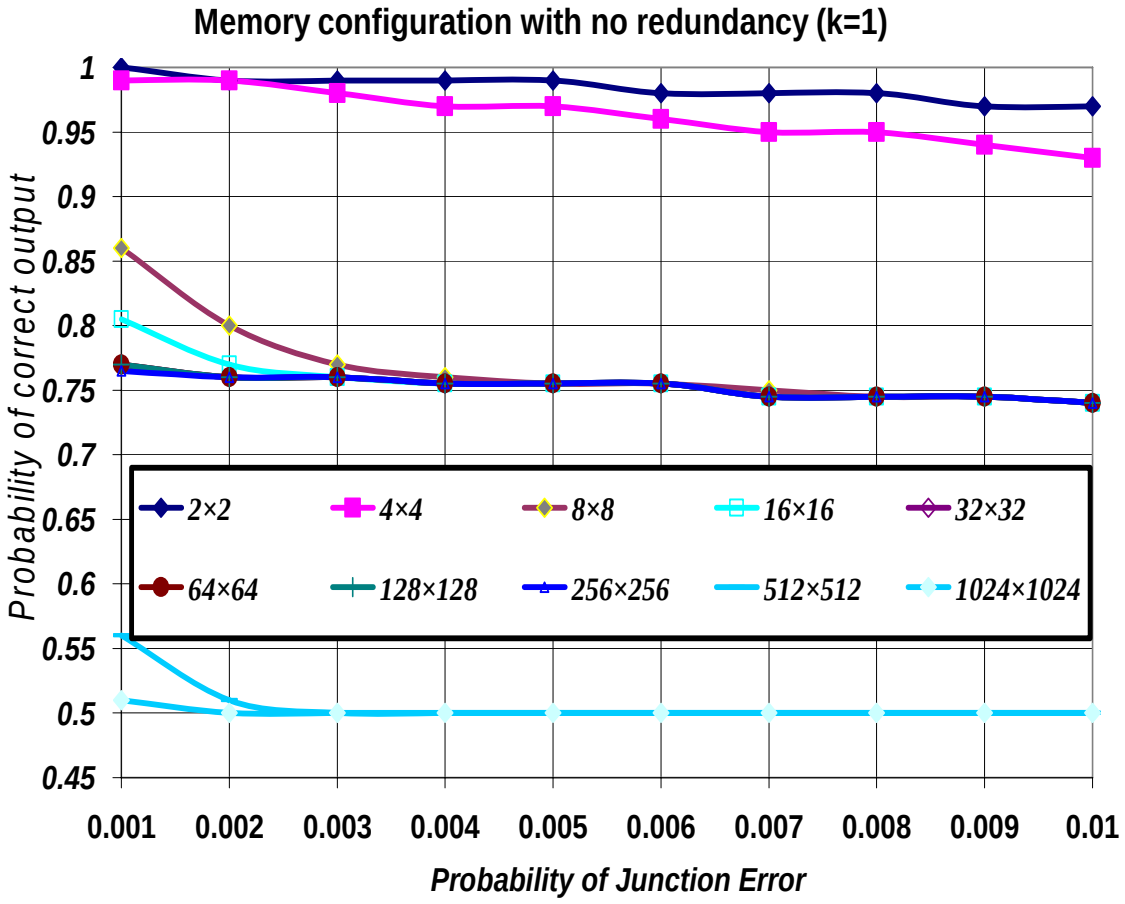

Figure 53. Probability of obtaining correct output from the molecular switch crossbar memory, when no redundancy is implemented in the device.

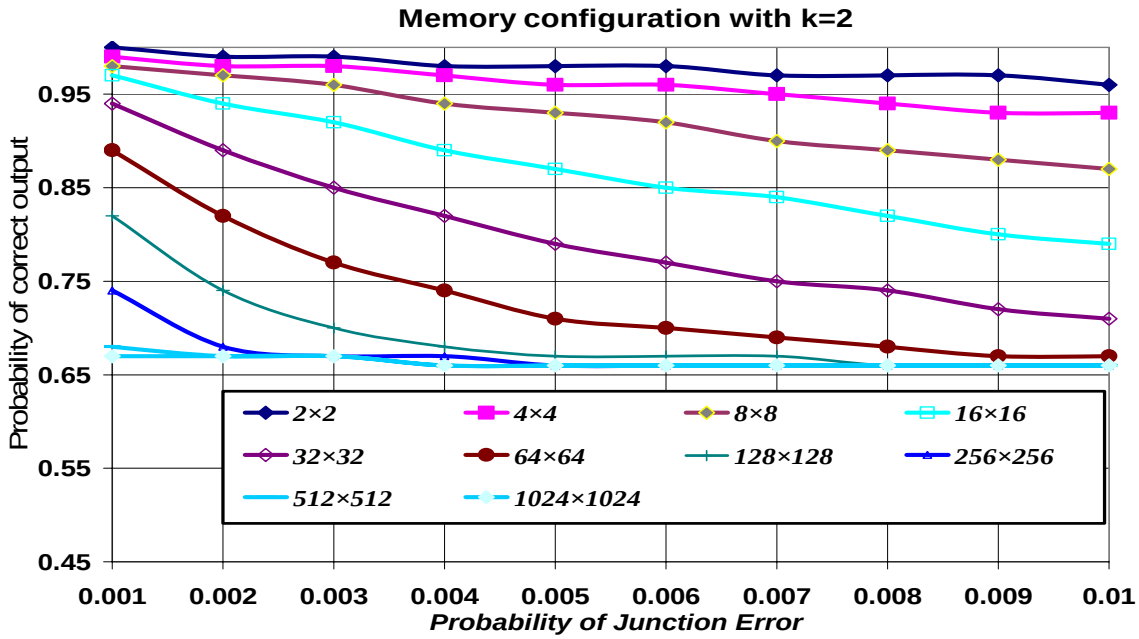

Figure 54. Probability of obtaining the correct output for $k = 2$.

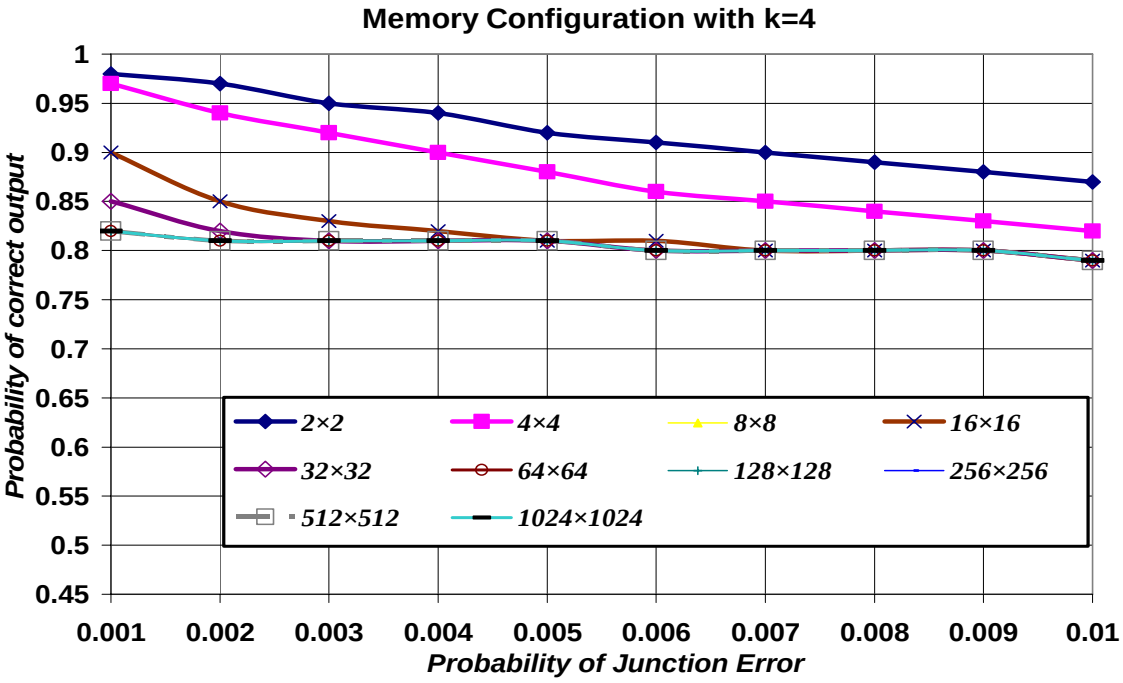

Figure 55. Probability of obtaining the correct output for $k = 4$.

7.2 Reliability of a Single-Wall Nanotube Demultiplexer: Introduction

In this section the probability of achieving fault tolerance in a crossbar molecular switch demultiplexer in which the nanowire interconnects are replaced by a bundle of SWNT is investigated; the bundle SWNT is illustrated in Figures 32 & 33. This is the follow-up reliability analysis to the delay and power performance analysis presented in bundled SWNT approach of Section 6.2.

7.2.1 Probability Analysis

The probability of yielding a desired number of non-defective molecular switch junctions from a cumulative yield of molecular switches available for a desired SWNT address-line is determined using a binomial distribution. Given an address consisting of i bits with $(k-1)$ redundant bits and x ECC bits, the Binomial (i, p) distribution is computed as the probability of yielding i useable bits from N total bits, given a specified fault or defect probability p. The probability of yielding i non-defective molecular switch junction from N total junctions is given by the following binomial distribution:

$$P_{yield}(N, i) = \left(\binom{N}{i} P^{i} (1 - P)^{N - i} \right) \tag{7-1}$$

where,

$$N = i + (k - 1) + x \tag{7-2}$$

The binomial coefficient $\binom{N}{i}$ is the number of ways of selecting i good molecular switch junctions from a total of N junctions. For each of these cases, there is a yield probability of $P^i((1-P)^{N-i})$. Total yield is thus given by the following cumulative distribution equation:

$$P_{MofN} = \sum_{M \le i \le N} \left(\binom{N}{i} P^i (1-P)^{N-i} \right) \tag{7-3}$$

The probability analysis assumes an idealized model where defects are uniformly spread with uniform probability and independence. In the probability analysis graph provided in Figure 61, $k = 1$ is considered the benchmark condition; i.e. only ECC is implemented in the demultiplexer. It can thus be inferred from the plots of Figure 54 that for lower defect probabilities, the molecular switch junction enhanced ECC demultiplexer (i.e. the MSJ-ECC demultiplexer) shows a higher degree of fault-tolerance than the (ECC- only) demultiplexer. However, Figure 56 also shows that at higher defect rates, indicated by the inflection point in the graph, the (ECC-only) demultiplexer shows a higher degree of fault-tolerance. This is a result of the defect probability weight (binomial coefficient) dominating the process. The defect probability is spread proportionally across each junction; higher redundancies will make the likelihood of making the correct address selection to be dominated by the increase in defect prone junctions. The point where the benchmark case $k = 1$, overtakes the higher redundancies occurs at $p \approx 65\%$.

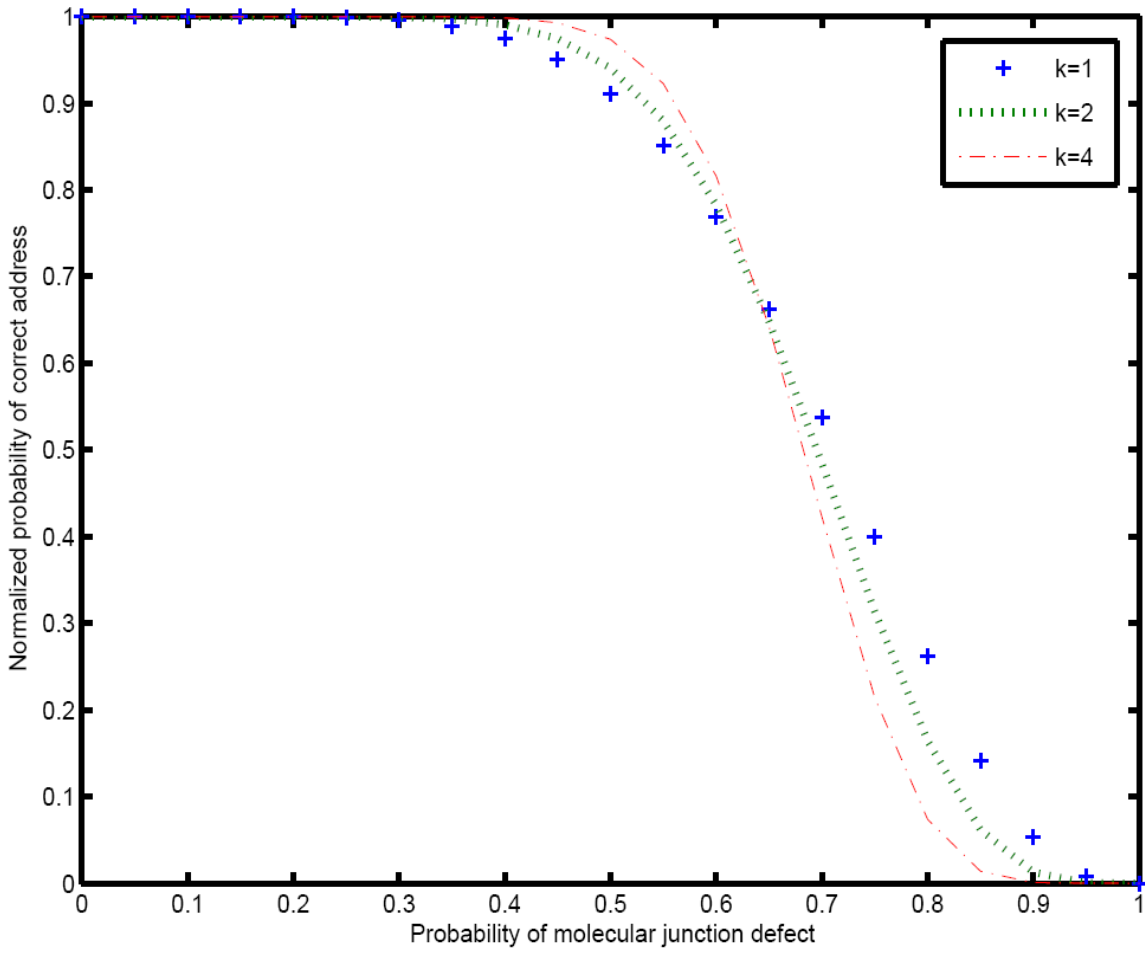

Figure 56. Probability analysis graph showing the normalized probability of selecting the correct input address given range of defects probability. $k = 1$ is the benchmark case indicating an ECC only demultiplexer.

7.3 Reliability: Scaled Analysis Results

In this section the probability of achieving fault tolerance in a crossbar molecular switch nanomemory and demultiplexer is studied and results reported. This section defers from section I in the PRISM analysis, which was conducted based on a scaled model, as well as the combined crossbar nanomemory and demultiplexer analysis. Two defect conditions, stuck-open and broken wire defects, are considered in this work. This section is also a corollary reliability analysis to the delay and power performance analysis presented in Section 6.3.

The MSJ architecture in this scaled analysis is also modeled as a generic MDP. When stuck-open faults occur, they prevent the flow of current through the molecular switch junctions. Hence, the MDP function of PRISM is used to model stuck-open faults as the likelihood of the molecular switch junction transitioning from one state to another; which is akin to the likelihood of the molecular switch junction to allow or prevent the flow of current. This model was validated using a base case binary model.

As in the parameterized analysis section, the model created here, also assumes an independent, identical and unclustered fault rate for all the molecular junctions. Both the maximum and minimum probability of correctness is also computed.

Validation Analysis

(50% Junction Error Utilized)

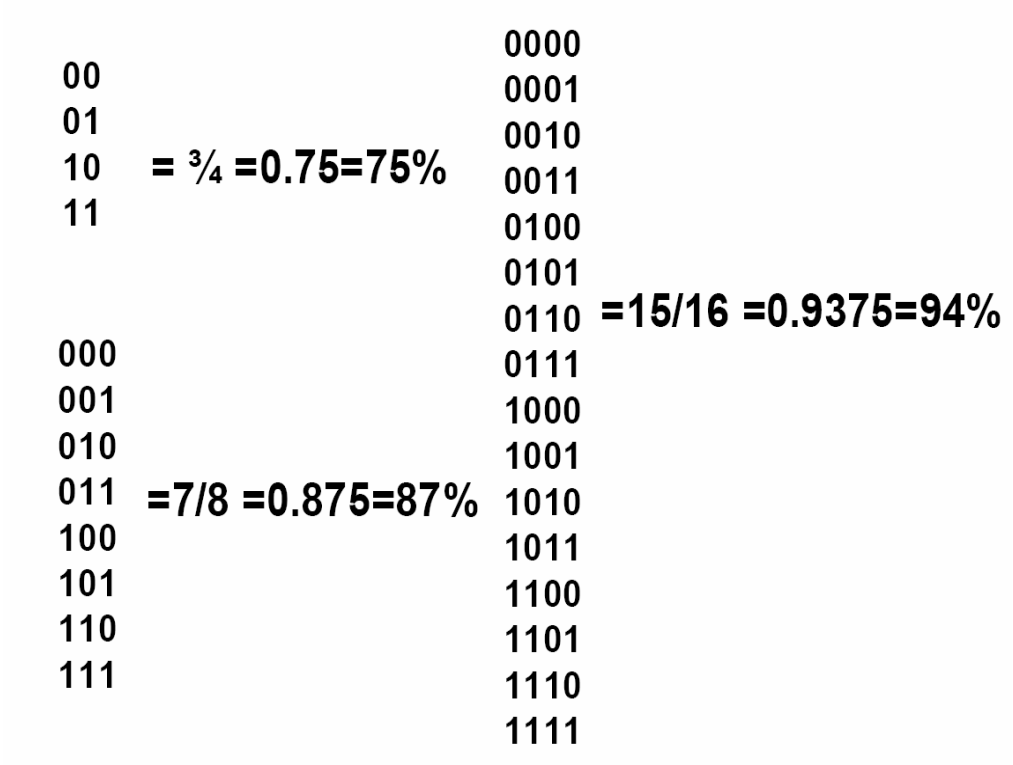

Figure 57. Probability analysis using binary numbers. All possible configurations of a one, two and three bit sequence were used. At least one of the binary bits in a configuration of a sequence must be "1" for it to be considered a success. Each bit is comuted as having a 50% chance of being a "1" or "0".

To validate the PRISM models, a binary probability analysis was conducted. In an MSJ crossbar nanomemory grid, at least one of the molecular switches must be working for a bit to be stored. Using fundamental probabilistic methodologies, a

straightforward analysis can be made as observed in Figure 57; a model using all possible configurations of a one, two and three bit sequence was used. At least one of the binary bits in a configuration of a sequence must be "1" for it to be considered a success. This model, in which each of the bits were simulated in PRISM as having a 50% chance of being a "1" or "0", was then applied to the PRISM simulator to validate the results of the quantitative analysis with the simulation results—which have been plotted in Figure 58. As can be observed, the results of Figure 58 validate the results of our simulator in Figure 58.

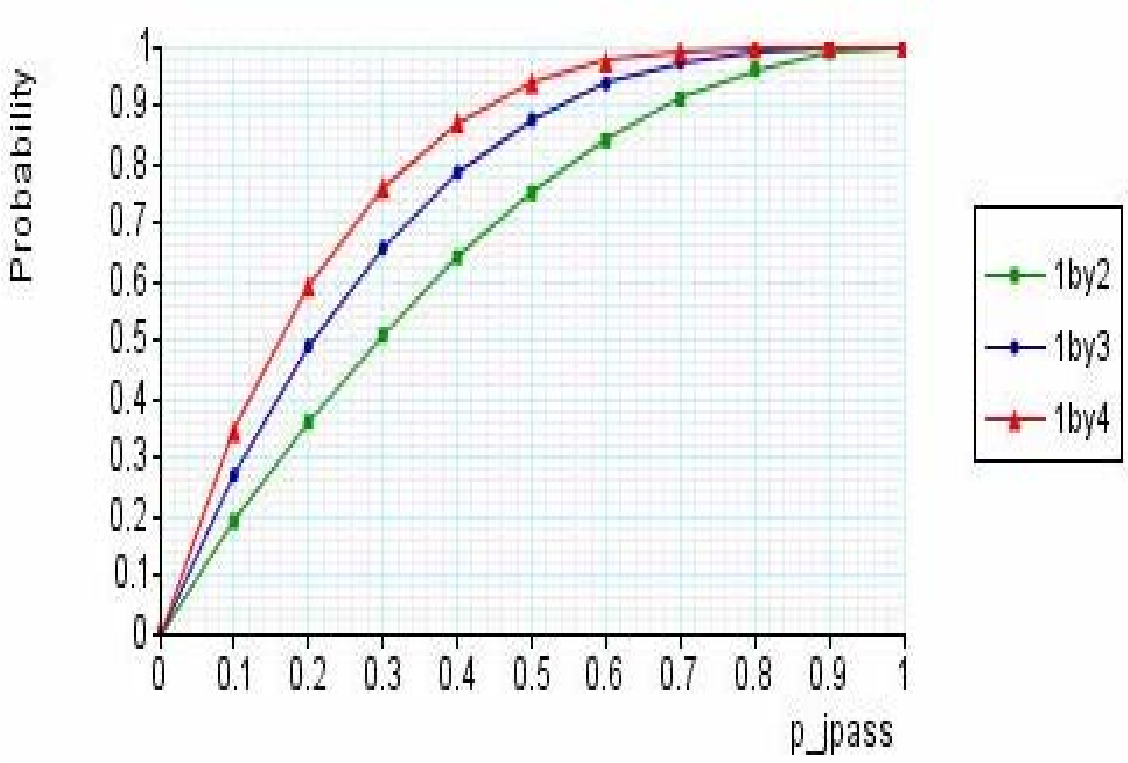

Figure 58. Validation results using PRISM. 1by2, 1by3 and 1by4, represent the 2, 3 and 4 bits sequence shown in Figure 57.

7.3.1 Crossbar Nanomemory Reliability Analysis Results

All simulations results indicate substantial improvements in the defect tolerance capabilities of the crossbar nanomemory and demultiplexer with the implementation of the MSJ scheme. As can be inferred from the results, the MSJ scheme has a greater impact on the reliability of smaller nanomemory arrays. This is mainly due to the fact that as the molecular switch junctions are replicated by implementing the MSJ scheme, more defect prone junctions are added to the simulated nanoelectronic devices—as was also the case in the reliability analysis of the section 7.2 of this chapter.

In the case of the crossbar nanomemory devices, our simulations show a high level of unreliability in the case where no MSJ is applied to the crossbar nanomemory, Figure 59 quantifies this point. In the plots of Figures 60 through 62, it can be observed that the reliability of the crossbar nanomemory improves substantially. The largest impact of the MSJ scheme can be observed in the plot of Figure 62, where nanomemory reliabilities greater than 92% are achieved for molecular switch junction defect rates greater than 50%, and redundancy ($k \geq 4$) is implemented for array sizes less than or equal to 64×64; reliability rates of approximately 74% is observed for the 128×128 array size and similar defect rates. Simulation results not provided here shows that reliability increases to greater than 99% when ($k \geq 5$) for similar array size ranges.

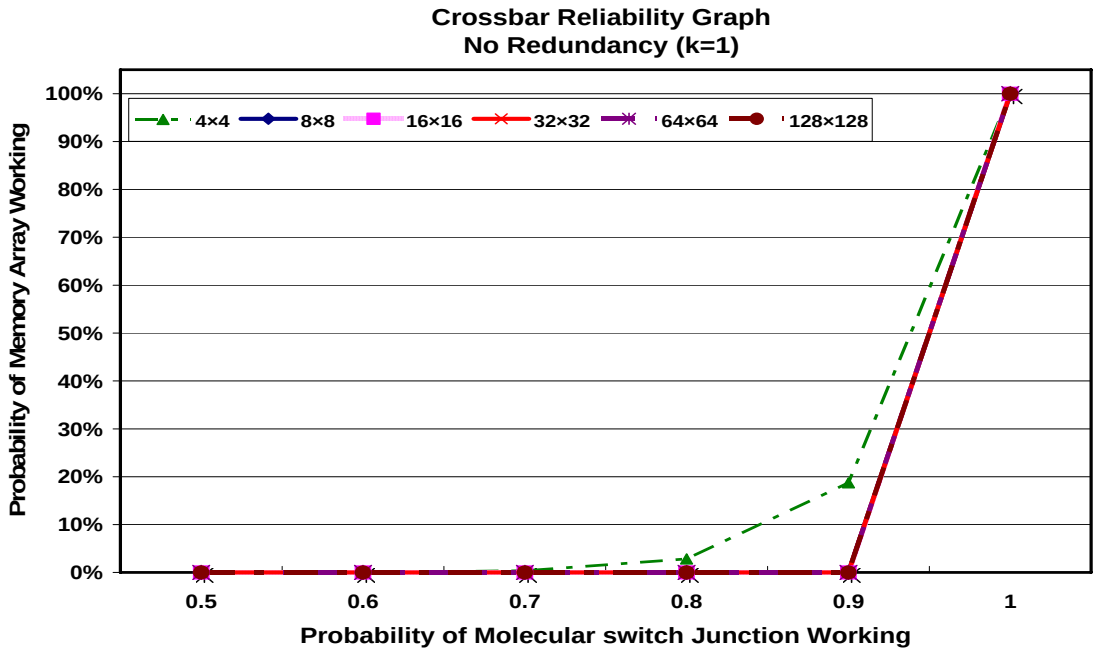

Figure 59. Simulation results showing the the reliability of a crossbar nanomemory with no implemented fault-tolerance.

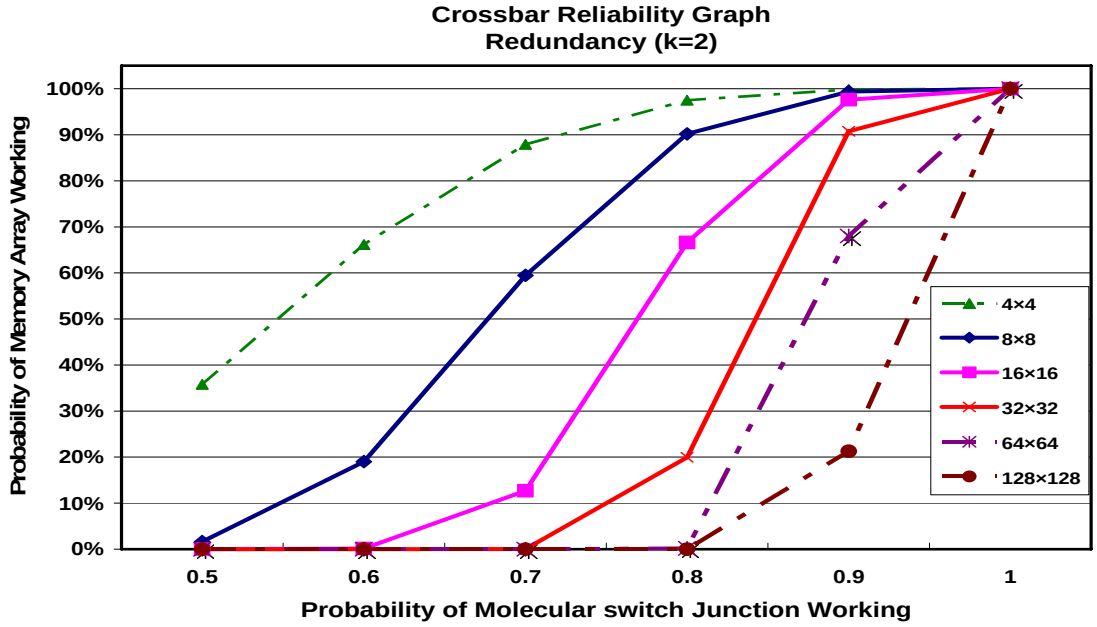

Figure 60. Simulation results showing the the clear improvement in the reliability of the nanomemory at redundancy $k = 2$, especially with regards to the smaller array sizes.

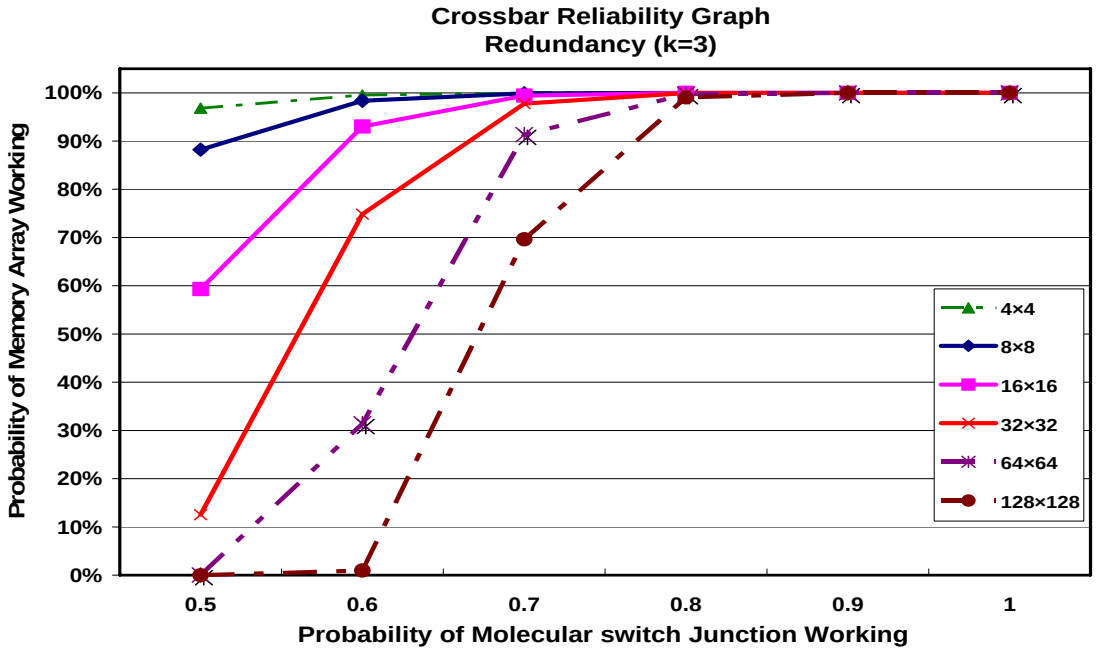

Figure 61. Simulation results showing the improved reliability over the $k = 2$ case when the nanomemory redundancy is increased to $k = 3$. Also increased reliability is noticable for the larger array sizes.

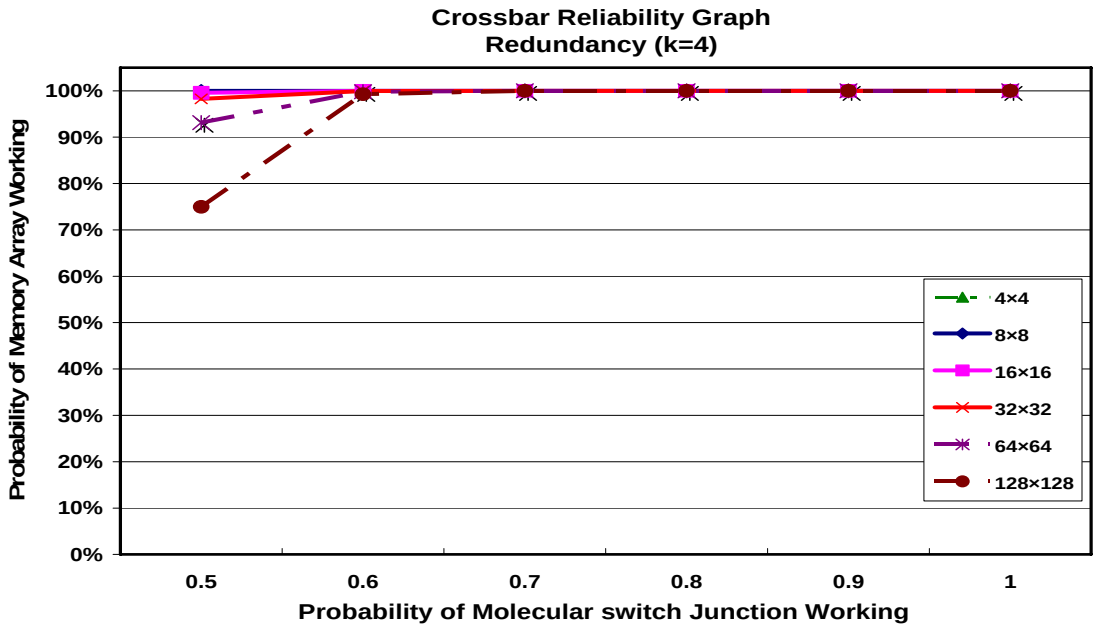

Figure 62. Graph shows the clear improvement in the reliability of the nanomemory at redundancy $k = 4$.

The following plots (Figures 63 to 67), are presented here to amplify the impact of implementing the MSJ scheme on small nanomemory arrays.

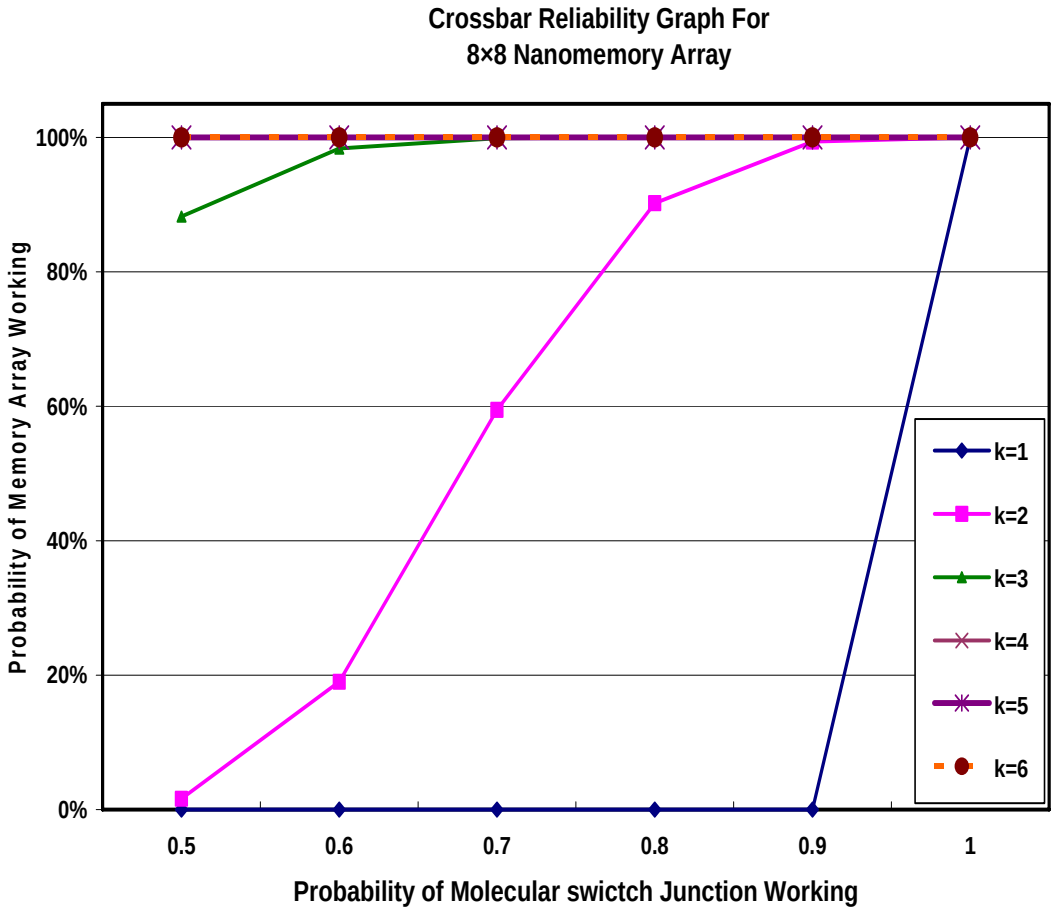

Figure 63. Simulation results showing the trend in reliability improvment with increasing *k* for a 8×8 nanomemory array.

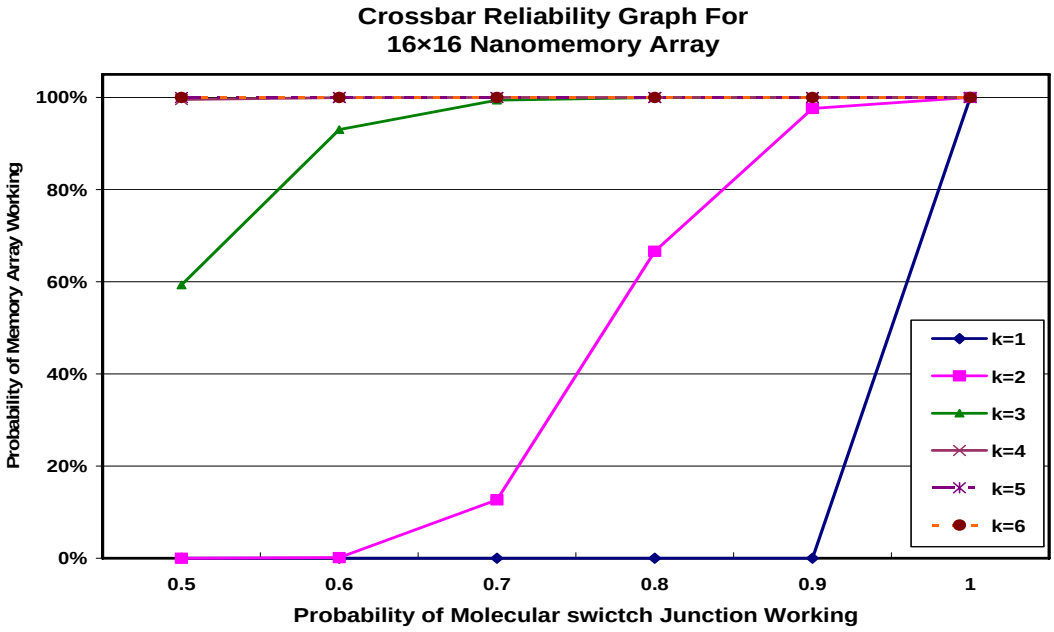

Figure 64. Simulation results showing the trend in reliability improvment with increasing *k* for a 16×16 nanomemory array.

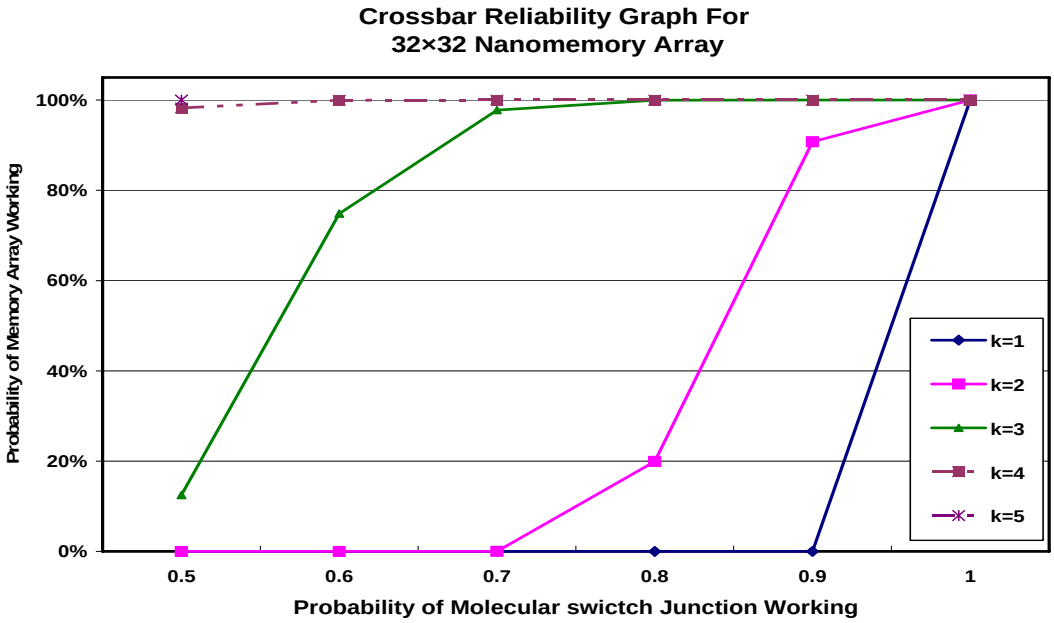

Figure 65. Simulation results showing the trend in reliability improvment with increasing *k* for a 32×32 nanomemory array.

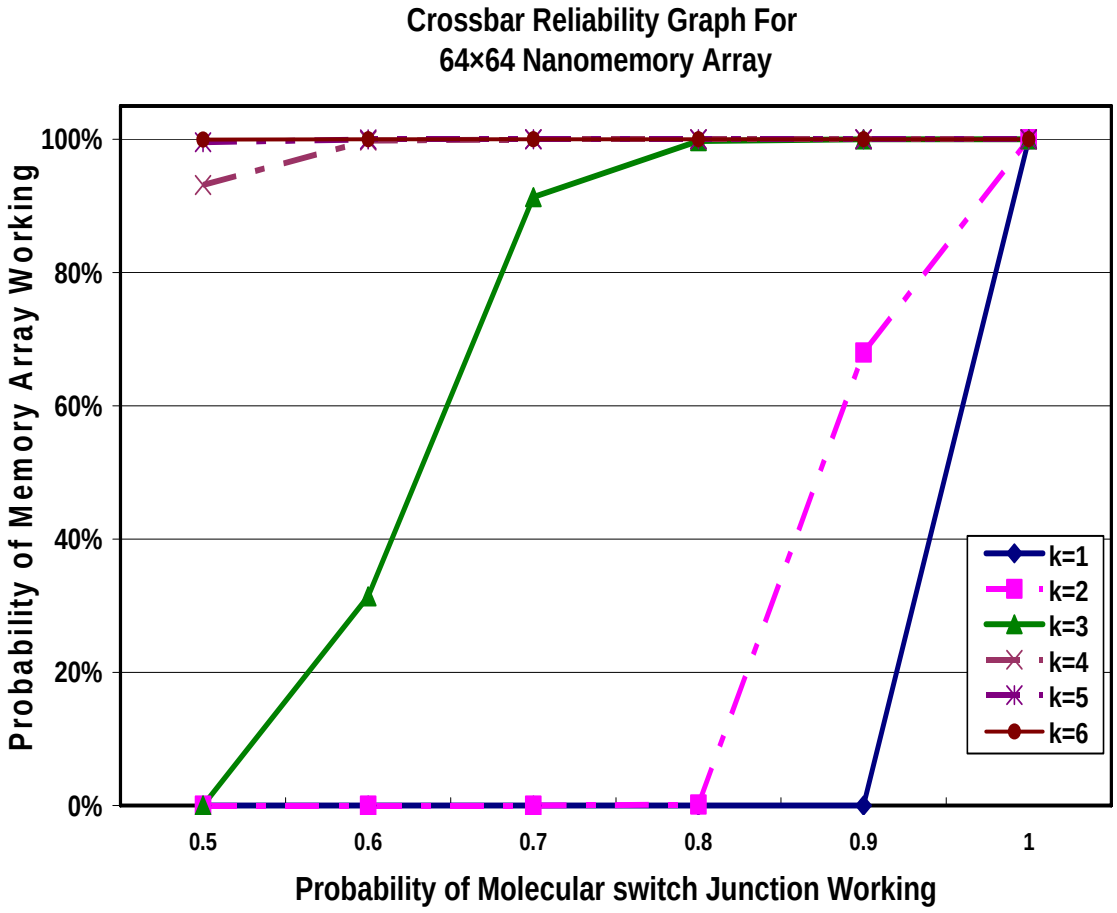

Figure 66. Simulation results showing the trend in reliability improvment with increasing *k* for a 64×64 nanomemory array.

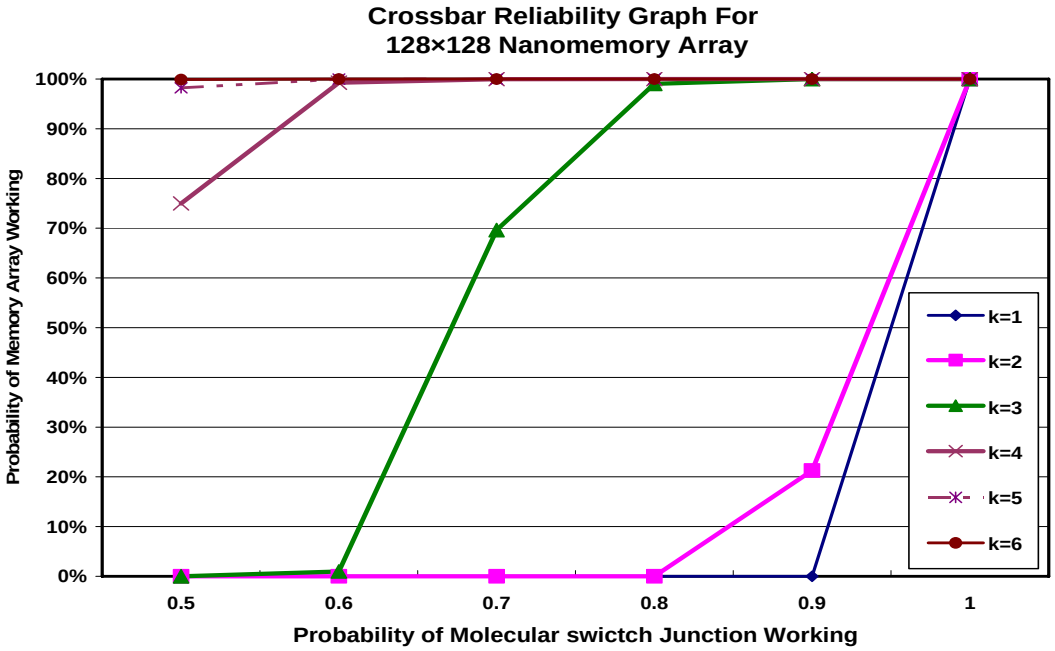

Figure 67. Simulation results showing the trend in reliability improvment with increasing *k* for a 128×128 nanomemory array.

7.4 Demultiplexer Nanowire Stuck-Open Faults Reliability Analysis Results

In Figure 68, the model implemented in the stuck-open fault reliability analysis is illustrated. When junction faults occur, the address nanowire rows and columns are still functional, however, the connection switch between the row and column address nanowires behave like an open circuit. As observed in Figure 68, a redundant MSJ is implemented; hence the 4×4 array effectively represents a single molecular switch junction. For the MSJ to be considered effective there has to be a row and column nanowire output path for current. Hence, in the situation where all the molecular switch junctions in the first row and second column address-lines are defective, the information

stored is not lost because of the nine remaining functional molecular switches representing the replicated data.

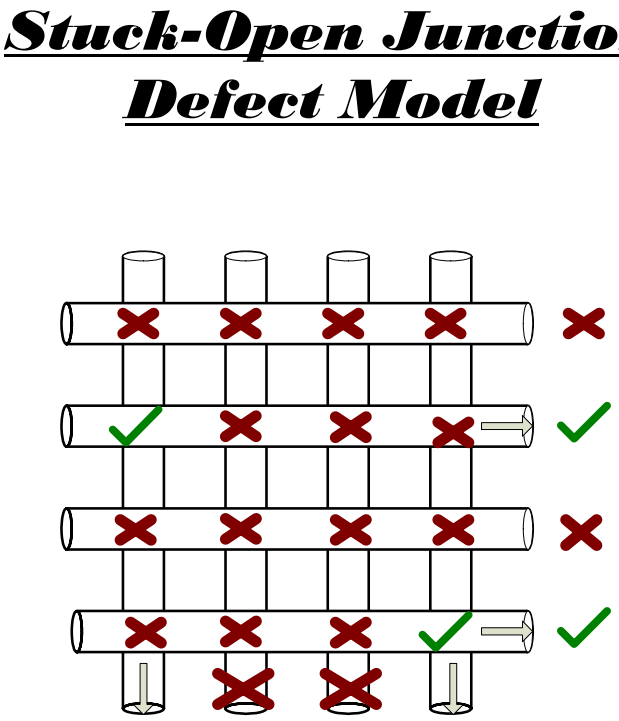

Figure 68. Stuck-open fault defect model illustrating conditions under which current woud be allow to pass to the row and column outputs in the presence of defective molecular switch junction. The **X** mark, symbolizes a defective molecular switches as well as a failed output, and the checks (√) and arrows symbolize the successful transsmision of current to the row and column outputs. This model represents a $k = 4$ MSJ implementation.

In the case of the demultiplexer, a greater degree of redundancy is required to obtain a reliability rate equivalent to that of the crossbar nanomemory. When the MSJ scheme is implemented in the demultiplexer, molecular switches are replicated at a value equal to k, because only the nanowires are made redundant. In the crossbar nanomemory

the replicated switches are equal to k^2, which is a function of the crossbar nanomemory row and column nanowires redundancy, illustrations of the MSJ implementation in the crossbar nanomemory and demultiplexer provided in Figures 16 and 22 demonstrates the fact. Due to large simulation runtime and resource requirements, only array sizes of up to 128×128 were simulated. These array sizes were sufficient for the goals of this work, in that the performance benefits of the MSJ schemes and its impact are discernibly observable in the provided results.

The ensuing result plot show the reliability of the demultiplexer when no redundancy is implemented, Figure 69. Just by adding two extra nanowires ($k = 3$) to each demultiplexer address-line, as indicated in Figure 70, it can be observed that for example, when the molecular switch junction failure rate is 10%, a greater than 70% device reliability can be achieved in array sizes less than or equal to 32×32—as opposed to the less than 20% reliability observed for the same array size range in the no redundancy plot of Figure 69. In the case of redundancy $k > 5$, we can increase the reliability of a demultiplexer to approximately 100% when the probability that each molecular switch junction could fail is 10%; these results can be observed in Figures 72 and 73.

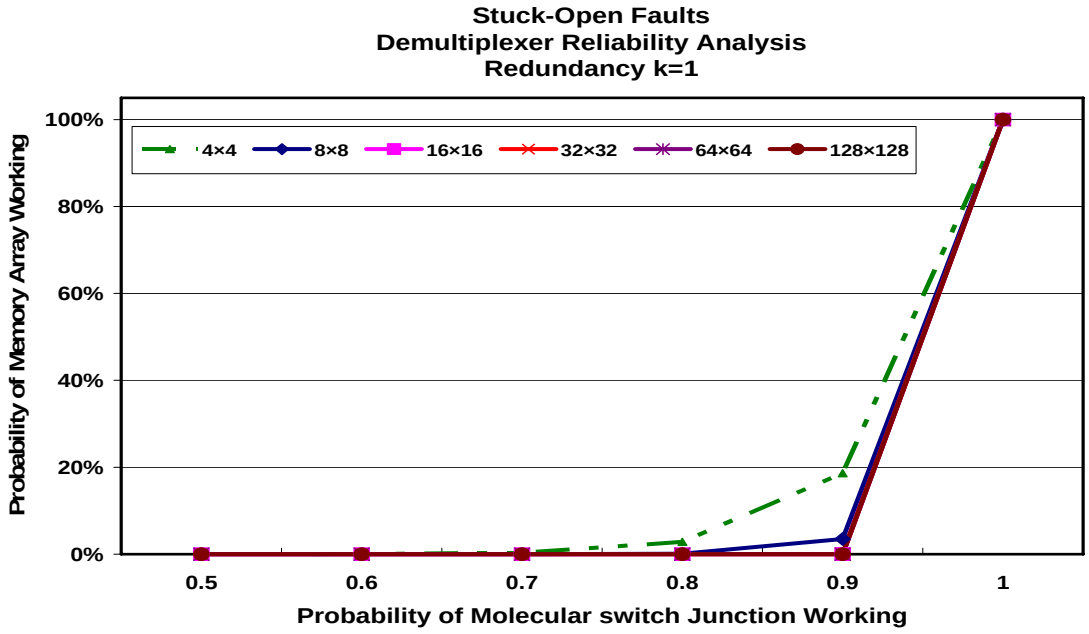

Figure 69. Demultiplexer reliability under stuck-open fault conditions when no redundancy is implemented. The high degree of device unreliabiloity is clearly observed.

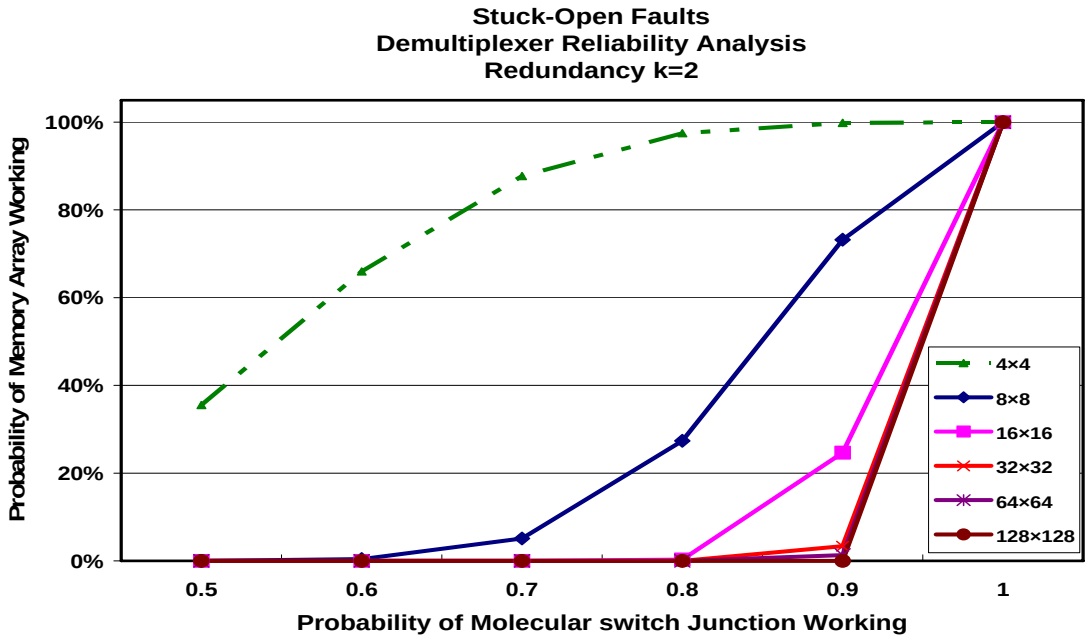

Figure 70. Demultiplexer reliability under stuck-open fault conditions, when redundancy $k = 2$ is implemented and array size is varied.

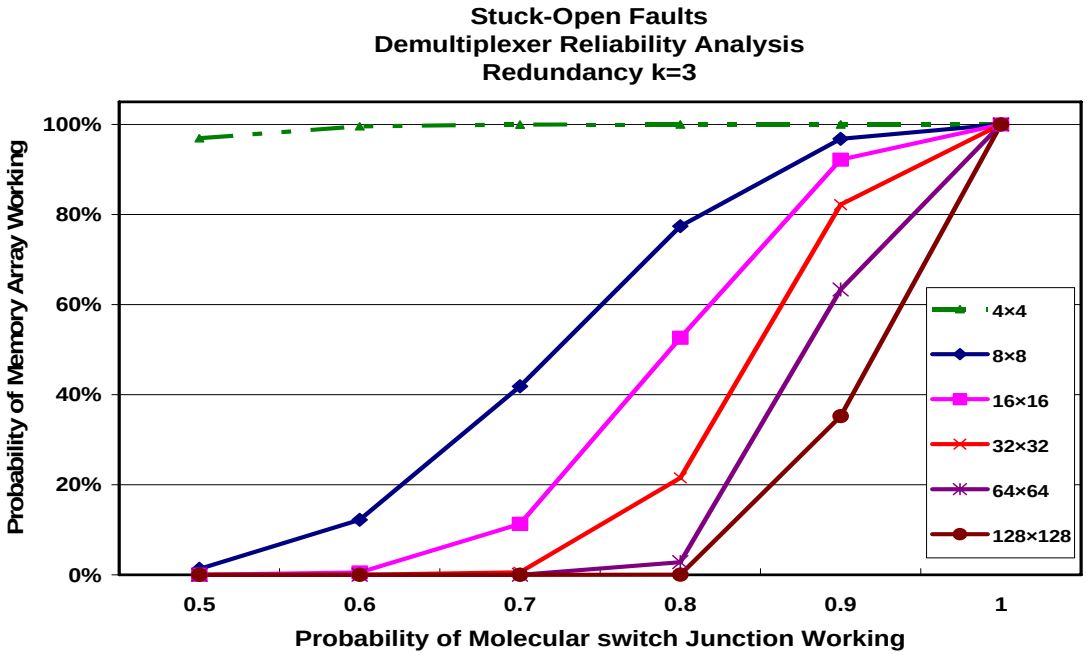

Figure 71. Demultiplexer reliability under stuck-open fault conditions, when redundancy $k = 3$ is implemented and array size is varied.

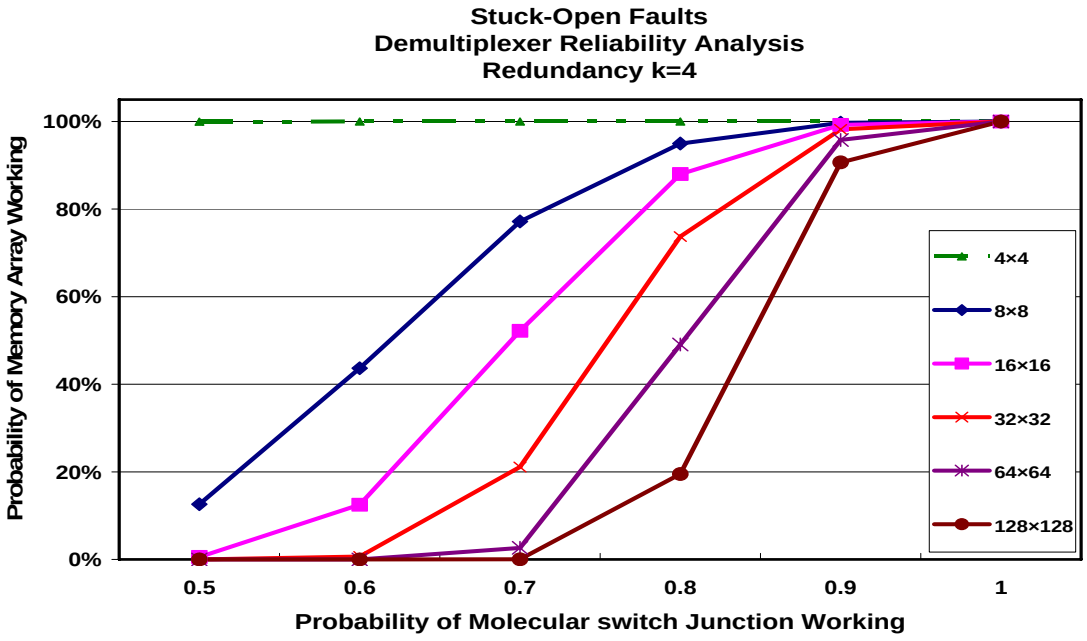

Figure 72. Demultiplexer reliability under stuck-open fault conditions, when redundancy $k = 4$ is implemented and array size is varied.

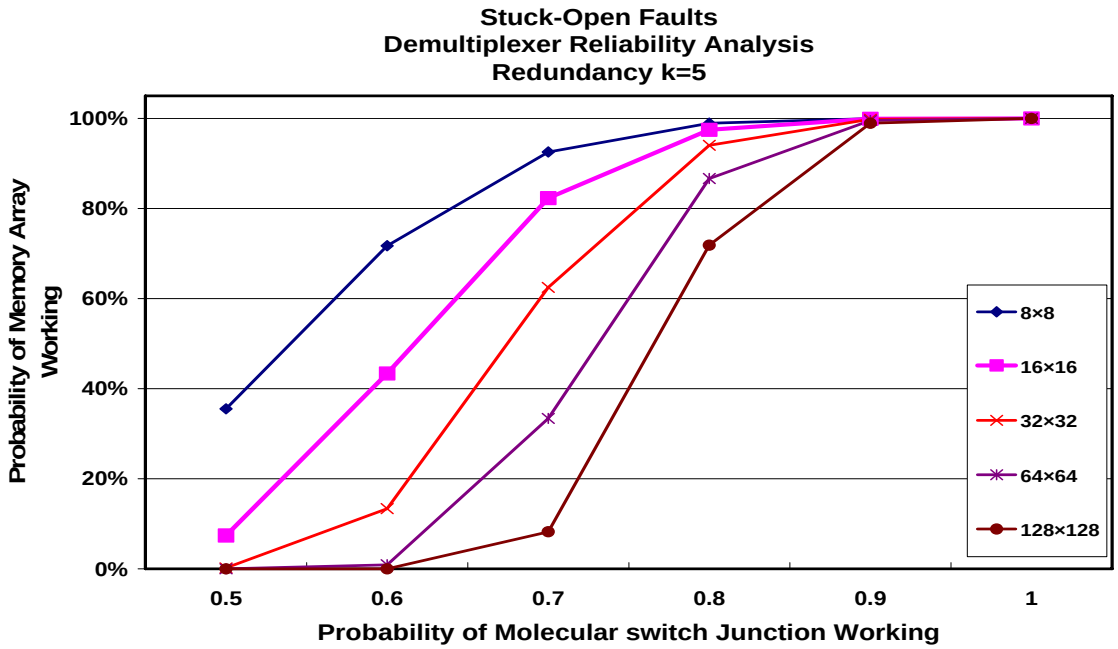

Figure 73. Demultiplexer reliability under stuck-open fault conditions, when redundancy $k = 5$ is implemented and array size is varied.

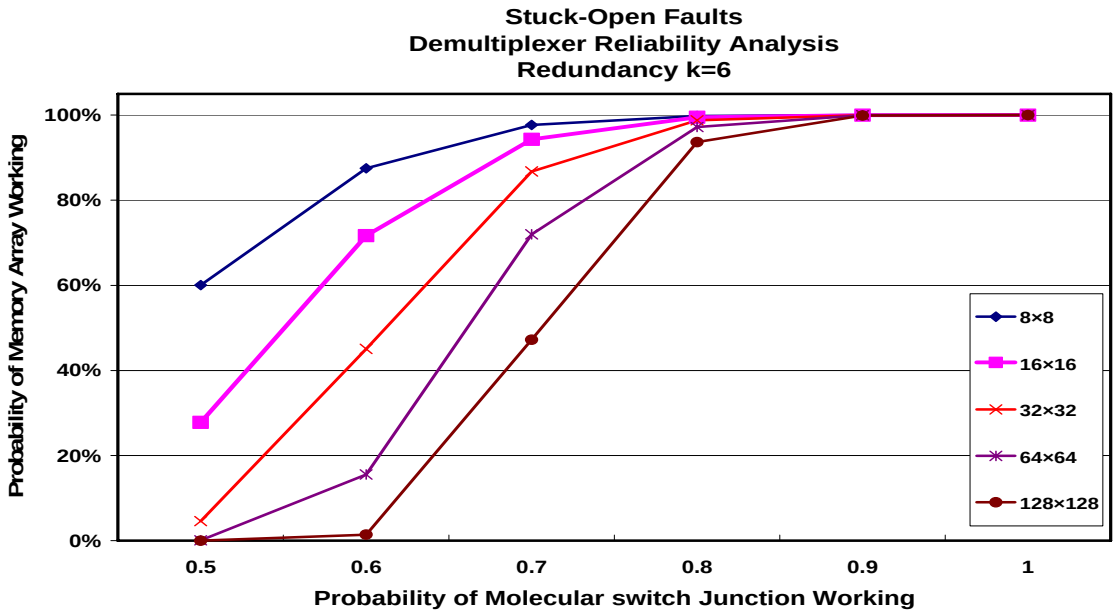

Figure 74. Demultiplexer reliability under stuck-open fault conditions, when redundancy $k = 6$ is implemented and array size is varied.

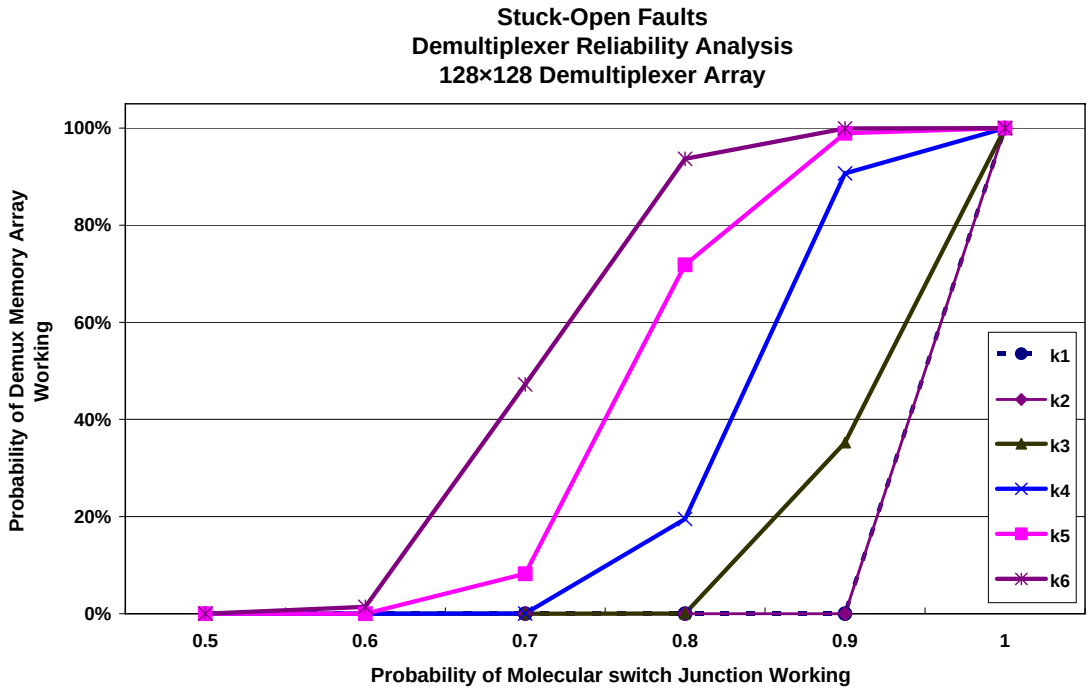

Figure 75. Demultiplexer reliability in a 128×128 demultiplexer array, measured as a function of the probability of the molecular switch junction working and increasing redundancy. Results demonstrates a clear improvement in demultiplexer reliability due to the implementation of the MSJ scheme.

7.5 Demultiplexer Broken Nanowire Defects Reliability Analysis Results

Nanowire defects are more difficult to tolerate. When nanowires break due to fabrication abnormalities, they interrupt the flow of current from the address microwires to the output electrodes of the demultiplexer—stuck-open faults merely render molecular switch junctions inoperable. The plots of Figures 77 to 80 demonstrate these points. When $k = 1$ in Figure 76, the reliability of the demultiplexer is slightly less than 10% for when the probability of there being a break in the nanowires is 10%. Implementing a redundancy of $k = 2$ fairs no better, except in the 16×16 array where the where reliability improves to 90% under the same conditions. However at $k = 4$ we get a 100% reliability

in the demultiplexer across the board.

The illustration provided in Figure 76, can be used to better understand the broken nanowire defect implementation utilized in the analysis of this section. Figure 76 represents the implementation of the MSJ scheme with redundancy $k = 4$; hence the 4×4 array effectively represents a single molecular switch junction. In Figure 76, the bottom two row nanowires will transmit because they have no breaks or defects and also because the second and fourth (read from left to right) column nanowires also possess no breaks. This was also the case for the column nanowire outputs as observed by the green symbols (*checks* and *arrows*). Break which occurs in the row or column nanowires effectively shuts that nanowire down, rendering all the molecular switches in that row or column useless. In the defect configuration of Figure 76, the MSJ implementation works quite well, allowing for the flow of current in both the row and column dimensions; an essential requirement, because for the MSJ to be considered effective there has to be a row and column nanowire output path for current. These were the model conditions implemented and simulated in the PRISM.

Broken Nanowire Defect Model

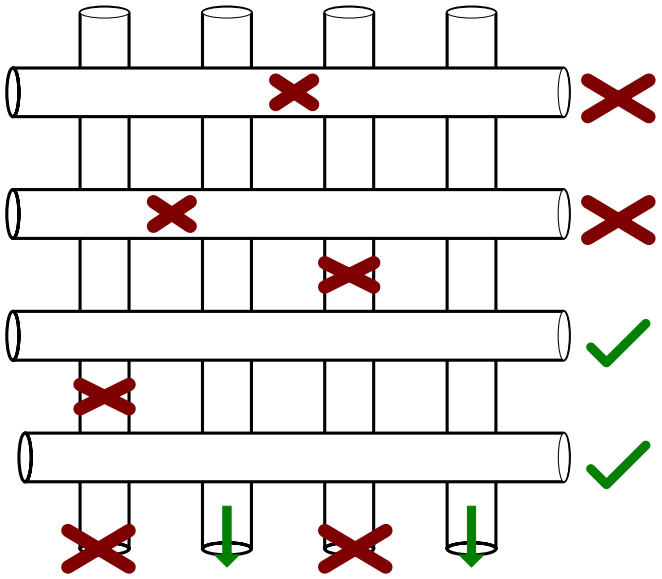

Figure 76. Broken nanowire defect model illustrating condition under which current woud be allow to pass to the row and column outputs in the presence of defects. The **X** mark, symbolizes a break or defect as well as a failed output, and the green check (√) and arrow symbolize successful current output transmission. This model represents a $k = 4$ MSJ implementation.

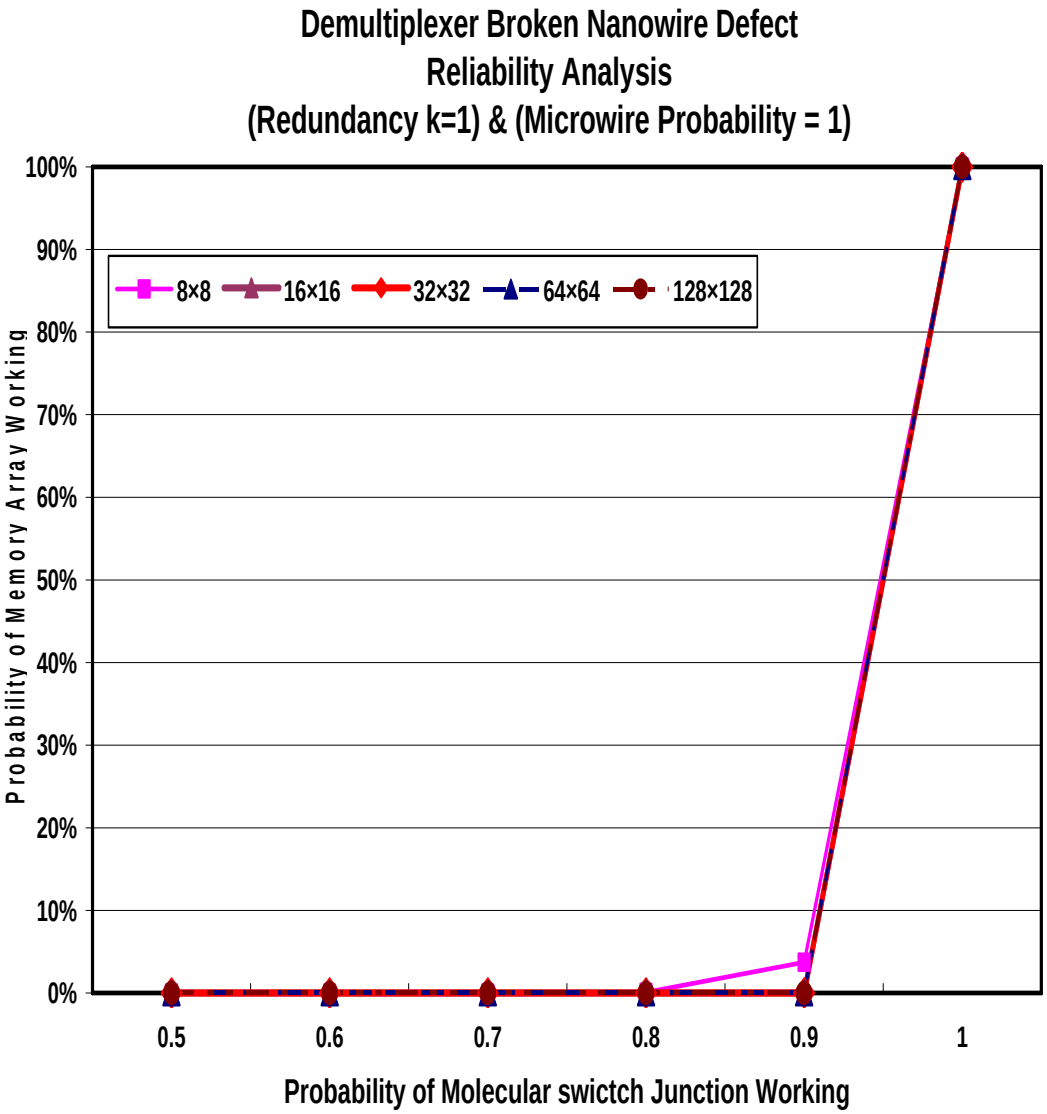

Figure 77. Demultiplexer reliability under broken nanowire defect conditions, when no redundancy $k = 1$ is implemented, and as a function of the probability of the molecular switch junctions working, and the array sizes are varied in increasing order.

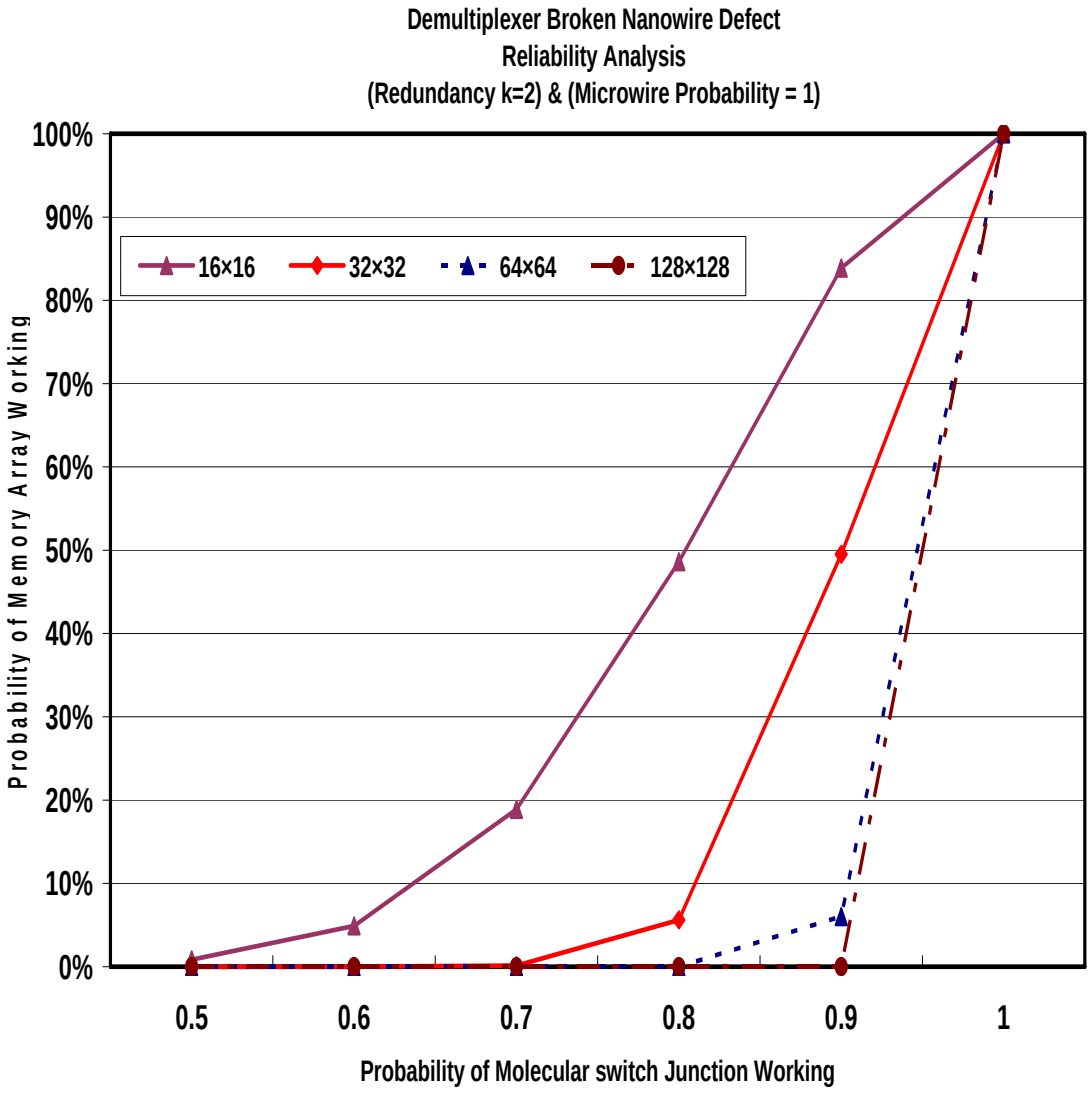

Figure 78. Demultiplexer reliability under broken nanowire defect conditions, when redundancy $k = 2$, under the same conditions as Figure 77. Results show better reliability improvements in the smaller arrays. In particular, the reliability trend lines improve sharply with decreasing array sizes.

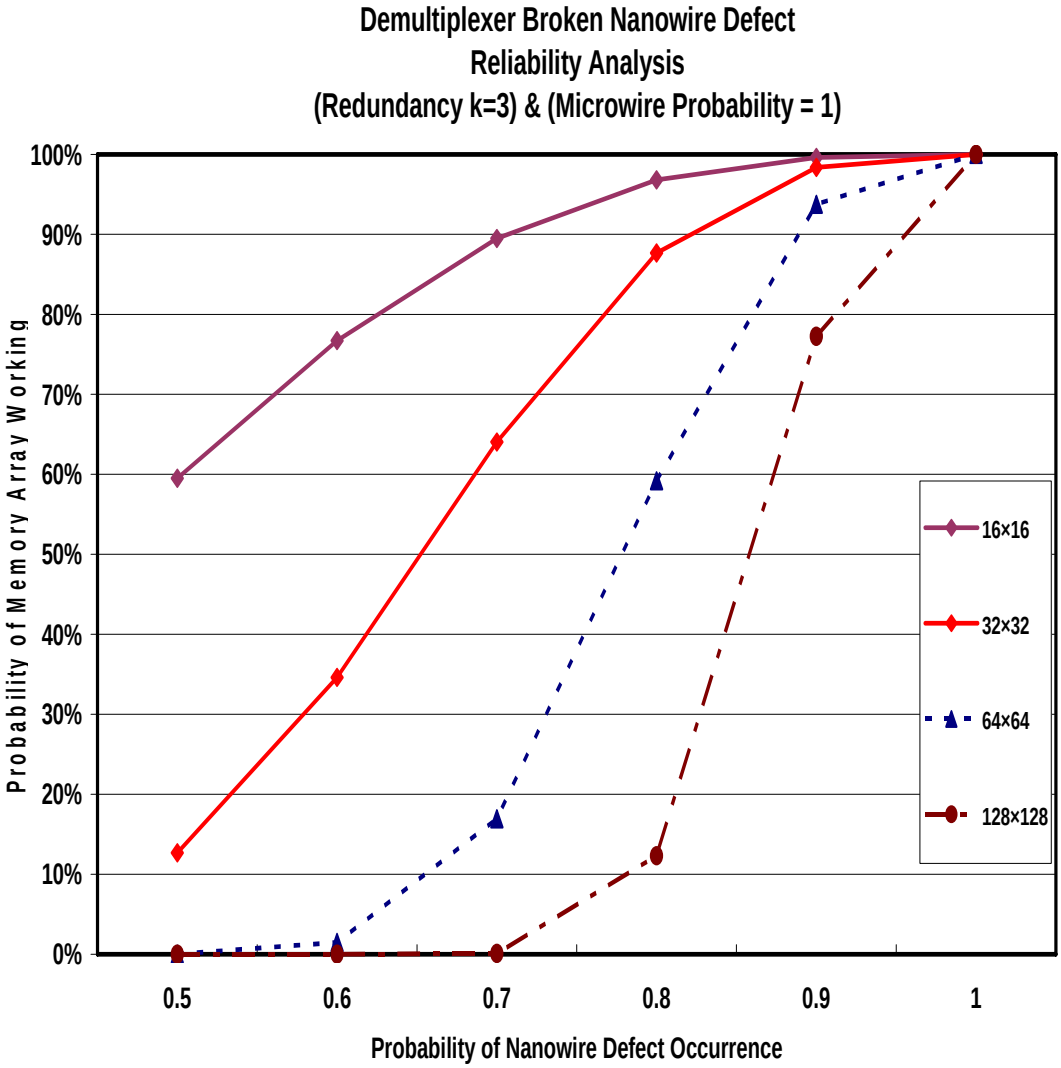

Figure 79. Demultiplexer reliability under broken nanowire defect conditions, when redundancy $k = 2$, under the same conditions as Figure 77. Results show better reliability improvements in the smaller arrays. In particular, the reliability trend lines improves with decreasing array sizes as was the case in the $k = 2$ implementation of Figure 77. In this case a sharper rise in the trend lines can be observed as indicated by the approximately 78% improvement when the probability of the molecular switch junction working is 0.9 (or 90%) for the 128×128 array. Contrast this with the 0% reliability achieved by implementing $k = 2$ under the same conditions and parameters as observed in Figure 78.

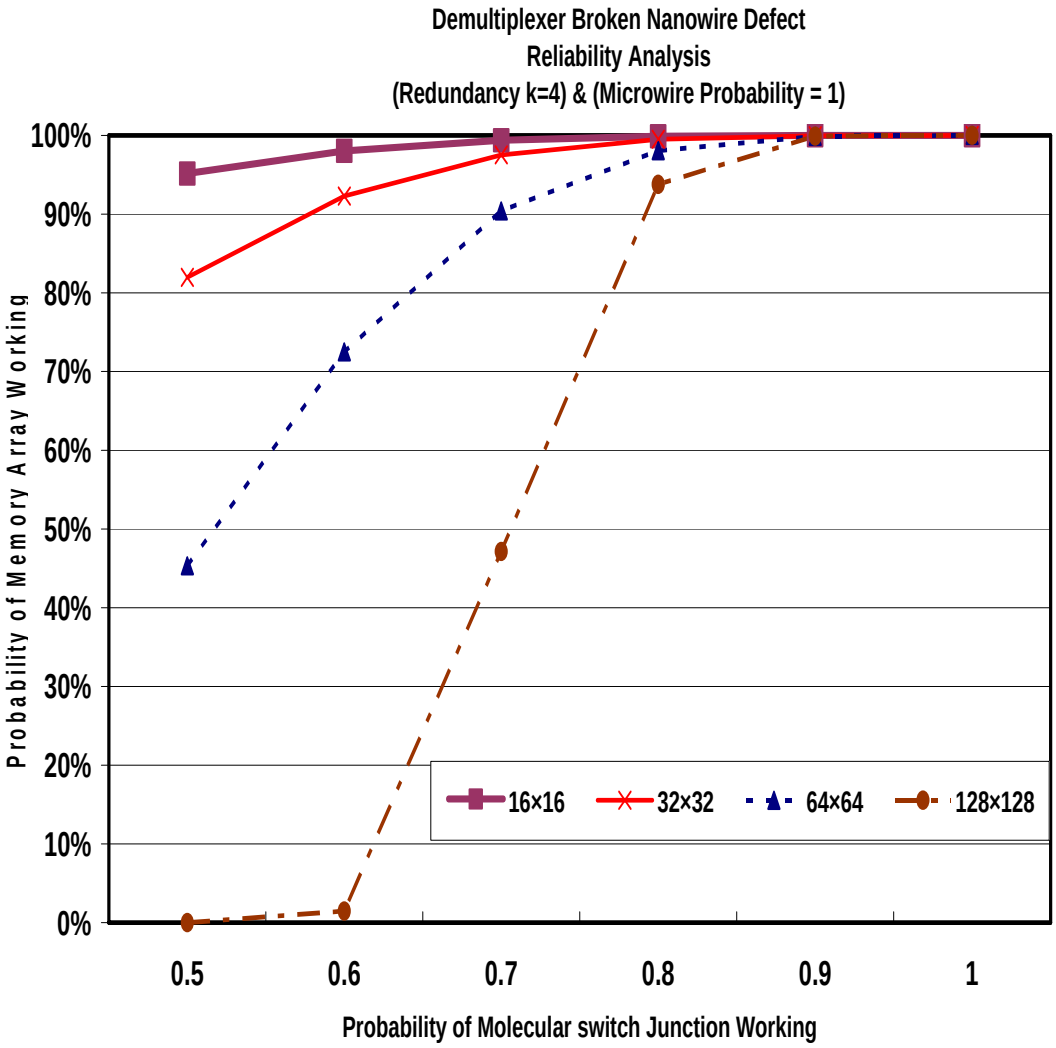

Figure 80. Demultiplexer reliability under broken nanowire defect conditions, when redundancy $k = 4$, under the same conditions as Figure 77. Results show better reliability improvements in the smaller arrays. In particular, the reliability trend lines improves with decreasing array sizes. In this case the 128×128 array was not simulated because of tool constraints. The reliability improvemtents are however, clearly visible, as reflected by the 100% reliability observed for working molecular junction probabilities greater thn 0.8 or 80%. A feat no achieved for k =2 and k = 3.

7.6 Demultiplexer MSJ vs. Banking Scheme Reliability Analysis Results

Comparisons between the MSJ and banking schemes were also studied. In this experiment, PRISM was used to simulate a condition where each molecular switch junction has a 1% probability of failing; at this error rate the MSJ can ensure the memory works with a greater than 99% probability for an 8×8 array with a redundancy of $k = 3$, as in the plot of Figure 72. The reliability of a non-defect tolerant 8×8 nanomemory array was also simulated— results show that given a molecular switch failure probability of 1%, the reliability of the nanomemory array was 53%. The cumulative binomial distribution formula was used then used to determine the probability of choosing a specified number of good banks from a total number of defect prone redundant banks. To implement a 16×16 nanomemory array configuration, four 8×8 banks are required. In the presence of a 1% molecular junction failure probability, over four times the number of required banks are needed to achieve a greater than 99% probability of the nanomemory array working, Similarly, a 128×128 nanomemory array will require 256 (8×8) banks for implementation and over 535 (8×8) banks at a 1% molecular junction failure probability, to achieve greater than 99% probability of the nanomemory array working; results of this analysis is provided in the plot of Figure 81. In comparison, MSJ requires $k \geq 3$ to achieve the same level of reliability (Figure 72). Delay and Power simulation results indicate similar performance advantages for the MSJ scheme over the banking scheme.

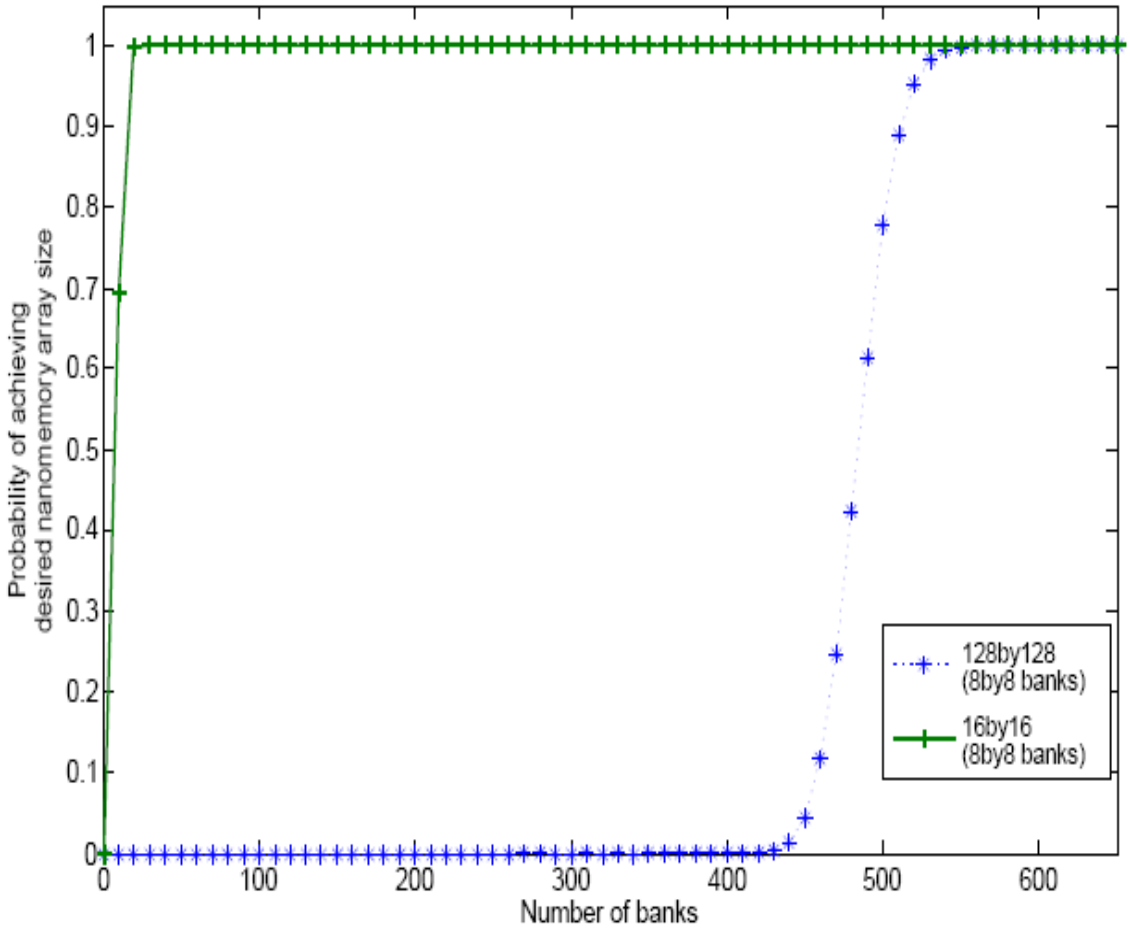

Figure 81. Data showing the number of 8×8 banks required to fabricate a 128×128 and 16×16 nanomemory array with a 1% molecular junction failure probability.

7.7 ECC Demultiplexer Reliability Analysis

This work builds upon the crossbar demultiplexer work of Kuekes et al. The goal here is to further enhance the reliability of a crossbar demultiplexer by implementing a combined error correction code and multi-switch junction scheme into the architecture of the crossbar demultiplexer. The primary aim of this approach is to find the optimal combination of the ECC-MSJ that will yield the best reliability using the smallest ECC-

MSJ configuration.

The reliability analysis is conducted by computing the probability of a nanowire functioning given a certain probability of a stuck-open fault occurring. When stuck-open faults occur on a nanowire, which causes it to become defective or fail, the nanowire is classified as a villain nanowire. Defective nanowires can then propagate their errors to cause another nanowire to fail; the affected nanowire can be labeled a victim nanowire. The probability of obtaining functioning nanowires can then be computed as the probability of a villain nanowire and a victim nanowire not occurring in the demultiplexer. To calculate the ECC-MSJ probability, the well thought derivation of Kuekes et al utilized is utilized and extended to include the effects of the MSJ on the reliability of the demultiplexer.

To compute the error correction probability of a linear code, the weight of the linear code must first be established. The weight of a linear code comprised of binary vectors is the number of ones contained in the code. Given any set S of binary vectors of length n, its weight profile function $W_S(i)$ can be defined, which describes for any integer $i \in [0, n]$, the number of elements of the set S with weight i. Let the set S be the code U with accompanying weight profile $W_U(i)$—denotes the number of codewords in U at each Hamming weight i. We can now introduce a set of dominating n-bit binary vectors $T(U)$ for code U, defined as $T(U) = \{\mathbf{e} \mid \mathbf{e}$ dominates y for some non-zero $y \in C\}$, where the expression "**e** dominates y" denotes the fact that **e** has a 1 at every bit position for which y has a 1. The weight profile of $T(U)$ is given by $W_{T(U)}(i)$.

A conflict occurs between two nanowire addresses when there are defective

junctions in one nanowire address causing it to be the same as another nanowire address—stuck-open defects usually manifest themselves as 0's in the afflicted nanowire addresses, and can cause conflicts such that two nanowire addresses become indistinguishable.

First we can calculate the probability of an output line being functional due to its own defects. Assuming defects to be statistically independent, $p_{villain}$ can be computed as follows.

$$p_{villain} = \sum_{i=1}^{n} W_{T(U)}(i)\, p^{i} (1-p)^{(n-i)} \tag{7-4}$$

Let the probability of an output line ceasing to be functional or usable be defined as p_{victim}. This output line is described as the *victim* because it does not lose its functionality due to its own defects, but because of interference from another defective output line. The probability p_{victim} can be calculated using the following equation.

$$\boldsymbol{p}_{victim} = 1 - \prod_{i=1}^{n} (1 - \boldsymbol{p}^{i})^{W_{U}(i)} \tag{7-5}$$

where $\boldsymbol{p}$ is the probability of the redundant molecular switches not working—attributed to the implementation of the MSJ scheme. The probability $\boldsymbol{p}$ is computed using PRISM. When U is the identity code then $W_U(i)$ is the binomial coefficient $W_U(i) = \binom{n}{i}$. The

probability of obtaining a functional output line can the be calculated as;

$$p_{funtional} = (1 - p_{villain}) \cdot (1 - p_{victim}) \tag{7-6}$$

Simulation results using these equations are presented in the following plots.

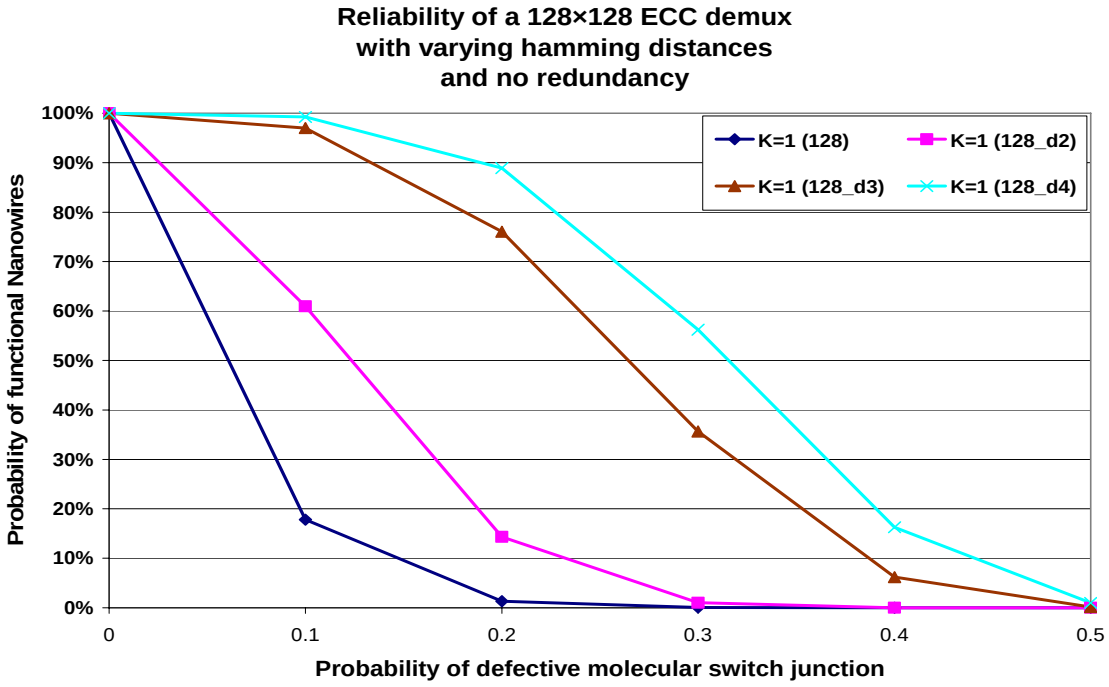

Figure 82. Shows the improvement in the reliability of the nanomemory demultiplexer when only ECC is implemented with no redundancy .

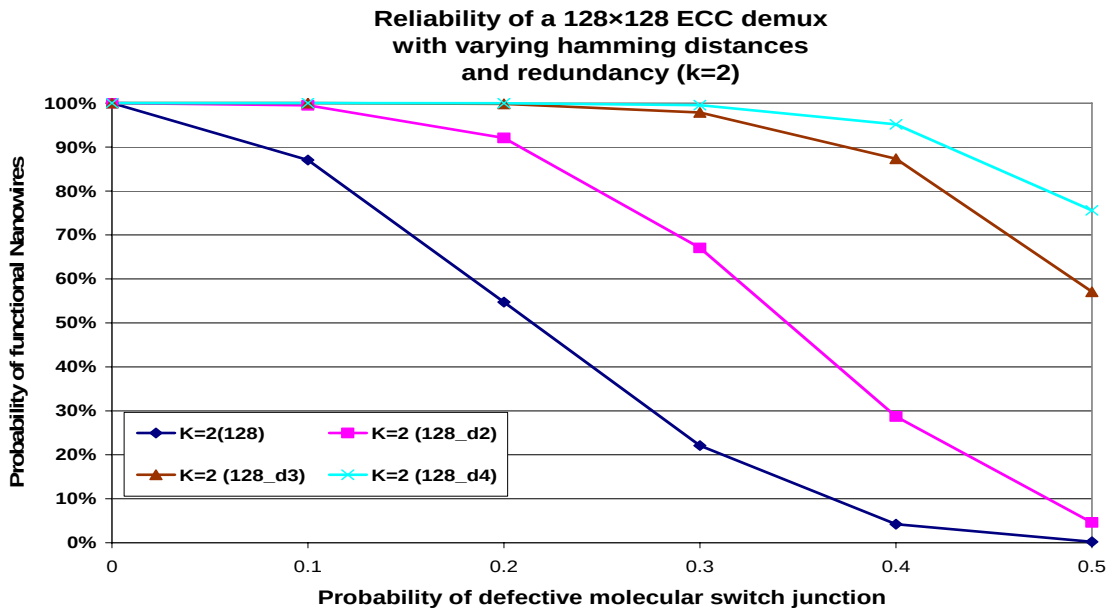

Figure 83. Shows the improvement in the reliability of the nanomemory demultiplexer when only ECC is implemented with with a redundancy of $k = 2$.

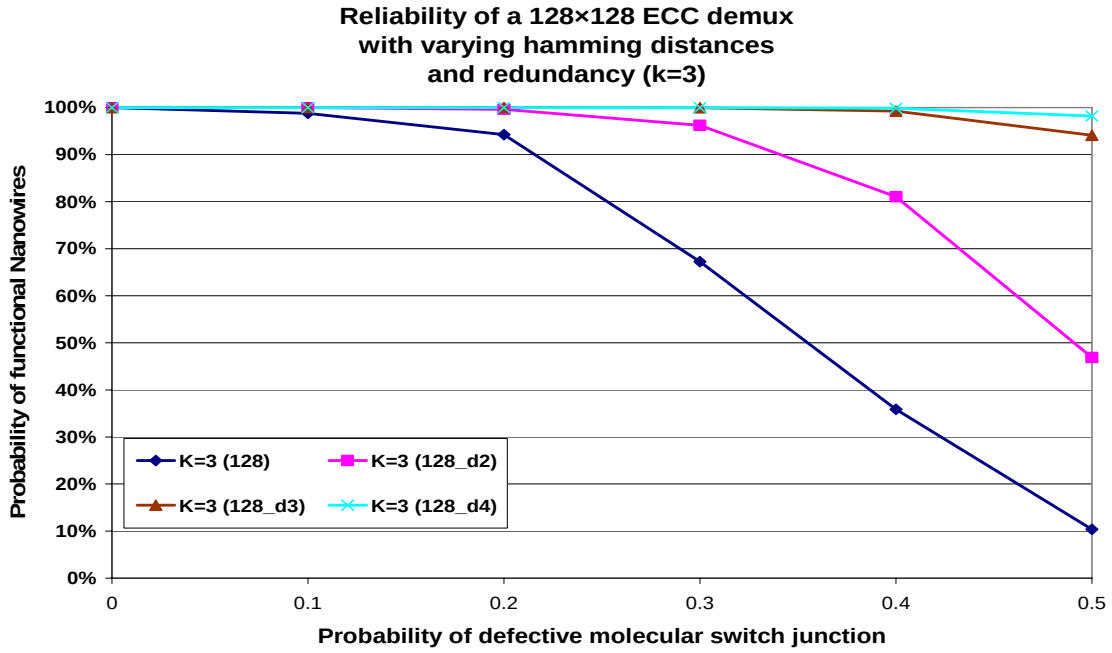

Figure 84. Shows the improvement in the reliability of the nanomemory demultiplexer when only ECC is implemented with with a redundancy of $k = 3$.

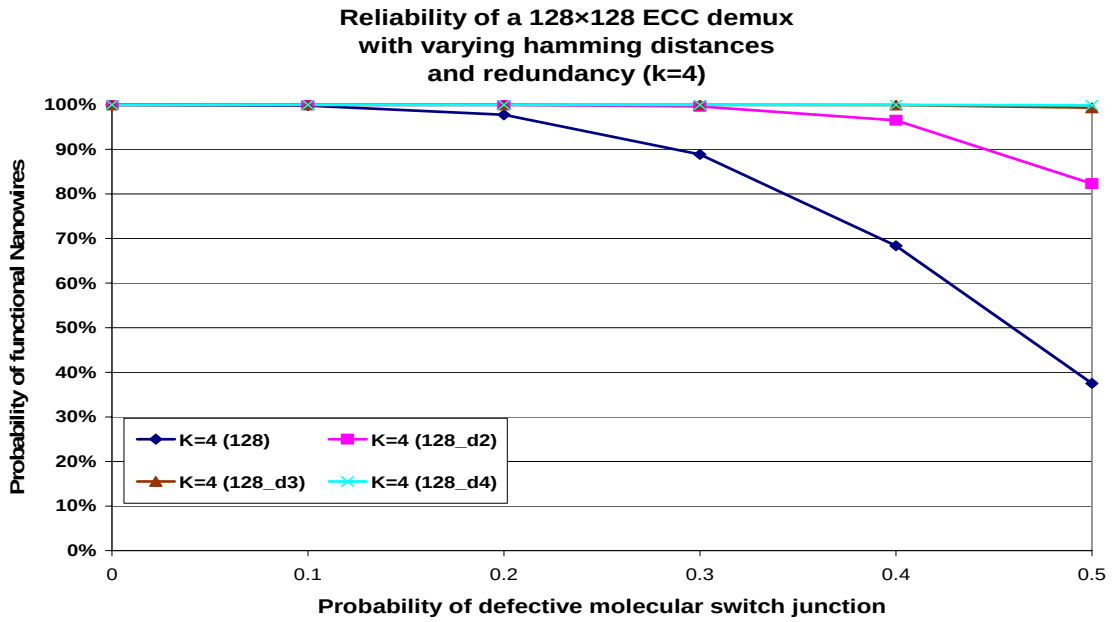

Figure 85. Shows the improvement in the reliability of the nanomemory demultiplexer when only ECC is implemented with a redundancy of $k = 4$.

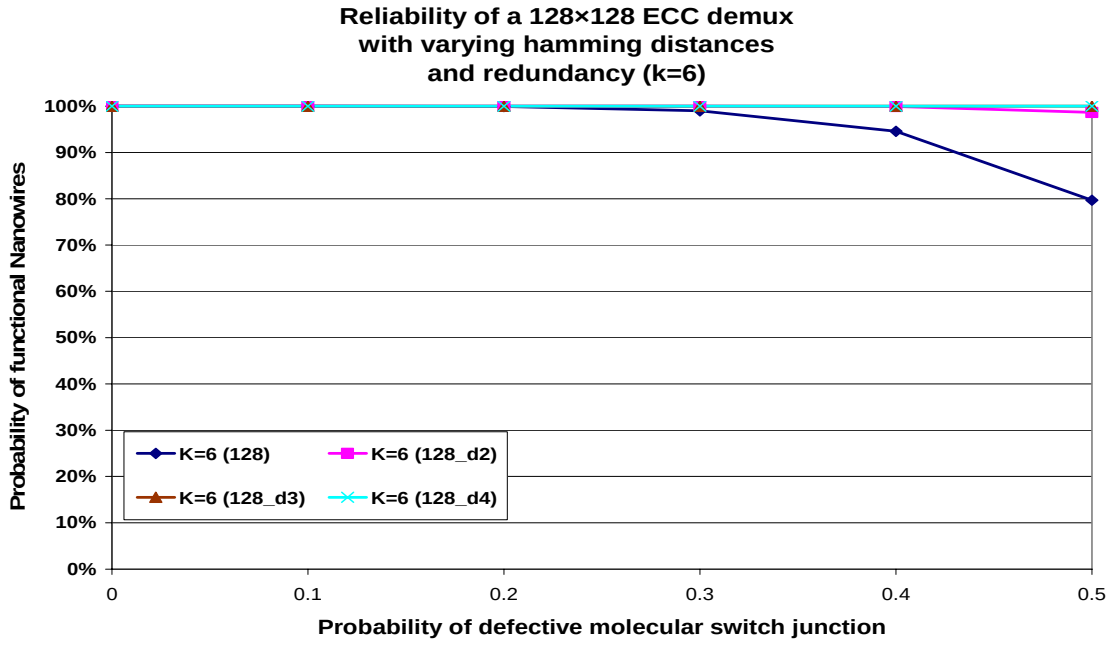

Figure 86. Shows the improvement in the reliability of the nanomemory demultiplexer when only ECC is implemented with with a redundancy of $k = 6$.

From the results observed in the plots of Figures 82 through 86, the relationship between the probability of obtaing a funtional demultiplexer nanowire addressline (ploted on the y-axis) is compared against the probability of stuck-open molecular switch junction faults ocurring. Error rates can not be too high if nanoelectronic devices are to be practical, hence a conservative essimate would be fabricating devices with defects in the 10% - 20% range. At this range, a fault tolerant scheme which implements a combination MSJ-ECC scheme with redundancy $k = 3$ and a hamming diatance $\mathbf{d} = 2$ could be used to obtain approximately 99% reliability in the demultiplexer, as observed in Figure 84. The penalties associated with the ECC implementation are more costly interms of power dissipation, as explained in chapter VI, due to the microwires. Hence, an efficient strategy for combining the MSJ and ECC schemes, would be to first optimize the number of MSJ redundancy, and then enhance the demultiplexer reliability by incorporating ECC. This approach is made clearer in the results ploted in Figures 87 through 89. Table 7 below, also paints a clearer picture of this point.

Table 7
Results from the plots of Figures 87 to 89, showing the improvements in demultiplexer reliability with increasing redundancy (*k),* and increasing hamming distance (*d*)

Array Size	***128×128***	***128×128***	***128×128***	***128×128***
Probability $P_{fail} = 0.2$	**Demultiplexer** ***d = 1***	**Demultiplexer** ***d = 2***	**Demultiplexer** ***d = 3***	**Demultiplexer** ***d = 4***
Redundancy	**Reliability**	**Reliability**	**Reliability**	**Reliability**
k=1	**1.32%**	**14.32%**	**76.05%**	**88.89%**
k=2	**54.71%**	**92.05%**	**99.82%**	**99.98%**
k=3	**94.22%**	**99.62%**	**100%**	**100%**
k=4	**97.74%**	**99.99%**	**100%**	**100%**
k=5	**99.52%**	**100%**	**100%**	**100%**
k=6	**99.92%**	**100%**	**100%**	**100%**

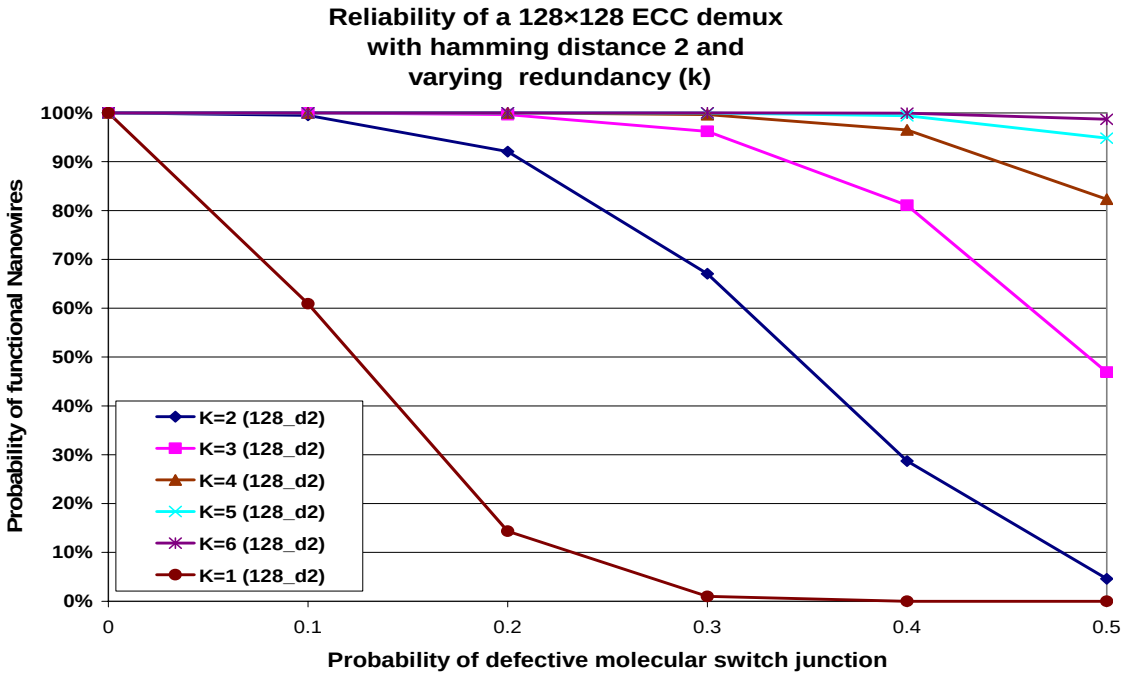

Figure 87. Shows the reliability improvement in a 128×128 nanomemory demultiplexer at an ECC hamming distance of 2 and increasing degrees of redundancy *k*.

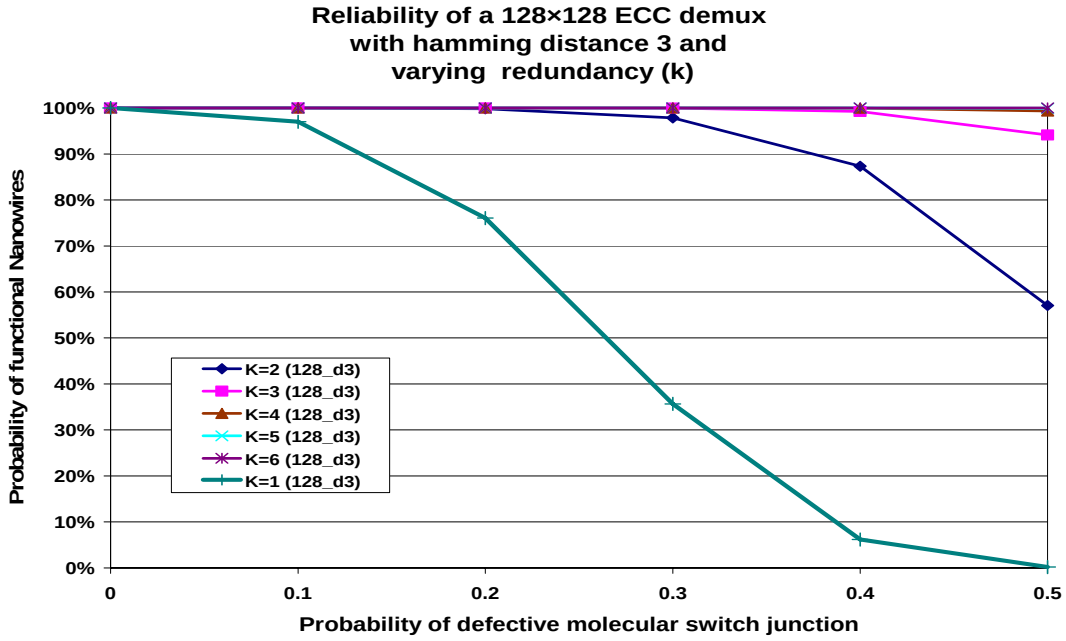

Figure 88. shows the reliability improvement in a 128×128 nanomemory demultiplexer at an ECC hamming distance of 3 and increasing degrees of redundancy *k*.

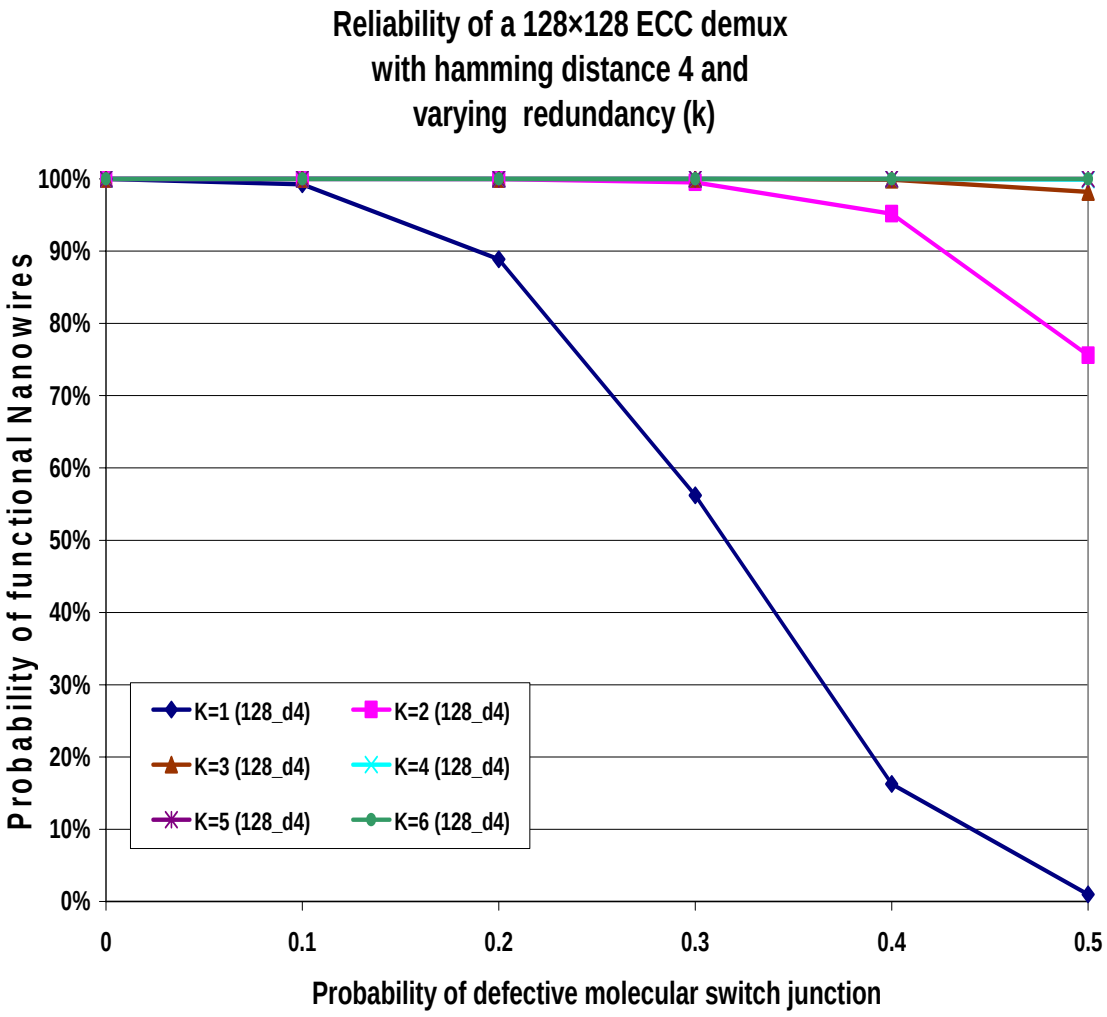

Figure 89. Shows the reliability improvement in a 128×128 nanomemory demultiplexer at an ECC hamming distance of 4 and increasing degrees of redundancy *k*.

In the plot of Figure 87, where the reliability of the demultiplexer is simulated with increasing redundancy and **d** = 2, approximately 99% demultiplexer reliability can be achieved for error rates equaling 20% at a redundancy of $k = 3$. If **d** = 3, that error rate can be extended to stuck-open faults rates of 30% for the same $k = 3$ implementation, as presented in Figure 88.

CHAPTER VIII

SUMMARY AND FUTURE WORK

8.1 Summary

In this work, an in-depth study of the impact imposed on the performance of a novel class of nanoelectronic molecular switch memories as a result of implementing defect and fault tolerant schemes in their architecture. In particular, the defect and fault tolerant architectures implemented here were the Multi-Switch Junction (MSJ), Banking, and Error Correction Coding (ECC) schemes.

In this dissertation, a defect and fault tolerant scheme, the MSJ scheme was developed and shown to improve the reliability of crossbar molecular switch nanomemories and demultiplexers. No commercial tools exist for evaluating the reliability and performance of nanomemories; as a result we developed an HSPICE simulation scheme based on PERL scripts, to evaluate the access time delay and power dissipation of crossbar nanomemories and demultiplexers. The probabilistic model checker, PRISM, was then modified to evaluate the reliability of the nanomemory and demultiplexer under stuck-open and broken nanowire defect conditions; Matlab was also used reliability evaluations using the banking scheme.

As a verification platform, a parameterized model of the crossbar molecular switch nanomemory and demultiplexer was first developed. The MSJ scheme, which is based on the use of redundancy, interweaved into the nanomemory fabric to form

multiple molecular switch junctions for single bit storage was also examined. The performance penalties imposed by the redundancy was analyzed and shown to scale reasonably well. Limits to the capacity of the memory were found to be inhibited by the lack of gain in the nanomemory circuit model. This was an important design parameter, as it affected the degree to which the nanomemory array sizes could be scaled; this is a problem that must be addressed if practical crossbar nanomemories are to be achieved.

Area analysis of the MSJ and banking architectures indicates a slight area utility advantage for the banking scheme, when compared to the MSJ scheme. In this work, our analysis was made with respect to the main crossbar nanomemory. The advantages of the banking scheme become less apparent when consideration is given to the area overhead incurred due to the additional CMOS logic and peripheral circuitry required for its implementation. The MSJ scheme on the other hand requires no additional peripheral circuitry or logic for implementation. The reliability of the molecular switch crossbar RAM was also investigated using PRISM. Results indicated improvements in defect and fault tolerance with increasing *k*. Further-more, as was *k* increased, the reliability of the MSJ architecture was found to improve with increasing memory array sizes.

In chapter VI it was determined that by including redundant SWNT bundles in the AND gate address-line of a nanomemory demultiplexer implemented with ECC, access time delay performance of the demultiplexer could be improved, as can be observed in the result plots of Figures 34 and 35. Furthermore, reliability computations using Matlab also showed the improvements yielded by this implementation on the fault-tolerance for a large array of defect probabilities. Utilizing bundles of SWNTs also resulted in better

signal conduction; it has been shown that using only single SWNT interconnects possess worse performance than current Cu interconnects. Bundle stacks are also desirable because they can be densely packed in a vertical geometry.

Simulations of a combined crossbar nanomemory and demultiplexer were performed and ensuing results presented. The introduction of the MSJ defect and fault tolerant scheme into the architecture of the nano-devices was found to improve the reliability of both the crossbar nanomemory and demultiplexer at a minimal cost to access time delay and power dissipated; which is in direct correlation with the results of the parameterized analysis. Simulation results also validated the previous findings that the MSJ was most effective for smaller nanomemory array sizes. The banking scheme of defect and fault tolerance was also studied, and it was found to provide less reliability than the MSJ scheme. Current simulation results suggest a combined MSJ and Banking scheme as the best way of achieving the desired optimal reliability. Smaller degrees of redundancy are needed to achieve high rates of reliability in smaller crossbar configurations, by having smaller more reliable redundant crossbar nanomemory and demultiplexer modules, a faster nanomemory with a conceivable parallel operation can be achieved.

Selected results have been tabulated to give a more descriptive and quantitative picture of the trade-offs between reliability and device performance. Cumulative results of the 32×32 and 128×128 crossbar nanomemory and demultiplexer arrays, as observed in Tables 8 and 9, are used to surmise these trade-offs. In Table 8(A) reliability using the MSJ is less effective than in the crossbar case because redundancy is only applied to row

nanowires; the molecular switches are multiplied by k not k^2 as is the case in the crossbar nanomemory. The access time delay penalties can are also much less than the power dissipated penalties because microwires are better conductors than nanowires, but on the other hand microwires generate more power than nanowires. Table 8(B) shows the relative similarities the access time delay and power penalties because the crossbar nanomemory is entirely comprised of nanowires and molecular switches. The MSJ is also more effective in this case because of the use of k^2 redundant molecular switches. The same analysis of Table 8 also apply to Table 9, however, it can be observed that the MSJ is much more effective in the 32×32 array than it is 128×128 array. Taking a look at the $k = 2$ case of Table 8(B) and 9(B), the contrast in achievable reliability can be clearly observed by the approximately 50% difference in the reliability at that level (90.7% and 21.26% respectively). Thus leading to previously stated conclusions that the MSJ is most effective at smaller array sizes due to the smaller number of error prone devices implemented.

Table 8
Trade-off between reliability and performance for the nanomemory demultiplexer (A), and the crossbar nanomemory (B) for a 32×32 array

Array Size	32×32	(A)	
Probability	P=0.9		
No ECC Demux Redundancy	Reliability	Delay	Peak Power
k=1	0	6.13E-09	2.42E-02
k=2	3.34%	6.53E-09 (6.64%)	4.84E-02 (82%)
k=3	82.19%	6.88E-09 (12.33%)	7.25E-02 (161.6%)
k=4	98.20%	7.19E-09 (17.43%)	9.65E-02 (247.7%)

Array Size	32×32	(B)	
Probability	P=0.9		
Crossbar Nanomemory Redundancy	Reliability	Delay	Peak Power
k=1	0	7.363E-11	1.04E-03
k=2	90.75%	1.37E-10 (86.00%)	2.06E-03 (100%)
k=3	100%	2.01E-10 (173.38%)	3.09E-03 (199.85%)
k=4	100%	2.66E-10 (265.15%)	4.11E-03 (298.92%)

Table 9
Trade-off between reliability and performance for the nanomemory demultiplexer (A), and the crossbar nanomemory (B) for a 128×128 array

Array Size	**128×128**	**(A)**	
Probability	**P=0.9**		
No ECC Demux Redundancy	**Reliability**	**Delay**	**Peak Power**
k=1	**0**	**9.62E-09 (Baseline)**	**1.11E-01 (Baseline)**
k=2	**0.004%**	**1.23E-08 (27.90%)**	**2.02E-01 (81.98%)**
k=3	**35.22%**	**1.48E-08 (54.00%)**	**2.90E-01 (161.26%)**
k=4	**90.676%**	**1.73E-08 (79.70%)**	**3.86E-01 (247.75%)**

Array Size	**128×128**	**(B)**	
Probability	**P=0.9**		
Crossbar Nanomemory Redundancy	**Reliability**	**Delay**	**Peak Power**
k=1	**0**	**2.03E-10 (Baseline)**	**3.48E-03 (Baseline)**
k=2	**21.26%**	**3.93E-10 (93.62%)**	**6.91E-03 (98.56%)**
k=3	**100%**	**6.40E-10 (215.07%)**	**1.04E-02 (198.85%)**
k=4	**100%**	**7.77E-10 (282.62%)**	**1.38E-02 (296.55%)**

8.2 Future Work

This body of work represents a first known trade-off analysis of the reliability induced penalty on performance of this class of nanomemories. Results described in his dissertation, represent a first order analysis of the crossbar molecular switch nanomemory and demultiplexer, a higher order analysis, while not necessary for the goals of this work, was not possible due to the lack of detailed empirical data. In the future, steps should be taken to engage the laboratories or research institutions which fabricate these devices in a mutually beneficial collaboration, so that the models described in here can be optimized using data not provided in literature. In addition, the creation of higher order defect models which take into account the simultaneous occurrence of transients and permanent faults during device operation will be very important. There is also a need for a new class to simulation engines, designed to evaluate and model nano devises at higher levels of abstraction.

REFERENCES

[1] Y. Chen, G. Jung, D. A. A. Ohlberg, X. Li, D. Stewart, J. O. Jeppesen, K. A. Nielsen, J. F. Stoddart, and R. S. Williams, "Nanoscale molecular-switch crossbar circuits," *Nanotechnology*, vol. 14, pp. 462-468, 2003.

[2] 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,* vol. 289, pp. 94, 2000.

[3] W. Goddard, D. Brenner, S. Lyshevski, G. Lafrate, *Handbook of Nanoscience Engineering and Technology*, CRC Press, Boca Raton, FL, pp. 3-12, 2003.

[4] P. G. Depledge, "Fault-tolerant computer systems," *IEEE Proc.* vol. 128, pp. 257–72, 1981.

[5] S. Spagocci and T. Fountain, "Fault rates in nanochip devices," *Electrochem. Soc. Proc,* vol. 99, pp. 354–368, 1999.

[6] J. Von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in *Automata Studies,* C.E. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton Univ. Press, 1956, pp. 43-98.

[7] B. Culbertson, R. Amerson, R. Carter, P. Kuekes, G. Snider, "Defect tolerance on the teramac custom computer," in *Proceedings of the 1997 IEEE Symposium on FPGA's for Custom Computing Machines*, pp. 116-123, 1997.

[8] C. Dekker, "Carbon nanotubes as molecular quantum wires," *Phys. Today*, pp.22-28, May 1999.

[9] S. Jin, D. Whang, M. McAlpine, and R. Friedman, "Interconnection and integration of nanowire devices without registration," *Nano Letters,* vol. 4, pp. 915-919, 2004.

[10] R. Beckman, E. Johnston-Halperin, Y. Luo, N. Melosh, J. Green, and J. R. Heath, "Fabrication of conducting silicon nanowire arrays," *J. Appl. Phys. Communication,* vol. 96 , no. 10, pp. 5921-5923, 2004.

[11] Y. Cui, X. Duan, J. Hu and C. M Lieber, "Doping and electrical transport in silicon nanowires," *J. Phys.Chem. B,* vol. 104, pp. 5213-5216, June (2000).

[12] W. Goddard, D. Brenner, S. Lyshevski, G. Lafrate, *Handbook of Nanoscience Engineering and Technology,* CRC Press, Boca Raton, FL, pp. 3-14, 2003.

[13] R. Waser (Editor), *Nanoelectronics and Information Technology: Advanced Electronic Materials and Novel Devices*, Wiley-VCH, Berlin, 2003.

[14] J. G. Kushmerick, S. K. Pollack, J. C. Yang, J. Naciri, D. B. Holt, M. A. Ratner, and R. Shashidhar, "Understanding charge transport in molecular electronics," *Ann. N.Y. Acad. Sci,* vol. 1006, pp. 277–290, 2003.

[15] G. Snider, P. Kuekes, T. Hogg and R. S. Williams, "Nanoelectronics architectures," *Appl. Phys*, A vol. 80, pp. 1183-1195, 2005.

[16] Z. L. Wang, *Nanowires and Nanobelts Materials, Properties and Devices Metal and Semiconductor Nanowires Volume I,* Kluwer Academic Publishers, New York. 2003.

[17] R. M. Metzger, B. Chen, U. Hopfner, M. V. Lakshmikantham, D. Vuillaume, T. Kawai, X. Wu, H. Tachibana, T. V. Hughes, H. Sakurai, J. W. Baldwin, C. Hosch, M. P. Cava, L. Brehmer and G. J. Ashwell, "Unimolecular electrical rectification in hexadecylquinolinium tricyanoquinodimethanide," *J. Am. Chem. Soc.*, vol. 119, pp. 10455, 1997.

[18] Z. Yao, C. L. Kane, and C. Dekker, "High-field electrical transport in single-wall carbon nanotubes," *Phys Rev Lett,* vol. 84, pp. 2941-2944, 2000.

[19] M. Buttiker, H. Thomas and A. Pretre, "Mesoscopic capacitors," *Physics Letters A*, vol. 364, pp. 180, 1993.

[20] W. Honlein, “New prospects for microelectronics: Carbon nanotubes,” *Jpn. J. Appl. Phys.,* vol. 41, pp. 4370, 2002.

[21] T. H. Rueckes, K. Kim, E. Joselevich, G. Y. Tseng, C. Cheung and CH. M. Lieber, “Carbon nanotube-based nonvolatile random access memory for molecular computing,” *Science*, vol. 289, pp. 94 - 97, July 2000.

[22] NRAM (Nanotube/Nonvolatile RAM) operational video [Online]. Available: http://www.nantero.com/nram.html. *Accessed on November 2007*. Fabricated by Nantero.

[23] F. Seker, K. Meeker, T. F. Kuech, and A. B. Ellis, “Surface chemistry of prototypical bulk II-VI and III-V semiconductors and implications for chemical sensing,” *Chem. Rev.,* vol. 100, no. 7, pp. 2505–2536, 2000.

[24] R. K. Iler, *The Chemistry of Silica*, Wiley, New York (1979).

[25] X. Duan, Y. Huang, Y. Cui, J. Wang, and C. M. Lieber, “Indium phosphide nanowires as building blocks for nanoscale electronic and optoelectronic devices,” *Nature,* vol. 409, pp. 66–69, Jan. 2001.

[26] Y. Cui and C. Lieber, "Functional nanoscale electronic devices assembled using silicon nanowire building blocks," *Science,* vol. 291, pp. 851, 2001.

[27] Y Huang, X. Duan, Y. Cui, L. Lauhon, K. Kim and C. M Lieber, “Logic gates and computation from assembled nanowire building blocks,” *Science,* vol. 294, pp. 1313, 2001.

[28] C. M. Lieber, “One-dimensional nanostructures: chemistry, physics & applications,” *Solid State Commun.,* vol. 107, pp. 607–616, 1998.

[29] J. Hu, T. W. Odom and C. M. Lieber, "Chemistry and physics in one Dimension: synthesis and properties of nanowires and nanotubes," *Acc. Chem. Res,* vol. 32, pp. 435, 1999.

[30] Y. Cui, L. J. Lauhon, M. S. Gudiksen, J. Wang and C. M Lieber, "Diameter-controlled synthesis of single-crystal silicon nanowires," *Appl. Phys. Lett.,* vol. 78, pp. 2214, 2001.

[31] J. E. Green, J. W. Choi, A. Boukai, Y. Bunimovich, E. Johnston-Halperin, E. DeIonno, Y. Luo, B. A. Sheriff, K. Xu, Y. Shik Shin, H-R. Tseng, J. F. Stoddart and J. R. Heath, "A 160-kilobit molecular electronic memory patterned at 10^11 bits per square centimetre," *Nature*, vol. 445, pp. 414-417, 2007.

[32] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams, "A defect-tolerant computer architecture: Opportunities for nanotechnology," *Science*, vol. 280, pp. 1716-1721, 1998.

[33] P. Yang and F. Kim, "Langmuir-Blodgett assembly of one-dimensional nanostructures," *ChemPhysChem*, vol. 3, pp. 503-506, 2002.

[34] J. W. Choi, "Bistable Rotaxane Based Molecular Electronics: Fundamentals and Applications," PhD Thesis, California Institute of Technology, Pasadena, CA. May 2007.

[35] N. Melosh, A. Boukai, F. Diana, B. Gerardot, A. Badolato, P. Petroff, and J. R. Heath, "Ultrahigh density Nanowire Lattices and Circuits," *Science*, vol. 300, pp.112, 2003.

[36] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective,* Prentice Hall Series in Electronics and VLSI, Prentice Hall, Englewood Cliffs, NJ, 1996.

[37] W. Goddard, D. Brenner, S. Lyshevski, G. Lafrate, *Handbook of Nanoscience Engineering and Technology*, CRC Press, Boca Raton, FL, pp. 6-5, 2003.

[38] W. Goddard, D. Brenner, S. Lyshevski, G. Lafrate, *Handbook of Nanoscience Engineering and Technology*, CRC Press, Boca Raton, FL, pp. 4-8, 2003.

[39] Hideki Imai, *Essentials of Error-Control Coding Techniques,* Academic Press Inc, New York, July 1990.

[40] I. Amlani, A. O. Orlov, G.L. Snider, C. S. Lent, and G. H. Bernstein "Demonstration of a six-dot quantum cellular automata system", *Appl. Phys. Lett,* vol. 72, pp. 2179–2181, 1998.

[41] A. Kumar and S. Tiwari, "Defect tolerance for nanocomputer architecture," in *Proceedings of the 2004 international workshop on System level interconnect prediction SESSION: Unconventional interconnects,* pp. 89-96, February 14–15, 2004.

[42] O. Turel and K. Likharev, "CrossNets: possible neuromorphic networks based on nanoscale components," *Int J Circuit Theory Appl*, vol. 31, pp. 37-53, January/February 2003.

[43] K. Nikolic, A. Sadek and M. Forshaw, "Fault-tolerant techniques for nanaocomputers," *Nanotechnology,* vol. 13, pp. 357-362, June 2002.

[44] X. D. Cui, A. Primak, X. Zarate, J. Tomfohr, O. F. Sankey, A. L. Moore, T. A. Moore, D. Gust, G. Harris, S. M. Lindsay, "Reproducible measurements of single-molecule conductivity," *Science,* vol. 294, pp. 571, 2001.

[45] Y. Chen, D. A. A. Ohlberg, X. Li, D. R. Stewart, R. S. Williams, J. O. Jeppesen, K. A. Nielsen, J. F. Stoddart, D. L. Olynick and E. Anderson, "Nanoscale molecular-switch devices fabricated by imprint lithography," *Appl. Phys. Lett,* vol. 82, no. 10, pp. 1610-1612, 2003.

[46] G. Y. Jung, S. Ganapathiappan, D. A. A. Ohlberg, D. L. Olynick, Y. Chen, W. M. Tong, and R. S. Williams, "Fabrication of a 34×34 crossbar structure at 50nm half-pitch by uv-based nanoimprint lithography," *Nano Letters,* vol. 7, pp. 1225 - 1229, 2004.

[47] Z. Yu, W. Wu, G-Y. Jung, D. L. Olynick, J. Straznicky, X. Li, Z. Li, W. Tong, J. A Liddle, S-Y. Wang and R Stanley Williams, "Fabrication of 30 nm pitch imprint moulds by frequency doubling for nanowire arrays," *Nanotechnology,* vol. 1, pp. 4956-4961, 2006.

[48] The International Technology Roadmap for Semiconductors (ITRS): process integration, devices, and structures. (Semiconductor Industry Association, San Jose, California, 2005). [Online]. Available: http://www.itrs.net/reports.html. *Accessed in 2006*.

[49] W. Goddard, D. Brenner, S. Lyshevski, G. Lafrate, *Handbook of Nanoscience Engineering and Technology,* CRC Press, Boca Raton, FL, p. 4-4, 2003.

[50] M. Khatun, T. Barclay, I. Sturzu, P. D. Tougaw, "Fault tolerance properties in quantum-dot cellular automata devices," *J. Phys. D: Appl Phys*. vol. 39, pp. 1489-1494, 2006.

[51] P. L. McEuen, "Artificial atoms: new boxes for electrons," *Science*, vol. 278, p. 1729, 1997.

[52] K. Goser, P. Glosekotter, J. Dienstuhl, *Nanoelectronics and Nanosystems: From Transistors to Molecular and Quantum Devices,* Springer-Verlag Heidelberg, 2004.

[53] C. S. Lent, M. Liu, and Y. Lu, "Bennett clocking of quantum-dot cellular automata and the limits to binary logic scaling," *Nanotechnology*, vol. 17, pp. 4240-4421, 2006.

[54] A. O. Orlov, I. Amlani, G. H. Bernstein, and G. L. Snider, "Realization of a functional cell for quantum-dot cellular automata," *Science*, vol. 277, pp. 928-930, 1997.

[55] I. Amlani, A. O. Orlov, G. L. Snider, C. S. Lent, and G. H. Bernstein, "External charge state detection of a double-dot system," *Appl. Phys. Lett,* vol. 71, pp. 1730–1732, 1997.

[56] I. Amlani, A. O. Orlov, G. L. Snider, C. S. Lent, and G. H. Bernstein "Demonstration of a six-dot quantum cellular automata system", *Appl. Phys. Lett,* vol. 72, pp. 2179–2181, 1998.

[57] I. Amlani, A. O. Orlov, G. Toth, C. S. Lent, G. H. Bernstein, and G. L. Snider "Digital logic gate using quantum-dot cellular automata," *Science*, vol 284, pp. 289–91, 1999.

[58] A. O. Orlov, I. Amlani, G. Toth, C. S. Lent, G. H. Bernstein, and G. L. Snider, "Experimental demonstration of a binary wire for quantum-dot cellular automata," *Appl. Phys. Lett,* vol. 74, pp. 2875–2877, 1999.

[59] I. Amlani, A. O. Orlov, R. K. Kummamuru, G. H. Bernstein, C. S. Lent, and G. L. Snider, "Experimental demonstration of a leadless quantum-dot cellular automata cell," *Appl. Phys. Lett,* vol. 70, pp. 738–40, 2000.

[60] R. K. Kummamuru, A. O. Orlov, C. S. Lent, G. H. Bernstein, and G. L. Snider, "Operation of a quantum-dot cellular (QCA) shift register and analysis of errors," *IEEE Trans. Electron Devices*, vol. 50, pp. 1906–13, 2003.

[61] H. Qi, S. Sharma, Z. Li. G. L. Snider, A. O. Orlov. C. S. Lent, and T. P. Fehlner, "Molecular QCA cell. Electric field driven stitching of a silicon surface bound array of vertically oriented two-dot molecular quantum cellular automata," *J. Am. Chem. Soc*, vol. 125, pp. 15250-15259, 2003.

[62] J. Jiao, G. J. Long, F. Grandjean, A. M. Beatty, and T. P. Fehlner, "Building blocks for the molecular expression of quantum cellular automata. Isolation and characterization of a covalently bonded square array of two ferrocenium and ferrocene complexes," *J. Am. Chem. Soc,* vol. 125, pp. 7522, 2003.

[63] K. S. Kwok, "Materials for future electronics," *Materials Today*, vol. 6, no. 12, pp. 20-27, December 2003.

[64] J. M Tour, W. L. Van Zandt, C. P. Husband, S. M. Husband, L. S. Wilson, P. D. Franzon, and D. P. Nackashi, "Nanocell logic gates for molecular computing". *IEEE Trans on Nanotechnology*, vol. 1, no 2, June 2002.

[65] J. H. Smet, T. P. E. Broekaert, and C. G. Fonstad, "Peak-to-valley current ratios as high as 50:1 at room temperature in pseudomorphic $In_{0.5}Ga_{0.47}As$/AIAs/InAs resonant tunneling diodes," *J. Appl. Phys.,* vol. 71, pp. 2475-2477, Mar. 1992.

[66] J. M Tour, L. Cheng, D. P. Nackashi, Y. Yao, A. K. Flatt, S. K. St Angelo, T. E. Mallouk, and P. D. Franzon, "NanoCell electronic memories". *J. Amer. Chem. Soc.,* vol. 125, no. 43, pp. 13279-13283, 2003.

[67] J. M. Seminario, Y. Ma and V. Tarigopula. "The nanoCell: a chemically assembled molecular electronic circuit", *IEEE Sensor Journal*, vol. 6 (6), pp. 1614-1626, 2006.

[68] L. J. Richter, C. S. C. Yang, P. T. Wilson, C. A. Hacker, R. D. van Zee, J. J. Stapleton, and D. L. Allara, "Optical characterization of oligo(phenylene-ethynylene) self-assembled monolayers on gold," *J. Phys. Chem., B*, vol. 108, no. 33, pp. 12547–12559, 2004.

[69] M. P. Stewart, F. Maya, D. V. Kosynkin, S. M. Dirk, J. J. Stapleton, C. L. McGuiness, D. L. Allara, and J. M. Tour, "Direct covalent grafting of conjugated molecules onto Si, GaAs, and Pd surfaces from aryidiazonium salts," *J. Amer. Chem. Soc.*, vol. 126, no. 1, pp. 370–378, 2004.

[70] J. J. Stapleton, P. Harder, T. A. Daniel, M. D. Reinard, Y. X. Yao, D. W. Price, J. M. Tour, and D. L. Allara, "Self-assembled oligo(phenylene-ethynylene) molecular electronic switch monolayers on gold: Structures and chemical stability," *Langmuir*, vol. 19, no. 20, pp. 8245–8255, 2003.

[71] S. Frey, A. Shaporenko,M. Zharnikov, P. Harder, and D. L. Allara, "Self assembled monolayers of nitrile-functionalized alkanethiols on gold and silver substrates," *J. Phys. Chem., B*, vol. 107, no. 31, pp. 7716–7725, 2003.

[72] B. C. Haynie, A. V. Walker, T. B. Tighe, D. L. Allara, and N. Winograd, "Adventures in molecular electronics: How to attach wires to molecules," *Appl. Surf. Sci.*, vol. 203, pp. 433–436, 2003.

[73] G. P. Berman, V. I. Tsifrinovich, and D. L. Allara, "Entangled spin states in self-assembled monolayer systems," *Phys. Rev. B, Condens. Matter*, vol. 66, no. 19, pp. 193406, Nov. 2002.

[74] W. Goddard (Editor), D. Brenner (Editor), S. Lyshevski (Editor), G. Lafrate (Editor), *Handbook of Nanoscience Engineering and Technology*, CRC Press, Boca Raton, FL, pp. 3-2, 2003.

[75] R. W. Keys, "Fundamental limits of silicon technology," *Proc. IEEE*, vol. 89, pp. 227-239, 2001.

[76] J. D. Plummer, and P. B. Griffin, "Material and process limits in silicon VLSI technology," *Proc. IEEE,* vol. 89, pp. 240-258, 2001.

[77] R. L. Harriott, "Limits of lithography," in *Proceedings of IEEE,* vol. 89, pp. 366-374, 2001.

[78] W. Goddard (Editor), D. Brenner (Editor), S. Lyshevski (Editor), G. Lafrate (Editor), *Handbook of Nanoscience Engineering and Technology*, CRC Press, Boca Raton, FL, pp. 4-3, 2003.

[79] Y. Luo, C. P. Collier, K. Nielsen, J. Jeppesen, J. Perkins, E. DeIonno, A. Pease, J. Fraser Stoddart, and J. R. Heath, "Molecular electronics random access memory circuits," *ChemPhysChem,* vol. 3, pp. 519, 2002.

[80] K. K. Likharev, "Nano and Giga Challenges in Microelectronics," in *Electronics below 10 nm,* A. Korkin (Editor), Elsevier, Amsterdam, pp. 27–68, 2003.

[81] K. K. Likharev and D. B. Strukov, "CMOL: devices, circuits, and architectures," in *Introducing Molecular Electronics*, G. Cuniberti, G. Fagas, and K. Richter, (Editors), pp. 447–478. Springer, Berlin, 2005.

[82] D. B Strukov, and K. K. Likharev, "CMOL FPGA: a reconfigurable architect for hybrid digital circuits with two-terminal nanodevices," *Nanotechnology,* vol. 16, pp. 888-900, 2005.

[83] D. B Strukov, and K. K. Likharev, "CMOL FPGA: a reconfigurable architect for hybrid CMOS/nanodevices circuits," *FPGA '06* (Monterey, CA, USA, Feb 2006).

[84] G. Snider and R. Williams, "Nano/CMOS architecture using a field programmable nanowire interconnect," *Nanotechnology,* vol. 18, pp. 11, 2007.

[85] R. Iris Bahar, "Trends and future directions in nano structure based computing and fabrication," *in proceedings of ICCD 2006,* pp. 522-527, Oct 1-4, 2006.

[86] R. Beckman, E. Johnston-Halperin, Yi Luo, Jonathan E. Green, and James R. Heath, "Bridging Dimensions: Demultiplexing ultrahigh-density nanowire circuits," *Science*, vol. 310, 21 Oct 2005.

[87] R. P. Feynman, *Lectures in Computation*, A. J. G. Hey, R. W. Allen, Editors. Addison-Wesley, Menlo Park, CA, 1996.

[88] A. Coker, V. Taylor, "Performance and reliability analysis of a scaled multi-switch junction crossbar nanomemory and demultiplexer," in *Proceedings of 7th IEEE Conference on Nanotechnology*, Aug 2-5, 2007.

[89] P. J. Kuekes, Warren Robinett, Gadiel Seroussi, and R. Stanley Williams, "Defect-tolerant interconnect to nanoelectronic circuits: internally redundant demultiplexers based on error correcting code," *Nanotechnology,* vol. 16, pp. 869-882, 2005.

[90] P. J. Kuekes R. S. Williams, "Demultiplexer for a molecular wire crossbar network," U.S Patent 6,256,767 (3 July 2003).

[91] A. Dehon, P. Lincoln, and J. Savage, "Stochastic assembly of sublithographic nanoscale interfaces," *IEEE Trans. Nanotechnol.,* vol. 2, no. 3, pp. 165-174, Sep. 2003.

[92] M. S. Gudiksen, J. Wang, and C. M. Lieber, "Synthetic control of the diameter and length of semiconductor nanowires," *J. Phys. Chem. B,* vol. 105, pp. 4062–4064, 2001.

[93] Y. Wu, R. Fan, and P. Yang, "Block-by-block growth of single-crystalline Si/SiGe superlattice nanowires," *Nano Lett.*, vol. 2, no. 2, pp. 83–86, Feb. 2002.

[94] M. T. Björk, B. J. Ohlsson, T. Sass, A. I. Persson, C. Thelander, M. H. Magnusson, K. Depper, L. R. Wallenberg, and L. Samuelson, "One-dimensional steeplechase for electrons realized," *Nano Lett.*, vol. 2, no. 2, pp. 87–89, Feb. 2002.

[95] D. R. Stewart, G. Gibson, G. Y. Jung, W. Wu, J. Straznicky, W. Tong, Z. Li and R. Stanley Williams, "Direct-write programming of nanoscale demultiplexer arrays," *Nanotechnology* vol. 18, pp. 5, 2007.

[96] P. J. Kuekes, W. Robinett and R. S. Williams, “Defect-tolerance in resistor-logic demultiplexers for nanoelectronics.” *Nanotechnology,* vol. 17, pp. 2466-2474, 2006.

[97] P. J. Burke, “An RF circuit model for carbon nano-tubes,” *IEEE Trans Nanotechnology*, vol. 2, no. 1, pp. 55-58, March 2003.

[98] S. K. Shukla (Editor), and R. I. Bahar (Editor), *Nano, Quantum and Molecular Computing: Implications to High Level Design and Validation,* Kluwer Academic Publishers, Boston, MA, 2004.

[99] D. Crawley (Editor), K. Nikolic (Editor), M. Forshaw (Editor), *3D Nanoelectronic Computer Architecture and Implementation*, IOP series in Materials Science and Engineering, IOP Publishing Ltd Bristol, UK, 2005.

[100] H. Naeimi and A. Dehon “Fault-tolerant sub-lithographic design with rollback recovery,” *Nanotechnology,* vol. 19, 115708 (17pp), 2008.

[101] K. Nikolic, A. Sadek and M. Forshaw, “Architecture for reliable computing with unreliable devices,” in *Proceedings of IEEE NANO,* pp. 254-259, 2001.

[102] J. Han, and P. Jonker, “A systems architecture solution for unreliable nanoelectronic devices,” *Transactions on Nanotechnology*, vol. 1, no. 4, December 2002.

[103] M. M. Ziegler, M. R. Stan, "CMOS/nano co-design for crossbar-based Molecular electronic systems," *IEEE Transactions on Nanotechnology*, vol. 2, no. 4, pp. 217-230, December 2003.

[104] M. Butts, A. DeHon, and S. Goldstein, “Molecular electronics: devices, systems and tools for gigagate, gigabit chips,” in *Proceedings of the 2002 IEEE/ACM International Conference on Computer Aided Design (IC-CAD) (San Jose, CA),* pp. 433-440, November 2002.

[105] G. Wang, T. Li, and Jianer Chen: "A probabilistic approach to fault-tolerant routing algorithm on mesh networks," in *Proceedings of ICPADS*, pp. 577, 2004.

[106] G. Wang and J. Chen, "Probabilistic analysis of connectivity on mesh networks," in *Proceedings of 1st IEEE Conference on Information Technology (ITCC 2003)*, pp. 362-366, 2003.

[107] A. Coker, V. Taylor, D. Bhaduri, S. Shukla, A, Raychowdhury, and K. Roy, "Multi-Junction fault tolerance architecture for nanoscale crossbar memories," *IEEE Transactions on Nanotechnology*, vol. 7, no. 2, pp. 202-208, March 2008.

[108] A. Dehon, S.C. Goldstein, P. J. Kuekes and P. Lincoln, "Nonphotolithographic nanoscale memory density prospects," *IEEE Transactions on Nanotechnology*, vol. 4, no. 2, pp. 215-228, March 2005.

[109] P. J. Kuekes, W. Robinett and R. Stanley Williams, "Improved voltage margins using linear error-correction codes in resistor-logic demultiplexers for nanoelectronics," *Nanotechnology,* vol. 16, pp. 1419-1432, 2005.

[110] S. Lin, D. Costello, *Error Control Coding,* 2nd Edition, Prentice-Hall, Inc. Upper Saddle River, NJ, 2004.

[111] H. Imai, *Essentials of Error-Control Coding Techniques,* Academic Press Inc, New York, 1990.

[112] J. Han, and P. Jonker, "A defect and fault-tolerant architecture for nanocomputers," *Nanotechnology,* vol. 14, pp. 224-230, 2003.

[113] M. Ziegler, C. Picconatto, J. Ellenbogen, A. Dehon, D. Wang, Z. Zhong and C. Lieber *"*Scalability simulations for nanomemory systems integrated on the molecular scale,*" Ann N.Y. Acad. Sci,* vol. 1006, pp. 312-330, 2003.

[114] Y-H Choi and Y. K. Kim "A nanoscale scalable memory architecture for molecular architecture electronics," *Nanotechnology*, vol. 15, no. 10, pp. S639-S644(1), October 2004.

[115] C. Amsinck, N. Di Spigna, S. Sonkusale, D. Nackashi, and P. Franzon, "Scaling constraints in nanoelectronic random access memories," *Nanotechnology,* vol. 16 pp. 2251-2260, 2005.

[116] K. Hirose, and H. Yasuura, "A bus delay reduction technique considering crosstalk," in *Proceedings of DATE*, France, pp. 441-445, March 2000.

[117] P. J. Burke, "Luttinger liquid theory as a model of the gigahertz electrical properties of carbon Nanotubes," *IEEE Transactions on Nanotechnology*, vol. 1, pp. 129-144, 2002.

[118] R. Boylestad, *Introductory Circuit Analysis*, Prentice-Hall, Inc, Upper Saddle River, NJ, pp. 317-20, 1997.

[119] Cosmoscope (TM) Version 3.0, Copyright © 1986-2004, Synopsys Inc.

[120] S. Salahuddin, M. Lunstrom, and S. Datta, "Transport effects on the signal propagations in quantum wires," *IEEE Transactions on Electron Devices*, vol. 52, no. 8, pp. 1734-1742, August 2005.

[121] A. Naeemi, R. Sarvari, J. D. Meindl, "Performance comparison between Carbon nanotube and copper interconnects for gigascale integration (GSI)," *IEEE Electron Device letters*, vol. 26, no. 2, February 2005.

[122] A. Raychowdhury and K. Roy "Modeling and Analysis of Carbon Nanotube Interconnects and their Effectiveness for High Speed VLSI Design," in *Proceedings of IEEE/ACM International Conference on Computer Aided Design, San Jose, USA,* pp. 237–240, November 2004.

[123] C. T. White and T. N. Todorov, “Nanotubes go ballistic,” *Nature*, vol. 411, pp. 649-651, June 2001.

[124] H. Stahl, J. Appenzeller, R. Martel, P. Avouris, and B. Lengeler, “Intertube coupling in ropes of single-wall carbon nanotubes,” *Phys. Rev. Lett.,* vol. 85, pp. 5186-5189, 2000.

[125] M. Bockrath, D. H. Cobden, P. L. McEuen, N. G. Chopra, A. Zettl, A. Tess, and R. E. Smalley, "Single electron tunneling transport in ropes of carbon nanotubes," *Science*, vol. 275, pp. 1922-1925, March 1997.

[126] S. Li, Z. Yu, C. Rutherglen, and P. J. Burke, “Electrical properties of 0.4 cm long single walled carbon nanotubes,” *Nano Lett.,* vol. 4, pp 2003-2007, 2004.

[127] J. Y. Park, S. Rosenbelt, Y. Yaish, V. Sazonova, H. Ustunel, S. Braig, T. A. Arias, and P. L. McEuen, “Electron-phonon scattering in metallic single-walled carbon nanotubes,” *Nano Lett.*, vol. 4, pp. 517-520, 2004.

[128] International Technology Roadmap for Semiconductors. [Online]. Available: http://public.itrs.net/. *Accessed in 2006*.

[129] The International Technology Roadmap for Semiconductors (ITRS): process integration, devices, and structures. (Semiconductor Industry Association, San Jose, California, 2005). [Online]. Available: http://www.itrs.net/reports.html. *Accessed in 2007*.

[130] P. J. Burke, "AC performance of nanoelectronics: towards a ballistic THz nanotube transistor," *Solid-State Electronics*, vol. 48, pp. 1981–1986, 2004.

[131] Saber® Sketch Version 3.2 Copyright © 1986-2004, Synopsys Inc.

[132] W. Wu, G.-Y. Jung, G. Jung, D. L. A. Olynick, J. Straznicky, Z. Li, X. Li , D. A. A. Ohlberg, Y. Chen, S –Y Wang, J. A. Liddle, W. M. Tong, R. Stanley Williams, "One-kilobit crossbar molecular memory circuits at 30nm half-pitch

fabricated by nanoimprint lithography," *Appl. Phys. A.* vol. 80, pp. 1173-1178, 2005.

[133] P. Kapur, J. P. McVittie and K. C. Saraswat, "Technology and reliability constrained future copper interconnects-Part II: Performance Implications," *IEEE Trans. Electron Devices,* vol. 49, no. 4, April 2002.

[134] M. Kwiatkowska, G. Norman, and D. Parker "PRISM: probabilistic symbolic model checker," *in TOOLS 2002, LNCS,* vol. 2324, pp. 200-204, (Springer-Verlag, April 2002).

[135] [Online]. Available: www.cs.bham.ac.uk/~dxp/prism/. *Accessed June 2006*.

[136] A. Coker, V. Taylor, "Performance analysis of a fault-tolerant crossbar molecular switch memory demultiplexer," in *Proceedings of IEEE Region 5 Conference*, April 7, 2006.

VITA

Name: Ayodeji O. Coker
Address: Department of Computer Science, H. R. Bright Building, 3112 TAMU, College Station, TX 77843-3142.
Email: cdeji@yahoo.com

EDUCATION

TEXAS A&M UNIVERSITY, College Station, TX
Department of Computer Science
PhD in Computer Engineering, May 2008.
Nanotechnology: Performance Analysis of Fault-Tolerant Nanoelectronic Memories.

NORTHWESTERN UNIVERSITY, Evanston, IL
Department of Computer and Electrical Engineering
Master of Science in Electrical Engineering (Optical Communications), June 2002.
Master's Project Topic: Dispersion Characterization of Holey optical fibers.

STATE UNIVERSITY OF NEW YORK AT ALBANY, Albany, NY
Bachelor of Science in Physics. Minor in Mathematics, May 1999

PUBLICATIONS

- A. Coker, V. Taylor, D. Bhaduri, S. Shukla, A. Raychowdhury, and K. Roy, "Multi-junction fault-tolerance Architecture for Nanoscale Crossbar Memories," *IEEE Transactions on Nanotechnology*, vol. 7, no. 2, pp. 202-208, March 2008.
- A. Coker, V. Taylor, "Performance and reliability analysis of a scaled multi-switch junction crossbar nanomemory and demultiplexer," In *Proceedings of 7th IEEE Conference on Nanotechnology*, Aug 2-5, 2007.
- A. Coker, V. Taylor, "Performance analysis of a scaled multi- junction molecular switch crossbar nanomemory with ECC enhanced demultiplexer," *Design Automation and Test in Europe*, Nice, France, April 20, 2007, Invited Talk.
- D. Bhaduri, D. Coker, S. Shukla, V. Taylor, P. Graham, M. Gokhale "A hybrid framework for design and analysis of fault-tolerant architectures for nanoscale molecular crossbar memories," in *Proceedings Publication and Interactive Presentation of Design Automation and Test in Europe*, Munich, Germany, March 2006.
- M. Fiorentino, J. E. Sharping, A. Coker, and P. Kumar, *Fiber generated quantum correlations for quantum-optical communications,* in Coherence and Quantum Optics VIII, N. P. Bigelow, J. H. Eberly, C. R. Stroud Jr., and I. A. Walmsley, Eds., (Kluwer/Plenum, New York, 2003), pp. 185-186.
- J. E. Sharping, A. Coker, M. Fiorentino, P. Kumar, and R. S. Windeler, "Four-wave mixing in Microstructure Fiber," in *CLEO '01 Technical Digest*, pp. 508, 2001.