Probabilistic analysis of defect tolerance in asynchronous nano crossbar architecture

Shikha Chaudhary

PROBABILISTIC ANALYSIS OF DEFECT TOLERANCE IN ASYNCHRONOUS NANO CROSSBAR ARCHITECTURE

by

SHIKHA CHAUDHARY

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN COMPUTER ENGINEERING

2008

Approved by:

Minsu Choi, Advisor

Theodore McCracken

Sahra Sedigh

PUBLICATION THESIS OPTION

This thesis consists of two articles that have been published or submitted for publication as follows:

The first paper presented in pages 2-15 entitled "Clock-Free Nanowire Crossbar Architecture based on Null Conventional Logic (NCL)", was published in the PROCEEDINGS OF IEEE INTERNATIONAL CONFERENCE ON NANOTECHNOLOGY, 2007.

The second paper presented in pages 16 to 35 entitled "Probabilistic Analysis of Design Mapping in Asynchronous Nanowire Crossbar Architecture" is submitted to 2009 IEEE INTERNATIONAL INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE (I2MTC).

ABSTRACT

Among recent advancements in technology, nanotechnology is particularly promising. Most researchers have begun to focus their efforts on developing nano scale circuits. Nano scale devices such as carbon nano tubes (CNT) and silicon nanowires (SiNW) form the primitive building blocks of many nano scale logic devices and recently developed computing architecture. One of the most promising nanotechnologies is crossbar-based architecture, a two-dimensional nanoarray, formed by the intersection of two orthogonal sets of parallel and uniformly-spaced CNTs or SiNWs. Nanowire crossbars offer the potential for ultra-high density, which has never been achieved by photolithography. In an effort to improve these circuits, our research group proposed a new Null Convention Logic (NCL) based clock-less crossbar architecture. By eliminating the clock, this architecture makes possible a still higher density in reconfigurable systems. Defect density, however, is directly proportional to the density of nanowires in the architecture. Future work, therefore, must improve the defect tolerance of these asynchronous structures.

The thesis comprises two papers. The first introduces asynchronous crossbar architecture and concludes with the validation of mapping a 1-bit adder on it. It also discusses various advantages of asynchronous crossbar architecture over clock based nano structures.

The second paper concentrates on the probabilistic analysis of asynchronous nano crossbar architecture to address the high defect rates in these structures. It analyzes the probability distribution of mapping functions over the structure for varying number of defects and proposes a method to increase the probability of successful mapping.

ACKNOWLEDGMENTS

It gives me immense pleasure to thank all the people who made this thesis possible. I would like to thank my advisor, Dr. Minsu Choi, for his support and encouragement which have helped me throughout my degree. It has been a pleasure working with him all these years.

I express my sincere gratitude to my committee member, Dr. Ted McCracken, with whom I enjoyed taking three courses. His knowledge and encouragement taught me a lot about Computer Architecture and Embedded Systems. I would also like to thank Dr. Sahra Sedigh, another committee member, who was the coordinator of my TA lab. She was very understanding and helpful, and I enjoyed working under her. I thank Dr. Shoukat Ali for his course in Computer Architecture and Dr. Scott Smith for his course in Digital Logic. I would also like to thank Ms. Regina Kohout for helping me with the paperwork and departmental guidelines. I am grateful to Dr. Kurt Kosbar for providing me a position as Teaching Assistant in the department.

I would like to thank Yadunandana Yellambalase and Ravi Bonam for working with me in this research.

Special thanks to my friends Amith Shetty, Poorna Marappa, Mandar Joshi, Alankar Sharma, Kavish Goel, Mohit Chopra, and Anshuman Singh for supporting me. I also thank my friends Arish, Anand, Jai, Tarang, Manoj, and Dheeraj for the fun environment they created.

Most importantly, I thank my parents, Jeetendra and Kamla Singh, and my brother Vishal for their love and continuous support that made me strive for my dreams.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

PAPER I

PAPER II

LIST OF TABLES

Table Page

PAPER II

1. INTRODUCTION

Since the inception of the idea of nanotechnology, the area of interest of researchers has shifted from CMOS circuits to find ways for the improvement of nano structures. The nano crossbar architectures formed by carbon nano tubes (CNT) or silicon nanowires (SiNW) are amongst the most important of these improvements. Much work has been done to improve the defect tolerance of crossbars since bottom up fabrication technology yields architectures with defect densities of 10% or higher.

The new asynchronous structure proposed by our research group, works on the principle of Null Convention Logic (NCL) rather than the clock to synchronize its operation. NCL is an asynchronous logic paradigm that works on the principle of local handshaking by integrating data and control signals into a single signal. The need for a clock is thus eliminated along with many problems related to the clock in conventional circuits, including delay sensitivity and space overhead.

This thesis highlights the opportunities offered by asynchronous architectures. These circuits have high nanowire density. The delay insensitive nature of NCL also makes them faster. These architectures do present challenges; however that have not been experienced with the conventional circuits.

The first paper describes in detail the proposed architecture, which uses Programmable Gate Macro Block (PGMB) as the building block on which all the NCL gates are mapped. It concludes with illustrations of the architecture, including the design of a full bit adder. The second paper discusses the distribution of mapping probability for various numbers of defects in the PGMB. Mathematical analysis provides a basis for improving programmability.

Paper I

CLOCK-FREE NANOWIRE CROSSBAR ARCHITECTURE BASED ON NULL CONVENTIONAL LOGIC (NCL)

Ravi Bonam, Shikha Chaudhary, Yadunandana Yellambalase and Minsu Choi
Dept of ECE, University of Missouri-Rolla, MO, USA

Abstract—**There have been numerous nanowire crossbar architectures proposed to date, although all are envisioned to be synchronous (i.e., clocked). The clock is an important part of a circuit, and it must to be connected to all the components to synchronize their operation. Considering the nondeterministic nature of nano scale integration, realizing the functions on a nanowire crossbar system would be quite cumbersome. This paper proposes a new clock-free crossbar architecture to resolve the issues with clocked counterparts. This architecture is implementing with a delay-insensitive logic encoding technique called Null Convention Logic (NCL). A delay-insensitive full adder has been implemented on the proposed architecture to demonstrate its feasibility.**

Index Terms — **Nanowire crossbar, Asynchronous computing, Null conventional logic (NCL), Manufacturability, Robustness, Scalability, Defect and Fault-tolerance.**

1. INTRODUCTION

The end of photolithography as the driver for Moore's Law is predicted within seven to twelve years, and emerging nanotechnologies are expected to continue the technological revolution. Recently, numerous nanoscale logic devices have been proposed based on nanoscale components such as carbon nanotubes (CNTs) and silicon nanowires (SiNWs); computing architectures are also being proposed using them as primitive building blocks. One of the most promising nanotechnologies is crossbar based architecture; a two-dimensional array (i.e., nanoarray) formed by the intersection of two orthogonal sets of parallel and uniformly-spaced nanometer-sized wires, such as CNTs and SiNWs. Experiments have shown that such wires can be aligned to construct an array with nanometer-scale spacing using a form of directed self-assembly. The crosspoints of nano scale wires can be used as programmable diodes, memory cells or Field-Effect Transistors (FETs), making nano scale logic devices realizable. Currently, the nanowire crossbars are either proposed to be or are used in a variety of applications, e.g. in bioelectric systems [1] and for digital circuits [2]. New techniques are being proposed for its fabrication [3] also.

Nanowire crossbars offer both an opportunity and a challenge. They would make possible an ultra-high density never achieved by photolithography. High-density systems consisting of nanometer-scale elements assembled in a bottom-up manner are likely to have many imperfections (raw fabrication defect densities, as high as 10%, are expected [4, 5]) and parametric variations. The challenge, therefore, is to make them simple enough for manufacturing and reliable enough for use in everyday computing applications. A computing system designed on a conventional design basis and top-down lithographic manufacturing would not be practical. Ultra-high density fabrication could be very inexpensive if researchers can actualize a chemical self- assembly; however such a circuit would require laborious testing, repair, and reconfiguration processes, implying significant overhead costs. Also, all reconfigurable computing architectures based on nanowire crossbars are commonly envisioned to be used for synchronous circuits and systems. Thus, a clock distribution network must be fabricated along with nanowire crossbars and precise timing control should be practiced to avoid all timing-related faults induced by physical design parameter variations resulting from nano scale non-deterministic assembly.

In order to be a viable nanotechnology, the nanowire crossbar based systems should be:

1. Structurally simple and scalable enough to be fabricated by bottom-up manufacturing techniques
2. Robust enough to tolerate extreme parametric variations

3. Defect and fault-tolerant enough to overcome extreme defect densities, aging factors, and transient faults
4. Able to support at-speed verification and reconfiguration

Addressing all these issues, this research proposes a new asynchronous architecture for carbon nanotube and silicon nanowire based reconfigurable nano computing systems as an alternative to conventional clocked counterparts.

The proposed asynchronous nano-architecture is based on a delay-insensitive data encoding and self-timed logic encoding scheme. No clock distribution network is needed, therefore, and all timing-related failure modes are also eliminated. Potential benefits from the proposed asynchronous architecture include enhanced manufacturability, scalability, robustness, and defect and fault tolerance.

2. PRELIMINARIES AND REVIEW

2.1 Null Convention Logic

Most traditional Boolean circuits that we have been using are clock driven. The clock is one of the most important parts of the circuit determining its speed and performance. All devices in a circuit have must be connected to the clock, creating a cumbersome network. Traditional Boolean circuits do not check for input completion when evaluating an expression. That is, they do not confirm that all inputs have arrived before beginning computation of an expression. Since they are dependent on the clock, traditional Boolean circuits are symbolically incomplete in terms of evaluating expressions. Null Conventional Logic (NCL) integrates data and control into a single signal, yielding circuits and systems that are inherently clockless and delay insensitive [8]. This technology uses two states, DATA and NULL, for synchronizing and I/O control. DATA wavefront contains the data to be processed by the combinational circuit. The NULL wavefront is a non-data value used to reset the logic gates in the circuit and is also used as a delimiter between two DATA wavefronts [8]. Circuits communicate with each other using local handshakes that provide synchronization. The concept of a global clock is eliminated, which in turn eliminates the clock network. The removal of the clock reduces power consumption and the circuit becomes data driven (i.e. data is processed as soon as it is available). In the DATA combinational evaluation period, the combinational circuitry processes the data passed on by the register, and the results are stored in the successive register. The successive register generates the request for NULL signal in the DATA Completion Acknowledgement period and propagates the signal to the previous register. The previous register will then transfer to the combinational circuitry a NULL, which is evaluated during the NULL combinational evaluation period.

The evaluated result is passed to the successive register, which then generates a Request for DATA signal. If the output of a gate is NULL, that output does not change until all inputs to the gate are DATA. When all inputs receive DATA, then the output changes to DATA and remains asserted as long as all the inputs do not change to NULL. This attribute of the threshold gates helps facilitates the input completeness feature, enabling the circuits to function without a clock [10]. To achieve input completeness, the inputs to the gates must be encoded using an encoding scheme. In a dual rail encoding scheme, each bit is represented with two rails. According to the representation in Table 1, the combination of rails (rail1, rails0) represents a single Boolean value. The value "00" is regarded as NULL state, which resets the circuit and does not represent any Boolean value. The value "11" is an undefined expression in the dual rail encoding scheme. NCL uses symbolic completeness [14] of expression to achieve self-timed behavior. A symbolically complete expression is defined as an expression that depends only on the relationships of the symbols present in the expression without reference to the time of evaluation. Symbolic completeness depends on the following conditions [14]:

1. The input-completeness criterion, which NCL circuits must maintain in order to be self-timed, requires that the outputs of a circuit may not transition from NULL to DATA until all inputs have transitioned from NULL to DATA, or vice versa.
2. In circuits with multiple outputs, those outputs that are dependent on arrived inputs can make transition, but all outputs can change only when all inputs arrive, which eliminates the possibility of a data cycle and null cycle overlapping.

3. No orphans may propagate through a gate. An orphan is defined as a wire that transitions during the current DATA wavefront, but is not used in the determination of the output. Orphans are caused by wire forks and can be neglected through the isochronic fork assumption, as long as they are not allowed to cross a gate boundary. This observability condition ensures that every gate transition is observable at the output.

3. PROPOSED ARCHITECTURE

3.1 The New Architecture: Asynchronous Crossbar Architecture

In this paper we are going to implement Null Conventional Logic on nanowire crossbar architecture to realize "Asynchronous Crossbar Architecture". The primary advantages of NCL for the proposed clock-free nano-architecture are as follows:

1. Larger, less complex circuits can be designed in a bottom-up manner and integrated directly without the need to synchronize each module [8].
2. In clock-driven circuits, the majority of power is consumed by the clock and its network. By removing the clock from the circuit, cumulative power consumption decreases [8].
3. The use of NCL makes the circuit insensitive to delay, and the circuits operate at the rate of the flow of data. The circuits can be described as delay-insensitive and self-timed [5, 7].
4. Problems associated with the clock, such as clock skew, race conditions, etc. are eliminated, making circuits more reliable [8].

Twenty seven threshold gate macros are implemented in NCL. These gates permit implementation of any possible expression involving two, three, or four variables. Inversion can be implemented by interchanging rail1 and rail0 in case of the dual rail encoding scheme.

3.2 Advantages of Asynchronous Crossbar Architecture

Normal crossbar architecture is similar to conventional Boolean circuits in that the clock must be circulated throughout the circuit to synchronize various blocks. The normal crossbar circuit cannot decide when to receive or release data; therefore, a clock must be added to control the flow of input and output. In contrast, the asynchronous crossbar architecture is data driven; instructions are acted upon the moment they are available, and output is available the moment it is completed. This architecture employs discrete threshold gates [8] that recognize only certain simultaneous combinations of values. Each of the gate acts as a synchronization node, making the circuit as a whole and symbolically complete. The DATA state follows the NULL state. It is processed by the gates and output is passed on to a register. The register contains completion circuitry that enables synchronization and checks the state of the output and generates an appropriate signal indicating the previous register to send the complementary state. That is, if the circuit is processing a NULL state then when the output arrives, register will send a request for data signal requesting for data to the previous register. The primary advantages of the asynchronous architecture are as follows.

1. Manufacturability

Asynchronous crossbar Architecture significantly increases the manufacturability of the nanowire crossbar systems in large scale manufacture. Such circuits are easier to manufacture than their clocked counterparts. Clocked synchronous architectures are difficult to map on crossbar architectures since they require complex placement and routing algorithms. Asynchronous crossbar architecture, however, permits mapping of gates onto discrete blocks of crossbars, eliminating the need of a global synchronous signal to coordinate all the blocks. All clock related hardware components can be removed from the overall hardware design making the circuits less complex and easier to design.

2. Scalability

The overall circuit is self-timed i.e. timing information is integrated with data in the encoding. Since the timing of each circuit is handled locally, scalability of these circuits is higher. Although the size of the circuit increases, timing complexity does not. Time required for any particular computation does not change due to the increased circuit size.

3. Robustness

Due to the non-determinism of the directed self-assembly paradigm, nanowire crossbar circuits are expected to exhibit large variations in physical parameters. Since any physical variation in an electrical parameter may have a negative effect on the timing behavior of the circuit, the ability to design delay-insensitive circuits (i.e., circuits that operate correctly is independent of timing) is important. This capacity greatly increases the robustness of the circuit to design parameter variations. As noted above, asynchronous crossbar architecture eliminates delays in processing data due to clock cycles. Instead data is processed as and when it is available.

4. Defect and Fault Tolerance

Since NCL circuits have a definite flow pattern (i.e., DATA or NULL and vice versa) the output can be identified as a data or null. Not only are all timing-related failure modes eliminated, but testing complexity is reduced. In particular, stuck-at-1 faults simply halt the circuit, since the NCL circuit cannot make a transition from DATA to NULL. Also, in dual-rail encoding, 11 is considered an invalid code. Therefore, any permanent or transient fault resulting in 11 can be eventually detected. Only stuck-at-0 faults and a few other transient faults need to be exercised with applied patterns. Design time and risk as well as circuit testing requirements are decreased because of the elimination of the clock with its complexity and critical timing issues.

3.3 Programmable Gate Macro Block

The basic unit of the proposed architecture is a programmable gate macro block (PGMB). Each block is made of an AND plane and an OR plane formed by the diode

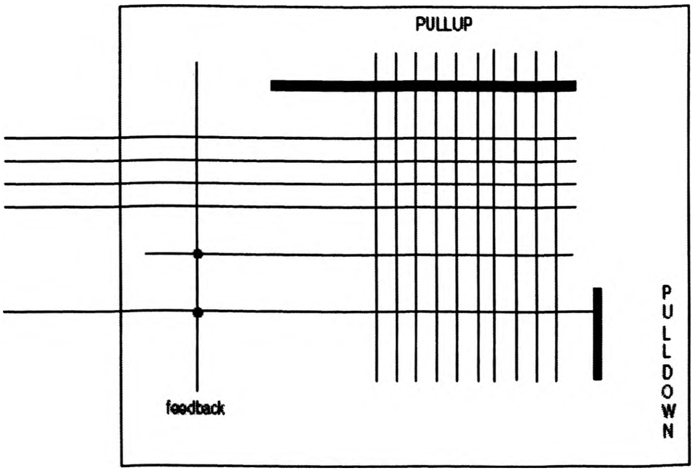
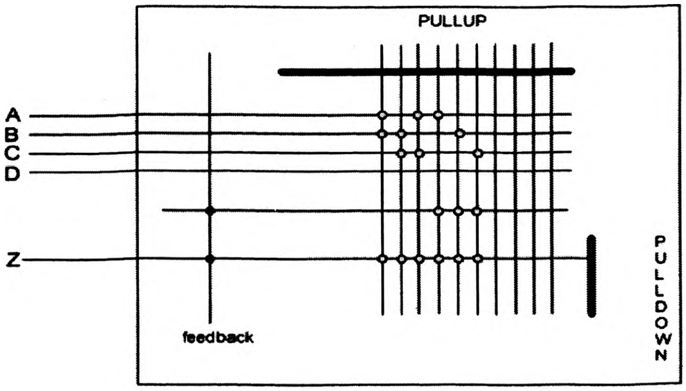


Figure 1 Basic structure of PGMB



TH23 realized on PGMB

Figure 2 TH23 realized on a PGMB

crossbars. Vertical nanowires with pull up resistors form product terms and horizontal wires with pull down resistor add them using OR logic. Each block also has a feedback loop that drives the output back to an input wire. The maximum number of inputs to any threshold gate is four. A feedback is required to implement any of the 27 threshold gates

[10]. Figure 1 shows the basic structure of a Programmable Gate Macro Block. It is a 6x10 crossbar structure which can take a maximum of 4 inputs as illustrated. Figure 2 shows the implementation of TH23 gate in the programmable gate macro block. The output of the TH23 gate is given by the logic Z = AB+BC+CA + (A+B+C)Z*, where Z* is the previous output of the TH23 gate, which is fed back to an input nanowire.

3.4 Physical Structure

The new architecture consists of array of PGMBs that are interconnected in a 2D grid structure. These blocks are surrounded by nanowires that are used to route the signals inside the grid structure. The PGMB's input and output nanowires cross the routing wires forming programmable crosspoints. By programming these crosspoints, signals can be routed to any of the programmable gate macro blocks. The input stage consists of programmable resistor crosspoints formed by the micro wires and nanowires. By programming relevant crosspoints, signals can be routed to the required PGMB. Each block can in turn be programmed to implement any of the threshold gates [10]. These blocks can tap the input signals by programming both a corresponding crosspoint which is formed by the nanowire column carrying the input signal, and a nanowire row, which is input to the macro block. The output of the implemented threshold gate [10] can be routed to the other gates in a similar fashion. Thus the number of columns of nanowires between programmable macro blocks determines the number of crosspoints available for routing signals. This number has to be sufficient to route all the required inputs and outputs to the macro blocks. The number of rows and columns of PGMBs in the grid is limited by the amount of signal degradation caused by propagation. Before complete degradation of the signal, a buffering stage can be implemented to restore signal strength. We show the implementation of a full adder using the new crossbar architecture and discuss feasibility of a multi-bit adder.

4. IMPLEMENTATION OF ONE BIT FULL ADDER

A full adder can be implemented using threshold gates, as shown in Figure 3. The proposed architecture will implement a 1 bit full adder by using two TH23 gates and two

TH34w2 gates, as shown in the Figure 3. This implementation requires 3 input bits: a and b for addition and c as the carry bit, all encoded in dual rail logic. These bits are represented by a0, a1, b0, b1, c0, and c1. By programming required crosspoints at the input crossbar, these signals are routed to the programmable gates. Complete implementation of the 1 bit full adder is shown in Figure 5.

The blocks in row 1 and columns 1, 2 are programmed as TH23 gates and blocks in row 2 and columns 1, 2 are programmed as th34w2 gates. The TH23 gates require three inputs, leaving one input row unused, where as in TH34w2 all the 4 input rows are used. The threshold gates realized on PGMB are shown in the Figures 1 and 4. The required signal is then routed to the corresponding input rows. Outputs from the threshold gates are also routed either to the input of other gates or to the output block, by programming routing crosspoints and using free nanowires.

The NCL register stage consists of two TH22 gates and a single TH12 gate that are used to generate a handshaking signal that will synchronize the circuit. Two kinds of signals, request for data and request for null, are generated by the registers and passed on to the previous register. Input from successive stage (Ki) and output to previous stage (Ko) are the handshaking signals. The input data rails are labeled as Do, D1 and Q0, Q1 are the output rails. The single bit register stage is shown in Figure 6.

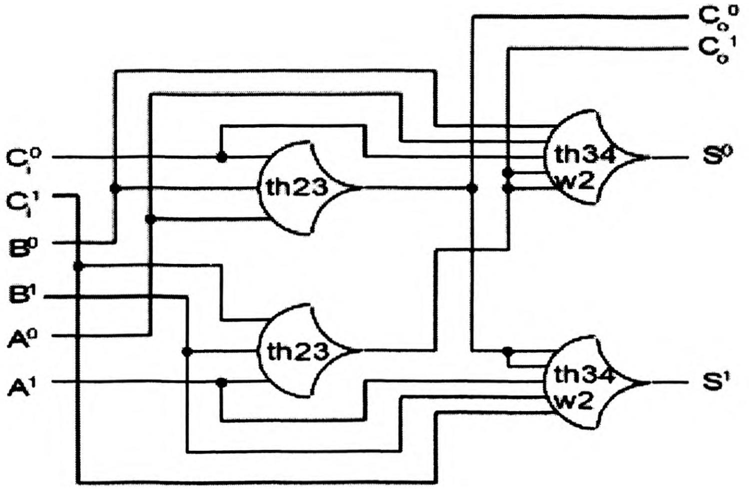


Figure 3 1-bit adder in NCL

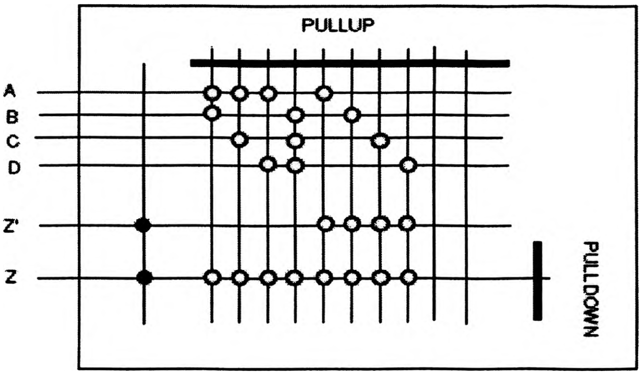


Figure 4 TH34W2 realized on PGMB

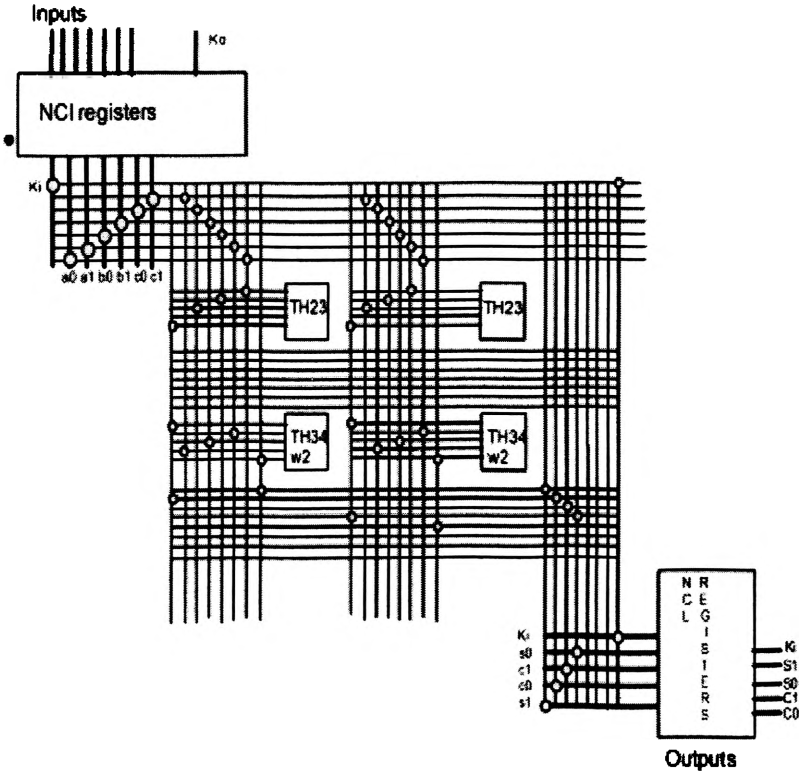


Figure 5 1-bit adder using proposed architecture

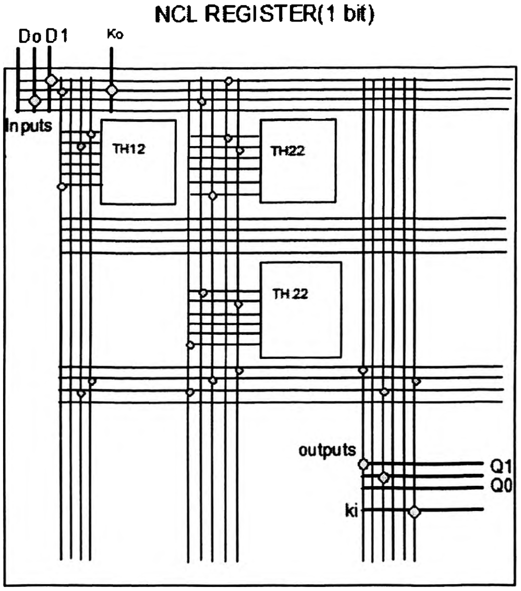


Figure 6 NCL one bit register on proposed architecture

5. CONCLUSIONS

This paper proposes a new clock-free nanowire crossbar architecture based on delay-insensitive logic known as Null Convention Logic, or NCL. The complex clock distribution network can be removed from the hardware, thus eliminating many clock related failure modes. To demonstrate the feasibility of this architecture, a delay-insensitive full adder design has been implemented on it. Future work will develop automated design optimization tools, testing schemes, and defect-tolerant logic mapping techniques for the proposed architecture.

6. REFERENCES

[1] M.-W. Shao, Y.-Y. Shan, N.-B. Wong, S.-T. Lee, "Silicon nanowire sensors for bioanalytical applications: Glucose and Hydrogen Peroxide detection", Wiley Interscience, Volume 15 Issue 9, Pages 1478 – 1482, 2005.

[2] Snider, Greg, "Molecular junction nanowire crossbar based inverter, latch, and flip-flop circuits, and more complex circuits composed, in part, from molecular junction nanowire crossbar based inverter, latch, and flip-flop circuits", United States Patent 6919740, 2005.

[3] Wessels, Jurina, Ford William E., Yasuda, Akio, "Method for preparing a nanowire crossbar structure and use of a structure prepared by this method", United States Patent 7276172, 2007.

[4] J. Huang, M. B. Tahoori and F. Lombardi, "On the defect tolerance of nano-scale two-dimensional crossbars", IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, pp. 96-104, Oct 2004.

[5] M. Jacome, C. He, G. de Veciana, and S. Bijansky, "Defect tolerant probabilistic design paradigm for nanotechnologies", IEEE/ACM Design Automation Conference (DAC), pp. 1-6, 2004.

[6] Jiangtao Hu et. al., "Chemistry and Physics in One Dimension : Synthesis and properties of Nanotubes and Nanowires," Acc. Chem. Res. , Vol. 32, pp. 435-445, 1999.

[7] Matthew M. Ziegler and Mircea R.Stan, "Design and analysis of Crossbar Circuits for Molecular Nanoelectronics," IEEE Nanotechnology Conference, pp. 323-327, 2002.

[8] Karl M. Fant, Scott A. Brandt, "NULL Convention Logic : A Complete and Consistent Logic for Asynchronous Digital Circuit Synthesis," IEEE International Conference on Application-Specific Systems, Architectures and Processors, pp. 261-273, 1996.

[9] R. Smith and M. Ligthart, "High-Level Design for Asynchronous Logic," Design Automation Conference, pp. 431-436, 2001.

[10] S. C. Smith, R. F. DeMara, J. S. Yuan, D. Ferguson, and D. Lamb, "Optimization of NULL Convention Self-Timed Circuits," Integration, The VLSI Journal, Vol. 37, No. 3, pp. 135-165, 2004.

[11] D. Whang, S. Jin and C. M. Lieber, "Large-Scale Hierarchical Organization of Nanowires for Functional Nanosystems," Japanese Journal of Applied Physics, Vol. 43, No. 7B, 2004.

[12] Y. Cui and C. M. Lieber, "Functional nanoscale electronic devices assembled using silicon nanowire building blocks," Science, Vol. 291, pp. 851-853, 2001.

[13] Nicolas A. Melosh, Akram Boukai, Frederic Diana, Brian Geradot, Antonio Badolato, Pierre M. Petroff, James R. Health, "Ultrahigh-Density Nanowire Lattices and Circuits," Science, Vol. 300, pp. 112-115, 2003.

[14] S. Smith, R. DeMara, J. Yuan, M. Hagedorn and D. Ferguson, "Delay- Insensitive gate-level pipelining," Integration, the VLSI journal, Vol. 30,
pp. 103-131, 2000.

Paper II

PROBABILISITC ANALYSIS OF DESIGN MAPPING IN ASYNCHRONOUS NANOWIRE CROSSBAR ARCHITECTURE

Shikha Chaudhary and Minsu Choi
Dept of ECE, University of Missouri-Rolla, Rolla, MO 65401, USA
{sc8tc, choim}@umr.edu

Abstract

Recent publications have introduced the concept of NCL into nanotechnology, resulting in the removal of the clock circuit overhead from the crossbar architecture, making possible a higher density in reconfigurable systems. Defect density, however, is directly proportional to the density of nanowires in the architecture. This work examines a number of ways to avoid defects while mapping functions. Since nano crossbar architecture has many defects due to manufacturing constraints and its extremely small size, a more practical approach is to route the ON crosspoints away from the defects. This approach analyzes quantitatively the variations in mapping probabilities with factors such as defect rate, size of crossbar matrix, and type of threshold gate, to achieve an optimal design.

1. INTRODUCTION

With advancements in nanotechnology, most researchers have begun to focus their efforts on the development of nano scale circuits. Nanowires are one dimensional structures which exhibit interesting electrical properties. A nano crossbar architecture is a two dimensional array of intersecting sets of orthogonal nanowires, which can be programmed electronically to exhibit properties of various active and passive devices [1]. Depending on doping concentration and the alignment of the nanowires, the crosspoints can exhibit properties of a conventional diode, a Field Effect Transistor (FET), or a resistor [1, 2]. This work used diode crossbar architecture, which can realize AND-OR logic functions [3]. These AND-OR logic planes can be cascaded in the form of logic tiles to realize complex functions. A Programmable Gate Macro Block (PGMB) is a nanowire crossbar matrix with a discrete number of rows and columns on which the functions can be programmed. Figure 1 shows a PGMB. The vertical wires with pull up resistors form the product terms plane and the horizontal wires with pull down resistors add them together. It has a feedback wire to provide the current output at the input.

A 6x10 defect-free crossbar can be used to program any of the 27 threshold gates (described in section 2.2). The crosspoints can be programmed as ON or OFF by applying a voltage to decrease or increase the distance between the two orthogonal nanowires. Thus the ON crosspoints form a diode junction, while the OFF crosspoints offer a high resistance. The current bottom up assembly will lead to many imperfections which are an unavoidable aspect of any nano crossbar architecture.

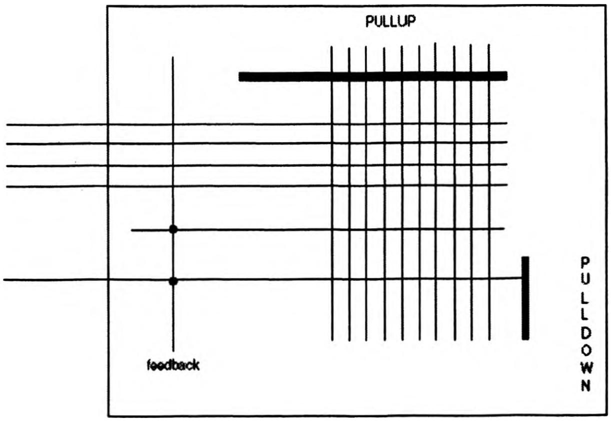


Figure 1 Programmable Gate Macro Block

The PGMB may contain the number of defective crosspoints that cannot be programmed as a "closed" junction. When a threshold (TH) gate macro is mapped onto the PGMB, therefore, zero or more ON-inputs may coincide with these defects. Instead of trying to reduce the number of defects, this work takes the practical approach of routing away from them. The most challenging step in improving defect tolerance is to find an optimum mapping technique to avoid defects. Mapping requires knowledge of the exact distribution of probabilities with a variable number of coinciding defects. This work relies on the defect unaware approach to generate the probability distribution and thus to successfully map the gate on the defective PGMB. The factors that affect the mapping probability of a gate are the defect rate (i.e., the average number of defects in the crossbar), the type of the TH gate, and the size of the crossbar matrix.

2. PRELIMINARIES AND REVIEW

2.1 Conventional Clocked Nanowire Crossbar Architecture

The clock is one of the most important parts of a clocked circuit, and it determines the speed and performance of the circuit. All devices in a circuit must be connected to the clock; therefore, the clock network is cumbersome. Traditional Boolean circuits do not check for input completion when evaluating an expression. That is, they do not confirm that all inputs have arrived before beginning computation of the expression. Traditional Boolean circuits, then, are symbolically incomplete in terms of evaluating expressions since they are dependent on the clock. One of the major disadvantages of clocked architectures is that the clock time, and thus the total circuit time depends on the worst case delay. Combinational logic blocks bracketed by registers store the current state results. The data to be latched into the register should be present before a certain time called set-up time, t_{su}, ahead of the triggering clock edge. Similarly hold time, t_{hold}, is the time for which the input to a register should remain constant. The combinational delay, t_{comb}, is the delay that occurs inside the combinational logic. The total time for the system can be represented as $T = t_{su} + t_{hold} + t_{comb}$ [4]. This delay depends on the worst case path for the whole system. Since the worst case delay applies to faster logic, it retards the whole system. Clockless circuits offer various advantages over clocked circuits; these are discussed below.

2.2 A New Approach: Clockless Crossbar Architecture [5]

This author's research group has recently proposed a new clock-free architecture that circumvents many issues associated with conventional clocked nanowire crossbar systems. This architecture is based on an asynchronous logic paradigm known as Null Convention Logic (NCL) [6], which works on the principle of logic/control encoding and handshaking. It integrates data and control (i.e., handshaking) into a single signal, thus providing inherently clockless delay-insensitive operation [6]. Two states DATA and NULL, synchronize functioning. The DATA wavefront contains the binary data (i.e., either 0 or 1) that is processed by the combinational circuit. The NULL wavefront which tells the circuit that new data will be coming in, is a non data value used to reset the logic block. It separates the two DATA wavefronts. As soon as the DATA or NULL is available to the register, it provides the handshaking signal and requests the next DATA or NULL. The global clock is thus eliminated, which reduces power consumption, and the circuit becomes data driven (i.e., data is processed as soon as it is available). This complete elimination of the clock distribution network and clock-related failure modes is crucial to the proposed asynchronous nanowire crossbar architecture.

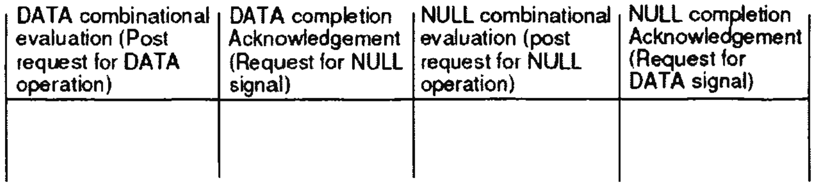

DATA combinational evaluation (Post request for DATA operation)
DATA completion Acknowledgement (Request for NULL signal)
NULL combinational evaluation (post request for NULL operation)
NULL completion Acknowledgement (Request for DATA signal)

Figure 2 NCL Timing Diagram [7]

The DATA to DATA cycle timing diagram is shown in Figure 2. NCL uses special gates called threshold (TH) gates. The NCL TH gates have hysteresis state-holding capability such that the output once asserted does not deassert until all inputs deassert. This attribute of the TH gates facilitates input completeness, thus enabling the circuits to function without a clock [7, 8]. The output of such a gate can be described as F = set + (F'*Hold), where F' is the previous output value. The set equation determines when the gate will be asserted, and the hold equation determines how long the gate remians asserted. For example, the set equation for TH23 is AB + BC + AC, and the hold equation is A + B + C. The gate is thus asserted when any two inputs assert, and it deasserts only when all inputs deasssert.

The proposed architecture introduced the NCL paradigm to nanowire crossbar architecture. Normal crossbar architecture is similar to a conventional, clock-based, Boolean circuit; it requires a clock to synchronize the flow of data. In contrast, asynchronous crossbar architecture is data driven. Instructions are acted upon the moment they are available, and output is available the moment it is completed. The proposed architecture consists of an array of PGMB, which are interconnected in the form of a 2D grid structure. These blocks are surrounded by nanowires that are used to route the signals inside the grid structure. The PGMB's input and output nanowires cross these routing wires, forming programmable crosspoints. By programming these crosspoints, signals can be routed to any of the programmable gate macro blocks [5]. The input stage consists of programmable resistor crosspoints formed by the micro wires and nanowires. By programming relevant crosspoints, signals can be routed to the required PGMB. Each block can in turn be programmed to implement any of the threshold gates [5]. These blocks can tap the input signals by programming corresponding crosspoints. These crosspoints are formed from the nanowire column carrying the input signal and the nanowire row, which is an input to the macro block. The output of the TH gate thus implemented [5] can be routed to the other gates in a similar fashion. Thus the number of columns of nanowires between programmable macro blocks determines the number of crosspoints available for routing signals. This former number must be sufficient to route all the required inputs and outputs to the macro blocks. However the number of rows and columns of PGMBs in the grid is limited by the amount of signal degradation caused by propagation. Before complete degradation of the signal, a buffering stage can be implemented to restore signal strength.

3. DEFECT DENSITY PROBLEMS IN NANOWIRE CROSSBAR ARCHITECTURE

Nanowire crossbar systems are prone to defects due to the non-deterministic nature of unconventional nanoscale assembly. A defect rate as high as 10% is usually anticipated. The crosspoints may be stuck-open (i.e., always OFF) or stuck-closed (i.e., always ON). A stuck-open crosspoint can never be used to program an ON-input since it will never conduct, thus producing a wrong output. Similarly, a stuck-closed crosspoint will always conduct, thus issuing faulty output as well. Nanowires may be broken, accounting for unreliable outputs. In case of stuck-open and stuck-closed defects, only the particular crosspoint involved becomes unusable. In case of a broken wire, however, no part of the wire can be used for programming any function. These types of manufacturing defects are unavoidable and need to be tolerated. A high number of defects occur in the crossbar due to localized imperfections and variations in nanofabrication. Figure 3 shows a threshold gate TH34w2 implemented on PGMB, with each dot indicating a location of ON-input. Although the minimum number of rows and columns required in a PGMB provides flexibility to implement any of the 27 gate macros, the defects present at the PGMB crosspoints will prevent implementation of some crosspoints as shown in Figure 4 so that the gate cannot be programmed.

In a grid formed by the intersection of orthogonal cross wires, vertical nanowires having the pull-up resistors form the AND plane and horizontal nanowires with pull-down resistors form the OR plane and add the product terms to give the output. The TH gates are mapped onto the diode crossbar structures on the AND and OR planes [9]. Thus each vertical wire in the crossbar provides a product term of the TH gate equation, and the horizontal nanowire adds these terms to give the result in Sum of Products form. The defects do not affect the logic of a gate as long as they coincide with the OFF crosspoints. Columns can be shuffled to route away from the defects since the product terms are commutative; however shuffling the rows is not allowed and is restricted to the respective planes. Research is going on to improve these circuits in terms of defect tolerance [10].

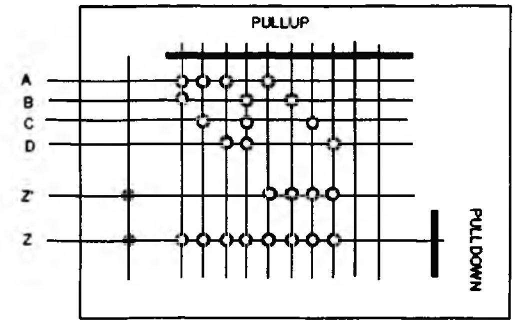


Figure 3 TH34W2 realized on PGMB

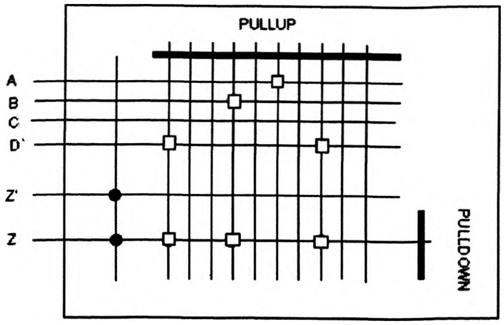


Figure 4 PGMB with defective crosspoints

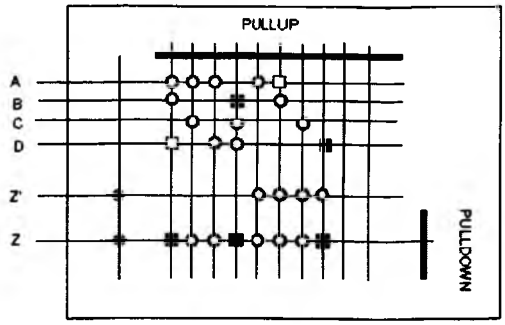


Figure 5 TH34W2 on the defective PGMB

4. NOMENCLATURE

The following notations will be used throughout this paper:

λ : Defect rate, $0 <= \lambda <= 1$.

m: Number of columns having ON crosspoints (since there might be unused columns in the crossbar) .

n_i : Number of ON crosspoints in i^{th} column.

k : Total number of defects in the given PGMB.

k_i : Number of defects in i^{th} column.

P(k): Probability of mapping the given TH gate when there are exactly k coinciding defects in the PGMB.

$p(k')$: Probability of mapping a column of the PGMB when k^0 defects coincide with the ON crosspoints of the column.

$P_n(k_i)$: Probability of mapping a function when k_i defects coincide with the ON crosspoints in one column.

n: Total number of columns in the PGMB.

r: Total number of rows in the PGMB.

a: Number of defect free crosspoints.

b: Number of defective crosspoints.

5. MAPPING NCL GATES AND THE DEFECTS

5.1 Modeling the mapping probability

Each PGMB should have at least a dimension of 6x10 to program any given TH gate macro. The product terms are mapped on the vertical cross wire corresponding to a single pull-up register, and the horizontal wires with pull-down register add the terms together. Various factors affect the calculation of the probability of mapping a TH gate onto the crossbar architecture for defects coinciding with the ON crosspoints. These factors include the number of columns having ON crosspoints (*m*), the number of ON crosspoints in each column ($n_{i,}$ where subscript i denotes the ith column), the size of the PGMB, and the defect rate (λ).

The defect rate indicates the average number of defects in a crossbar system. The TH23 gate can be expressed as F = $AB + BC + AC + AF' + BF' + CF'$, where A, B and C are the primary inputs, and F' is the feedback term. For a defect rate of λ, the probability of having a defect free crosspoint is (1 - λ). If k is the number of total defects in the given PGMB, then $P(k)$ is the probability of mapping the logic function on the PGMB for k coinciding defects. The probability of successfully mapping the first column for 0 defects would be

$$p(0) = (1 - \lambda)^3$$

Each of the six columns in TH23 has three ON crosspoints. Therefore the total probability of mapping the given function on the PGMB in case when no defect coincides with any of the ON crosspoints is $p(0)^6$

$$P(0) = (1-\lambda)^{(3 \cdot 6)}$$

The probability $p(1)$ of mapping the single column when one defect coincides with one of the ON crosspoints in the PGMB can be calculated as:

$$p(1) = \binom{3}{2}(1-\lambda)^2\lambda\left\{\binom{3}{3}(1-\lambda)^3 + \binom{3}{2}(1-\lambda)^2\lambda + \binom{3}{1}(1-\lambda)^1\lambda^2 + \binom{3}{0}\lambda^3\right\}$$

Considering all the six columns, the probability is:

$$P(1) = \binom{6}{1}p(1)p(0)^5$$

Three defects in the architecture can coincide with the ON crosspoints in the PGMB in following ways:

1. All defects in one column:

$$P_1(3) = \binom{6}{1}\lambda^3 P'(0)^5$$

2. Two defects in one column, and one defect in another column:

$$P_2(3) = \mathrm{P}^6_2\binom{3}{2}(1-\lambda)\lambda^2\binom{3}{1}(1-\lambda)^2\lambda P'(0)^4$$

3. All defects in different columns:

$$P_3(3) = \binom{6}{3}\left\{\binom{3}{1}(1-\lambda)^2\lambda\right\}^3$$

The above calculations are specific to the TH23 gate for a PGMB of size 6x10. Since the number of programmable crosspoints per column (n_i) and the number of columns having programmable crosspoints (m) differ for each TH gate along with the change in the PGMB dimension, a general equation for total probability can be given in terms of the above parameters as follows:

$$P(k) = \sum_{i=1}^{n}\left\{\left(\mathrm{P}^m_{\#\text{of defective columns}}\binom{n_i}{k_i}(1-\lambda)^a\lambda^b\right)\sum_{i=1}^{n}\binom{r-n_i}{k_i}(1-\lambda)^a\lambda^b\right\}$$

where a is the number of defect-free crosspoints, and b is the number of defective crosspoints. The total number of rows and columns is represented by r and n respectively. Since the addition of the product terms is commutative, columns can be added to a PGMB to obtain optimal mapping. This option provides the flexibility to choose the appropriate column for mapping a particular function by choosing a term from the function that does not use the same crosspoint as its ON crosspoint. Thus even a defective PGMB can be used to map a function successfully. This is why the column-wise probability is calculated.

The proposed probability model and results can be used in various ways as design criteria. Possible applications include:

1. Redundancy optimization

The primitive building block of the proposed architecture is PGMB. Each PGMB can be designed to have redundant rows and columns to improve defect-tolerance.

2. Analysis optimization

It is not practical to test all possible faults with 100% test coverage. Therefore, faults should be prioritized, and an appropriate number of highly-probable faults should be included in the fault set. Also, the overall testing efficiency and the overall testing overhead should be properly balanced to achieve the optimal result.

3. Repair optimization

There are various ways to tolerate defects in PGMB. Firstly, the order of rows and columns can be rearranged to circumvent the defects. Secondly, more rows and columns can be added to the base dimension of 6 x 10 to provide local redundancy. Finally, it is also possible to allocate redundant PGMBs to provide global redundancy.

5.2 Optimal PGMB dimension for gates

The probability calculation demonstrated above can be extended to find the optimum PGMB size for a given threshold gate. The defect aware approach, which generates the defect map before mapping the function on the PGMB, has the time and space overhead since it needs a defect map and a library of these maps to program the gate successfully. The defect unaware approach, however, is much faster because the gate is programmed over the PGMB without any knowledge of the positions of defects. By calculating the probability of mapping a gate successfully on a PGMB of a given size, the defect unaware approach can be used very efficiently.

This work has developed an analysis to find the optimum PGMB size for mapping all threshold gates. As mentioned above, a PGMB of size 6x10 can be used to program any threshold gate; therefore, this dimension will be used for our calculations. The analysis presents three ways to increase PGMB size. First, the number of columns is increased while keeping the row count constant. Second, the number of rows is increased while keeping the column count constant. Finally, the number of both rows and columns is increased. These three methods deliver different results, and the optimum size is determined according to the programmability threshold required.

Since the addition of AND terms is commutative, columns can be added to the PGMB block easily. When adding rows, on the other hand, the AND and OR planes must be considered separately. This work simulated the two blocks individually to find the probability of each. Since the OR plane has more ON crosspoints on a single row, the first row is always added to the OR plane. The next redundant row is added to the plane which has a row with maximum number of ON crosspoints, which then becomes the critical row. Thus, the extra row can be allocated to either of the planes according to the algorithm discussed in the section below.

5.3 Critical row algorithm

The following algorithm, called critical row algorithm is used when two or more redundant rows are added to the given PGMB. The first redundant row is added to the OR plane by default since it is critical. This row is also programmed like the existing row. Consider a gate whose location of ON crosspoints is already known. Suppose n_on(i) be the number of ON crosspoints of the i^{th} row and n be the number of redundant rows to be added.

```
% Critical Row Algorithm
    i=1;
    while(i<=n)
    {
            find the row with maximum n_on
            and save its index as j;
            add a redundant row as (j+1)th row;
            program (j+1)th row same as jth row;
            n_on(j)=n_on(j)/2;
            n_on(j+1)=n_on(j+1)/2;
            i++;
    };
```

This algorithm finds the row with maximum number of ON crosspoints in the whole PGMB and allocates the redundant row to the plane corresponding to that row. Thus it makes the best use of available redundant rows.

6. PARAMETRIC SIMULATION AND RESULTS

6.1 Simulation and results for calculating mapping probability

The graph in figure 6 shows how the probability of mapping changes for TH23 with progression in the defect rate. Various plots on the graph show the corresponding probabilities with no coinciding defect, with one coinciding defect, with two coinciding defects, and so on, with a defect rate ranging from 1% to 10%. The probability curve with zero defects has the highest slope, because it is unlikely that no defect would overlap any of the ON crosspoints as the defect rate rises. In case of one or more coinciding defects, the probability of defective mapping increases as the defect rate rises.

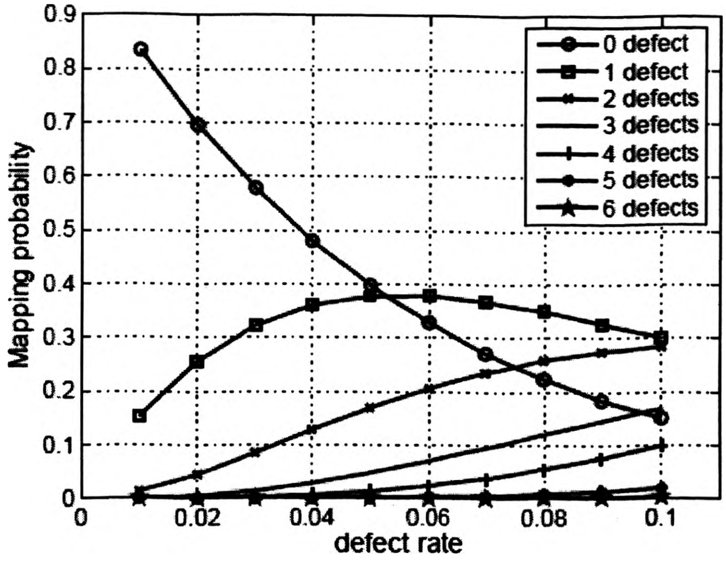


Figure 6 Probability map for TH23 for varying number of defects

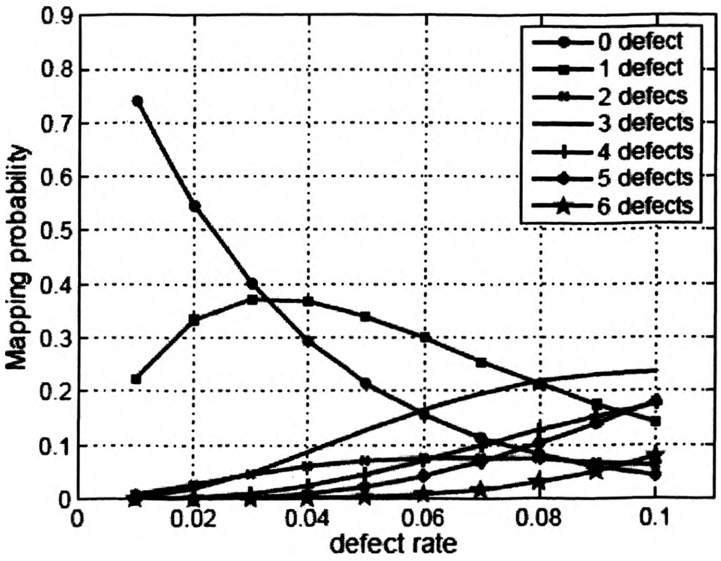


Figure 7 Probability map for TH24 for varying number of defects

Simulation results for TH24 are shown in figure 7. For the low defect rate range, both TH23 and TH24 have similar results. However, the two graphs differ substantially for high defect rate cases. In these cases, a higher number of coinciding defects is shown for TH24. This can be attributed to the larger number of programmable columns in TH24.

Table 1 shows the trend in the probabilities of other gates with the change in defect rate and in the number of defective ON crosspoints. Note that δ denotes the number of coinciding defects.

Table 1

Probability table for TH12, TH23W2, TH34 and TH54W22 gate macros

TH Gate	δ	λ = 0:01	λ = 0:03	λ = 0:05	λ = 0:08	λ = 0:1
TH12	0	0.904	0.7374	0.598	0.4343	0.3486
	1	0.09	0.22806	0.315	0.3777	0.3874
	2	0.0415	0.0317	0.0746	0.1478	0.1937
	3	0.000111	0.0026	0.01	0.0342	0.0111
TH23W2	0	0.8687	0.6528	0.4876	0.311	0.22878
	1	0.1228	0.2826	0.35933	0.3788	0.355
	2	0.00806	0.0568	0.1129	0.2141	0.257
	3	3.25 e-4	7.02 e-3	0.02588	0.0744	0.11422
TH34	0	0.7471	0.4134	0.2259	0.089	0.047
	1	0.21886	0.3707	0.3448	0.2246	0.1517
	2	0.0309	0.1605	0.254	0.2735	0.23608
	3	2.8 e-3	0.04468	0.12	0.214	0.236
TH54W22	0	0.8179	0.54379	0.35848	0.1886	0.12157
	1	0.1652	0.3363	0.377	0.3281	0.27
	2	0.0158	0.0988	0.1886	0.271	0.2851
	3	9.6 e-4	0.01833	0.05958	0.1414	0.19011

6.2 Results for row/column redundancy cases

The effects of increasing the PGMB size on the programming probability were analyzed for these five gates: TH22, TH33W2, TH23, TH44W322 and TH24. The corresponding number of ON crosspoints is 9, 15, 18, 24 and 30 respectively. There are three cases for the simulations: increasing columns only, increasing rows only, and increasing both rows and columns. Lets us consider the defect rate of 10%. Figure 8 shows the successful mapping probability variations according to PGMB size for these five gates when only columns are added. The probability increases with the increase in the PGMB size and is more for the gates with lower ON crosspoint count.

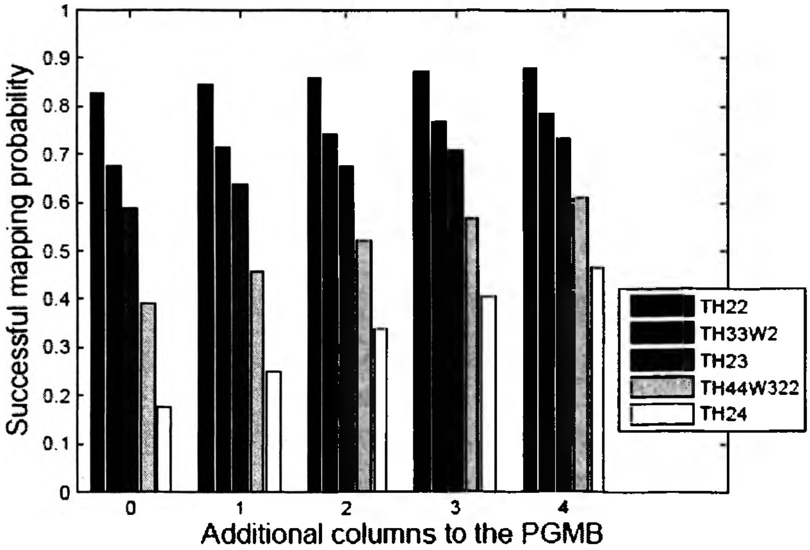


Figure 8 Probability map for columns added to the PGMB at 10% defect rate

The rows are added to either of the planes according to Critical Plane Algorithm. The results of the simulation are shown in figure 9. As seen from the results, the increase in probability is more for adding the rows than adding the columns. This is because the algorithm adds the rows in an optimized way and thus makes the best use of redundancy.

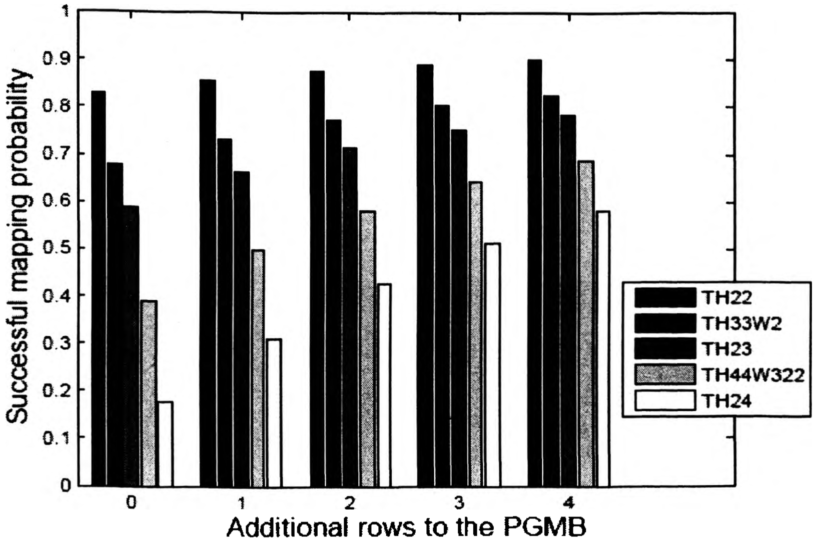


Figure 9 Probability map for rows added to the PGMB at 10% defect rate

Now we consider the third case of adding both rows and columns to the PGMB. The columns are simply added to the PGMB, while for adding the rows again the algorithm is used. The results are shown in figure 10. The results show a significant improvement in the defect tolerance of the PGMB for the increased defect rate. The optimum PGMB size can thus be chosen to map different functions with better success.

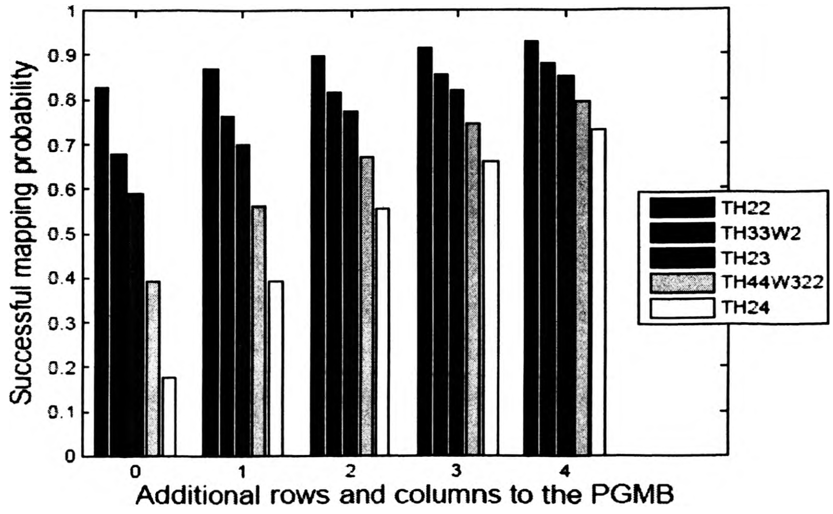


Figure 10 Probability map for rows and columns added to the PGMB at 10% defect rate

The variation in probability for all the gates is also shown in the plots. The plot in figure 11 shows the probability variation when only columns are added to the PGMB. Gate type indicates TH22, TH33W2, TH23, TH44W322 and TH24 indicated by 1 through 5 respectively. The five planes correspond to 5 defect rates ranging between 1% to 10%, top plane being 1%. The darker portion in the plots indicates lower probability. As expected, the probability slopes down for higher defect rates and small PGMB size. The slope is maximum for TH24 which has the maximum ON crosspoints. As we increase the PGMB size and the probability increases in the corresponding defect plane. The plot in figure 12 is the corresponding graph when only rows are added to the PGMB using Critical Plane Algorithm. The graph has similar trend but with a slightly higher probability attributed to the algorithm.

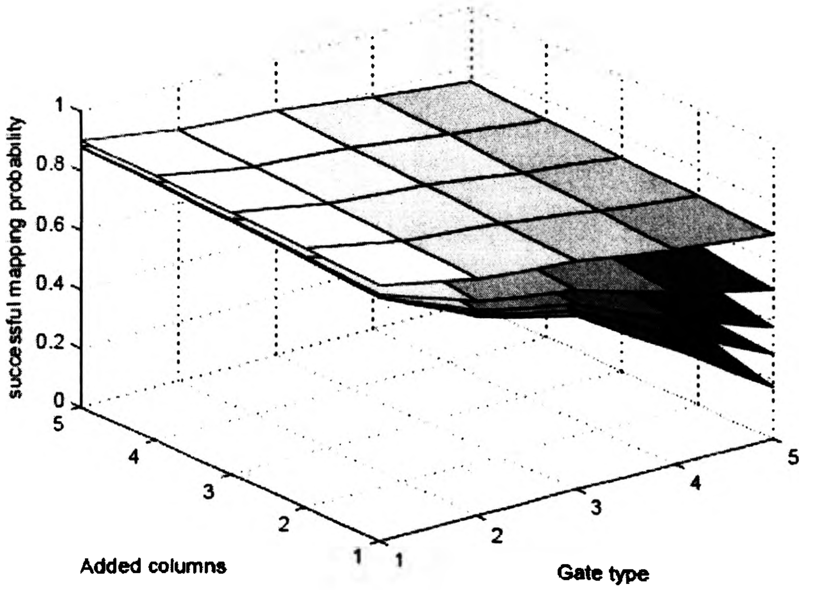


Figure 11 Probability map for added columns and defect rate ranging from 1% to 10%

Finally the graph for increasing both row and column count is shown in figure 13. The probability increased further in this case, as indicated by the lighter shade of black in the plot. The optimum size of the PGMB can be chosen according the type of gate and the defect rate. For example if the programmability threshold is 80% at a defect rate of 10%, the optimum size for TH33W2 will be 8x12. Thus the analysis provides a method to choose the best PGMB size to map all the gates, thus making the circuit more tolerant.

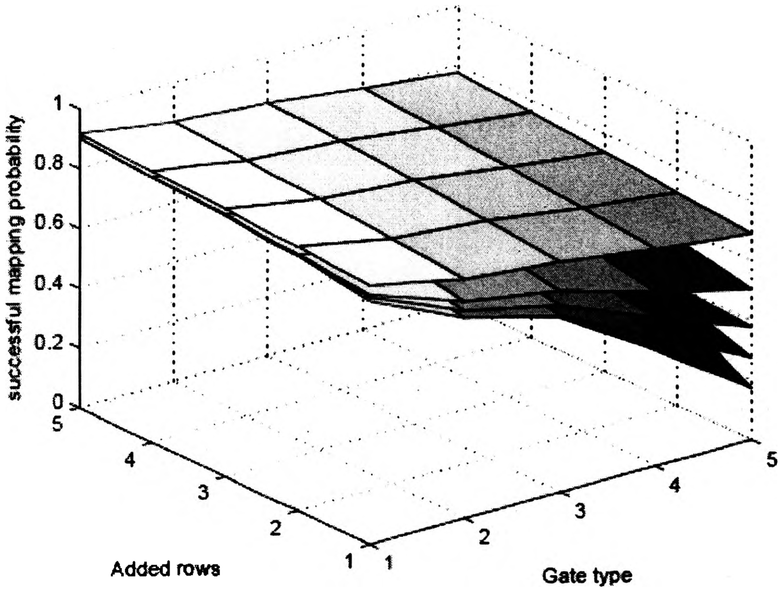


Figure 12 Probability map for rows added and defect rate ranging from 1% to 10%

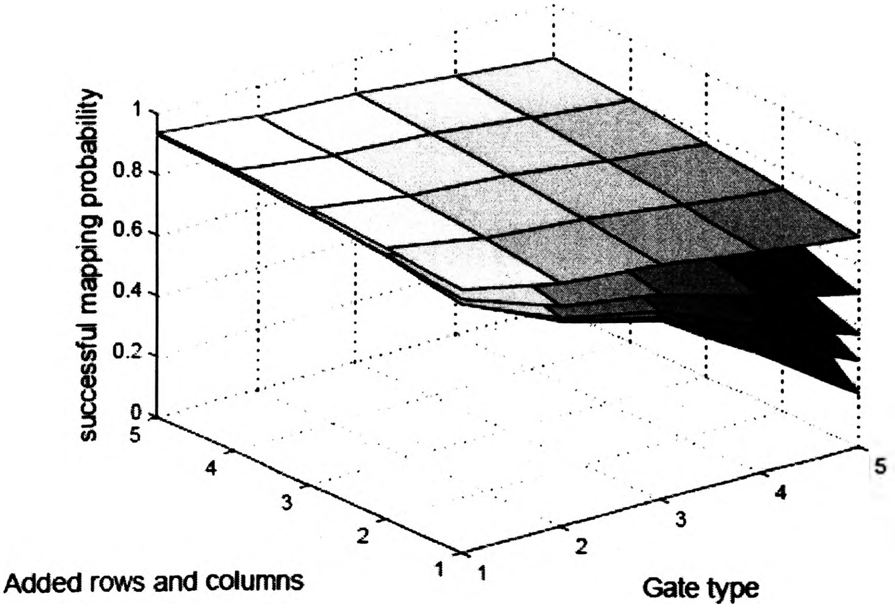


Figure 13 Probability map for rows and columns added and defect rate ranging from 1% to 10%

7. CONCLUSIONS

Although the recently proposed asynchronous nanowire crossbar architecture offers better manufacturability, scalability and robustness than its clocked counterpart, it has a high defect rate due to nondeterministic nanoscale assembly. This issue must be addressed, and physical systems based on the clockless architecture should be designed, tested, and repaired to maximize programmability and fault tolerance while minimizing the overhead. In this paper, a new numerical model is initially proposed to measure the probability of mapping as a function of coinciding defect(s). Then, the proposed model has been used to measure the programmability of various redundancy allocation cases and to find the optimal PGMB dimension for asynchronous nanowire crossbar architecture. This defect unaware approach avoids the complex pre-mapping analysis for creating defect map library to map the function on the PGMB. So this approach avoids time overhead of defect aware approach to map the gates in less time and complexity.

8. REFERENCES

[1] Matthew M. Ziegler and Mircea R. Stan, "Design and analysis of Crossbar Circuits for Molecular Nanoelectronics," IEEE Nanotechnology Conference, pp. 323-327, 2002.

[2] Andre Dehon "Nanowire-based programmable architectures" ACM Journal on Emerging Technologies in Computing Systems (JETC), Volume 1 , Issue 2 , July 2005

[3] Nicolas A. Melosh, Akram Boukai, Frederic Diana, Brian Geradot, Antonio Badolato, Pierre M. Petroff, James R. Health, "Ultrahigh-Density Nanowire Lattices and Circuits," Science, Vol. 300, pp. 112-115, 2003.

[4] Neil H. E. Weste, K. Eshraghian, "Principles of CMOS VLSI design", Second Edition, pp. 322-325.

[5] R. Bonam, S. Chaudhary, Y. Yellambalase and M. Choi, "Clock-Free Nanowire Crossbar Architecture based on Null Convention Logic (NCL)", 7th IEEE International Conference on

Nanotechnology (IEEE-Nano), Apr 2007.

[6] Karl M. Fant, Scott A. Brandt, "NULL Convention Logic: A Complete and Consistent Logic for Asynchronous Digital Circuit Synthesis," IEEE International Conference on Application-Specific Systems, Architectures and Processors, pp. 261-273, 1996.

[7] S. Smith, R. DeMara, J. Yuan, M. Hagedorn and D. Ferguson, "Delay-Insensitive gate-level pipelining," Integration, the VLSI journal, Vol. 30, pp. 103-131, 2000.

[8] S. C. Smith, R. F. DeMara, J. S. Yuan, D. Ferguson, and D. Lamb, "Optimization of NULL Convention Self-Timed Circuits," Integration, The VLSI Journal, Vol. 37, No. 3, pp. 135-165, 2004.

[9] Y. Yellambalase , M. Choi and Y. Kim, "Inherited Redundancy and Configurability Utilization for Repairing Nanowire Crossbars with Clustered Defects",pp. 98-106, IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2006.

[10] Zhiyong Li, Matthew D Pickett, Duncan Stewart, Douglas A A Ohlberg, Xuema Li, Wei Wu, Warren Robinett and R Stanley Williams, “Experimental demonstration of a defect-tolerant nanocrossbar demultiplexer”, Quantum Science Research, Hewlett-Packard Laboratories, Palo Alto, CA 94304, USA.

VITA

Shikha Chaudhary was born in Nagpur, Maharashtra, India on 28th of October, 1983. She did her schooling from Ghaziabad, Uttar Pradesh. She received her Bachelor's degree in Electronics and Communication from Uttar Pradesh Technical University, Uttar Pradesh in June 2006. She pursued her Masters from Missouri University of Science and Technology (Formerly University of Missouri Rolla), Rolla, Missouri, USA in Computer Engineering starting August 2006. She was a Teaching assistant for 2007 and Research Assistant from 2006 through 2008. She did a Co-op at Advanced Micro Devices, Boston, USA for a period of seven months in 2008 and graduated in December 2008.