01 Aug 2008

Evaluating Performance Tradeoff in Defect-Tolerant Gate Programming Techniques for the Clock-Free Nanowire Crossbar Architecture

Minsu Choi
Missouri University of Science and Technology, choim@mst.edu

Ravi Bonam

Recommended Citation

Evaluating Performance Tradeoff in Defect-Tolerant Gate Programming Techniques for the Clock-Free Nanowire Crossbar Architecture

Ravi Bonam and Minsu Choi

Dept of ECE, Missouri University of Science & Technology

Rolla, MO 65409-0040, USA

{rkbcdf, choim}@mst.edu

Abstract— **A novel asynchronous nanowire crossbar architecture has been recently proposed by authors' research group. The proposed clock-free architecture provides numerous significant benefits over its clocked counterparts which include better manufacturability, scalability, modularity and robustness. We also proposed various gate mapping and reconfiguration algorithms for defect-tolerant programming of PGMB (programmable gate macro blocks) - which is the primary building block of the proposed architecture. These algorithms were tested by simulations and a variety of parameter values were applied to show their performance characteristics. The most important performance metric of the proposed techniques is the programmability (i.e., the ratio of successfully programmed gates to the total number of gates). However, algorithms with higher programmability should come with higher time/space requirements. In this work, we will evaluate the tradeoff between programmability and time/space requirements and suggest a way to find the most suitable algorithm with acceptable combination of programmability and time/space requirements.**

I. INTRODUCTION

Clock-Free Nanowire crossbar architecture is based on two emerging technologies, nanowire crossbar architecture and NCL(Null Conventional Logic) [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13].

The nanowire crossbar architecture consists of a two dimensional array of orthogonal nanowires. It has been proved that the crosspoints of the array can be realized as programmable diodes, memory cells or FETs(Field effect transistors) [5, 6].

NCL is a delay insensitive asynchronous paradigm and uses NULL-DATA convention to synchronize the operation of the circuit. Basic components used in this technology are the threshold gates which are analogous to the logic gates in boolean logic. The logic expression of the gates depend on the number of inputs given to the gate(n, maximum of 4) and number of asserted inputs(m) and the gate is represented as THmn. Hence, TH23 gate's expression would be $F = AB + BC + AC + AF' + BF' + CF'$, where A, B, C are the primary inputs and F' is the output feedback. The last three product terms in the expression satisfy the hysteresis behavior. [11, 12, 13, 14]

The asynchronous nanowire crossbar architecture is built around using uniformly sized programmable crossbar blocks, PGMBs (Programmable Gate Macro Blocks) which can be configured to realize the function of any given threshold gate [14]. Mapping and Placement algorithms were also proposed [2] to efficiently program the PGMBs and realize any given logic expression. The algorithms were simulated on PGMBs with randomly placed defects which gave a brief perspective of their performance.

The most important result used to distinguish them was programmability i.e., ratio of programmed gates to the total number of gates. They were simulated for various defect rates at variable inherent redundancy which provided us with a brief perspective to evaluate their performance. This paper addresses time and space related issues to give a comprehensive view of the algorithm's performances.

II. REVIEW OF MAPPING AND PLACEMENT ALGORITHMS

There are four different mapping and placement algorithms proposed, each of them have its own pros and cons. Defect Unaware and Defect Aware techniques [2] are considered to be the two extremities and the shift and modified shift techniques act as compromised approaches (briefed in this section).

A. Defect Unaware approach

This approach maps a predefined gate pattern onto the PGMB without the knowledge of position of the defects. This approach minimizes programming time at the cost of programmability. The programmability of the gates in this approach is highly dependent on the manufacturer's ability to fabricate defect-free PGMB's.

B. Defect Unaware - Shift Approach

This approach is an extension of the Defect-Unaware approach and employs a circular shift procedure which shifts the columns in both the AND and OR planes collectively. Shifting the planes creates greater number of representable patterns of the gates which increase the probability of successful mapping while maintaining proper functionality. This approach creates a better trade-off between the time required to program and programmability. It would yield better programmability compared to the Defect-unaware Approach described in the previous section.

C. Defect Unaware - Modified Shift Approach

This approach is an annexure to the Shift algorithm and applies the shift algorithm's property on the AND plane's rows and columns, in addition to shifting the columns of the OR plane. This technique creates greater number of gate patterns than the previous methods which comparatively increases the probability of successfully programming a defective PGMB. It yields better programmability compared to the previous methods at noticeable defect rates.

D. Defect Aware Approach

This algorithm will scan through the entire PGMB and generate a defect map. The defect map will provide information about the defects which will allow efficient utilization of inherent redundancy. This algorithm is greedy, exhaustive and one of its important property is, it starts programming the OR plane and then goes on to program the corresponding AND plane's column. This property helps reduce programming time at high defect rates.

III. PRELIMINARY RESULTS

The algorithms (Defect Aware and Defect Unaware) have already been simulated under various defect rates and the programmability pertaining to each of them has been analyzed in [2]. This section analyzes the time and space required to perform these simulations and correlates them to the programmability of the corresponding algorithms. We will further extend this work and propose a method to find the most suitable mapping and placement algorithm with balanced combination of programmability and time/space requirements among different candidates.

Time and Space are important factors of the algorithms that have to be taken into consideration by the manufacturer before implementing them. These parameters would directly influence the cost of programming and eventually the entire cost of production.

Results presented in this section are based on simulations performed on a Pentium Core 2 Duo 2.2 GHz system, number of gates programmed in each case were 10000 and defects were randomly introduced onto a 6X10(rows X columns) PGMB.

Figure 1 illustrates the execution time of the Defect Unaware approach for programming various gates at different defect rates. The graph clearly demonstrates similar programming times for all defect rates. This is completely logical due to the fact that it does not employ any sort of intelligence or reconfiguration based on defects locations.

This approach utilized a total of 6360 bytes for programming 10000 gates over 6X10 PGMBs.

Figure 2 shows the effect of defect rate on execution time for the Shift Approach. We can see the increase in execution time as defect rate increases and it seems to stabilize after 15% defect rate. This anomaly can be correlated to the fact that programmability reduces to a zero percent at such defect rates as illustrated in figure 3. Hence, at high defect rates the algorithm would be unable to find a successful mapping
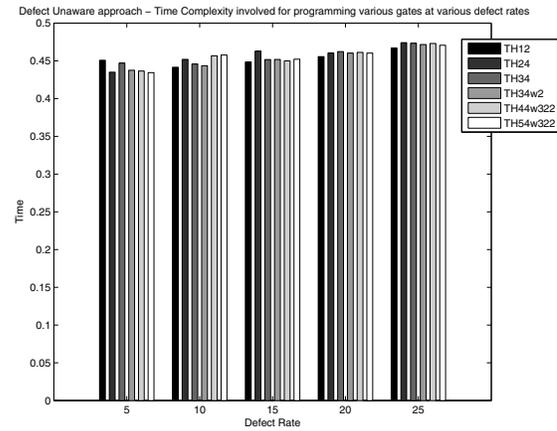

Fig. 1. Execution time for PGMB programming at various defect rates-Defect Unaware approach.

configuration of the gate and would be inefficient. At defect rates greater than 15%, the algorithm would be confronted with its worst case and would be shifting through all its possible configurations which explains the reason for high execution times.

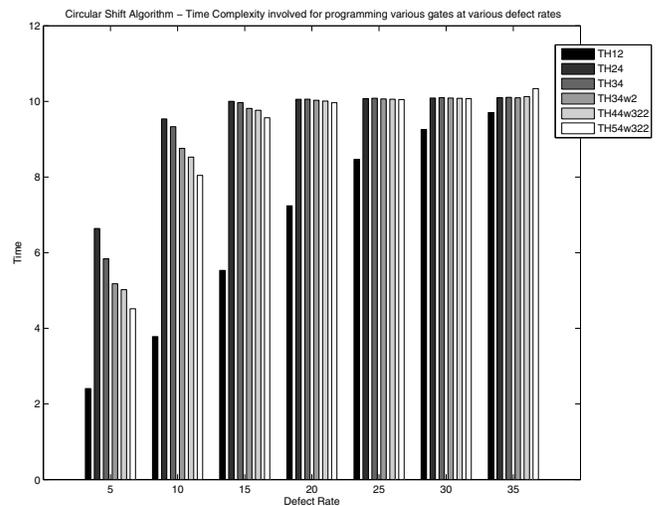This approach utilized a total of 6664 bytes for programming 10000 gates over 6X10 PGMBs.

Fig. 2. Execution time for PGMB programming at various defect rates - Defect Unaware Shift approach.

The execution time for the Modified shift approach would be similar to the Shift approach. The fact that this approach creates greater set of possible gate configurations would increase the worst case time at higher defect rates. The only difference between both the approaches would be the worst
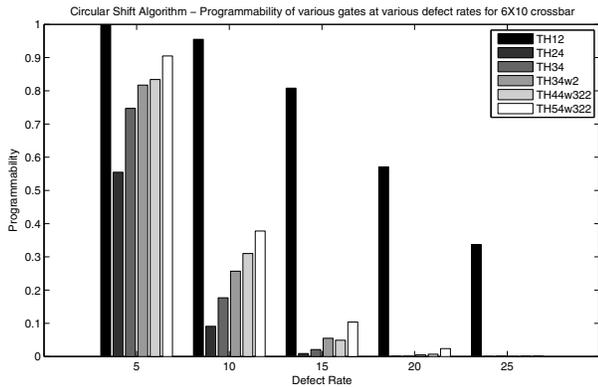
Fig. 3. Programmability at various defect rates - Defect Unaware Shift approach.
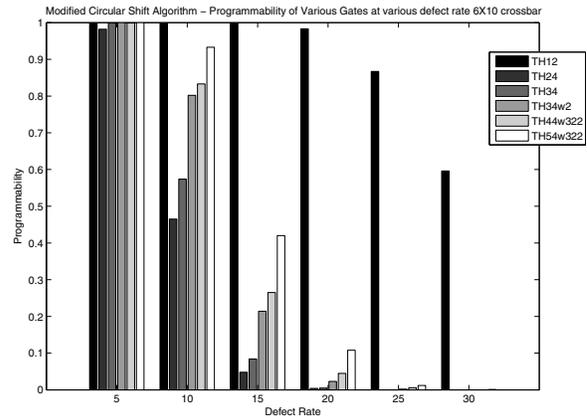

Fig. 5. Programmability at various defect rates - Defect Unaware Modified Shift approach.

case defect rate, which would be higher in the latter case.

Figure 4 illustrates the Modified Shift approach's execution time at various defect rates. The results presented in this plot are analogous to those in figure 2. We can also observe the increase in time to program the gates at higher defect rates. These results can be correlated to the programmability of these gates at similar defect rates as shown in figure 5. Similar to the previous approach the worst case is met at defect rates greater than 20% as the algorithm cycles through all the possible configurations.

This approach utilized a total of 8900 bytes for programming 10000 gates over 6X10 PGMBs.
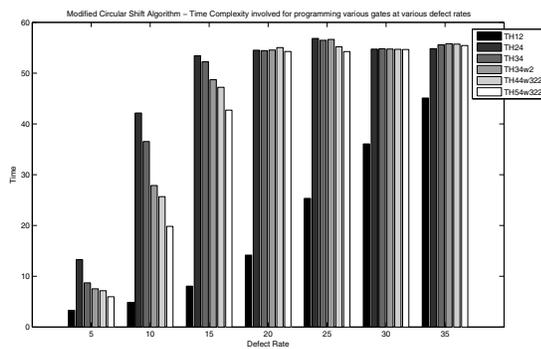
decrease in time can be correlated to programmability and the algorithm's design [2]. The algorithm was designed in a way to avoid unnecessary programming of gates if terms in the logic expression of the gate couldn't be summed up in the OR plane. Programming starts with the OR plane which has lesser dimensions(20% of rows in the PGMB are dedicated to the OR plane) and moves on to the AND plane.

This approach utilized a total of 19384 bytes for programming 10000 gates over 6X10 PGMBs.

Fig. 4. Execution time for PGMB programming at various defect rates - Defect Unaware Modified Shift approach.
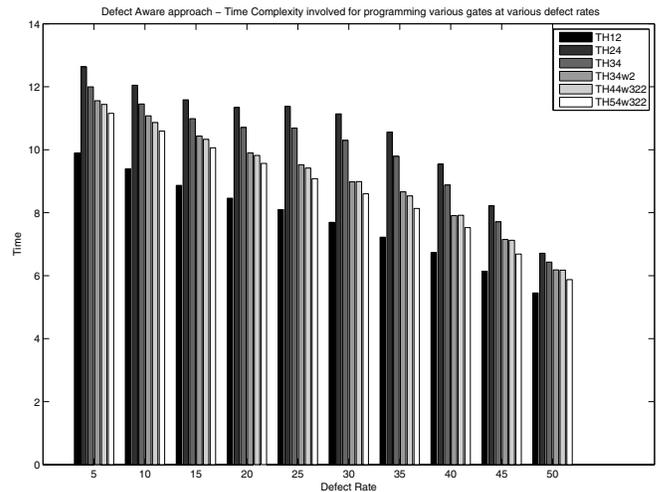

Fig. 6. Execution time for PGMB programming at various defect rates - Defect Aware approach.

The execution times at various defect rates for the defect Aware approach are illustrated in figure 6. On careful analysis, it seems to contradict the expected plot i.e. as defect rates increase the execution time should also increase. The anomalous

IV. CONCLUSION AND FUTURE WORK

The results produced in this paper purely address the Asynchronous Crossbar architecture and will be required by
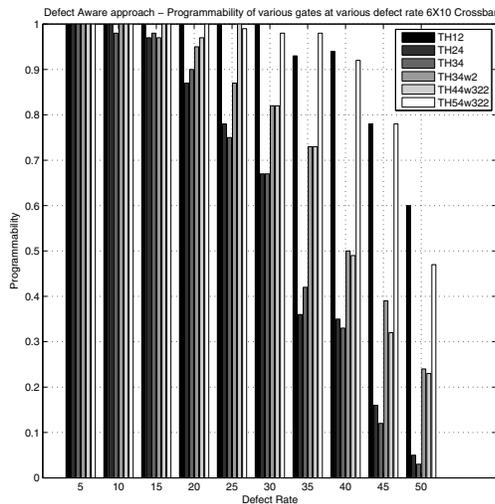
690

Fig. 7. Programmability at various defect rates - Defect Unaware Modified Shift approach.

manufacturers to select the appropriate technique in mapping and placement of PGMBs.

As predicted, Defect Unaware and Defect Aware approaches give two extremes of programming the PGMB based on both execution time and space consumed, excepting worst scenarios where the shift algorithms seem to contradict the statement(in case of execution time). The Shift and Modified shift approaches act as compromised approaches, with moderate times of execution and space consumed.

The shift algorithms should be given a test run to get an approximate threshold defect rate at which they fail to be efficient. Due to their defect unaware nature they will shift through all the possible combinations before they declare that a particular PGMB cannot be programmed, which would be an unnecessary waste of resources.

Complexity of the circuits being programmed is also a significant measure in choosing the appropriate algorithm. If the circuits are simple and bound to utilize less crosspoints, they can be programmed using a simple algorithm like the Defect-Unaware approach or the Shift Approach which would ensure greater programmability than the former.

Moderately complex circuits can be programmed using the shift approaches which would yield significant programmability, and would have high programmability given some inherent redundancy. Manufacturers looking at higher programmability without introducing inherent redundancy can use the Defect Aware approach which would cost some resources.

Complex circuits to be programmed on PGMBs need to use the Defect Aware approach as they would be using up most of the crosspoints and to get a successful configuration in a defective environment would be impossible for the other approaches.

This gives a comprehensive view of the algorithms and the issues still to be addressed are the interconnection strategies.

These issues will be discussed in future publications which would help in developing larger circuitry.

REFERENCES

[1] 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.
[2] Ravi Bonam, Yong-Bin Kim, Minsu Choi, "Defect-Tolerant Gate Macro Mapping and Placement in Clock-Free Nanowire Crossbar Architecture", 22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems, 2007. DFT '07. 26-28 Sept. 2007 Page(s):161 - 169
[3] Mark T. Bohr, Senior Member, IEEE "Nanotechnology Goals and Challenges for Electronic Applications", IEEE TRANSACTIONS ON NANOTECHNOLOGY, VOL. 1, NO. 1, MARCH 2002
[4] Forshaw, M., Stadler, R., Crawley, D., Nikolic, K. "A short review of nanoelectronic architectures " Nanotechnology, Volume 15, Issue 4, April 2004
[5] G. Snider, P. Kuekes, T. Hogg, R. Stanley Williams "Nanoelectronic architectures", Applied Physics A: Materials Science and Processing,Volume 80, Number 6, March 2005
[6] Andr Dehon "Nanowire-based programmable architectures" ACM Journal on Emerging Technologies in Computing Systems (JETC), Volume 1 , Issue 2 , July 2005
[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] 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.
[9] Y. Cui and C. M. Lieber, "Functional nanoscale electronic devices assembled using silicon nanowire building blocks," Science, Vol. 291, pp. 851-853, 2001.
[10] 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.
[11] 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.
[12] 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.
[13] 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.
[14] S. C. Smith, "Gate and Throughput Optimizations for NULL Convention Self-Timed Digital Circuits", Ph.D. Dissertation, School of Electrical Engineering and Computer Science, University of Central Florida, May 2001.

691