

Probabilistic Analysis of Design Mapping in Asynchronous Nanowire Crossbar Architecture

Shikha Chaudhary and Minsu Choi
Dept of Electrical & Computer Engineering
Missouri University of Science & Technology
Rolla, MO 65409
Email: {sc8tc, choim}@mst.edu

Yong-Bin Kim
Dept of Electrical & Computer Engineering
Northeastern University
Boston, MA, 02115
Email: ybk@ece.neu.edu

Abstract—There have been numerous nanowire crossbar architectures proposed till date and they are envisioned as clock-driven. To deal with numerous issues caused by clocking, a new asynchronous architecture based on Null Convention Logic (NCL) has been recently proposed, resulting in the removal of the clock circuit overhead from the crossbar architecture. The proposed architecture is easier to be manufactured and all clocking-related issues are also eliminated. Even though the proposed architecture has considerable merits over its clocked counterpart, it is still prone to defects due to nondeterministic nature of nanoscale assembly and the defect density is directly proportional to the density of nanowires in the architecture. Hence a number of ways are being examined to effectively avoid the defects to make the architecture defect tolerant. Considering the fact that the nano crossbar architecture has a large number of defects due to manufacturing constraints and its extremely small size, the variations in mapping probabilities with certain factors, such as defect rate, size of crossbar matrix, and type of threshold gate are numerically measured to get an optimized design for resilient clock-free nanowire crossbar systems.

Index Terms—Defect-tolerance, Probabilistic modeling, Asynchronous nanowire crossbar system.

I. INTRODUCTION

With advancements in Nanotechnology, researchers have begun to focus their efforts on the development of viable nanoscale systems. Nanowires are one dimensional structures which exhibit interesting electrical properties. The 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 computing architecture recently proposed. 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], such as conventional diode, Field Effect Transistor or a resistor [1, 2].

We recently proposed a new asynchronous nanowire crossbar architecture with considerable merits [4]. The proposed architecture is based on a delay-insensitive logic paradigm called Null Convention Logic (NCL) [5]. In the proposed architecture, we use 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 discrete number of rows and

columns on which any NCL gate can be programmed. Figure 1 shows a PGMB with a threshold gate (described in section II.B) mapped on it. The vertical wires with pull up resistor form the product terms or the AND plane and the horizontal wires with a pull down resistor add them together with the OR logic. It has a feedback wire to provide the current output at the input.

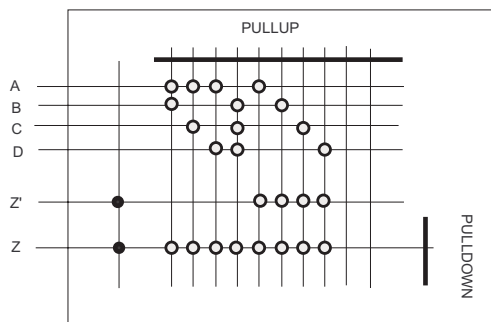


Fig. 1. TH34w2 realized on PGMB.

A 6×10 defect-free PGMB can be used to program any of the 27 NCL gates. The cross points can be programmed as ON or OFF by applying a voltage but the current bottom up process of assembling will lead to many imperfections in the nano crossbar architecture. Instead of trying to reduce the number of defects, a more practical approach is to route the ON crosspoints away from them. The most challenging part of this approach is to find an optimum mapping technique to make the structure defect tolerant. Thus knowing the distribution of probabilities having variable number of coinciding defects becomes important. This paper relies on the defect unaware approach, which maps the function without the previous knowledge of defect locations on the PGMB and hence is faster. The factors that affect the mapping probability of a gate are defect rate (i.e. average number of defects in the crossbar), type of the TH (Threshold) gate and the size of crossbar matrix.

II. PRELIMINARIES AND REVIEW

A. Conventional Clocked Nanowire Crossbar Architecture

In order to realize a large scale nano system, its hardware architecture should have very simple periodic structure which is suitable for bottom-up nanoscale assembly. Since the clock distribution network has poor scalability, it is hard to correctly assemble using nondeterministic nanoscale manufacturing techniques. Also, there are numerous clocking-related issues that may impact the overall system functionality. So, clocking is considered as one of the significant overhead in nanoscale crossbar systems. The proposed clockless circuits offer various advantages over the clocked counterparts, as discussed in the further section.

B. The New Approach: The Proposed Clockless Crossbar Architecture

A new clock-free architecture has been recently proposed to circumvent various issues associated with conventional clocked nanowire crossbar systems [4]. The proposed architecture is based on an asynchronous logic paradigm known as NCL (Null Convention Logic) [5]. NCL 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 [5]. Two states DATA and NULL synchronize the functioning. 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, the register 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 removal of clock distribution network and clock-related failure modes is especially contributing in the proposed asynchronous nanowire crossbar architecture.

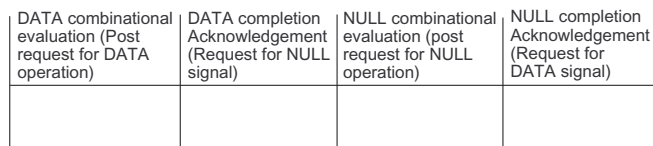


Fig. 2. NCL Timing diagram [6].

The DATA to DATA cycle timing diagram is shown in figure 2. It uses special gates called threshold (TH) gates. The NCL TH gates have the hysteresis state-holding capability such that the output once asserted does not deassert until all inputs deassert. This attribute of the threshold gate helps in achieving input completeness feature enabling the circuits to function without the clock [6, 7]. The output of such a gate can be described as $F = \text{set} + (F' * \text{hold})$ where F' is the

previous value. The 'set' equation determines when the gate will be asserted and the 'hold' equation determines till when would the gate be asserted once it is in the asserted state. For example, the set equation for TH23 is $AB + BC + AC$ and the hold equation is $A + B + C$. So the gate is asserted when any two inputs assert and it deasserts only when all the inputs deassert.

NCL uses special gates called threshold (TH) Gates. which have the hysteresis state-holding capability such that the output once asserted does not de-assert until all the inputs de-assert. This attribute of the TH gates facilitates input completeness, thus enabling the circuits to function without a clock [6, 7]. The output of such a gate can be described as $F = \text{set} + (F' * \text{hold})$ where F' is the previous output. The 'set' equation determines when the gate will be asserted and the 'hold' equation determines how long the gate remains asserted. 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. The proposed architecture consists of array of PGMBs which are interconnected in the form of 2D grid structure to route the signals to the required PGMB [4].

III. DEFECT 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. These manufacturing defects are unavoidable and must be tolerated in the architecture. Figure 3 shows a defective PGMB. Even though the minimum required rows and columns of PGMB gives flexibility to implement any of the 27 gate macros, the defects present at the PGMB cross points will prevent implementation of some of them as shown in figure 4 where TH34w2 is implemented on the defective PGMB. The dark squares are the coinciding defects.

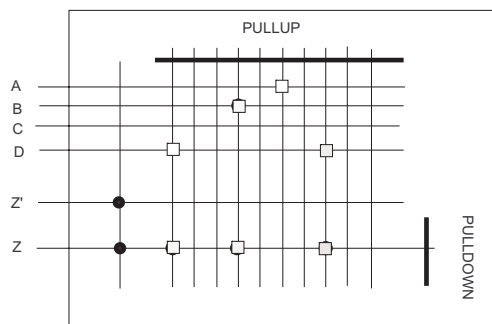


Fig. 3. PGMB with defective crosspoints.

The TH gates are mapped onto the diode crossbar structures on the AND and OR planes [8]. Columns can be shuffled to route away from the defects since the product terms are commutative; however shuffling the rows within the two planes is not allowed and is restricted to the respective planes.

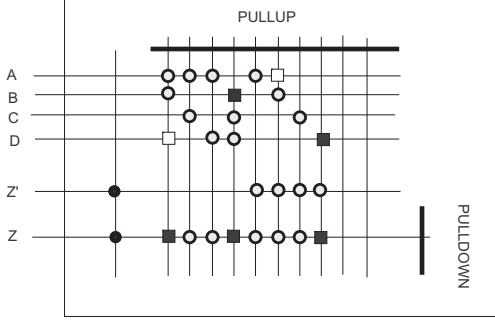


Fig. 4. TH34W2 on the defective PGMB. Black squares are ON-crosspoints with coinciding defects.

IV. NOMENCLATURE

The following notations will be used throughout this paper:

- 1) λ : Defect rate, $0 \leq \lambda \leq 1$.
- 2) m : Number of columns having ON crosspoints, since there might be unused columns in the crossbar.
- 3) n_i : Number of ON crosspoints in each column, where i indicates the i^{th} column.
- 4) k : Total number of defects in the given PGMB.
- 5) k_i : Number of defects in i^{th} column.
- 6) $P(k)$: Probability of mapping the given TH gate when there are exactly k coinciding defects in the PGMB.
- 7) $p(k')$: Probability of mapping a column of the PGMB when k' defects coincide with the ON crosspoints of the column.
- 8) $P_n(k_i)$: Probability of mapping a function when in one column k_i defects coincide with the ON crosspoints.
- 9) n : Total number of columns in the PGMB.
- 10) r : Total number of rows in the PGMB.
- 11) a : Number of defect free crosspoints.
- 12) b : Number of defective crosspoints.

V. MAPPING NCL GATES ON DEFECTIVE PGMBs

A. Modeling the mapping probability

Each PGMB has at least 6×10 dimension to program any given TH gate macro. The product terms are mapped on the AND plane and the horizontal wires add these 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 the ON crosspoints (m), the number of ON crosspoints in each column (n_i), where subscript i denotes the i^{th} column), the size of the PGMB and the defect rate (λ).

The defect rate indicates the percentage of defective crosspoints in a crossbar system. For instance, 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 successfully mapping the first column without any coinciding defect 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 $P(0) = (1 - \lambda)^{(3 \cdot 6)}$.

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

$$p(1) = (C_2^3 \cdot (1 - \lambda)^2 \cdot \lambda) \cdot \sum_{i=3}^0 C_i^3 \cdot (1 - \lambda)^i \cdot \lambda^{(3-i)} \quad (1)$$

Then, the probability of having one column with one coinciding defect among the given six columns is $P(1) = C_1^6 p(1) p(0)^5$.

The above calculations are specific to TH23 gate for a PGMB size of 6×10 . Since the number of programmable crosspoints per column (n_i) and the number of columns having programmable crosspoints (m) differ for each TH gate according to the change in PGMB dimensions, a general equation for overall probability can be given in terms of the above parameters as follows:

$$P(k) = \sum_{i=1}^n \left\{ \left(P_c^m \cdot C_{k_i}^{n_i} \cdot (1 - \lambda)^a \lambda^b \right) \cdot \sum_{i=1}^n C_{k_i}^{(r-n_i)} \cdot (1 - \lambda)^a \lambda^b \right\} \quad (2)$$

where m is the number of columns with ON-crosspoint(s), c is the number of columns affected by one or more defects, n_i is the number of ON-crosspoints in i^{th} column, k_i is the number of defects in i^{th} column, r is the total number of rows, a is the number of defect free crosspoints, and b is the number of defective crosspoints. Total number of rows and columns is represented by r and n respectively. Since the addition of the product terms is commutative, the columns can be added in a PGMB to obtain the optimal mapping. This provides us with 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. This is why the column-wise probability is calculated above.

B. Optimal PGMB dimension for gates

The probability calculation demonstrated above can be extended to find the optimum PGMB size for a given threshold gate. We have come up with an analysis to find the optimum PGMB size for mapping all the threshold gates. As mentioned previously, a PGMB of size 6×10 can be used to program any threshold gate, we start with this dimension for our calculations. The analysis presents three ways to increase PGMB size: increasing number of column, increasing number of rows, and increasing number of both row and columns. The three methods of increasing the size deliver different results and the optimum size is decided 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. Since the OR plane has more ON crosspoints on a single row, the first redundant row should be always added to the OR plane.

C. Critical Row Algorithm

The following algorithm, called critical row algorithm is used when two or more redundant rows are added to the given PGMB. 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 redundant rows available.

VI. PARAMETRIC SIMULATION AND RESULTS

A. Simulation and results for calculating mapping probability

Figure 5 shows how the probability of successful mapping changes with progression in defect rate. Various plots on the graph show the corresponding probabilities with no coinciding defect, with one coinciding defect, and so on, for defect rate from 1% to 10%. In case of one or higher coinciding defects, the probability of having defective mapping increases with defect rate.

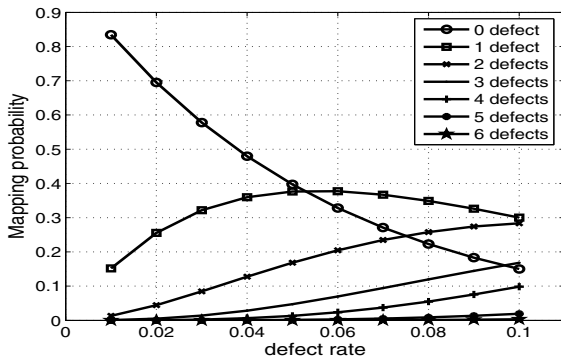


Fig. 5. Probability distribution curves for TH23 gate

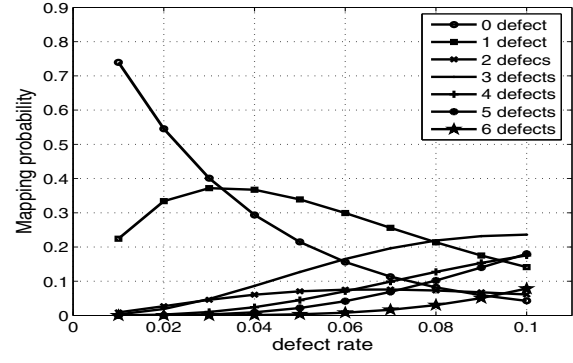


Fig. 6. Probability distribution curves for TH24 gate

Simulation results for TH24 are shown in figure 6. For the low defect rate range, both TH23 and TH24 have similar results. However, the two graphs differ substantially for the 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.

B. Results for row/column redundancy cases

The results of increasing PGMB size on the programming probability were analyzed for these five gates: TH22, TH33W2, TH23, TH44W322 and TH24 in three cases: increasing columns only, increasing rows only and increasing both rows and columns.. The corresponding number of ON crosspoints is 9, 15, 18, 24 and 30 respectively. For a defect rate of 10%, figure 7-a shows the successful mapping probability variation according to the PGMB size for these five gates when only columns are added. The probability increases with the increase in the PGMB size and is more obvious for the gates with lower ON crosspoint count.

The rows are added to either of the planes according to Critical Plane Algorithm. The results of the simulation are shown in figure 7-b.

As seen from the results, the increase in probability is more when rows are added than when columns are added. This is because the algorithm makes the best use of redundancy. For the 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 7-c. 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.

The graph in figure 8 shows the probability map for increasing both row and column count according to variation in probability. 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%. 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

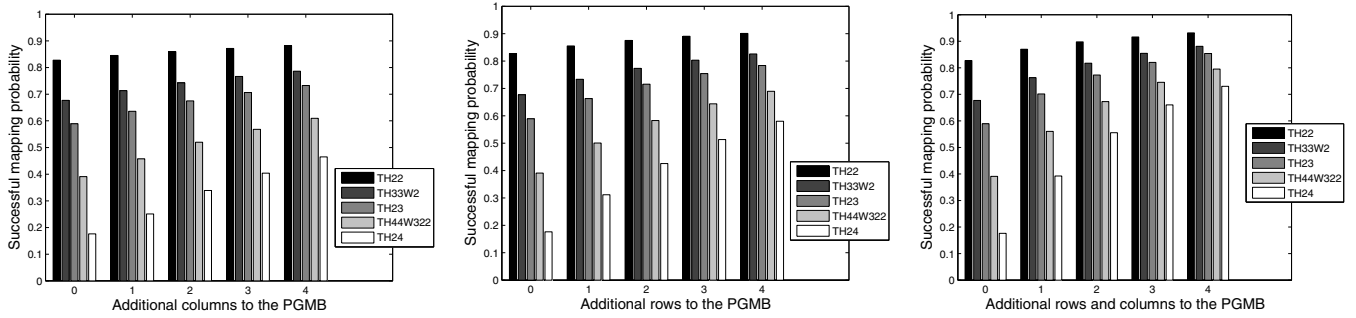


Fig. 7. Probability of successful mapping plots for: a) added columns, b) added rows and c) added rows and columns to PGMB.

crosspoints. As we increase the PGMB size and the probability increases in the corresponding defect plane.

For a particular gate and defect rate, the optimum size of PGMB blocks can be chosen according to the threshold of programmability desired. For example if a programmability of 80% is required at a defect rate of 10%, the optimum size for TH33W2 will be 8×12 . Thus the analysis provides a method to choose the best PGMB size to map all the gates, thus making the circuit more tolerant.

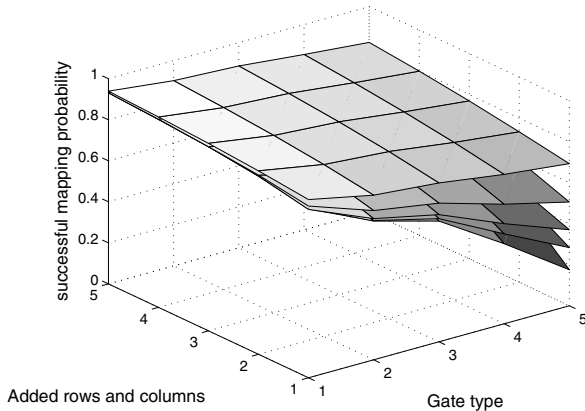


Fig. 8. Probability map for added rows and columns and varying defect rate

VII. CONCLUSION

Although the recently proposed asynchronous nanowire crossbar architecture offers better manufacturability, scalability, and robustness than its clocked counterpart, high defect rate due to nondeterministic nanoscale assembly is still one of the major issues that should be addressed. Thus, physical systems based on the clockless architecture should be designed, tested and repaired to maximize the 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.

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] R. Bonam, S. Chaudhary, Y. Yellambalase and M. Choi, "Clock-Free Nanowire Crossbar Architecture based on Null Convention Logic (NCL)", Accepted to appear in the 7th IEEE International Conference on Nanotechnology (IEEE-Nano), Apr 2007.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.