On the Defect Tolerance of Nano-scale Two-Dimensional Crossbars

Jing Huang, Mehdi B. Tahoori and Fabrizio Lombardi
Dept of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115
{hjing,mtahoori,lombardi}@ece.neu.edu

Abstract

Defect tolerance is an extremely important aspect in nano-scale electronics as the bottom-up self-assembly fabrication process results in a significantly higher defect density compared to conventional lithography-based process. Defect tolerance techniques are therefore essential to obtain an acceptable manufacturing yield. In this paper we investigate defect tolerance properties of a two-dimensional (2D) nano-scale crossbar, which is the basic block of various nano architectures which have been recently proposed. Various nano-wire and switch faults are studied and their impact on the routability of a crossbar are investigated. In the presence of defects, it is still possible to utilize a defective crossbar at reduced functionality, i.e. as a smaller defect-free crossbar. Simulation results for different sizes and defect densities are presented. This proposed approach can be utilized by architecture designers to determine the expected size of functional (defect-free) crossbar based on defect density information obtained from the fabrication process.

1 Introduction

Nano-scale devices and molecular electronics promise to overcome the fundamental physical limitation of lithography-based silicon VLSI technology. It is projected that molecular electronics can achieve density of 10^{12} devices per cm^2 and operate at THz frequencies [2]. Researchers have demonstrated several successful nano-scale electronic devices including *carbon nanotubes (CNT)* [1], *silicon nano-wires (NW)* [14][15], and *quantum dot cells* [19].

It has been shown that using bottom-up self-assembly techniques, it is possible to build nano devices without relying on lithography to define the smallest feature size [2]. Due to the fabrication regularity imposed by the chemical self-assembly process, it is unlikely that complex aperiodic circuits can be constructed. It is projected that only regular structures such as the two-dimensional (2D) crossbar can be manufactured. Reconfigurability is therefore an integral part of nanoelectronics as the circuit will be programmed into the desired functionality at post-fabrication. These architectures share some similarities with conventional Field Programmable Gate Arrays (FPGAs).

Regular programmable architectures for nano-scale devices are currently being investigated [5][6][11][20]. The 2D crossbar structure plays a central role in these architectures as it can act as memory cores, PLA planes (for programmable logic) and interconnects [5]. Several publications have stressed the need for defect tolerance for circuits realized by nano-scale devices [2][3][5][11], since the inherent non-determinism in bottom-up self-assembly chemical processes at molecular scale results in more defects compared to highly controlled lithography-based manufacturing processes currently used in CMOS technologies. The fully populated 2D crossbar provides defect tolerant capabilities; even in the presence of some defects, an $N \times N$ crossbar can be still used as a $K \times K$ ($K \leq N$) crossbar

Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'04)
1550-5774/04 $ 20.00 IEEE

at an architectural level. This approach is similar to conventional memory defect tolerant scheme which utilizes spare rows and columns. Efficient test and diagnostic techniques are also needed to identify and locate the defects and then reconfigurability is utilized to configure the circuit to bypass defective elements. These techniques are therefore integral parts of any defect tolerance scheme for nanotechnology.

In this paper, defect tolerance capabilities of a 2D crossbar is analyzed in detail. The impact of switch and nano wire faults on the routability of a crossbar is assessed. In this paper, *defect density* is defined as the number defective elements (nano-wires or switches) over the total number of elements. From the fabrication process information on the defect density of the crossbar can be obtained. A probabilistic approach based on the defect density level is used to find the maximum number of signals that can be simultaneously routed through the crossbar. So, instead of using the original $N \times N$ crossbar, a smaller crossbar configuration can be used at the architectural level. Results for crossbars of various sizes are reported under different types of faults. Our work establishes the expected capacity of a crossbar for a given defect density. It also provides guidelines for system design by assessing the level of redundancy required for a specific defect density.

The rest of the paper is organized as follows. In Sec. 2 a brief review of previous work as well as preliminaries are presented. The proposed method is described in Sec. 3. Experimental results are discussed and analyzed in Sec. 4. Finally, Sec. 5 concludes the paper.

2 Background

2.1 Previous Work

Researchers have demonstrated that CNTs, fullerene molecules, and semiconductor NWs, are all potential building blocks for circuits with significantly smaller dimensions than those achieved using a conventional process based on lithography patterns. Logic devices build with CNTs have been proposed in [1]. Semiconductor NWs (of a few nanometers in diameter and microns long) have been synthesized. The electrical properties of NWs can be controlled by dopants, yielding p-type or n-type semiconductors. Doped NWs also exhibit Field Effect Transistor (FET) properties and can be used for constructing active devices with gain [6]. Chemical self-assembly techniques can be used for aligning CNTs or NWs in parallel rows. Two layers of orthogonal NWs/CNTs can thus be built and transferred to a surface to construct a 2D crossbar. In [16], a suspended CNT crossbar memory has been proposed. This structure consists of a set of parallel CNTs suspended above another set of perpendicular CNTs on a substrate. At each crossing point, there is a programmable switch. The bottom layer can also be replaced by silicon NWs, in which the junction exhibits diode-like rectifying behavior [5]. Researchers at UCLA and HP have demonstrated that a molecular monolayer of rotaxanes sandwiched between metal NWs can be used to build an 8×8 crossbar [9]. Each crosspoint acts as a reprogrammable switch which can be used as a memory cell. These switches can be programmed using the signal lines by applying a higher voltage. Unlike conventional silicon systems, no dedicated programming circuit is needed.

The chemically assembled electronic nanotechnology (CAEN) FPGA-like architecture called *NanoFabric* has been proposed in [11]; this consists of an array of connected logic blocks (Nanoblocks). A 2D molecular array inside each Nanoblock provides reprogrammable resistor-diode logic since each crosspoint is a diode. Signal restoration is performed by using a molecular latch. The 2D crossbar is used for providing interconnection between the Nanoblocks. An array-based architecture which is similar to PLAs is proposed in [6]. This structure is also based on self-assembled crossed arrays of NWs with non-volatile diode switches at the intersections. Signal restoration is accomplished using NW FET devices as buffers. A stochastic approach to address individual NWs without using lithography-scale processing at nanoscale dimension, has been described in [7].

A general defect tolerance approach based on hierarchically sparing is discussed in [5]. In [6], manufacturing yield of the proposed array-based architecture is analyzed. This study is based on the NW defect density, estimating the yield of the stochastic decoder and the stochastic buffering. A

Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'04)
1550-5774/04 $ 20.00 IEEE

fault tolerance technique for asynchronous cellular array has been proposed in [17]. Error correcting code is utilized to achieve fault tolerance with respect to transient errors during system operation.

Testing and diagnosis are integral parts of fault tolerance techniques for reconfigurable molecular computing, as defective elements are required to be identified and located before reconfigurability can be utilized to configure the circuit to bypass faults. Testing and diagnosis techniques for reconfigurable molecular computing have been addressed in [18].

2.2 Preliminaries

In this paper, a fully populated $N \times N$ crossbar (such as those demonstrated in [16][9]) as the interconnect network, is considered. An example of a 4×4 crossbar is shown in Figure 1(a). The crossbar consists of two sets of orthogonal nano-wires; the horizontal nano-wires are the inputs while the vertical nano-wire are the outputs. There is a programmable switch at each intersection (crossing), which connects the corresponding input to an output when it is programmed "on". When the switch is in the "off" state, it is assumed that the horizontal nano-wires can not influence the vertical nano-wires. The crossbars used here are assumed to have no gain. Signal restoration can be obtained by adding a buffer stage of doped NWs (which act as field effect transistors) at the output of the crossbar [6].

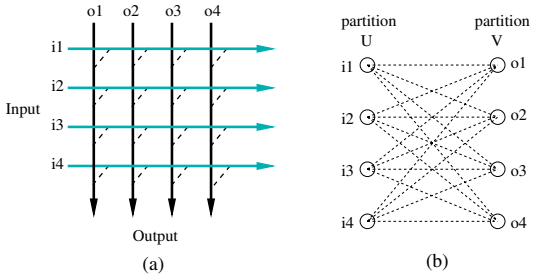


Figure 1. 2D Nano-scale Crossbar and Bipartite Graph Representation

The 2D crossbar can be represented by a *bipartite* graph $B = (U, V, E)$. The partition U represents the input nano-wires, while the partition V represents the output nano-wires; E represents the programmable switches in the crossbar, as illustrated in Figure 1(b). A *matching* T is a set of edges such that no two edges share the same vertex. If an edge (v_i, v_j) is in the matching, then vertices v_i and v_j are said to be matched. A *perfect* matching of a graph is a matching such that all vertices are matched. A matching T of size K corresponds to the K signals that can be routed through the crossbar simultaneously. Therefore the maximum matching of a bipartite graph B represents the so-called routing *capacity* of the crossbar. In the fault-free case, a perfect matching exists in a fully populated crossbar and thus N input signals can be routed to N outputs. However, in the presence of defects, certain nano-wires or switches might become unusable and routability may drop. When the defect density is sufficiently high, the probability of finding a perfect matching will be small. However, the crossbar can still be used as a $K \times K$ ($K \leq N$) crossbar if a matching of size K can be found with a high probability. In this case, signals can be routed from the K inputs to the K outputs. The address decoder similar to the one proposed in [7] can be used for selecting fault-free nano-wires to carry valid signals.

The proposed technique identifies the probability of finding a $K \times K$ (so that K inputs can be routed to K outputs) crossbar out of a faulty $N \times N$ ($K \leq N$) crossbar at a specified defect density level. The following Metric is used as figure of merit throughout this paper:

Metric $M^d_{N,K}$: *The probability of finding a matching of size K in an $N \times N$ crossbar when the defect density is d.*

In this paper, both switch faults and nano-wire faults are considered. Switch faults include switch stuck-open faults and switch stuck-closed faults: a *stuck-open* fault means that the switch is permanently disconnected; a switch *stuck-closed* fault causes the two endpoints of the switch to be shorted. Nano-wire faults including open faults and bridging faults are also considered: a

Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'04)
1550-5774/04 $ 20.00 IEEE

nano-wire *open* fault refers to a broken nano-wire which is disconnected; a nano-wire *bridging* fault causes two adjacent nano-wires in the same layer to be shorted together. This paper investigates the manufacturing yield analysis of nano-scale crossbars. Therefore only permanent faults are considered here. Transient faults happen in the system operation and are outside the scope of this paper.

3 Proposed Method

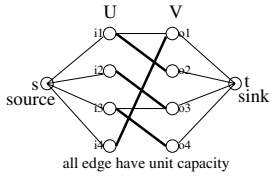


Figure 2. Maximum Flow Algorithm

First consider the problem of finding the maximum matching in an arbitrary bipartite graph $B = (U, V, E)$, $|U|=|V|=N$, $|E|=e$. This can be done by using the maximum flow algorithm [4]. A bipartite graph with $|U|=|V|=4$ is illustrated in Figure 2, a maximum (also perfect in this case) matching is highlighted. we add a source vertex s and a sink vertex t to the original bipartite graph and construct the corresponding flow network $B_f = (V_f, E_f)$ (Figure 2), such that: $V_f = U \cup V \cup \{s, t\}$ and $E_f = \{(s, u) : u \in U\} \cup E \cup \{(v, t) : v \in V\}$. Then we set unit capacity for all edges in E_f. It has been proved [4] that a maximum matching in B corresponds directly to a maximum flow in B_f. The algorithm to find the bipartite matching problem can be found in [4] with time complexity $O(\sqrt{|V|}|E|) = O(\sqrt{N}N^2)$=$O(N^{\frac{3}{2}})$.

In a fully populated $N \times N$ crossbar, there are altogether $2N$ nano-wires and N^2 switches. In the case of switch faults, the defect density is d and $\lceil N^2 d \rceil$ switch faults can be present. The number of possible fault patterns is given by the binomial relation $\binom{N^2}{\lceil N^2 d \rceil}$, which is clearly exponential to the size of the graph. This can be resolved by considering only a sample of all possible fault patterns. In this paper, $FP_N^{d,total}$ fault patterns are randomly generated, for an $N \times N$ crossbar with defect density d. If $FP_{N,K}^d$ of these fault patterns will result in no matching of size bigger or equal to K, then Metric $M_{N,K}^d$ can be calculated as:

$$M_{N,K}^d = 1 - \frac{FP_{N,K}^d}{FP_N^{d,total}} \tag{1}$$

Switch stuck-open faults: The construction of the bipartite graph corresponding to the faulty crossbar in the presence of switch stuck-open faults is shown next. Faults are introduced to the original (fault-free) bipartite graph $B = (U, V, E)$ to obtain a new (faulty) bipartite graph $B' = (U, V, E')$. A switch stuck-open fault corresponds to a missing switch at the crosspoint and the faulty bipartite graph B' can be obtained by simply deleting from B the edges corresponding to the faulty switches between the two partitions. Obviously $|E'| < |E| = N^2$. The algorithm outlined below finds the metric $M_{N,K}^d$, which is the probability of finding a matching of size K or bigger in an $N \times N$ crossbar with defect density d.

Function Permutation_open(B,d,K)
input: bipartite graph $B = (U, V, E)$ representing fault-free switch block
$|U|=|V|=N$, $|E|=N^2$
defect density d
required minimum matching size K
number of faults $j \leftarrow \lceil N^2 d \rceil$
randomly generate $FP_N^{d,total}$ fault patterns
$FP_{N,K}^d \leftarrow 0$

Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'04)
1550-5774/04 $ 20.00 IEEE

for each possible fault pattern with j faults **loop**
    introduce the fault pattern to B to generate B'
    find maximum matching of B'
    if matching size $< K$ **then**
        $FP^d_{N,K} \leftarrow FP^d_{N,K} + 1$
return 1-$(FP^d_{N,K}/FP^{d,total}_N)$

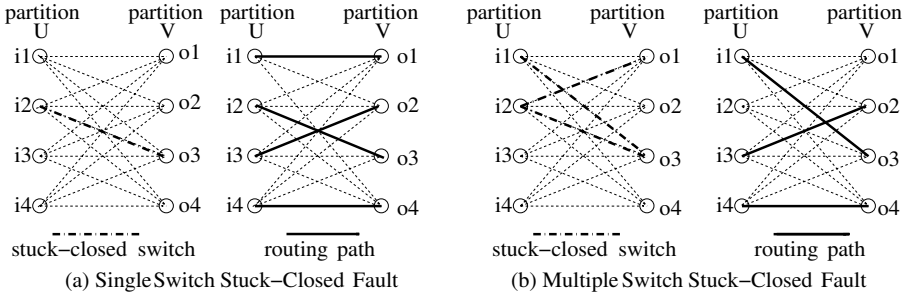


Figure 3. Signal Routing in the Presence of Switch Stuck-closed Faults

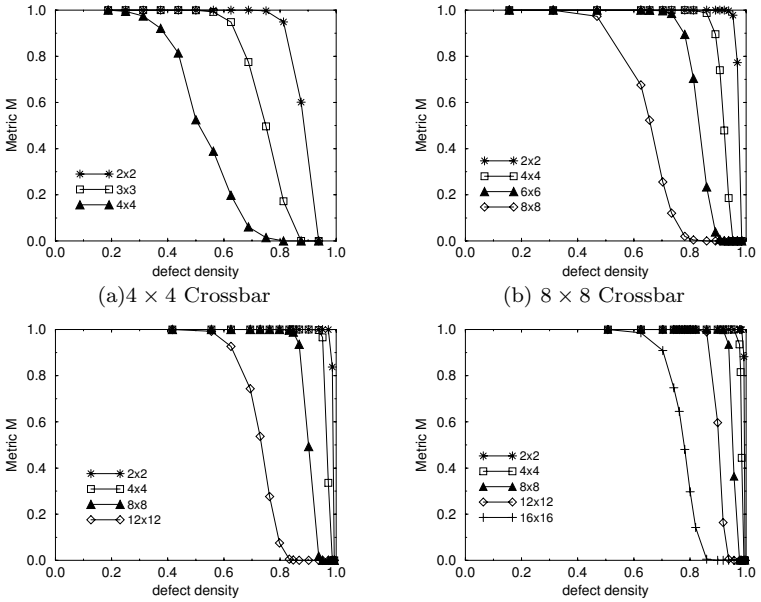
Switch stuck-closed faults: First assume the case of a single switch stuck-closed fault. An example is shown in Figure 3(a) with the input nano-wire $i2$ shorted to the output nano-wire $o3$. In this case, if $i2$ is used for routing the signals through the crossbar, the signal can be only routed to $o3$. In the presence of this fault, it is still possible to route four signals through the crossbar, with the constraint that $i2$ must be routed to $o3$. A routing arrangement is shown in Figure 3(a): $i1 \rightarrow o1$, $i2 \rightarrow o3$, $i3 \rightarrow o2$, $i4 \rightarrow o4$. When multiple switch stuck-closed faults are present, a group of nano-wires (input and output wires) can be shorted together. Obviously, the group can be used to route at most one signal. An example of multiple switch faults is illustrated in Figure 3(b) with three switch stuck-closed faults. Here, one group of resources (the input nano-wires $i1$,$i2$ and the output nano-wires $o1$,$o3$) are shorted together, and at most one signal can be routed with this group. For example, this group can route $i1$ to $o3$. Consider the routability of the fault-free resources. This is done by deleting all resources in the shorted group (i.e. $i1, i2, o1, o3$), and using the remaining fault-free resources. In this example, it is still possible to route two signals through the crossbar, for example $i3 \rightarrow o2$ and $i4 \rightarrow o4$. So, altogether three signals can be routed through the faulty crossbar, as shown in Figure 3(b).

In general, with multiple switch stuck-closed faults, the groups of resources that are shorted together must be identified first. This corresponds to the maximum number of signals, K_1, that can be routed through these faulty resources. Next, the faulty bipartite graph B' is obtained from the original bipartite graph B by deleting all vertices (nano-wires) involved in short faults. The maximum flow algorithm can be used for finding the maximum number of signals K_2 that can be routed by fault-free elements. The maximum number of signals that can be routed through the faulty crossbar is given by $K = K_1 + K_2$. Based on this approach an algorithm similar to the one described above can be used to find the metric $M^d_{N,K}$.

Nano-wire open and bridging faults: In some architectures, precharge/evaluation logic is used [6][7]. As a result, broken nano-wires are of no use at all. In this work we consider nano-wires with open faults to be completely defective, so they can not be used to carry signals. So B' is obtained by deleting the faulty vertices (broken nano-wires) from B. In a nano-wire bridging fault, two adjacent nano-wires (both inputs or both outputs) are shorted together. Obviously, one of the nano-wires becomes unusable. For multiple bridging faults, only one of the nano-wires in a group of shorted nano-wires can be used to carry signals. When constructing the faulty bipartite graph B', we delete all but one nano-wire in each nano-wire bridging group. The maximum flow algorithm can then be used to find the maximum number of signals that can be routed within the faulty crossbar.

Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'04)
1550-5774/04 $ 20.00 IEEE

4 Experimental Results



(a)4×4 Crossbar (b) 8×8 Crossbar

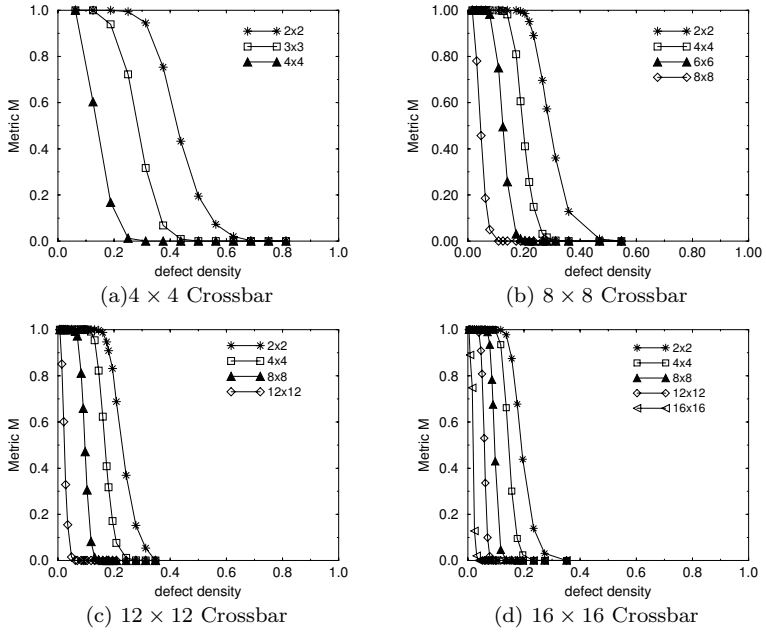
(c) 12×12 Crossbar (d) 16×16 Crossbar

Figure 4. Results for Switch Stuck-Open Fault

Experimental results on all four types of faults addressed in the previous section and their effects on the routability of the crossbar are presented. The following sized crossbars are discussed in this paper: 4×4, 8×8, 12×12 and 16×16 ($N = 4, 8, 12, 16$). An $N \times N$ crossbar consists of $2N$ nano-wires and N^2 switches. Recall that the defect density refers to the percentage of faulty elements. For each crossbar of size N (with a specific type of fault), the metric $M^d_{N,K}$ versus defect density for different K ($K \leq N$) are presented in Figure 4 to Figure 7. For example, the results for switch stuck-open faults are shown in Figure 4. The plots for a 16×16 crossbar can be found in Figure 4(d), in which the probability of routing $2, 4, 8, 12$ and 16 signals through the crossbar versus the defect density is plotted. As per the simulation results, if 80% of the switches are stuck-open, then the probability of using the crossbar as a 16×16 crossbar is rather small, i.e. less than 20%. However we can most probably still use it as a 12×12 crossbar. This means that if we need a 12×12 crossbar for a specified design, then by providing redundancy and constructing a 16×16 crossbar (even if the defect density of switches is high) the yield can still be almost 100%.

Similarly, the results for switch stuck-closed faults are shown in Figure 5. Clearly, switch stuck-closed faults have a much more severe impact on the routability than switch stuck-open faults, for the same defect density. As an example, for a 12×12 crossbar with 50% of the switches having stuck-open faults, it is still possible to route 12 signals over 95% of the times. However with 50% of the switches being stuck-closed faults, most likely it is not possible to use the crossbar even as a 2×2 crossbar. This can be explained as follows. A switch stuck-open fault is equivalent to a missing switch. The nano-scale crossbar considered in this paper are fully populated, so even with a large percentage of missing switches, the probability of having enough fault-free switches to route signals is sufficiently high. However, multiple switch stuck-closed faults cause nano-wires to be shorted together. For example consider a 4×4 crossbar as shown in Figure 1. Assume four switches, $(i1, o1)$,$(i1, o2)$,$(i1, o3)$,$(i1, o4)$, are stuck-closed. Then in effect all output nano-wires are shorted together, so only one signal can be routed through this crossbar. On the contrary, in the case of four stuck-open switch faults, at least three signals can be routed through this crossbar.

Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'04)
1550-5774/04 $ 20.00 IEEE



(a)4 × 4 Crossbar (b) 8 × 8 Crossbar

(c) 12 × 12 Crossbar (d) 16 × 16 Crossbar

Figure 5. Results for Switch Stuck-Closed Fault

One interesting observation is that for the same defect density, large crossbars are more (less) defect tolerant than small ones in the presence of switch stuck-open (stuck-closed) faults. In the fabrication process, it is possible to tailor the chemical synthesis technique to decrease the probability of having switch stuck-closed faults at the expense of increasing the total number of switch faults [11]. In this case we expect to have a much higher defect density of stuck-open faults than stuck-closed faults, thus resulting in a better defect tolerance for large crossbars.

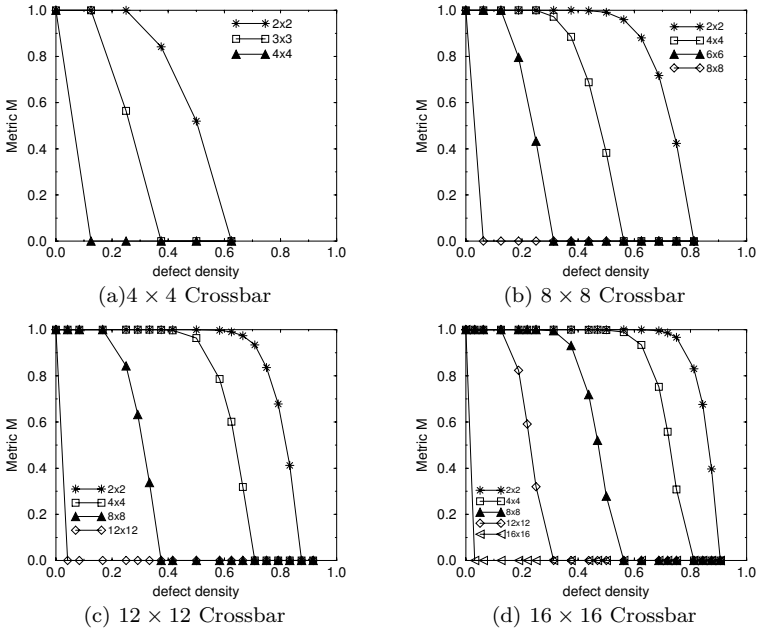
In the presence of an open defect in a nano-wire, it is assumed that the entire nano-wire is unusable, because precharge/evaluation logic is employed in some architectures. When two adjacent input (output) nano-wires are bridged, one of them is unusable. Therefore the results for nano-wire open and nano-wire bridging faults are similar, as shown in Figure 6 and Figure 7 respectively. Since a nano-wire open or bridging fault makes the entire nano-wire unusable, it has a more severe impact on routability than a switch stuck-open fault.

Currently the value of the defect density level that will be expected from a chemical self-assembly process is not clearly understood, although it is known it will be much higher than a conventional CMOS lithography-based process [2]. Researchers at UCLA-HP have reported that only $15-20\%$ of the manufactured junctions work properly for the nano-scale 2D crossbar [10]. In [13], it is suggested that 10% of all fabricated resources, including switches, nano-wires and logic elements, are expected to be defective. From the simulation results, it can be seen that fully populated crossbars are quite defect tolerant to switch stuck-open faults. Extra redundancy can be provided by using an $N \times N$ crossbar as a $K \times K$ $(K < N)$ crossbar, such that even with a high defect density ($\sim 70\%$) an acceptable yield can be still achieved.

5 Conclusion

Nano-technology is expected to develop and yield molecular sized electronics that operate at THz frequencies. Bottom-up chemical self-assembly processes will be used to achieve significantly smaller feature sizes than those obtained by lithography. However these processes are expected to have a

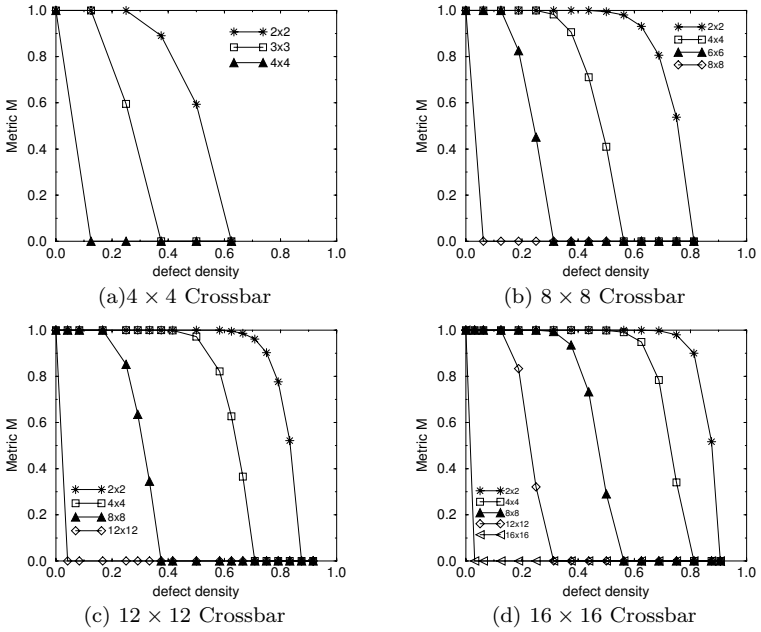
Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'04)
1550-5774/04 $ 20.00 IEEE



(a)4 × 4 Crossbar (b) 8 × 8 Crossbar

(c) 12 × 12 Crossbar (d) 16 × 16 Crossbar

Figure 6. Results for Nano-wire Open Fault



(a)4 × 4 Crossbar (b) 8 × 8 Crossbar

(c) 12 × 12 Crossbar (d) 16 × 16 Crossbar

Figure 7. Results for Nano-wire bridging fault

significantly higher defect density than that for conventional top-down patterning processes. Defect tolerant techniques are therefore essential for nano-electronic systems to increase the manufacturing yield. In this paper, defect tolerance of the 2D crossbar, which is used as the basic building block for logic, interconnect and memory in various nano architectures, has been studied. Switch stuck-open

Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'04)
1550-5774/04 $ 20.00 IEEE

and stuck-closed faults, as well as nano-wire open and bridging faults have been considered. A new metric is proposed, which indicates the probability of using an $N \times N$ crossbar as a smaller $K \times K$ crossbar for a given defect density. Simulation results of the proposed metric versus defect density for various sized crossbars with the four different types of faults have been presented. It can be concluded that in general, switch stuck-closed faults have a significantly higher impact on the manufacturing yield compared to switch stuck-open faults. This fact suggests that it is beneficial to bias the chemical synthesis process during manufacturing to decrease the probability of having switch stuck-closed faults, even at the expense of increasing the total number of switch faults. Our work gives the architecture designer an indication about the expected number of signals that can be routed through a crossbar based on the fabrication defect density. In addition, effective testing and diagnostic techniques are needed to identify defective elements such that they can be bypassed by reconfiguration.

References

[1] A.Bachtold, P.Harley, T.Nakanishi, C.Dekker, "Logic Circuits with Carbon Nanotube Transistors" *Science*, vol. 294, pp 1317-1320, 2001.

[2] M.Butts, A.DeHon, S.C.Goldstein, "Molecular Eletronics: Devices, Systems and Tools for Gigagate, Gigabit Chips", *Proc. International Conference on Computer-Aided Design*, pp . 443-440, 2002.

[3] C.P. Collier, E.W. Wong, M.Belohradsky, F.M.Raymo, J.F.Stoddart, P.J.Kuekes, R.S.Williams, J.R.Heath, "Electronically Configurable Molecular-Based Logic Gates", *Science*, vol 285, pp 391-394, 1999.

[4] T.H.Cormen, C.E.Leiserson, R.L.Rivest, C.Stein "Introduction to Algorithms", *McGraw-Hill* , pp. 643-693, 2001.

[5] A.DeHon, "Array-Based Architecture for FET-Based, Nanoscale Electronics", *IEEE Trans. on Nanotechnology*, vol 2, No. 1, pp. 23-32, 2003.

[6] A.DeHon, M.J.Wilson, "Nanowire-Based Sublithographic Programmable Logic Arrays", *Proc. International Symposium on Field-Programmable Gate Arrays (FPGA2004)*, pp. 123-132, 2004.

[7] A.DeHon, P.Lincoln, J.E.Savage, "Stochastic Assembly of Sublithographic Nanoscale Interfaces", *IEEE Trans. on Nanotechnology*,vol 2, No. 3, pp. 165-174 , 2003.

[8] A.DeHon, "Law of Large Numbers System Design", *Nano, Quantum and Molecular Computing: Implications to High Level Design and Validation* pp. 213-241, 2004.

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

[10] Y.Luo, C.P.Collier, J.O.Jeppesen, K.A.Nielsen, E.Delonno, G.Ho, J.Perkins, H.R.Tseng, T.Yamamoto, J.F.Stoddart, J.R.Heath. "Two-dimensional Molecular Electronics Circuits", *Chemphyschem*, vol 3(6), pp. 519-25, 2002.

[11] S.C.Goldstein, M.Budiu, "NanoFabrics: Spatial Computing Using Molecular Electronics", Proc. International Symposium on Computer Architecture, pp. 178-189, 2001

[12] S.C.Goldstein, D.Rosewater, "Digital Logic Using Molecular Electronics", *IEEE International Solid-State Circuits Conference, Digest of Technical Papers*, vol 2, 2002.

[13] M.Mishra, S.C.Goldstein, "Defect Tolerance at the End of the Roadmap", *Proc. International Test Conference*, pp . 1201-1211, 2003,

[14] Y.Cui, C.M.Lieber, "Functional Nanoscale Electronics Devices Assembled Using Silicon Nanowire Building Blocks", *Science*, vol 291, pp . 851-853, 2001.

[15] Y.Huang, X.Duan, Y.Cui, L.J.Lauhon, K.H.Kim, C.M.Lieber, "Logic Gates and Computation From Assembled Nanowire Building Blocks", *Science*, vol 294, pp . 1313-1317, 2001.

[16] T.Rueckes, K.Kim, E.Joselevich, G.Y.Tseng, C.L.Cheung, C.M.Lieber, "Carbon Nanotube-based Nonvolatile Random Access Memory for Molecular Computing", *Science*,vol 289(5476), pp . 94-97, 2000.

[17] F.Peper, J.Lee, F.Abo, T.Isokawa, S.Adachi, N. Matsui, S.Mashiko, "Fault-tolerance in nanocomputers: a cellular array approach", *IEEE Trans. on Nanotechnology*,vol 3, No. 1, pp. 187- 201, 2004.

[18] M.B.Tahoori, S.Mitra, "Defect and Fault Tolerance in Reconfigurable Molecular Computing", *Field Custom Computing Machines (FCCM)*, 2004.

[19] P.D.Tougaw, C.S. Lent, "Logical Devices Implemented Using Quantum Cellular Automata", *Applied Physics*, vol 75(3), pp 1818-1825, 1994.

[20] M.M.Ziegler, M.R.Stan, "Design and analysis of crossbar circuits for molecular nanoelectronics", *Proc. IEEE International Conference on Nanotechnology*, 2002.

Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'04)
1550-5774/04 $ 20.00 IEEE