Logic Level Fault Tolerance Approaches Targeting Nanoelectronics PLAs

Wenjing Rao
UC San Diego
CSE Department
wrao@cs.ucsd.edu

Alex Orailoglu
UC San Diego
CSE Department
alex@cs.ucsd.edu

Ramesh Karri
Polytechnic University
ECE Department
rkarri@poly.edu

ABSTRACT

A regular structure and capability to implement arbitrary logic functions in a two-level logic form have placed crossbar-based Programmable Logic Arrays (PLAs) as promising implementation architectures in the emerging nanoelectronics environment. Yet reliability constitutes an important concern in the nanoelectronics environment, necessitating a thorough investigation and its effective augmentation for crossbar-based PLAs. We investigate in this paper fault masking for crossbar-based nanoelectronics PLAs. Missing nanoelectronics devices at the crosspoints have been observed as a major source of faults in nanoelectronics crossbars. Based on this observation, we present a class of fault masking approaches exploiting logic tautology in two-level PLAs. The proposed approaches enhance the reliability of nanoelectronics PLAs significantly at low hardware cost.

1. INTRODUCTION

As device scales shrink, traditional CMOS based devices are facing physical limits due to quantum effects and fabrication challenges occurring at the nano scale. A number of nanoelectronic devices such as SET [1], RTD [2], Carbon Nanotubes [3], QCA [4] and molecular electronics [5], are proposed as promising device candidates for the next generation nanoelectronics [6].

A number of new opportunities and challenges emerge in the nanoelectronic devices. The new nanoelectronics promise to deliver device densities of up to 10^{12}device/cm^2, thus indicating a huge advantage in terms of hardware abundance [6]. On the other hand, the traditional top-down fabrication process becomes exceedingly expensive for the increasingly shrinking nano devices. The only possible method of precisely and economically fabricating a nanoelectronics based system is the bottom-up self-assembly process, which is limited to the generation of regular structures, thus necessitating a post-fabrication reconfiguration process to impose the desired functionality [7].

Perhaps one of the most severe challenges in the nanoelectronic system is unreliability, due to their nano scale dimensions [6]. Not only are manufacturing defect rates projected to be extremely high, but also nanoelectronic systems suffer from a significantly increased occurrence rate of run-time faults. Consequently, extensive fault tolerance schemes are necessitated for future nanoelectronics based systems to guarantee the fundamental correctness requirement.

The abundant hardware resources supported by the nanoelectronic devices can be exploited for fault tolerance purposes. These fault tolerance schemes typically utilize redundant hardware to mask the fault effect, thus achieving the correct result in spite of fault occurrence, at the expense of hardware overhead, while avoiding significant performance loss.

In this paper we focus on the fault tolerance issue for a nano crossbar based PLA structure. Such a crossbar based PLA structure is quite promising for the nanoelectronic logic system, since its highly regular structure can be easily fabricated with the bottom-up self-assembly process and its implementation is shown to be supported by multiple nanoelectronic devices [8, 9, 10].

Based on the major fault type that has been identified up to now in the crossbar based PLA, the missing device fault at the crosspoints in a crossbar structure [9, 10], we develop a specific genre of fault tolerance schemes that exploits the logic functionality of the PLA structure so as to avoid the tremendous hardware requirements of a TMR scheme.

2. MOTIVATION

The bottom-up self-assembly fabrication process of nanoelectronic systems results in a regular system structure. Extensive research work has been carried out for regular structure based, particularly crossbar based logic, nanoelectronic systems [11, 12, 13, 14, 15]. A PLA like crossbar based logic has been shown to be a quite promising candidate for nanoelectronic systems for the following reasons. First, a PLA logic is highly regular in its structure; therefore, it can be easily manufactured with the bottom up self-assembly process in nanofabrication. Second, the reconfigurability supported by nanoelectronic devices can be effectively utilized in a PLA logic to form the logic functions, thus supporting arbitrary logic functions in the two-level logic form. Also, it is shown that in certain two terminal nanoelectronic devices, such as molecular devices, it is hard to construct inverter gates, thus necessitating the inversion of a signal through switching back to CMOS level inverters [15]. With an AND/OR plane based 2-level PLA, an logic function can be implemented without inversion as long as the complement forms of all the input signals are given. Due to this reason, the switching back and forth between nanoelectronic and CMOS transistors for signal inversion in a combinational logic is eliminated, thus avoiding tremendous performance overhead. Consequently, PLA logic is highly compatible with multiple nanoelectronic device candidates including the two terminal devices.

It is widely acknowledged that reliability constitutes one of the most challenging issues in nanoelectronic system construction. First, manufacturing defects increase significantly, because the fabrication process in nano environments is prone to defects due to the small scale of devices and the bottom-up self-assembly process. In comparison with the defect rates of 10^{-9} to 10^{-7} in current CMOS systems, the defect rates of nanoelectronic systems are projected to be extremely high, of the order of 10^{-3} to 10^{-1} [16]. Second, a high occurrence of online faults is expected during run-time [17]. This is essentially caused by device scales and the low voltage uti-

978-3-9810801-2-4/DATE07 © 2007 EDAA

lized in nano transistors, which result in extremely high sensitivity to environmental influences, such as temperature, cosmic ray particles and background noise.

In fact, online faults have been observed increasingly in current CMOS based systems as the device scales down to the deep submicron stage. Single event upset caused by cosmic particles has already been observed in large amounts in memory systems and sequential logic state elements. The ultra low power utilized as well as the quantum effects nanoelectronic devices rely on, both result in significantly reduced noise margins and increased sensitivity to environmental effects. Therefore, a significant number of online faults are expected to be triggered due to variances in temperature, cosmic particles, background noise, and crosstalk effects [17, 18]. Due to the highly unreliable devices which are extremely sensitive to environmental influences, online fault tolerance is of significant importance for guaranteeing the basic correctness requirement of a nanoelectronics based system.

Fault models for PLAs have been developed early in the research history of PLA testing methodologies [19]. Research on reliability issues of PLAs heretofore mainly targets the issues of manufacturing defects, by performing reconfiguration to bypass the defective devices in a post-fabrication process [10, 20, 21]. However, the research on online fault tolerance schemes for PLAs, particularly in a high fault rate environment, has not been well investigated. Previous related research mainly focuses on online fault detection techniques [22, 23, 24], leaving the issue of fault tolerance unresolved.

Fault tolerant computation typically explores redundancy to guarantee correctness in the presence of faults. A hardware redundancy based approach, generally applicable to all computational systems, is particularly supported in the nanoelectronic environment and is advantageous in terms of performance. These approaches typically utilize straightforward hardware redundancy to mask the occurrence of faults, thus introducing very low performance sacrifice. Perhaps one of the best known examples of traditional fault masking approaches is the N-modular redundancy (NMR) scheme, which utilizes identical redundant modules to execute the same function and a majority voter to mask out faulty outputs.

In the crossbar structure nanoelectronic environment, where a massive number of two-terminal molecular devices are sandwiched at the crosspoint of two perpendicular nanowires, it has been observed that the dominant occurrence of fault behavior is the device missing effect [9, 10]. A missing device fault at the crosspoint of a PLA structure results in distinct effects depending on the location of the fault [25]:

- When a missing device fault occurs in the AND plane, a variable is dropped from a product term and the outputs connected to the product term change unidirectionally from 0's to 1's.
- When a missing device fault occurs in the OR plane, a product term is dropped and the outputs connected to the dropped product term change unidirectionally from 1's to 0's.

Table 1 exhibits the two types of manifestations of the missing device fault.

When focusing on the dominant missing device fault, it is possible to develop, according to the characteristics of nano PLA, techniques that utilize highly reduced hardware while providing an efficient fault tolerance capability. In the following sections we propose a number of fault tolerance approaches for the nanoelectronic PLAs.

3. FAULT TOLERANCE IN NANO PLA

location	example	fault effect in logic	K-map	output
AND plane	$f = ab + cd$ $f' = b + cd$	missing a variable (a) in a product term (ab)	growth	$0 \rightarrow 1$
OR plane	$f = ab + cd$ $f' = cd$	missing a product term (ab) in a logic output (f)	disappearance	$1 \rightarrow 0$

Table 1: Missing device fault manifestation in nano PLA

Hardware redundancy based fault masking is a general fault tolerance technique that can be applied straightforwardly to arbitrary functions. In the traditional NMR based fault tolerance approach, to achieve the single fault masking capability, at least triple the amount of hardware is required, plus the additional overhead of a majority voter. However, for the device missing fault in the nanoelectronic PLAs, a class of new fault masking schemes can be developed by integrating the redundancy within the logic function, thus necessitating no majority voting process and significantly reduced hardware overhead.

Consider the tautology of an AND Boolean function:

$$f = ab \equiv aabb = \widehat{f_{AND}}$$

$\widehat{f_{AND}}$ is logically equivalent to f, but is able to mask any single variable dropping fault with double the amount of hardware. Similarly, any single variable dropping fault in the OR Boolean function can be masked with a redundant form of logic with double hardware:

$$f = a + b \equiv a + a + b + b = \widehat{f_{OR}}$$

In the examples above, by exploiting the tautology in Boolean logic functions, fault tolerance can be achieved with double the amount of hardware, and without any dedicated voting hardware.

However, for a sum-of-product (SOP) two level Boolean logic[1], where both variable and product term dropping faults are considered, the different fault manifestation directions in the AND function and the OR function raise difficulties for hardware efficiency:

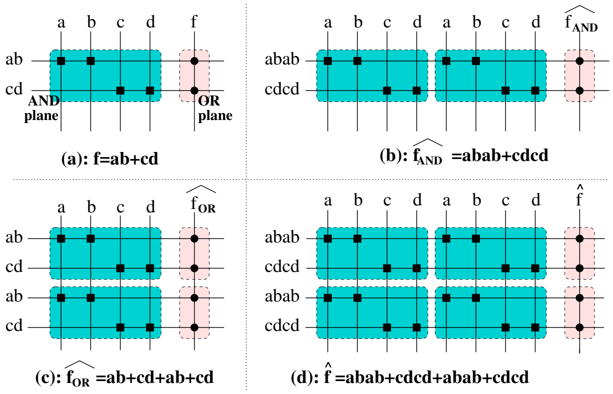
Example two-level logic function: $f = ab + cd$
Masking variable dropping: $\rightarrow \widehat{f_{AND}} = abab + cdcd$
Masking product term dropping: $\rightarrow \widehat{f_{OR}} = ab+cd+ab+cd$

It can be observed that $\widehat{f_{AND}}$ and $\widehat{f_{OR}}$ each masks one type of dropping fault, yet is susceptible to the other type of dropping fault. Specifically, any variable dropping in $\widehat{f_{OR}}$ and any product term dropping in $\widehat{f_{AND}}$ result in an erroneous function. In order to mask the device dropping faults in both the AND and the OR planes of a two-level PLA logic, the following tautology needs to be utilized:

$$f = ab + cd \equiv \widehat{f} = abab + cdcd + abab + cdcd$$

Figure 1 shows the fault tolerance scheme in a PLA structure for the missing device fault. Specifically, figure 1(a) shows the original logic function of $f = ab + cd$. Figure 1(b) shows the tautology of $\widehat{f_{AND}} = abab + cdcd$ with duplicated input variable wires and a duplicated AND plane. Figure 1(c) shows the tautology of $\widehat{f_{OR}} = ab+cd+ab+cd$ with duplicated product term wires and a duplicated OR plane. Although the vantage point of a tautological logic form indicates that $\widehat{f}$ necessitates a quadrupling in the number of logic variables and product terms, the hardware overhead is significantly diminished when implemented in the PLA structure. It

[1] We perform our discussion based on the sum-of-product form of two-level logic, without loss of generality to its duality, the product-of-sum form.



(a): f=ab+cd

(b): $\widehat{f_{AND}}$ =abab+cdcd

(c): $\widehat{f_{OR}}$ =ab+cd+ab+cd

(d): $\hat{f}$ =abab+cdcd+abab+cdcd

Figure 1: Fault masking two-level PLA without majority vote

can be seen from figure 1(d) that, device-wise, such an approach requires a quadrupling in the number of devices in the original AND plane and a doubling in the number of devices in the original OR plane. In terms of wire hardware overhead, this approach demands a doubling both in the number of variable wires and product term wires, but requires no extra logic output wires.

4. VARIATIONS IN TAUTOLOGY

It has been demonstrated in [13] that multi-level logic solely within the nano layer with no additional access to the CMOS layer can be implemented in the crossbar based nanoelectronic PLAs. By exploiting the capability of using separate planes for multi-level logic in the nanoelectronic PLAs, the proposed fault tolerance scheme can be extended into a class of tautology based variations, where multiple tradeoffs in performance and hardware overhead can be exploited according to the parameters of the logic function implemented in the PLA.

4.1 A-O-O: the 3-level logic approach

In the previous subsection, fault tolerance is essentially achieved by the formula of $\widehat{f} = abab + cdcd + abab + cdcd$. By extending one more logic level in the PLA structure, such a fault masking effect can be alternatively achieved through the equivalent formula $\widehat{f} = \widehat{f_{AND}} + \widehat{f_{AND}} = (abab + cdcd) + (abab + cdcd)$.

Figure 2 illustrates an example of this approach. It is easy to see, both from the figure and from the formula, that an additional level of logic is necessitated in this situation. The PLA therefore consists of a 3-level logic with an AND-OR-OR plane structure. The addition of an extra level of logic obviates the necessity for a quadrupling in the number of devices in the AND plane. Fault tolerance for variable dropping can be achieved with double the number of devices in the AND plane.

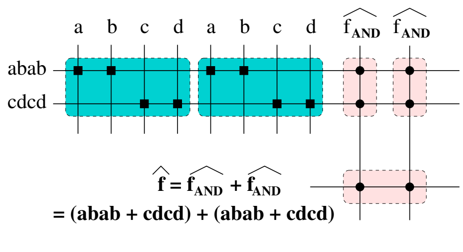

a
b
c
d
a
b
c
d
$\widehat{f_{AND}}$
$\widehat{f_{AND}}$
abab
cdcd
$\hat{f} = \widehat{f_{AND}} + \widehat{f_{AND}}$
= (abab + cdcd) + (abab + cdcd)

Figure 2: Fault masking in a 3-level PLA with A-O-O planes

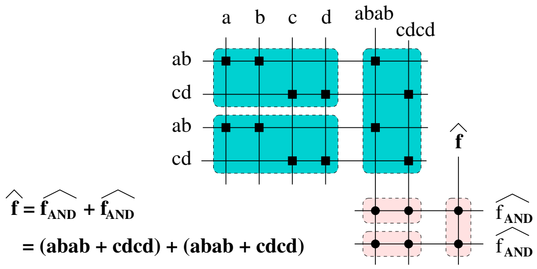

a
b
c
d
abab
cdcd
ab
cd
ab
cd
$\hat{f}$
$\hat{f} = \widehat{f_{AND}} + \widehat{f_{AND}}$
= (abab + cdcd) + (abab + cdcd)
$\widehat{f_{AND}}$
$\widehat{f_{AND}}$

Figure 3: Fault masking PLA with A-A-O-O planes

Any device dropping in the AND plane is propagated to all the product terms in this approach. For example, if the device connecting variable wire a and product term wire $abab$ is dropped, then bab, instead of $abab$, is duplicated in the OR plane and the logic output essentially becomes $(bab + cdcd) + (bab + cdcd)$. However, this fault is masked since the variable dropping fault is locally masked in $\widehat{f_{AND}}$ already.

4.2 A-A-O-O: the 4-level logic approach

In the previous 3-level A-O-O approach, the AND plane is duplicated to make the product term wires robust against the missing variable faults. The two copies of the AND plane are placed horizontally in a row, so the number of input wires is doubled while the number of product term wires remains the same. For a PLA with a large number of input variables but with a relatively small number of product terms, such an approach proves costly in terms of wire number. To avoid the high cost in terms of input wires, the AND plane can be duplicated vertically, thus doubling the number of product term wires and keeping the input number unchanged. This approach requires an additional level of AND logic to implement the $\widehat{f_{AND}}$, as is shown in figure 3. Two additional levels of logic are necessitated in this approach, making it an AND-AND-OR-OR plane PLA structure.

4.3 A-O-O-A: an alternative 4-level logic approach

Since a PLA can easily implement both $\widehat{f_{AND}} = aabb + ccdd$ and $\widehat{f_{OR}} = ab + ab + cd + cd$, an alternative tautology representation of logic function f is in the form of

$$\widehat{f} = \widehat{f_{OR}} \cdot \widehat{f_{OR}} = (ab + ab + cd + cd)(ab + ab + cd + cd).$$

Based on this representation, an approach shown in figure 4(a) can be easily constructed. However, a closer analysis reveals that, in contrast to the approaches described in the previous subsections, this approach actually does not cover the device dropping faults in the AND plane. For example, if the device at the crosspoint of wire a and the first ab wire is missing, then the logic function essentially becomes $\widehat{f} = \widehat{f_{OR}} \cdot \widehat{f_{OR}} = (b + ab + cd + cd)(b + ab + cd + cd) = b + cd$, which is incorrect.

This incorrect result can be attributed to the fact that the two $\widehat{f_{OR}}$ functions in figure 4(a) consist of exactly the same set of product terms. Therefore, a single device dropping fault in any of the two AND planes is propagated simultaneously to both of the two $\widehat{f_{OR}}$ functions. In this case, the masking of a single device dropping fault in the AND plane is to be performed at the end stage in the third AND plane. Since both $\widehat{f_{OR}}$ functions receive the same faulty effect, the fault masking stage at the third AND stage fails.

It is therefore crucial to separate the propagation of variable dropping faults in the two $\widehat{f_{OR}}$ functions. A fourth level of logic is therefore necessitated in this approach, as is shown in figure 4(b). Consider the same example, where the device connecting the left-

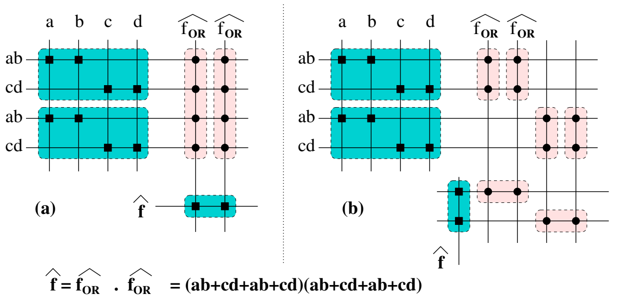

a b c d
$\widehat{f_{OR}}$ $\widehat{f_{OR}}$
ab
cd
ab
cd
(a)
$\widehat{f}$
a b c d
$\widehat{f_{OR}}$ $\widehat{f_{OR}}$
ab
cd
ab
cd
(b)
$\widehat{f}$
$\widehat{f} = \widehat{f_{OR}} \cdot \widehat{f_{OR}}$ = (ab+cd+ab+cd)(ab+cd+ab+cd)

Figure 4: (a): A scheme in a 3-level A-O-A PLA that fails to mask the device missing faults and (b): a 4-level A-O-O-A PLA with fault masking capability

most variable wire a and the uppermost product term wire ab is missing; in the case of figure 4(a) the end result becomes $(b+cd+ab+cd)(b+cd+ab+cd) = (b+cd)$, which would be incorrect. However, such a fault is successfully masked in the case of figure 4(b), where the end result is $((b+cd)+(b+cd))((ab+cd)+(ab+cd)) = (ab+cd)$. It can be similarly observed that any device dropping fault in all the OR planes can be masked in the approach of figure 4(b) as well. Two additional levels of logic are necessitated in this approach, making it an AND-OR-OR-AND plane 4-level PLA structure.

5. TRADEOFF ANALYSIS

Assume that the nanoelectronic PLA implements a logic function with I input wires, O function outputs and P product terms. Figure 5 illustrates the overall schematics for the four different fault tolerance schemes discussed.

Figure 5(a) shows the original implementation of a 2-level PLA without any fault tolerance capability, where I input wires are crossed by P product term wires in the AND plane, and the P product term wires are crossed by O output wires in the OR plane.

Figure 5(b) shows the 2-level “A-O” fault masking PLA approach. The AND plane is duplicated 4 times with double the number of input and product term wires. The original OR plane is duplicated and the O fault tolerant functional outputs cross the $2P$ product term wires in the two identical OR planes.

In figure 5(c), the 3-level “A-O-O” approach is shown where both the AND plane and the OR plane are duplicated. An extra logic level of OR planes is added with every logic output wire ORed with its duplication, using two devices for each logic output wire. Therefore, the extra level of OR logic uses an additional $2O$ number of devices and O wires.

In figures 5(d) and 5(e), two extra logic levels are added. Figure 5(d) shows the 4-level “A-A-O-O” approach with the original AND and OR planes duplicated and two extra planes added: an AND plane with $2P$ devices and an OR plane with $2O$ devices. The “A-O-O-A” approach shown in figure 5(e) adds two extra OR planes with $4O$ devices and a final AND plane with $2O$ devices. The device overhead for this approach is dominated by the quadrupling of the original OR plane and the doubling of the original AND plane.

The hardware overhead for a fault masking PLA scheme needs to be analysed from the vantage point of both the device and the wiring aspects. Furthermore, depending on the logic function parameters of the original PLA, basically the input variable number, product term number and output number, the various fault masking PLA schemes exhibit distinct hardware overhead. Assuming that the number of devices utilized in the original PLA to be D_A for the AND plane and D_O for the OR plane, table 2 summarizes the hardware overhead of the discussed fault masking schemes.

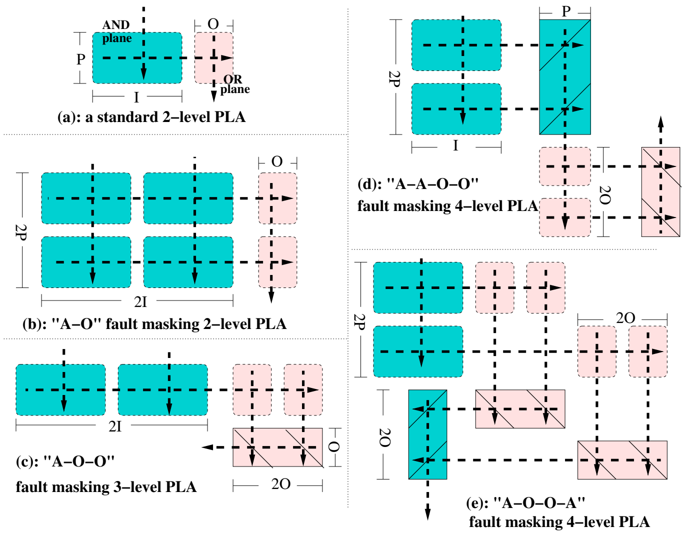

AND plane
O
P
I
OR plane
(a): a standard 2-level PLA
O
2P
2I
(b): "A-O" fault masking 2-level PLA
2I
O
2O
(c): "A-O-O"
fault masking 3-level PLA
P
2P
I
(d): "A-A-O-O"
fault masking 4-level PLA
2O
2P
2O
2O
(e): "A-O-O-A"
fault masking 4-level PLA

Figure 5: Fault masking schemes summary

As a reference point, we consider first the hardware overhead of a TMR fault masking approach for a PLA structure. In a TMR approach, both the AND plane and the OR plane need three identical copies, thus requiring $3D_A + 3D_O$ devices. In terms of wires, a tripling in the number of product term wires and output wires is necessary; however, one copy of the input wires can be extended to cross the three AND planes placed in a column. For a TMR approach, a majority vote process is required for every final output wire. This in turn imposes two extra logic levels in a PLA structure, since a majority vote logic of three output bits o_1, o_2, o_3 is represented in the AND-OR form as $o = o_1o_2 + o_2o_3 + o_1o_3$. This extra voting stage adds an additional $9O$ devices and $7O$ wires to the original PLA structure.

As can be seen in table 2, the proposed four fault masking schemes have distinct hardware overhead in terms of devices and wires, depending on the multiple parameters of the PLA structure. In terms of the number of devices, the 3-level scheme of A-O-O and the 4-level scheme of A-A-O-O introduce in general the least hardware overhead. The dominating hardware in these cases consists of doubling the number of both the AND and OR plane devices, i.e., $(2D_A + 2D_O)$.

From the wire aspect vantage point, the hardware overhead of the fault masking schemes depends heavily on the PLA parameters. According to table 2, it can be observed that for a PLA with a large number of output wires (O), the 2-level fault masking approach is most promising with the least amount of hardware overhead. For a PLA with a large number of input wires (I), the 4-level fault masking approaches can be considered. For a PLA where the wires are

fault masking scheme	hardware		logic level
	device	wire	
original	$D_A + D_O$	$I + P + O$	2
TMR	$3D_A + 3D_O + 9O$	$I + 3P + 7O$	4
A-O	$4D_A + 2D_O$	$2I + 2P + O$	2
A-O-O	$2D_A + 2D_O + 2O$	$2I + P + 3O$	3
A-A-O-O	$2D_A + 2D_O + 2P + 2O$	$I + 3P + 3O$	4
A-O-O-A	$2D_A + 4D_O + 6O$	$I + 2P + 7O$	4

Table 2: HW overhead for the fault masking schemes

dominated by the product term lines, the 3-level A-O-O approach requires the least hardware overhead. Overall, the four proposed fault masking schemes provide a variety of choices to be exploited according to the particular specification of any PLA logic so as to achieve fault masking capability with low hardware overhead, both in terms of nano devices and nano wires.

6. FUTURE WORK

In this paper we have presented an initial effort at a fault tolerance framework for nanoelectronics based PLA logics. However, the severe reliability challenge in the nanoelectronic environment demands further investigation, in order to construct workable systems. Specifically, the significantly increased fault rates, possible clustering fault behavior as well as the other topological constraints imposed by the underlying nanoelectronic devices, all require further research on aggressive fault tolerance schemes that are particularly designed for such nanoelectronic systems.

We envision our future work on the fault tolerance of nanoelectronic PLA logics to consist of two main directions. First, we plan to research the enhancement of the fault tolerance capability based on the proposed fault masking schemes. Particularly, the challenges of fault tolerance on defective PLA structures with irregularity, multiple fault occurrences in the nanoelectronic PLAs, the clustering behavior of fault occurrences, and the assignment of redundant input variable / product term wires under the severe interconnect constraint need to be addressed. Second, we plan to investigate reconfiguration based online-repair schemes for nanoelectronic PLAs. Such online reconfiguration based fault tolerance is promising in providing high flexibility in dealing with the unreliability problem while necessitating comparatively low hardware overhead.

7. CONCLUSIONS

When addressing the reliability challenge in the nanoelectronic PLA system, hardware redundancy based fault tolerance schemes are especially promising. In this paper, we propose a class of fault tolerance approaches in a nanoelectronic PLA structure, concentrating on the dominant device missing fault occurring online.

We have developed in this paper based on Boolean logic tautology a class of fault masking approaches with no requirement for majority voting. These fault masking approaches achieve fault tolerance with significantly reduced hardware overhead by targeting the dominant missing device type of faults in nanoelectronic PLAs and by exploiting the particularity in the PLA logic structure. The proposed genre of fault tolerance schemes can generate correct results without performance degradation with efficient hardware, thus setting up a framework for the online fault tolerance approach in the nanoelectronic PLA logic by investigating multiple hardware redundancy based fault tolerance possibilities.

8. REFERENCES

[1] M. A. Kastner, "The Single-Electron Transistor", *Review of Modern Physics*, vol. 64, pp. 849–858, 1992.

[2] P. Mazumder, S. Kulkarni, M. Bhattacharya, J. P. Sun and G. I. Haddad, "Digital Circuit Applications of Resonant Tunneling Devices", *Proceedings of the IEEE*, vol. 86, n. 4, pp. 664–686, April 1998.

[3] P. Avouris, J. Appenzeller, R. Martel and S. Wind, "Carbon Nanotube Electronics", *Proceedings of the IEEE*, vol. 91, n. 11, pp. 1772–1784, 2003.

[4] C. S. Lent, P. D. Tougaw, W. Porod and G. H. Bernstein, "Quantum Cellular Automata", *Nanotechnology*, vol. 4, pp. 49–57, 1993.

[5] Y. G. Krieger, "Molecular Electronics: Current State and Future Trends", *J. Structural Chem*, vol. 34, pp. 896–904, 1993.

[6] ITRS, *International Technology Roadmap for Semiconductors Emerging Research Devices*, 2005.

[7] J. R. Heath, P. J. Kuekes, G. S. Snider and S. Williams, "A Defect-Tolerant Computer Architecture: Opportunities for Nanotechnology", *Science*, vol. 280, pp. 1716–1721, June 1998.

[8] Y. Huang, X. Duan, Y. Cui, L. J. Jauhon, K. Kim and C. M. Lieber, "Logic Gates and Computation from Assembled Nanowire Building Blocks", *Science*, vol. 294, pp. 1313–1317, November 2001.

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

[10] H. Naeimi and A. DeHon, "A Greedy Algorithm for Tolerating Defective Crosspoints in NanoPLA Design", in *ICFPT*, pp. 49–56, 2004.

[11] P. J. Kuekes, D. R. Stewart and R. S. Williams, "The Crossbar Latch: Logic Value Storage, Restoration, and Inversion in Crossbar Circuits", *Journal of Applied Physics*, vol. 97, n. 3, pp. 034301, July 2005.

[12] G. Snider, P. J. Kuekes and R. S. Williams, "CMOS-like Logic in Defective, Nanoscale Crossbars", *Nanotechnology*, vol. 15, pp. 881–891, Aug 2004.

[13] R. Rubin and A. DeHon, "Nanowire-based Sublithographic Programmable Logic Arrays", in *FPGA*, pp. 123–132, 2004.

[14] A. DeHon, "Array-Based Architecture for FET-Based, Nanoscale Electronics", *IEEE Transactions on Nanotechnology*, vol. 2, n. 1, pp. 23–32, 2003.

[15] D. B. Strukov and K. K. Likharev, "CMOL FPGA: A Reconfigurable Architecture for Hybrid Digital Circuits with Two-terminal Nanodevices", *Nanotechnology*, vol. 16, pp. 888–900, Apr 2005.

[16] K. Nikolic, A. Sadek and M. Forshaw, "Architectures for Reliable Computing with Unreliable Nanodevices", in *IEEE-NANO*, pp. 254–259, 2001.

[17] P. Beckett and A. Jennings, "Towards Nanocomputer Architecture", in *Asia-Pacific Computer System Architecture Conference*, pp. 141–150, 2002.

[18] M. Forshaw, R. Stadler, D. Crawley and K. Nikolic, "A Short Review of Nanoelectronic Architectures", in *Nanotechnology*, volume 15, pp. 220–223, 2004.

[19] M. Abramovici, M. A. Breuer and A. D. Friedman, *Digital Systems Testing and Testable Design*, IEEE Press, 1990.

[20] G. Snider and W. Robinett, "Crossbar Demultiplexers for Nanoelectronics Based on n-Hot Codes", *IEEE Transactions on Nanotechnology*, vol. 4, pp. 249–254, 2005.

[21] J. Emmert, C. Strout, B. Skaggs and M. Abramovici, "Dynamic Fault Tolerance in FPGAs via Partial Reconfiguration", in *FCCM*, pp. 165–174, 2000.

[22] P. K. Lala and D. L. Tao, "On Fault-tolerant PLA Design", in *IEEE Southeastcon*, pp. 945–947, 1990.

[23] J. Khakbaz and E. J. McCluskey, "Concurrent Error Detection and Testing for Large PLA's", *IEEE Journal of Solid-State Circuits*, vol. 17, n. 2, pp. 386–394, April 1982.

[24] W. K. Fuchs, C. R. Chen and J. A. Abraham, "Concurrent Error Detection in Highly Structured Logic Arrays", *IEEE J. Solid-State Circuits*, vol. 22, pp. 583–594, August 1987.

[25] P. Bose and J. A. Abraham, "Test Generation for Programmable Logic Arrays", in *DAC*, pp. 572–580, 1982.