01 Apr 2007

Nanofabric PLA Architecture with Double Variable Redundancy

Mandar V. Joshi

Waleed K. Al-Assadi
Missouri University of Science and Technology, waleed@mst.edu

Recommended Citation

M. V. Joshi and W. K. Al-Assadi, "Nanofabric PLA Architecture with Double Variable Redundancy," *Proceedings of the IEEE Region 5 Technical Conference, 2007*, Institute of Electrical and Electronics Engineers (IEEE), Apr 2007.
The definitive version is available at https://doi.org/10.1109/TPSD.2007.4380347

Nanofabric PLA Architecture with Double Variable Redundancy

M.V. Joshi and W. K. Al-Assadi
Department of Electrical and Computer Engineering, University of Missouri-Rolla, 1870 Miner Circle, Rolla, MO 65401
Email :{ mvjvx8, waleed} @umr.edu

***Abstract*- It has been shown that fundamental electronic structures such as Diodes, and FET's can be constructed using selectively doped semiconducting Carbon Nanotubes or Silicon Nanowires (CNT's, SiNW's) at nanometer scale. Memory and Logic cores using these technologies have been proposed, that use the configurable junctions in two-dimensional crossbars of CNT's. These Memories and Logic Arrays at this scale exhibit significant amount of defects that account for poor yield. Configuration of these devices in presence of defects demands for an overhead in terms of area and programming time. In this work, we introduce a PLA (Programmable Logic Array) configuration that makes use of design-specific redundancy in terms of number of nanowires, in order to simplify the process of programming the PLA, increase the yield and reduce the time complexity and in turn, the cost of the system.**

I. INTRODUCTION

The advances in Photolithographic techniques of today have made the miniaturization of electronic circuits possible. According to Moore's Law, the number of transistors per unit area would continue to double, approximately every two years. However, the applicability of Moore's law will cease to continue as the pitch sizes approach molecular dimensions. It therefore becomes necessary to explore devices and technologies that can match these trends of increase in transistors per area. We propose a technique to tolerate defects at nanometer scale using Carbon Nanotubes and Si Nanowires.

Semiconducting Carbon-Nanotubes and SiNW's exhibit electronic properties similar to those of conventional lithographic-scale CMOS devices, in terms of electron and hole mobilities. It has been shown that chemical passivation of SiO_x shell surrounding single crystal SiNW cores can significantly enhance conductance-gate voltage behavior making these wires highly suitable to be used as Field Effect Transistors [1], and in turn building blocks for digital circuits. The electronic applications of NWs are based on diode and FET-like properties of NW junctions or *"crosspoints"* in two-dimensional arrays, called as *Nanofabrics* or *Crossbars.* Crosspoints can be grouped together to form a memory or logic device. Cha et al [2] have shown electro-mechanical switching devices using suspended nanotubes. The crosspoints at the junctions are programmed using this "Bistable" property that they exhibit. Their ON-state behavior is similar to that of a diode. When the two wires forming a junction are in close contact, the junction resistance is very small, and when the wires are far away, their resistance increases by a great extent (~33MΩ in closed state and ~10kΩ in open state)[2]. A crosspoint can be programmed ON or OFF by applying a voltage Differential of ~3.6V.

Synthesis of Boolean expressions can be made possible on PLA's based on Crossbars. A row in a Crossbar can be made to act as a Boolean product/sum term by programming ON only the junctions or crosspoints that correspond to the variables that take part in the term, as shown in Fig. 1. Inputs A, C and D are called the ON inputs, as they take part in the evaluation of the product term. Rests of the inputs are called the OFF inputs. The programmability of a crosspoint is statistical in nature, and therefore such a configuration of PLAs gives a poor number of successfully configured crosspoints even for a small number of junctions to be programmed on a NW. "Yield" of configuration can thus be defined as the ratio of number of NWs successfully configured and the total number of NWs available for configuration. We propose to develop scheme to tolerate this statistical behavior of crosspoints to obtain an acceptable yield for PLA configuration.

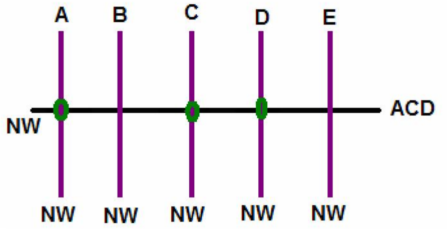


Fig.1: Generation of a product term on a NW grid

II. CROSSPOINT DEFECT MODEL

The bistable property of crosspoints can be used to implement *logic blocks* in a PLA [3], or *memory cells* [4]. This is useful for implementing *NAND and NOR functions* in NanoPLA. The crosspoints can lose their programmability because of the mechanisms discussed below. For these simulations we assume a random distribution of such defects throughout the crossbar.

Breaks in Nanowires: It has been observed that the probability of having breaks in a Nanowire increases with increase in its length. Breaks can be introduced during the fabrication of nanowires, due to limitation of the fabrication techniques, and axial stress. As more length adds to axial stress and increases occurrence of breaks, their lengths should, nominally, not exceed a few tens of microns (typically 20 to 25 microns). It is reasonable to assume that as high as 5% of the Nanowires exhibit breaks, and therefore are unusable [5].

1-4244-1280-3/07/$25.00 ©2007 IEEE

Non-Programmable crosspoints: These defects are characterized by the inability of a crosspoint to be programmed "closed" or "open". The latter is observed to be rare [7], and therefore not considered in the present discussion. The occurrence of defective crosspoints is a function of fabrication technique, size of the array, and the random distribution of molecules at the junction area. With reasonable assumptions of operating conditions, it can be proved that the occurrence of a "closed non-programmable" defect is largely due to absence of sufficient electrons at the junction area.

III. RELATED WORK

A defect-tolerant methodology that is proposed in [7] uses a Greedy Heuristic Algorithm to find a solution to the mapping problem discussed above. The algorithm sorts the function rows in matrix F in descending order of number of ON inputs contained in them, which enhances the probability of a successful match. This algorithm is intended to be used with NanoPLA architecture proposed by DeHon et. al. in [3].

The limitation of this algorithm is its very high time complexity in sorting elements of F. It is seen that the Time Complexity for sorting is an exponential function of defect rate and number of crosspoints to be programmed. It therefore becomes infeasible to use this algorithm for NanoPLA with high defect rates and number of ON inputs more than 10%.

Our redundancy technique can be used in conjunction with NanoPLA in [3] or *Nanofabric Molecular Logic Array* (MLA) Proposed by Goldstein et. al. in [6].

IV. OUR APPROACH: DOUBLE VARIABLE REDUNDANCY

We propose to obtain greater yield rates for the configuration of a NanoPLA without having to compromise for Time Complexity, seen in [7]. It can be noted that the Greedy Algorithm discussed above has a very high time complexity for higher defect rates. To minimize this time complexity, we introduce redundancy in terms of number of nanowires and observe the yield rates and area overhead. We allocate two vertical Nanowires per Product (or Sum) term in order to achieve a better coverage in the presence of crosspoint defects. This is illustrated in Fig. 2. If any of the two Vertical Nanowires have a programmable junction with any of the two Horizontal Nanowires in consideration, the condition is equivalent to having a programmable resource. We term this method as Double Variable Redundancy (DVR), and examine the yield obtained by DVR in comparison with the Greedy Heuristic Algorithm. Since DVR uses sequential PLA configuration, it eliminates the sorting used in [7].

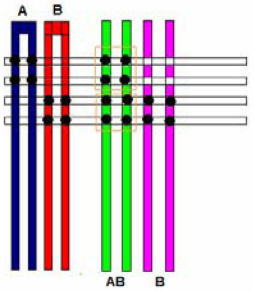


Fig.2: Generation of a product term on a DVR based PLA

It can be mathematically proved that the probability of getting a pair of NWs unsuitable for a given minterm is significantly low as compared to the defect rate in the crossbar. This fact is the result of the three redundant crosspoints present. On account of this, the need of sorting the rows/columns to configure the PLA for given logical functions can be eliminated. The address decoding scheme can be same as that in [2]. The location of Pull-up and Pull-down networks determines the working of the array as AND array or OR array, as seen in Fig. 3. In this work, we only make use of the "Diode-like" working of a crosspoint during the evaluation of Boolean functions. Therefore the NanoBlock *does not have* an ability to invert the inputs, and hence we introduce both, the real and inverted inputs externally. This also eliminates the need of "inverting block" in [2]. The DVR based PLA architecture is illustrated in Fig. 3. It shows implementation of product-sum terms using DVR-based PLA.

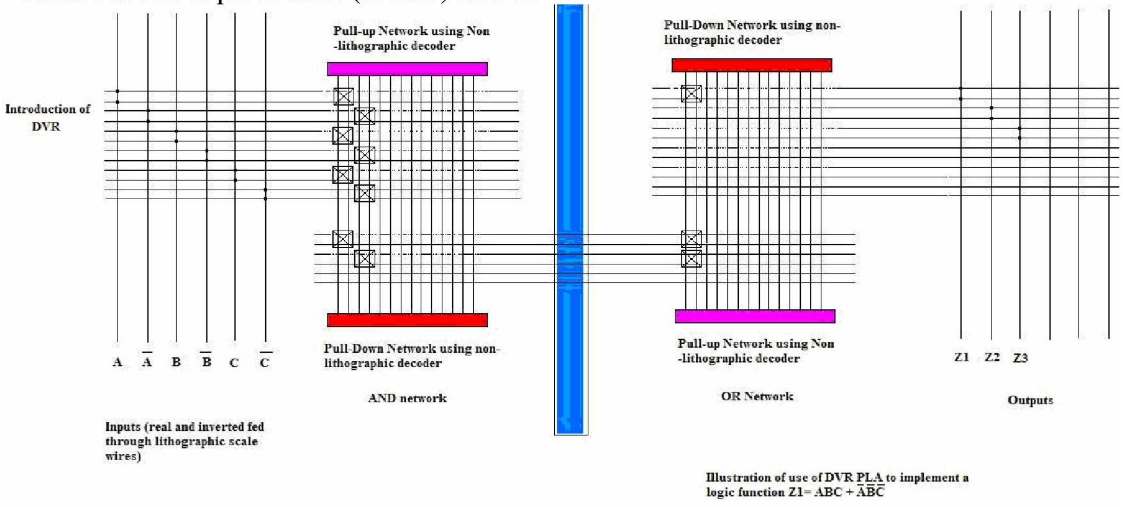

Introduction of DVR
Pull-up Network using Non -lithographic decoder
Pull-Down Network using non-lithographic decoder
Pull-Down Network using non-lithographic decoder
Pull-up Network using Non -lithographic decoder
A
$\bar{A}$
B
$\bar{B}$
C
$\bar{C}$
AND network
OR Network
Z1
Z2
Z3
Outputs
Inputs (real and inverted fed through lithographic scale wires)
Illustration of use of DVR PLA to implement a logic function Z1= ABC + $\bar{A}\bar{B}\bar{C}$

Fig. 3: The DVR based proposed PLA Architecture with an illustration of pre-configured Boolean function

1-4244-1280-3/07/$25.00 ©2007 IEEE

V. Defect Rate Calculation In Dvr Based PLA

Assumed Parameters:

P_{cp}= 0.05 to 0.2 (Probability that a single crosspoint is non programmable)
c= number of columns in PLA
r= number of rows in PLA
P_{on}= Probability that a given crosspoint-quad is to be programmed

Each product/sum term to be programmed has to follow a certain "path" through each section of the PLA. The failure of a product/sum term would mean that there is *at least one* quad that did not get programmed.

A quad being non-programmable has a probability equal to fourth power of the probability, that two consecutive crosspoints are defective. If we have "n" crosspoints, and the probability of occurrence of defective crosspoint is "P_{cp}", the expected number of defective crosspoints is given by (1),

$$d = \text{integer}\,(n * P_{cp}) \qquad (1)$$

The probability of a given Quad of crosspoints being defective is:

$$P\,(\text{defective quad}) = {}^{(n-4)}C_{(d-4)} / {}^{n}C_{d} \qquad (1.1)$$

Now, the probability that a given programmable quad is to be programmed and it is defective is given by,

$$P_{on} * {}^{(n-4)}C_{(d-4)} / {}^{n}C_{d} \qquad (1.2)$$

This applies to all the quads in the same column. We know that the total number of quads in a column is same as the number of rows in the device.

Therefore, the probability of finding one defect on a given NW column is given by combining equations 1.1 and 1.2

$$P\,(1) = r * P_{on} * {}^{(n-4)}C_{(d-4)} / {}^{n}C_{d} \qquad (2)$$

The NW column is unusable if there is *at least one* defective quad at a location that needs to be programmed. Therefore, the probability of getting a non-programmable NW column is given by,

$$P_{dcrosspoint} = \sum_{i=1}^{r} P\,(i)$$

$$= \sum_{i=1}^{r} \left\{ r * (P_{on})^{i} * {}^{(n-4i)}C_{(d-4i)} / {}^{n}C_{d} \right\} \qquad (3)$$

Expression (3) gives us the defect probability exclusively due to crosspoint-defects. For a 50x50 NanoPLA having a crosspoint defect rate of 15%, this probability is of the order of 2%.

We also consider NW defects to find the yield expression.

Let the probability that a single nanowire is faulty be given by,

P_{wd} = 0.05

H= total number of horizontal wires available for programming (implies that total number of inputs = H/2)
V= total number of vertical wires available for programming (We assume V=H for symmetry)
Therefore, the probability of a fault due to NW defect can be given as,

$$P_{dNW} = \sum_{i=1}^{r} {}^{(H-2i)}C_{(H*Pwd-2i)} / {}^{H}C_{(H*Pwd)} \qquad (4)$$

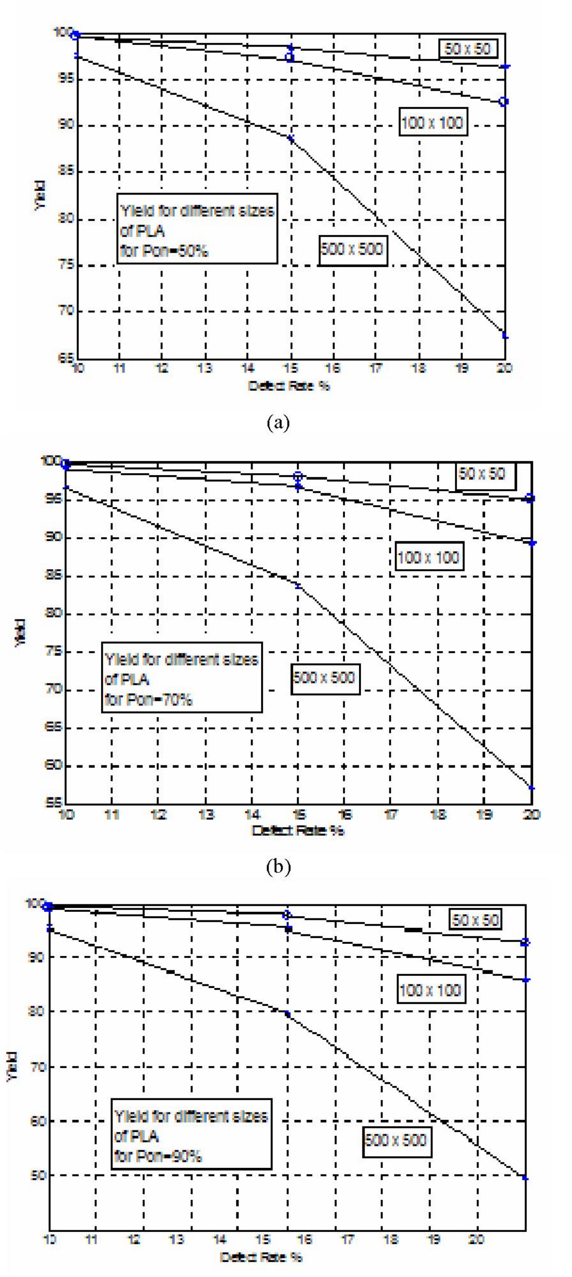
The value of Pw_{d} under consideration is 5%. Substituting this in expression (4), taking H=100 (Assuming the size of the PLA to be 50x50) gives us a defect probability of the order of 10^{-4}, indicating that NW defects have virtually insignificant effect on the overall yield of the PLA in consideration. It is quite intuitive, as a defective NW introduces two defective crosspoints in every quad it is contained in. For the quad to be defective, both the other crosspoints also need to be defective. Therefore, we *neglect the effect of NW breaks in the further discussion*

VI. Simulation Results

A. Yield and Area Overhead

The MATLAB simulations based on the configuration sequence gives us the following results for different array sizes and defect rates. The simulations are carried out by varying the following parameters:

Defect rate: 10% to 20%
PLA size: 50x50 to 500x500
P (on) =50% to 90%



(c)

Fig. 4: Yield Vs Defect Rates for different PLA sizes. It also establishes the relation between Yield and the Probability Pon. For (a) Pon= 50%, (b) Pon=70% and (c) Pon=90%

The results illustrated in Fig. 4 above indicate that the yield is significantly high for PLA sizes up to 100 X 100, or defect rates up to 15%. It can be seen that the yield is a weak function of Pon, as the curves retain the pattern for all the three cases where the values of P_{on} are significantly different. For array sizes as high as 500 X 500, however, the yield becomes smaller for defect rates greater than 10%. It therefore follows that the logic functions having the number of variables considerably greater than 100 result in very low yield. Such functions require more than one PLA for realization. This requirement is different from "Fan-in Bounding" discussed in [7], where the Number of crosspoints to be programmed in a row/column is restricted to a certain number. DVR based PLA does not need fan-in bounding on account of added redundancy.

B. Time Complexity

In this section we illustrate and compare the time complexity involved in the programming of PLA's based on DVR, and the time complexity that exists in the programming technique using Greedy Heuristic Algorithm. Summarizing the results in [7], we have the time complexity for Greedy Heuristic Algorithm as:

$$T_{c(GA)} = O\,(|F| \log (|F|)) \; + O\,(|F| \cdot P_J^{-cm} \cdot c_m) \qquad (5)$$

Where,
$T_{c(GA)}$= Time complexity for Greedy Heuristic Algorithm
|F| = size of the array of Boolean functions
cm= maximum number of crosspoints to be
Programmed in a minterm
P_J= Probability that the given junction is Programmable.

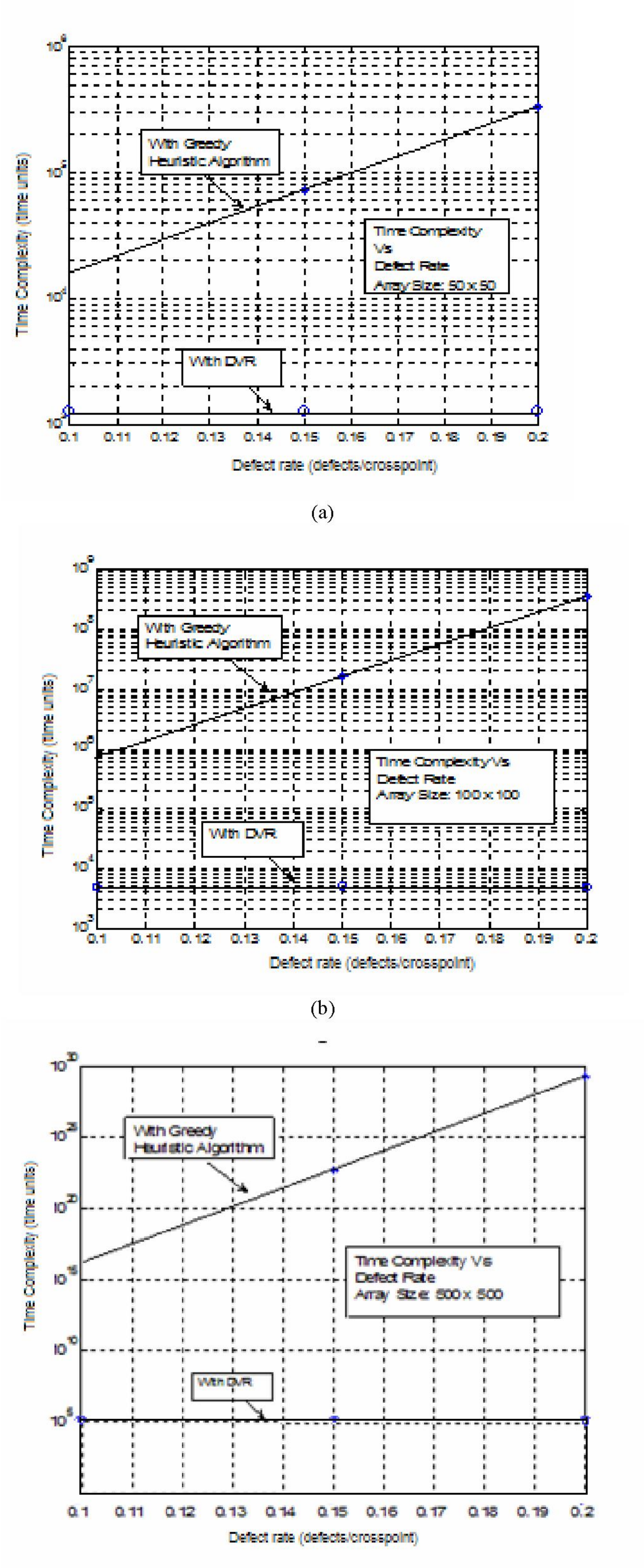
Now,
The time complexity involved in the configuration of a DVR based PLA is given as:

T_c= Total number of crosspoints to be Programmed
Therefore the Time complexity in DVR is given by the expression (6).

$$T_{c(DVR)} = O(r*c*P_{on}) \qquad (6)$$

Double Variable Redundancy is therefore observed to be distinctly advantageous in terms of yield and time complexity over Greedy Heuristic Algorithm, as seen in Fig. 5. It clearly shows that the time complexity is independent of defect rate.

1-4244-1280-3/07/$25.00 ©2007 IEEE



(a)

(b)

(c)

Fig. 5: Comparison between the time complexities in the PLA configuring mechanisms: Greedy Heuristic Algorithm and DVR for different Array sizes. (a) Array size 50 x 50 (b) Array size 100 x 100 (c) Array Size 500 x 500

Conclusions

We have successfully simulated the configuration of this DVR in MATLAB to verify the yield obtained. It is seen that it matches with the theoretical calculations. The DVR approach gives excellent improvement with respect to yield and time complexity. It is less area efficient, as the Redundancy introduced demands for double the number of horizontal and vertical Nanowires. It reduces the time complexity of configuration by a great extent without compromising the yield. The development of the DVR approach demands for change in the NanoPLA architecture in order to enable the introduction of an input variable via different, but predefined, nanowires.

References

[1] Y Cui, Z Zhong, D Wang, W Wang, C Lieber, *"High Performance Silicon Nanowire Field Effect Transistors"*, Nano Letters Vol. 3, No. 2, 149-15, 2003

[2] S Cha, J Jang, Y Choi, G Amaratunga, D Kang, D Hasko, J Jung, N Kim, *"Fabrication of a nanoelectromechanical switch using a suspended carbon nanotube,* Applied Physics Letters, 083105, 2005

[3] A. DeHon and M. J. Wilson, "*Nanowire-Based Sublithographic Programmable Logic Arrays*", in FPGA, pp. 123–132, 2004.

[4] A. DeHon, S. Goldstein, P. Kuekes, "*Nonphotolithographic NanoscaleMemory Density Prospects",* IEEE TRANSACTIONS ON NANOTECHNOLOGY, VOL. 4, NO. 2, MARCH 2005

[5] T. Hogg, G. Snider, *"Defect-tolerant logic with Nanoscale crossbar circuits"*, HP Labs, May 2004. URL:http://www.hpl.hp.com/research/idl/papers/molecularAdder/circuits.pdf

[6] S. C. Goldstein and M. Budiu, "NanoFabrics: Spatial Computing Using Molecular Electronics," in ISCA, June 2001, pp. 178–189

[7] H. Naeimi,, *"A Greedy Algorithm for Tolerating Defective Cross points in Nano PLA Design",* MS Dissertation, California Inst. Of Technology, 2004