Designing Fault Tolerant FSM by Nano-PLA

S. Baranov*, I. Levin**, O. Keren*, and M. Karpovsky***

* Bar Ilan University/School of Engineering, Ramat Gan, Israel

** Tel Aviv University/School of Education, Tel Aviv, Israel

***Boston University/Department of Electrical, Computer and System Engineering, Boston, USA

samary@012.net.il, ilia1@post.tau.ac.il, kereno@eng.biu.ac.il, markkar@bu.edu

Abstract— **The paper deals with designing fault tolerant finite state machines (FSMs) by nanoelectronic programmable logic arrays (PLAs). Two main critical parameters of the fault tolerant nano-PLAs, the area and the number of crosspoint devices, are considered as optimization criteria for the synthesis. The paper introduces a method for synthesizing fault tolerant nano-PLA based FSMs. The method is based on decomposing an initial PLA description of the FSM into a three interacting portions. The proposed solution provides significant reduction of the area without meaningful increasing of a number of crosspoint devices in comparison with known solutions and provides a trade-off between the area and the number of devices in designing FSMs by PLAs.**

I. Introduction

Programmable logic arrays (PLAs) have a bright history in logic design. Being introduced in the early 1970-s, the PLA became a very popular and successful base for the logic design. Extensive research has been done in the fields of area minimization and testability. Usually, successful computation structures return back at the next round of the technological spiral. The PLA technology can certainly be considered as such a promising basis owing to its regularity, manufacturing suitability, efficiency of the synthesis methods, etc. Recently, the phenomenon of returning to the PLA basis can be observed in the modern nanoelectronics. The nano crossbar PLA structures can easily be fabricated using the bottom-up self-assembly process. PLA implementations can be built using various nanolectronic devices.

The nanoelectronic systems' reliability becomes a critical bottleneck when they are utilized for the practical design. Both the manufacturing defects and a lot of possible run-time faults may disturb proper functioning of the nano-PLA circuits. The high probability of both the presence and the appearance of faults in the circuits define specific challenges of the research, to achieve effective use of the nano-PLA technology in the modern logic design.

In the case of nano-PLA, the number of faults is expected to become so large that the conventional ideology of fault detection (to detect a fault as soon as possible) becomes doubtable. Moreover, satisfying such a requirement may even disturb the normal functioning of a nano circuit. To allow the circuit to function even in the presence of a fault, a fault-tolerance technique has to be applied. One well-known way to increase fault-tolerance of a circuit is increasing its redundancy to guarantee functioning of the circuit in presence of faults. Due to the significant increase of fault occurrence in nanoelectronic circuits, and in particular in nano-PLAs, intensive fault tolerant techniques are required. Two main techniques were proposed for providing fault-tolerance to the nano-PLAs: *on-line repair* [12] and *fault masking* [7]. The fault masking-based techniques are preferable, due to their on-line nature and lower hardware penalties.

Fault tolerant techniques for nano-PLAs were studied in [14], [15]. The authors developed a *tautology technique* and a corresponding PLA architecture. Following [14] denote A as the AND plan, O as the OR plan of the initial logic system description. Four different fault tolerant PLA architectures were introduced: A-O, A-A-O-O, A-O-O, A-O-O-A. The area overhead required for each of the tautology architectures may be easily estimated. Generally, these techniques provide better overhead penalty than the well-known triple modular redundancy (TMR) method. However, in many cases the approach [14] is inefficient due to the high area overhead. In the present paper, we propose a method that allows increasing efficiency of the tautology-based techniques for a widely used class of digital circuits – the combinational part of FSMs.

It has been observed ([4], [11], [12]) that the device missing dominants in nano-PLAs. PLA crosspoints may be with or without devices. PLA including a small percent of devices in both of its plans is considered to have low density. Obviously, between two PLAs of identical square, the dense PLA is less fault-tolerant. Both the number of PLA devices and the PLA area has to be considered as optimization criterion in synthesis of logic circuits by nano-PLA.

All the tautology methods are based on doubling rows and/or columns of PLA plans. These methods basically double empty crosspoints of the PLA, which results in unreasonable overhead. Namely, known methods for synthesis of the tautology based PLAs don't use the density parameter to reduce the resulting area.

In this paper, we introduce a new approach for synthesis of fault tolerant circuits, which method is highly suitable in non-dense PLAs. In general, the combinational part of FSM corresponds to a low density PLA. Our method is based on decomposition of the initial PLA into a chain of high-density component PLAs. Such high-density components can be then transformed into a certain fault tolerant form, without excessive doubling of PLA's empty crosspoints. As shown in Section IV, our approach provides the fault tolerant property with about 15% area overhead in respect to the conventional implementation of the non fault tolerant PLA based FSM.

978-1-4244-4595-0/09/$25.00 © 2009 IEEE

The proposed decomposition of the combinational circuit transforms an initial FSM specification into three interacting components in order to reduce the total area. To achieve the fault tolerance, each of the components can be implemented independently by using one of the known tautology-based techniques. It allows to select the best solution for a specific FSM by applying different fault tolerant techniques for each of the components of the proposed PLA structure.

In addition to the area minimization, the proposed SM architecture has a diagnostic potential that may be utilized for implementation of reconfigurable nano-structures. On-line reconfigurability, that is highly desirable in the post-fabrication processes, requires diagnosis of a circuit to bypass possible defects.

The paper is organized as follows. Preliminaries and previous research are discussed in Section II. The proposed six-matrix PLA architecture is described in Section III. Benchmarks results and the estimation of the proposed solution are presented in Section IV. Conclusions are provided in Section V.

II. PRELIMINARIES AND RELATED WORK

The problem of FSM synthesis by PLA was deeply investigated in 70-th when the CMOS PLA structure was introduced. There were two main challenges in the above field: minimization of a number of standard PLA blocks implementing a given FSM, and minimization of the area required for the implementation of the FSM. A number of studies were performed by the authors of the present paper ([2][8][9]). The majority of the works utilizes the following properties of FSMs in order to optimize the resulting solution:

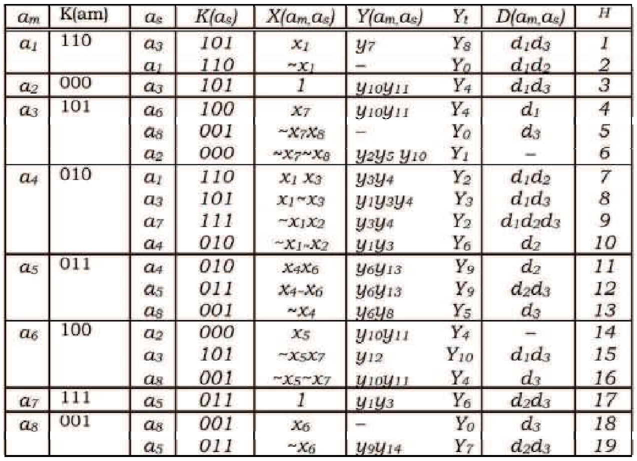
- A system of logic functions, corresponding to a combinational part of FSM is defined by a set of disjoint cubes. This property was used for on-line checking of PLAs ([1], [5], [6], [10]).
- A number of input variables affecting transitions between two certain FSM's states is much smaller than the total number of the FSM input variables. Moreover, a number of input variables affecting all transitions from a certain state is also much smaller than the total number of the FSM input variables.
- A number of possible FSM output vectors is limited and known in advance.
- A number of "ones" in output vectors is much less than a number of "zeros".

The above FSM properties may be interpreted as low density of the corresponding PLA. They were utilized in a number of solutions toward effective implementation of FSMs both as a network of standard PLAs having a minimized number of elements, and as a homogeneous matrix structure with the minimized area.

In [2], one of the authors of the present paper proposed Six Matrix (SM) implementation of CMOS based PLAs. The main goal of the method was to minimize an area required for implementation of the FSM combinational part. In this paper, we develop the SM approach for the case of fault tolerant PLAs.

We assume that the FSM with L input variables, N outputs variables is defined by its structural table as shown in Table I.

TABLE I. STRUCTURAL TABLE OF EXEMPLARY FSM



a_m	K(am)	a_s	$K(a_s)$	$X(a_m, a_s)$	$Y(a_m, a_s)$	Y_t	$D(a_m, a_s)$	H
a_1	110	a_3	101	x_1	y_7	Y_8	d_1d_3	1
		a_1	110	$\sim x_1$	–	Y_0	d_1d_2	2
a_2	000	a_3	101	1	$y_{10}y_{11}$	Y_4	d_1d_3	3
a_3	101	a_6	100	x_7	$y_{10}y_{11}$	Y_4	d_1	4
		a_8	001	$\sim x_7x_8$	–	Y_0	d_3	5
		a_2	000	$\sim x_7\sim x_8$	$y_2y_5\ y_{10}$	Y_1	–	6
a_4	010	a_1	110	$x_1\ x_3$	y_3y_4	Y_2	d_1d_2	7
		a_3	101	$x_1\sim x_3$	$y_1y_3y_4$	Y_3	d_1d_3	8
		a_7	111	$\sim x_1x_2$	y_3y_4	Y_2	$d_1d_2d_3$	9
		a_4	010	$\sim x_1\sim x_2$	y_1y_3	Y_6	d_2	10
a_5	011	a_4	010	x_4x_6	y_6y_{13}	Y_9	d_2	11
		a_5	011	$x_4\sim x_6$	y_6y_{13}	Y_9	d_2d_3	12
		a_8	001	$\sim x_4$	y_6y_8	Y_5	d_3	13
a_6	100	a_2	000	x_5	$y_{10}y_{11}$	Y_4	–	14
		a_3	101	$\sim x_5x_7$	y_{12}	Y_{10}	d_1d_3	15
		a_8	001	$\sim x_5\sim x_7$	$y_{10}y_{11}$	Y_4	d_3	16
a_7	111	a_5	011	1	y_1y_3	Y_6	d_2d_3	17
a_8	001	a_8	001	x_6	–	Y_0	d_3	18
		a_5	011	$\sim x_6$	y_9y_{14}	Y_7	d_2d_3	19

In this table, each row corresponds to a certain transition of the FSM from state a_m to state a_s. Columns of the table are defined as follows:

a_m and $K(a_m)$ - the present state and its code;

a_s and $K(a_s)$ - the next state and its code;

$X(a_m, a_s)$ - the product of input variables that is equal to 1 when FSM transits from a_m to a_s;

$Y(a_m, a_s)$ - a subset of output variables that are equal to 1 on the transition from a_m to a_s. Denote this subset by Y_j - set, $j = 0, \ldots, J$;

$D(a_m, a_s)$- a subset of next state functions equal to 1 on the transition from a_m to a_s;

H - FSM transition number.

The structural table can be directly implemented by the PLA structure. This implementation can be achieved by the direct mapping of each row of the structural table into the PLA plans. The direct matrix implementation of the FSM from Table I is presented in Fig 1.

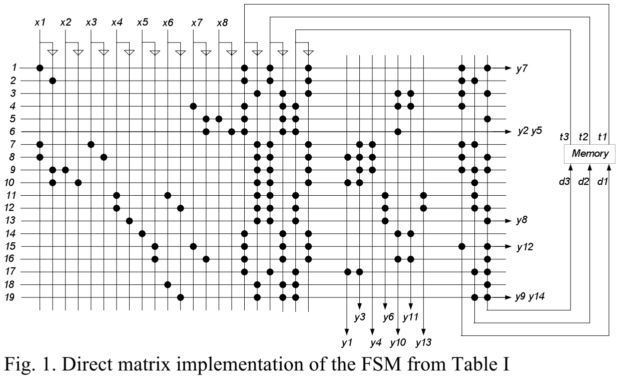


Fig. 1. Direct matrix implementation of the FSM from Table I

It is easy to examine the properties of the obtained FSM. FSM transitions and, consequently, rows of the PLA are defined by products of the small rank. It leads to a low density of the PLA. At the same time, there are some places of high density. Such places correspond to codes of the FSM states.

A number of fault masking approaches (A-O, A-A-O-O, A-O-O, A-O-O-A) with no requirement for majority voting were presented in [14][15]. These approaches achieve fault tolerance with lower overhead by targeting the dominant missing device type of faults in nanoelectronic PLAs. The above approaches are based on the tautology and implement various duplications of AND and OR PLA plans. For these approaches the area of the PLA's structures having *I* input crossbars, *P* product crossbars, *O* outputs, is calculated according to the following formulae:

$$S_{org} = IP + OP;$$

$$S_{TMR} = 3IP + 3OP + 9P;$$

$$S_{AO} = 4IP + 2OP;\ S_{AOO} = 2IP + 2OP + 2O;$$

$$S_{AAOO} = 2IP + 2OP + 2P + 20;$$

$$S_{AOOA} = 2IP + 4OP + 6O.$$

These formulae will be used in Section IV for comparison of the proposed solution with the above fault-tolerant architectures.

III. Six Matrix architecture for nano-PLA

In this chapter, we introduce the Six Matrix (SM) PLA architecture of the FSM oriented on the fault tolerant nano-PLA. We use an example of the FSM defined by Table I. The general structure of the SM PLA architecture is presented in Fig. 2.

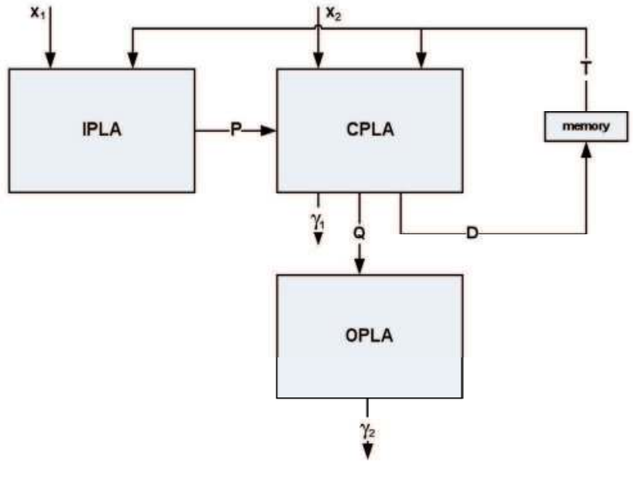


Fig. 2. Six Matrix architecture

Let the combinational part of the initial FSM implements the transformation $(X \cup T) \Rightarrow (Y \cup D)$, where:

- $X = \{x_1, \ldots, x_L\}$; is the set of input variables, $X = X_1 \cup X_2,\ X_1 \cap X_2 = \varnothing$,
- $Y = \{y_1, \ldots, y_N\}$; is the set of output variables, $Y = \Upsilon_1 \cup \Upsilon_2,\ \Upsilon_1 \cap \Upsilon_2 = \varnothing$;
- $T = \{t_1, \ldots, t_R\}$; is the set of present state variables;
- $D = \{d_1, \ldots, d_R\}$; is the set of the next state variables.

Two main transformations form the SM architecture: the transformation of input variables and the transformation of output variables. Both of the transformations are implemented by corresponding PLAs. As a result, the final SM structure comprises three PLAs: an inputs transformation PLA (IPLA), a core transformation PLA (CPLA) and an outputs transformation PLA (OPLA). Transformations performing by the PLAs are the following:

1. Inputs transformation $(X_1 \cup T) \Rightarrow P$;
2. Core transformation: $(X_2 \cup P \cup T) \Rightarrow (Q \cup \Upsilon_1 \cup D)$;
3. Outputs transformation: $Q \Rightarrow \Upsilon_2$.

Two additional sets of auxiliary variables are introduced into the SM architecture:

$P = \{p_1, \ldots, p_K\}$ - the set of output variables of IPLA (auxiliary input variables of the CPLA);

$Q = \{q_1, \ldots, q_M\}$ - the set of output variables of CPLA (auxiliary input variables of the OPLA).

While the two transformations PLAs perform combinational functions of the corresponding transformations, the core transformation PLA corresponds to a core FSM that performs the main functions of the initial FSM. Notice that the design of the Core PLA depends on the functionality of the IPLA and the OPLA.

A. Inputs transformation PLA

Let $X(a_i)$ be a set of input variables that determine the transitions from state a_i . In our example:

$$X(a_1) = \{x_1\};\ X(a_5) = \{x_4, x_6\};\ X(a_2) = \varnothing;$$

$$X(a_6) = \{x_5, x_7\};\ X(a_3) = \{x_7, x_8\};\ X(a_7) = \varnothing;$$

$$X(a_4) = \{x_1, x_2, x_3\};\ X(a_8) = \{x_6\}.$$

The idea of transformation of the original input variables can be formulated as a problem of replacing the set of variables X with a set of new variables P . A number of variables in the set P may be smaller or equal to the maximal number of variables in sets $X(a_i)$. In our example: $|P| \le 3$.

To optimize the PLA implementation of the inputs transformation PLA, a specific state assignment has to be applied. Let, in our example:

$$K(a_1)=110;\ K(a_2)=000;\ K(a_3)=101;$$
$$K(a_4)=010;\ K(a_5)=011;\ K(a_6)=100;$$
$$K(a_7)=111;\ K(a_8)=001$$

Table II illustrates one of possible variants of the replacement of variables X with variables P.

TABLE II. REPLACEMENT OF INPUT VARIABLES

a_m	p_1	p_2	p_3
a_1	x_1	-	-
a_2	-	-	-
a_3	x_7	x_8	
a_4	x_1	x_2	x_3
a_5	x_4	x_6	
a_6	x_7	x_5	
a_7	-	-	-
a_8	-	x_6	-

It is clear from the table that variable x_3 may be routed directly to the Core PLA. Namely:

$$X_2=\{x_3\},\ \text{and}\ X_1=X\setminus X_2.$$

After minimization, these encoding variables can be expressed as follows:

$$p_1=t_2\bar{t}_3x_1+\bar{t}_2x_7+t_2t_3x_4;$$
$$p_2=t_1t_3x_8+t_2\bar{t}_3x_2+\bar{t}_1t_3x_6+\bar{t}_2\bar{t}_3x_5.$$

Obviously, the above expressions can be directly implemented by a PLA structure having a small area.

B. Outputs transformation PLA

The outputs transformation PLA transforms a set of variables Q to a subset of output variables Υ_2.

Notice that each of the output variables y_i is present in a number of Y_j -sets. In our example (Table I), $J=10$ and:

$$Y_0=\varnothing;\ Y_1=\{y_2,y_5,y_{10}\};Y_2=\{y_3,y_4\};Y_3=\{y_1,y_3,y_4\};$$
$$Y_4=\{y_{10},y_{11}\};Y_5=\{y_6,y_8\};Y_6=\{y_1,y_3\};$$
$$Y_7=\{y_9,y_{14}\};Y_8=\{y_7\};Y_9=\{y_6,y_{13}\};Y_{10}=\{y_{12}\}$$

Each Y_j -set may be associated with a binary code (vector) $K(Y_j)$ of length M, $M\geq\lceil\log_2 J\rceil$. Let B_j be a minterm of variables $q_1,\ldots,q_M$ corresponding to $K(Y_j)$. Consequently, each of the Y_j -sets can be mapped to the minterm and the output variables $y_i\ (i=1,\ldots,N)$ can be expressed as a sum of the minterms $B_j,(j=1,\ldots,J)$. In our example:

$$y_1=B_3+B_6;\ y_2=B_1;\ y_3=B_2+B_3+B_6;$$
$$y_4=B_2+B_3;\ y_5=B_1;\ y_6=B_5+B_9;\ y_7=B_8;$$
$$y_8=B_5;\ y_9=B_7;\ y_{10}=B_1+B_4;\ y_{11}=B_4;$$
$$y_{12}=B_{10};\ y_{13}=B_9;\ y_{14}=B_7.$$

Obviously, these expressions can be implemented by a PLA structure.

The set Υ_1 consists of all the output variables produced by single product terms. The remaining output variables form the set Υ_2. The output variables of Υ_2 define a set of punctured $\widetilde{Y}_j$ -sets. In our example,

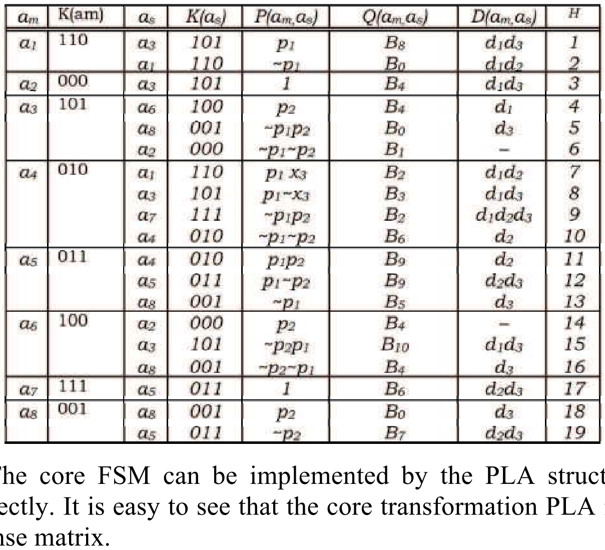
$$\Upsilon_2=\{y_1,y_3,y_4,y_6,y_{10},y_{11},y_{13}\}\ \text{and}\ \Upsilon_1=Y\setminus\Upsilon_2.$$

Therefore the punctured $\widetilde{Y}_1$ -set corresponding to the original Y_1 -set equals to $\widetilde{Y}_1=\{y_{10}\}$. Each punctured $\widetilde{Y}_j$ -set is associated with a certain code. The number of the coded bits is $M=\log_2\lceil\{\widetilde{Y}_j\}\rceil$. In our case: $|\{\widetilde{Y}_j\}|=8$, hence each code is represented by a minterm of the variables q_1,q_2,q_3.

C. Core PLA

The core transformation PLA implements the Core FSM, which means the transformation $(X_2\cup P\cup T)\Rightarrow(Q\cup\Upsilon_1\cup D)$. The Core FSM differs from the initial FSM by its inputs and outputs. It uses new P variables instead of input variables X_1, and new Q variables instead of output variables Υ_2. The transition functions and the next state functions of the core FSM remain the same as in the initial FSM. The core FSM for our example is presented by Table III.

TABLE III. STRUCTURAL TABLE OF THE CORE FSM



a_m	K(am)	a_s	$K(a_s)$	$P(a_m,a_s)$	$Q(a_m,a_s)$	$D(a_m,a_s)$	H
a_1	110	a_3	101	p_1	B_8	d_1d_3	1
		a_1	110	$\sim p_1$	B_0	d_1d_2	2
a_2	000	a_3	101	1	B_4	d_1d_3	3
a_3	101	a_6	100	p_2	B_4	d_1	4
		a_8	001	$\sim p_1p_2$	B_0	d_3	5
		a_2	000	$\sim p_1\sim p_2$	B_1	–	6
a_4	010	a_1	110	$p_1\ x_3$	B_2	d_1d_2	7
		a_3	101	$p_1\sim x_3$	B_3	d_1d_3	8
		a_7	111	$\sim p_1p_2$	B_2	$d_1d_2d_3$	9
		a_4	010	$\sim p_1\sim p_2$	B_6	d_2	10
a_5	011	a_4	010	p_1p_2	B_9	d_2	11
		a_5	011	$p_1\sim p_2$	B_9	d_2d_3	12
		a_8	001	$\sim p_1$	B_5	d_3	13
a_6	100	a_2	000	p_2	B_4	–	14
		a_3	101	$\sim p_2p_1$	B_{10}	d_1d_3	15
		a_8	001	$\sim p_2\sim p_1$	B_4	d_3	16
a_7	111	a_5	011	1	B_6	d_2d_3	17
a_8	001	a_8	001	p_2	B_0	d_3	18
		a_5	011	$\sim p_2$	B_7	d_2d_3	19

The core FSM can be implemented by the PLA structure directly. It is easy to see that the core transformation PLA is a dense matrix.

After the minimization, the resulting six-matrix structure for our example looks as shown in Fig. 3.

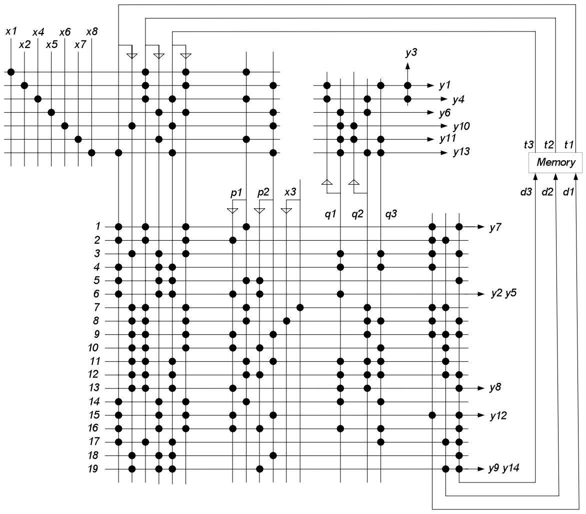


Fig. 3. Optimized Six Matrix structure for the example

The resulting SM scheme built for our example has the total area equals 479 crosspoint. Taking into account that the direct matrix implementation requires 608 crosspoints, we have 22% area reduction by using the proposed SM scheme.

The use of the SM synthesis approach in the TMR nano-PLA structure yields area of 1530. The TMR scheme, for the standard direct FSM implementation, gives 1914.

After applying the four tautology techniques to the SM implementation for our example, we obtain the following values of the area:

$$S_{AO} = 1664;\ S_{AOO} = 984;\ S_{AAOO} = 1048;\ S_{AOOA} = 1288.$$

IV. Experiments and estimation of the proposed solution

Two main characteristics are commonly used as optimization criteria in synthesis by nano-PLAs, the required area and the required total number of devices.

Notice that the SM technique has a potential of reducing the additional devices amount by using optimized encoding of auxiliary variables and other optimization techniques. However, to show the potential of our approach, we have used a simple straightforward encoding toward area minimization and without optimizing the number of devices.

After decomposing a PLA corresponding to the initial FSM into three interacting matrices, a big number of fault tolerant nano-PLA structures implementing the FSM can be generated. Indeed, each of the three portions of the SM architecture can be implemented by using one of the four tautology schemes [14] or by using the TMR scheme. This provides a significant level of flexibility in searching the optimal solution.

In our experiments, we used a set of FSM benchmarks (http://www.tau.ac.il/~ilia1/benchmarks.htm) to test the proposed SM approach in various tautology schemes. For this purpose, we decomposed all the benchmarks according to the proposed approach and estimated the required area overhead and the devices overhead, when all three of the SM PLAs were implemented by the same technique, that is: TMR, A-O, A-O-O, A-A-O-O or A-O-O-A architectures. The experiments were performed by using a system EDA tool *Abelite* [3].

Experimental results for the area and the devices are shown in Tables IV and V respectively. In both of the tables, each benchmark is represented by two rows – the first one corresponds to the direct implementation and the second – to the decomposed (SM) implementation. Columns of the table are: a benchmark name, numbers of inputs, outputs and product terms in the benchmark correspondingly, numbers of devices (Table V), the method of implementation (direct and SM), the area for the non fault tolerant FSM implementation, the area for the TMR scheme. The last four columns correspond to the four tautology schemes correspondingly.

The last two rows of Tables IV and V comprise average area for the direct $\left(W_{direct}\right)$ and the SM $\left(W_{SM}\right)$ solutions normalized by the corresponding area required for the direct PLA implementation of the original FSM.

TABLE IV. BENCHMARK RESULTS - AREA

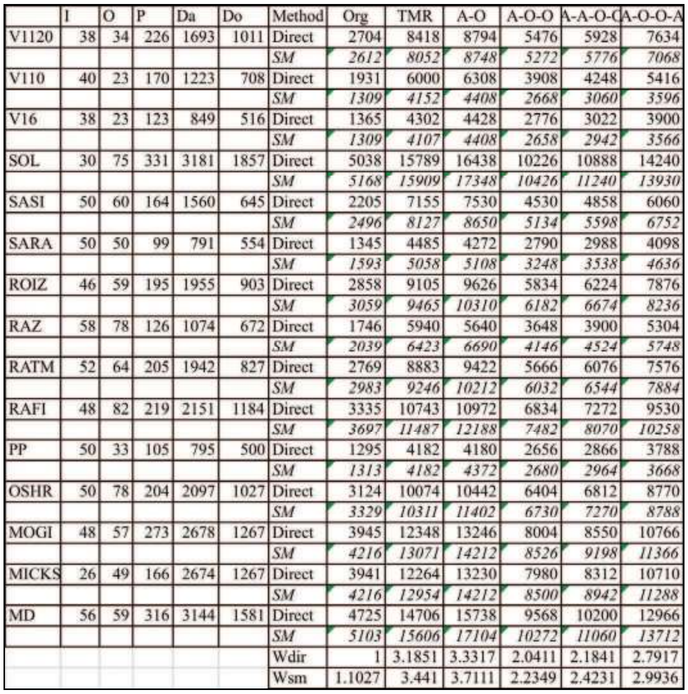
	I	O	P	Method	Org	TMR	A-O	A-O-O	A-A-O-O	A-O-O-A
V1120	38	34	226	Direct	16272	49122	49720	32612	33064	48116
				SM	*10020*	*30276*	*35132*	*20088*	*20592*	*25132*
V110	40	23	170	Direct	10710	32337	35020	21466	21806	29378
				SM	*7503*	*22734*	*26536*	*15056*	*15448*	*18632*
V16	38	23	123	Direct	7503	22716	24354	15052	15298	20802
				SM	*5133*	*15579*	*18094*	*10306*	*10590*	*12824*
SOL	30	75	331	Direct	34755	104940	89370	69660	70322	119610
				SM	*21037*	*63516*	*72406*	*42164*	*42978*	*54086*
SASI	50	60	164	Direct	18040	54660	52480	36200	36528	56120
				SM	*9494*	*29121*	*28166*	*19130*	*19594*	*29224*
SARA	50	50	99	Direct	9900	30150	29700	19900	20098	30000
				SM	*6376*	*19407*	*21844*	*12814*	*13104*	*16598*
ROIZ	46	59	195	Direct	20475	61956	58890	41068	41458	64314
				SM	*11164*	*33780*	*38568*	*22392*	*22884*	*28608*
RAZ	58	78	126	Direct	17136	52110	48888	34428	34680	54396
				SM	*8280*	*25146*	*28464*	*16628*	*17006*	*21420*
RATM	52	64	205	Direct	23780	71916	68880	47688	48098	74184
				SM	*12218*	*36951*	*42204*	*24502*	*25014*	*31302*
RAFI	48	82	219	Direct	28470	86148	77964	57104	57542	93348
				SM	*15488*	*46860*	*52072*	*31064*	*31652*	*41120*
PP	50	33	105	Direct	8715	26442	27930	17496	17706	24558
				SM	*5352*	*16299*	*18872*	*10758*	*11042*	*13402*
OSHR	50	78	204	Direct	26112	79038	72624	52380	52788	84516
				SM	*12172*	*36840*	*41692*	*24416*	*24956*	*31556*
MOGI	48	57	273	Direct	28665	86508	83538	57444	57990	88794
				SM	*17184*	*51975*	*58332*	*34462*	*35134*	*45054*
MICKS	26	49	166	Direct	12450	37791	33532	24998	25330	41462
				SM	*9649*	*29253*	*33400*	*19366*	*19808*	*24698*
MD	56	59	316	Direct	36340	109551	108072	72798	73430	110322
				SM	*19250*	*58047*	*67436*	*38566*	*39354*	*48262*
				Wdir	1	3.0256	2.910408	2.005689	2.02546	3.106658
				Wsm	0.575	1.74173	1.979668	1.153746	1.17901	1.481571

The results presented in Tables IV and V clearly indicate efficiency of the SM method. The SM has advantages over the conventional direct solution in all the four fault tolerant schemes. The most impressive advantage is achieved for the A-O-O scheme giving more than 50% reduction in the area overhead in comparison with the direct solution. In the case of direct implementation, using fault tolerant schemes A-O and A-O-O-A seems not efficient since the required overhead occurs to be approximately the same as in the TMR scheme. Nevertheless, after applying the proposed SM approach, these schemes A-O and A-O-O-A produce overhead reduction of about 40% and 50% respectively. At the same time, when using the SM approach, the best solutions are provided by the A-A-O and the A-A-O-O schemes, while the A-O and the A-

O-O-A schemes still give the overhead close to the TMR SM solution.

The benchmark results show that all fault tolerant FSMs implemented according to the SM architecture have the significantly lower area than the same FSMs would they be implemented by conventional fault-tolerant techniques. For one and the same fault-tolerant technique, the proposed method provides the area reduction of about 50%. The SM method requires certain amount of additional devices to be introduced into the resulting scheme. For the majority of the benchmarks, the devices overhead does not exceed 10%, which have to be considered as relatively small penalty for achieving the above area reduction.

TABLE V. BENCHMARK RESULTS - DEVICES



	I	O	P	Da	Do	Method	Org	TMR	A-O	A-O-O	A-A-O-O	A-O-O-A
V1120	38	34	226	1693	1011	Direct	2704	8418	8794	5476	5928	7634
						SM	*2612*	*8052*	*8748*	*5272*	*5776*	*7068*
V110	40	23	170	1223	708	Direct	1931	6000	6308	3908	4248	5416
						SM	*1309*	*4152*	*4408*	*2668*	*3060*	*3596*
V16	38	23	123	849	516	Direct	1365	4302	4428	2776	3022	3900
						SM	*1309*	*4107*	*4408*	*2658*	*2942*	*3566*
SOL	30	75	331	3181	1857	Direct	5038	15789	16438	10226	10888	14240
						SM	*5168*	*15909*	*17348*	*10426*	*11240*	*13930*
SASI	50	60	164	1560	645	Direct	2205	7155	7530	4530	4858	6060
						SM	*2496*	*8127*	*8650*	*5134*	*5598*	*6752*
SARA	50	50	99	791	554	Direct	1345	4485	4272	2790	2988	4098
						SM	*1593*	*5058*	*5108*	*3248*	*3538*	*4636*
ROIZ	46	59	195	1955	903	Direct	2858	9105	9626	5834	6224	7876
						SM	*3059*	*9465*	*10310*	*6182*	*6674*	*8236*
RAZ	58	78	126	1074	672	Direct	1746	5940	5640	3648	3900	5304
						SM	*2039*	*6423*	*6690*	*4146*	*4524*	*5748*
RATM	52	64	205	1942	827	Direct	2769	8883	9422	5666	6076	7576
						SM	*2983*	*9246*	*10212*	*6032*	*6544*	*7884*
RAFI	48	82	219	2151	1184	Direct	3335	10743	10972	6834	7272	9530
						SM	*3697*	*11487*	*12188*	*7482*	*8070*	*10258*
PP	50	33	105	795	500	Direct	1295	4182	4180	2656	2866	3788
						SM	*1313*	*4182*	*4372*	*2680*	*2964*	*3668*
OSHR	50	78	204	2097	1027	Direct	3124	10074	10442	6404	6812	8770
						SM	*3329*	*10311*	*11402*	*6730*	*7270*	*8788*
MOGI	48	57	273	2678	1267	Direct	3945	12348	13246	8004	8550	10766
						SM	*4216*	*13071*	*14212*	*8526*	*9198*	*11366*
MICKS	26	49	166	2674	1267	Direct	3941	12264	13230	7980	8312	10710
						SM	*4216*	*12954*	*14212*	*8500*	*8942*	*11288*
MD	56	59	316	3144	1581	Direct	4725	14706	15738	9568	10200	12966
						SM	*5103*	*15606*	*17104*	*10272*	*11060*	*13712*
						Wdir	1	3.1851	3.3317	2.0411	2.1841	2.7917
						Wsm	1.1027	3.441	3.7111	2.2349	2.4231	2.9936

V. CONCLUSIONS

The paper deals with fault tolerance techniques for nanoelectronic PLAs. Tautology based fault masking schemes are used as a base for our study. We have proposed to decompose a combinational part of an FSM into three component PLAs connected in series (SM architecture). The proposed decomposition allows selectively applying various fault tolerant techniques to the components. Our decomposition is performed in such a way that each component PLA has high density, i.e. quite a low percent of the PLA cross points include devices. As a result, such decomposition minimizes the area required for the FSM implementation.

The proposed SM decomposition method brings the following benefits, which are significant in synthesis of nano computing structures.

1. Diagnostics possibility. Since the SM architecture comprises three smaller PLA portions, it may be tested and diagnosed easier than the standard one-PLA scheme.
2. Flexibility. In the SM architecture, each of the component PLAs may be implemented by any of the known fault tolerant technique and the best solutions can be selected for the resulting scheme.
3. Optimization possibility. The SM architecture provides a possibility of the “area - devices number” trade-off, which may be achieved by optimized encoding of auxiliary variables, states assignments etc.

The obtained results demonstrate the high potential of the presented Six Matrix approach and open future research directions.

REFERENCES

[1] Astaf’ev, M., Levin, I., Matrosova, A., Sinelnikov, V. “Self-Testing Automaton Networks: Their Design in Programmable Logical Matrices”, *Automation and Remote Control*, 63(10), 1637-1652, 2002.

[2] Baranov S., *Logic Synthesis For Control Automata*. Kluwer Academic Publishers, Norwell, MA, USA, Pages: 408, 1994.

[3] Baranov, S. *Logic and System Design of Digital Systems.* TTU Press and SiB Publishers, Tallinn, 2008.

[4] Chen Y., Jung G. Y., Ohlberg D., Stewart D. R., Jeppesen J. 0., Nielsen K. A., Stoddart J. F. and Williams R. S., “Nanoscale Molecular-switch Crossbar Circuits”, *Nanotechnology*, vol. 14, pp. 462-468, 2003.

[5] Fuchs W. K., Chen C. R., Abraham J. A. “Error Detection in Highly Structured Logic Arrays”, *IEEE Journal of Solid-State Circuits*, Vol. 22, n. 4, pp. 583-594, August 1987.

[6] J. Khakbaz, E. J. McCluskey, “Concurrent Error Detection and Testing for Large PLA's”. *IEEE Journal of Solid-State Circuits*, Apr 1982, Volume: 29, Issue: 4, pages: 756- 764.

[7] Lala, P. K., Tao, D. L. “On fault-tolerant PLA design”, *IEEE Proceedings Southeastcon '90*, vol.3, 1990, pp. 945-947.

[8] Levin I. “Decompositional Design of Automata Based on PLA with Memory”, *Automatic Control and Computer Sciences*, Vol. 20, No. 2, 1986, pp. 61-68.

[9] Levin, I. “Hierarchical Model of the Interaction of Microprogrammed Automata”, *Automatic Control and Computer Sciences*, Vol. 21, No. 3, 1986, 67-73.

[10] Levin I., Karpovsky M., “On-line Self-Checking of Microprogram Control Units”, *4-th IEEE International On-line Testing Workshop*, Capri, Compendium of papers, 1998, pp. 153-159.

[11] H. Naeimi and A. DeHon, “A Greedy Algorithm for Tolerating crosspoints in Nano PLA design”, *Proceedings IEEE International Conference on Field-Programmable Technology*, 2004, pp. 49-56,

[12] Nikolic, K. Sadek, A. Forshaw, M., “Architectures for reliable computing with unreliable nanodevices”, *Proceedings of the 2001 1st IEEE Conference on Nanotechnology,* 2001, pp. 254-259.

[13] Tahoori M. B., “Defect Tolerance in Crossbar Array Nano-Architectures”, In: *Emerging Nanotechnologies. Test, Defect Tolerance, and Reliability, Series: Frontiers in Electronic Testing*, M. Tehranipoor, (Ed.), Chapter 5, Pages 121-152, Springer, 2008, pp.121-152.

[14] Wenjing Rao, Alex Orailoglu, Ramesh Karri, “Logic Level Fault Tolerance Approaches Targeting Nanoelectronics PLAs”, *Design, Automation & Test in Europe Conference & Exhibition,* 2007, pp. 1 – 5.

[15] Wenjing Rao, Alex Orailoglu, Ramesh Karri, “Fault Tolerant Approaches to Nanoelectronic Programmable Logic Arrays”, *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks,* 2007, pp. 216 –224.