IEEE Int. Test Conf., pp.574-582, Nov. 1985.

A TESTABLE DESIGN OF PROGRAMMABLE LOGIC ARRAYS WITH UNIVERSAL CONTROL AND MINIMAL OVERHEAD

Kewal K. Saluja

Dept. Elect. & Comp. Eng.
University of Newcastle
N.S.W. 2308 Australia

Hideo Fujiwara

Dept. Electronic& Communications
Meiji University
Kawasaki 214 Japan

Kozo Kinoshita

Dept. Inf. & Beh. Sci.
Hiroshima University
Hiroshima 730 Japan

ABSTRACT

In this paper we propose a new design technique for designing testable PLAs with minimal overhead. In our method we marry a number of existing design methods. We then define silicon area overhead as a cost function and minimize the cost (i.e. overhead) while maintaining the testability property of PLAs.

INTRODUCTION

With the growth in the complexity of VLSI circuits, the only way to carry out a circuit design to completion is by not only enlisting the help of design automation (DA) tools but by making use of regular structures in the design process. Although, with the use of DA tools almost any regular structure can be designed without much difficulty, yet some regular structures have become integral parts of DA systems and of Integrated Circuits (IC). Most commonly used regular structures being RAMs, ROMs and PLAs (Programmable Logic Arrays). Use of PLAs in ICs is gaining increasing popularity for reasons:

(1) It is a powerful structure to realize arbitrary combinational as well as sequential circuits. Therefore, its use reduces the overall complexity of the chip design.
(2) PLAs can often be implemented as testable structures[1-11] thus making the otherwise complex problem of testing VLSI circuits of manageable proportions.
(3) It is easy to include engineering design changes in IC's designed using PLAs.
(4) Hardware and silicon area required to implement a PLA can often be further reduced by using PLA minimization methods[12-13].

In almost all testable designs of PLAs, enhancement in testability is achieved through use of additional logic to control individual product lines in test mode. Typically, such a control is achieved by one of the following two methods:

(1) Using a shift register (SR) or shift register with multiplexer[1,3-7].
(2) Using extra bit lines to form a decoder or decoder like structure[2,8,9].

As both the methods provide almost equal fault-coverage, the superiority of a method depends on the amount of silicon area overhead. Bozorgui-Nesbat and McCluskey[9] argue that it is difficult to construct register cells with the same pitch as PLA pitch, thus a design employing SR is likely to be inferior than a design which uses extra bit lines. On the otherhand, Hua et al[7] and others[14] argue the use of SR with multiplexers and SR with extra inputs or SR wrapped around a PLA to obtain a low silicon area overhead.

In general depending on the function realized by a PLA, number of inputs, number of product lines and number of outputs; any of the two methods can results into a testable design with lower silicon area overhead. Although in Built-in Self Test PLAs, use of shift registers and shift register like structures are more prevalent[7,17-19]

In this paper we marry the two approaches and show that the resulting design not only has the required fault coverage but results into an area overhead lower than the either approach.

This paper is organised as follows. In Section 2 of this paper we give preliminaries and describe the required notation and the details of the methods which are to be integrated in this paper. In Section 3 we propose the design of a Universal Control and discuss its properties. In Section 4 we describe as to how partitioning can be employed to merge the universal control and the use of SR concepts. In Section 5 we find the optimal length of SR and size of control, analytically, such that overhead is minimized. Asymptotic bounds on overhead and further reduction of overhead for non-universal control are discussed in Sections 6 and 7 respectively.

2. PRELIMINARIES

In this Section we present the notation and known testable designs. We also include some of the known results for the sake of completeness of this paper.

From logical description point of view, PLAs are two level sum-of-product realizations of combinational logic functions. Although in a given technology their implementation may not be AND/OR. For example nMOS PLAs are NOR-NOR implementations. However, for our analysis and presentation of results we choose AND/OR realization. Conversion of the tests derived in this paper to suit NOR-NOR and other forms is a simple matter. Similarly, fault coverage results also apply to other forms of PLAs.

A general PLA structure is shown in Figure 1. Input decoders, D_i, shown in this figure will be assumed to be one-bit decoders providing true and compliment bit lines to the AND plane. For simplicity of presentation other details of a PLA structure, (e.g. pull-ups, grounds, etc.) will now be shown. Program points of a PLA are intersection of bit lines and product lines in the AND-plane and product lines and output lines in the OR-plane. A PLA can be completely described by its personality matrices for AND and OR planes. We shall denote the size of a PLA by an ordered triple (n, m, ℓ) having n: number of inputs; m: number of product-lines and ℓ: number of outputs. The personality matrix A for AND plane is a nxm matrix whose entries are from the set $\{1,0,-\}$. A 1(0) in the position a_{ij} means that ith input (complement of ith input) and jth product line have a crosspoint present in the PLA. Entry a_{ij} = - means jth product line is independent of ith input.

The personality matrix Q for OR plane is a ℓxm matrix whose entries are from the set $\{1,-\}$.

Typically, all testable designs incorporate extra logic to enhance testability. A general structure of a testable PLA is shown in Figure 2. While discussing different testable design we shall often refer to this figure.

2.1 Fault Model

We shall assume that only following faults can be present in a PLA:
(1) Any number of s-a-faults,
(2) Any number of extra devices, and
(3) Any number of missing devices.

Though we do not assume the presence of adjacent line bridging faults, yet they can be detected with only a minor modification in some of the cases discussed in this paper. Also, as Saluja et al[11] have shown that s-a-faults in a PLA are equivalent to missing device or output s-a-0 faults, therefore, while presenting the proofs we shall only consider the fault set modified accordingly.

2.2 Bozorgui-Nesbat and McCluskey's (BM) Approach[9]

In this approach LP (Figure 2) consists of a decoder-like structure. Extra inputs are added such that the resulting A matrix has certain distance properties. LI essentially consists of circuit which can be enabled only during test mode (in n-MOS it will consist of pull down transistors on extra lines). We shall call the PLA obtained with this approach a BM-PLA. In our notation, P_i denotes ith product line as well as function realized by ith product line. An n-bit input vector will be denoted as $X = (c_1, c_2, \ldots, c_n)$.

Definition 1: A set of inputs S_i, is called a *select set of* P_i if $S_i = \{X/P_i(X) = 1\}$.

Definition 2: An input vector $t_{i,0}$ is called *main test pattern of* P_i if : 1. $t_{i,0} \in S_i$; 2. $d_H(t_{i,0},X) \geq 2$ for all X and S_j such that $X \in S_j$ and $j \neq i$. Where $d_H(a,b)$ is Hamming distance between vectors a and b.

Definition 3: For a main test pattern $t_{i,0} = (c_1, c_2, \ldots, c_n)$ for P_i we define *n auxiliary test patterns of* P_i as $t_{i,j} = (c_1, c_2, \ldots, c_{j-1}, \overline{c}_j, c_{j+1}, \ldots, c_n)$; $j = 1, 2, \ldots, n$.

Definition 4: *Test patterns for* P_i, TP_i, is the set defined as: $TP_i = \{t_{i,0}, t_{i,1}, \ldots, t_{i,n}\}$.

Definition 5: *Test set for BM-PLA*, T_{BM} is defined as: $T_{BM} = TP_1 \cup TP_2 \cup \ldots \cup TP_m$.

Theorem 1[9]: A BM-PLA can be tested for all faults by the test set T_{BM}.

A procedure for adding extra inputs such that the resulting PLA is easily testable is given in 9. Number of extra inputs depend on the A matrix of the original PLA. Computational complexity of the procedure which changes a PLA to BM-PLA is $O(m^3)$.

2.3 Khakbaz's (K) Approach[5]

In this approach LP consists of a SR to control the individual product lines. No extra logic is required at LI. An extra observable output is added. We shall call the PLA obtained by this approach a K-PLA. In deriving the tests for K-PLA, a test vector will be denoted by a (m+n)-bit vector, $(r_1, r_2, \ldots, r_m;\ c_1, c_2, \ldots, c_n)$. The first m-bits denote the contents of SR, and remaining n-bits indicate the inputs. We shall often group them together and denote the test vector as (R, X). Following R and X vectors are of special interest for deriving tests for K-PLA.

Definition 6: Vectors R_0 and R_i are defined as follows: $R_0 = (0, 0, \ldots, 0)$; $R_i = 0, 0, \ldots, \underset{\text{ith}}{0, 1}, 0, \ldots, 0)$

Definition 7: An input X_i is called a *Test Input for* P_i if $X_i \in S_i$ (see Definition 1 for S_i).

Definition 8: If X_i (test input for P_i) is denoted as $(c_1, c_2, \ldots, c_n)$, we define *n auxiliary test inputs for* P_i as $X_{i,j} = (c_1, c_2, \ldots, \overline{c}_j, \ldots, c_n)$; $j = 1, 2, \ldots, n$.

Definition 9: *Test Vectors for* P_i, TV_i, is the set $TV_i = \{(R_i;X_i), (R_i;X_{i,1}), (R_i;X_{i,2}), \ldots, (R_i,X_{i,n})\}$.

Definition 10: *Test Set for K-PLA*, T_K, is defined as $T_K = TV_0 \cup TV_1 \cup TV_2 \cup \ldots \cup TV_m$ where $TV_0 = \{(R_0;X_1), (R_0;X_2), \ldots, (R_0;X_m)\}$.

Theorem 2[5]: A K-PLA can be tested for all faults by the test set T_K.

2.4 Saluja et al (SKF)[4] and Fujiwara (F)[6] Approaches

In these approaches LP consists of a SR to control the individual product lines. LI consits of additional controllable logic in the form of SR[4] or gates[6] to provide inputs to the AND plane of the PLA such that the resulting PLA can be tested by test sets independent of the personality of a PLA. We do not describe the necessary test sets here, as it will make this paper unduly long. However, following theorems is a consequence of these works. In a later section in this paper we will apply the key idea used in SKF-PLA and F-PLA to arrive at a new design.

Theorem 3[4,6]: SKF-PLA and F-PLA can be tested for all faults by test sets independent of the PLA. Furthermore, SKF-PLA and F-PLA can also be tested for adjacent line bridging faults.

2.4 Comparison

Comparative advantages and disadvantages of the different designs discussed above are listed in Table 1.

Table 1 Comparison of Different Testable Designs

	BM-PLA	K-PLA	SKF-PLA & F-PLA
Extra Logic, L_P			
(1) Type	Decoder like	SR	SR
(2) Dependence on PLA	Yes	No	No
(3) Extra inputs	f(PLA)	1	1
(4) Computation complexity	$O(m^3)$	Nil	Nil
(5) Cost	f(PLA)	f(m)	f(m)
Extra Logic L_I			
(1) Type	Controllable extra inputs	Nil	SR/gates
(2) Delay inserted	Nil	Nil	Yes
(3) Cost	Constant	Nil	f(n)
Test Generation			
(1) Required	Yes	Yes	No
(2) Length	f(n, m)	f(n, m)	f(n, m)
(3) Fault Coverage	s-a and cross-points	s-a and cross-points	s-a., cross-points and bridging
(4) Stored test patterns	m	m	constant (2)

f(x, y, ..) means a function of x,y etc

f(PLA) means a function of personality of PLA

3. UNIVERSAL CONTROL

In this Section we propose a design of LP which makes use of extra inputs and bit lines similar to BM-PLAs. However, the design proposed will be universal in nature, i.e. independent of the personality of PLA. The algorithm described in 9 for adding extra inputs to the PLA has a computational complexity of $O(m^3)$. For large PLAs, use of such an algorithm can be prohibitively expensive, thus restricting the use of this method only to small PLAs.

We shall first study some properties of a special AND-array, called Decoder-Parity-AND-Array (DPAA), defined below.

Definition 11*: A *DPAA* is an AND-Array with m product lines $\lceil \log_2 m \rceil + 1$ inputs and with the A matrix A_D as follows: If m columns of A_D matrix are numbered from A_0 to A_{m-1}, and $A_i = [a_{1,i}, a_{2,i}, \ldots, a_{n,i}, a_{n+1,i}]^T$ then $(a_{n,i}, \ldots, a_{2,i}, a_{1,i})$ is binary representation of i and $a_{n+1,i} = a_{n,i} \oplus \cdots \oplus a_{2,i} \oplus a_{1,i}$.

Following lemmas describe some properties of DPAA.

Lemma 1: $d_H(A_i, A_j) \geq 2; \quad 0 \leq i,j < m; \quad i \neq j$.

Lemma 2**: For the DPAA, the select set S_i of P_i is uniquely determined and consists of a single pattern. Notationally, $|S_i| = 1$ and $S_i = A_i^T$.

Following definition is a minor variation of the definition in 9 and is stated here for the sake of completeness of this paper.

Definition 12: *Distance matrix* D, of a PLA is a mxm matrix whose entries are defined as follows:
$$D_{i,j} = \min\{d_H(X,Y)/X \in S_i, Y \in S_j\}; \quad i \neq j;$$
$$= 2 \quad \text{for} \quad i = j.$$

Lemma 3: Every element of the distance matrix for the DPAA is greater than or equal to 2.

Proof: It follows from Lemmas 1 and Definition 12.

Design 1†: Augment a given (n, m, ℓ) PLA by concatinating a DPAA to it as shown in Figure 3(b). We shall call this PLA as D1-PLA. In this Figure we have not shown the extra logic which can be used to disable DPAA part of the PLA during normal operation of the PLA. In nMOS PLAs[16] such circuit will be of the form shown in Figure 3(c). Note that total number of extra inputs to D1-PLA are (1+h) where $h = \lceil \log_2 m \rceil$.

We now derive a test set for D1-PLA.

Let S_i be the select set of P_i for the original PLA. Let $\{u_i\}$ be the select set of P_i for the decoder-parity PLA. Then, the select set of P_i for the augmented PLA is

$$u_i.S_i = \{(u_i,X) \mid X \in S_i\} \quad .$$

Lemma 4: For all $X \in S_i$ and $Y \in S_j$, $d_H[(u_i,X), (u_j,Y)] \geq 2$.

This lemma guarantees that any element of the select set of P_i for the augmented PLA can be a main test pattern of P_i.

Let $t_{i,0}$ be any one of the elements of $u_i.S_i$. For a $t_{i,0}$, we can obtain the auxiliary test patterns

* Y^T means transpose of a vector Y.

** $|x|$ means cardinality of a set x.

† $\lceil y \rceil$ means smallest integer z such that $z \geq y$.

Table 2 Comparison of BM-PLA at D1-PLA

PLA name		Master	New ALU	Bas New	Recur	Traffic	ALU Test	CERBERUS	COND	BAR	RIMP
		15x27x13	15x26x28	8x33x27	7x9x9	5x8x7	14x36x21	18x50x37	10x24x2	8x29x25	12x39x21
Number of Extra inputs for:	BM-PLA	4	4	5	2	3	4	4	2	4	4
	D1-PLA	6	6	7	5	4	7	7	6	6	7
Computation complexity of generating	BM-PLA	$O(27^3)$	$O(26^3)$	$O(33^3)$	$O(9^3)$	$O(8^3)$	$O(36^3)$	$O(50^3)$	$O(24^3)$	$O(29^3)$	$O(39^3)$
	D1-PLA	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil

of P_i. Let these be $t_{i,1}, t_{i,2}, \ldots, t_{i,n+\lceil \log_2 m\rceil+1}$.

Then, the test patterns for the D1-PLA are:

$$
\begin{aligned}
T_{D1} = \{ & t_{1,0}, t_{1,1}, \ldots, t_{1,n+\lceil \log m\rceil+1}, \\
& t_{2,0}, t_{2,1}, \ldots, t_{2,n+\lceil \log m\rceil+1}, \\
& \vdots \\
& t_{m,0}, t_{m,1}, \ldots, t_{m,n+\lceil \log m\rceil+1} \} .
\end{aligned}
$$

Theorem 4: The D1-PLA of Figure 3(b) can be tested for all faults by the test set T_{D1} mentioned above.

Proof: It follows from Theorem 1 and Lemma 4.

Theorem 5: Length of the test set to test D1-PLA is $m(2+n+\log_2 m)$. All these test patterns can be generated from m main test patterns.

Comparison of this method with BM-PLA is given in Table 2 for 10 PLAs taken from 9. The number of extra inputs in D1-PLA is not much larger than BM-PLA, but the computational complexity for generating D1-PLA is nil. Minimization of overhead (number of inputs as well as silicon area) with the use of DPAA is the subject of the next two Sections.

4. PARTITIONING

In this Section we propose a partitioning procedure which helps reduce the number of extra inputs without sacrificing the testability properties of the PLA. The basic idea being a PLA is partitioned into k blocks, each block containing m/k product lines. A DPAA is appended to each block as shown in Figure 4. Furthermore, an SR of length k is introduced to control these blocks. The formal design description and the testability properties of the resulting PLA are given below.

Design 2: Product lines of a PLA are divided into k blocks with each block containing h product lines, i.e. $h = \lceil m/k \rceil$. Note that all but kth block have h product lines, whereas number of product lines in kth block are $m-(k-1)h$. A DPAA is appended to each block. All DPAAs receive the same inputs thus total number of extra inputs to the PLA are $1+\lceil \log_2 h \rceil$. An SR of length k is appended. The resulting PLA is called D2-PLA (Figure 4).

While discussing testing of D2-PLA we must not lose sight of the actual realization and physical failures. Keeping this in mind we make the following observations.

Observation 1: Saluja et al[11] have shown that in a PLA all s-a-faults can be reduced to multiple missing cross-point or outputs s-a-0 faults. However, s-a-1 faults at $r_{i_1}, r_{i_2}, e_{i_0}, e_{i_1}, \ldots$ etc. for Block i shown in Figure 5 must be tested explicitly. Although, s-a-1 faults at these locations will not interfere in the normal operation of a PLA, yet their presence may invalidate other tests. Thus our fault set is

(1) multiple cross-points (extra and/or missing) and output(s) s-a-0 faults.
(2) s-a-1 faults at location marked in Figure 5. Notice that this fault set includes the fault set given in Section 2.1.

Observation 2: In Figure 4, number of extra inputs are

(1) 1 for SR

(2) $1+\lceil \log_2 h \rceil$ for DPAAs.

However, if we take $h = 1$ then the design 2 should be degenerated to K-PLA requiring no DPAA. Similarly, for $h = m$ design 2 should be degenerated to D1-PLA requiring no SR.

Further, in the presence of SR we would need to observe the output r_{k_1} to test SR.

Observation 3: We can introduce an extra output, z^*, in the OR-plane similar to 4-6. In this case AND-plane can be completely tested by observing only z^*. However, to test OR-plane, as well as outputs s-a-0, all outputs need to be observed, therefore it is not necessary to introduce z^* output and complete testing can be carried through by observing the normal outputs of D2-PLA.

We now describe the testing of D2-PLA. Testing is carried in a number of conceptual steps described below. Though, in practice these steps can be merged to obtain a reduced test set.

Step 1: Testing SR - SR is tested by applying a sequence of 0s, followed by a 1, followed by 0s and observing the r_{k_1} output. This will also detect s-a faults at r_{i_1}'s in any block.

Step 2: Testing s-a-1 faults at locations marked in Figure 5 - Set $r_1 = r_2 = \ldots = r_k = 0$. Now apply m main test patterns. All normal outputs of the PLA must stay zero during this test for a fault free PLA.

Step 3: Testing PLA - Each block of the PLA is now tested independently. To test block i set $r_i = 1$ and $r_j = 0$ for $j \neq i$. Apply $h(2+n+\lceil \log_2 h \rceil)$ test patterns required to test block i.

This completes the testing procedure.

Following theorem states this result formally. The proof has not been included as it is straightforward.

Theorem 6: The D2-PLA of Figure 4 can be tested for all faults by tests stated in above three steps.

Theorem 7: Length of the test set to test D2-PLA is $m(3+n+\lceil \log_2 h \rceil)$.

Proof: This is total number of tests in Steps 2 and 3 above. Test for SR is included in these steps.

5. OPTIMAL PARTITIONING

In the previous Section we discussed as to how by using partitioning one can reduce the number of extra inputs without sacrificing the fault coverage. Of course, if reduction of number of extra inputs is the only objective than SKF-PLA, F-PLA or K-PLA are optimal (in general) by choosing $h = 1$. Here we set out objective as follows:

Objective: Find optimal partitioning such that the area of LP in D2-PLA is minimized. Notice that in D2-PLA only added logic is in the form of LP (other than disabling transistors for extra inputs - these are not shown in the figure to keep the figure simple). Therefore, minimizing area of LP will result in a minimal overhead PLA.

We now define the cost function. Let w_1 be the area of a PLA cell and w_2 be the area of an SR cell.

It is difficult, if not impossible, to have pitch of SR cells same as PLA pitch. However, in D2-PLA we need pitch of SR cells to be $h \times$ PLA pitch, to avoid any interconnection overheads. In the following discussion we assume such to be the case, i.e. h will be large enough to make interconnection overhead as zero. Notice interconnection overhead can also be reduced to zero by having SR wrap around PLA[14] or by use of SR with multiplexers[7].

In a D2-PLA increase in area of PLA by adding an extra bit line is mw_1. Therefore an extra input increases the area of PLA by $2mw_1$. Thus in D2-PLA the overhead

$$g = 2mw_1(1+\lceil \log h \rceil) + kw_2 \qquad (1)$$

where $k = \lceil m/h \rceil$.

Our objective is to find $h = f(m, w_1, w_2)$ such that g is minimized.

Note: (i) There are two special cases which are not included in equation (1) to keep the presentation simple. These are : (a) for $h = 1$ $g = mw_2$; (b) for $h = m$ $g = 2mw_1(1+\lceil \log m \rceil)$.

(ii) The solution for h must be found in the set of integers.

Following theorem simplifies the solution space for h considerably.

Theorem 8: If g is minimized for some h such that $2^{\alpha-1} < h < 2^\alpha$ then g is also minimized for $h = 2^\alpha$.

Proof: If g is minimized for $2^{\alpha-1} < h < 2^\alpha$ then $g_{min} = 2mw_1(1+\alpha) + \lceil m/h \rceil w_2$;

$g_{h=2^\alpha} = 2mw_1(1+\alpha) + \lceil m/2^\alpha \rceil . w_2$.

But $\lceil m/2^\alpha \rceil \leq \lceil m/h \rceil$ for $h < 2^\alpha$.

Therefore $g_{h=2^\alpha} \leq g_{min}$. But g_{min} is minimum therefore $g_{h=2^\alpha} = g_{min}$.

Implication of Theorem 8 is that while finding minimum value of g we need only consider those h which are some powers of 2. Therefore a solution can easily be found either by hand computations or by using computer. Problem can also be solved analytically by solving the equation

$$\frac{dg}{dh} = 0 \qquad (2)$$

To gain some insight into the solution method we first consider the case where $2^\alpha (=h)$ divides m.

Note a sufficient condition for this case is m to be a power of two.

Theorem 9: If h divides m then g is minimized for $h = 2^{\lceil \log_2(s/2 \ln 2)\rceil}$ or $2^{\lfloor \log_2(s/2 \ln 2)\rfloor}$ where where $s = w_2/w_1$.

Proof: By Theorem 8 we need only consider $h = 2^{\alpha}$. Equation (1) in this case reduces to

$$g = 2mw_1(1+\alpha) + \frac{m}{2^{\alpha}}.w_2 \qquad (3)$$

setting $dg/d\alpha = 0$ we have $d/d\alpha\ (g) = 2mw_1 - mw_2/2^{\alpha} \ln 2 = 0$, i.e. $2^{\alpha} = \frac{1}{2}.w_2/w_1.\ln 2$; $\alpha = \log_2(s/2 \ln 2)$; where $s = w_2/w_1$. However, α must be an integer. Therefore $\alpha = \lceil \log_2(\frac{1}{2}\ w_2/w_1\ \ln 2)\rceil$ or $\lfloor \log_2(\frac{1}{2}\ w_2/w_1\ \ln 2)\rfloor$.

The general solution is not as straightforward and leads to a recursive but approximate relation given by the following theorem.

Theorem 10: g is minimized when $h = 2^{\lfloor \log_2(s/2.m_1/m \ln 2)\rfloor}$ or $2^{\lceil \log_2(s/2\ m_1/m \ln 2)\rceil}$ where $s = w_2/w_1$; and using division algorithm we write $m = q.h + \gamma = m_1 + \gamma$; $0 \le \gamma < h$.

Proof: Proof is similar to the proof for Theorem 9. We set $\lceil m/h\rceil = \lceil m_1+\gamma/h\rceil = \lceil m_1/h\rceil + 1 = m_1/h + 1$ and $h = 2^{\alpha}$ (by Theorem 8) in the expression for g and set $dg/d\alpha = 0$.

Note that although Theorem 10 gives a recursive solution it is very easy to solve by successive approximation by first approximating the value of h using the result of Theorem 9 and then making a correction in the solution. In our experience, the solution never took more than two steps of approximations. It is interesting to note that the solution is almost independent of m and depends only on s, i.e. the ratio (area of SR cell/area of a PLA cell).

Table 3 gives values of α for different values of m and s. It is evident from this table that BM-PLA are only likely to be optimal for small PLAs and for large s. In other conditions partitioning is likely to provide lower overhead. We don't know the ratio s for the layout of PLA used in 7, but we conjecture that s was approximately 10 and therefore, instead of one bit multiplexer, use of two bit multiplexer would have resulted into a PLA with still lower overhead.

Table 3 Optimal Value of α for Different m and s

s \ m	16	20	30	32	40	50	60	64	70	80	90	100	128
10	2	2	2	2	2	2	2	2	2	2	2	2	2
20	4	2/3	3	3	3	3	3	3	3	3	3	3	3
30	4	3	3/4	3	3	3	3/4	3	3	3	3/4	3	3

6. ASYMPTOTIC OVERHEAD

In this section we derive simplified expression of overhead for large PLAs. The total area of an nMOS PLA is given by area of AND/OR planes and area of input inverters, pull-ups, etc. For derivation of simple expression we shall consider only the area of AND/OR planes, i.e. PLA area = $m(2n+\ell).w_1$. In D2-PLA, the area of extra logic $g_{min} = 2m(1+\alpha)w_1 + \lceil m/2^{\alpha}\rceil.w_2$. If we simplify $\lceil m/2^{\alpha}\rceil$ as $m/2^{\alpha}$ then extra area $g = m[2(1+\alpha) + s/2^{\alpha}]w_1$. Therefore

$$\%\text{ overhead} = \frac{100m[2(1+\alpha) + s/2^{\alpha}]w_1}{m(2n+\ell)w_1}$$

$$= \frac{100[2(1+\alpha) + s/2^{\alpha}]}{2n+\ell} \qquad (4)$$

From Table 3 we observe that α is 2, 3 and 3 for value of s of 10, 20 and 30 respectively for large m. On substitution of these values of α and s in equation (4) we obtain

$$s = 10 \qquad \%\text{ overhead} = \frac{850}{2n+\ell}$$

$$s = 20 \qquad \%\text{ overhead} = \frac{1050}{2n+\ell}$$

$$s = 30 \qquad \%\text{ overhead} = \frac{1175}{2n+\ell}$$

These values are plotted in Figure 6. Note that the above expressions are on conservative side as the actual area of original PLA will be more than the area of AND/OR planes only. Thus percentage overhead in practice is likely to be less than the above bounds.

7. FURTHER REDUCTION IN OVERHEAD

In the previous Sections we have described a way of reducing the overhead through the use of DPAA. However, the basic reason a DPAA was appended to each block of Figure 4 was that each element of the distance matrix for each block should be no less than 2. We now propose an alternate design and discuss some of its properties.

Design 3: Product lines of a PLA are divided into k blocks. For a Block i number of extra inputs, e_i, and cross points are determined by using the algorithm given by Bozorgui-Nesbat and McCluskey[9]. Now each block of the PLA has desired distance property, the extra inputs are connected together along with an SR of length k, as shown in Figure 7. We call this design a D3-PLA. The total number of extra inputs to D3-PLA are $e = \max\{e_1, e_2, \ldots, e_k\}$.

D3-PLA can be tested in the same manner as D2-PLA. We therefore can state the following theorem based on Theorems 6 and 7.

Theorem 11: The D3-PLA of Figure 7 can be tested for all faults by a test set of length $m(2+n+e)$.

To show as to how D3-PLA can result into an overhead lower than D2-PLA as well as BM-PLA let us consider CERBERUS PLA[9]. It is a 18 × 50 × 37 PLA.

It is evident that D3-PLA can be no worse than D2-PLA because if we find that the number of extra inputs for any block are larger than $1+\lceil \log h \rceil$, we can realize such a block by appending DPAA.

To show that we can actually improve on BM-PLA we note the following fact.

Fact 1: $e \le$ number of extra inputs for BM-PLA.

Now let us say we divide the 50 lines of CERBERUS PLA into 4 blocks for realization of D3-PLA. It can be expected that the number of extra inputs will reduce by at least 1, i.e. from 4 for BM-PLA to 3 in D3-PLA. Whereas, an SR of length 4 will be added.

Thus total change in the area from BM-PLA is

$$K = -2 \times 50 \times w_1 + 4w_2$$

$$= -100w_1 + 4w_2$$

This change is negative as long as $s < 25$. In general for large PLAs we can expect a larger reduction in the number of extra inputs while changing a BM-PLA into D3-PLA.

Similar arguments hold for other larger examples in *9*. An additional benefit in adopting D3-PLA over BM-PLA is reduced computational complexity which is stated in the form of the following lemma.

Lemma 5: Computational complexity to generate D3-PLA is $O(m_3/k^2)$ as opposed to $O(m^3)$ for BM-PLA.

8. CONCLUSION

In this paper we have proposed three testable designs of PLAs. These designs can be seen as methodologies to improve over the existing designs. We have set different goals at different stages of the design. While moving from D1-PLA to D2-PLA we retained all the properties of D1-PLAs yet reduced the overhead. While introducing D3-PLA, again we were able to reduce the overhead, complexity of generation of PLA, still maintaining the fault coverage. It is simple to incorporate many other variations of these designs. For example, it is possible to design PLAs which are testable by a universal test set and use the partitioning and distance concept to reduce overhead. Such a PLA is discussed in *15*.

The methods proposed in this paper are straightforward to incorporate in Design Automation systems. In this paper we have also described how a number of designs can be merged to give rise to a design superior than the all known designs in the case of PLAs. These concepts can be included in expert systems and may possibly result into still improved designs and design methodologies.

REFERENCES

[1] H. Fujiwara and K. Kinoshita, "A design of programmable logic arrays with universal tests" *IEEE Trans. Comput.*, vol. C-30, no. 11, pp.823-828; and *IEEE Trans. Circuits Syst.*, vol. CAS-28, no. 11, pp.1027-1032, Nov. 1981.

[2] S. J. Hong and D. L. Ostapko, "FITPLA: A programmable logic array for function-independent testing", in *Dig. 10th Int. Symp. on Fault-Tolerant Computing*, pp.131-136, Oct. 1980.

[3] S. Yajima and T. Aramaki, "Autonomously testable programmable logic array", in *Dig. 11th Int. Symp. on Fault-Tolerant Computing*, pp.41-43, June 1981.

[4] K. K. Saluja, K. Kinoshita and H. Fujiwara, "An easily testable design of programmable logic arrays for multiple faults", *IEEE Trans. Comput.*, vol. C-32, no. 11, pp.1038-1046, Nov. 1983.

[5] J. Khakbaz, "A testable PLA design with low overhead and high fault coverage", *IEEE Trans. Comput.*, vol. C-33, no. 8, pp.743-745, Aug. 1984.

[6] H. Fujiwara, "A new PLA design for universal testability", *IEEE Trans. Comput.*, vol. C-33, no. 8, pp.745-750, Aug. 1984.

[7] K. A. Hua, J. Y. Jou and J. A. Abraham, "Built-in tests for VLSI finite-state machines", *Proc. 14th Int. Symp. on Fault-Tolerant Computing*, pp.292-297, June 1984.

[8] T. Sato and Y. Tohma, "A new configuration of PLA with function independent test", Technical Report, Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan, Oct. 1982.

[9] S. Bozorgui-Nesbat and E. J. McCluskey, "Lower overhead design for testability of programmable logic arrays", *Proc. 1984 Int. Test Conf.*, pp.856-865, Oct. 1984.

[10] Y. Min, "A PLA design for ease of test generation", *Proc. 14th Int. Symp. on Fault-Tolerant Computing*, pp.436-442, June 1984.

[11] K. K. Saluja, K. Kinoshita and C. Boswell, "A design of parallel testable programmable logic arrays", *Proc. Int. Symp. on Circuits and Systems*, Japan, June 1985.

[12] G. D. Hachtel, A. R. Newton and A. L. Sangiovanny-Vincentelli, "An algorithm for Optimal PLA folding", *IEEE Trans. CAD of Integrated Circuits and Systems*, vol. CAD-1, no. 2, pp.63-77, April 1982.

[13] R. H. Mak, "Optimization of programmable logic arrays", *Integration, the VLSI Journal*, vol. 2, no. 2, pp.149-162, June 1984.

[14] C. Boswell, "A comparative study of testable design of programmable logic arrays", M.E. Thesis (under preparation), University of Newcastle, N.S.W., Australia.

[15] K. K. Saluja and J. S. Upadhyaya, "Divide and conquer strategy for testable design of programmable logic arrays", *Dig. 4th Australian Microelectronics Conference*, Sydney, pp.121-127, May 1985.

[16] C. Mead and L. Conway, *Introduction to VLSI Systems*, Reading, MA, Addison-Wesley, 1980.

[17] W. Daehn and J. Mucha, "A hardware approach to self-testing of large programmable logic arrays" *IEEE Trans. Comp.*, vol. C-30, no. 11, pp.829-833, Nov. 1981.

[18] S. Z. Hassan and E. J. McCluskey, "Testing PLAs using multiple parallel signature analysers", *Proc. 13th Int. Symp. on Fault-Tolerant Computing*, pp.422-425, June 1983.

[19] .Treuer, H. Fujiwara and V. Agarwal, "A low overhead, high coverage, built-in self-test PLA design", *Proc. 15th Int. Symp. on Fault Tolerant Compiling*, pp. 112-117, June 1985.

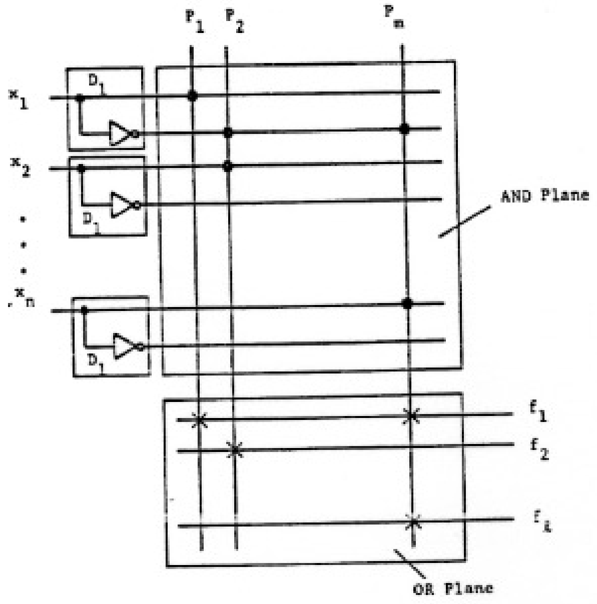


Figure 1 A General PLA Structure

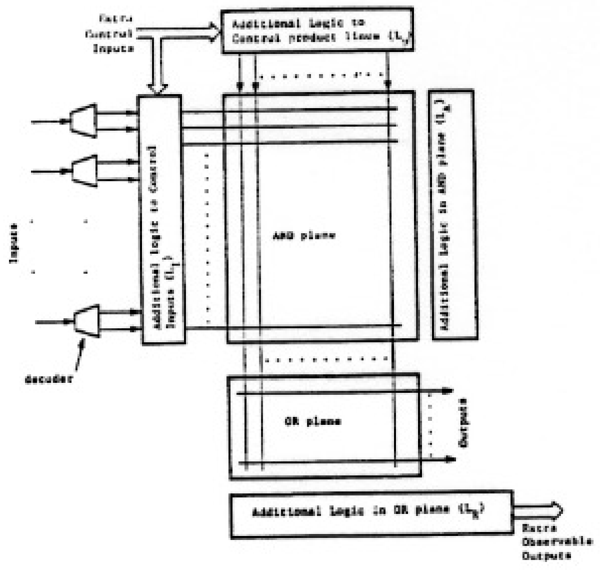
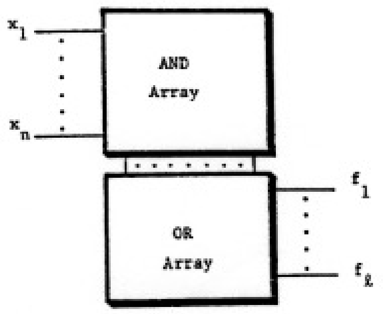
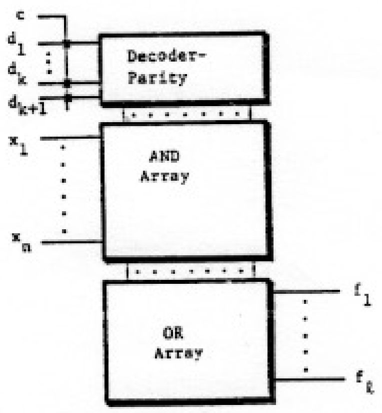

Extra Control Inputs
Additional Logic to Control product lines (L_p)
Inputs
Additional logic to Control Inputs (L_I)
AND plane
Additional Logic in AND plane (L_A)
decoder
OR plane
Outputs
Additional Logic in OR plane (L_R)
Extra Observable Outputs

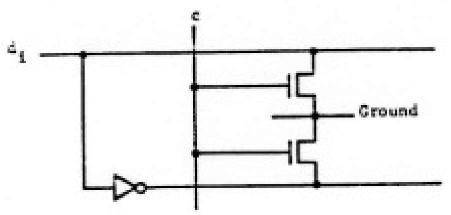
Figure 2 A General Structure of Testable PLAs



(a) Original PLA



(b) Augmented PLA



(c) Extra transistors on inputs of DPAA in case of nMOS realization

Figure 3 Testable Design of a PLA (D1-PLA)

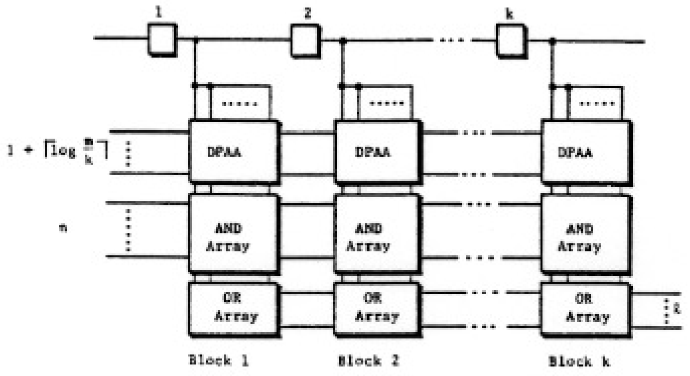


Figure 4 Partitioning of a PLA (D2-PLA)

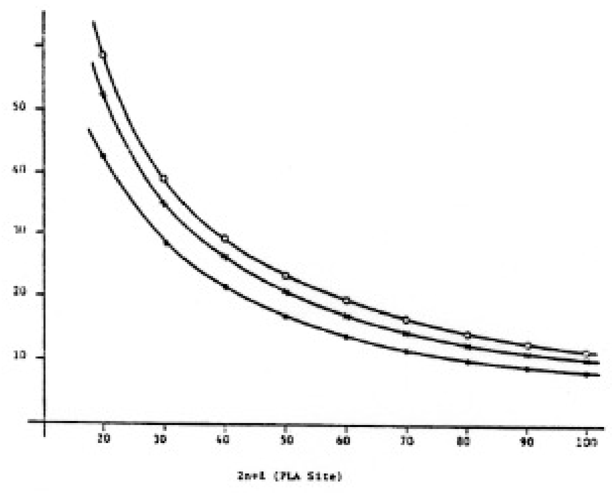


Figure 6 Percentage Overhead for D2-PLAs

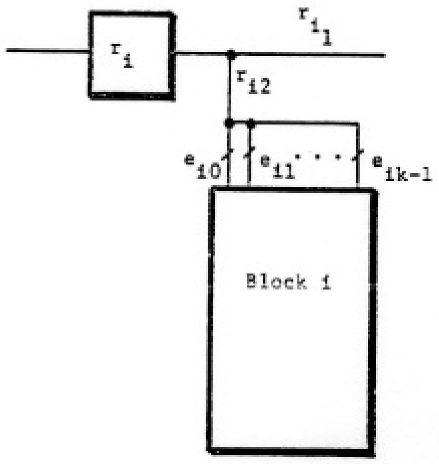


Figure 5 Certain Fault Sites in Block i

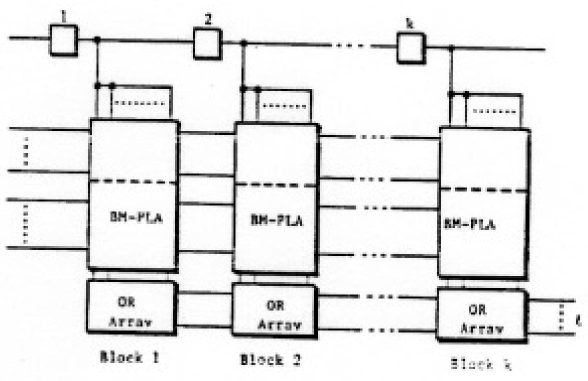


Figure 7 A D3-PLA