Design of Pseudoexhaustive Testable PLA with Low Overhead

Wen-Zen Shen, Gwo-Haur Hwang, Wen-Jun Hsu, and Yun-Jung Jan

Abstract—**The pseudoexhaustive testing (PET) scheme is a economic approach to test a large embedded programmable logic array (PLA). In this paper, we propose an efficient algorithm named low overhead PET (LOPET) to partition the product lines. By applying our algorithm, both the area overhead and test length are reduced significantly.**

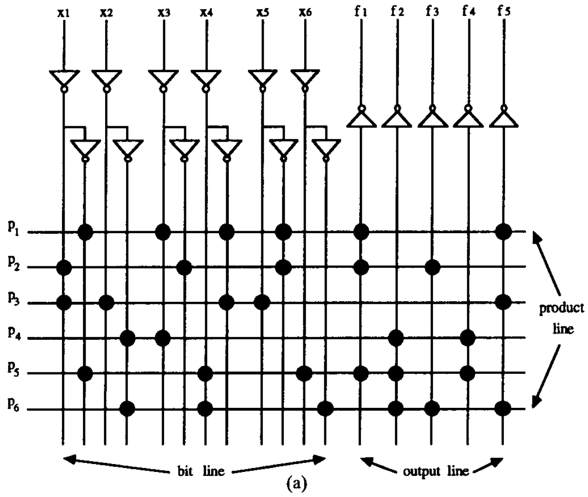
Index Terms—**Built-in self-test, design for testability, programmable logic array, pseudoexhaustive testing.**

I. Introduction

Due to its simplicity, regularity, and flexibility, the programmable logic array (PLA) is used extensively in the design of VLSI circuits and systems. The simplest way to test a PLA is exhaustive testing. But for large PLA's, exhaustive testing is impractical. Thus, there are various PLA test pattern generation algorithms [1]–[4] and PLA testable designs [5]–[9] have been proposed. As VLSI circuits become more dense and complex, PLA's are usually embedded in the chip. This results in poor controllability and observability. Thus, the traditional test pattern generation algorithms or testable designs cannot be applied in such conditions. One effective way to solve the above problem is to design a built-in self-test (BIST) PLA. There are many BIST PLA designs [10]–[16]. Among them, the pseudoexhaustive testable (PET) PLA [10] is claimed to have lower area overhead and fewer test patterns than the others.

The earlier designs, such as BIST [16], must scan both input and product lines, so every input and product line needs a shift register cell. Since a shift register cell is wider than a product line, there exists a pitch mismatch problem. Although the mismatch problem can be solved by multiplexing, the area overhead is still too high. The PET PLA, on the other hand, partitions both input and product lines into groups. While testing, a selected group of input and product lines are tested exhaustively. Since every group needs only one shift register cell, the area overhead of PET is less than earlier designs.

The most important step of PET is the partitioning of product lines. However, the result is not satisfactory. For example, the PLA [in1] (refer to Table II) which has 106 product lines is partitioned into 103 groups. The root of the problem is that the partitioning rule only considers the OR plane conditions, resulting in a rule that is too harsh. In this paper, we propose a new grouping method called low overhead PET (LOPET) PLA, which loosens the OR plane condition while making some restriction on the AND plane. LOPET results in larger but fewer groups than PET. Since the area overhead and number of test patterns are proportional to the group number, LOPET has a lower area overhead and fewer test patterns than PET, while keeps other performance unchanged. In the next section, we will present the preliminaries. Section III is our proposal. In section IV, the partitioning algorithms and experimental results on 30 PLA's are given. Section V is the conclusions.



$$\begin{array}{cccccc} 0&2&1&0&0&2\\ 1&2&0&2&0&2\\ 1&1&2&0&1&2\\ 2&0&1&2&2&2\\ 0&2&2&1&2&1\\ 2&0&2&1&2&0 \end{array} \quad \begin{array}{ccccc} 4&3&3&3&4\\ 4&3&4&3&3\\ 3&3&3&3&4\\ 3&4&3&4&3\\ 4&4&3&4&3\\ 3&4&4&3&4 \end{array}$$

(b)

Fig. 1. (a) A six-input, five-output NOR-NOR PLA. (b) Characteristic matrix of (a).

II. Preliminaries

There are various types of IC chip fabrication technologies. Our proposal can be applied to all of them. In this paper, all the examples are assumed to be fabricated by nMOS technology. Fig. 1(a) shows a six-input five-output NOR-NOR PLA. The fault models we consider in this paper are: 1) crosspoint faults, 2) stuck-at faults, and 3) bridging faults. A crosspoint is the intersection of a bit line or an output line with a product line. There may or may not exist a transistor, called a device, at a crosspoint. In order to simplify our discussion, we have to define the "characteristic matrix" of a PLA. Assume a PLA with n input lines, p product lines, and m output lines.

Definition—Characteristic Matrix of a PLA: The characteristic matrix of a PLA is a p-by-$(n+m)$ matrix denoted by M whose entries are defined as follows: $M(i,j)=0$ if a device exists on the intersection of the ith product line and the true bit line of the jth input line. $M(i,j)=1$ if a device exists on the intersection of the ith product line and the complement bit line of the jth input line. $M(i,j)=2$ if no device is on the intersection of the ith product line with either the true or complement bit line of the jth input line. $M(i,j)=3$ if no device exists on the intersection of the ith product line and the $(j-n)$th output line. $M(i,j)=4$ if a device exists on the intersection of the ith product line and the $(j-n)$th output line.

Fig. 1(b) shows the characteristic matrix of Fig. 1(a). The notation of the characteristic matrix can also be applied to a group of product lines. Fig. 2 is the PET form of Fig. 1(a). In normal operation, all

Manuscript received June 18, 1991; revised March 12, 1992. This work was supported by National Science Council of the Republic of China under Grant NSC80-0404-E009-35.

W.-Z. Shen, G.-H. Hwang, and W.-J. Hsu are with the Institute of Electronics, National Chiao Tung University, Hsin-Chu, Taiwan, Republic of China.

Y.-J. Jan is with the Computer and Communication Laboratory of ITRI, Hsin-Chu, Taiwan, Republic of China.

IEEE Log Number 9207294.

0018–9340/93$03.00 © 1993 IEEE

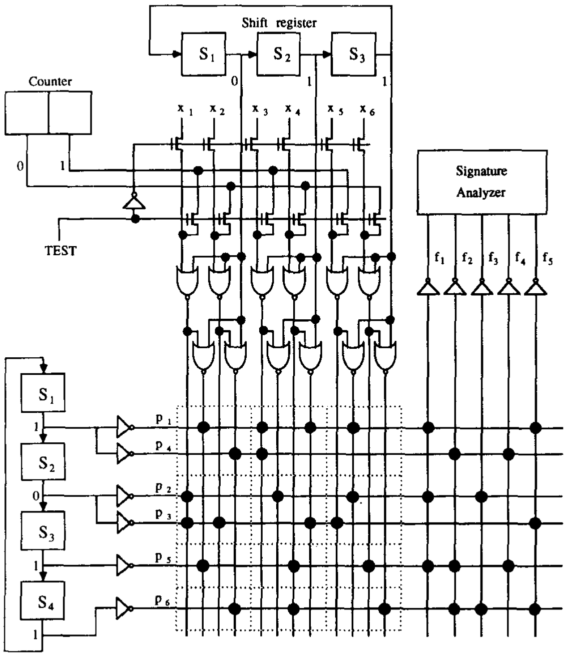


Fig. 2. PET form of Fig. 1(a).

the cells of the shift registers and the TEST signal are set to 0. During testing, the control input called TEST is set to 1 to disconnect the PET PLA from other circuits on the chip. The counter starts counting to apply test patterns. The function of the signature analyzer is to compress test data which may result in some loss of fault coverage. The main spirit of PET is the adding of two shift registers associated with the inputs and the product lines. These two shift registers logically partition the AND plane of a PLA into blocks (see dotted lines in Fig. 2). For each register, only one cell is at 0 and the rest of them are at 1. Thus, at each time only one block is activated, and the counter applies exhaustive patterns to the block. In order to make the test patterns minimal, the input group size (counter length) is fixed to 2. So, four possible input combinations are applied to the activated block. Since the rest of the input shift register cells are set to 1, the corresponding bit lines are all set to 0. Due to the NOR gate feature of the nMOS, all the crosspoint faults on the block can be sensitized and propagated through the AND plane. The above process would scan through all the blocks via the control of the two shift registers.

In order to propagate the AND plane crosspoint faults through the OR plane and to detect the OR plane crosspoint faults, some rules must be applied to partition the product lines into groups. If we use the characteristic matrix to describe a group, the partitioning rule can be stated as follows:

In a group, every column of the OR plane can have at most one "4."

According to [1], the above condition will guarantee to propagate all the single detectable crosspoint faults through the OR plane. In fact, all single crosspoint faults, all single stuck-at faults on bit lines, product lines, and output lines, and all single bridging faults between bit lines, product lines, and output lines are detected in a PET PLA. The proofs can be found in [17].

Definition—Mono-device: In a group, if a column has only one "4," the "4" is called a "mono-device."

According to the above definition, we can find that every OR plane device in PET must be a mono-device. Such a restriction is very harsh which results in small group size and large group number.

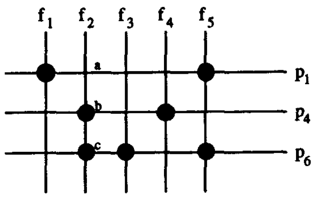


Fig. 3. The OR plane of the first group in Fig. 1.

III. Our Proposal

In order to reduce the group number, we have to loosen the OR plane condition. The new rule is as follow:

Rule 1: In a group, every product line must have at least one mono-device.

According to [1], all the crosspoint faults on the AND plane part of a product line will be propagated via the mono-device of this product line. Therefore, we just have to consider the OR plane faults. Unlike PET, our proposal permits OR plane to be overlapped, which means some columns may have more than one "4" in the group. When the overlap occurs, some OR plane crosspoint faults may be masked and turn out to be untestable. As shown in Fig. 3, f_2 and f_5 are overlapped. In order to detect all the crosspoint faults on the OR plane, we must make some restriction on the AND plane.

Definition—Sharp Operator #: The sharp operator # (refer to [1]) is used to find the vertices in one cube (or list of cubes), but not in a second cube (or a list of cubes).

For example, given two lists of cubes A and B, if A # B = C, then C consists of all the vertices which are contained in A, but not in B. In this paper, the sharp operation between product terms is defined to take the sharp operation on every input group separately. When the results of # operation are null sets for every input group, we use a notation ϕ to represent such a condition. We shall use the following example to show how the restriction on the AND plane is made.

Fig. 3 is the OR plane part of the first group in Fig. 2. Originally there are only p_1 and p_4. Since it can meet the PET condition, there is no problem. When p_6 is added, overlap occurs. We can find that only the crosspoint faults on the overlapping columns may be masked. Let's take a look at f_2. If we want to detect "b," there must be some patterns to set p_4 on but p_6 off. In other words, $p_4 \# p_6 \not\equiv \phi$. In the same way, we can find: to detect "c" $\rightarrow p_6 \# p_4 \not\equiv \phi$. To detect "a," there must be some patterns to set p_1 on but p_4 and p_6 off. In other words, it must be $p_1 \# (p_4 \cup p_6) \not\equiv \phi$. The above process must be performed for f_5 in the same manner. If all the above conditions are met, we can include p_6 into the group. Now, we can define the AND plane conditions as follows:

Rule 2: In a group, for every column which has more than one "4," the following conditions must be met:

1) Every product line that has "4" on this column must sharp to the union of the other product lines that also have "4" on this column, and the result cannot be ϕ.
2) Every product line that has "3" on this column must sharp to the union of all the product lines that have "4" on this column, and the result cannot be ϕ.

If both Rule 1 and Rule 2 are met, the group is a LOPET group.

Example 1: Fig. 4 shows the LOPET form of Fig. 1(a). Let's check its correctness: for Group 1, the mono-device of p_1, p_4, p_6 is on f_1, f_4, f_3 respectively. Therefore, the Rule 1 is met. From the above discussion, we have:

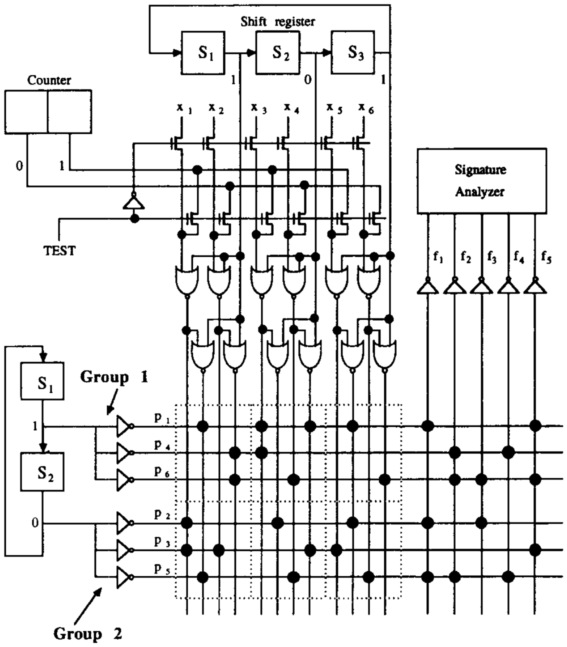


Fig. 4. LOPET form of the PLA in Fig. 1.

$$\begin{aligned}
&p_1 = 02 : 10 : 02\\
&p_4 = 20 : 12 : 22\\
&p_6 = 20 : 21 : 20\\
\text{for } f_2\text{:}\quad &p_4 \# p_6 = \phi : 10 : 21 \not\equiv \phi\\
&p_6 \# p_4 = \phi : 01 : \phi \not\equiv \phi\\
&p_1 \# (p_4 \text{ u } p_6) = 01 : \phi : \phi \not\equiv \phi\\
\text{for } f_5\text{:}\quad &p_1 \# p_6 = 01 : 10 : 01 \not\equiv \phi\\
&p_6 \# p_1 = 10 : 21 : 10 \not\equiv \phi\\
&p_4 \# (p_1 \cup p_6) = \phi : \phi : 11 \not\equiv \phi
\end{aligned}$$

$\therefore$ Group 1 meets the LOPET rules.

In the same manner, the correctness of Group 2 can also be proved. Thus, in this example, LOPET reduces the group number from 4 to 2. For a large PLA, the effect of LOPET is significant, as shown in the next section. Now, let's discuss the fault coverage of the LOPET.

Lemma 1: All the crosspoint faults on the AND plane can be detected in a LOPET PLA.

Proof: Assume the faulty crosspoint c_{ij} is the intersection of input x_i and product line p_j. The key point to detect c_{ij} is: except for the device on c_{ij}, all the devices on the AND plane part of p_j must not be activated. When the block containing c_{ij} is activated, all the bit lines of the inactivated input groups are set to 0. No devices on these inputs are activated. Since the test patterns for the activated block are exhaustive, we can set all the inputs of the block except c_{ij} to some values which don't activate any devices on the intersection of p_j and these inputs. Now, for c_{ij}, we can test the 2→1 or 1→2 fault by setting x_i to 0, and 2→0 or 0→2 fault can be tested by setting x_i to 1. The above condition will cause the p_j to be improperly on or off, and the outputs corresponding to the mono-devices of p_j will also be improperly on or off. So, c_{ij} is detected. Q.E.D.

Lemma 2: All the single crosspoint faults on the OR plane can be detected in a LOPET PLA.

Proof: According to the discussion of the Rule 2, we can conclude that the Rule 2 can guarantee to detect all the crosspoint faults on the OR plane. Q.E.D.

Lemma 3: All the bridging and the stuck-at faults of bit lines and product lines can be detected in a LOPET PLA.

Proof: We know that all the bridging faults and stuck-at faults of bit lines and product lines must be propagated by causing the corresponding product lines to be improperly on or off. Every time a product line in PET is improperly on or off, all the mono-devices of this product line will output the error signals. Since every product line in LOPET has at least one mono-devices, we can conclude that if PET can detect the fault stated in this lemma, then LOPET can detect them as well. Therefore, the proof of this lemma is the same as the PET condition which is given in [17]. Q.E.D.

Lemma 4: All the single bridging faults between adjacent output lines can be detected in a LOPET PLA.

Proof: Assume two adjacent output lines are bridging. There must exist at least one product line which has crosspoint "34" or "43" on these two outputs. The test patterns which detect the above "3" will set these two outputs to be 0,1 or 1,0. Thus, the bridging can be detected. Q.E.D.

Lemma 5: All the single output line stuck-at 0 faults can be detected in a LOPET PLA.

Proof: For any "4" on the output line, the test patterns which detect this "4" will set the output to 1. Thus, the output stuck-at 0 is detected. Q.E.D.

Lemma 6: If there is at least one "3" on an output line, this output line stuck-at 1 can be detected in a LOPET PLA.

Proof: For any "3" on this output line, the test patterns which detect this "3" will set the output to 0. Thus, the output stuck-at 1 is detected. Q.E.D.

The only kind of faults that LOPET can't detect is: when every crosspoint of an output line has a device, the output line stuck-at 1 may not be detected. Such a condition is unlikely to happen. If it happens, LOPET can detect such a condition. We just need to arbitrarily choose one group, and arbitrarily partition it into two groups. Then, the problem is solved. Assume we partition group X into group Y and group Z. According to Rule 2, there must be at least one pattern which turns on one product line but turns off all the other product lines in group X. Assume the above mentioned product line, which is uniquely on, is in group Z. Then, the patterns which make the product line in group Z uniquely on will turn off all the product lines in group Y. Therefore, we can detect the originally undetectable fault by these patterns. The above method only needs to add a shift register cell. While for PET, such a condition is a nightmare. Since the group number is equal to the product number.

From Lemmas 1–6 and the above discussion, we have the following theorem.

Theorem 1: All single crosspoint faults, all single stuck-at faults on bit lines, product lines, and output lines, and all single bridging faults between bit lines, product lines, and output lines are detected in a LOPET PLA.

From Theorem 1, we can find that LOPET has the same fault coverage as PET.

IV. Experimental Results

Now, we want to see how good LOPET can be improved from PET. The problem can be stated as follows:

"Given a PLA, we want to find a partitioning algorithm that can result in minimum number of groups, each of which meets both Rule 1 and Rule 2 of LOPET."

At first, we used a simple heuristic algorithm called SCAN to partition the product lines. SCAN is similar to the partitioning algorithm used in [10], the only difference is that the grouping condition is changed from PET to LOPET. We have applied SCAN to 30 PLA's [18] which all have at least 15 inputs. These PLA's

TABLE I
THE RESULTS OF VARIOUS LOPET ALGORITHMS

Name	SCAN	REST	REVERSE	REVERSE1	REVERSE2	Rule 1
bc0	72	69	*68	68	68	68
bca	55	51	*50	50	50	48
bcb	45	46	43	*42	42	40
bcc	42	40	40	40	*39	38
bcd	48	46	*44	44	44	44
chkn	73	73	*73	73	73	73
cps	25	24	*21	22	23	15
dk48	3	3	*3	3	3	3
exep	35	35	*35	35	35	35
gary	27	25	22	*21	22	20
in0	26	25	22	*21	22	20
in1	32	33	32	31	*30	29
in2	34	32	30	30	*29	29
in3	17	17	*15	15	18	13
in4	77	74	68	68	*66	65
in5	17	16	*15	15	15	14
in6	9	9	*8	8	8	7
in7	16	12	*12	12	12	12
jbp	11	10	*10	10	10	9
misg	33	33	*33	33	33	33
mish	5	5	*5	5	5	5
opa	14	14	*13	13	13	8
rckl	14	16	*14	14	14	11
ti	31	25	24	*23	25	20
vg2	40	40	*40	40	40	40
x1dn	40	40	*40	40	40	40
x2dn	10	10	*10	10	10	10
x6dn	35	36	*35	35	35	35
x7dn	63	63	*63	63	63	63
x9dn	40	40	*40	40	40	40
Total	**989**	**962**	**928**	**924**	**925**	**925**

(total * = 918)

TABLE II
THE COMPARISON BETWEEN LOPET AND PET

Name	#in	#out	p.[10]	p.ESP	g.PET	g.LOPET	t.PET	t.LOPET
bc0	26	11	179	179	121	68	6292	3536
bca	26	46	180	180	98	50	5096	2600
bcb	26	39	156	155	106	42	5512	2184
bcc	26	45	137	.137	99	39	5148	2028
bcd	26	38	117	117	81	44	4212	2288
chkn	29	7	140	140	73	73	4234	4234
cps	24	109	162	163	63	21	3024	1008
dk48	15	17	21	22	3	3	90	90
exep	30	63	109	110	33	35	1980	2100
gary	15	11	107	107	38	21	1140	630
in0	15	11	107	107	39	21	1170	630
in1	16	17	104	106	103	30	3296	960
in2	19	10	135	136	48	29	1824	1102
in3	35	29	74	74	34	15	2380	910
in4	32	20	212	212	96	66	6144	4224
in5	24	14	62	62	39	15	1872	720
in6	33	23	54	54	10	8	660	528
in7	26	10	54	54	24	12	1248	624
jbp	36	57	122	122	15	10	1080	720
misg	56	23	69	69	34	33	3808	3696
mish	94	43	82	82	5	5	940	940
opa	17	69	79	79	44	13	1496	442
rckl	32	7	32	32	19	14	1216	896
ti	47	72	213	213	53	23	4982	2162
vg2	25	8	110	110	40	40	2000	2000
x1dn	27	6	110	110	40	40	2160	2160
x2dn	82	56	104	104	10	10	1640	1640
x6dn	39	5	81	82	39	35	3042	2730
x7dn	66	15	538	538	63	63	8316	8316
x9dn	27	7	120	120	40	40	2160	2160
Total			**3770**	**3776**	**1510**	**918**	**88162**	**58398**

Name : name of PLA
#in : number of PLA inputs
#out : number of PLA outputs
p.[10] : number of product lines listed in [10]
p.ESP : number of product lines by ESPRESSO-MV
g.PET : number of PET groups listed in [10]
g.LOPET: number of LOPET groups
t.PET : number of test patterns of PET
t.LOPET : number of test patterns of LOPET

were also used in [10]. There are various PLA logic minimization algorithms which may result in different outcomes. In this paper, we used ESPRESSO.MV [19]. The little difference of the results between ESPRESSO.MV and the algorithm used in [10] is negligible.

As shown in Table I, total groups for the SCAN are 989, which are much less than PET. This result has proven the effectiveness for the loosen of the OR plane condition. With an attempt for further reduction of the group number, we use the results of ESPRESSO.MV to run a program called REST [20]. REST is a PLA restructuring algorithm which can increase the distance between product lines while keeps the logic function of the PLA unchanged. Since the distance is increased, the probability for the LOPET sharp operation to be ϕ is reduced. We use the results of REST to run SCAN. As shown in Table I, the results have only 962 groups. Since PET has no concern with the personality of the AND plane, REST has no effect on PET.

Extensively investigating the results, we found another weak point of the SCAN. Since the SCAN merely scans through the PLA to choose product lines for grouping, there may exist a problem: "Those product lines, which are easy to be grouped with others, may be grouped at first, while leave the 'hard' product lines to stand alone." To solve the above problem, we propose three new algorithms: REVERSE, REVERSE1, and REVERSE2 [21]. For every benchmark PLA, we mark the best result among REVERSE, REVERSE1, and REVERSE2 with a "*" in Table I. The total group number of the results marked with "*" in Table I is 918.

Finally, if we only consider Rule 1 while omit Rule 2 in all the steps of REVERSE1, total groups are 887. In Table I, the results are under the column denoted by "Rule 1." Such results do not meet LOPET conditions, but they can tell us the degree of restriction imposed by Rule 2. Since 918−887=31, it means that only 31 more groups are needed if we take both Rule 1 and Rule 2 into consideration. Under the above observation, we can conclude that the restriction imposed by Rule 2 are not significant. At last, we can conclude that, with all the other conditions unchanged, the total group number is reduced from 1510 to 918. The results of comparison between LOPET and PET are listed in Table II. The reduction ratio is (1510−918) / 1510 = 39.2%. Since the shift register cells occupy the majority of the area overhead, we can calculate the shift register reduction ratio to estimate the total area overhead reduction ratio. Total shift register cells for PET are 2072, while for LOPET are 1480. The reduction ratio is 28.6%. The total test patterns are reduced from 88 162 to 58 398. The reduction ratio is 33.8%.

In order to compare with the PET results listed in [10], we only consider the input group size of 2 in the above discussions. Now, we want to discuss the effects of increasing the input group size.

Since the PET product line grouping rule has no concern with the AND plane conditions, the product line group number does not change when the input group size is increased. However, for LOPET, if the input group size is increased from 2 to 4, the possible input combinations for each input group are increased from 4 to 16. Therefore, the possibility for the result of sharp operation to be ϕ is reduced. Such effects are similar to that of REST.

Table III is the results of REVERSE1 algorithm with input group size from 3 to 10. Comparing with the results of only considering Rule 1, we can find that the restriction of Rule 2 has almost vanished. Again, we mark the best results of each PLA with "*," and find that the total group number is 890.

Of course, there are some disadvantages when the input group size is increased. First of all, we can find that the number of test patterns is increased. But for a BIST PLA, test pattern number under 1M is acceptable. Moreover, when the length of the counter in Fig. 4 is increased, it may also induce some routing problems.

TABLE III
THE PRODUCT LINE GROUPS FOR VARIOUS INPUT GROUP SIZE

Name	3	4	5	6	7	8	9	10	Rule 1
bc0	*68	68	68	68	68	68	68	68	68
bca	*48	48	48	48	48	48	48	48	48
bcb	*40	40	40	40	40	40	40	40	40
bcc	*38	38	38	38	38	38	38	38	38
bcd	*44	44	44	44	44	44	44	44	44
chkn	*73	73	73	73	73	73	73	73	73
cps	*15	15	15	15	15	15	15	15	15
dk48	*3	3	3	3	3	3	3	3	3
exep	*35	35	35	35	35	35	35	35	35
gary	*20	20	21	21	21	20	20	21	20
in0	*20	20	21	21	21	20	20	20	20
in1	*29	29	29	29	29	29	29	29	29
in2	*29	29	29	29	29	29	29	29	29
in3	16	*14	14	14	14	14	14	14	13
in4	*65	65	65	65	65	65	65	65	65
in5	*14	14	14	14	14	14	14	14	14
in6	8	8	*7	7	7	8	8	7	7
in7	*12	12	12	12	12	12	12	12	12
jbp	*10	10	10	10	10	10	10	10	10
misg	*33	33	33	33	33	33	33	33	33
mish	*5	5	5	5	5	5	5	5	5
opa	10	10	9	9	9	10	9	*8	8
rckl	13	13	13	*12	13	13	12	12	12
ti	*20	20	20	20	20	20	20	20	20
vg2	*40	40	40	40	40	40	40	40	40
x1dn	*40	40	40	40	40	40	40	40	40
x2dn	*10	10	10	10	10	10	10	10	10
x6dn	*35	35	35	35	35	35	35	35	35
x7dn	*63	63	63	63	63	63	63	63	63
x9dn	*40	40	40	40	40	40	40	40	40
Total	**896**	**894**	**894**	**893**	**894**	**896**	**892**	**892**	**887**

(total * = 890)

By evaluating the various outcomes, a designer can have the flexibility to choose an appropriate input group size.

V. CONCLUSIONS

In this paper, we propose a new algorithm called LOPET to improve the original PET algorithm which was claimed to be the best in this area. The key point for PET is to partition the product lines into groups. The area overhead and test pattern number are proportional to the group number. Both PET and LOPET use the mono-device concept to propagate the faults, but PET only considers the OR plane conditions, which results in small group size and large group number. LOPET, which considers both the AND plane and the OR plane conditions, has fewer groups than PET. We have proven that LOPET can detect all the single crosspoint, bridging, and stuck-at faults which are the same as PET. We also have explicitly developed a series of heuristic algorithms. At first, a simple algorithm called SCAN is proposed. Besides, the PLA restructuring algorithm which can increase the distance between product lines so as to reduce the group number is also introduced. Finally, we propose the REVERSE algorithm for the further reduction of the group number.

For 30 benchmark examples, the LOPET PLA's have less area overhead and test patterns than the PET PLA's, while keeping other performance unchanged. The group number reduction ratio is 39.2%. The shift register cell reduction ratio is 28.6%. The test pattern reduction ratio is 33.8%. When the input group size is increased, the restriction of Rule 2 almost vanishes, and the group number is further reduced. We have listed all the results of various input group sizes.

REFERENCES

[1] J. E. Smith, "Detection of faults in programmable logic arrays," *IEEE Trans. Comput.*, vol. C-28, pp. 845–853, Nov. 1979.

[2] D. L. Ostapko and S. J. Hong, "Fault analysis and test generation for programmable logic arrays (PLA's)," *IEEE Trans. Comput.*, vol. C-28, pp. 617–626, Sept. 1979.

[3] E. B. Eichelberger and E. Lindbloom, "A heuristic test pattern generation for programmable logic arrays," *IBM J. Res. Develop.*, vol. 24, pp. 15–22, Jan. 1980.

[4] R. S. Wei and A. Sangiovanni-Vincentelli, "PLATYPUS: A PLA test pattern generation tool," *IEEE Trans. Comput.-Aided Design*, vol. CAD-5, pp. 633–644, Oct. 1986.

[5] H. Fujiwara, "A new PLA design for universal testability," *IEEE Trans. Comput.*, vol. C-33, pp. 745–750, Aug. 1984.

[6] J. Khakbaz, "A testable PLA design with low overhead and high fault coverage," *IEEE Trans. Comput.*, vol. C-33, pp. 743–745, Aug. 1984.

[7] D. S. Ha and S. M. Reddy, "On the design of testable domino PLA's," in *Proc. Int. Test Conf.*, Nov. 1985, pp. 567–572.

[8] S. Bozorgui-Nesbat and E. J. McCluskey, "Lower overhead design for testability of programmable logic arrays," *IEEE Trans. Comput.*, vol C-35, pp. 397–383, Apr. 1986.

[9] S. M. Reddy and D. S. Ha, "A new approach to the design of testable PLA's," *IEEE Trans. Comput.*, vol. C-36, pp. 201–211, Feb. 1987.

[10] S. M. Reddy and D. S. Ha, "On the design of pseudoexhaustive testable PLA's," *IEEE Trans. Comput.*, vol. 37, pp. 468–472, Apr. 1988.

[11] D. S. Ha and S. M. Reddy, "On BIST PLA's," in *Proc. Int. Test Conf.*, Sept. 1987, pp. 342–351.

[12] K. A. Hua, J. Y. Jou, and J. A. Abraham, "Built-in test for VLSI finite-state machine," in *Proc. 14th Int. Symp. Fault-Tolerant Comput.*, June 1984, pp. 302–307.

[13] D. L. Liu and E. J. McClusky, "Design of large embedded CMOS PLA's for BIST," *IEEE Trans. Comput.-Aided Design*, vol. 7, pp. 50–59, Jan. 1988.

[14] J. Salick, M. R. Mercer, and B. Underwood, "Built-in self-test input generator for programmable logic arrays," in *Proc. Int. Test Conf.*, Nov. 1985, pp. 115–125.

[15] S. J. Upadhyaya and K. K. Saluja, "A new approach to design for BIST PLA's for high fault coverage," *IEEE Trans. Comput.-Aided Design*, vol. 7, pp. 60–67, Jan. 1988.

[16] R. Treuer, H. Fujiwara, and V. K. Argawal, "Implementing a built-in self-test PLA design," *IEEE Design Test Comput.*, pp. 37–48, Apr. 1985.

[17] D. S. Ha, "On the design and built-in self-test of PLA's," Ph.D. dissertation, Dep. Elect. Comput. Eng., Univ. Iowa, Iowa City, May 1987.

[18] R. K. Brayton, G. D. Hachtel, C. T. McMullen, and A. Sangiovanni-Vincentelli, "Logic Minimization Algorithms for VLSI Synthesis." Norwell, MA: Kluwer, 1984.

[19] R. L. Rudell and A. Sangiovanni-Vincentelli, "Multiple-valued minimization for PLA optimization," *IEEE Trans. Comput.-Aided Design*, vol. CAD-6, pp. 727–751, Sept. 1987.

[20] G. H. Hwang and W. Z. Shen, "Restructure algorithm for testable PLA," in *Proc. 1990 IEEE Int. Symp. Circuits Syst.* (New Orleans, LA), May 1–3, 1990, pp. 2740–2743.

[21] W. Z. Shen, G. H. Hwang, W. J. Hsu, and Y. J. Jan, "Design of pseudoexhaustive testable PLA with low overhead," in *Proc. 1991 Int. Symp. VLSI Technol., Syst., Applicat.* (Taipei, Taiwan, R.O.C.), May 22–24, 1991, pp. 409–413.