IEEE Int. Symp. on Circuit and Systems, pp.1301-1304, June 1985.

AN IMPLEMENTATION OF A NEW BUILT-IN, SELF-TEST PLA DESIGN

R. Treuer, H. Fujiwara* and V.K. Agarwal

Dept. Electrical Engineering, McGill University

3480 University St., Montreal, Canada H3A 2A7

Abstract

An nMOS implementation of a new BIST PLA design (from a companion paper [FTA85]) is described. For large PLA's, the additional test circuitry uses less than 20% extra area, which is a significantly better overhead than that of any existing scheme. Both the input test patterns and the output response (which is compressed into a string of parity bits) are independent of the functions that the PLA realizes, and the 20% overhead even includes the storage needed for the fault free compressed output data. Our approach, as proven in [FTA85], covers all single and $(1-2^{-(2n+m)})$ of all multiple stuck, crosspoint and bridging faults in the original PLA and in the added test circuitry (n and m are the numbers of input variables and product terms, respectively).

1. Introduction

Due both to the wide use of Programmable Logic Arrays (PLAs) in VLSI chips, and to their structural regularity, the PLA test problem has attracted much attention recently. Built-in self-test (BIST) PLAs with low overhead and high fault coverage appear to be a possible solution to this testing problem.

In this paper, we assume all PLAs to have n inputs (with single bit decoders), 2n bit lines, m (an even number) product lines, p sum lines and p outputs. The table below compares several schemes:

BIST Scheme	Delay per Test	Output Response	Number of Test Patterns
[DM81]	O(1)	Dependent	O(n+m+p)
[HM83]	O(1)	Dependent	$O(2^n)$
[SKF83]	O(1)	Dependent	O(nm)
[YA81]	O(m)	Independent	O(n+m)
[FK81]	O(m)	Independent	O(n+m)
NEW	O(p)	Independent	O(nm)

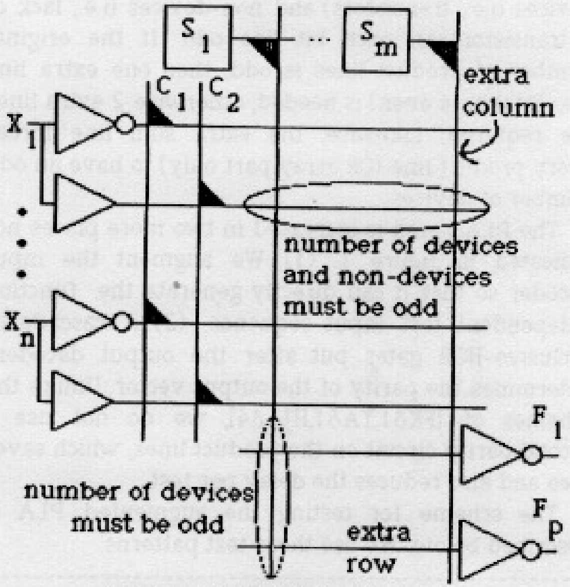

S 1
S m
C 1
C 2
extra column
X 1
X n
number of devices and non-devices must be odd
F 1
F P
number of devices must be odd
extra row

Figure 1: Augmented PLA

The two most important criteria for comparing BIST PLA designs, (1) area overhead and (2) fault coverage, are not in the above table because the NEW scheme has a lower area overhead and a higher fault coverage than each of the other 5 schemes (due to limited space, proof of this claim is not given here). Since minimizing area overhead and maximizing fault coverage are contradictory goals, it is natural to ask how the new scheme obtained improvements in both fault coverage and overhead. The answer: (1) less extra area was needed because a long test sequence was used and the compressed output is function independent, and (2) higher fault coverage was gained by exploiting all of the parity information.

* Dr. H. Fujiwara was a Visiting Professor while this research was carried out. He is with the Dept. of Electronic Eng., Osaka University, Japan. This research was supported by a fellowship to R. Treuer, and by a Strategic Grant, both from the Natural Sciences and Engineering Research Council of Canada.

2. New Built-in Self-test Approach

Figure 1 shows a PLA augmented by adding a shift register, two control lines, one product line, and one sum line. The shift register can disable all product lines but one, to allow the effect of a single product line on the outputs to be observed. The 2 control lines C_1 and C_2 can disable all ¬x's and x's, respectively. The extra product line is used to make the number of devices (i.e., transistors) and non-devices (i.e., lack of a transistor) on each bit line odd. If the original number of product lines is odd, then one extra line (i.e.: total m is even) is needed; otherwise 2 extra lines are required. Likewise, the extra sum line forces every product line (OR array part only) to have an odd number of devices.

The PLA's area is increased in two more places not indicated in figure 1. (1) We augment the input decoder so that it can directly generate the function independent test input sequence. (2) A cascade of exclusive-NOR gates put after the output decoders determines the parity of the output vector. Unlike the schemes of [FK81,YA81,HJA84], we do not use a second parity circuit on the product lines, which saves area and also reduces the delay per test.

The scheme for testing the augmented PLA is described below. We use these test patterns:

	x_1	…	x_{i-1}	x_i	x_{i+1}	…	x_n	c_1	c_2	s_1	…	s_{j-1}	s_j	s_{j+1}	…	s_m
I^1	0	…	0	0	0	…	0	1	0	1	…	1	1	1	…	1
I^2_j	0	…	0	0	0	…	0	1	0	1	…	1	0	1	…	1
I^3_j	1	…	1	1	1	…	1	0	1	1	…	1	0	1	…	1
I^4_{ij}	0	…	0	1	0	…	0	1	0	1	…	1	0	1	…	1
I^5_{ij}	1	…	1	0	1	…	1	0	1	1	…	1	0	1	…	1

Theorem When the Universal Test Sequence

$$I^1 \bullet \prod_{j=1}^{m}(I^2_j) \bullet \prod_{i=1}^{n}\prod_{j=1}^{m}(I^4_{ij}) \bullet \prod_{j=1}^{m}(I^3_j) \bullet \prod_{i=1}^{n}\prod_{j=1}^{m}(I^5_{ij})$$

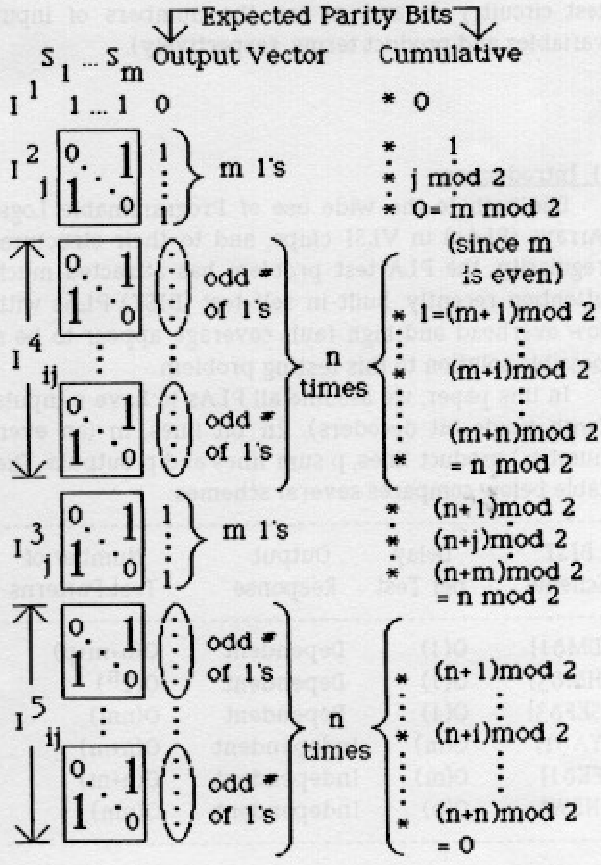
is applied, all single and almost all multiple crosspoint, stuck and bridging faults are detected by the following scheme (fewer than $2^{-(m+2n)}$ of all multiple faults remain undetected; for example, for a large PLA, with m = 180 and n = p = 60, our scheme will cover better than $(1-2^{-300})$ of all multiple faults, including faults in the added test circuitry). Every new output vector is compressed into a single parity bit, and then this bit is exclusive-ORed with a bit representing the cumulative parity of all previous output vectors, to obtain a new cumulative parity bit. The cumulative parity bit is compared to its function independent expected value at (2n+2m+1) specific times, as indicated by the asterisks in figure 2. The expected parity bits of the Output Vectors were derived as follows:

I^1: If the PLA is fault-free, then the output vector is a string of all zeroes, whose parity is zero.

I^2_j: If the PLA is fault-free, then each of the m output vectors has an odd number of 1's because each product line in the OR array has an odd number of devices. Thus, the parity of each output vector is 1. (Similarly for I^3_j.)

I^4_{ij}: Assume the PLA to be fault-free. For simplicity, consider the partial test pattern Π_j (I^4_{ij}), with i being constant, which activates the crosspoints on bit line x_i only. If the crosspoint (x_i,j) has a device, then the

Figure 2: Cumul. Parity Comparison scheme



product line j is pulled down to 0, which produces an output of all zeroes, and thus a parity bit of zero. If the crosspoint (x_i,j) has NO device, then the product line j stays at 1, which produces an output with an odd number of ones, and thus a parity bit of 1. Recall that the number of non-devices of each bit line is odd, therefore an odd number of 1's parity bits are created for each bit line. (Similarly for I^5_{ij}.)

Interestingly, the cumulative parity bit sequence of length (2n+2m+1) is simply an alternating sequence of 0's and 1's., thus allowing the fault free bit sequence to be generated on-line by a simple circuit. The test length for our scheme is (2nm+2m+1) and the delay per test is (p+3).

The complete proof of the above theorem is given in [FTA85] as a series of 12 lemmas and 2 theorems. Due to space limitations, we present abridged proofs of 2 lemmas dealing with crosspoint faults only:

Lemma 1 All single and almost all of the multiple crosspoint faults in the AND array can be detected by the test patterns $\prod_{i=1}^{n} \prod_{j=1}^{m} (I^4_{ij})$ and $\prod_{i=1}^{n} \prod_{j=1}^{m} (I^5_{ij})$

Proof A crosspoint fault at (x_i,j) inverts the expected value of the product line j, and hence the parity of the output is also inverted. I^4_{ij} and I^5_{ij} activate all m crosspoints of a bit line to produce one new cumulative parity bit; thus, if the total number of faults on a bit line is even, the cumulative parity bit indicates no fault since the number of devices and non-devices for the line remains odd. Thus, the only multiple faults that are not detected are those which have an even number of faults (or zero faults) on each bit line. The total number of possible multiple crosspoint faults in the AND array is $2^{(2n)(m)}$ The number of possible even faults (including zero faults) for a single bit line is $2^{(m-1)}$. Thus, the fraction of undetectable faults in the AND array is: $(2^{m-1})^{2n} / 2^{2nm} = 2^{-2n}$

Lemma 2 All single and almost all of the multiple crosspoint faults in the OR array can be detected by applying either $\prod_{j=1}^{m} (I^2_j)$ or $\prod_{j=1}^{m} (I^3_j)$

Proof As in Lemma 1, undetectable multiple faults have an even (or zero) number of faults per product line (OR array part only). The fraction of undetectable faults in the OR array is: $(2^{p-1})^m / 2^{pm} = 2^{-m}$.

3. nMOS Implementation of Augmented PLA

We add the following four small circuits to PLAs:

(1.) Figure 3 shows a shift register cell whose output ("Next Value") is shared by two adjacent product lines by multiplexing. The cell's layout (using [MC80]'s design rules) is 16 lambda wide and 170 lambda long. Multiplexing is used because no more than one shift register cell can occupy the narrow 16 lambda width which 2 product lines share.

(2.) The augmented input decoder, shown in figure 4, uses a shift register cell (composed of 2 NAND gates) for each pair of bit lines. The "Hold" signal (which is the value of the last cell S_m of the product lines' shift register) is needed to hold the value x in each cell, since shifting occurs only every m^th^ clock cycle, unlike the shift register of figure 3 where shifting occurs at every clock cycle. The shift register cell's output is multiplexed to the 2 bit lines. The cell's layout is 16 lambda by 175 lambda, which is 135 lambda longer than the unaugmented input decoder.

(3.) The parity checker cell, shown in figure 5, uses a chain of exclusive-NOR gates (i.e., NOT(EXOR)) to find the parity of the output vector. An EXNOR gate uses less area than an EXOR gate (see figure 6). When several EXNOR gates are cascaded, a long chain of pass transistors without any interposing inverters is created, which causes serious delays in signal propagation (pp.22-25 of [MC80]). We minimize this

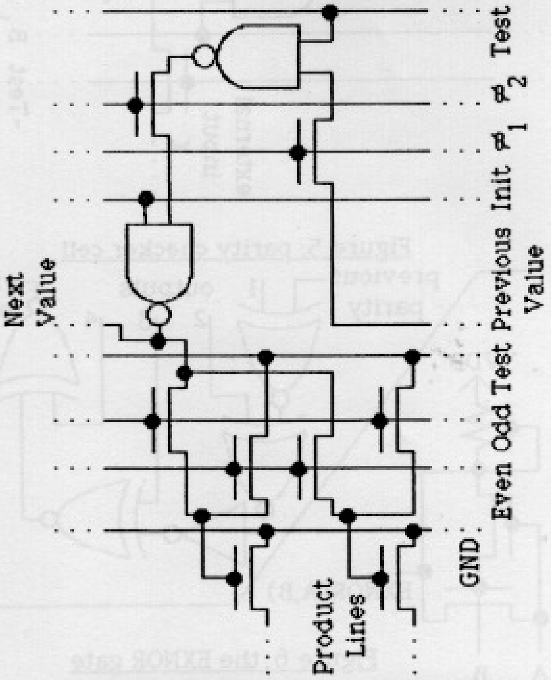
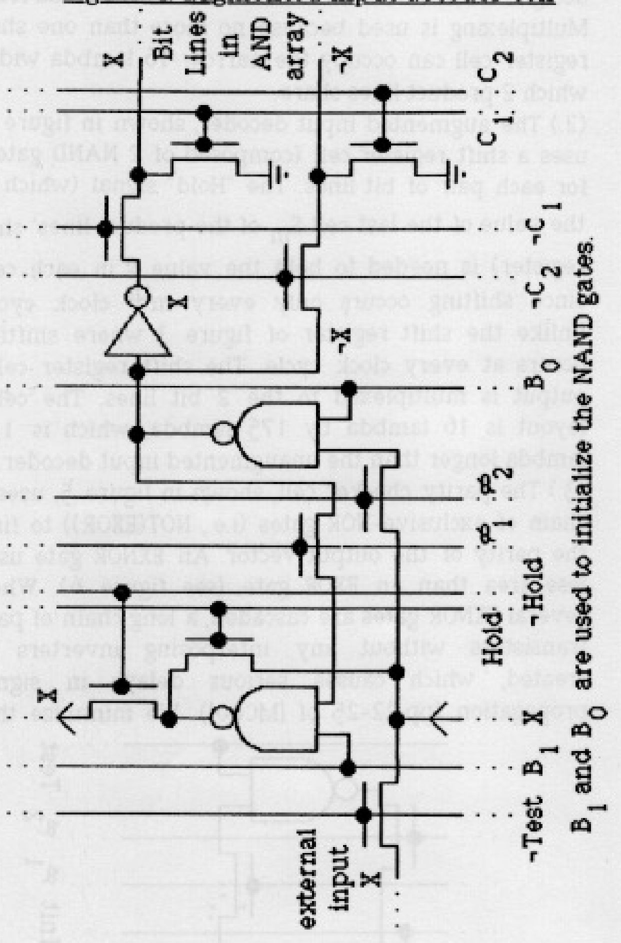


Figure 3: product lines' shift register

delay by placing an inverter after every fourth EXNOR gate, as shown in figure 5.

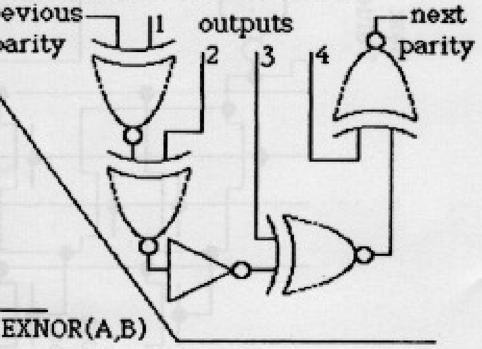
(4.) The cumulative parity comparator circuit is shown in figure 7. The signal Toggle has the value 1 only when the cumulative parity bit should be compared with its expected value emanating from the Toggle flip-flop. The signal ¬Hold stops the comparison after $I^2{}_j$ and $I^3{}_j$, and causes it during $I^4{}_{ij}$ and $I^5{}_{ij}$. B_0 clears the T flip-flop and starts the comparison at I^1, and similarly B_1 starts the comparison at $I^3{}_j$.

Figure 4: augmented input decoder cell



B_1 and B_0 are used to initialize the NAND gates.

Figure 5: parity checker cell



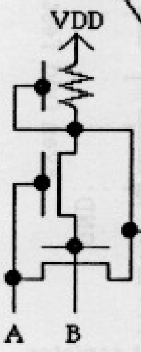


Figure 6: the EXNOR gate

The total area of the unaugmented PLA is 65m(2n+p) + 300(3n+m) + 550(4+p). The total extra area is 1360m + 2160n + 760p. For example, the area overhead of a 60 input, 60 output, 180 product term PLA is 420000 / 2249200 = 18.7%.

4. References

[DM81] W. Daehn and J. Mucha, "A Hardware Approach to Self-Testing of Large Programmable Logic Arrays", 1981, IEEE Trans. Comp., vol. C-30, pp. 829-833.

[F84] H. Fujiwara, "A New PLA Design for Universal Testability", 1984, IEEE Trans. Comp., vol. C-33, pp. 745-750.

[FK81] H. Fujiwara and K. Kinoshita, "A Design of Programmable Logic Arrays with Universal Tests", 1981, IEEE Trans. Comp., vol. C-30, pp. 823-828.

[FTA85] H. Fujiwara, R. Treuer and V.K. Agarwal, "A Low Overhead, High Coverage, Built-In, Self-Test PLA Design", McGill Report No. 84-13, Electrical Engineering Department, McGill University, Montreal.

[HJA84] K.A. Hua, J.-Y. Jou and J.A. Abraham, "Built-In Tests for VLSI Finite-State Machines", 1984, FTCS-14, pp. 292-297.

[HM83] S.Z. Hassan and E.J. McCluskey, "Testing PLAs Using Multiple Parallel Signature Analyzers", 1983, FTCS-13, pp. 422-425.

[MC80] C.A. Mead and L.A. Conway, Introduction to VLSI Systems, Addison-Wesley, Reading, MA, 1980.

[SKF83] K.K. Saluja, K. Kinoshita and H. Fujiwara, "An Easily Testable Design of Programmable Logic Arrays for Multiple Faults", 1983, IEEE Trans. Comp., vol. C-32, pp. 1038-1046.

[YA81] S. Yajima and T. Aramaki, "Autonomously Testable Programmable Logic Arrays", 1981, FTCS-11, pp. 41-43.

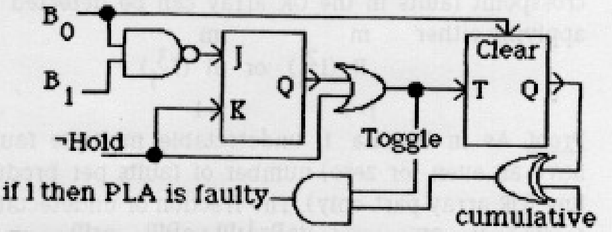


Figure 7: cumul. parity comparator