Configurations for I_{DDQ}-Testable PLAs

MANOJ SACHDEV
University of Waterloo, Canada
HANS KERKHOFF
MESA Research Institute

In these two PLA configurations, adjacent precharge lines activate, and adjacent evaluation lines evaluate, to complementary logic levels. This design-for-test technique makes it possible to use I_{DDQ} tests to detect all likely bridging faults—for the most part independently of the PLA's implemented function.

PROGRAMMABLE LOGIC ARRAYS are widely used in integrated circuits because they provide a simple, automated way of implementing complex Boolean functions. Often, microprocessors use PLAs to implement such functions as instruction decoding.[1,2] In its simplest form, a PLA is a highly uniform structure capable of implementing any Boolean function expressed in the sum-of-products form. The PLA structure consists of an AND plane and an OR plane. Earlier, PLAs were implemented using wired logic. A pull-up transistor (or resistor) pulls each output high. Depending on the input data on switching transistors connected to output lines, output lines are pulled low—or evaluated.

For embedded applications, designers generally prefer dynamic PLAs because of their smaller area compared to static PLAs, their low power dissipation, and their greater throughput via pipelined processing. Generally, we implement them as INV-NOR-NOR-INV structures. In a typical dynamic PLA, precharge and evaluation functions replace wired logic. Figure 1 shows an example of a dynamic PLA configuration with three inputs, four product terms, and three outputs.

The I_{DDQ} test method is a powerful technique for detecting bridging defects in digital circuits;[3,4] it is incomparable in terms of quality, simplicity, and cost. At the same time, it is a relatively slow method for testing logic. Therefore, there is a strong motivation to improve defect coverage for each I_{DDQ} measurement. With the two PLA configurations presented in this article, we attempt to achieve this goal.

PLA fault model and defects

Traditionally, a PLA fault model consists of the following faults:

- *Line stuck-at faults.* These include single- or multiple-line stuck-at faults. The PLA fault model considers faults on input lines, product lines, output lines, input and output registers, pull-up logic, and so on.
- *Crosspoint faults.* A crosspoint may exist at an undesirable location, or a crosspoint may be missing from a desirable location. These faults are known as extra- and missing-crosspoint faults. They are also known as crosspoint growth and crosspoint shrinkage faults.
- *Bridging faults.* These can exist among the input, product, and output lines. The model also considers bridging faults between input and product lines and between product lines and output lines.
- *Open faults.* The model considers these on input, product, and output lines. Open faults can cause sequential behavior in CMOS logic circuits. However, this behavior is not likely in PLAs because dynamic PLAs rarely contain CMOS logic gates other than inverters. Most open faults in a PLA occur on lines or in single transistors and lead to stuck-at fault behavior on lines or in pull-up or pull-down transistors. As a result, tests for stuck-at faults detect opens, so the fault model does not include them explicitly.

Tamir and Sequin[5] developed a PLA fault model based upon realistic physical defects.

0740-7475/99/$10.00 © 1999 IEEE

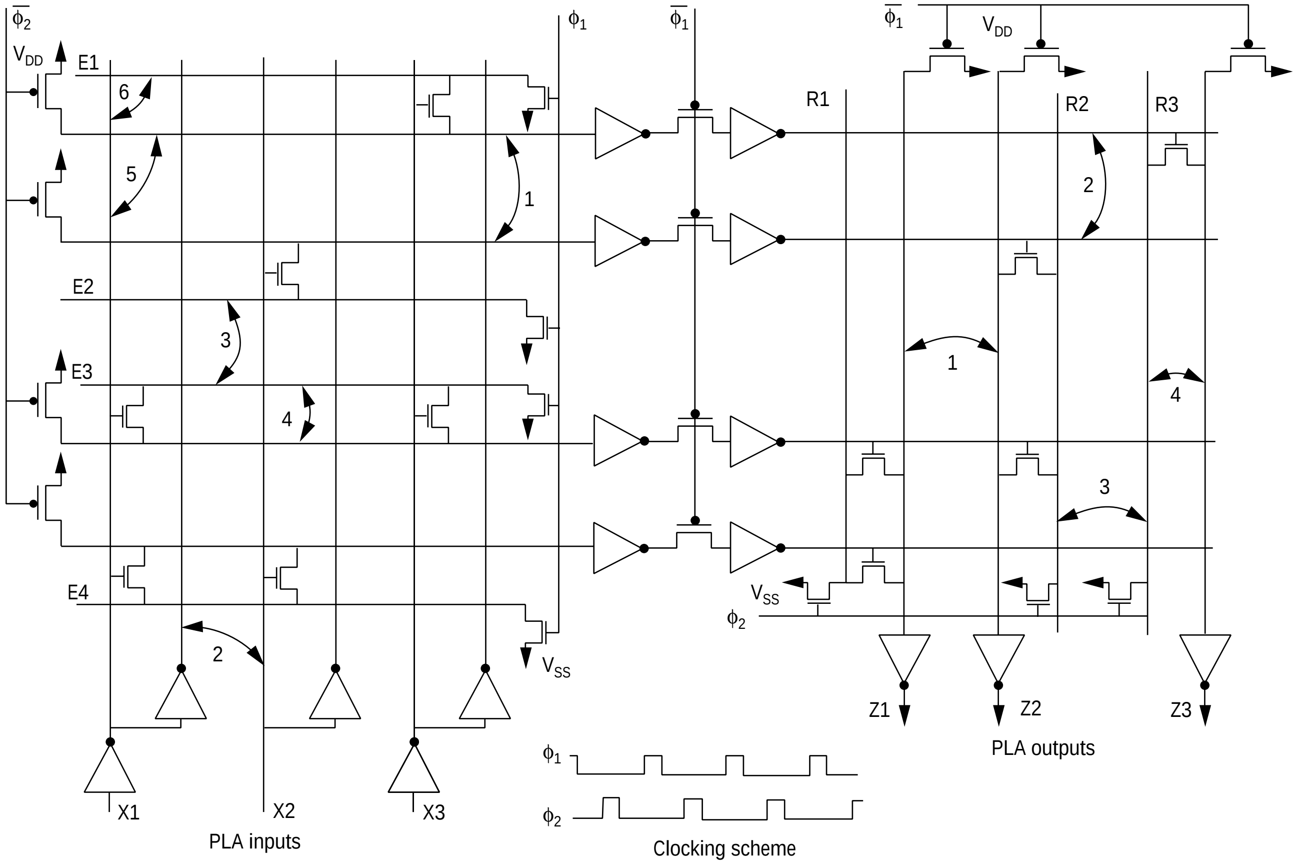


Figure 1. *A dynamic PLA with possible bridging faults. Evaluation lines are Ei in the AND plane and Rj in the OR plane.*

They argued that a missing-crosspoint fault in the AND array has the same effect as a weak logic 1 on an input line; instead of turning on the crosspoint transistor, the fault turns it off. Similarly, a missing-crosspoint fault in the OR array is equivalent to a weak logic 1 on the corresponding product line. A break in an input line can cause the line to float, which is equivalent to a weak 0 fault, a weak 1 fault, or both. Therefore, a model that considers weak 0/1 faults on input lines does not need to consider break faults on the input lines.

Furthermore, Tamir and Sequin suggested that troublesome sequential faults are not possible on product and output lines, because these lines connect either to pull-up transistors or to drains of crosspoint transistors. In the first case, a break will give a line stuck-at-0 behavior. In the second case, a break is the same as a missing- or extra-crosspoint fault.

In developing a PLA fault model, Fujiwara considered bridging faults in addition to stuck-at and crosspoint faults.[6] He proposed two new augmented PLA architectures for universal testability, which solved the problems of extra hardware and performance degradation associated with previous solutions.

Arguably, the bridging fault is the most dominant fault type in modern CMOS processes. Bridging faults in PLAs are much more likely than in random logic, owing to the extensive, closely spaced interconnects. Chandramouli et al.[7] analyzed bridging faults in a CMOS PLA and argued that previous works either ignored bridging faults or assumed a wired-AND behavior. This assumption is not valid in MOS technologies. Furthermore, PLAs implemented with dynamic logic put extra constraints on bridging-fault detection owing to the dynamic nature of their operation.

The distinctive feature of a dynamic PLA is the presence of precharge and evaluation lines in the AND and OR planes. Dynamic PLAs generally use a two-phase nonoverlapping clock scheme. The ϕ_1 phase evaluates the AND plane and precharges the OR plane. The ϕ_2 phase evaluates the OR plane and precharges the AND plane. Dynamic latches between the AND plane and the OR plane buffer the AND plane's output.

During layout, designers usually rearrange the precharge and evaluate lines in a dynamic PLA to make efficient use of the area. Therefore, in the AND plane, sometimes two product lines are adjacent to each other and sometimes two evaluation lines are adjacent. Similarly, in the OR plane, sometimes sum lines are adjacent and sometimes evaluation lines are adjacent.[7] Four types of bridging faults are possible in the AND and OR planes; Figure 1 shows these as faults 1, 2, 3, and 4:

- *Type 1.* These are bridging faults between the product lines in the AND plane (and between sum lines in the OR plane). These faults are similar to the output of a dynamic logic gate that is precharged by a PMOS transistor and evaluated by NMOS transistors. Only functional testing can detect such faults. I_{DDQ} does not detect them because the voltage conflict across the fault cannot be sustained in steady state.
- *Type 2.* These are bridging faults between the input lines in the AND plane and product lines in the OR plane; they are similar to bridging faults in completely static CMOS circuits. We should use I_{DDQ} to detect these faults. Functional testing for such faults is not efficient without a design-for-test scheme, which can increase area and reduce performance. A single test vector can excite a bridging fault between any two input lines in the AND plane. The requirement for this vector is that all adjacent bitlines in the AND plane, as well as all adjacent product lines in the OR plane, must have complementary logic values. Under these conditions, voltage across a bridging fault results in elevated I_{DDQ}.
- *Type 3.* This category includes bridging faults between adjacent evaluation lines—such as defect 3 between E2 and E3 in Figure 1. These faults do not affect the PLA's functionality and are not detectable.
- *Type 4.* This category consists of the defects between precharge and evaluation lines (see defect 4 in Figure 1). These faults are identical to crosspoint device stuck-on faults, and test strategies for crosspoint faults will detect them as well. I_{DDQ} does not detect them because both precharge and evaluation lines are dynamically excited—that is, they precharge logic high and evaluate low. Therefore, we cannot create and sustain a logical conflict for I_{DDQ} testing.

Besides these four categories, other bridging faults are possible (see faults 5 and 6 in Figure 1). These faults, between input lines and product (or evaluation) lines, are less likely because the lines affected are routed on different metallization levels.

Logic testing of PLAs

Test pattern generation for PLAs is a complex problem for a variety of reasons. First, PLAs contain logic redundancies that cause untestable faults. Second, PLAs contain a reconvergent fan-out that makes the test generation task difficult. Furthermore, an embedded PLA has interdependencies among its inputs that may exclude the input patterns that stimulate and propagate the fault effect through the PLA.[8]

At the same time, PLAs have a very regular test structure. Therefore, DFT schemes potentially make test pattern generation considerably simpler. The key to the design of testable PLAs is the independent control of input and product lines. Other researchers have proposed several DFT schemes with and without BIST.[9] Typically, the area overhead and performance impacts of such schemes are prohibitive. They need a large number of test vectors to test logic redundancies and reconvergent fan-outs. Moreover, these schemes either do not consider bridging faults, or they make unrealistic assumptions about bridging-fault detection in PLAs.[7] All these arguments provide motivation to devise I_{DDQ}-testable PLAs that will allow us to test most realistic defects easily and independently of the PLA's implemented function.

I_{DDQ}-testable dynamic CMOS PLAs

Although both PLA testing and I_{DDQ} testing are well-researched topics, researchers have paid very little attention to I_{DDQ} testing of PLAs. In general, bridging defects in dynamic PLAs are much more difficult to detect with I_{DDQ} than those in PLAs with static logic. A study by Chandramouli et al.[7] on bridging faults in CMOS PLAs confirms this hypothesis. I_{DDQ} testing detects only type 2 bridging faults. Furthermore, I_{DDQ}-based tests do not detect transistor leakage faults in a dynamic PLA, owing to its dynamic character.

However, I_{DDQ} testing can detect both bridging and leakage faults in PLAs that have minor architectural modifications. For those faults that go undetected by I_{DDQ} tests (such as missing-crosspoint faults and opens), we can use quite simple tests. For example, as mentioned before, open defects in PLAs lead to stuck-at behavior, which is easy to test. Similarly, we can apply logic and DFT procedures for crosspoint faults.[9]

In some aspects, PLAs are similar to RAMs. For example, both contain orthogonal data and control lines to decode appropriate data. With RAMs, we can reduce test complexity significantly by creating an I_{DDQ} test mode.[10] Similarly, we should be able to reduce PLA test cost with an I_{DDQ} test mode. Besides reducing test cost, modifications would make testing largely independent of the PLA's implemented function.

First enhanced PLA configuration

Figure 2 is the schematic diagram of a dynamic PLA with DFT features. The PLA is the same as that in Figure 1, except in the highlighted areas, which show modifications. In the dynamic implementation of a logic gate, the output precharges to V_{DD} (or V_{SS}) and evaluates to the complementary Boolean level V_{SS} (or V_{DD}). Similarly, in a dynamic PLA, the evaluation lines should evaluate to logic levels that are complementary to the corresponding precharge lines—product lines in the AND plane and sum lines in the OR plane. For example, in a conventional PLA (Figure 1), all the product lines precharge to V_{DD} and evaluate to V_{SS}. As long as we

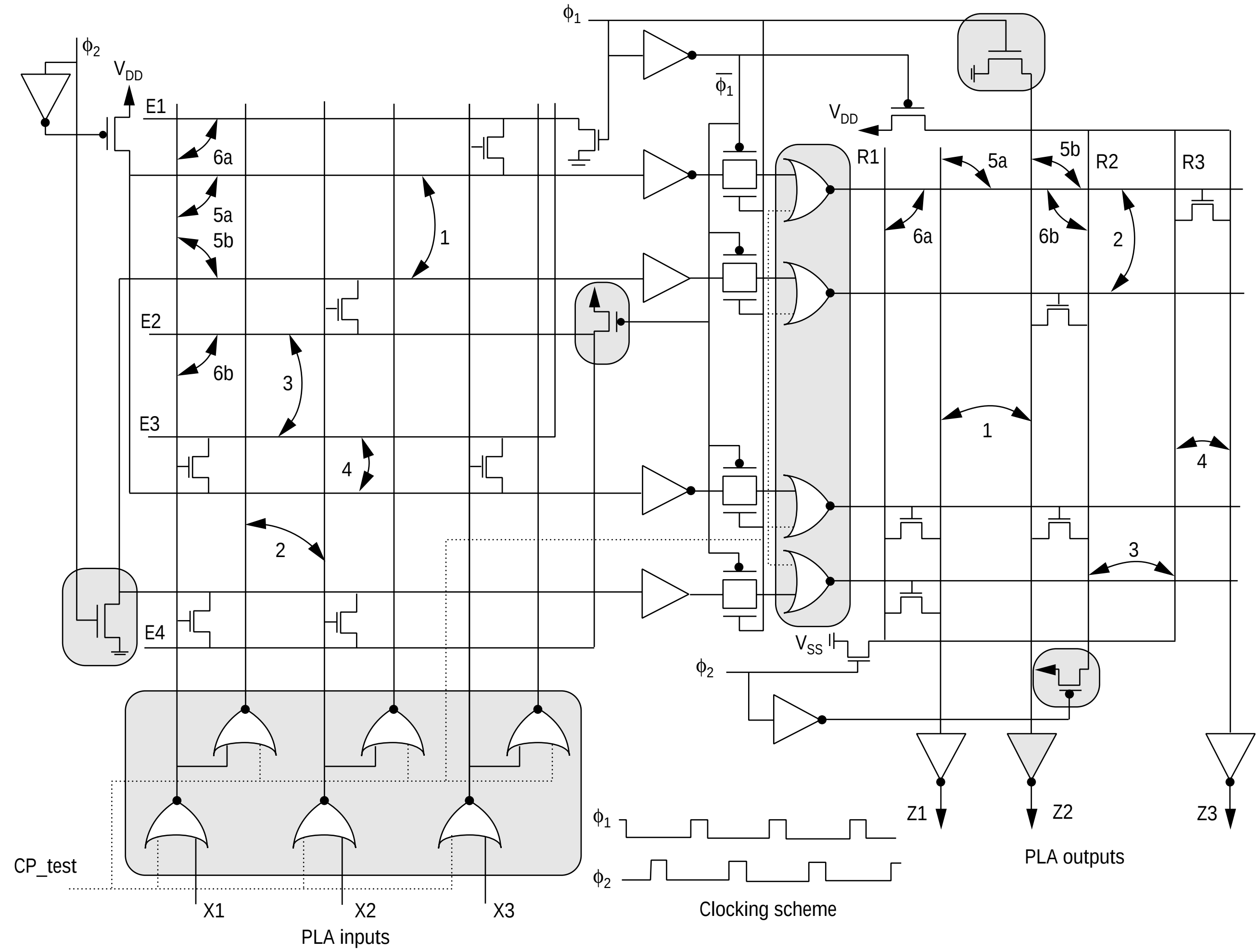


***Figure 2.** An I_{DDQ}-testable dynamic PLA. Evaluation lines are Ei in the AND plane and Rj in the OR plane. The dotted line represents the test control signal, CP_test.*

meet the constraint of precharging and evaluating to complementary logic levels, we are free to choose V_{DD} or V_{SS} for precharge or evaluation for individual product or sum lines.

We have exploited this feature to enhance the I_{DDQ} testability of PLAs. The odd product lines in the AND plane and the odd sum lines in the OR plane precharge to V_{DD}, as before. However, the even product lines in the AND plane and the even sum lines in the OR plane precharge to V_{SS} (shaded NMOS transistors). Similarly, we have also modified corresponding even evaluation lines in the AND and OR planes; these lines evaluate to V_{DD} (shaded PMOS transistors). Noninverting drivers buffer the output of even product and sum lines to maintain the proper logic operation.

This design has a test control signal, CP_test. In normal mode, CP_test stays at logic-low level, which ensures normal PLA operation. A logic high at CP_test puts the PLA into test mode. The I_{DDQ} test mode puts all crosspoint transistors in the off state. Such an arrangement is necessary to distinguish currents due to bridging defects from current via crosspoint transistors.

Primary inputs of the PLA and product lines going to the OR plane are gated by two-input NOR gates. Therefore, when CP_test is high, all input lines in the AND and OR planes are pulled down, ensuring that all crosspoint transistors are in the off state.

Furthermore, pass transistor pairs replace the single *p*-channel pass transistors between the AND and OR planes. The single *p*-channel pass transistors cause a logic low to appear as volts at the input of the subsequent inverter, resulting in a steady flow of current. Replacing them with pass transistor pairs keeps this undesirable situation from occurring.

Bridging-fault types 1 through 4. For the first I_{DDQ} measurement, we keep both clock phases high, ensuring that all

Table 1. *I_{DDQ} tests and detection conditions for bridging and leakage faults in the PLA shown in Figure 2.*

Test	Test conditions	Detected faults	Comments
1	$\phi_1 = 1$, $\phi_2 = 1$, CP_test = 1	AND plane: bridging faults types 1, 3, and 4; all stuck-on crosspoint faults; OR plane: bridging faults types 1, 3, and 4; all stuck-on crosspoint faults	Test is function-independent.
2	$\phi_1 = 1$, $\phi_2 = 0$, CP_test = 0 Inputs with complementary data	AND plane: bridging faults types 2 and 3	Test is function-independent.
3	$\phi_1 = 1$, $\phi_2 = 0$, CP_test = 0 Inputs with appropriate data	OR plane: bridging faults type 2	Test is function-dependent.

the precharge lines and evaluation lines in both planes are active at the same time. In other words, the lines precharge and evaluate at the same time. Now, depending upon input stimuli conditions, some crosspoint transistors may be on; this will cause contention between their precharge and evaluation lines. Such a situation will result in elevated current and invalidate the test. Therefore, we keep CP_test high as well, to put all crosspoint transistors in the off state. These conditions drive all adjacent precharge and evaluation lines to complementary logic levels. Hence, any bridging fault among them will result in an elevated quiescent current.

Now, let's reconsider the bridging-fault categories we listed earlier. This I_{DDQ} measurement will detect all bridging faults of types 1, 3, and 4 in both planes. It will also detect leakage faults in crosspoint transistors (stuck-on faults, for example). This test is independent of the function implemented in the PLA.

We need the second I_{DDQ} measurement to test for type 2 faults in the AND plane. During this measurement, we keep clock phase ϕ_1 high, clock phase ϕ_2 low, and test signal CP_test low. These conditions ensure that no crosspoint transistor is on in the AND plane. Now, we drive adjacent inputs to complementary logic levels. This excites all type 2 faults in the AND plane, and elevated quiescent current detects them. The same arrangement also ensures detection of type 3 faults in the AND plane. This test is also independent of the function implemented in the PLA.

Similarly, we can detect type 2 faults in the OR plane by keeping adjacent product lines at complementary logic values. However, the effectiveness of this test depends on the AND plane's implemented function. Table 1 lists the necessary conditions for these three I_{DDQ} measurements and the faults each measurement detects.

Bridging-fault types 5 and 6. For detection of these bridging faults and their derivatives, we do not need an additional DFT scheme. Instead, we must determine whether the bridge affects an odd or even product (or evaluation) line. This is because alternative product lines precharge (and alternative evaluation lines evaluate) to complementary logic levels. Thus, they require different fault detection conditions. However, in general, we can detect these defects with a proper combination of input data, clock phases, and CP_test. For example, logic high on both clock phases together with high CP_test will ensure detection of faults 5a and 6b in both planes. We can detect other faults similarly.

Discussion. The three I_{DDQ} measurements listed in Table 1 will not detect open defects. An open defect in a precharge transistor will cause the corresponding product or sum lines to have multiple stuck-at-0 or stuck-at-1 faults. Similarly, an open defect in the evaluation lines will also cause multiple stuck-at-0 or stuck-at-1 faults, which are easy to detect with logic testing. An open defect can also cause a crosspoint transistor to be open. For detection, this requires either functional testing or structural testing with DFT schemes described elsewhere.[9]

In this DFT scheme, even product lines evaluate to logic high through NMOS crosspoint transistors. Such a scheme will result in evaluation to voltage on even product lines. Therefore, we must take care to size the subsequent buffer to take into account the threshold voltage drop. The voltage threshold drop on even product lines may be an issue for many design applications. It may also result in dc power dissipation in the buffer, which would in turn increase power consumption. Finally, in low-voltage applications, it may result in unacceptably reduced noise margins.

There are several possible solutions to the problem of threshold voltage drop on the even product and sum lines. Of these, replacing NMOS crosspoint transistors with PMOS crosspoint transistors for even product and sum lines is probably the simplest. Although such an arrangement would not cause the threshold voltage drop, it would have some other consequences. For example, testing of type 4 bridging defects would require an elaborate arrangement because forcing logic zero would not switch off PMOS crosspoint tran-

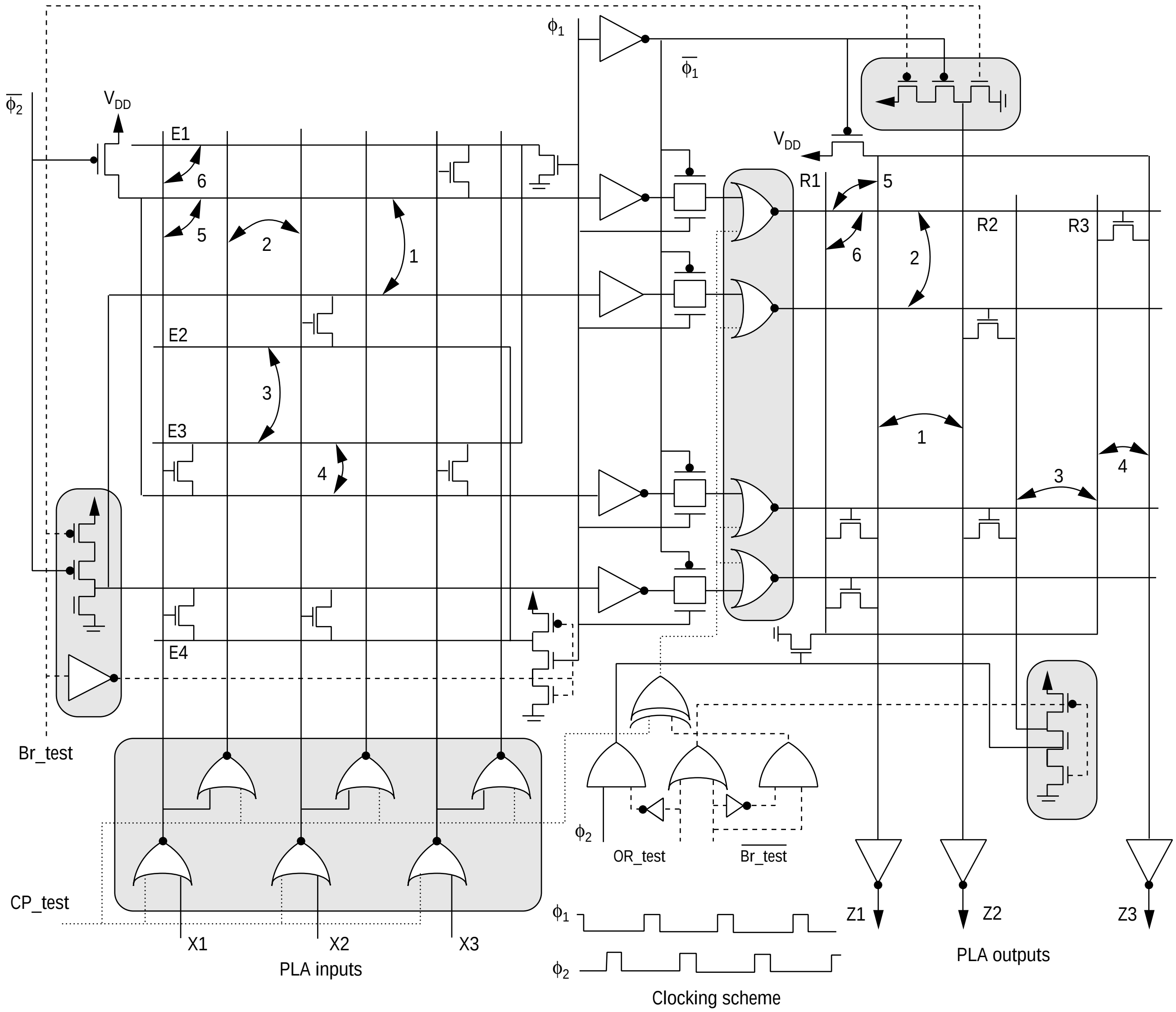


Figure 3. *An alternative DFT scheme for I_{DDQ}-testable PLAs. Evaluation lines are Ei in the AND plane and Rj in the OR plane. The dotted line represents CP_test; the dashed lines represent two additional test control signals, Br_test and OR_test.*

sistors. Furthermore, there would be constraints on logic implementation in the AND and OR planes.

The application of latches or sense amplifiers instead of inverters (buffers) to restore the logic level is yet another solution. However, this might increase the PLA's area overhead and pitch, and reduce performance without really being of significant benefit.

Second enhanced PLA configuration

For situations in which the threshold voltage drop on the even product lines is unacceptable, we can suitably modify the first DFT scheme. Figure 3 shows a modified version of the first DFT scheme, which retains the first scheme's fault coverage, alleviates the problem of the threshold voltage drop, and does not result in excessive area and performance penalties.

However, this scheme requires two additional test control signals. Signal Br_test controls even product and evaluation lines in the AND and OR planes in test mode. Similarly, signal OR_test controls evaluation lines in the OR plane in test mode. In normal mode, both the Br_test and OR_test signals remain at logic low, and the PLA functions as usual. That is, product and sum lines precharge to logic high and evaluate to logic low. However, test conditions drive these signals logic high to excite different types of faults. (We ex-

Table 2. *I_{DDQ} tests and detection conditions for bridging and leakage faults in the PLA shown in Figure 3.*

Test	Test conditions	Detected faults	Comments

1	$\phi_1 = 1$, $\phi_2 = 1$, CP_test = 1,
	Br_test = 1, OR_test = 0
		AND plane: bridging faults types 1, 3, and 4;
		all stuck-on crosspoint faults
		OR plane: bridging faults types 1, 3, and 4;
		all stuck-on crosspoint faults
			Test is function-independent.

2	$\phi_1 = 1$, $\phi_2 = 0$, CP_test = 0, Br_test = 0, OR_test = 0 Inputs with complementary data	AND plane: bridging faults types 2 and 3	Test is function-independent.
3	$\phi_1 = 1$, $\phi_2 = 1$, CP_test = 1, Br_test = 1, OR_test = 1	OR plane: bridging faults types 1 and 2	Test is function-independent.

plain these conditions in the subsequent section.)

There may be situations in which it is not possible to have a total of three inputs for test purposes. In such cases, we can design an onboard state machine with only one or two inputs, whose outputs may be decoded to derive the test signals. Alternatively, CP_test may be decoded from both clock phases. (As is clear from Tables 1 and 2, a high CP_test is necessary only when both clock phases are high.)

Bridging-fault types 1 through 4. Detection of bridging faults in this PLA is similar to that in the first configuration we discussed. For type 1, 3, and 4 bridging faults in the AND plane, the following conditions are necessary: CP_test must be at logic high so that all crosspoint transistors in the PLA are in nonconduction state. Clock phase ϕ_1 must be kept high, to drive logic low on odd evaluation lines. Also, clock phase ϕ_2 must be kept high to enable odd product lines to be logic high. Furthermore, Br_test must be at logic high, which enables even product lines in the AND plane to be driven logic low and even evaluation lines in the OR plane to be driven logic high. Under these conditions, I_{DDQ} measurement will detect type 1, 3, and 4 bridging faults in both planes as well as leakage faults in all crosspoint transistors in both planes. This measurement, listed as the first measurement in Table 2, is independent of the function implemented in the PLA.

Detection of type 2 bridging faults in the AND plane requires an additional I_{DDQ} measurement. For this measurement, we drive adjacent input lines in the layout to complementary logic values. We keep clock phase ϕ_1 at logic high and clock phase ϕ_2 at logic low. Furthermore, we keep all the test inputs (CP_test, Br_test, and OR_test) at logic low. In other words, we keep the PLA in normal mode and detect faults by giving appropriate inputs. Such an arrangement ensures the detection of bridging faults between input lines. This measurement will also detect type 3 bridging faults in the AND plane. This measurement, listed as test 2 in Table 2, is independent of the function implemented in the PLA.

Detection of type 2 faults in the OR plane requires a bit of explanation. To detect these faults independently of the implemented function, we apply the following scheme. We keep clock phase ϕ_2 high, which makes odd product lines logic high. We also keep Br_test high, which drives even product lines to logic low. We keep CP_test high so that all crosspoint transistors in the AND plane are in the off state and do not invalidate the test.

Now, we keep clock phase ϕ_1 high, enabling the product lines to drive the OR plane. However, for that to happen, CP_test must not be high, because a high CP_test will make all the product lines in the OR plane low. Such a condition would not allow detection of type 2 defects. Therefore, we EXOR CP_test with the logic AND of Br_test and OR_test, which are both high for this test. Essentially, keeping both these signals high disables the application of CP_test on the OR plane. Such an arrangement ensures that the adjacent product lines in the OR plane have complementary logic values. Unfortunately, keeping both clock phases high excites sum and evaluation lines to complementary logic values. This may cause leakage through crosspoint transistors, depending upon the logic state of the product lines. Therefore, to avoid invalidation of the test, the additional signal, OR_test, controls the evaluation lines in the high impedance state. The OR_test signal, which normally remains at logic low, is at logic high for this test. Under these conditions, a type 2 bridging fault will give rise to an elevated I_{DDQ} level.

Bridging-fault types 5 and 6. Detection of these bridging fault types is simpler in this configuration because all product lines precharge high and all evaluation lines evaluate low. However, as with the first PLA configuration, a proper combination of input data, clock phases, and CP_test ensures detection of these faults. For example, logic high on both clock phases together with high CP_test ensures detection of type 5 faults in both planes. We can detect other faults similarly.

Implementation cost

We can quantify the cost of building I_{DDQ} testability into PLAs as area overhead and potential performance degradation. In general, we should weigh these costs against the potential test benefits of the proposed configurations.

The area overhead of implementing these schemes is small. In both of the configurations we propose, placement of extra two-input NOR gates on input and product lines takes the most of extra area. The total number of gates required equals the sum of input and product lines. However, we must take care to fit additional transistors within the PLA pitch, which may require careful layout planning. In percentage terms, area overhead reduces as the PLA complexity increases. The second configuration requires a few more gates to provide extra control over the clock.

Besides the area overhead of extra transistors, the first configuration requires one control input, and the second configuration requires two. In embedded applications, where direct access is often expensive or impractical, this small area overhead is acceptable.

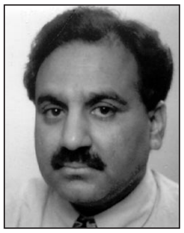
The performance impact of incorporating I_{DDQ} testability is also small. The constraint to a PLA's performance is the data path. How quickly the lines can precharge and evaluate largely determines the overall PLA performance. Our proposed configurations introduce two 2-input NOR gates in the data path; this should not result in large performance degradation. Furthermore, PLAs contain long interconnects. In state-of-the-art CMOS technologies, interconnect delay constitutes a major part of the overall delay and is often the limiting factor.[11] Therefore, the addition of a couple of logic gates in the data path should not result in unacceptable degradation in PLA performance.

EMBEDDED PLAS are popular means of realizing Boolean functions. In this article, we have proposed two I_{DDQ}-testable dynamic PLA configurations. They efficiently test all likely bridging faults with the I_{DDQ} test technique. These configurations require extra logic gates and a test control signal to implement the testability scheme. However, performance and area penalties are significantly less than those of other PLA design-for-test schemes.

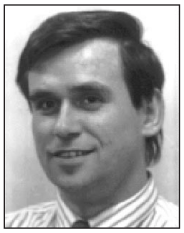
What is the true area overhead and performance impact of our configurations? What is the actual bridging-fault sensitivity? A layout and silicon implementation will answer these and many other practical questions about this concept. We will address these topics in our future research.

References

1. A. Carbine and D. Feltham, "Pentium Pro Processor Design for Test and Debug," *Proc. Int'l Test Conf.,* IEEE Computer Society Press, Los Alamitos, Calif., 1997, pp. 294-303.
2. R.S. Fetherston, I.P. Shaik, and S.C. Ma, "Testability Features of AMD-K6 Microprocessor," *Proc. Int'l Test Conf.,* IEEE CS Press, 1997, pp. 406-413.
3. Y. K. Malaiya and R. Rajsuman, *Bridging Faults and I_{DDQ} Testing,* IEEE CS Press, 1992.
4. S. Chakravarty and P.J. Thadikaran, *Introduction to I_{DDQ} Testing,* Kluwer Academic Publishers, Boston, 1997.
6. Y. Tamir and C.H. Sequin, "Design and Application of Self-Testing Comparators Implemented with MOS PLAs," *IEEE Trans. Computers,* Vol. C-33, No. 6, June 1984, pp. 493-506.
6. H. Fujiwara, "A New PLA Design for Universal Testability," *IEEE Trans. Computers,* Vol. C-33, 1984, pp. 745-750.
7. V. Chandramouli et al., "Bridging Faults and Their Implication to PLAs," *Proc. Int'l Test Conf.,* IEEE CS Press, 1990, pp. 842-859.
8. D.L. Liu and E.J. McCluskey, "Design of Large Embedded CMOS PLAs for Built-In Self Test," *IEEE Trans. Computer-Aided Design,* Vol. 7, No. 1, Jan. 1988, pp. 50-59.
9. V.K. Agarwal, "Easily Testable PLA Design," *VLSI Testing,* T.W. Williams, ed., Elsevier Science Publishing Company, Inc., New York, 1986, ISBN 0 444 87895 5, pp. 65-93.
10. M. Sachdev, *Defect Oriented Testing for CMOS Analog and Digital Circuits,* Kluwer Academic Publishers, Dordrecht, the Netherlands, 1997.
11. T. Chang Lee and J. Cong, "The New Line in IC design," *IEEE Spectrum,* Mar. 1997, pp. 52-58.



Manoj Sachdev is an associate professor in the Electrical and Computer Engineering Department at the University of Waterloo, Canada. His research interests include various aspects of VLSI testing, low-power and high-performance circuit design techniques, and VLSI design for manufacturability. He has written extensively in these areas, and he received the best paper award at the 1997 European Design and Test Conference. Sachdev holds six patents and has several patents pending. He is a senior member of the IEEE.



Hans Kerkhoff heads the Testable Design and Test of Microsystems group for the MESA Research Institute and the MESA Test Centre. He also works part-time for the Philips Research Laboratories at Eindhoven in IC design and testing. His interests are in testable design and testing of mixed-signal microsystems. Kerkhoff received his MSc degree in telecommunication from the Technical University of Delft; he received the Delft Hogeschool Award for his MSc thesis. He obtained a PhD in technical science (microelectronics) from the University of Twente.

Send questions and comments to Manoj Sachdev, Electrical and Computer Engineering, University of Waterloo, 200 University Ave. West, Waterloo N2L 3G1 Canada; msachdev@ece.uwaterloo.ca.