SAND--90-2900C

DE91 010579

1991

CMOS IC Fault Models, Physical Defect Coverage, and I_{DDQ} Testing

Ronald R. Fritzemeier, Charles F. Hawkins*, and Jerry M. Soden

Sandia National Laboratories
Division 2118
Albuquerque, NM 87185
Office (505) 845-8575
Fax (505) 846-8755

* Electrical and Computer Engr. Dept.
The University of New Mexico
Albuquerque, NM 87131
Office (505) 277-2264

Abstract

The development of the stuck-at fault (SAF) model is reviewed with emphasis on its relationship to CMOS integrated circuit (IC) technologies. The ability of the SAF model to represent common physical defects in CMOS ICs is evaluated. A test strategy for defect detection, which includes I_{DDQ} testing, is presented.

Introduction

This paper reviews the origin and use of the stuck-at fault model, the detection of physical defects, the use of quiescent power supply current (I_{DDQ}) monitoring, and proposed test methodologies for improving the quality and reliability of CMOS integrated circuits. The present urgency to follow methods that lead to zero defects in production should drive CMOS IC test philosophies toward defect detection strategies and away from strategies based upon the SAF model. The SAF model originated with current-switching technologies and evolved into the primary test metric for bipolar ICs and subsequently for CMOS ICs. The adequacy of the SAF model as the test quality metric for CMOS ICs is reviewed. A test strategy which includes I_{DDQ} testing for quantitative defect coverage is discussed.

Development of the Stuck-At Fault Model

In 1959 Eldred devised structural tests for the detection of faulty components in logic circuits [1]. These tests were based on the physical connectivity of diode and vacuum tube components for logic functions. Eldred defined four logic conditions that were used to generate test patterns to evaluate the correct logic behavior of combinational logic gates. These conditions for test pattern generation utilized the path sensitization concept to test for stuck-at faults on the logic gate nodes. Although Eldred did not refer to these as stuck-at faults, he clearly was evaluating single stuck-at faults. He demonstrated this test pattern generation method for a data processor circuit.

In the early 1960s the single stuck-at fault model was formalized and Boolean algebra was used to analyze the effects of these faults in combinational logic. Poage discussed stuck faults in 1963 [2]. Galey et al restricted their work to the subset of failures which they found to be "the universal choice of diagnosticians" [3]. They assumed that any "line" of the circuit might be stuck and gave a simplified example of transistors that always had a logic output of 0 or 1. In 1966 Armstrong similarly restricted his investigation to failures that caused any "wire" to be (or appear logically to be) stuck at logic 0 or 1 [4]. He also defined that all "cut" (open-circuited) combinational gate inputs were logically equivalent to having that input stuck at 0 (OR and NOR gates) or 1 (AND and NAND gates).

The stuck-at fault model therefore originated from the observed electrical behavior of common physical defects and provided a convenient logical abstraction suitable for modeling with a digital computer. Physical defects, breakdown mechanisms, and assembly flaws in tubes, circuit boards, discrete transistors, and passive components often caused either short circuit connections to the circuit's power supply nodes or open circuits. The defects producing power supply short circuits caused the "physical" stuck-at fault. Defects producing open circuits caused "logical" stuck-at behavior due to the nature of the current-switching logic used, such as discrete diode logic and integrated circuit DTL and TTL logic [5]. Since current-switching logic requires current drive to produce follow-on logic transitions, an open-circuit path causes a node to behave the same as a node stuck to a power rail.

The presumed dominance of physical and logical stuck-at behavior over the vast majority of effects produced by these common defects profoundly influenced the development of test philosophy and practice. Diagnosis of circuit failures typically began with the assumption that a stuck node existed somewhere in the circuit. Test strategies were developed that assumed a complete test was one in which all "detectable" stuck-at faults would be detected [4]. As a result, the "classical" SAF model became widely accepted as the model for logic faults, i.e., single, permanent, stuck-at-zero (SA0) and stuck-at-one (SA1) faults.

The physical basis for the SAF model was appropriate but not perfect for DTL and TTL IC technologies in the 1960s and into the 1970s. The relation between TTL defects and the SAF model was reported by Beh et al [6]. They found that "hard" shorts and even opens caused SAF behavior in a TTL gate. They also found certain types of TTL defects whose detection would not be guaranteed by a SAF test set. Their study concluded that the SAF test set would detect many TTL defects but a second test strategy was needed for delay defects not detected by SAF tests.

As mentioned, Armstrong stated that open-circuited logic gate inputs behaved as SAFs [4]. Fig. 1 (a) shows a 2-NAND gate with an open-circuited input and Fig. 1 (b) shows simplified 2-NAND circuitry for TTL technology. The open-circuit defect interrupts the current path of the emitter input. Absence of sinking current forces the affected input to behave as though it was permanently stuck in the non-controlling logic 1 state for NAND or AND gates. An open circuit in a TTL NOR or OR gate input would similarly cause it to behave as though it was permanently stuck in the non-controlling logic 0 state.

The natural fit of the SAF model to computer modeling allowed tremendous progress to be made in both the simulation of SAF behavior and the algorithmic generation of "complete" test sets that would find all detectable SAFs. Unfortunately, the strengths of the SAF model also tended to encourage people to look for ways to express every new defect behavior in terms of the SAF. Further developments and time thus tended to obscure the original intent and assumptions made by the SAF model. For instance, Armstrong stated, in his procedure for finding more nearly minimal SAF tests for combinational logic circuits, "when detecting 'all faults' is spoken of, it will mean 'all detectable faults' " [4]. The issue of the various ways that a defect could be detected was not directly considered; rather, the SAF model was strictly and exclusively used. This assumption of

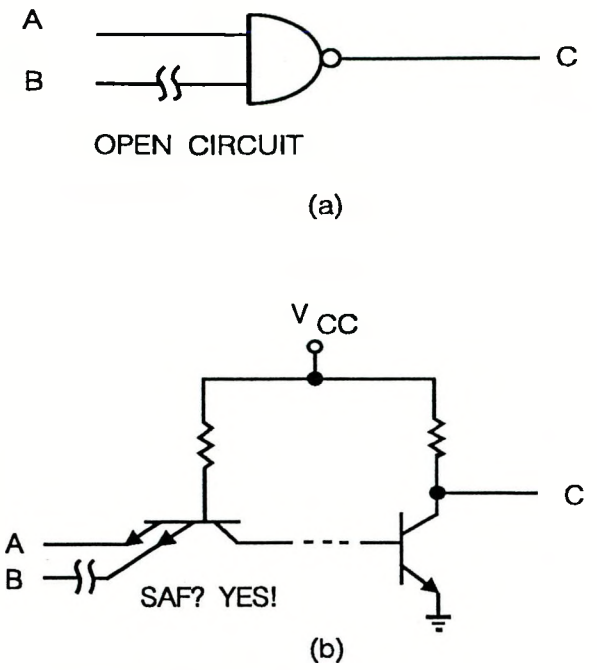

Fig. 1. (a) 2-NAND with open circuit in input B, and (b) TTL 2-NAND with open circuit in input.

detectability has carried on into present practice even where logically "undetectable" faults can clearly produce significant problems [7,8].

A large volume of literature developed around the stuck-at fault model in the 1960s due to the rapidly increasing use of ICs in products such as computers. This trend continued and even accelerated on into the 1970s. These issues paved the way for the common perception that the stuck-at fault model provides the following advantages: (1) SAFs can be modeled at the logic level and are "independent of technology", (2) SAFs can be analyzed by "known" methods, and (3) SAF coverage has been "proven" to be an effective measure of test quality [9].

In the late 1970s and the 1980s, researchers began addressing the issue of whether the SAF model was appropriate for CMOS ICs. Some authors felt that the SAF model was still appropriate for CMOS ICs [9-13]. Other investigators began to discuss certain CMOS IC failure modes that could not be accurately modeled as stuck-at faults [14-17]. Some of these defects and fault conditions include stuck-open faults [18-20], floating transistor gates [8,21,22], leakage [23,24], gate shorts [7], and delay faults [25].

As a result, it is now common to find publications on CMOS IC testing that seem to present an inconsistent, two-sided approach. On one hand the inability of the stuck-at fault model to represent the effects of common physical defects and failure mechanisms in CMOS ICs is mentioned, but then the stuck-at fault model is used for test generation and test quality measurement. Relatively few papers even attempt to quantify test coverage against any metric other than SAF coverage. Exceptions include the work of Teixeira et al who evaluated test coverage in terms of fault types closely linked to defects [26].

CMOS IC physical defects and the SAF Model

SAF coverage is the most widely recognized metric for test quality in industry. The recently released RADC MIL-I-38535 military specification for Qualified Manufacturer Listing (QML) requires a design-for-testability (DFT) methodology that can provide $\geq$ 99% SAF coverage [27].

When the "classical" SAF model is applied to CMOS ICs, the relationship of stuck-at faults to physical defects and failure mechanisms is treated in various ways. The following method has often been used [18,19]. The number of SAFs for an n-input combinational logic gate is reduced from the full $2n+2$ faults (one SA0 and SA1 fault for each input and for the one output) to $n+2$ "distinct" faults (one SAF for each input and both SAFs for the output). The one SAF eliminated for each input is the "dominant" state (0 for AND/NAND gates and 1 for OR/NOR gates) which is defined to be logically indistinguishable from one of the output SAF states. The output SA0 and SA1 faults, which now include the effect of the dominant SAF for each input, are said to correspond to low impedance short circuits to V_{SS} or V_{DD}, respectively. The n non-dominant input SAFs are sometimes called "input open from" (IOF) faults, which are defined as open inputs to the logic gate. Note that these are logic gate open circuits; i.e., the gate input is open-circuited to both the p-channel and the n-channel transistor in the gate. This "classical" approach therefore defines a CMOS OR/NOR gate open input as a SA0 and a CMOS AND/NAND gate open input as a SA1, in the same manner as previously discussed for bipolar logic.

Sandia National Laboratories has designed, fabricated, and tested high reliability CMOS ICs for over 15 years for military and space applications. Because of this, Sandia developed a uniquely thorough and sophisticated IC failure analysis capability. It has been observed that very few CMOS IC failures are caused by defects or failure mechanisms whose electrical effects are accurately represented by the stuck-at fault model [8,28]. Specifically, only a few failures have occurred due to logic element inputs or outputs which were permanently SA0 or SA1. Most failures have nonlinear characteristics not well modeled even by analog simulators like SPICE. Therefore fault modeling based on "SPICE-like" simulation is clearly questionable for many defects.

This apparent lack of correlation between typical CMOS IC defect behavior and the SAF model clearly brings into question the third perceived advantage of the SAF model, its "proven" effectiveness as a measure of test quality. For example, CMOS circuits with open inputs act differently than previously shown for the TTL 2-NAND in Fig. 1 (b). Fig. 2 shows a CMOS 2-NAND gate with input B open-circuited due to a defect. The input of a CMOS gate is voltage sensitive not current sensitive. It has been reported that this type of defect does not behave as a stuck-at fault in most situations but instead behaves more as a delay fault [8,21,22]. This is contrary to the "classical" input SAF assumption for open circuits.

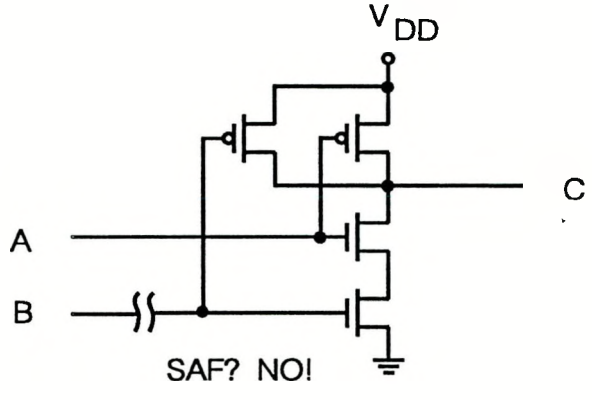

Fig. 2. CMOS 2-NAND with open circuit in input B.

ICs with open-circuited CMOS gates have been observed to operate at MHz frequencies [8,21,22,29]. Some open-circuit defects in CMOS circuits analyzed at Sandia have exhibited the effect of DC clamping to one of the power rails, but most often there is significant AC signal coupling that dominates any DC influence on such a gate.

In addition, open circuits in the source or drain interconnections of the individual p-channel or n-channel transistors have been widely acknowledged to produce electrical behavior not consistent with the SAF model [18-20]. These "nonclassical" effects include the

temporary retention of the previous logic state due to transistor source or drain open circuits, which is a behavior represented by the stuck-open fault model.

This experience and information leads to the conclusion that only CMOS IC defects and failure mechanisms whose electrical effect is a permanent, low impedance conducting path from a circuit node to V_{SS} (V_{DD}) can be modeled accurately as a SA0 (SA1) fault. This is shown in Fig. 3. The shunt path can be due to several types of defects, including metallization bridges to the power supply or parasitic transistors. To produce a SAF, the effective resistance of this shunt path must be low enough to overcome the lowest "on" resistance of the "on" p-channel or n-channel network. The shunt path shown in the figure implies that the responsible defects in effect "add" new circuitry. Circuit malfunctions that cause the "off" network to stay on, such as transistors which do not turn off ("stuck-on" faults), would not necessarily produce SAFs. They would produce output voltage levels dependent upon the drive strengths of the competing transistors, similar to bridged-node circuits [8].

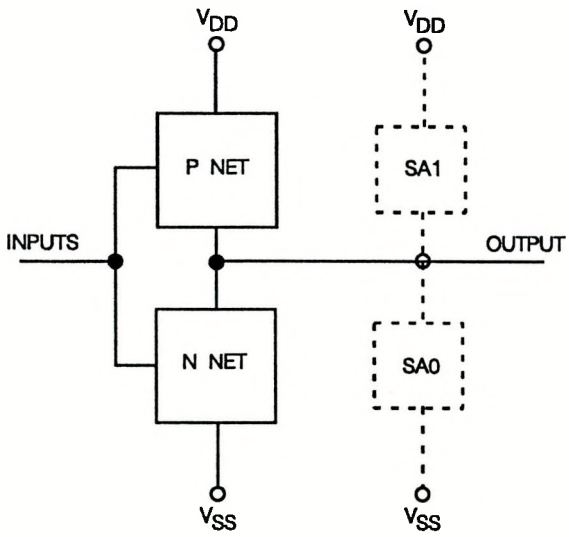

Fig. 3. CMOS circuit with SA0 or SA1 fault.

The significance of these open-circuit properties for CMOS inputs is that automatic test pattern generation adds no measurable detection value by attempting to generate SAF tests for individual branch lines of fanout networks. If a defect is assumed to cause an input to a logic gate to be open (not driven), then the test strategy must target open-circuit behavior, not SAF behavior. The appropriate test strategy to detect a floating node on a CMOS gate input may be to treat it as a delay fault. A SAF test may detect an open circuit defect, but it is not a guaranteed or quantified test strategy. In some design and topological situations, the open circuit input defect also can be detected by an I_{DDQ} test, but detection with I_{DDQ} tests cannot be guaranteed for every case [20].

The perception that SAFs can be used independent of technology to provide good defect detection has been clearly shown not to be the case for CMOS technologies [8]. Even so, significant reluctance to move on to better fault methodologies exists. Arguments for retaining the SAF model range from acceptance of SAF coverage as the only "universal" test metric to the desire to retain the use of SAF-based tools that companies have spent millions of dollars and uncountable man-years developing.

The prior investment in SAF-based tools is not necessarily lost. An issue with computer-modeling is the ability of the tool to summarize accurately the behavior of defect types such that detection can be assured. SAF-based tools may still be of significant benefit if they can be shown to provide useful defect detection information. For example, a sequential automatic test pattern generation program was slightly modified to provide more efficient test pattern generation for defect detection [30]. Also, for many common defects, the requirement for fault simulation can be reduced to "good" circuit logic simulation.

While the I_{DDQ} test has been shown to be a very sensitive detection method for CMOS IC defects, it is also very efficient for detecting CMOS SAFs in those situations where customers demand this test metric [30,31]. Data show that the number of I_{DDQ} test vectors required for 100% SAF detection can be reduced by factors up to 100 or more compared to conventional SAF test sets with comparable SAF coverage [24,30].

Fault Models, Defects, and Quality Levels

Williams developed a relation (Eqn. (1)) for the test escape percentage of defective ICs as a function of process yield (Y) and test coverage (T) [32]. He used the stuck-at fault as the measure of test coverage and labelled the percentage of defective ICs escaping to the customer as defect level (DL). This simple relation has had a large influence in establishing SAF coverage as the

test metric and has led to demand for near-100% SAF coverage by customers including RADC and others [33]. Some companies are putting significant resources into raising SAF coverage above 99%. There are features of this model that should be reviewed regarding its application to CMOS circuits. Equation (1) predicts unacceptable defect levels if test coverage is not close to 100%.

$$DL = 1 - Y^{(1-T)} \quad (1)$$

A fundamental improvement in the model occurs if SAF test coverage is not linked to the word "defect level." Data presented by many papers and a major point of this paper is that very few physical defects produce SAF behavior, so resources being applied to raising SAF coverage would be better applied to improving physical defect coverage. A production test set on a CMOS IC could have 100% SAF coverage but still pass significant defects on to customers. Gate oxide shorts, open-circuits, and bridged nodes are just three examples of defects whose detection by SAF test patterns is accidental [8]. An improvement in defect level modeling occurs if we do not assume a one-to-one mapping between SAFs and physical defects (an implicit assumption in past usage of the model). A 100% SAF coverage test guarantees only that the tested ICs have zero "detectable" SAFs, not zero physical defects.

The model retains much of its original intent if we redefine test coverage to be the true physical defect coverage. Teixeira et al provide an example of defining test coverage in terms of fault types that are closely linked to physical defects (see Fig. 5 in [26]). Fig. 4 shows how we should view the model.

The assumptions under which (1) was derived still have some validity if test coverage now means defect coverage. The basic assumption is that if defects are random and test coverage is incomplete, then a non-zero defect level has to exist. That basic assumption is as true for defects as it is for faults.

An improvement in the accuracy of the model was derived by R. Williams and Hawkins where tester error that fails good parts was taken into account [34]. This tester error is called α–error or Type I error and (1) is modified in the denominator to

$$DL = (1 - Y^{(1-T)})/(1 - \alpha Y^{(1-T)}) \quad (2)$$

where α is the probability that a tester will erroneously fail a good part. For small values of α, (2) reverts to (1).

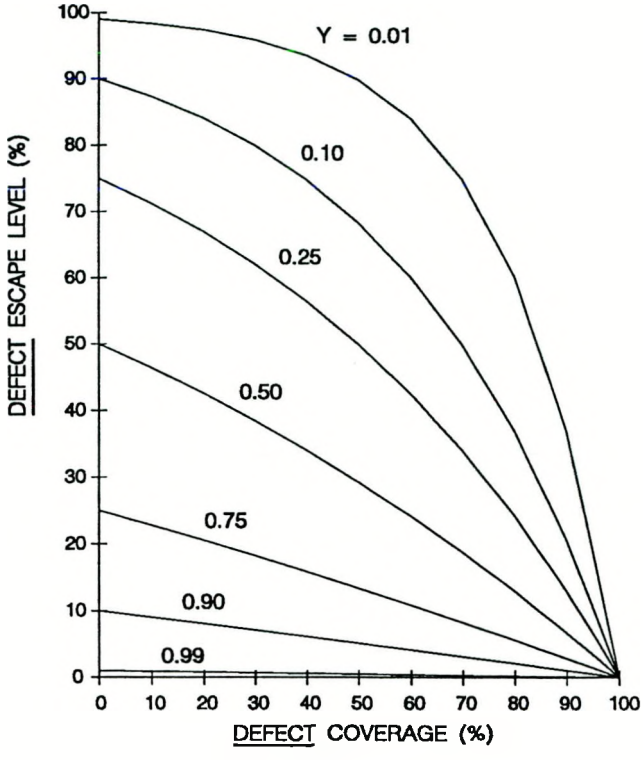

Fig. 4. Defect coverage versus defect escape level.

It has also been observed that defects often occur in clusters and therefore do not fit the assumption of random defect occurrence.

What is the impact of these factors on the fault model? The use of SAFs instead of real defect coverage for the "T" parameter means that the true defect level is higher than that predicted by SAF coverage. The inclusion of the tester α-error also would raise the true defect level higher than that predicted by (1). The tendency of defects to cluster on wafers and die probably would lower the defect level predicted by (1). There is little data to quantify the magnitude of the effect of these variables.

The validity of the model has been difficult to assess. The "Kokomo" experiments provided supportive information on the relation between increased SAF test coverage and the increased number of SAF test failures [33,35]. In fact, real defect levels in those experiments would have been higher than the SAF escape levels cited since true defect levels were not quantitatively evaluated.

Manufacturers have stated privately that they don't experience the unacceptable defect levels predicted by (1), even on ICs that don't have design-for-testability and use empirically-derived test sets. However, it has been noted that many manufacturers' test sets evolve significantly as a result of customer failures due to insufficient test coverage. Unfortunately, many failures at the next stage of assembly or in the "field" are often simply thrown away rather than returned to the vendor. The manufacturers in this case are "rewarded" for selling defective ICs. A mitigating factor not included in (1) is that the vendor's parametric tests may detect many defects that might otherwise escape a SAF test set [36]. It has also been suggested that customers often do not use the full functional capability of an IC and therefore may sometimes not encounter logic failures in their application.

Testing Strategy for Defect Detection

An effective test strategy for CMOS IC technologies must be based on physical defect coverage rather than SAF coverage. The goal is to use the most efficient means possible to achieve 100% physical defect coverage. The best method currently available begins with the development of a baseline test set that provides 100% node state coverage. Such a test set, when coupled with I_{DDQ} current measurement, appears to provide the most comprehensive coverage of physical defects. A 100% node state test utilizing I_{DDQ} current measurement provides excellent defect coverage for many common CMOS IC physical defects [30]. This type of test is efficient not only because it provides a quantitative measure of coverage for common physical defect types, but is also significantly easier to generate automatically and typically has a test set much smaller than an equivalent conventional single SAF test set.

There is, however, more than one way to acquire such a baseline test set. Mao et al developed and demonstrated a tool (QUIETEST) for identifying and selecting small subsets of vectors within large functional test sets that provide the same node state coverage as the entire functional test set [24]. Given that functional test sets, or other types of test sets, will be used as part of the overall test process, this method provides an alternative to economically identify I_{DDQ} test vectors and provide the initial coverage of physical defects. The test procedure used with the QUIETEST methodology runs the functional test set at normal speed and slows the test application rate down only for selected I_{DDQ} vectors in order to allow the necessary I_{DDQ} measurement. If 100% node state coverage is not provided by the functional test set then additional test generation is necessary to bring the baseline defect coverage up to the desired level.

The existence of a baseline test set allows the next logical step which is to evaluate actual defect coverage. "Fault" simulation must be performed for defect-types not guaranteed to be completely detected by a 100% node state test utilizing I_{DDQ} measurement [30]. Maximum benefit is thus drawn from the baseline test set. After determining defect coverage, additional test generation efforts can specifically target the remaining defect sites to increase overall defect coverage.

Improving the test coverage of bridging faults, for example, could be achieved with the following test development plan. Given a list of physically likely bridge-pair nodes obtained from layout information, one would first simulate the circuit using the baseline node state test set. A simple means for tracking the coverage of bridge-pair nodes, assuming the use of I_{DDQ} current monitoring, would be to add an XOR gate with inputs connected to the nodes of each bridge-pair. If, during the ensuing logic simulation, the output of an XOR gate is driven to logic 1, then the associated bridge-pair would be identified as detected. Detection is assured since an XOR output of 1 denotes that the two nodes of the bridge-pair have been set to opposite logic states. During test, when the nodes of each bridge-pair are set to opposite logic states, a bridging defect will be detected by monitoring I_{DDQ} for an increase due to the logic contention. After simulating the baseline test set to measure its effectiveness at detecting bridge-pairs, all XOR gates that have been detected would be removed and then the resulting modified circuit could be run through node-state test generation which would be specifically targeted at setting the remaining XOR gate outputs to logic 1.

For defects that behave exclusively as delay faults (without increasing I_{DDQ}), generation of a test that can provide 100% coverage is an extremely difficult problem. If one could limit the delay faults of concern to only critical paths of a circuit, then one could postulate that maximum speed tests along with AC parametric measurements could provide significant fault coverage. Unfortunately, delay faults can occur in any path within the circuit. Test generation alone is probably incapable of producing such a test even for moderately complex sequential circuits. This is one area where design-for-testability can play a significant role in reducing the effective complexity of designs so that existing or soon-to-exist tools will be able to provide the needed analysis and physical defect coverage.

Summary

Current-switching technologies, and in particular TTL, have been shown to have defects that predominantly can be mapped into two kinds of faults [6]. These are the stuck-at fault and the delay fault. However, for CMOS technologies the mapping of defects into fault models is much more complex. A majority of common CMOS IC defects produce increased quiescent power supply current and could therefore be considered "leakage" faults. To quantify the test coverage of defects which cause leakage faults, the use of I_{DDQ} testing has been proposed. CMOS IC failure analysis has also shown that many physical defects cause bridged nodes. These bridging defects can be quantitatively detected with I_{DDQ} tests using appropriate test vectors. Other CMOS defects are most likely modeled appropriately only by the delay fault model.

Even though a few defects in CMOS technologies can behave like SAFs, it is clear that the vast majority of common CMOS defects are not properly modeled by the SAF model. For instance, data from one report show SAF test escape rates of approximately 40% for gate oxide shorts [7]. Therefore SAF test coverage is not an appropriate test metric to be used for ensuring high quality CMOS products. I_{DDQ}-based testing clearly provides an efficient means of detecting many common CMOS defects, but even this method does not presently provide a quantifiable test metric for all types of CMOS physical defects.

Acknowledgement

We would like to express our gratitude to the following people for their review and contributions to this paper: Ed Cole (Sandia Div. 2142) and Scott Davidson (AT&T Bell Laboratories Engineering Research Center). We also would like to thank Ravi Gulati (Ford Microelectronics) for his helpful conversations. This work was performed at Sandia National Laboratories and supported by the U. S. Department of Energy under contract DE-AC-04-76DP00789.

References

[1] R. Eldred, "Test Routines based on symbolic logical statements," J. of Assoc. for Computing Mach. (ACM), vol. 6, pp. 33-36, 1959.

[2] J. Poage, "Derivation of optimum tests to detect faults in combinational circuits," Proc. Symp. on Math. Theory of Automata, Brooklyn, N. Y.: Polytechnic Press, pp. 483-528, 1963.

[3] J. Galey, R. Norby, and J. Roth, "Techniques for the diagnosis of switching circuit failures", IEEE Trans. on Comm. and Electron., vol. 83, no. 74, pp. 509-514, Sept. 1964.

[4] D. Armstrong, "On finding a nearly minimal set of fault detection tests for combinational logic nets," IEEE Trans. on Electron. Computers, vol. EC-15, no. 1, pp. 66-73, Feb. 1966.

[5] J. Millman and H. Taub, *Pulse, Digital, and Switching Waveforms*, Mcgraw-Hill Inc., Chapter 9: Logic Circuits, pp. 306-361, 1965.

[6] C. Beh, K. Arya, C. Radke, and K. Torku, "Do stuck fault models reflect manufacturing defects?" Proc. Int. Test Conf., pp. 35-42, 1982.

[7] C. Hawkins and J. Soden, "Electrical characteristics and testing considerations for gate oxide shorts in CMOS ICs," Proc. Int. Test Conf., pp. 544-555, Nov. 1985.

[8] J. Soden and C. Hawkins, "Electrical properties and detection methods for CMOS IC defects," Proc. European Test Conf., pp. 159-167, Paris, France, April 1989.

[9] V. Agrawal and S. Seth, "Test Generation for VLSI Chips," Computer Society Press, Chapter 1: Introduction, pp. 1-5, 1988.

[10] D. Edwards, "Testing for MOS integrated circuits failure modes," Proc. Int. Test Conf., pp. 407-416, 1980.

[11] C. Timoc, M. Buehler, T. Griswold, C. Pina, F. Stott, and L. Hess, "Logical models of physical failures," Proc. Int. Test Conf., pp. 546-553, 1983.

[12] T. Williams and N. Brown, "Defect level as a function of fault coverage," IEEE Trans. Computers, pp. 987-988, Dec. 1981.

[13] M. Turner, D. Leet, R. Prilik, and D. McLean, "Testing CMOS VLSI: tools, concepts, and experimental results," Proc. Int. Test Conf., pp. 322-328. Nov. 1985.

[14] P. Banerjee and J. Abraham, "Characterization and testing of physical failures in MOS logic circuits," IEEE Design and Test, vol. 1, pp. 76-86. Aug. 1984.

[15] N. Burgess and R. Damper, "The inadequacy of the stuck-at fault model for testing MOS LSI circuits: a review of MOS failure mechanisms and some implications for computer-aided design and test of MOS LSI circuits," Software & Microsystems, vol. 3, no. 2, pp. 30-36, April 1984.

[16] J. Zasio, "Non stuck fault testing of CMOS VLSI," Proc. 13th IEEE Computer Society Int. Conf. (Compcon), pp. 388-391, Spring, 1985.

[17] J. Galiay, Y. Crouzet, and M. Vergniault, "Physical versus logical fault models MOS LSI Circuits: Impact on their testability," IEEE Trans. on Computers, vol. C-29, no. 6, pp. 527-531, June 1980.

[18] G. Case, "Analysis of Actual Fault Mechanisms in CMOS Logic Gates," Proc. Des. Auto. Conf., pp. 265-270, 1976.

[19] R. Wadsack, "Fault modeling and logic simulation of CMOS and MOS integrated circuits," Bell Sys. Tech. J., vol. 57, pp. 1449-1488, May-June 1978.

[20] J. Soden, R. Treece, M. Taylor and C. Hawkins, "CMOS IC stuck-open fault electrical effects and design considerations," Proc. Int. Test Conf., pp. 423-430, Aug. 1989.

[21] M. Renovell and G. Cambon, "Topology dependence of floating gate faults in MOS circuits," Electron. Letters, vol. 22, no. 3, pp. 152-153, Jan. 1986.

[22] W. Maly, P. Nag, and P. Nigh, "Testing oriented analysis of CMOS ICs with opens," Proc. Int. Conf. Comp. Aid. Des. (ICCAD), pp. 344-347, Nov. 1988.

[23] C. Hawkins, J. Soden, R. Fritzemeier, and L. Horning, "Quiescent power supply current measurement for CMOS IC defect detection," IEEE Trans. on Indus. Electron., vol. 36, no. 2, pp. 211-218, 1989.

[24] W. Mao, R. Gulati, D. Goel, and M. Ciletti, "QUIETEST: A quiescent current testing methodology for detecting leakage faults," Proc. ICCAD, 1990.

[25] J. Carter, V. Iyengar, and B. Rosen, "Efficient Test Coverage for Delay Faults," Proc. Int. Test Conf., pp. 418-427, Sept. 1987.

[26] J. Teixeira, I. Teixeira, C. Almeida, F. Goncalves, and J. Goncalves, "A methodology for testability enhancement at layout level," J. of Electron. Testing: Theory and Applications (JETTA), accepted for publication on Sept. 17, 1990.

[27] MIL-I-38535, "General Specification for Integrated Circuits (Microcircuits) Manufacturing," 18 Dec. 1989. Available from the Naval Publications and Forms Center, (ATTN: NPODS), 5801 Tabor Avenue, Philadelphia, PA 19120-5099.

[28] C. C. Timoc, Personal communication, January, 1991.

[29] C. Henderson, "The behavior of open circuits in VLSI applications," Univ. of New Mexico MSEE thesis, 1990.

[30] R. Fritzemeier, J. Soden, K. Treece, and C. Hawkins, "Increased CMOS IC stuck-at fault coverage with reduced I_{DDQ} test sets," Proc. Int. Test Conf., pp. 427-435, 1990.

[31] Y. Malaiya and S. Su, "A new fault model and testing techniques for CMOS Devices," Proc. Int. Test Conf., pp. 25-34, Nov. 1982.

[32] T. Williams, "Sufficient testing in a self-testing Environment," Proc. Int. Test Conf., pp. 167-172, 1984.

[33] E. McCluskey and F. Buelow, "IC quality and test transparency," IEEE Trans. Indust. Electron., vol. 36, no.2, pp. 197-202, May 1989.

[34] R. Williams and C. Hawkins, "Errors in Testing," Proc. Int. Test Conf., pp. 1018-1027, 1990.

[35] R. Daniels and W. Bruce, "Built-in self-test trends in Motorola Microprocessors," IEEE Design & Test, pp. 64-71, April, 1985.

[36] K. Baker, Personal communication, Philips Research, Nov. 1990.