Some Faults Need an I_{ddq} Test

Samy R. Makar* and Edward J. McCluskey

Center for Reliable Computing
Gates Hall 2A
Stanford University
Stanford, CA 94305

Abstract

Fault Simulation results of different implementations of 2-1 multiplexers and D-latches are presented. These results show that some faults can only be detected by I_{ddq} test. Simulation results also show that the "importance" of I_{ddq} as a test method can vary considerably with implementation.

Introduction

I_{ddq} testing has been used in the last several years as a technique to improve quality of CMOS chips [2], [9], and [11]. In CMOS, there are some faults whose presence does not change the functionality of the circuit under test. Some of these cannot be detected (and thus are untestable or redundant). Others that cannot be detected by a Boolean voltage test (since the circuit functionality is correct) can, nevertheless, be discovered by a current test or a delay test [4], [6]. There are special difficulties in using I_{ddq} testing: determining the I_{ddq} threshold, ensuring that the design adheres to I_{ddq} design rules (so that I_{ddq} current of fault-free circuits is low enough to be differentiated from I_{ddq} of faulty circuits), and the increase in fault-free I_{ddq} values with the shrinking of circuit technology. In this work we examine different implementations of the same circuit to determine how "important" I_{ddq} is for each of the implementations. Importance of I_{ddq} for a circuit implementation is measured by the percentage of faults that are only detected by an I_{ddq} test. We show that regardless of the implementation, some faults can only be detected by an I_{ddq} test. However, some implementations have a smaller percentage of faults that can only be detected by an I_{ddq} test. In other words, if we cannot use an I_{ddq} test, then we will miss fewer faults with such implementations.

In this paper we analyze three different implementations of 2-to-1 multiplexers and three different implementations of D-latches. Multiplexers and D-latches are interesting because there are many different ways of implementing them, and are common in many designs. Our analysis shows that the percentage of faults that require I_{ddq} for detection vary considerably with different implementations.

In our analysis we perform fault simulation using the CrossCheck fault models [1], [12]. The fault models comprise shorted interconnects, open interconnects, short-to-power, short-to-ground, transistor stuck-on, and transistor stuck-open. In the simulations, faults are injected by modifying a copy of the circuit description. The faulty circuits are simulated using HSpice [5].

The current limit for I_{ddq} testing is often determined experimentally, by measuring the Iddq values of many good and bad dies, and selecting an appropriate threshold that would detect as many faulty circuits as possible without discarding many good ones [3], [10]. For our simulations, the current limit is determined by first measuring the maximum observed current for the fault-free circuit and for each faulty circuit (circuit with fault injected). An appropriate threshold is selected based on these numbers.

Multiplexer

A 2-to-1 multiplexer is shown in Fig. 1. Three different 2-to-1 multiplexer implementations (transmission gate, AOI, NAND/NAND) were simulated [8]. Two different tests were used in the simulations: an exhaustive test, which contains 8 patterns, and a test that applies all single bit transitions which contains 25 patterns (see Table 1). Each test was run twice, once with a cycle time (*cycle time* here is defined to be the time between the application of inputs) of 100 ns, and once with a cycle time of 10 ms. Outputs were measured just before applying the next input. The 100 ns cycle time is a typical test time for a boolean test, and the 10 ms cycle time is needed to allow I_{dd} to settle to its quiescent value. The results of both tests are

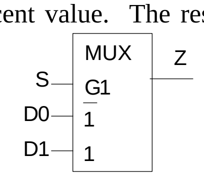

Figure 1 2-to-1 Multiplexer.

Table 1a Exhaustive Test For Multiplexer.

S	0	1	1	0	0	1	1	0
D0	0	0	1	1	0	0	1	1
D1	0	0	0	0	1	1	1	1
Z	0	0	0	1	1	1	1	0

Table 1b Single Transition Bit Exhaustive Test.

S	0 1 1 0 0 1 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 1 0
D0	0 0 1 1 1 1 0 0 1 0 0 0 0 0 0 1 1 1 1 1 0 1 1 0 0
D1	0 0 0 0 1 1 1 1 1 1 0 1 1 0 1 1 0 1 1 0 0 0 0 0 0
Z	0 0 0 1 1 1 1 0 1 0 0 0 1 0 1 1 0 1 1 1 0 1 0 0 0

*Samy Makar is presently with Cirrus Logic Inc., Fremont CA. 94536 shown in Tables 2 and 3. In these tables several numbers are reported: the total number of faults injected, the number of faults detected when both a boolean and an I_{ddq} test are done, the number of faults detected when only a boolean test is done, the number of faults detected when only an I_{ddq} test is done, and the number of faults detected by an I_{ddq} test but missed by a Boolean test. These tables show that a significant percentage of faults are missed if no I_{ddq} test is done. The AOI implementation (b) with the exhaustive transition test had the lowest percentage of faults (Table 3) detected by I_{ddq} missed by boolean test.

Table 2 Number of Faults for Test in Table 1a.

Impl	Total Faults	Det by Boolean & Iddq	Det by Boolean Alone	Det by Iddq Alone	Not detectable	Detected by Iddq not by Boolean
a	75	69	47	65	6	22 (31.9%)
b	102	102	78	98	0	24 (23.5%)
c	121	121	95	110	0	26 (21.4%)

a=Transmission gate, b=AOI, c=NAND/NAND

Table 3 Number of Faults for Test in Table 1b.

Impl	Total Faults	Det by Boolean & Iddq	Det by Boolean Alone	Det by Iddq Alone	Not detectable	Detected by Iddq not by Boolean
a	75	69	47	65	6	22 (31.9%)
b	102	102	86	98	0	16 (15.7%)
c	121	121	100	115	0	21 (17.3%)

a=Transmission gate, b=AOI, c=NAND/NAND

D-Latch

The D-latch circuit is shown in Fig. 2. Three different D-latch implementations were simulated. These are shown in Fig. 4. The test used here, shown in Table 4, is a checking experiment from [7]. A checking experiment guarantees the detection of all faults that do not increase the number of states in the circuit. In [7], we only presented and analyzed results for the first implementation, and showed that the test detects all detectable faults in the circuit. The other two implementations use the second and third multiplexer implementations described in [8]. The output of the multiplexer is connected to the D0 input and renamed Q, the S input is renamed E (for enable), and the D1 input is renamed D. The results of the simulations are shown in Table 5. As with the multiplexers, Table 5 shows that a large number of faults are missed if no I_{ddq} test is done. The AOI implementation (b) has the lowest percentage of faults that are detected only by I_{ddq} measurement.

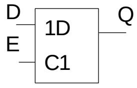

Figure 2 D-Latch Circuit.

Table 4 Checking Experiment for D-Latch.

E	1	0	1	0	1	0	0	1	0	1	0	0	1	0
D	0	0	0	0	0	0	1	1	1	1	1	0	0	0
Q	0	0	0	0	0	0	0	1	1	1	1	1	0	0

Table 5 Number of Faults for Test in Table 4.

Impl	Total Faults	Det by Boolean & Iddq	Det by Boolean Alone	Det by Iddq Alone	Not detectable	Detected by Iddq not by Boolean
a	67	66	47	61	1	19 (28.4%)
b	92	92	86	56	0	6 (6.5%)
c	110	110	100	72	0	10 (9.1%)

a=Transmission gate, b=AOI, c=NAND/NAND

Conclusion

Simulation results presented in this paper show that a large number of faults can be missed if no I_{ddq} test is performed. Even though these faults may not affect the functionality of the circuit, they may have an impact on early life failure or timing.

Acknowledgments

The authors thank Jonathan Chang and Siyad Ma for their valuable comments. This work was supported in part by the Ballistic Missile Defense Organization, Innovative Science and Technology (BMDO/IST) Directorate and administered through the Department of the Navy, Office of Naval Research under Grant No. N00014-92-J-1782, in part by the Advanced Research Projects Agency under Contract No. DABT63-94-C-0045, and in part by the National Science Foundation under Grant No. MIP-9107760.

References

[1] Chandra, S., et. al., "CrossCheck: An Innovative Testability Solution," IEEE Design and Test of Computers, Vol. 10, No. 2, pp. 56-67, June, 1993.

[2] Wiscombe , P.C. "A Comparison of Stuck-at Fault Coverage and I_{ddq} Testing on Defect Levels," *Proceedings ITC*, pp. 293-299, 1993.

[3] Hawkins, CF., "Quiescent Power Supply Current Measurement for CMOS IC Defect Detection," *IEEE Trans. on Industrial Electronics*, pp. 211-218, May, 1989.

[4] Hawkins, CF., et. al, "Defect Classes - An Overdue Paradigm for CMOS IC Testing," *Proceeding ITC*, pp. 413-425, 1994.

[5] Kielkowski, R., *Inside SPICE Overcoming the Obstacles of Circuit Simulation*, McGraw Hill, USA, 1994.

[6] Ma, S., P. Franco and E.J. McCluskey, "An Experimental Chip to Evaluated Test Techniques, Experimental Results," *Proceedings ITC*, pp. 663-672, 1995.

[7] Makar, S.R. and E.J. McCluskey, "Functional Tests for Scan Chain Latches," *Proceedings ITC*, pp. 606-615, 1995.

[8] Makar, S.R. and E.J. McCluskey, "Some Faults Need an I_{ddq} Test," Stanford CRC Technical Note 96-1, 1996.

[9] Maxwell, P.C. and R.C. Aitken, V. Johansen and I. Chiang, "The Effectiveness of Iddq, Functional and Scan Tests: How Many Fault Coverages do We Need?," *Proceedings ITC*, pp. 168-177, 1992.

[10] Perry, R., "I_{ddq} Testing in CMOS Digital ASICs - Putting It All Together," *Proceedings ITC*, pp. 151-157, 1992.

[11] Sawada, K. and S. Kayano, "An Evaluation of Iddq Versus Conventional Testing for CMOS Sea-of-Gates IC's," *Proceedings ITC*, pp. 158-167, 1992.

[12] Sucar, H., "High Performance Test Generation for Accurate Defect Models in CMOS Gate Array Technology," *ICCAD*, pp. 166-169, 1989.