Test Sets for Internal Access Test Methods

V.N. Yarmolik, Y.V. Bykov
Computer Systems Department,
Belarussian State University
of Informatics and Radioelectronics,
BELARUS 220027, Minsk, P. Brovki 6,
e-mail: yarmolik%csd.bsuir.minsk.by@brc.minsk.by

M.G. Karpovsky
Department of Electrical, Computer and System Engineering, Boston University, 44 Cummington Str., Boston, Massachusetts, 02215 USA
e-mail: mr@buenga.bu.edu

Abstract: This paper presents test generation approach for internal access test methods providing massive observability (I*ddq*, CrossCheck, Electron Beam, Bed-of-Nails and Guided Probe Test Approaches).

1 Introduction

As electronic systems technologies become more complex, test development and design for testability issues become increasingly critical. With the increasing complexity of VLSI circuits, new techniques are required to ensure IC quality and reliability. This trend is especially true in the current environment with its rising quality standards and unabated time-to-market pressures.

Standard test approaches for the detection of faults (for example, stuck-at and bridging faults) require two condition to occur: *sensitization and propagation.* These conditions traditionally are measured by the circuit's characteristics to control (Controllability) and observe (Observability) faulty nodes [1]. Additional logic usually may be added to increase both controllability and observability of deeply buried logic elements.

New test techniques have recently been developed that provide visibility, either virtually or directly, to the logic node or even the electrical node level, which ensure massive obsevability in a circuit-under-test. Such techniques are reffered to as *internal access test methods* (I*ddq*, CrossCheck, Electron Beam). All of these techniques guarantee a good testability for the circuits with a good controllability.

One of the most attractive technique among them is monitoring of the quiescent power supply current (I*ddq*-testing) for static CMOS circuit. In the following sections we will consider the only Iddq-testing since the results may be easily modified for any other internal access test method.

The main idea of I*ddq*-testing is based on the propeties of the CMOS circiuts. In general, a complex CMOS gate has an nMOS pull down part with complementary pMOS pull up part. In steady state, only one part, either nMOS or pMOS, conducts. In switch level modeling each transistor is considered as a switch and it takes one of the two possible states (open or closed). An open switch means that the transistor is not conducting, while a closed switch means that it is conducting, offering a very low resistance path.

Normally, the leakage current of CMOS circiut under a quiescent state is very small and negligible. When a failure occurs and causes a conducting current path from power (V*dd*) to ground (V*ss*), it may draw an excessive supply current. The measurement of the quiescent power supply current to detect leakage faults is reffered to as I*ddq* which allows to detect the defects in CMOS circuits, such as: gate oxid shorts; bridging defects; punchthrough; parasitic transistor leakage; open drain or source and transmission gate opens [2].

With I*ddq* technique, observation is at the power supply current terminal that is connected to each of macro cells. Therefore an increase in I*ddq* indicates the presence of a detect in a macro cells for a given test pattern. This is the main advantage of I*ddq* testing which allows to get "automatic observability". This increase results in a rapid fault simulation and grading, because only fault sensitization and not propagation is necessary.

Iddq testing is slow. Typically, less than 1,00 test patterns can be applied. Thus the problem of generation of short tests is very important.

2 Fault Models

In this paper we will concentrate our investigation on the bridging faults (shorts) as a most common fault model covering physical defects in CMOS circuits. Bridging faults (BFs) between normally unconnected signal nodes, are common failures in CMOS circuits [3 - 5]. Studies examining realistic faults in CMOS digital circuits suggest that BFs may account from thirty to fifty percent of all faults [3].

It is shown [6] that a bridging fault may cause an anomalous output and create metastability in the circuit. Such fault cannot be detected by logic testing. To detect these faults, internal access test methods with high level of fault observability should be used [6].

Stuck-at faults (stuck-at-0, stuck-at-1) can be represented as BFs between the logic node and V*dd* and V*ss* nodes. Thus BFs between any two logic nodes, as well as between logic nodes and V*dd* and Vss nodes, can be considered as universal bridging faults, which cover all stuck-at faults, all BFs, [1], as well as several previously intractable or unmeasurable types of faults. These include logically redundant single stuck-at faults, multiple stuck-at faults and many common CMOS parametric failures that cannot be modeled by the standard fault models [7].

The number P of BFs for a circuit with N logic nodes is given by

$$P = 2N + \binom{N}{2} = (N^2 + 3N)/2 \qquad (1)$$

The conditions for bridge detectability by I*ddq* testing have been investigated by many authors and the most commonly used fault detectability criterion consists of controlling the bridged nodes to complementary logic values in the fault-free circuit [3, 4].

3 Test Matrices

A test with T test patterns for a given network with m inputs and N nodes we define as an $(T \times m)$ *input matrix, H_I,* where rows of H_I are test patterns and a $(T \times N)$ *test matrix, H,* where *i*th rows of H represents logical values at all nodes of the network for *i*th test pattern $(i = 1, \ldots, T)$.

The necessary and sufficient conditions for BFs detection can be formulated as following theorem.

Theorem 3.1 *All BFs of the circuit-under-test are detectable iff all columns in a test matrix H are different and not equal to all ones or all zeros, where a row in H represents an input test pattern and responses on internal and output nodes.*

The number of rows in test matrix H determines the I*ddq* test complexity. The following theorem shows the lower and upper bounds for the I*ddq* test complexity, T.

Theorem 3.2 *The Iddq test complexity, T, to detect all BFs for nonredundant circuit with m input, p internal and n output nodes, satisfies the following inequalities*

$$\lceil \log_2(m+p+n+2) \rceil \le T \le \lceil \log_2(m+2) \rceil + p + n, \qquad (2)$$

where $N = n + m + p$.

Proof. According to Theorem 3.1 any test pattern generation procedure should generate a test matrix with different columns which are not equal to all ones and all zeros. The minimal number of rows for a test matrix H to satisfy this condition equals to $\lceil \log_2(m+p+n+2) \rceil$ (Lower bound).

We note that $\lceil \log_2(m+2) \rceil$ rows are sufficient to generate m different columns for m input nodes, which are differ from all ones and all zeros columns. For any subset of S identical columns no more then S - 1 additional rows should be added to a test matrix to make these columns different. One extra row are sufficient to avoid all zeros and all ones columns in S. As result the upper bound will satisfy (2). Q.E.D.

We note that lower and upper bounds (2) given by Theorem 3.2 are attainable. The lower bound is attained for the case of one gate with N - 1 inputs and the upper bound for the case of two-level AND-OR networks with orthogonal product terms.

According to theorem 3.2 test complexity for detection of all $P = (N^2 + 3N)/2$ BFs satisfies the inequality $T \le N$.

4 Test Generation

4.1 Universal Test Sets

The fact that I*ddq* provides good observability implies that for every bridging faults (BF) the number of test vectors will be rather high. From this one is tempted to conclude that BFs can be effectively tested using a small test set.

Let us consider a universal test sets for input-output (IO) BFs. It is easily to show that universal test set for combinational circuit described by the set of boolean functions $F_j(x_1, x_2, \ldots, x_n)$, $j \in \{1, 2, \ldots, p\}$ consists of T_u test patterns where $\lceil \log_2(p+n+2) \rceil \le T_u \le \lceil \log_2(n+2) \rceil + p$. The first $q = \lceil \log_2(p+n+2) \rceil$ rows of test matrix H represents the set of 2^q - ary numbers in a binary representation, where a column x_1 is 1, a column x_2 equals 2 and a column x_n represents n. The columns $F_1, F_2, \ldots, F_p$ have the values depending on boolean functions $F_j(x_1, x_2, \ldots, x_n)$. If column F_i has a q - ary representation r, $r \in \{0, 1, 2, \ldots, n, q - 1\}$, then the test matrix H includes additional row generated by the test pattern $F_j(x_1, x_2, \ldots, x_n) \oplus x_r = 1$, where $x_0 = 0$ and $x_{q-1} = 1$.

Let us generate universal test set for the circuit described by $F = x_1x_2\overline{x_4}\overline{x_5} + x_1x_2x_3$. The first $q = \lceil \log_2(p+n+2) \rceil = \lceil \log_2(5+1+2) \rceil = 3$ rows represent 2^3 - ary numbers, where a column $x_1 = 1$, $x_2 = 2$, $x_3 = 3$, $x_4 = 4$ and $x_5 = 5$. As a result

$$H = \begin{vmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & F \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ - & - & - & - & - & - \\ 1 & 2 & 3 & 4 & 5 & 0 \end{vmatrix}$$

Column $F = 0$. Then, any solution of the equality $F \oplus 0 = 1$ has to be used as additional row to H (for example, 11100).

For $n = 2^k$ - bit adder universal test set consists of $k + 3$ patterns of $3 \cdot 2^k$ bits. Optimal universal test set for $n = 2^k$ - bit multiplier has a complexity $k + 3$. The tests for 8 - bit adder and multiplier are presented in Table 4.1.

$Z = X + Y$			$Z = X \cdot Y$		
X	Y	Z	X	Y	Z
00000000	00000000	00000000	1000	1110	01110000
11111111	00000000	11111111	0111	0001	00000111
00000000	11111111	11111111	0011	1100	00100100
00001111	00001111	00011110	0101	0110	00011110
00110011	00110011	01100110	1100	1011	10000100
01010101	01010101	10101010			

Table 4.1: Universal test set for adder and multiplier

4.2 Pseudoexhaustive Tests

As alternative for standard approaches for test generation there is pseudoexhaustive testing, what allow to reduce test complexity. In the case of internal access test methods pseudoexhaustive testing can be implemented as a generation of exhaustive BFs tests for

every segment (portion) of original circuit, such that the number of logical nodes of a segment significantly smaller than the number of nodes of the circuit.

As an example of a pseudoexhaustive test generation for the internal access test methods let us consider example of full adder (Fig. 4.1), where the smallest segments are

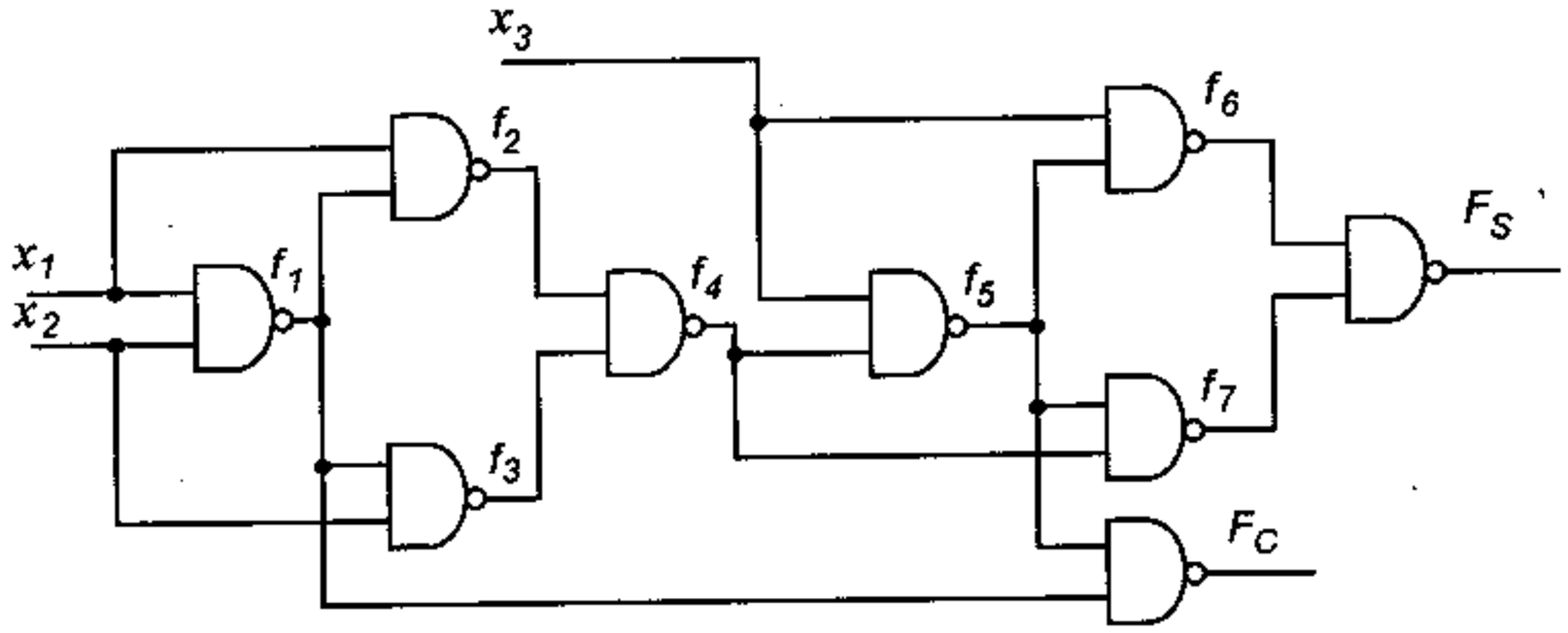

Fig. 4.1: Full adder

2NAND gates. A full list of the exhaustive tests with minimal complexties detecting all Bfs of 2NAND gate are presented in table 4.2. It should be mentioned that the 2NAND octal operation is commutative. To generate pseudoexhaustive test pattern for internal access test method any standard *Backtrace* algorithm [8] can be applied.

Procedure of test generation for the adder (Table 4.3) starts with the output logic node F_S. As a result we have got pseudoexhaustive test set $x_1\, x_2\, x_3 = 633$, what ensures exhaustive BFs tests for all 2NAND gates of full adder.

2NAND gate $f = \overline{x_1 x_2}$								
0·1 = 1	0·1 = 1	0·0 = 1	1·1 = 0	1·1 = 0	1·1 = 0	0·1 = 1	0·0 = 1	1·0 = 1
0·0 = 1	1·0 = 1	0·1 = 1	0·1 = 1	0·0 = 1	0·1 = 1	1·1 = 0	1·1 = 0	1·1 = 0
1·1 = 0	1·1 = 0	1·1 = 0	0·0 = 1	0·1 = 1	1·0 = 1	0·0 = 1	0·1 = 1	0·1 = 1
1·5 = 6	3·5 = 6	1·3 = 6	4·6 = 3	4·5 = 3	5·6 = 3	2·6 = 5	2·3 = 5	6·3 = 5

Table 4.2: Exhaustive tests and its octal notation for 2NAND gate

x_1	x_2	x_3	f_1	f_2	f_3	f_4	f_5	f_6	f_7	F_S	F_C
								3	5	6	
		3				5	6	3	5		
				3	6	5					
6	3		5	3	6						
			5								3

Table 4.3: Backtrace procedure for pseudoexhaustive test set generation for full adder.

4.3 Test Generation Procedure

General test generation procedure for internal access test methods consists of the construction of the test matrix H according the Theorem 3.1. This procedure can be formulated as an *Algorithm G* for given combinational circuit CC and a set of BFs. Let CC has an n input, m internal and p output logic nodes and N - tuple $t = t^0, t^1, t^2, \ldots, t^{N-1}$, is a set of 2^q - ary boolean function described all logic nodes, where $N = n + m + p$.

Algorithm G

Begin

$q = \lceil \log_2(N + 2) \rceil$

\#

Find the solutions S of the set of boolean 2^q - ary inequalities:
$t^i \neq 0$; $t^i \neq q - 1$; and $t^i \neq t^j$;
$i \neq j \in \{0, 1, \ldots, N\text{-}1\}$;

if S empty,
then $q=q+1$;
goto #; *else*;

End

For example, for 2NAND gate we have $q = \lceil \log_2(3 + 2) \rceil = 3$ and the set of boolean octal inequalities has a form

$x_1 \neq 0$; $x_1 \neq 7$; $x_2 \neq 0$; $x_1 \neq 7$; $x_1 \neq x_2$; $\overline{x_1 x_2} \neq 0$; $\overline{x_1 x_2} \neq 7$; $\overline{x_1 x_2} \neq x_1$; $\overline{x_1 x_2} \neq x_2$;
and their solution is $x_1 x_2 = \{13, 15, 23, 26, 35, 36, 45, 46, 56\}$.

It is easily to show that a computing tests for BFs is NP-hard. To reduce the complexity of a test generation we propose to use combine approaches, when during the first phase of test generation any standard tests, IO universal test sets, pseudoexhaustive tests or random tests can be applied, then at the second phase an *Algorithm G* can be used.

5 Summary

In this paper we have presented lower and upper bounds for I*ddq* test complexity for detection of brifging faults, method for design of IO universal test sets, pseudoexhaustive tests and general test procedure for BFs have been presented. These results can be easily adapted to any other internal access test method.

References

[1] K.M. Butler, R. Kapur, M.R. Mercer, D.E. Ross "The Roles of Controllability and Observability in Design for Test", Digest of papers IEEE VLSI Symposium, 1992, pp. 211-216.

[2] R.R Fritzemeier, J.M. Soden, R.K. Treece, C.F. Hawkins "Increased CMOS IC Stuck-at Fault Coverage with Reduced Iddq Test Sets", Proceedings International Test Conference 1990 pp. 427-434.

[3] Y. K. Malaiya, S.Y. H. Su "A New Fault Modeling and Testing Technique for CMOS Devices", Proceedings International Test Conference, 1982 , pp. 25 - 34.

[4] E. Isern, J. Figueras "Analysis of Iddq Detectable Bridges in Combinational CMOS Circuits", Proceedings International Test Conference, 1994, pp. 386 - 373.

[5] S. W. Bollinger, S.F. Midkoff "On Test Generation for Iddq Testing of Bridging Faults in CMOS Circuits", Proceedings International Test Conference, 1991, pp. 598 - 607.

[6] R. Rajsuman "An Analysis of Feedback Bridging Faults in MOS VLSI", Digest of papers IEEE VLSI Test Simposium, 1991, pp. 53 - 58.

[7] R. Rajsuman, Y.K. Malaiya, A.P. Jayasumana "On Accuracy of Switch-Level Modeling Of Bridging Faults in Complex Gates" Proceedings Design Automation Conference, 1987, pp. 244 - 250.

[8] M. Abramovici, M.A. Breuer, A.D. Friedman "Digital Testing and Testable Design", Computer Science Press, New York, 1990.