Delay Fault Testing: Choosing Between Random SIC and Random MIC Test Sequences

Arnaud Virazel, René M. G. David, Patrick Girard, Christian Landrault, Serge Pravossoudovitch

Delay Fault Testing: Choosing Between Random SIC and Random MIC Test Sequences

A. VIRAZEL
Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier Université Montpellier II/CNRS, 161 rue Ada, 34392 Montpellier Cedex 5, France
virazel@lirmm.fr

R. DAVID
Laboratoire d'Automatique de Grenoble, INPG–CNRS–UJF BP 46, 38402 St Martin-d'Hères, France
Rene.David@lag.ensieg.inpg.fr

P. GIRARD, C. LANDRAULT AND S. PRAVOSSOUDOVITCH
Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier Université Montpellier II/CNRS, 161 rue Ada, 34392 Montpellier Cedex 5, France
girard@lirmm.fr
landraul@lirmm.fr
pravo@lirmm.fr

Received August 1, 2000; Revised October 1, 2000

Editor: H.J. Wunderlich

Abstract. The combination of higher quality requirements and sensitivity of high performance circuits to delay defects has led to an increasing emphasis on delay testing of VLSI circuits. In this context, it has been proven that Single Input Change (SIC) test sequences are more effective than classical Multiple Input Change (MIC) test sequences when a high robust delay fault coverage is targeted. In this paper, we show that random SIC (RSIC) test sequences achieve a higher fault coverage than random MIC (RMIC) test sequences when both robust and non-robust tests are under consideration. Experimental results given in this paper are based on a software generation of RSIC test sequences that can be easily generated in this case. For a built-in self-test (BIST) purpose, hardware generated RSIC sequences have to be used. This kind of generation will be shortly discussed at the end of the paper.

Keywords: delay testing, random testing, robust test, non-robust test, BIST

1. Introduction

Delay fault testing allows to test for *delay faults*. A delay fault occurs in a circuit when one or more paths in the circuit fail to propagate a signal within the time interval specified by the clock period. Detection of delay faults requires *two-pattern tests*. An *initialization vector* is applied and the circuit is allowed to stabilize. Then, the *test vector* is applied and the circuit outputs are sampled at clock speed. The response is then compared to that of the fault-free circuit to determine the presence or the absence of a delay fault. Hence, correct operation of a circuit at the intended speed can only be guaranteed if there is no delay fault in the circuit.

With the continuous increase in the operating speed of VLSI circuits, delay fault testing is likely to become industrially accepted in the near future [14]. With delay fault testing (i.e. application of two-pattern tests to ensure temporal correctness of the design), at-speed testing (i.e. test at the intended operating speed of the circuit) is also becoming an essential part of the verification process of today's VLSI circuits since it allows to optimize the test time and provides the means to test for delay faults. From a general point of view, at-speed testing does not necessarily lead to delay fault testing, but delay fault testing needs to be performed at-speed to detect timing defects and test the performance of the circuit.

A problem that occurs with currently-used external testers, however, is that they are several times slower than the designs they have to test. Purchasing high speed testers that meet the performances of new designs requires a huge investment. Moreover, even with those high speed testers, it is not always possible to have a timing accuracy comparable to the IC internal speed [21]. In this context, BIST represents an attractive test solution since it allows at-speed testing of the circuit under test, thus solving timing accuracy and test time related problems encountered with traditional external testers. In addition, BIST drastically reduces the amount of test data exchanged with the tester, thus reducing the need for complex external testing equipment. BIST can hence be run on a very low cost tester. Finally, BIST solves the problem of tester capacity (very often, external testers do not have enough memory to store the entire test set to cover stuck-at, transition and path delay faults [11]) and the problem of low accessibility of internal nodes of the design, that increases the test complexity [20].

Although the work presented in this paper comes within the general frameworks of delay testing and random testing, the targeted application is logic BIST for delay faults. BIST is generally based on pseudo-random testing [11]. Pseudo-random testing refers to the application of test patterns that exhibit randomness, but which are generated using special-purpose hardware (LFSR or Cellular Automata), and are thus repeatable. In terms of delay fault coverage, conventional pseudo-random test patterns in which more than one bit change between two consecutive patterns are not efficient to robustly test combinational circuits in a reasonable test time [19]. This fact is what has motivated the development of pseudo-random BIST techniques in which Single Input Change (SIC) test pairs are generated for testing delay faults [8, 19]. SIC test pairs are sufficient to detect all robustly detectable path delay faults [17], with a test length shorter than that required with Multiple Input Change (MIC) test pairs [19]. Note that robustness of delay tests is important to guarantee timing correctness of the CUT since it allows to detect a delay fault even in the presence of other delay faults in the circuit. Additionally, as not all delay faults have a robust test, the effectiveness of SIC test pairs (compared with MIC test pairs) has also been observed in the case of a fault detected by a validatable non-robust test (a non-robust test validated by another test in the sequence) [5].

Several authors have studied the performance of test sequences which are SIC but not truly random. In [4, 8, 19], after each generation of a classical **pseudo-random** vector, a string of SIC vectors is applied to the CUT. In these approaches, a high robust fault coverage is obtained thanks either to a weighted random generation [19] or to an efficient shifting mechanism [8]. Another approach, proposed in [10], consists in using a deterministic sequence for the stuck-at faults. From this set of deterministic patterns, the authors produce a SIC sequence which is very effective for the path delay fault testing. From a general point of view, these papers concentrate their analysis on the coverage of faults having at least one robust test.

Our paper differs from those previous studies into two points. Firstly, we are interested in measuring the performance of a truly Random SIC (RSIC) sequence to test path delay faults, in comparison with the performance of a truly Random MIC (RMIC) sequence. Secondly, we aim at taking into account all the delay faults, even those that do not have a robust test. A fault with no robust test is worth being tested too! (a circuit can be synthesized such that all path faults are robustly testable [15], but most of the existing circuits do not have this property).

In practice, a truly RSIC sequence is difficult to produce since any generation (software or hardware) is pseudo-random in nature. However, a "careful" generation allows to obtain test sequences whose randomness is very good (we mean that they have roughly the same properties than a pure random generation). Since a good generation is easier by software than by hardware [6], the results in this paper are based on a software generation briefly explain in Section 2. Hardware generation (required for BIST) will be shortly discussed by the authors at the end of the paper.

The rest of the paper is organized as follows. Section 2 gives some preliminary definitions on delay fault testing. Section 3 analyses the delay fault coverage of RSIC test sequences on a case study, and Section 4 generalizes the results obtained to other circuits and other fault models. Concluding remarks and future work are discussed in Section 5.

2. Preliminaries

Correct operation of a circuit at the intended speed requires that any path delay exceeds the value determined by the clock period. This is usually verified by delay testing, using the path delay fault model [17]. In this model, it is assumed that the presence of a delay fault increases the delay along the path. This model represents *distributed delays* in the circuit, often caused by *device parameter* variation [16], as well as single isolated failures. This is the main advantage of this model over other existing delay fault models, namely the gate delay fault model and the transition delay fault model [1, 12]. However, an important feature of the path delay fault model is that the single fault assumption is not realistic since a single defect usually affects a large number of paths. For this reason, a robust test is preferred to detect a path delay fault. A *robust test* is a test that detects a delay fault regardless of all other delays and delay faults in the circuit [17]. In contrast, a test that detects a fault with the assumption that no other delay fault can exist in the circuit is called a *non-robust test*.

A test for a path delay fault consists of propagating a transition along the target path $P = (g_0, g_1, \ldots, g_n)$, where each g_i is a gate except g_0 and g_n which are the source and the destination of the path respectively (usually an input and an output of the circuit). The inputs of g_i other than the output of g_{i-1} are called *side-inputs* of g_i (or side-inputs of P). Each connection between g_{i-1} and g_i is called an *on-input* of P. Sensitization conditions for non-robust and robust testability of a path delay fault are as follows. A non-robust test for a rising (or a falling) transition on a path P sets every side-input of P to a final non-controlling value. In a robust test, side-inputs must be at stable non-controlling values (with no static hazards) when on-path inputs have transitions to final controlling values. These conditions are illustrated in Fig. 1 for an AND-type logic gate.

As the purpose in this work was not to develop a delay fault simulator to evaluate the fault coverage of the experimented delay test sequences (random SIC and MIC test sequences), we used an existing industrial test evaluation package, TestGen of Synopsys [18], to perform test validations. In the delay fault tool suite of this package, several models are supported for testing delay paths. The conventional non-robust delay fault model is referred to as the *weak non-robust delay model* in TestGen, and delay fault simulation or test generation from this model is possible. Conversely, the conventional robust delay fault model is also supported by TestGen, but one of the main sensitization constraints in this model (no glitches on the side-inputs when on-path inputs have transitions to final controlling values) is not verified during delay fault simulation or test generation. As a consequence, fault-free results cannot be ensured when using such delay fault model with TestGen.

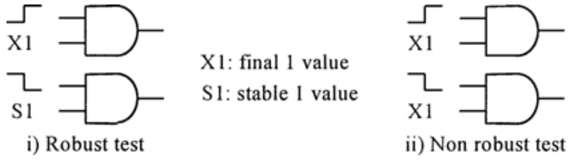

Fig. 1. Robust and non-robust test conditions for path delay faults.

The closest delay fault model handled by this tool is the *strong non-robust delay fault model*, for which the sensitization conditions are the following: in a strong non-robust test, side-inputs must be at initial and final non-controlling value when on-path inputs have transitions to final controlling values (the only difference with the conventional robust delay model states in the acceptance of static hazards on side-inputs). We used the strong non-robust delay model (in addition to the non-robust delay model) as a metric to evaluate the quality of the test sequences generated for testing path delay faults. As the strong non-robust delay model is very close to the conventional robust delay model, this model will be referred to as the ***pseudo-robust delay model*** in the sequel.

Apart from robust and non-robust testable paths, two other classes of paths have been defined in the literature: *functional sensitizable* paths and *functional redundant* paths [13]. Functional redundant paths can never determine the performance of the circuit and do not have to be tested [3]. On the other hand, defects on functional sensitizable paths may degrade the circuit performance when several path delay faults occur simultaneously. Although the number of functional sensitizable paths in a circuit may be not negligible, this model of path delay faults is not handled by TestGen. For this reason, only pseudo-robust and non-robust tests are considered in the rest of this study.

In general, two-pattern tests may vary in multiple bit positions. In this case, they are called multiple input change (MIC) pattern pairs. Test pattern pairs that differ in exactly one bit are called adjacent or single input change (SIC) pattern pairs. Let us now define what a RMIC and a RSIC sequences should be from a theoretical point of view (we assume implicitly the case of equal likelihood of all vectors). Let $S = V(1)\,V(2)\ldots V(l)\ldots V(L)$ be a test sequence composed of L successive n-bit vectors $V(l)$. Each vector takes a value from the set $V = \{V_0,\ V_1, \ldots, V_j, \ldots, V_{2^n-1}\}$, where V_j corresponds to the n-bit vector associated with the decimal value j. For example, for $n = 5$, $V_9 = 01001$, i.e., $x_5 = x_3 = x_2 = 0$ and $x_4 = x_1 = 1$. In a RMIC sequence, the probability $Pr[V(l) = V_j] = 1/2^n$ for any l and any j, and the probability $Pr[V(l) = V_j]$ is independent of the values $V(i),\ i = 1,\ \ldots, l-1$. In a RSIC sequence, $Pr[V(l) = V_j | V(l-1) = V_k] = 1/n$ iff $|j-k| = 2^a$, where a is a non-negative integer (in other words, $V(l)$ differs from $V(l-1)$ by exactly one bit randomly drawn, $V(l+1)$ differs from $V(l)$ by exactly one bit, etc).

The RSIC sequence is software generated as follows. We use an instruction "random", drawing a (pseudo) random number Y uniformly distributed in the range [0,1). Let $V(l) = x_1(l)x_2(l)\ldots x_n(l)$ the vector at time l. If the random value $Y(l+1)$ is drawn, then the input variable x_i such that $i = \lfloor Y\,(l+1) \bullet n \rfloor$ is changed, i.e., $x_i(l+1) \neq x_i(l)$ and $x_j(l+1) = x_j(l)$ for any $j \neq i$ (Section 10.2 in [6]). The principle of the hardware generation is similar, the vector Y is obtained from a LFSR.

A delay test consisting of adjacent or SIC vectors is called an *adjacency* or an *asynchronous* test. As a single transition is applied at the primary inputs of the CUT in an adjacency test, the probability of delay test invalidation due to hazards or multiple delay faults is greatly reduced. This is one of the main reasons for using such kind of tests. Moreover, SIC test pairs are sufficient to detect all robustly detectable path delay faults [17]. Finally, the universe of pattern pairs considered for SIC test generation ($\mathrm{O}(n \cdot 2^n)$) is significantly smaller than that for MIC test generation ($\mathrm{O}(2^{2n})$). Hence, SIC fault coverage can be higher than MIC fault coverage for the same test length [5, 19].

3. Delay Fault Coverage—A Case Study

This case study corresponds to experiments performed on the combinational part of circuit s382 of the ISCAS'89 benchmark set [2]. The performance of the random SIC and MIC test sequences is evaluated based on the results obtained by a deterministic ATPG (TestGen Tg3.0.2 of Synopsys [18]). For further comparison purpose, the results obtained from this ATPG are given in Table 1 for some of the ISCAS89 circuits (combinational part). Notations in Table 1 are as follows. Fc SA, Fc PR and Fc NR are the stuck-at fault coverage, the pseudo robust delay fault coverage and the non-robust delay fault coverage respectively. *Ls* and *Ld* are the test length of stuck-at fault and delay fault test sequences respectively. These lengths are the test lengths required to achieve the maximum fault coverage.

Table 1 should be interpreted as follows. Circuit s382 (its combinational part) has 24 inputs. All the stuck-at faults are detected with a test sequence of length 36. The delay test sequence is composed of 1398 input vectors. When this sequence is applied to the CUT, 88% of the delay faults are pseudo-robustly (PR) tested (12% of the faults have no robust test), 91.75% are non-robustly (NR) tested, and 8.25% are functionally sensitizable or redundant [3]. As the set of faults covered by NR tests includes the set of faults covered by PR tests, there are exactly $(91.75 - 88.00) = 3.75\%$ of delay faults that have at least one NR test but no PR test.

In order to measure the performance of a test sequence, let us denote by *Eff* (Efficiency) the ratio of

Table 1. Results obtained on the ISCAS'89 circuits.

Circuits	*n*	*Ls*	Fc SA	*Ld*	Fc PR	Fc NR
s298	17	38	100	688	76.19	78.79
s382	24	36	100	1398	88	91.75
s386	13	86	100	736	100	100
s420	35	55	100	1322	100	100
s510	25	67	100	1366	100	100
s526	24	68	99.82	1396	86.34	87.8
s641	54	46	100	2778	66.16	71.02
s713	54	90	93.46	4418	22.54	37.13
s1238	32	196	94.80	4992	62.17	62.93
s1494	14	158	99.19	3518	98.83	98.89
s3330	172	238	100	15862	89.87	90.94
s5378	214	143	99.13	29730	74.02	82.2

faults detected by the sequence over the maximum number of testable faults. For example, the pseudo-robust efficiency of a sequence S that detects 44% of the PR testable faults in circuit s382 is 44/88 = 0.5 (i.e. 50%). If it detects 88% of the PR testable faults, its PR efficiency is *Eff* = 100%.

Table 2 given above presents simulation results obtained from one RMIC and one RSIC sequences. Note that results would be quite similar with others RMIC or RSIC sequences. In Table 2, the results are given for various test lengths. For example, for $Ld = 1398$ vectors, the following results are obtained:

1) RMIC sequence: a PR test has been found for 33.24% of the PR testable faults and a NR test has been found for 91.01% of the NR testable faults.
2) RSIC sequence: a PR test has been found for 52.84% of the PR testable faults and a NR test has been found for 57.22% of the NR testable faults.

The graph in Fig. 2 is a graphical representation of the results given in Table 2.

The following observations can be made:

1) The PR efficiency of the RSIC sequence is close to the NR efficiency, while there is a huge gap between these efficiencies for the RMIC sequence. The reason is that, due to the adjacency of successive vectors, most of the tests found in the RSIC sequence are robust.

Table 2. PR and NR efficiencies of given RMIC and RSIC sequences.

s382	Number of vectors	RMIC sequence *Eff PR*	RMIC sequence *Eff NR*	RSIC sequence *Eff PR*	RSIC sequence *Eff NR*
	10	12.78	16.49	1.70	1.63
	100	25.57	52.04	14.35	15.40
Ld/4	349	30.97	73.97	24.57	25.61
Ld/2	699	33.09	86.78	43.47	46.05
Ld	1398	33.24	91.01	52.84	57.22
2Ld	2796	33.52	96.86	61.36	66.49
5Ld	6990	33.52	99.31	76.56	84.33
10Ld	13980	33.52	99.86	81.53	90.32
20Ld	27960	33.52	99.86	86.65	96.19
50Ld	69900	33.52	99.86	89.06	98.64
100Ld	139800	33.52	99.86	90.19	99.73
	1000000	33.52	99.86	90.48	100.00

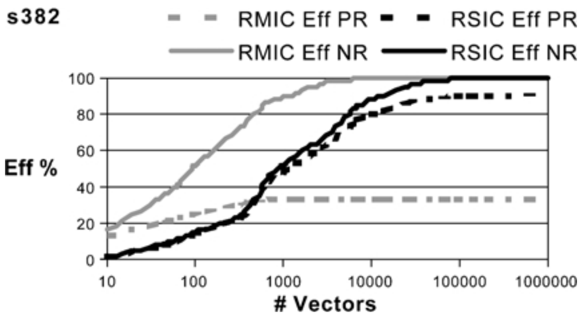

Fig. 2. Efficiency comparison of RMIC and RSIC sequences.

2) For a short test length (not interesting in practice), the fault efficiencies achieved by the RSIC test sequence are lower than those achieved by the RMIC sequence. *For average and long test lengths, the NR efficiency of the RMIC sequence is much higher than that of the RSIC sequence, while the RSIC sequence is more efficient in terms of PR efficiency.*
3) For very long test lengths, the RSIC test sequence may be as efficient as the RMIC sequence regarding the NR efficiency (last line in Table 2). It is much more efficient regarding the PR efficiency.

To summarize, we can say that the RMIC sequence achieves a better NR efficiency while the RSIC sequence leads to a better PR efficiency (for average and long test lengths). Consequently, a question should be asked: what is the best test sequence to be used when both PR efficiency and NR efficiency are targeted? Let us then try to find a single criterion to help selecting the best test sequence.

Let us remind that, for circuit s382, 88% of the faults have at least one PR test and 91.75% of the faults have at least one NR test (Table 1). In other words, **91.75% of the delay faults are NR testable** and $q = 88/91.75 = 0.959$, i.e. **95.9% of the NR testable faults have at least one PR test**. Let us try now to answer the following question: what is the fraction of NR testable faults which is effectively detected by a RMIC sequence of length $Ld = 1398$?

According to Table 2, a PR test has been found for 33.24% of the faults having at least one PR test. Since the fraction of NR testable faults having at least one PR test is 0.959, a PR test has been found for $33.24 \times 0.959 = 31.88\%$ of the NR testable faults. According to the basic property of a robust test (detection of a fault irrespective of the presence of other delay faults), at least 31.88% of the NR testable faults will

Table 3. Weighted efficiency for various assumptions about the success rate S_{NR} of NR tests.

	Weighted efficiencies for circuit s382								
		RMIC sequence				RSIC sequence			
	Number of vectors	*Eff NR* − *Eff PR*	S_{NR} 10%	S_{NR} 50%	S_{NR} 90%	*Eff NR* − *Eff PR*	S_{NR} 10%	S_{NR} 50%	S_{NR} 90%
	10	3.71	12.60	13.98	15.36	−0.07	1.63	1.60	1.58
	50	16.02	21.92	27.88	33.83	0.73	8.51	8.78	9.06
	100	26.48	26.98	36.82	46.66	1.05	13.86	14.25	14.64
Ld/4	349	43.01	33.69	49.67	65.65	1.04	23.66	24.05	24.43
Ld/2	699	53.69	36.72	56.67	76.62	2.58	41.92	42.88	43.84
Ld	1398	57.77	37.24	58.71	80.18	4.38	51.08	52.71	54.34
2Ld	2796	63.34	38.03	61.57	85.11	5.12	59.32	61.23	63.13
5Ld	6990	65.79	38.26	62.71	87.16	7.77	74.14	77.03	79.91
10Ld	13980	66.34	38.31	62.96	87.61	8.79	79.01	82.27	85.54
20Ld	27960	66.34	38.31	62.96	87.61	9.54	83.98	87.53	91.07
50Ld	69900	66.34	38.31	62.96	87.61	9.58	86.30	89.86	93.42
100Ld	139800	66.34	38.31	62.96	87.61	9.53	87.38	90.92	94.47
	1000000	66.34	38.31	62.96	87.61	9.52	87.65	91.19	94.73

be actually detected during testing of the CUT. On the other hand, at most 91.75% of the NR testable faults will be actually detected. As a matter of fact, there are $91.75 - 31.88 = 59.87\%$ of the NR testable faults for which no PR test has been found but for which a NR test has been found. A question now arises: what is the fraction of faults that will be actually detected among the faults that are only NR testable? This depends on the number and the nature of the distributed defects that affect the circuit and that can invalidate the non-robust tests. Let us call this fraction the *success rate* of NR tests, and denote by S_{NR}.

Since we have no answer to this question, assumptions can be stated. Let us assume that the success rate is $S_{NR} = 0.5$ (i.e., 50%), then the weighted efficiency for our example is $31.88 + (91.75 - 31.88) \times 0.5 = 61.82$. Let us now remind the notation and generalize the calculation: $q = (\text{Fc PR}/\text{Fc NR})$ is the fraction of NR testable faults having at least one PR test, S_{NR} is the assumed value of the success rate of NR tests, and $\mathit{Eff}(S_{NR})$ is the weighted efficiency of a test sequence for a given S_{NR}. The weighted efficiency of a test sequence for a given success rate S_{NR} is given by Eq. (1) described below:

$$\begin{aligned}\mathit{Eff}(S_{NR}) &= \mathit{Eff\,PR} \times q \\ &\quad + (\mathit{Eff\,NR} - \mathit{Eff\,PR} \times q) \times S_{NR} \quad (1)\end{aligned}$$

The application of Eq. (1) to circuit s382 for the RMIC and RSIC test sequences and for $S_{NR} = 10\%$, 50% and 90% are presented in Table 3 for various test lengths. These results are also presented in the graph of Fig. 3. In this figure, it clearly appears that for a long test length, the weighted efficiency is higher for the RSIC sequence than for the RMIC sequence, for all assumed values of the success rate S_{NR}.

As a conclusion, these results demonstrate that, even with a lower non-robust delay fault coverage, a RSIC test sequence may often give rise to a better test quality than that obtained with RMIC delay test sequences. This is the most important result we wanted to demonstrate in this paper. Note that this conclusion drawn from a study on circuit s382 is also valid for most of the ISCAS'89 benchmark circuits.

4. Application to Other Circuits and Other Fault Models

An important comment on the validation results given in this paper is that the ISCAS'85 circuits family has not been used in our experiments. This is because the number of path faults in these circuits is too huge, and TestGen [18] is unable to generate the corresponding delay fault dictionaries for fault simulation. However,

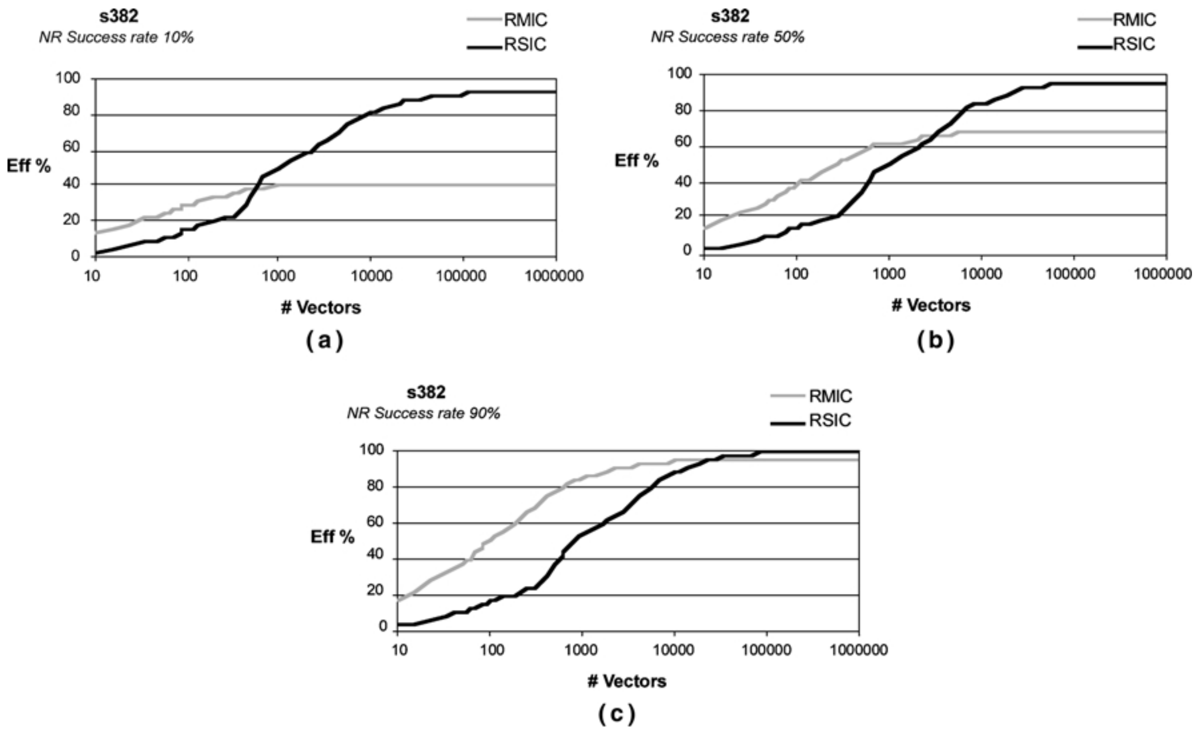

Fig. 3. Comparison of weighted efficiencies of RMIC and RSIC sequences for various values of S_{NR}: (a) 10%, (b) 50%, (c) 90%.

results similar to those reported in the previous section for circuit s382 were obtained for various circuits of the ISCAS'89 benchmark set. For example, results obtained for circuits s1238 are given in Fig. 4. Results obtained for other ISCAS'89 circuits are listed in Table 4 (the test length considered are 100 Ld for every circuit).

A RSIC sequence has the important property that it is made of adjacent vectors. Hence, it may be efficient for testing delay faults (the aim of this paper) or for testing stuck-open faults that also require pairs of vectors. In addition, it may be interesting to study the efficiency of RSIC sequences for other fault models. For stuck-at faults, for example, it is expected that the test length would be longer than for a RMIC sequence

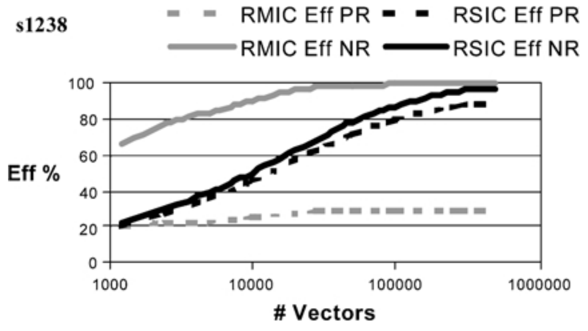

Fig. 4. Comparison of RSIC and RMIC sequence efficiencies for circuit s1238.

Table 4. Comparison between pseudo-robust fault coverage of RSIC and RMIC testing.

Circuit	ATPG FC %	Number of patterns ⇒	RSIC *Eff %*	RMIC *Eff %*
s298	76.19	68800	91.48	36.65
s382	88.00	139800	91.19	33.38
s386	100	73600	98.79	40.82
s420	100	132200	66.86	37.24
s510	100	136600	91.11	31.78
s526	86.34	139600	90.40	30.09
s641	66.16	277800	76.01	18.12
s713	22.54	441800	83.72	20.76
s1238	62.17	499200	89.11	29.35
s1494	98.83	351800	94.51	30.83
s3330	89.87	158620	62.28	23.47
s5378	74.02	297300	64.77	24.51

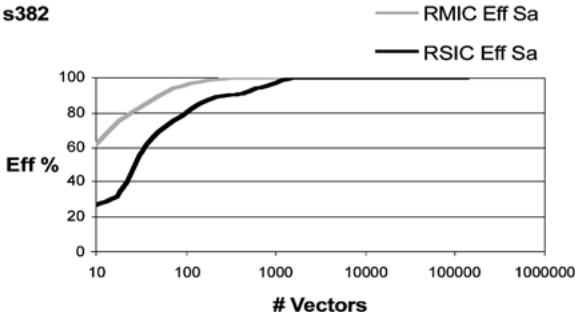

Fig. 5. Comparison between stuck-at fault coverage of RSIC and RMIC—circuit s382.

since a RMIC sequence samples more quickly all the input state space. However, all this space can be covered when the RSIC sequence length increases. This observation is confirmed by the example given in Fig. 5.

5. Concluding Remarks and Future Work

The efficiency of RSIC test sequences for delay fault testing was analyzed in this paper. The performance measurement takes into account both robust tests and non-robust tests. The main conclusion drawn from this study is that, even with a lower non-robust delay fault coverage, a RSIC test sequence may often give rise to a better test quality than that obtained with RMIC delay test sequences. This conclusion is a fundamental result that can be further used by test engineers when they will have to provide test sequences for delay fault testing.

Several extension to this study have been foreseen for the near future.

A first one concerns with the development of a complete hardware generation structure providing random SIC sequences [1]. This generation will require that:

1) a random number is hardware transformed into a random bit;
2) the bit changed at time l is completely independent from the bit changed at time $l - 1$;
3) the period of the RSIC sequence is large enough.

Solutions to points 1 and 2 are explicitly given in [6], and point 3 is to be developed. In addition, the area overhead has to be estimated and compared with other methods.

Another extension of this study concerns with the evaluation of SIC test sequences in testing faults other than delay faults, i.e. stuck-at faults, bridging faults, etc. A preliminary study has been carried out on some ISCAS circuits. Results and analysis can be found in [9]. Some discussions on the universal nature of SIC sequences for the testing purpose will be tackled in a future paper.

References

1. J.A. Abraham and H. Shih, "Testing of MOS VLSI Circuits," in *IEEE Int. Symp. on Circuits and Systems*, 1985, pp. 1297–1300.
2. F. Brglez, D. Bryant, and K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits," in *IEEE Int. Symp. on Circuits and Systems*, 1989, pp. 1929–1934.
3. K.T. Cheng and H.C. Chen, "Delay Testing for Non-Robust Untestable Circuits," in *IEEE Int. Test Conf.*, 1993, pp. 954–961.
4. G.L. Craig and C.R. Kime, "Pseudo-Exhaustive Adjacency Testing: A BIST Approach for Stuck-Open Faults," in *Proc. of Int. Test Conf.*, 1985, pp. 126–137.
5. S. Crepaux-Motte, M. Jacomino, and R. David, "An Algebraic Method for Delay Fault Testing," in *IEEE VLSI Test Symposium*, 1996, pp. 308–315.
6. R. David, *Random Testing of Digital Circuits: Theory and Applications*, New York: Marcel Dekker, Inc., 1998.
7. R. David, P. Girard, C. Landrault, S. Pravossoudovitch, and A. Virazel, "On Hardware Generation of Random Single Input Change Test Sequences," in preparation.
8. P. Girard, C. Landrault, V. Moreda, and S. Pravossoudovitch, "An Optimized BIST Test Pattern Generator for Delay Testing," in *IEEE VLSI Test Symposium*, 1997, pp. 94–99.
9. P. Girard, C. Landrault, S. Pravossoudovitch, and A. Virazel, "Comparison Between Random and Pseudo-Random Generation for BIST of Delay, Stuck-at and Bridging Faults," in *IEEE Int. On-Line Testing Workshop*, 2000, pp. 121–126.
10. C.T. Glover and M.R. Mercer, "A Deterministic Approach to Adjacency Testing for Delay Faults," in *IEEE Design Automation Conf.*, 1989, pp. 351–356.
11. G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassad, A. Hassan, and J. Rajski, "Logic BIST for Large Industrial Designs: Real Issues and Case Studies," in *IEEE Int. Test Conf.*, 1999, pp. 358–367.
12. S. Koeppe, "Modeling and Simulation of Delay Faults in CMOS Logic Circuits," in *IEEE Int. Test Conf.*, 1986, pp. 530–536.
13. W.K. Lam, A. Saldanha, R.K. Brayton, and A.L. Sangiovanni-Vicentelli, "Delay Fault Coverage, Test Set Size, and Performance Trade-Offs," *IEEE Trans. on CAD*, Vol. 14, No. 1, pp. 32–44, January 1995.
14. A.K. Majhi and V.D. Agrawal, "Tutorial: Delay Fault Models and Coverage," *Proc. 11th International Conf. VLSI Design*, 1998, pp. 364–369.
15. A.K. Pramanick and S.M. Reddy, "On Multiple Path Propagating Tests for Path Delay Faults," in *IEEE Int. Test Conf.*, 1991, pp. 393–402.
16. M. Sivaraman and A.J. Strojwas, "Diagnosis of Parametric Path Delay Faults," in *IEEE Int. Conf. on VLSI Design*, 1996, pp. 412–417.
17. G.L. Smith, "Model for Delay Faults Based upon Paths," in *IEEE Int. Test Conf.*, 1985, pp. 342–349.

18. TestGen, version Tg3.0.2 User Guide, Synopsys Inc., 1999.
19. W. Wang and S.K. Gupta, “Weighted Random Robust Path Delay Testing of Synthesized Multilevel Circuits,” in *IEEE VLSI Test Symposium*, 1994, pp. 291–297.
20. H.J. Wunderlich and Y. Zorian, *Built-In Self Test (BIST): Synthesis of Self-Testable Systems*, Kluwer Academic Publishers, 1997.
21. Y. Zorian, “Testing the Monster Chip,” *IEEE Spectrum*, Vol. 36, No. 7, pp. 54–60, 1999.

Arnaud Virazel was born in France in 1974. He received the B.Sc. (1995) and M.Sc. degrees (1997) in Electrical Engineering from Sciences University of Montpellier, France. He is presently Ph.D. student, and works in the Microelectronics Department of the LIRMM (Laboratory of Computers Sciences, Robotics and Microelectronics of Montpellier, France). His research interests are in Delay Fault Testing and Built-In-Self-Test.

René David was born in Saint-Nazaire, France, in 1939. He received the Engineering Degree in 1962 at the Ecole Nationale Supérieure d'Arts et Métiers. In 1963 an accident stopped his work and left him paraplegic. In 1965 he received the Automatic Control Engineering degree at the University of Grenoble and in 1969 he received the Doctorat d'Etat ès-Sciences degree at the same University. He is now Directeur de Recherche Emérite at the Centre National de la Recherche Scientifique, working at the Laboratoire d'Automatique de Grenoble (in Institue National Polytechnique de Grenoble). He was Deputy Director of this laboratory from 1987 to 1990. His works covers design of logic controllers, testing of digital circuits, modelling and performance evaluation of manufacturing systems. He is the author of *Random Testing of Digital Circuits: Theory and Application* (Marcel Dekker Inc., New-York, 1998). He is a co-author of *Du Grafcet aux réseaux de Petri* (Hermès, Paris, 1989 then 1992) and *Petri Nets & Grafcet: Tools for Modelling Discrete Even Systems* (Prentice-Hall, London, 1992); this book was translated into Chinese (Machinery Press, Beijing, 1996).

Patrick Girard was born in France in 1964. He is presently Researcher at CNRS (National Center for Scientific Research), and works in the Microelectronics Department of the LIRMM (Laboratory of Informatics, Robotics and Microelectronics of Montpellier—France). His research interests include the various aspects of digital testing, with special emphasis on logic BIST, delay fault testing, and low power test. He has authored and co-authored numerous papers on these fields, and has supervised several PhD dissertations. He has also participated to several European project (Esprit, MEDEA). Patrick GIRARD holds the BSc degree and the MSc degree in Electrical Engineering, and obtained the PhD degree in Microelectronics from the University of Montpellier in 1992.

Christian Landrault was born in Orléans (France) on December 8, 1946. He has held a degree in aeronautical engineering from “Ecole Nationale d'Ingénieurs de Constructions Aéronautiques,” Toulouse in 1970. From 1970 to 1980, he worked at “Laboratoire d'Automatique” et d'Analyse des Systèmes (LAAS, Toulouse) on self checking circuits, dependability architecture and dependability evaluation of digital systems. In 1980, he joined the Laboratory of Informatics, Robotics and Microelectronics of Montpellier (LIRMM). He is presently Research Director at CNRS and head of the Microelectronics Department at LIRMM, he works mainly in testing. He has held the “Docteur-Ingénieur” degree in 1973 and the “Docteur d'Etat” degree in 1977 both from the LAAS in Toulouse. He has published over 100 papers in journals and conference, he has participated to 25 research contracts and has managed 18 of them. He is an Editor of the *Journal of Electronic Testing: Theory and Applications*.

Serge Pravossoudovitch was born in 1957. He is currently professor in the electrical and computer engineering department of the University of Montpellier and his research activities are performed at LIRMM (Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier). He received the master degree in electrical engineering in 1979 from the University of Montpellier. He got his PhD degree in electrical engineering in 1983 on symbolic layout for IC design. Since 1984, he was been interested in the testing domain. He obtained the “doctorat d'état” degree in 1987 for his works on switch level automatic test pattern generation. He is presently interested in delay fault testing, design for testability and power consumption optimization. He has authored and co-authored numerous papers on these fields, and has supervised several PhD dissertations. He has also participated to several European project (Microelectronic regulation, Esprit, Medea).