An Interval-based Diagnosis Scheme for Identifying Failing Vectors in a Scan-BIST Environment*

Chunsheng Liu†, Krishnendu Chakrabarty† and Michael Goessel‡

†Dept. Electrical & Computer Engineering
Duke University, Durham, NC 27705, USA

‡ Institute of Computer Science
University of Potsdam, 14469 Potsdam, Germany

Abstract

We present a new scan-BIST approach for determining failing vectors for fault diagnosis. This approach is based on the application of overlapping intervals of test vectors to the circuit under test. Two MISRs are used in an interleaved fashion to generate intermediate signatures, thereby obviating the need for multiple test sessions. The knowledge of failing and non-falling intervals is used to obtain a set S of candidate failing vectors that includes all the actual (true) failing vectors. We present analytical results to determine an appropriate interval length and the degree of overlap, an upper bound on the size of S, and a lower bound on the number of true failing vectors; the latter depends only on the knowledge of failing and non-failing intervals. Finally, we describe two pruning procedures that allow us to reduce the size of S, while retaining most true failing vectors in S. We present experimental results for the ISCAS 89 benchmark circuits to demonstrate the effectiveness of the proposed scan-BIST diagnosis approach.

1 Introduction

As process technologies shrink and designs become more complex, built-in self-test (BIST) is gaining increasing acceptance as an industry-wide test solution [1]. In particular, the combination of scan design and BIST, commonly referred to as scan-BIST, is now especially common [2]. Scan-BIST techniques typically apply a large number of patterns from a pseudorandom pattern generator (PRPG) to the circuit under test (CUT) via scan chains. The test responses are then captured by the scan chain and a compact signature is generated using a multiple-input signature register (MISR); see Figure 1. However, a problem with this approach is that the signature provided by the MISR does not contain enough diagnostic information, either to identify failing vectors or to precisely identify error-capturing scan cells. The pass/fail information obtained from the MISR at the end of the test session is usually insufficient to diagnose the failure via effect-cause analysis.

Fault diagnosis is essential for the identification of manufacturing defects and for yield learning. The cost of diagnosis is proportional to the time required for failure analysis, which can be extremely high for a scan-BIST scheme involving tens of thousands or millions of vectors [3]. Therefore, there is a pressing need for BIST schemes that provide adequate diagnostic information, without burdening the failure analysis process with superfluous information.

The diagnostic information in scan-BIST can be classified as *space information* and *time information*, respectively. The former refers to the set of scan cells that capture errors during the BIST session. This problem has received a lot of attention recently, and a number of methods involving scan chain partitioning methods with multiple test sessions have been proposed for precisely identifying the failing scan cells [4-7]. A more difficult problem in scan-BIST diagnosis is that of identifying the set of failing vectors. This is because the length of a scan chain in a typical BIST scheme is usually much smaller than the number of test vectors applied to the CUT. As a result, fewer practical techniques are available today for rapidly identifying a small set of candidate failing vectors.

Early work on failing vector identification was based on the analysis of LFSR sequences [8], and the use of cycling registers [9] and error-correcting codes [10,11]. An alternative approach that does not require intermediate signatures was presented in [12]. These techniques suffer from the drawback of limitations on error multiplicity [8], diagnostic aliasing [9,10,11], and high overhead [10,12]. Recently, a method based on the combination of cycling registers and pruning techniques was proposed for failing vector identification [13]. While this approach is useful in narrowing down the set of candidate failing vectors, it suffers from the drawback that it does not identify *all* the failing vectors.

In this paper, we present a new technique for failing vector identification based on the use of overlapping intervals of test vectors. An interval is a set of consecutive test vectors. An advantage of this approach is that *all* failing vectors are included in a reduced set of candidate vectors for failure analysis. The overlap allows us to prune the set of candidate failing vectors. An interval that does not contain a failing vector can be omitted from the set of candidate failing vectors. The overlap ensures that if a failing interval I_1 is followed by a non-failing interval I_2, only the set difference $I_1 - I_2$ needs to be retained in the set of candidate failing vectors. In order to reduce the candidate set further, a pruning procedure is performed as optional post-processing step prior to failure analysis.

The proposed interval-based approach also allows us to determine a lower bound on the number of actual (true) failing vectors. This bound is obtained via a simple graph model based on the pass/fail status of the intervals.

In addition to facilitating effect-cause analysis, the proposed approach can also benefit cause-effect analysis [14]. For example, the knowledge of candidate failing vectors can improve the resolution provided by a compact fault dictionary [15].

*This research was supported in part by National Science Foundation under grant No. CCR-9875324, and in part by Deutsche Forschungsgemeinschaft, Germany.

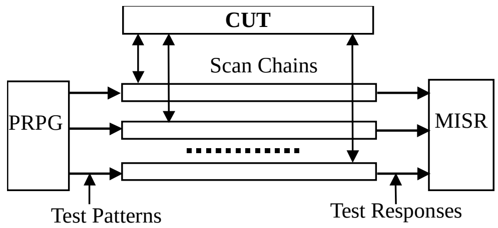

Figure 1. Generic scan-BIST scheme.

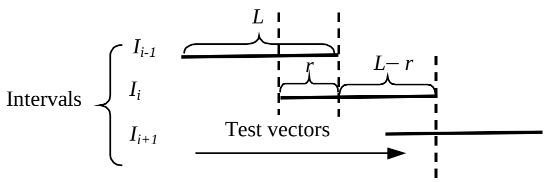

Figure 2. Test patterns in intervals.

The organization of this paper is as follows. In Section 2, we present the scan-BIST architecture for this interval-based scheme. In Section 3, we present theoretical analysis to determine the interval length and the amount of overlap. We also determine an upper bound on the number of candidate failing vectors and a lower bower bound on the number of true failing vectors. Finally in Section 4, we present experimental results for the ISCAS 89 benchmark circuits.

2 Interval-Based Diagnosis

In this section, we introduce the notion of intervals of test vectors for diagnosis. Let T be an ordered set of patterns that is applied to the CUT by the test pattern generator (TPG). An interval I corresponds to a subset of consecutive vectors from T. The basic idea of interval-based diagnosis is to divide T into a set of overlapping intervals $I_1, I_2, \ldots, I_N$ such that $T = I_1 \cup I_2 \cup \cdots \cup I_N$ and $I_i \cap I_{i-1} \neq \Phi$.

Figure 2 illustrates three consecutive intervals. The entire test sequence is split into intervals of length L and overlap r. Note that I_i represents the i^{th} interval. The test patterns in an interval are applied to the CUT and the signature is compared for every interval. We assume that the aliasing in the MISR used for signature analysis can be neglected. If an interval contains one or more failing vectors, the corresponding signature is different from the fault-free signature. The diagnosis procedure therefore relies on the knowledge of failing intervals, from which the candidate failing vectors are derived.

A hardware implementation of the proposed scheme is shown in Figure 3. We augment the basic scan-BIST scheme in Figure 1 by using two MISRs with reset input to capture the signature for every interval. The "MISR Selector" determines the start and stop positions of the intervals and thus controls the MUX to output the signature from one MISR at a time. Since two adjacent intervals are overlapping, two output MISRs are used to avoid applying the vectors twice. If $MISR_1$ is used for the current interval, then $MISR_2$ is selected at the start of the next interval. Therefore, $MISR_1$ captures signature of the current interval while $MISR_2$ captures signature of the next, and these MISRs provide the interval signatures in an interleaved fashion. The fault-free signatures can be stored off-

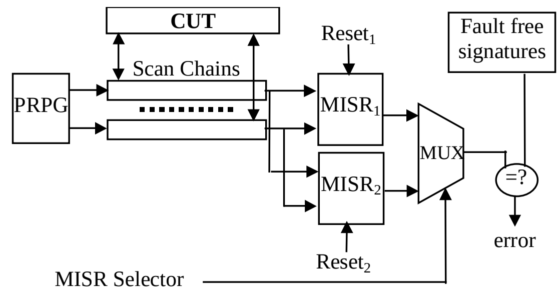

Figure 3. Scan-BIST architecture for interval-based diagnosis.

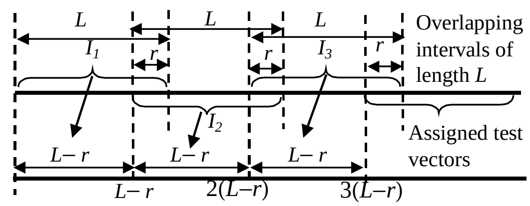

Figure 4. Test vector assignment for calculating size of S

chip since the intermediate signatures can be downloaded after every interval. An alternative approach is to use time redundancy and a standard BIST scheme based on a single MISR. Note that the proposed interval-based approach may potentially be combined with signature analysis methods that use multiple intermediate signatures but a single reference signature [17]. This offers the advantage of reduced storage requirement for diagnosis—the details are currently being investigated.

The use of overlapping intervals allows us to reduce the number of candidate failing vectors. This can be explained as follows: If I_i is a failing interval and I_{i-1} is not a failing interval, the set of vectors in $I_i \cap I_{i-1}$ can be eliminated from the set of candidate failing vectors. In this way, I_i and I_{i-1} together contribute only $L - r$ vectors to the set of candidate failing vectors. In order to limit the number of intervals and MISR signatures, we assume that a vector belongs to at most two intervals. In general, let I_F be the set of failing intervals and let I_{NF} be the set of non-failing intervals. The set of candidate failing vectors Ω is then given by $\Omega = \bigcup_{i \in I_F} I_i - \bigcup_{j \in I_{NF}} I_j$.

An appropriate value of interval length L must be chosen to trade-off failing vector resolution with the test application time and storage cost. Short intervals lead to better failing vector resolution; however, they require a larger number of signatures for diagnosis. We determine L using a pre-processing step to estimate the fault detection probability α.

The pre-processing step is motivated by the fact that the use of intervals is more effective when every interval contains only one failing vector. The pre-processing for interval-based diagnosis is performed by a sampling technique. We apply a small number of vectors N_i (typically two to three orders of magnitude less than the number of BIST vectors) to the faulty CUT. We set the interval length L to 1 and determine the number of failing vectors D. The detection probability α is estimated as D/N_i and L is estimated by $\lfloor 1/2\alpha \rfloor$, which is twice the average distance between two failing vectors. This estimate increases the likelihood that failing intervals are separated by non-failing intervals, which leads to smaller S.

3 Analytical Results

In this section, we analyze the effectiveness of interval-based diagnosis. We characterize the failing vector candidate set S in terms of L and r. To ensure that we do not count in our analysis test vectors that belong to two overlapping intervals twice, we assign the first $L-r$ vectors $(1,\ldots,L-r)$ to the first interval, the second $L-r$ vectors $(L-r+1,\ldots,2(L-r))$ to the second interval, and so on. This is illustrated in Figure 4.

Let α be the probability that a faulty CUT is detected by a vector from the BIST test set. The probability p that an interval is failing is given by $p = 1-(1-\alpha)^L$. Consider now two consecutive intervals I_{i-1}, I_i. The function $v(I_i)$ is used to denote the failing/non-failing status of the interval I_i. Table 1 lists the four possible combinations of the status (failing/non-failing) of the intervals. Note that a 0 (1) in the first two columns of the table indicates that the interval is non-failing (failing). The rightmost column gives the contribution of I_i to the set S corresponding to each combination. For example, if $v(I_{i-1})v(I_i) = 01$, the contribution is $(L-r)-r = L-2r$; if $v(I_{i-1})v(I_i) = 11$, the contribution is $L-r$. The total contribution of the combinations due to I_i can be obtained by simply adding the entries in the last column. However, each entry must be weighted with the corresponding probability measure. Moreover, if N is the number of BIST test vectors, the number of intervals k is given by $k = \lceil N/(L-r) \rceil$.

$v(I_{i-1})$	$v(I_i)$	*Probability of occurrence*	*Contribution to S*
0	0	$(1-p)^2$	0
0	1	$(1-p)p$	$L-2r$
1	0	$(1-p)p$	0
1	1	p^2	$L-r$

Table 1. Contribution of intervals to the failing vector set.

The weighted sum leads to the following expression of $|S|$, the size of the candidate failing vector set:

$$|S| = k[(1-p)\,p(L-2r) + p^2(L-r)\,] = Np[1-(1-p)/(L/r-1)].$$

Note that here we ignore the first and the last interval. These boundary cases can be omitted because the BIST length N is much greater than the interval length L. Since $1-p>0$ and we assume that a vector belongs to at most two intervals, the above equation shows that $|S|$ decreases as r is increased from 1 to $L/2$. This justifies the choice of $r = L/2$, and the value of $|S|$ for $r = L/2$ is $\lceil Np^2 \rceil$.

Figure 5 plots the size of S as a function of L for $r = L/2$ and various values of α. As expected, we find that for a given circuit characterized by a certain value of α, using smaller interval length L results in smaller $|S|$ and thus leads to better resolution. However, more MISR signatures are produced with smaller L; as a result both storage and diagnosis time are increased. On the other hand, for a given interval length L, better resolution can be obtained with smaller detection probability α. An appropriate value of L should be therefore chosen to balance resolution with storage cost and diagnosis time.

If the failing vector set determined for a specific interval length L is not satisfactory in either resolution or diagnosis time, a different L can be chosen and the BIST sessions run again. This procedure can be made more efficient by investigating the rate of increase dS of the candidate set size

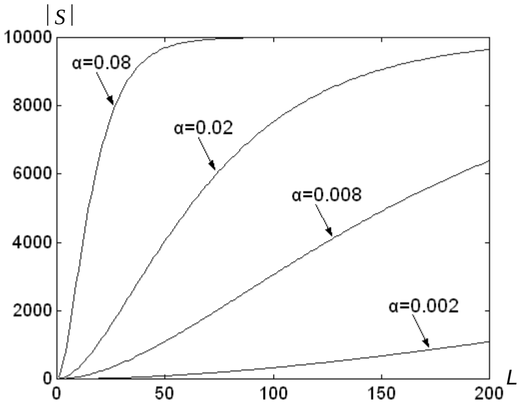

Figure 5. Size of failing candidate failing vector set $|S|$ versus interval length L.

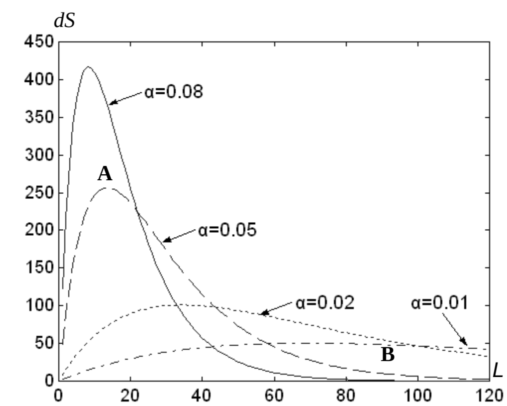

Figure 6. The rate of increase dS of size of the candidate failing vector set $|S|$ versus interval length L.

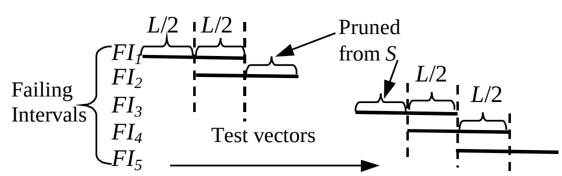

Figure 7. Pruning procedure on failing intervals for $r = L/2$.

$|S|$ with L, i.e. $dS = \partial|S|/\partial L = 2Np(\partial p/\partial L) = -2Np(1-p)\log(1-\alpha)$. Figure 6 plots dS versus interval length L. If better resolution is preferred, L should be chosen near the peak (area A) of the dS curve such that as L varies, $|S|$ decreases at the highest rate. On the other hand, if shorter diagnosis time is required, L should be chosen near area B such that the variation of L will not result in sharp deterioration in resolution.

Note that for $r = L/2$, every vector appears in exactly two intervals (except the $L/2$ vectors in the first interval and the $L/2$ vectors in the last interval). If the boundary case is omitted, a failing vector always leads to two consecutive failing intervals. This observation can be used to prune S as follows. Let FI_1, $FI_2,\ldots, FI_m$ be the sequence of failing intervals where, for any $j > i$, vectors in FI_i are applied before those in FI_j. The candidate failing vector set can be obtained as $S = FI_1 \cup (FI_1 \cap FI_2) \cup (FI_2 \cap FI_3) \;\ldots\; \cup (FI_{m-1} \cap FI_m) \cup FI_m$. This

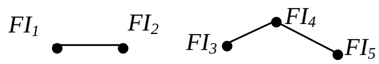

Figure 8. Graph model for the failing vectors in Figure 7.

assumes that FI_1 is the first and FI_m is the last interval applied to the CUT. If this is not the case, they each contributes only $L/2$ vectors to S and the above equation for S must be slightly modified. Figure 7 illustrates this pruning procedure for $m = 5$, where $|S| = L+L+L/2 = 5L/2$.

Next we note that if a hard-to-detect fault is detected by only one vector, $|S| = L/2$. In general, if the number of true failing vectors is f ($f>1$), an upper bound on $|S|$ is given by $|S| \leq fL+L/2$. This bound is reached if all the intervals are consecutive failing intervals and the spacing between two consecutive failing vectors is greater than $L/2$.

Finally, we present a graph-theoretic model that can be used to further analyze the interval-based diagnosis. In particular, it can be used to determine a lower bound LB on the number of true failing vectors. Consider an undirected graph G in which every failing interval corresponds to a vertex, and there exists an edge between two vertices if and only if the corresponding intervals overlap. Let the total number of components in G be t and let v_i be the number of vortices in the i^{th} component of G, $1 \leq i \leq t$. It is clear from the definition of G that every component in it is simply a path. A lower bound is given by: $LB = \sum_{i=1}^{t} \lfloor (v_i + 1)/2 \rfloor$. Consider the failing vectors in Figure 7 as illustrated in Figure 8, this corresponds to $t = 2$, $v_1 = 2$ and $v_2 = 3$, hence $LB = 3$.

The knowledge of the upper and lower bounds is useful to evaluate the procedure for determining failing vectors. The lower bound on the number of true failing vectors provides an important baseline for refining the failing vector identification procedure.

4 Experimental Results

In this section, we present simulation results for the larger ISCAS89 benchmark circuits. We inject a single random fault in each case and use a BIST test sequence of 10,000 patterns. The test patterns are randomly generated and fault simulation is performed using the FSIM program [16]. Although we limit our experiments to single stuck-at fault, the proposed interval-based diagnosis can be easily used for other fault models.

The results are presented in Table 2. While we validated our approach for a large number of faults, we report results here for three faults with different values of α for each circuit. The number of intermediate signatures varied from 40 to 1250. The number of true failing vectors (out of the 10,000 applied to the CUT) is shown in Column 2. Columns 3, 4, and 5 present results of failing vector identification without pruning for interval length $L = \lfloor 1/2\alpha \rfloor$ and overlap $r = L/2$. Column 3 gives the number of suspect vectors $|S|$. Columns 4 and 5 list lower bounds on the number of true failing vectors LB and the number of failing intervals, respectively. Note that the former provides a close estimate for the number of true failing vectors, while the later is near twice of this number, which corresponds to $r = L/2$. In Column 6 the interval length L is decreased to $\lfloor 1/4\alpha \rfloor$; as a result the number of suspect vectors is now reduced to almost half of that in Column 3. Note that without any pruning algorithm, all the true failing vectors out of the 10,000 BIST patterns are included in the candidate set. This feature can greatly facilitate the hardware diagnosis procedures that are employed during failure analysis.

After the candidate failing vectors are identified, we apply a pruning-based post-processing step to reduce the number of candidates. We consider two pruning procedures, which are facilitated by our choice of interval-length—every interval is expected to contain only a few failing vectors.

In the first pruning procedure I, we assume that a failing interval contains exactly one failing vector. We then use a simple binary search procedure with a pre-defined searching depth to determine the subset of the failing interval that contains the failing vector. This procedure leads to a significant reduction on the size of S. A drawback however is that a small number of failing vectors are not included in S. Nevertheless, as shown in Column 7 and 8 of Table 2, this number is quite small in all cases.

In the second pruning procedure II, we assume that a failing interval contains at most two failing intervals. We now augment Procedure I to continue the search on both branches of the root node at each step during binary search. Once again, the search is restricted to a pre-defined depth. An advantage of II over I is that fewer true failing vectors are dropped from S; see Columns 9 and 10 of Table 2. The drawback is that more post-processing and intermediate signatures are now necessary. Alternative post-processing pruning procedures can also be employed to reduce the candidate set S provided by the interval-based scan-BIST scheme [8]. Even smaller candidate sets of failing vectors (with no loss of information) can be obtained using additional information from the interval signatures, beyond the simple pass/fail interval status that we have used for our experiments.

A direct comparison with related prior work [13] is difficult since we are using different test vector sets in this work. Moreover, the goal of [13] was to obtain a small (incomplete) set of true failing vectors. Here, the primary objective is to retain all failing vectors in the candidate set. A secondary objective is to make the candidate set as small as possible.

5 Conclusion

We have presented a new scan-BIST diagnosis technique for the identification of failing vectors. This technique is based on the concept of overlapping intervals of test vectors within a BIST sequence. The scan-BIST architecture allows us to collect interval signatures without the need for multiple sessions. This is made possible through the use of two MISRs working in an interleaved fashion. We have presented a tight lower bound on the number of true failing vectors. This lower bound is directly obtained from the pass/fail status of the intervals; no additional information is necessary. Experimental results for the ISCAS 89 benchmark circuits show that the basic interval-based scan-BIST method can reduce the size of the candidate failing vectors set S significantly, and all failing vectors are included in S. These results also show that the lower bound on the number of true failing vectors is almost equal to half of the number of failing intervals. Finally, we have presented results for two simple pruning methods that lead to much smaller candidate sets that include almost all the true failing vectors.

Circuit	True failing vectors in 10,000 BIST vectors	$r = L/2$, without pruning				$L = \lfloor 1/2\alpha \rfloor$, $r = L/2$, with pruning			
		$L = \lfloor 1/2\alpha \rfloor$			$L = \lfloor 1/4\alpha \rfloor$	Pruning procedure I		Pruning procedure II	
		No. of suspect failing vectors $\lvert S \rvert$	Lower bound *LB* on failing vectors	No. of failing intervals	No. of suspect failing vectors $\lvert S \rvert$	No. of true failing vectors included in *S*	No. of suspect failing vectors $\lvert S \rvert$	No. of true failing vectors included in *S*	No. of suspect failing vectors $\lvert S \rvert$
S9234	9	1250	5	9	875	8	20	8	20
	36	1900	28	57	850	32	171	33	156
	47	2200	40	73	1200	41	140	46	137
S13207	22	2700	19	38	1178	22	208	22	204
	92	2500	80	148	1152	77	350	85	299
	138	2538	116	217	1269	108	641	126	570
S15850	49	2400	38	70	1300	41	157	45	135
	68	2625	56	106	1314	62	344	67	282
	94	2275	78	147	1152	80	289	87	271
S38417	45	2300	35	60	1125	37	157	41	131
	89	2425	82	153	1080	80	319	84	265
	109	2507	79	153	1344	82	369	95	342
S38584	90	2225	75	141	1116	76	292	86	263
	132	2346	111	209	1233	103	618	111	555
	148	2635	121	229	1413	113	687	135	618
S35932	293	2288	245	462	1188	273	1148	293	1104
	270	2358	236	450	1112	259	1184	270	1036
	293	2456	260	483	1256	282	1232	293	1132

Table 2. Experimental results on failing vector identification using interval-based scan-BIST.

References

[1] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing*, Kluwer Academic Publishers, Norwell, MA, 2000.

[2] P. H. Bardell and W. H. McAnney, "Self-Testing of Multichip Logic Modules", *Proc. Int. Test Conf.*, pp. 200-204, 1982.

[3] P. Song, F. Motika, D. R. Knebel, R. F. Rizzolo and M. P. Kusko, "S/390 G5 CMOS Microprocessor Diagnostics", *IBM Journal of R&D*, vol. 43, No. 5/6, 1999.

[4] Y. Wu and S. Adham, "BIST Fault Diagnosis in Scan-Based VLSI Environment", *Proc. Int. Test Conf.*, pp. 48-57, 1996.

[5] J. Rajski and J. Tyszer, "Fault Diagnosis in Scan-Based BIST", *Proc. Int. Test Conf.*, pp. 894-902, 1997.

[6] I. Bayraktaroglu and A. Orailoglu, "Improved Fault Diagnosis in Scan-based BIST via Superposition", *Proc. DAC*, pp. 55-58, June 2000.

[7] I. Bayraktaroglu and A. Orailoglu, "Diagnosis for Scan-Based BIST: Reaching Deep into the Signatures", *Proc. DATE Conf.*, pp. 102-109, March 2001.

[8] W. H. McAnney and J. Savir, "There is Information in Faulty Signatures", *Proc. Int. Test Conf.*, pp. 630-636, 1987.

[9] J. Savir and W. H. McAnney, "Identification of Failing Tests with Cycling Registers". *Proc. Int. Test Conf.*, pp. 322-328, 1988.

[10] T. R. Damarla, C. E. Stroud and A. Sathaye. "Multiple Error Detection and Identification via Signature Analysis", *JETTA*, vol. 7, pp. 193-207, 1995.

[11] M. G. Karpovsky and S. M. Chaudhry, "Design of Self-Diagnosing Boards by Multiple Signature Analysis", *IEEE Trans. Comp.*, vol. 42, pp. 1035-1044, September 1993.

[12] R. C. Aitken and V. K. Agarwal. "A Diagnosis Method Using Pseudo-random Vectors without Intermediate Signatures", *Proc. Int. Conf. CAD*, pp. 574-580, 1989.

[13] J. Ghosh-Dastidar, D. Das and N. A. Touba, "Fault Diagnosis in Scan-based BIST using Both Time and Space Information", *Proc. Int. Test Conf.*, pp. 95-102, 1999.

[14] B. Chess and T. Larrabee, "Creating Small Fault Dictionaries", *IEEE Trans. CAD*, vol. 18, March, 1999.

[15] I. Pomeranz and S. M. Reddy, "On Dictionary-Based Fault Location in Digital Logic Circuits", *IEEE Trans. Comp.*, vol. 46, pp. 48-59, January 1997.

[16] H. K. Lee and D. S. Ha, "An Efficient Forward Fault Simulation Algorithm Based on the Parallel Pattern Single Fault Propagation", *Proc. Int. Test Conf.*, pp. 946-955, 1991.

[17] Y. Wu and A. Ivanov, "Single-reference Multiple Intermediate Signature (SREMIS) analysis for BIST", *IEEE Trans. Comp.*, vol 44. pp. 817-825, June 1995.