

# Fault Diagnosis in Scan-Based BIST Using Both Time and Space Information

Jayabrata Ghosh-Dastidar, Debaleena Das, and Nur A. Touba

Computer Engineering Research Center  
Department of Electrical and Computer Engineering  
University of Texas, Austin, TX 78712-1084

E-mail: {dghosh, ddas, touba}@cat.ece.utexas.edu

## Abstract

*A new technique for diagnosis in a scan-based BIST environment is presented. It allows non-adaptive identification of both the scan cells that capture errors (space information) as well as a subset of the failing test vectors (time information). Having both space and time information allows a faster and more precise diagnosis. Previous techniques for identifying the failing test vectors during BIST have been limited in the multiplicity of errors that can be handled and/or require a very large hardware overhead. The proposed approach, however, uses only two cycling registers at the output of the scan chain to accurately identify a subset of the failing BIST test vectors. This is accomplished using some novel pruning techniques that efficiently extract information from the signatures of the cycling registers. While not all the failing BIST test vectors can be identified, results indicate that a significant number of them can be. This additional information can save a lot of time in failure analysis.*

## 1. Introduction

Fault diagnosis in a built-in self-test (BIST) environment is an important problem for current technologies. As feature sizes continue to shrink and integration densities continue to increase, more powerful diagnostic tools are needed to reduce the time for failure analysis. BIST allows a large number of test vectors to be applied to the circuit-under-test (CUT) at-speed. The output response of the circuit is compacted using a signature analyzer. If the final signature is incorrect, then the circuit is known to be faulty. The problem being addressed here is how to rapidly diagnose the cause of the faulty behavior.

There are two pieces of information in BIST diagnosis that will be referred to here as *time information* and *space information*. *Time information* is which test vectors applied during the BIST session produced a faulty

response (i.e., the vectors for which the CUT failed). *Space information* is which scan cells in the CUT captured a faulty response during the BIST session. For example, consider the case where 10,000 vectors are applied during the BIST session to a CUT with 200 scan cells. Time information would refer to which of the 10,000 vectors failed, and space information would refer to which of the 200 CUT scan cells captured a faulty response. This paper presents a low-cost approach to obtain *both* time and space information for diagnosis in a scan-based BIST environment.

In a scan-based BIST environment, the output response of the CUT is shifted out of the scan chain and into a serial signature register (or multiple-input signature register, MISR, if there are multiple scan chains). The final signature at the end of the BIST session is so highly compacted that it provides very little time or space information for diagnosis unless the number of errors is only one or two (which is very unlikely). In general, there is no bound on the multiplicity of errors during BIST since a single defect can cause a large number of vectors to produce faulty responses. Thus the only way to obtain useful time or space information for diagnosis without any assumptions on the multiplicity of errors is to add additional hardware and/or get more signatures.

Obtaining space information for BIST diagnosis is much easier than obtaining time information. This is because the number of CUT scan cells is usually much smaller than the number of vectors applied during the BIST session. Recently, two low-cost schemes have been proposed for obtaining space information for BIST diagnosis with no assumptions on the multiplicity of errors. Wu and Adham [Wu 96] proposed a technique that uses a programmable MISR to collect multiple signatures. The MISR is programmed with different polynomials, and the BIST session is repeated for each polynomial to produce a signature. A set of non-linear equations is then solved to identify the set of scan cells that had faulty responses. Rajsiki and Tyszer [Rajsiki 97]

proposed a technique that uses an LFSR to pseudo-randomly mask out different sets of scan cell responses when collecting multiple signatures. The BIST session is repeated and each time a different set of scan cell responses are pseudo-randomly masked out. By analyzing the signatures, the scan cells that had faulty responses can be identified. With both techniques, the maximum number of scan cells with faulty responses that can be identified depends on how many signatures are collected. Neither technique provides any time information for BIST diagnosis. They can only locate the cone of logic where the fault exists. Having time information would allow a much faster and more precise diagnosis.

Previously proposed techniques that can provide time information for BIST diagnosis either are limited in the multiplicity of errors that can be handled [Savir 88], [Stroud 95], or require a very large overhead [Aitken 89], [Karpovsky 93], [Damarla 95]. Identification of the failing test vectors or *time diagnosis* is a challenging problem due to the large number of test vectors applied and the high degree of test response compaction. The number of error sequences that can map to any given faulty signature is well beyond millions in practice [Wu 96]. LFSRs have been used to extract information about failing vectors. [McAnney 87] gives a technique using a single LFSR that guarantees correct diagnosis of single error sequences, [Savir 88] and [Stroud 95] use two LFSRs to diagnose single and double error sequences. [Damarla 95] proposes a method based on error correcting codes (a  $r$ -error correcting BCH code is used). The hardware associated with error correcting codes is high and  $r$  has to be usually kept to 4 or less [Damarla 95]. Thus, there is no effective method for practical time diagnosis. In this paper, we propose a technique which provides a practical solution for the problem.

A low-cost technique that provides both time and space information is presented here. Several techniques for correlating the time information to derive an accurate subset of failing test vectors with a high degree of confidence are described in the paper. Given the set of scan cells that fail and a subset of the failing vectors, standard combinational circuit diagnostic procedures can then be used to precisely locate the fault.

The paper is organized as follows: an overview of the proposed scheme is given in Sec. 2. Details of the time diagnosis scheme are presented in Sec. 3. Section 4 presents experimental results for practical circuits, and Sec. 5 concludes the paper.

## 2. Proposed Scheme

In this section we present the overall scheme for fault diagnosis. Time diagnosis of each output (scan cell) is done using cycling registers as will be explained in detail in Sec. 3. The cycling register signatures are scanned out and the diagnostic computation is done off-line. Time diagnosis based on cycling registers was proposed by Savir and McAnney in [Savir 88]. However, the diagnostic aliasing in [Savir 88] can be very large. In the next section, we propose techniques to substantially reduce diagnostic aliasing and make the approach practical.

A block diagram illustrating the scheme is shown in Fig. 1. One serial signature register and two cycling registers of different sizes are placed at the output of the scan chain. Consider the output response as a matrix where each row corresponds to the output response for one scan vector (i.e., is one scan out) and each column corresponds to a scan cell in the scan chain. Our strategy is to do time diagnosis one column at a time. The response for only one scan cell (selected by a column counter) is shifted into the cycling registers. When the BIST session is finished, the cycling registers contain a signature for only one scan cell. Time information for that scan cell can be extracted from the signature with some *diagnostic aliasing* (i.e., different sets of errors responses will map to the same signature).

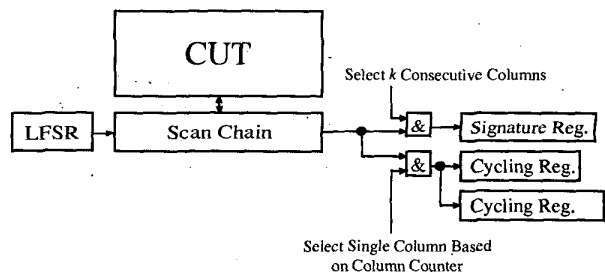


Figure 1. Block Diagram of Diagnosis Scheme

For long scan chains, the number of BIST sessions may be too large if time diagnosis is done for all the scan cells. A "lookahead" register can be used to control the number of BIST sessions required. The response for a small number  $k$  of consecutive scan cells in the scan chain is shifted into the serial signature register during each scan out (e.g.,  $k$  could be 8). The serial signature register contains space information for the next  $k$  scan cells. It serves as a "lookahead" indicating if any of the next  $k$  scan cells need to be analyzed. The signature is compared with the corresponding fault-free signature

stored on-chip. If it is incorrect, then the column counter is incremented by one and time diagnosis is done for each of the  $k$  consecutive columns, otherwise it is incremented by  $k$  to jump ahead to the next set of  $k$  columns. The BIST session is then repeated to collect the next set of signatures. This process continues until a signature in the cycling register has been obtained for each of the scan cells containing errors.

An example is given to illustrate the tradeoff between on-chip signature storage requirements and the number of BIST sessions based on  $k$ . Consider a circuit with 120 output bits. Assume that an internal fault causes errors in bits 12 and 118. The cases are arranged in increasing amount of hardware overhead and corresponding decrease in amount of test application time.

**Case 1:** No "lookahead" signature register is used. Time diagnosis is done for all the output bits. Thus, the number of BIST sessions is 120.

**Case 2:**  $k = 20$ . The number of signatures to be stored for space diagnosis is  $120/20 = 6$ . Time diagnosis has to be done for  $2k + N/k$  output bits. Thus, the number of BIST sessions is  $40 + 6 = 46$ .

**Case 3:**  $k = 8$ . The number of signatures to be stored for space diagnosis =  $120/8 = 15$ . Time diagnosis has to be done for  $2k + N/k$  output bits. Thus, the number of BIST sessions is  $16 + 15 = 31$ .

Note that the extra BIST sessions are only necessary when performing diagnosis. The normal production test procedure would involve only one BIST session as usual.

### 3. Identification of Failing Test Vectors

In the previous section, we presented the overall scheme for collecting data for fault diagnosis. In this section, we explain the procedure for time diagnosis of each faulty scan cell. Identification of failing test vectors with cycling registers as in [Savir 88] is explained briefly. The advantage of this method is that it requires relatively low area overhead compared to other methods, the disadvantage lies in the high diagnostic aliasing. We propose two elegant techniques for pruning the solution space of the cycling registers such that diagnostic aliasing is significantly reduced.

#### 3.1 Time Diagnosis with Cycling Registers

A cycling register is a LFSR whose only feedback is from the last stage to the first stage [Savir 88]. The error sequence from an output is fed into two such registers of length  $m$  and  $n$ , where  $m$  and  $n$  are relatively prime. Further, the product  $mn$  has to be greater than the total test length. The error signature polynomial in the cycling register is equal to the actual error sequence modulo

$(1+x^n)$ . The error positions (or failing vectors) are identified as follows [Savir 88]: Let  $M = (m_1, m_2, \dots, m_u)$  and  $N = (n_1, n_2, \dots, n_v)$  be the error positions in the  $m$ -register and  $n$ -register respectively. Then the following equation has to be solved for all pairs  $(i, j)$  where  $i \in M$  and  $j \in N$ :

$$T - \sigma m - i = T - \tau n - j \quad (1)$$

$T$  is the total test length.  $\sigma$  and  $\tau$  are any two non-negative numbers upper bounded by  $N/m$  and  $N/n$  respectively.

The intuition behind the solution is as follows. Consider any one error position,  $m_i$ , in the  $m$ -register. This error could have been caused by the  $(N-m_i)$ th failing vector or any vector at distance of multiples of  $m$  from this vector. Thus, every error position in the  $m$ -register corresponds to a set of possible failing vectors. Similarly every error position in the  $n$ -register corresponds to a set of possible failing vectors. Eq. (1) yields the intersection of the two sets which is the set of "suspect vectors". Note that some vectors in this suspect set may not actually be failing vectors.

The diagnosis solution is illustrated with an example. Fig. 2 gives the pictorial representation of the example. Let the test length,  $T$ , be 35 and  $m$  and  $n$  are chosen as 9 and 8 respectively ( $mn > T$ ). Let the error sequence entering the cycling registers be such that the failing test vectors are at 1, 6, 8, 13, 19, 20, 28, 32, and 35. Let the error positions in the signature of the  $m$ -register be at  $e_1, e_2, e_3, e_4,$  and  $e_5$  and in the  $n$ -register be at  $e_6, e_7,$  and  $e_8$ . Eq. (1) is solved with these  $(i, j)$  pairs to generate the solution space of suspect vectors. Henceforth, we shall refer to this solution space of suspect vectors as  $S$ . For this example  $S = \{1, 5, 6, 13, 14, 29, 22, 33\}$ . There are three correctly identified failing test vectors out of a total of seven.

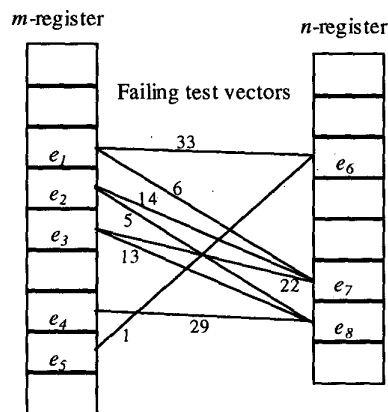


Figure 2. Block Diagram of Time Diagnosis Computation

Fig. 2 illustrates the computation of  $S$ . The error positions in the signature of the  $m$ -register and  $n$ -register are marked on the block diagram of the  $m$ -register and  $n$ -register respectively. The  $(i, j)$  pairs, where  $i \in M$  and  $j \in N$ , that yield a solution from eq. (1) have been shown linked. The diagnosed test vector corresponding to the  $(i, j)$  pair is shown adjacent to the line linking the  $(i, j)$  pair.

Two kinds of aliasing can occur: non-failing vectors may be included in the suspect set (e.g. 5, 14, 22, 29, and 33) and failing vectors may be missing from the suspect set (e.g. 8, 19, 20, 28, 32, and 35). In this paper, we propose methods to minimize the former kind of aliasing. Our aim is to generate a set of suspect vectors that are with very high probability failing vectors. We can afford to miss some failing vectors since identifying even a few failing vectors correctly can greatly aid fault diagnosis. The solution space of suspect vectors generated by eq. (1) is large. Many non-failing test vectors are often contained in this solution space of suspect vectors. We have implemented elegant pruning techniques to significantly decrease the number of non-failing vectors included among the suspects.

### 3.2 Pruning Technique: Step I

The pruning techniques are based on properties of the solution space. Consider any pair  $(i, j)$  where  $i \in M$  and  $j \in N$ . Solving eq. (1) yields a solution for this pair as the failing vector  $T_j$ . Similarly another pair  $(i, k)$  yields the solution as  $T_k$ . Now, the error position  $i$  could have come from only one failing test vector. Thus, only one of  $T_j$  or  $T_k$  is the correct solution, the other one is a spurious solution. Consider the example in Fig. 2.  $(e_1, e_6)$  yields the test vector 33 and  $(e_5, e_6)$  yields the test vector 1. Therefore either 33 or 1 is a failing vector but not both.

The above observation can be formally stated as follows. Consider a bipartite graph where the error positions in sets  $M$  and  $N$  correspond to the two disjoint sets of vertices. Any error position in  $M$  should be matched to a unique error position in  $N$  and vice versa, i.e., we need to find a "maximum matching" in the bipartite graph [Cormen 90]. In our experiments we have observed that there is usually more than one maximum matching possible. One possible maximum matching in Fig. 2 is  $(e_1, e_7)$ ,  $(e_4, e_8)$  and  $(e_5, e_6)$ . Yet another solution is  $(e_1, e_7)$ ,  $(e_3, e_8)$  and  $(e_5, e_6)$ . We use the maximum matching algorithm to generate a subset of  $S$ , which will be referred to as the set  $P$ , that contains diagnosed failing test vectors that have been correctly identified as failing with a high probability. Computation of  $P$  is explained next.

Consider the example in Fig. 2. Error positions  $e_4$  and  $e_5$  in the  $m$ -register have unique matching with error positions  $e_8$  and  $e_6$  in the  $n$ -register, respectively. Test vector 1 is a solution from  $(e_5, e_6)$  and 29 is a solution from  $(e_4, e_8)$ . Thus,  $P = \{1, 29\}$ . Aliasing can cause spurious results since an entry in  $M$  may have a unique matching with an entry in  $N$  and yet the matching may be wrong due to the correct matching having been aliased out. This is the case with test vector 29. The correct error position in the  $n$ -register corresponding to error position  $e_4$  in the  $m$ -register has been aliased out (due to one error feeding back around in the cyclical register just as another error is entering in such that they are exclusive-ORed together and cancel out). This leads to the erroneous conclusion that 29 is a highly probable failing vector.

The first pruning step yields  $P = \{1, 29\}$ . For larger examples, the cardinality of  $P$  may be very large. Note that if there is no aliasing,  $P$  will contain no spurious results. The diagnosed test vectors in  $P$  will be correctly diagnosed ones.

### 3.3 Pruning Technique: Step II

The second pruning step is based on aliasing properties. Note that test vectors that are at a distance of multiples of  $m$  or  $n$  would alias out in one of the cycling registers and thus the solution space should not contain any failing test vectors that are separated by multiples of  $m$  or  $n$ . The pruning based on the maximum matching algorithm yields the set  $P$ .  $P$  represents the set of diagnosed failing vectors that have been diagnosed correctly with a high probability. The second pruning step is therefore to prune out all test vectors in  $S$  that are separated by multiples of either  $m$  or  $n$  from the test vectors in  $P$ .

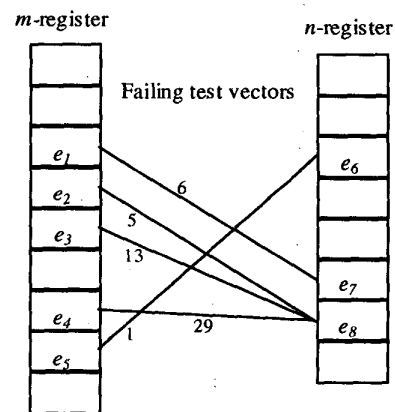


Figure 3. Solution Space After Pruning

Table 1. Experimental Results for Pruning Techniques

Circuit	Without Pruning: [Savir 88]			With Pruning		
	Number of Suspect Vectors $ S $	Failing Vectors in $S$	Non-Failing Vectors in $S$	Number of Suspect Vectors $ S $	Failing Vectors in $S$	Non-Failing Vectors in $S$
S5378	18	14	4	13	13	0
	25	20	5	20	19	1
S9234	39	28	11	27	24	3
	44	26	18	24	22	2
S13207	35	22	13	25	22	3
	57	36	21	34	30	4
S15850	37	26	11	22	22	0
	34	23	11	20	18	2
S38417	26	20	6	20	20	0
	46	34	12	31	28	3

Consider the example in Fig. 2.  $P = \{1, 29\}$ ,  $S = \{1, 5, 6, 13, 14, 22, 29, 33\}$ . 33 would be pruned out because  $(33-1=4*8)$ . Now the error position  $e_1$  in the  $m$ -register has a unique matching with the error position  $e_7$  in the  $n$ -register. This corresponds to the test vector 6. Therefore  $P$  is updated to  $\{1, 6, 29\}$  and  $S$  to  $\{1, 5, 6, 13, 14, 22, 29\}$ . Inclusion of 6 in  $P$  prunes out 14 and 22. Now  $P = \{1, 6, 29\}$  and  $S = \{1, 5, 6, 13, 29\}$ . Fig. 3 illustrates the modified solution space from the cycling registers after the two pruning steps.

Applying a final iteration of the maximum matching algorithm can further modify the solution space. Earlier iterations of the maximum matching algorithm were used to add elements to  $P$ ; no elements were pruned from  $S$ . This final iteration step will prune out all test vectors in  $S$  that have common error positions in the  $m$  or  $n$  cycling registers with the set  $P = \{1, 6, 29\}$ . This prunes out 5 and 13. Note that 13 is actually a failing test vector. It gets pruned out due to the initial erroneous inclusion of 29 in  $P$ . The test vectors remaining in  $S$  after this final pruning constitute our solution space for failing test vectors. In this example, the pruning steps have resulted in the equality of sets  $S$  and  $P$  but this may not happen in the general case.

The example is summarized below. Correctly identified failing vectors are indicated in bold:

Failing test vectors: 1, 6, 8, 13, 19, 20, 28, 32, and 35.

Initial step:  $S = \{1, 5, 6, 13, 14, 22, 29, 33\}$ ,  $P = \emptyset$ .

Before the final pruning by the maximum matching algorithm:  $S = \{1, 5, 6, 13, 29\}$ ,  $P = \{1, 6, 29\}$ .

After the final pruning by the maximum matching algorithm:  $S = P = \{1, 6, 29\}$ .

### 3.4 Pruning: Experimental Results

Table 1 presents experimental results for the case where a single random stuck-at fault was injected in the circuit. Time diagnosis was done for the first faulty output bit encountered. The second column gives the total number of vectors obtained from eq. (1), *i.e.*, the cardinality of the initial solution set of suspect vectors  $S$ . The third and fourth columns give respectively the number of failing vectors and non-failing vectors in the initial set  $S$ . The fifth column gives the cardinality of the final solution set of suspect vectors  $S$ , *i.e.*, the set  $S$  after the pruning steps. The sixth and seventh columns give respectively the number of failing vectors and non-failing vectors in the final solution set  $S$ .

Recall that our aim was to generate a set of suspect vectors that are with very high probability failing test vectors. Table 1 clearly indicates that the number of non-failing vectors in  $S$  is greatly reduced by the pruning steps. The average number of non-failing vectors in  $S$  after pruning (column seven) is around 2 whereas the number of non-failing vectors in  $S$  before pruning (column four) is around 12.

## 4. Practical Solution For Minimizing Aliasing

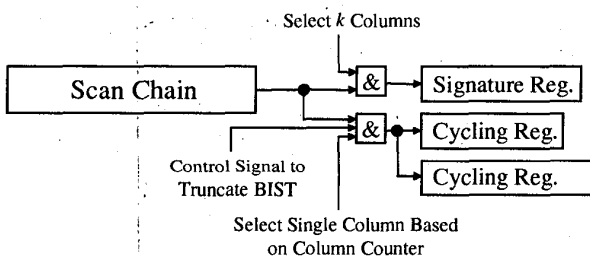
If the number of failing test vectors is large, then errors will cancel out in the cycling registers resulting in a lot of aliasing. In this section, we describe a practical approach for avoiding this problem.

Faults differ widely in their error responses. Easy to detect faults will cause erroneous output values for a large number of test vectors whereas other faults will get activated and propagate to the output for only a few test vectors. There are two extreme cases: a fault at the

primary output will be detected by any test vector that causes an opposite value at the output from the fault type, whereas a random-pattern-resistant fault may get detected only once in the entire test set. Because the number of errors is unpredictable and ranges widely, the average performance of any method (LFSRs, cycling registers, error correction codes) to identify failing test vectors will be poor. Our proposed strategy is to bound the number of failing test vectors that have to be analyzed.

The idea is to take two sets of signatures per output bit being analyzed. One set of signatures is generated for all the test vectors, and one set of signatures for only the first  $tl$  test vectors, where  $tl$  is a parameter that can be chosen based on the circuit-under-test. Our experiments have been performed with  $tl = 250$ .

Truncation of the BIST test set can be done using very simple circuitry. A control signal that is generated when  $tl$  test vectors have been applied to the circuit-under-test can be used to stop the BIST session after the application of  $tl$  test vectors. Alternatively, the control signal can be used to mask out the data entering the cycling registers after the  $tl$ -th BIST clock cycle. Instead of directly connecting the scan element to be diagnosed to the cycling registers, the AND of the scan element and the control signal is connected to the cycling registers. This is illustrated in Fig 4.



**Figure 4.** Collecting Signatures for Only  $tl$  Vectors

This scheme is very effective in reducing aliasing as will be shown in the experimental results. The reasoning behind the scheme is as follows. An easy-to-detect fault will very likely be detected in the first  $tl$  test vectors applied. Thus, the signature from  $tl$  test vectors can be used for diagnosis. However, a fault that does not get detected within the first  $tl$  test vectors is hard-to-detect, and hence will cause relatively few errors. In this case, signatures from the total test set can thus be used for diagnosis without excessive aliasing.

## 5. Experimental Results

Experiments using the techniques described in this paper were performed for some of the ISCAS 89 benchmarks circuits [Brglez 89]. Table 2 shows results where a single random fault was injected in the circuit-under-test in each case. The BIST test length was 10,000 vectors. Two cycling registers of size 101 and 107 were used. Note that one set of cycling registers can be reused when diagnosing each of the scan chains on a chip to save hardware (each scan chain need not have its own set of cycling registers). One possible way to further reduce overhead would be to configure the boundary scan chains to perform as the cycling registers.

Of the 10,000 vectors that were applied, the number of failing vectors (vectors for which the fault caused an error in at least one of the scan cells) is noted in column 2. We have incorporated in our results faults that cause errors in the scan cells a small number of times, moderate number of times, and large number of times. Each row in the table corresponds to a different fault. Column 3 gives the size of the suspect set without any pruning when 10,000 vectors were applied. Column 4 and 5 respectively give the number of failing and non-failing vectors present in the suspect set of column 2. Column 6 gives the suspect set size after the pruning techniques were applied on the suspect set of column 4. Column 7 and 8 respectively give the number of failing and non-failing vectors present in the suspect set of column 6. Note here that even after pruning the number of non-failing vectors in the suspect set is often large.

To get much better diagnostic accuracy, we collect signatures for two cases as was described in the previous section. Once for the full BIST session and once for a truncated BIST session. Column 9 gives the size of the suspect set generated after application of 250 vectors and using our pruning techniques. Column 10 and 11 respectively give the number of failing and non-failing vectors in the suspect set. For example, the second row for ISCAS 89 benchmark circuit s13207 shows that the inserted fault caused a scan cell to fail for 286 vectors. The suspect set without pruning had a size of 2395 out of which only 72 were actually failing vectors and after pruning the size of the suspect set reduced to 50 out of which 3 were actually failing vectors. The next column shows that the size of the suspect set after using the first 250 vectors and our pruning strategy had a size of 11, all of which were failing vectors. Though the number of vectors that caused an error in the output was 286, our final solution had only 11 of them. However, it is

Table 2. Experimental Results With Two Sets of Signatures

Circuit	No. of Failing Vectors in 10,000	Without Pruning			With Pruning					
		Diagnostic Resolution with Test Length = 10,000			Diagnostic Resolution with Test Length = 10,000			Diagnostic Resolution with Test Length = 250		
		No. of Suspect Vectors  S	Failing Vectors in  S	Non-Failing Vectors in  S	No. of Suspect Vectors  S	Failing Vectors in  S	Non-Failing Vectors in  S	No. of Suspect Vectors  S	Failing Vectors in  S	Non-Failing Vectors in  S
S5378	46	836	29	807	28	1	27	3	3	0
	151	2173	45	2128	49	4	45	5	5	0
	794	2637	211	2426	51	6	45	8	7	1
	1012	2215	235	1980	48	5	43	20	19	1
S9234	2	4	2	2	2	2	0	0	0	0
	442	2509	108	2401	47	1	46	6	5	1
	911	2388	256	2132	51	8	43	18	18	0
	1818	2450	453	1997	49	9	40	25	23	2
S13207	2	3	2	1	1	1	0	0	0	0
	286	2395	72	2323	50	3	47	11	11	0
	602	2988	206	2782	56	3	53	15	15	0
	3577	1602	594	1008	41	21	20	21	18	3
S15850	27	547	23	524	23	2	21	1	1	0
	128	1982	31	1951	40	2	38	5	5	0
	1190	2074	246	1828	44	2	42	20	19	1
	2371	2640	685	1955	53	12	41	22	22	0
S38417	1	1	1	0	1	1	0	0	0	0
	43	1142	31	1111	35	1	34	2	2	0
	342	1876	73	1803	42	2	40	13	12	1
	1268	2140	265	1875	46	7	39	20	20	0

important to note here that it is not necessary to identify all the failing vectors. Correctly identifying even a portion of the failing vectors for every fault present is very helpful. It allows further analysis using fault-simulation or critical path tracing [Abramovici 83], to more precisely locate the fault site. This saves a lot of time by reducing the search space for direct probing techniques like E-beam probing. So our objective has been to reduce the number of non-failing vectors in the suspect set as much as possible so that the efforts in identifying the actual defect is minimized and well directed.

### 6. Conclusion

In this paper we have presented a new approach for scan-based BIST diagnosis that provides time information in addition to providing space information. The time information comes in the form of a subset of the failing test vectors. Knowing some of the actual BIST vectors that fail enables a faster and more precise diagnosis.

The proposed technique for diagnosis is non-adaptive. No intermediate signatures have to be collected for on-tester decision making. Thus, this approach can be used for field diagnosis where the signatures are analyzed elsewhere. Further, the proposed approach requires small hardware overhead. Only two cycling registers are required for time diagnosis. The BIST session has to be run twice per scan cell if time diagnosis is done for all the scan cells. If the BIST running time is an issue, the "lookahead" operation with the serial signature register can be used to reduce test time. This will require some additional hardware overhead to store signatures on chip.

The proposed technique for time diagnosis can also be used with any of the existing techniques for identifying faulty scan cells. In this scenario, the diagnosis scheme would be adaptive. Information about the faulty scan cells would be passed on to the time diagnosis step. Time diagnosis could then be done only for the faulty scan cells.

## Acknowledgements

This material is based on work supported in part by the Defense Advanced Research Projects Agency (DARPA) under Contract No. DABT63-97-C-0024, and in part by the National Science Foundation under Grant No. MIP-9702236, and in part by the Texas Advanced Research Program under Grant No. 1997-003658-369.

## References

- [Abramovici 83] Abramovici, M., P.R. Menon, and D.T. Miller, "Critical Path Tracing - An alternative to Fault Simulation", *Proc. of 20<sup>th</sup> Design Automation Conference*, pp. 214-220, 1983.
- [Aitken 89] Aitken, R.C., and V.K. Agarwal, "A Diagnosis Method Using Pseudorandom Vectors Without Intermediate Signatures," *Proc. of International Conference on Computer-Aided Design (ICCAD)*, pp. 574-577, 1989.
- [Brglez 89] Brglez, F., D. Bryan, and K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits," *Proc. of Int. Symp. on Circuits and Systems*, pp. 1929-1934, 1989.
- [Cormen 90] Cormen, T.H., C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms*, The MIT Press, 1990.
- [Karpovsky 93] Karpovsky, M.G., and S.M. Chaudhry, "Design of Self-Diagnostic Boards by Multiple Signature Analysis," *IEEE Trans. on Computers*, Vol. 42, No. 9, pp. 1035-1043, Sept. 1993.
- [Darmala 95] Darmala, T.R., C.E. Stroud, and A. Sathaye, "Multiple Error Detection and Identification via Signature Analysis," *Journal of Electronic Testing: Theory and Applications*, Vol. 7, No. 3, pp. 193-207, 1995.
- [McAnney 87] McAnney, W.H., and J. Savir, "There is Information in Faulty Signatures," *Proc. of International Test Conference*, pp. 630-636, 1987.
- [Rajski 97] Rajski, J., and J. Tyszer, "Fault Diagnosis in Scan-Based BIST," *Proc. of International Test Conference*, pp. 894-902, 1997.
- [Stroud 95] Stroud, C.E., and T. Damarla, "Improving the Efficiency of Error Identification Via Signature Analysis," *Proc. of VLSI Test Symposium*, pp. 244-249, 1995.
- [Savir 88] Savir, J., and W.H. McAnney, "Identification of Failing Tests with Cycling Registers," *Proc. of International Test Conference*, pp. 322-328, 1988.
- [Wu 96] Wu, Y., and S. Adham, "BIST Fault Diagnosis in Scan-Based VLSI Environment," *Proc. of International Test Conference*, pp. 48-57, 1996.