Scan-Based BIST Diagnosis Using an Embedded Processor

Kedarnath J. Balakrishnan and Nur A. Touba

Computer Engineering Research Center
Department of Electrical and Computer Engineering
University of Texas at Austin

Email: {kjbala,touba}@ece.utexas.edu

Abstract

For system-on-chip designs that contain an embedded processor, this paper present a software based diagnosis scheme that can make use of the processor to aid in diagnosis in a scan-based built-in self-test (BIST) environment. The diagnosis scheme can be used to determine both the scan cells that capture errors as well as the failing test vectors during a BIST session thereby allowing a faster and more precise diagnosis. The BIST session needs to be run only once for diagnosis. The scheme is based on pseudo-random linear compaction of the output response bits. The proposed scheme uses word based linear operations that can be implemented very fast and efficiently on the embedded processor. Experimental results indicate that the proposed scheme is very powerful in identifying the failing test vectors and performs better than previous methods both in terms of the suspect set size and the accuracy of diagnosis.

1. Introduction

Complex system-on-a-chips (SoC) containing many cores are becoming ubiquitous due to advances in design technology. Most complex SoCs have an embedded processor core and memory. The processing power of the embedded processor can be harnessed to help in the testing and diagnosis of other cores in the SoC.

Because of the increasing use of intellectual property (IP) cores in SoC designs, most cores come with their own test requirements. Therefore, the same SoC might have some cores that require deterministic testing and others with pseudo-random scan-based BIST. In scan-based BIST, the patterns are generated on-chip and shifted through the scan chains, and then the output response is captured in the scan chains and compacted on-chip using a multiple input shift register (MISR) to generate a single signature. Since a large number of patterns are applied and the output response is compacted very highly into a signature, it only provides pass/fail information and as such contains very little diagnostic information.

Fault diagnosis to determine the location and cause of failures is very important during the initial manufacturing process. It helps in the identification of the manufacturing defects and is also used for yield learning to improve the production quality. Diagnostic information in scan-based BIST can be classified into two categories: *space information* which is the set of scan cells that capture the faulty responses, and *time information* which is the set of test vectors that fail.

Extracting space information during scan-based BIST is much easier than identifying the failing test vectors since the length of the scan chain is typically much smaller than the number of test vectors that are applied during BIST. There has been a lot of work in the past for identifying the scan cells that capture faulty information. Most involve intelligently collecting signatures over multiple BIST sessions and analyzing them later. In [Wu 96], a programmable MISR is used to collect multiple signatures with the MISR programmed with a different polynomial each time. The scan cells that had faulty responses are then identified by solving a set of non-linear equations. Other techniques involve partitioning the scan cells into different

Proceedings of the 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'03)
1063-6722/03 $17.00 © 2003 IEEE

groups and observing each partition separately. In [Rajski 99], an LFSR is used to pseudo-randomly mask out a set of scan cells. This process is repeated over multiple sessions and in each session a different set of scan cells are masked. This idea was improved upon by [Bayraktaroglu 00a] by using the principle of superposition. Properties of high quality partitions were identified in [Bayraktaroglu 00b] for this process and instead of randomly partitioning the scan cells, a deterministic partitioning technique was proposed. In this case, the improvement in accuracy involved additional hardware overhead. One drawback of all these methods is that the number of signatures collected determines the maximum number of scan chains with faulty responses that can be identified.

Only the cone of logic where the fault exists can be located by the information provided by identifying the scan cells that capture faults. Identifying which test vectors fail can provide much more information and hence much faster and more precise diagnosis. Most techniques proposed earlier for getting time information require additional hardware for diagnosis and are limited either by multiplicity of errors that can be handled or the hardware overhead. Methods using LFSRs were proposed by [McAnney 87], [Savir 88], and [Stroud 95]. A single LFSR is used in [McAnney 87] that guarantees correct diagnosis of single error sequences while two LFSRs are used in [Savir 88] and [Stroud 95] to diagnose single and double error sequences. Methods based on error correcting codes were proposed in [Karpovsky 93] and [Damarla 95]. An approach that does not require intermediate signatures was presented in [Aitken 89]. [McAnney 87], [Savir 88] and [Stroud 95] are limited by the multiplicity of errors that can be handled while [Aitken 89], [Karpovsky 93] and [Damarla 95] require very high hardware overhead. [Ghosh-Dastidar 99] presented a method that combines cyclic registers with pruning techniques to identify the failing test vectors and provide better diagnostic resolution than [Savir 88]. [Liu 02] presented a method for identifying failing vectors based on the use of overlapping intervals of test vectors. Signatures are collected for each interval of consecutive test vectors and the signatures are analyzed using pruning techniques to identify the set of candidate failing vectors. This method can require the collection of a large number of signatures.

This paper investigates the use of an embedded processor to aid in diagnosis for scan-based BIST. In contrast to all the previous schemes, the proposed method is software based and requires very little additional hardware overhead. The proposed method can be implemented with a small number of instructions. The proposed method provides both time and space information and requires that the BIST session be run only once. Note that most previous methods require that the BIST session be run a large number of times to collect many signatures.

2. Proposed Scheme

In this section, the proposed scheme for response compaction of BIST vectors is described. The response compaction is based on pseudo-randomly compacting the output response. It can be considered as the dual of the pseudo-random linear expansion scheme that was used to decompress deterministic test vectors in [Balakrishnan 03]. It can be efficiently implemented in software through linear operations.

Figure 1 shows the architecture of a typical SoC. It consists of multiple cores with different types of DFT techniques for each. It also consists of an embedded processor and embedded system memory connected to the different cores though a system bus. The embedded processor can be used to test the different cores in the SoC. On the input side, it can be used as an on-chip decompressor that decompresses the compressed deterministic test vectors coming from the tester or it can be used as a pseudo-random pattern generator for BIST. It can also be used for compacting the output response of the cores. For a BISTed core, the MISR performs very high compaction with negligible aliasing. Since the MISR signature has very little diagnostic information, we propose to use the embedded processor for diagnosis in BIST.

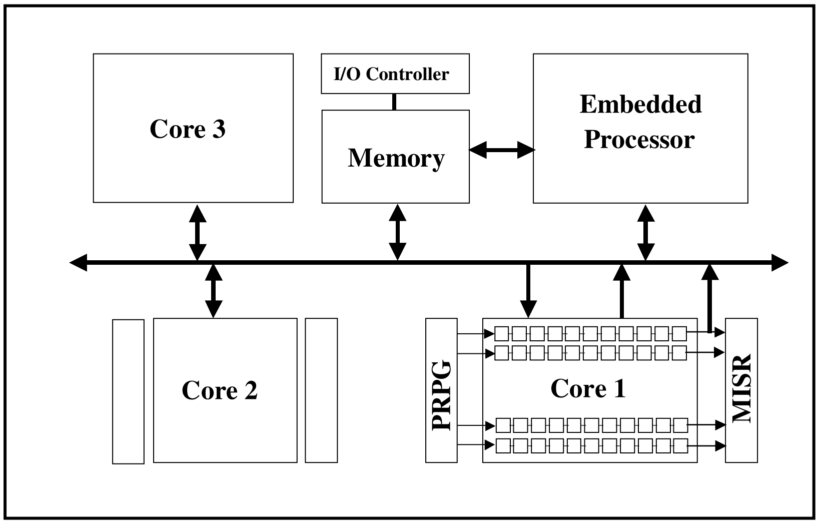

Figure 1. Architecture of a typical SoC

Consider *Core1* shown in Fig. 1 as the circuit-under-test (CUT). During normal testing, the scan-BIST scheme is working with the pseudo-random pattern generator (PRPG) generating the patterns and sending it to the scan chains and the MISR compacting the output responses. If the MISR signature doesn't match with the fault-free signature, then an error is detected. At this point, the diagnosis process can be initiated. During diagnosis, the same test vectors are applied by the PRPG, but this time the processor also reads the output responses. A software program running on the processor compacts the output responses and stores them on the system memory. The compacted output is then analyzed to get information about the failing vectors and the scan chains that capture faults.

The software program stores a small number of output responses at the beginning uncompacted, and then compacts the remaining output responses. The reason for leaving some output responses uncompacted is to help in diagnosing faults that are detected by a large number of vectors. Such faults have a large number of errors in the output response and thus having just a small set of uncompacted output responses is sufficient for diagnosing them. However, for the faults that are detected by very few vectors, it is difficult to find the vectors for which they fail in the long BIST sequence, and hence it is for these faults that the proposed compaction procedure is needed to aid the diagnosis process.

The response compaction algorithm is based on pseudo-randomly XORing the output responses together to form a much smaller compacted set. The steps of the compaction algorithm are:

1. Allocate a set of memory locations to store the compacted data and initialize them to zeros.
2. Read the output response word by word. Multiple scan chains (equal to word size of processor) can be read in parallel.
3. For each output response word, do the following multiple times
   - Randomly choose a location in allocated memory
   - Rotate the response word by a random number and
   - XOR the response word with the chosen memory location

The number of times each word is XORed together is represented by *numxors* and will determine the running time of the diagnosis program. The random number generation is done using the Mitchell and Moore generator [Knuth 97]. The generator produces a random number in $Z_m=\{0,1,2\ldots,m-1\}$ given by the equation:

$$X_n= (X_{n-24}+ X_{n-55}) \bmod m, \qquad n \geq 55.$$

It is an additive generator with very good random properties and can be efficiently implemented on a processor [Balakrishnan 03].

3. Diagnosis

The proposed pseudo-random linear compaction procedure can be represented by a binary matrix $\boldsymbol{A}$ (each entry is *0* or *1*), with one column for each bit in the output response and one row for each bit in the compacted response. The entry $\boldsymbol{a_{ij}}$ will be *1* if the output response bit corresponding to the column *j* is XORed into the compacted response bit corresponding to row *i*. For example, if the random number generator generates locations 3, 5 and 8 to XOR the first output response bit, then the entries $\boldsymbol{a_{31}}$, $\boldsymbol{a_{51}}$, *and* $\boldsymbol{a_{81}}$ will be *1*. All other entries in the first column will be *0*. Since the number of non-zero entries in each column is equal to *numxors*, it is much smaller than the number of rows, thus the diagnosis matrix $\boldsymbol{A}$ is a very sparse matrix. If the output response is represented by vector $\boldsymbol{r}$, then the compacted response, $\boldsymbol{c_r}$, can be constructed as $\boldsymbol{c_r} = \boldsymbol{A}\,\boldsymbol{r}$.

During diagnosis, the faulty output response is compacted using the procedure discussed in Sec. 2 to get the compacted response vector $\boldsymbol{c_{faulty}}$. This is then compared with the compacted fault-free output response vector, $\boldsymbol{c_{ff}}$, to get the compacted difference or error vector, $\boldsymbol{e} = \boldsymbol{c_{faulty}} \oplus \boldsymbol{c_{ff}}$. The identification of the location of bits in error for diagnosis can then be represented in the form of a matrix equation $\boldsymbol{A}\ \boldsymbol{x} = \boldsymbol{e}$, where $\boldsymbol{x}$ is the vector indicating the location of the bits of the output response that are in error. If an element of $\boldsymbol{e}$ is *1* (the faulty and fault free compacted bits differ) then an odd number of non-zero columns corresponding to that row of $\boldsymbol{A}$ are in error. Similarly, if an element of $\boldsymbol{e}$ is *0*, then either zero (no errors) or an even number of non-zero columns corresponding to that row of $\boldsymbol{A}$ are in error.

This matrix equation is actually a set of simultaneous linear equations in the variables of $\boldsymbol{x}$ that need to be solved to get the original output response. Since the number of equations is much smaller than the number of variables, the solution space for this set of equations will be enormous. The number of solutions depends on the compaction ratio. The higher the compaction ratio, the more the number of solutions. Since the solution space is enormous, a heuristic based approach is used to reduce it. The proposed heuristic is based on the fact that a lower number of errors is more probable than a higher number of errors. That is, if an erroneous compacted response bit (row) can be explained by errors in either 1 column or 3 columns, the 1 column error is a more probable solution and hence will be selected by our scheme. The reason why a lower number of errors is more probable than a higher number of errors is explained as follows. As was discussed in Sec. 2, a small set of output responses are initially left uncompacted. The faults that cause a large number of errors can be diagnosed using only these uncompacted responses. This leaves only the faults with a few number of errors for which diagnosis from the compacted responses is necessary. For these faults, the proposed heuristic is applicable.

If the location of all non-zero bits in a column match the rows in error (in the vector $\boldsymbol{e}$), we add the column (and hence the corresponding bit in the output response) to the suspect set. Note that the suspect set contains all the bit locations in the output response that are suspected to be in error. Consider the example shown in Fig. 2 where 12 output response bits are compacted to 8 bits. Assume that the difference vector between the faulty and fault-free compacted responses, ***e***, is as given in the figure. Since the position of 1's in the second column match with the position of 1's in the ***e*** vector (shown in bold in the figure), our method will add the output response bit corresponding to column 2 to the suspect set.

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & \mathbf{1} & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1} \\ 0 \\ 0 \\ \mathbf{1} \\ \mathbf{1} \\ 0 \\ 0 \end{bmatrix}$$

Figure 2. Identification of incorrect bits

Note that if a row has an even number of errors, the errors will cancel out. So some columns that are in error will not match in the rows in error because the errors in one or more of the rows may have cancelled out. Thus some output response bits that are in error may not appear in the suspect set. Also, it is possible for a column that is not in error to match all the rows in error. This can happen if the column vector is equal to a linear combination of other column vectors that are in error. However, since the number of bits in error is very small compared to the number of bits not in error, the probability of this happening is low. Thus the proposed heuristic is generally very effective at identifying a suspect set that accurately contains output response bits that are in error (which in turn identifies failing vectors and scan cells that capture errors). While not all output response bits that are in error will appear in the suspect set, having even a subset is very useful to guide the failure analysis process and greatly speed it up.

Note also that the performance of the diagnosis procedure depends on the parameter *numxors* which is the number of XOR operations that are performed for each word of the output response. The parameter *numxors* determines the number of non-zero entries in each column. The higher the value of *numxors*, the greater the accuracy of the diagnosis since the requirements for adding a column to the suspect set increase. However, the size of the suspect set goes down. Thus, the value of *numxors* can be used to tradeoff a more inclusive suspect set versus a more accurate one. One approach to increase the effectiveness of the proposed diagnostic scheme would be to run the BIST session multiple times using different values of *numxors* and then compare the resulting suspect sets.

Since the algorithm performs the check for each column of the matrix, the running time depends on the number of columns. In fact, the running time is *O(columns)* since the checking is done a linear number of times for each column.

4. Experimental Results

We have performed several experiments to validate the proposed diagnostic scheme. The experiments can be classified into two categories; in the first set of experiments, we randomly inject errors into output response and simulate our diagnosis scheme, while the second set of experiments consists of injecting faults into the ISCAS 89 circuits and diagnosing the compacted output response.

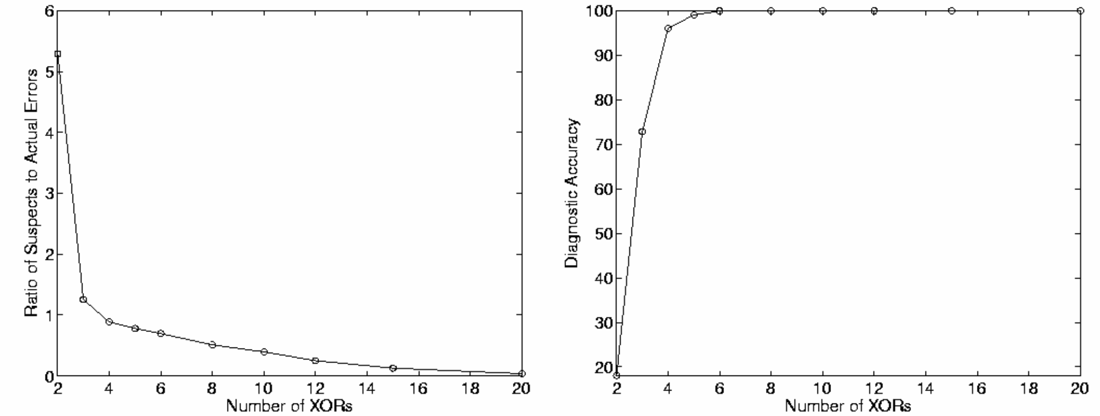

Figure 3. Variation of suspect set size and diagnostic accuracy with *numxors*

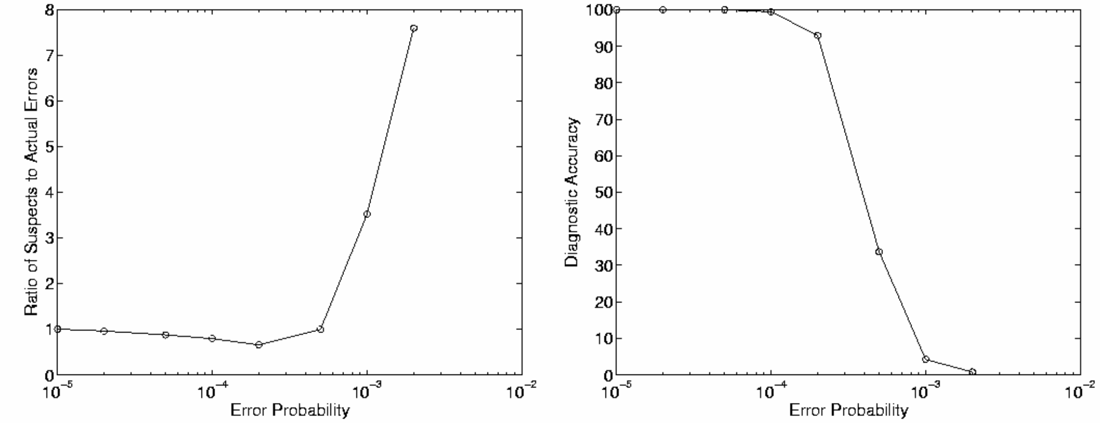

Figure 4. Variation of suspect set size and diagnostic accuracy with error probability

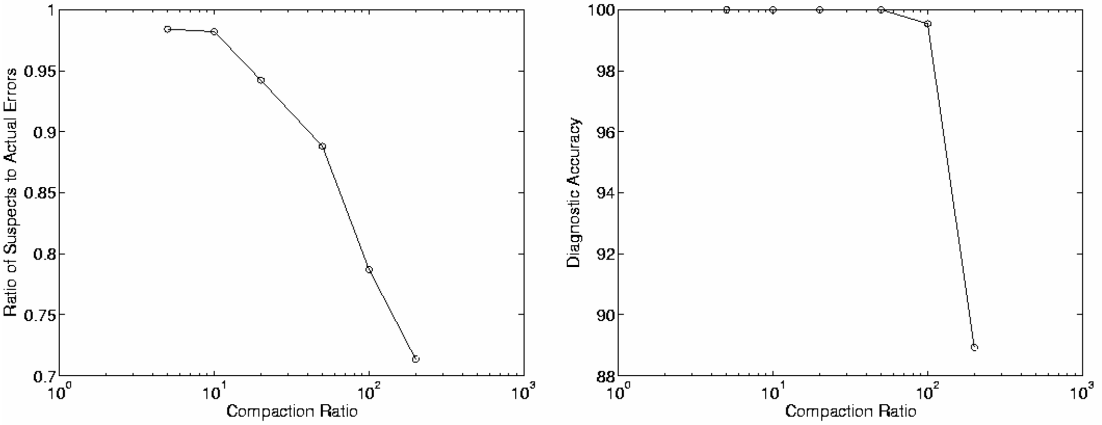

Figure 5. Variation of suspect set size and diagnostic accuracy with compaction ratio

The first set of experiments was done to study the impact of *numxors*, error probability (ratio of bits in error to total number of bits), and compaction ratio (ratio of uncompacted response size to compacted response size) on the number of suspects and diagnostic accuracy of our method. The results are shown in the following graphs. In Fig. 3, *numxors* is varied while the compaction ratio was held constant at 100 and the error probability was held constant at 0.001. As *numxors* increases, the ratio of suspects to actual errors decreases while the diagnostic accuracy increases. The reason for this was explained in Sec. 3. In Fig. 4, error probability is varied while the compaction ratio was held constant at 100 and *numxors* was held constant at 5. As the number of errors increases, the diagnostic accuracy initially remains relatively constant, but then drops off as the error probability exceeds 0.0002 in this case. The size of the suspect set decreases and reaches a minimum when the error probability is equal to 0.0002. By changing *numxors*, this inflection point can be moved. In Fig. 5, compaction ratio is varied while *numxors* was held constant at 5 and the error probability was held constant at 0.001. As the compaction ratio increases both the number of suspects and diagnostic accuracy decrease.

A second set of experiments was done on the larger ISCAS 89 benchmark circuits. A single random stuck-at fault was injected into the CUT in each case and a BIST sequence of 10,000 patterns was simulated. The results are reported in Table 1. Even though we simulated our method for a large number of faults, we report results here for a small set of representative faults. We have incorporated faults that cause various numbers of errors. We implemented the cyclic registers method described in [Savir 88] and the pruning techniques for it described in [Ghosh-Dastidar 99]. Column 2 shows the number of failing test vectors out of the 10,000 random test vectors applied. Columns 3, 4, and 5, respectively give the number of suspect vectors |S|, the number of failing vectors correctly included in the suspect set, and the number of non-failing vectors included in the suspect set for [Savir 88] for a particular fault. Columns 6-8 and columns 9-11 give the respective numbers for [Ghosh-Dastidar 99] and the proposed scheme for the same fault.

Table 1. Experimental results for ISCAS '89 benchmark circuits

Circuit	No. of Failing Vectors in 10,000	[Savir 88]			[Ghosh-Dastidar 99]			Proposed Scheme		
		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	9	74	9	65	9	2	7	9	9	0
	50	1253	32	1221	0	0	0	46	46	0
	187	2104	44	1960	43	1	42	148	146	2
	404	2481	102	2379	46	3	43	435	245	190
s9234	2	4	2	2	2	2	0	2	2	0
	23	111	10	101	10	2	8	22	22	0
	74	45	7	38	7	1	6	67	67	0
	161	23	5	18	5	3	2	122	122	0
s13207	4	15	4	11	4	1	3	4	4	0
	44	1052	30	1022	30	0	30	44	44	0
	276	2217	80	2137	45	2	43	268	261	7
	453	581	43	538	23	1	22	367	364	3
s15850	5	22	5	17	5	0	5	5	5	0
	43	740	18	722	23	1	22	41	41	0
	103	1123	28	1095	33	1	32	101	101	0
	385	3116	119	2997	57	1	56	348	348	0

As can be seen in Table 1, the proposed scheme performs better than the other methods for all the circuits both in terms of the size of the suspect set and the diagnostic accuracy. The number of non-failing vectors (last column of Table 1) in the suspect set for the proposed scheme is usually much smaller than the number of failing vectors. For all the faults (except the fourth row for circuit s5378), the number of non-failing vectors included in the suspect set is less than 8. This demonstrates the accuracy of our method for actual stuck-at faults. Note that in this comparison, the same amount of compacted response storage is used for all the techniques, so the improvement in the results is purely based on the algorithm and not simply due to using more storage.

5. Conclusions

In this paper, a new diagnosis scheme for a scan-based BIST environment was presented which uses the power of an embedded processor to help in the diagnosis. The scheme requires that the BIST session be run only once and provides both space and time information for failure analysis. Experimental results validate the claim that the proposed scheme performs better than previously methods for identifying failing test vectors.

Acknowledgements

This material is based on work supported in part by the Intel Corporation and in part by the National Science Foundation under Grant No. CCR-0306238.

References

[Aitken 89] Aitken, R.C., and V.K. Agarwal. “A Diagnosis Method Using Pseudo-random Vectors without Intermediate Signatures”, *Proc. of Int. Conf. on Computer-Aided Design*, pp. 574-580, 1989.

[Balakrishnan 03] Balakrishnan, K.J., and N.A. Touba, "Deterministic Test Vector Decompression in Software Using Linear Operations", *Proc. of VLSI Test Symposium*, pp. 225-231, 2003.

[Bayraktaroglu 00a] Bayraktaroglu, I., and A. Orailoglu, "Improved Fault Diagnosis in Scan-based BIST via Superposition", *Proc. of Design Automation Conference*, pp 55-58, 2000.

[Bayraktaroglu 00b] Bayraktaroglu, I., and A. Orailoglu, "Deterministic Partitioning Techniques for Fault Diagnosis in Scan-Based BIST", *Proc. of International Test Conference*, pp. 273-282, 2000.

[Damarla 95] Damarla, T.R., C. E. Stroud, and A. Sathaye. "Multiple Error Detection and Identification via Signature Analysis", *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 7, pp. 193-207, 1995.

[Ghosh-Dastidar 99] Ghosh-Dastidar, J., D. Das, and N. A. Touba, "Fault Diagnosis in Scan-Based BIST Using Both Time and Space Information", *Proc. of International Test Conference*, pp. 95-102, 1999.

[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.

[Knuth 97] Knuth, D.E. *The Art of Computer Programming*, Vol. 2. Addison-Wesley, Reading, MA, 3rd edition, 1997. ISBN: 0201896842.

[Liu 02] Liu, C., K. Chakrabarty and M. Goessel, "An Interval-Based Diagnosis Scheme for Identifying Failing Vectors in a Scan-BIST Environment", *Proc. of Design, Automation and Test in Europe (DATE) Conference*, pp. 382-386, 2002.

[McAnney 87] McAnney, W.H., and J. Savir. “There is Information in Faulty Signatures”, *Proc. of International Test Conference*, pp. 630-636, 1987.

[Rajski 99] Rajski, J., and J. Tyzer, "Diagnosis of Scan Cells in BIST Environment", *Proc. IEEE Transactions on Computers*, Vol. 48, No. 7, pp. 724-731, July 1999.

[Savir 88] Savir, J., and W. H. McAnney, "Identification of Failing Tests with Cyclic Registers", *Proc. of International Test Conference*, pp. 322-328, 1988.

[Stroud 95] Stroud, C.E., and T. R. Damarla. “Improving the Efficiency of Error Identification via Signature Analysis”, *Proc. of VLSI Test Symposium*, pp. 244-249, 1995.

[Wu 96] Wu, Y., and S. Adham, "BIST Fault Diagnosis in Scan-Based VLSI Environment", *Proc. of International Test Conference*, pp 48-57, 1996.