On Reducing Aliasing Effects and Improving Diagnosis of Logic BIST Failures

Ramesh C. Tekumalla
Sun Microsystems, Inc.
1 Network Drive
Burlington, MA 01803

Abstract

Diagnosing failing vectors in a Built-In Self Test (BIST) environment is a difficult task because of the highly compressed signature coming out of the Multiple Input Shift Register (MISR). The root cause of the failure must be initially narrowed down to the failing vectors and also the scan cells at which mismatches occurred. In this work, we propose a method for accurately determining the first cycle at which incorrect responses were captured in the scan chains along with the scan cells that captured the incorrect responses. We define a diagnosis methodology that describes a simple way of diagnosing one chain at a time and further shows that accurate diagnosis is possible with the aliasing nature of the MISR having no impact on diagnosis. The proposed technique results in identifying the failing vectors along with the mismatching scan cells. After determining the failing vectors and mismatching scan cells, we extend the method to identify potential faults in the circuit that may have resulted in the scan cells capturing incorrect responses. The method presented in this paper can be applied to diagnose multiple failing vectors also.

1 Introduction

Current application specific integrated circuit (ASIC) designs usually employ BIST for achieving their target fault coverage. There are a few reasons for using BIST in such designs and the most popular ones being at-speed testing of the circuit, avoiding the burden of generating test patterns and manipulating them for tester needs and the consistency in testing the chip at the wafer-level and at the system level. Most of the circuits with BIST have multiple scan chains that are driven by a pseudo-random pattern generator. The patterns being shifted out by the scan chains are fed into a MISR which compresses the values coming out of the scan chains and generates a signature. At the end of the BIST session, the generated signature is compared with the expected signature and the test is classified as a pass or fail depending on whether the actual signature is the same as the expected signature or otherwise.

The goal of BIST diagnosis is to determine the reason for the faulty signature that came out of the MISR. The root cause of this problem is a faulty response that was captured is some scan cells for some specific BIST patterns. Thus, the BIST diagnosis capability should be able to identify BIST patterns that caused the faulty response to be captured in the scan cells and also locate potential sites in the circuit, where faults could have caused the incorrect values to be captured in some scan cells. In this work, we make the following assumptions: the circuit is fully scanned with multiple scan chains. The number of bits in the pseudo-random generator shift register (LFSR) is equal to the number of scan chains in the design and the MISR is of the same length as the LFSR. The scan chains are driven by one bit each of the LFSR and the last scan-out of each scan chain drives one bit each of the MISR. A BIST pattern consists of shifting in all the scan cells in all the scan chains, followed by a complete shift-out of all the scan cells through the MISR. We consider only the stuck-at fault model for our experimentation purposes. The scan chain lengths are balanced as far as possible.

Original work on diagnosing scan based designs was presented in [1, 2]. The latter proposed a technique for setting the scan cells to specific values and collect the diagnosis information while the scan chain is unloaded. A method for identifying failing BIST vectors using error correcting codes was presented in [3]. The extra hardware needed for doing this was considerably high. A programmable MISR for collecting multiple signatures using multiple polynomials was presented in [4]. A fault simulation approach for locating multiple design errors is presented in [5]. This approach is the most fundamental method for diagnosing any kind of test vectors including BIST and scan vectors among others. A method for pseudo-randomly masking out the responses from different scan chains was presented in [6]. By analyzing the signatures involving a fewer set of scan cell

0-7803-8106-8/03 $17.00 Copyright 2003 IEEE

responses, the faulty responses are identified. The above approach involves some processing of the signatures leading to diagnostic complexity.

A method using implicit enumeration of lines in the circuit for multiple error diagnosis is presented in [7]. Appropriate intersections of lines on the paths that contribute to multiple errors are determined to diagnose the failing vectors. A technique using additional hardware for diagnosing scan errors is presented in [8]. Tests are applied several times to gather multiple signatures and determine the flip-flops driven by erroneous signals. A method for modeling the open defects and diagnosing multiple errors is presented in [9]. A method for diagnosing multiple errors affecting the same output is presented in [10]. A method that gradually determines the actual set of failing scan cells is proposed in [11]. Starting from an initial complete set of scan cells, the set is pruned to determine the non-faulty and certainly faulty cells. It is shown that the diagnosis time with this technique is reduced compared with earlier approaches. A three-step procedure for diagnosing scan chains is presented in [12]. Additional patterns are generated to narrow down the suspect set of failing scan cells. Stuck-at and transition type of failing defects are handled by this technique. A fast technique for eliminating false candidates during diagnosis is presented in [13]. This is achieved by techniques including X-propagation from fanout stems and flipping the values at the outputs.

The motivation for the work proposed in this paper is the need for an efficient hardware-software approach that can be easily deployed in an industrial environment. This solution must involve minimal circuit modifications and also enable a software solution to quickly determine the failing vectors and root cause the failures to a set of fault locations that caused the failures. The proposed technique requires circuit modifications only in the MISR circuitry, not affecting the timing of the functional logic. The software part of the solution post processes the results from the BIST fault simulation tools to determine the potential faults in the circuit that caused the failures. The proposed solution will work for any fault model that the BIST tools use for fault coverage analysis. In this effort, the different tools and techniques available in the industry for BIST vector generation, fault simulation and cone extraction are integrated together to create a practical solution that can be quickly deployed for analyzing large industrial designs.

The key contributions of this work are described below. We propose a method for diagnosing logic BIST failures in a multiple scan chain environment. The logic going into the MISR is modified such that each scan chain can be individually diagnosed. A one-hot controller is designed such that it selects one scan chain at a time that is targeted for diagnosing the incorrect values captured in that chain. An extra test signal ensures that the scan chain values go through the MISR unmodified and hence ensuring that the signature coming out of the MISR, sampled at specific time instants, is the same as the values captured in the corresponding scan chain. The above process, as we later show, helps us in identifying the vector number and scan cell(s) at which the mismatches(s) occurred during the BIST session. We follow this up by identifying the set of stuck-at faults that are responsible for the failing BIST test. This is done by identifying the faults detected by the failing vector and eliminating those that were detected by vectors before the first failing vector and also those that are detected by the failing vector at the scan cells that do not have any mismatches. Experiments are performed on industrial test cases to show the effectiveness of the proposed technique. We show that the proposed solution can be easily incorporated into the currently available tools, techniques and processes available in the industry.

The paper is organized as follows. Section 2 discusses the basic concepts of the BIST environment and also introduces some specifics about the circuits used in this work. A design enhancement technique that uses a one-hot finite state machine for controlling which scan chain to diagnose, is introduced in Section 3. This is followed up by the methodology for identifying the failing vectors and scan cells that have incorrect values captured in them. A method for identifying specific faults in the circuit that must have contributed to the failures is also presented. Experimental results on some industrial circuits are presented in Section 4. These results include additional hardware requirements for diagnosing one scan chain at a time and also the statistics of the number of faults that must be analyzed for debugging the failing vectors. Conclusion and future work are presented in Section 5.

2 Background

The basic BIST architecture in a multiple scan chain environment is as shown in Fig. 1. A pseudo-random generator determines the patterns to be loaded into the multiple scan chains. All the scan chains are loaded in parallel. The number of cycles needed to load all the scan chains is equal to the number of flip-flops in the largest scan chain. The output of each flip-flop in the pseudo-random generator is fed into one scan chain input. A BIST vector constitutes one complete scan-in of all the scan chains followed by one system clock and then a complete shift-out of the values captured in the scan chains through the MISR. The final scan-out from each scan chain is fed into a MISR and the signature generated in the MISR after each BIST vector is very highly compressed. A serial output from the MISR is used to shift out the compressed signature, which is compared with the expected signature to determine whether or not the BIST session passed. The logic BIST controller, which constitutes the pseudo-random generator and its associated circuitry, generates patterns at speed and the shifting through

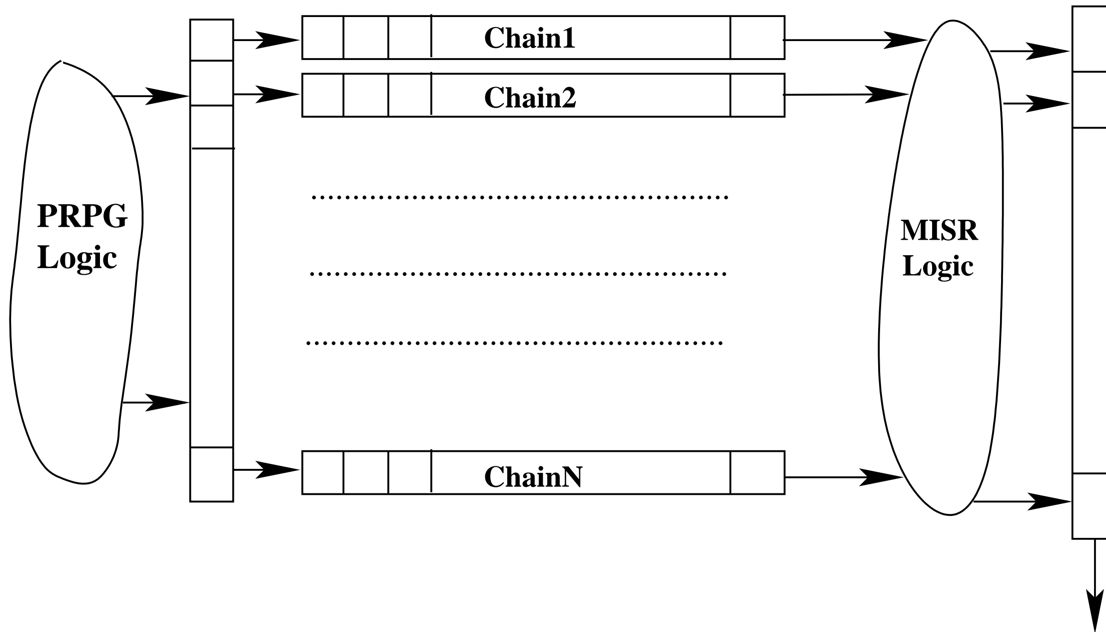

Figure 1. A basic BIST architecture.

the scan chains is also done at-speed. If parts of the logic to be tested operate at different frequencies, different controllers are used to test them at their respective frequencies. If only one controller must be used, then the entire logic is tested at the slowest possible frequency. The scan chain lengths are balanced as far as possible. The MISR logic can consist of complex combinational logic that manipulates the values captured and shifted out of the MISR each cycle.

For each BIST pattern generated, fault simulation is done to determine the faults detected each cycle. The rate at which the faults are detected is very high in the initial cycles and it tapers off towards the end of the BIST session. This is because faults may get detected multiple times and the new faults that were not detected before become fewer as the BIST session progresses. As a result, it may be expected that initial failures may happen at the very beginning of the BIST session since many of the faults get detected the first time in this phase. Since there are multiple scan chains and the possibility of logic sharing between different cones of logic going into the scan flip-flops, the same failures may be observed at multiple destination flip-flops. BIST failure diagnosis involves identifying the unique fault locations that caused incorrect values to be captured at one or more destination flip-flops.

3 Diagnosing BIST vectors

In this section, we will discuss the hardware design changes needed for diagnosing the BIST vectors. The proposed hardware changes can be easily accomplished for arbitrary designs and the accompanying method for diagnosis results in identifying the failing vector and the scan cells(s) at which the mismatches occurred. After the above are identified, we further describe how the actual fault locations in the circuit, that caused the mismatches, can be identified. It should be noted that the proposed methods are developed so that the solutions from various tools available in the industry can be put together to create a diagnostic solution for BIST failures. Details will be discussed later on in this section.

3.1 The diagnostic hardware

The modified BIST architecture in a multiple scan chain environment is shown in Fig. 2. Once the scan chains are completely loaded using the values coming from the LFSR logic, one system clock is executed and the values captured in the scan flip-flops are completely shifted out through the MISR. The above process constitutes one BIST vector. This is repeated for as many BIST vectors selected or the desired fault coverage is achieved. At the end of the BIST session, the signature coming out of the MISR is compared with the expected signature to determine whether or not the BIST session passed. Since the signature coming out of the MISR is highly compressed, a failing signature may not contain any clear information about which BIST vector(s) failed or which scan flip-flops captured incorrect values. We propose the following design changes to solve the above problem of identifying the failing BIST vectors and the scan cells capturing the incorrect values.

The proposed design modifications are shown in Fig. 2. The

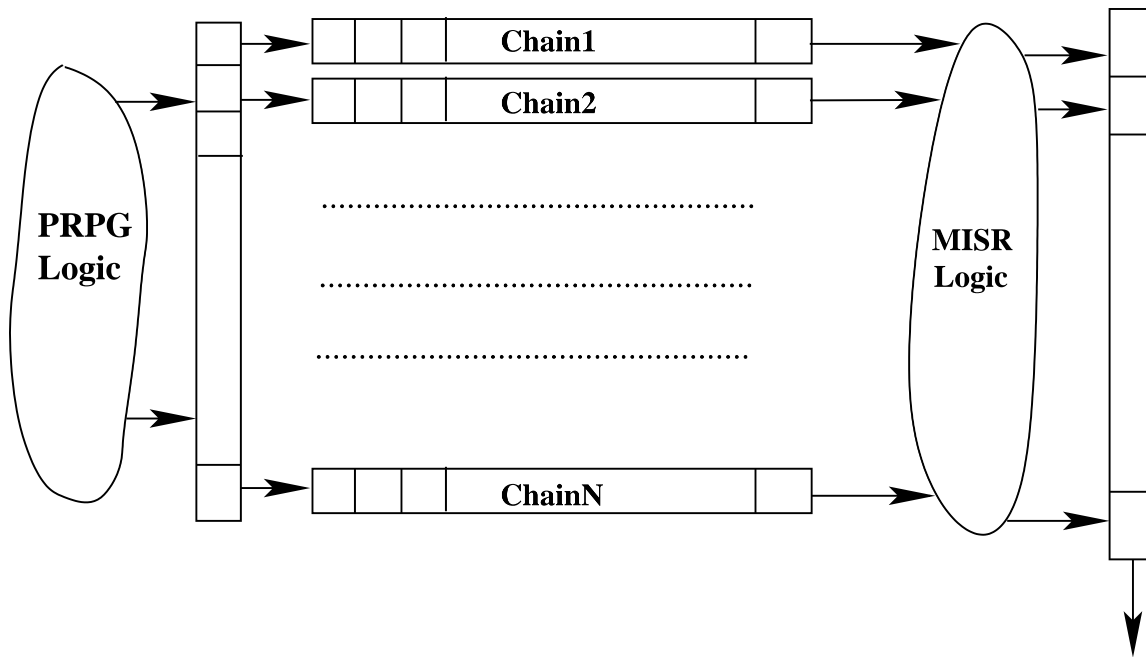

Figure 2. Selecting one scan chain for diagnosis.

final scan-out signal of each scan chain going into the MISR is intercepted with a 2-input multiplexer (MUX). One input to this MUX is held at a constant value 0 and the select line for each MUX is driven by a unique flip-flop output from a one-hot encoded finite state machine. The finite state machine has as many flip-flops as the number of scan chains in the design. The idea behind making the above design changes is as follows. Since the signature coming out of the MISR is a function of the values coming out of all the scan chains, the first step for determining which flip-flops captured incorrect values is to isolate the chains one at a time. This is not mandatory but it gives an easy and efficient way to locate scan cells with mismatches. The isolation of scan chains is achieved by allowing the values from only one scan chain to affect the values in the MISR. Since the select line for each MUX is driven by a unique flip-flop output from the one-hot finite state machine, constant 0 values from all the scan chains (with the corresponding MUX select line set to a 0) except one scan chain are fed into the MISR logic. There will be only one scan chain, whose values are shifted out as they are into the MISR logic since the select line for only its MUX is driven by a value 1. Hence, for each state of the one-hot finite state machine, one unique scan chain can be diagnosed independent of the values being shifted out of the other chains.

The above process ensures that one scan chain is debugged at a time. The next step is to somehow make sure that the values from the single scan chain being diagnosed, serially shifted out of the MISR logic unmodified. This helps us in observing the scan chain values every BIST cycle and in turn determine the BIST vector that failed and the scan cells having the mismatches in the failing cycle.

Fig. 3 shows the additional multiplexers needed to make sure the contents of one scan chain are transferred to the MISR for eventual shifting out before a new BIST vector is loaded into the scan chains. The connections to the XOR gates in the MISR logic are intercepted by a MUX whose other input is connected to a constant 0. The select lines of these additional multiplexers are driven by a test signal that selects the input driven by a constant 0, during the time the scan chain values are shifted through the MISR logic. It can be seen that the second input of the multiplexers used to select one chain for shifting its values through the MISR logic is also driven by a constant 0. During the diagnosis process, given the above described extra hardware, all the scan chains except the one being diagnosed shift out constant 0 values. The outputs of all the multiplexers driving the XOR gates are also 0's. As shown in Fig. 3, in this situation, only the values coming from the scan chain being diagnosed are captured in the MISR unmodified and shifted out through the MISR. For example, if the MISR length is 20 bits and the fifth scan chain is being diagnosed, the values from the fifth scan chain come out of the MISR serially after the first five shifts or fifteen shifts depending on whether the serial shift out of the MISR is the closest or farthest to the fifth bit of the MISR. This response from the scan chain must be compared to the expected values captured in the scan chain to see if the BIST vector failed and caused any mismatches in that particular scan chain.

The additional hardware needed to select one scan chain at

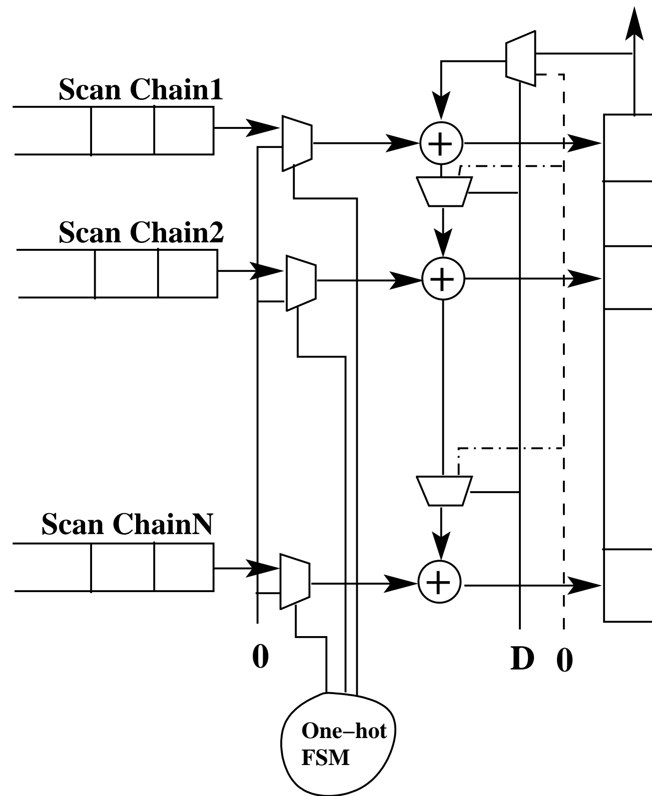

Figure 3. Single chain contents shifted out without modifications.

a time and to shift its values through the MISR is composed of the one-hot finite state machine, the routing needed for the constant 0 signal, the additional test signal D and the multiplexers in the scan chain selection logic and the MISR logic. The total number of 2-input multiplexers needed is equal to the number of scan chains plus the number of inputs to the XOR gates in the MISR logic, minus the number of scan chains. The number of scan chains must be subtracted in the above equation because, the values coming from the scan chains must be propagated through the MISR XOR logic as it is.

3.2 The diagnostic process

We shall now explain the overall operation of the modified BIST logic shown in Fig. 3. During normal operation, the one-hot finite state machine must be placed in the reset state so that the multiplexers at the final shift-out of the scan chains select the values coming from the scan chains. The D signal is asserted to a value such that the multiplexers select the input other that the one held at constant 0. In this mode, the circuit functions as if the extra hardware is not present at all.
If the signature at the end of the BIST session is different from the expected one, the additional hardware can be used as follows. The one-hot finite state machine is allowed to go into one of its legal states. Lets assume that the first legal state enables scan chain 1 to shift its values only through the MISR logic. The BIST patterns are shifted into the scan chains in the normal fashion. Once all the scan chains are loaded, the circuit goes through one system clock and the new values are captured into the scan chains. Note that even though the first scan chain is only allowed to shift its values through the MISR logic, all the scan chains are loaded so that all the interactions with the logic going into the first scan chain are appropriately handled. The values from the first scan chain are now shifted through the MISR logic by selecting the appropriate value of D. This process continues until differences are noticed in the values coming out of scan chain 1 for every cycle. This way, the first BIST vector that caused incorrect values to be captured into first scan chain can be identified. At the same time, the mismatching scan cells in the first scan chain can also be identified by determining which scan cells in that scan chain captured the incorrect values.

The above process can be repeated for each state of the finite state machine so that a different scan chain is selected each time for diagnosis purposes. Each time the failing vector and the corresponding faulty scan cells are identified, the individual fault sites in the circuit that caused the scan cells

to capture incorrect values must be identified. We explain this process in the next sub-section.

3.3 Identifying the responsible faults

After the failing BIST vector and the scan cells with mismatches are identified, the next step is to identify the faults in the circuit that caused the mismatches to occur. In order to do this, we need to identify the faults that are detected by the failing vector and eliminate those that are detected by earlier vectors. We also need to make sure we eliminate those faults that are detected by the failing vector at the scan cells with mismatches and also at other scan cells. This step avoids classifying some faults as responsible for the failure when in fact they are not. We consider only stuck-at fault based BIST diagnosis in this work.
The BIST tools available in the industry operate on designs that are so large that it requires a lot of memory to report all the faults in the design. They only list the faults that are not detected by the BIST vectors since the memory resources needed for listing them are more manageable. We make use of the list of undetected faults to figure out the set of faults detected by the failing BIST vector.
Initially, we simulate all the faults in the design for n cycles, where n is the failing BIST vector number. We collect the faults that are not detected by n BIST vectors. We then simulate all the faults for n - 1 cycles and collect the list of faults that are not detected by n - 1 BIST vectors. We get the list of faults detected by the nth BIST vector by taking the difference between the two lists of faults above. We have to prune this set of detected faults by considering only those that are detected by the nth vector at the mismatching scan cells only.

4 Experimental Results

We performed experiments on an ASIC that has several blocks in it. Every block has logic BIST in it and memory BIST wherever needed. Most of the blocks have a single clock domain. If a block has more than one clock domain, the methodology is to use a unique logic test controller for each domain or to use only one logic test controller running at the slowest frequency of the clock domains. The statistics for the block used in our experiments are shown in Table 1.

# Chains	96	Max # Flops	498
# Gates	433537	# Lines	798884
# PIs	46543	# POs	48650
# Faults	812911	# Clocks	1
# Vectors	16K	FC w/o test points	84.82%
FC with 100 test points	95%		

Table 1. Statistics for the block.

Columns 1 and 3 in Table 1 represent the various parameters related to the circuit. Columns 2 and 4 give the corresponding numbers for each parameter. The block has multiple scan chains and there are a total of 96 scan chains. The maximum number of scan flip-flops in any chain is 499. It can be seen that the number of gates and lines in the block are fairly large. There is only one main clock driving this block but in general, there can be more than one input clock to a block resulting in as many clock domains. The single stuck-at fault coverage for the block using 16K vectors and without any test points is 84.82%. The corresponding coverage with 100 test points and the same number of vectors is 95%. Out of the 100 test points, 78 are controlling type and 22 are observation type.
Since there are 96 scan chains in the design, 96 flip-flops are needed for the one-hot FSM for observing the values on one chain at a time, through the MISR. Also, 96 2-input multiplexers are needed as shown in Fig. 2 for disabling all the other scan chains when one scan chain is being diagnosed. The second input of these multiplexers is tied to the ground. A reset signal is needed to place the one-hot FSM in the all-0 state during normal operation. All the above logic is synthesized and added to the original BIST inserted design. As shown in Fig. 3, the MISR logic needs some modifications so that the values from the scan chain being diagnosed go through the MISR logic unmodified. The MISR logic for the above block is such that there are 91 2-input XOR gates for which one of the inputs is the scanout signal from a unique scan chain. As shown in Fig. 3, one of the inputs to the XOR gates, coming from the feedback paths, must be intercepted by a MUX so that the input to the XOR gate can be controlled to a 0. Hence, for this design, we need 91 2-input multiplexers whose select line selects the scanout signal from the scan chains during normal operation. The second input of each MUX is tied to a constant 0 value, which can be connected to the inputs of the multiplexers in Fig. 2 that are tied to a 0.
The next step in the experiments is to diagnose the failing patterns using a software approach. For experimental purposes, we assumed the first cycle numbers at which the actual signature in the MISR is different from the expected one. If the circuit under consideration is already manufactured, the proposed technique can be used to determine the first failing BIST vector and the corresponding scan cells with mismatches. This is accomplished by making the one-hot FSM select one scan chain at a time and get the signatures for each chains. These signatures can be compared with the expected signatures to determine the failing vector numbers and the exact scan cells with mismatches.
Table 2 shows the results assuming different failing vectors. It can be seen that most of the stuck-at faults are detected in

the early BIST vectors and hence the increased probability that failures may be observed in the initial vectors itself. Five different vectors were chosen as the failing vectors based on the feedback from the manufacturer where testing was also done. Column 1 shows the failing vector numbers and Columns 2 and 3 show the total number of detected faults until the failing vector and the number of faults detected by the failing vector respectively. Note that the BIST fault simulator we used, drops the faults as it goes along and hence the faults reported in columns 2 and 3 are the unique faults detected until the cycle in Column 1. Simulating a fault only until its first detection is acceptable for diagnosis purposes because if a fault detected in an earlier cycle is responsible for the BIST failure, then the BIST session would have failed in the earlier cycle itself.

vector #	Det. Faults		Suspects Failing Scan Cells		
	Total	Failing Vec.	2	4	6
32	629711	416	23	31	12
41	695271	184	12	19	27
65	701439	141	16	26	12
74	703150	225	36	27	31
100	705992	115	9	21	17

Table 2. Suspects for different failing vectors.

Columns 4, 5 and 6 in Table 2 show the actual number of suspects when 2, 4 and 6 scan cells have incorrect values captured in them for the corresponding failing vectors shown in Column 1. These suspects were determined as explained earlier by pruning the detected faults and retaining those that were detected at the failing scan cells only. The scan cells chosen in Columns 4, 5 and 6 are distinct and they do not have any common scan cells. Also, the scan chains to which they belong are also different. It can be seen that the pruning results in a large reduction in the faults responsible for the failure.

It is clear from the above results that the proposed technique can be used effectively for determining the suspects for BIST failures. The proposed hardware enhancements do not pose any restrictions on the area overhead in our designs since considerable area requirements for test purposes have been accommodated. It was seen how transparent the additional logic will be during normal operation and how easily the additional circuitry can be made operational in the diagnosis mode by making the one-hot FSM go into one of its legal states and activating the select lines of the multiplexers. Also, there is no timing impact on the functional logic since the additional hardware is associated only with the MISR logic. If there are multiple clocks in the design that has logic BIST in it, the BIST circuitry operates at a frequency less than or equal to the slowest clock in the design. Further investigation is needed to see if the proposed hardware enhancements are still valid in such a situation since the MISR modifications are combinational in nature and the one-hot FSM operates at the frequency of the BIST logic.

In the proposed diagnosis technique, larger number of scan chains can have a direct impact on the extra hardware since it requires a larger one-hot FSM and extra multiplexers in the MISR logic to make sure only one chain shifts out its contents through the MISR. A better technique that allows multiple chains to shift out their contents through the MISR reduces the hardware overhead needed in the MISR logic. If the PRPG and MISR sizes are larger than the number of scan chains in the design, the proposed method can be easily extended to add extra multiplexers in the MISR logic that ensure the values coming out of a scan chain shift out of the MISR in an unmodified manner. However, if the size of the MISR is smaller than the number of scan chains and the values coming out of the scan chains are compacted before they reach the MISR itself, the proposed hardware enhancements in the MISR logic cannot be applied in a straight-forward manner. The diagnosis methodology for such a situation becomes more complicated and better techniques are needed to ensure correct diagnosis of the BIST failures.

5 Conclusion

In this work, we presented a combination of software and hardware methods for diagnosing logic BIST failures. The hardware part of the solution provided a means for diagnosing one scan chain at a time by using a finite state machine that isolates the other chains from interacting with the MISR. The method further enhanced the visibility of the each scan chain by intercepting the XOR gates in the MISR logic to make sure the values from a single scan chain go through the MISR logic unmodified. We showed that this method can be used to determine the failing vectors and scan cells with mismatches. Once the mismatching scan cells are identified, a method is proposed to determine the faults in the circuit that caused the scan cells to capture incorrect values.

Experiments were performed on a block in an industrial design and results were obtained in terms of the suspects causing the BIST session to fail. The proposed technique was successfully implemented in the industry and the hardware changes were also easily incorporated into the industrial designs with minimal effort. It clearly shows that the solutions proposed in this work are practical and easy to implement using the current techniques and methodologies in practice. The techniques proposed here are applicable to designs with multiple scan chains and separate BIST controllers for different blocks in the design. Effort is in progress to evaluate the failures on the interconnect for different blocks in the

design. Methods must be proposed to test the interconnect and diagnose their failures. The proposed analysis is for determining the suspects must be extended to BIST methodologies that target newer defect models and the corresponding types of faults. The effect of multiple clock domains in a block creates new challenges for BIST diagnostic efforts. Work is in progress to evaluate the interactions of multiple clock domains and their impact on BIST diagnosis.

References

[1] R.C. Aitken and V.K. Agarwal, "A diagnosis method using pseudo-random vectors without intermediate signatures", *Proc. Intl. Conf. Computer Aided Design*, pp. 574-577, 1989.

[2] S. Kundu, "On diagnosis of faults on a scan-chain", *Proc. VLSI Test Symposium*, pp. 303-308, 1993.

[3] T.R. Damarla, 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.

[4] Y. Wu and S. Adham, "BIST fault diagnosis in scan-based VLSI environment", *Proc. Intl. Test Conf.*, pp. 48-57, 1996.

[5] S.-Y. Huang et. al., "Error Tracer: A fault simulation based approach to design error diagnosis", *Proc. Intl. Test Conf.*, pp. 974-981, 1997.

[6] J. Rajski and J. Tyszer, "Fault diagnosis in scan based BIST", *Proc. Intl. Test Conf.*, pp. 894-902, 1997.

[7] A. Veneris and I.N. Hajj, "Design error diagnosis and correction via test vector simulation", *IEEE Trans. CAD*, 18(12):1803-1816, Dec. 1999.

[8] J. Rajski and J. Tyszer, "Diagnosis of scan cells in BIST environment", *IEEE Trans. Computers*, vol. 48, Issue: 7, pp. 724-731, 1999.

[9] S. Venkataraman and S.B. Drummonds, "A technique for logic fault diagnosis of interconnect open defects", *Proc. VLSI Test Symposium*, pp. 313-318, 2000.

[10] S.-Y. Huang, "On improving the accuracy of multiple defect diagnosis", *Proc. VLSI Test Symposium*, pp. 34-39, 2001.

[11] I. Bayraktaroglu and A. Orailoglu, "Diagnosis of scan based BIST: Reaching deep into signatures", *Proc. Design Automation and Test in Europe*, pp. 102-109, 2001.

[12] R. Guo et. al., "A technique for fault diagnosis of defects in scan chains", *Proc. Intl. Test Conf.*, pp. 268-277, 2001.

[13] N. Sridhar and M.S. Hsiao, "On efficient error diagnosis of digital circuits", *Proc. Intl. Test Conf.*, pp. 678-687, 2001.