

Improving the proportion of At-Speed Tests in Scan BIST⁺

Y. Huang¹

I. Pomeranz²

S.M. Reddy¹

J. Rajski³

¹ Department of Electrical & Computer Engineering, University of Iowa, Iowa City, IA 52242

² School of Electrical & Computer Engineering, Purdue University, West Lafayette, IN 47907

³ Mentor Graphics Corporation, 8005 S.W. Boeckman Rd., Wilsonville, OR 97070

Abstract

A method to select the lengths of functional sequences in a BIST scheme for scan designs is proposed in this paper. A functional sequence is a sequence of primary input vectors applied when the circuit operates as a sequential circuit, without using scan. These sequences can be applied at-speed, i.e., at the normal circuit clock speed. The objectives set for choosing the lengths of the functional sequences are to increase the number of vectors applied at-speed, and to reduce the number of settings of functional sequence lengths, without compromising the fault coverage achieved. The experimental results presented demonstrate that compared to earlier methods, the proposed method achieves the above objectives while also achieving higher fault coverages for most of the benchmark circuits considered.

1 Introduction

Scan-based BIST methods can be divided into two classes [1]. The first class includes test-per-scan BIST methods, and the second class includes test-per-clock BIST methods. In the test-per-scan architecture, a random state vector is serially scanned into the flip-flops of the circuit. After the complete vector is scanned in, a test vector is applied to the primary inputs and its results are observed at the primary outputs. At the same time, the next state is captured into the scan chain. In the next cycle, a new state is scanned in and the captured response of the last cycle is scanned out and observed. In the test-per-clock architecture, a test vector is applied to the primary inputs every clock cycle. In the work reported here, we use a test-per-clock multiple scan chain BIST scheme. However, the analysis can be extended to the test-per-scan architecture.

In either BIST schemes for scan designs mentioned above, typically one vector of primary input values is applied after scanning in a state vector. The circuit response is captured in the memory elements and then scanned out. Thus, a single primary input vector is applied between scan operations. The input vectors applied and the states scanned in are generated by a pseudo-random pattern generator [1]. However, it may be advantageous to apply a sequence of input vectors of

length greater than one (i.e., use more than one capture cycle) between scan operations. Note that the input vectors applied between scan operations can be applied at-speed using the system clock. If the total number of test clock cycles (scan plus system clocks) is fixed, using longer input sequences between scan operations allows the application of a larger proportion of input vectors at-speed. The clock frequency of scan clocks and system/functional clocks are often different, with scan clock frequency being much lower in designs that do not optimize the scan path for speed. The clock frequencies may also be based on restrictions on power dissipation during testing, especially in scan based testing which may drive the circuit into illegal or functionally unreachable states. However, in many designs it may still be possible to use a higher frequency for functional clocks than scan clocks while meeting power dissipation constraints. Thus, applying more functional clocks and fewer scan clocks will allow a reduction in total test time. In addition, earlier research [2] has indicated that at-speed tests cover unique defects not covered by other tests. Furthermore, the recent work of Tsai et. al.[3] demonstrated that for a given total number of test clocks, using multiple test sessions with different numbers of capture cycles in each test session improves the fault coverage achieved.

In this work, we describe a procedure to select the lengths of the subsequences of input vectors applied between two consecutive scan operations (i.e., the number of capture cycles) that improves upon the method proposed by Tsai et. al.[3]. The proposed method selects a limited number of different subsequence lengths (two) in order to simplify test application. The lengths of the input sequences applied between scan operations are selected to be high in order to increase the proportion of tests that can be applied at-speed. The advantages compared to the earlier method of [3] are achieved due to an improved method for identifying the best subsequence lengths. This method allows us to use fewer different lengths and achieve higher fault coverage while applying a higher proportion of test vectors at-speed.

The method reported is motivated by an earlier work [4] on design for testability. In [4] it was shown

⁺ Research supported in part by SRC Grant No. 98-TJ-645.

that if the state of a sequential circuit can be set to any arbitrary value, then all the sequentially irredundant faults in the circuit can be detected by applying input vectors and observing only the circuit outputs. This result implies that controlling the states of a sequential circuit is sufficient, and specifically, observation of the state of the circuit is not necessary to detect sequentially irredundant faults. Experimental results using deterministic test generation and using BIST demonstrated the effectiveness of this approach [4]. In the context of BIST for scan designs this result implies that more than one input vector between consecutive scan operations could be useful in achieving higher fault coverage while utilizing a smaller total number of clock cycles. This observation arrived at from a different perspective was the basis for the work in Tsai et. al.[3].

The paper is organized in the following manner. In Section 2 we review the method proposed in [3] to enhance the effectiveness of scan-BIST. In Section 3 we present the proposed approach for scan-BIST. Experimental results are presented in Section 4. Section 5 concludes the paper.

2 Preliminaries

In this section we review the terminology and the procedure for scan-BIST given in [3].

Definition 1:

A *scan cycle* is the period during which a new circuit state is shifted into (and the current state is shifted out of) the scan chain. If the length of the longest scan chain is l , then one scan cycle corresponds to l clock cycles.

Definition 2:

A *functional cycle* is the period between two scan cycles. During this period the circuit functions as a sequential circuit. If an input sequence of length k is applied during this period, we say that the *length* of the functional cycle is k .

Definition 3:

A *test cycle* is one scan cycle followed by one functional cycle. The length of the test cycle (i.e. the number of clock cycles) is $l+k$, where l is the length of the scan cycle and k is the length of the functional cycle.

The probability of detection of a fault during a functional cycle is computed in [3] by using the following formula:

$$Pd_i^k = 1 - \prod_{j=1}^k (1 - Pd_{i,j}) \quad (1)$$

where Pd_i^k is the detection probability of fault i during a functional cycle of length k , and $Pd_{i,j}$ is the detection probability of fault i in the j th time frame of the functional cycle. The detection probabilities are computed by using COP [5] with appropriate modifications [3].

Based on the observation that the optimum length of a functional cycle varies from one fault to another, it was proposed in [3] that different lengths of functional cycles be used during testing of a circuit using

scan-BIST. A procedure is presented in [3] to determine a set of functional cycle lengths and to divide the total number of clock cycles budgeted for testing into multiple test sessions each with a different functional cycle length. The procedure used in [3] to determine the set of lengths of the functional cycles to be used is described next.

The procedure first identifies a set of hard-to-detect faults. Only these faults are considered in determining functional cycle lengths to be used. For the selection of functional cycle lengths, the procedure uses a metric called *Detection Probability per Clock* of a fault i defined as:

$$DPC_i^k = Pd_i^k / (l+k) \quad (2)$$

For each fault i , the procedure first determines the optimum value of k such that DPC_i^k for the fault is maximum. Each fault i is placed in a set S_k , $k=1,2,\dots$, where k is the optimum value of functional cycle length for fault i . The subsets S_1, S_2, \dots , are then ordered in decreasing order of their cardinalities. For a desired fault coverage, say C , the set of lengths of functional cycles to be used is next chosen in the following way. Let K be the set of functional cycle lengths to be used. The members of K correspond to the largest subsets of faults S_j defined above such that $| \bigcup_{j \in K} S_j | / |F| > C$, where $|X|$ is the cardinality of set X .

The optimum value k of the functional clock cycle length for every fault i (needed to place fault i in the appropriate subset of faults S_k) is determined by computing DPC_i^k for every fault i using increasing values of k . If $DPC_i^{k+1} > DPC_i^k$ when k is increased to $k+1$, then fault i is placed in S_{k+1} .

The highest functional clock cycle length is held down as follows in the procedure given in [3]. The value of k is increased only if the number of faults for which DPC_i^k increases from their current maximum value is over half of the number of targeted faults. The procedure terminates as soon as this condition fails for the first time.

The procedure of [3] for improving the effectiveness of scan BIST leads to improved fault coverages for a given number of clock cycles compared to the standard scan-BIST approach using a functional cycle length of one. However, typically, several different functional cycle lengths are obtained by the procedure of [3], and the lengths are typically small. As explained earlier, longer functional cycle lengths result in a higher proportion of tests that can be applied at-speed. Furthermore, reducing the number of different settings of functional cycle lengths may lead to a simplified BIST controller. In the next section, we propose an iterative procedure to select the lengths of the functional cycles in order to increase their lengths, as well as reduce the number of different functional cycle lengths used. As a by-product, a higher fault coverage is also obtained in many cases.

3 Iterative Procedure to Select Lengths of Functional Cycles

Similar to the methods in [3] and [4], the proposed method enhances the effectiveness of scan-BIST by using functional cycles with lengths greater than one. However, it differs in the following fundamental ways from the methods in [3] and [4].

1. The method investigated in [4] did not use scan, and did not propose specific methods to select different functional cycle lengths.
2. Unlike in [3] where all the functional cycle settings to be used are determined simultaneously, in the proposed method these lengths are determined one at a time. Specifically, after one functional cycle length is chosen, all the faults detected by tests using the chosen functional cycle length are dropped. The next functional cycle length is determined based on the yet undetected faults. In this paper we also restrict the different number of functional cycle lengths used to two.
3. The analysis used to select the functional cycle length settings is different from the one used in [3]. As demonstrated by the experimental results presented later, the proposed analysis leads to the selection of longer functional cycle lengths, leading to a higher proportion of at-speed tests and also a higher fault coverage in many cases. Furthermore, binary search over the lengths of the functional cycles is used to identify the optimal values for the second setting, unlike the linear search in [3].

The first setting of the length of the functional cycles is chosen to maximize the probability of detection during the functional cycle for a majority of the targeted faults.

Let $P_{di}(N, k, l)$ be the probability that a fault i is detected when a total of N clock cycles are applied, with the length of the functional cycle used equal to k and the length of the scan cycle equal to l . Equation (3) given below gives $P_{di}(N, k, l)$. Pd_i^k used in Equation (3) is defined in Equation (1).

$$P_{di}(N, k, l) = 1 - (1 - Pd_i^k)^{N/(k+l)} \quad (3)$$

In Equation (3), $N/(k+l)$ is the number of different test cycles applied to the circuit. (A test cycle consists of a scan cycle and a functional cycle.) A fault is detected if it is detected during any one of the test cycles. Using Equation (3) one can choose between two different functional cycle lengths k_1 and k_2 by picking, say k_1 over k_2 if:

$$P_{di}(N, k_1, l) > P_{di}(N, k_2, l)$$

Substituting Equation (3), we obtain:

$$\begin{aligned} 1 - (1 - Pd_i^{k_1})^{N/(k_1+l)} &> 1 - (1 - Pd_i^{k_2})^{N/(k_2+l)} \\ (1 - Pd_i^{k_1})^{N/(k_1+l)} &< (1 - Pd_i^{k_2})^{N/(k_2+l)} \\ N \lg(1 - Pd_i^{k_1}) / (k_1 + l) &< N \lg(1 - Pd_i^{k_2}) / (k_2 + l) \\ \lg(1 - Pd_i^{k_1}) / (k_1 + l) &< \lg(1 - Pd_i^{k_2}) / (k_2 + l) \quad (4) \end{aligned}$$

Therefore we define as our metric for selecting a functional cycle length k :

$$M_i^k = \lg(1 - Pd_i^k) / (k + l) \quad (5)$$

The first step in selecting the first functional cycle length is to partition the faults into subsets S_1, S_2, \dots, S_m according to their optimal functional cycle length. All the faults in S_k have an optimal length of k . The partitioning procedure is similar to the procedure from [3], except that the metric defined in Equation (5) is used to find the best value of k for each fault, instead of the metric defined in Equation (2) used in [3]. Note that a larger DPC_i^k means a better choice of k , while a smaller M_i^k means a better choice of k from Equation (4).

Once the sets S_1, S_2, \dots, S_m are found, we consider the two sets that contain the largest numbers of faults, say S_{k_1} and S_{k_2} . We select either k_1 or k_2 as the first functional cycle length, according to the following considerations. A larger set of faults S_k implies that more faults are likely to be detected using functional cycle length k . However, if k is too large, then fault effects may need to propagate through a large number of time frames before they can be observed. This may prevent us from detecting certain faults. In addition, the second cycle length will be longer than the first one. As a compromise between these considerations that may be conflicting, we use the value $|S_k|/k$. The value $|S_k|/k$ is larger for larger sets S_k , but smaller for larger values of k . If $|S_{k_1}|/k_1 > |S_{k_2}|/k_2$, we select k_1 as the first functional cycle length. Otherwise, we conclude that k_1 is too large, and we select k_2 .

Once we select either k_1 or k_2 as the first functional cycle length, we apply half of the budgeted clock cycles for testing the circuit with the selected functional cycle length, and drop the detected faults from further consideration.

To select the second functional cycle length, we consider the remaining, undetected faults. Using binary search on values of k between the value selected in the first step (say k_1) and 256, our goal is to find the value of k that maximizes the overall probability of detection of the faults that remain undetected. Binary search proceeds as follows.

We start searching from $k = k_1 + l$ with an upper limit of 256. For each k , we build a fault list C_k that consists of faults that are likely to be detected using k as the second functional cycle length. A fault i is put into C_k only if $M_i^k < (1 + \alpha) * M_i^{k_1}$, where α is a gain factor (α is initialized to 0.1). The building of the candidate fault list is based on the assumption that a fault is hard to detect if its metric M_i^k is larger than $(1 + \alpha) * M_i^{k_1}$. If $|C_k| < 10\% * |F|$, where $|F|$ is the total number of yet undetected faults, we move on to try the next value of k . (In this case, not enough of the faults will be detected, and a different value of k is needed.) The next value of k is determined by binary search as follows.

Suppose that $k_l=7$, we start our search with $k=8$. The next value of k is 16. If we get a better result for this value than for the previous one, we continue to search $k=32$. If the result shows that $k=32$ is worse than $k=16$, we go back to try $k=24$. In this way we will settle down at an optimal point with reasonable computational effort.

If for all the values of k considered, $|C_k| < 10\% * |F|$, we have to relax the requirement of the gain factor α . In our method, we reduce α to $\alpha - 0.01$ and repeat the above procedure until we settle down at an appropriate α . Note that M_i^k is a negative number, therefore, reducing α will allow more faults to be included in C_k .

As each value of k is considered, we use Equation (3) to find the probability of detection $P_{di}(N, k, l)$ of every fault i which has been put in C_k . Here, N is equal to half of the number of test clocks budgeted (since the first half of these clock cycles were used after choosing the first functional cycle length). Then we compute

$$\sum_{i \in C_k} P_{di}(N, k, l),$$

which gives the estimate of the number of faults that can be detected during N clock cycles with functional cycles of length k . Since $|C_k|$ usually varies with the value of k , we normalize the above sum with respect to the number of faults in the candidate fault list for functional cycle length of k , obtaining,

$$DI_k = \sum_{i \in C_k} P_{di}(N, k, l) / |C_k|.$$

DI_k stands for the *Detection Index* of k , which can be used to estimate the detection capability for different values of k . At the end of the process we select a value of k with maximum DI_k . This maximizes the probability of detection of the remaining faults. We use this value as the second functional cycle length. This process is described in Figure 1.

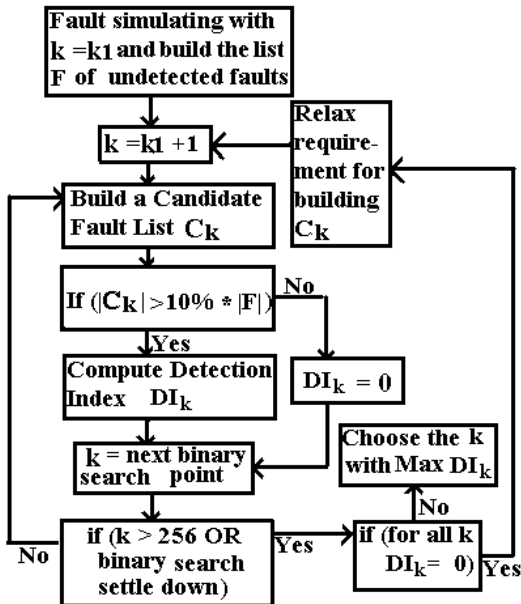


Figure 1: Procedure to select the second functional cycle length

4. Experimental Results

In Table 1 we report the results of applying the proposed procedure to select functional cycle lengths for larger ITC'99, ISCAS 89 and Addendum 93 benchmark circuits. The results using multiple functional cycle lengths using the procedure from[3] are reported under columns with the heading MTS, the results of the proposed procedure using two functional cycle lengths are reported under columns with the heading TTS, and the results of the standard scan BIST using a single input vector between consecutive scan operations are reported under columns with heading STS. (TS stands for test session; each test session uses a different length for its functional cycles) For each benchmark circuit the total number of clocks applied to test each circuit is 500K. In order to compare with the previous method, we also set the length of the longest scan chain to 10 as was done in [3]. Scanned state vectors and non-scan input vectors were generated randomly using a random number generator. For each circuit, ten different runs using ten different initial seeds for the random number generator were applied. For each circuit, we report the lengths of the functional cycles for MTS and TTS in columns 2 and 3, respectively. The percentages of at-speed tests among all the tests are shown for MTS and TTS in columns 4 and 5, respectively. The run times of the procedures to select the subsequence lengths are given in columns 6 and 7. The average fault coverages achieved over ten different runs for each procedure are given in the last three columns.

As an example consider circuit B21. The procedure MTS uses five different functional subsequence length settings of 2, 3, 5, 6 and 7, whereas the proposed procedure TTS uses two subsequence length settings of 14 and 256. The average fault coverage achieved by MTS over ten different random tests is 85.77% and the average fault coverage achieved by the proposed method is 87.50%. Furthermore, MTS would allow 30.4% test clocks to be at-speed whereas the proposed method allows 77% of test clocks to be at-speed.

In conclusion, for the set of benchmark circuits considered, the proposed procedure achieved higher fault coverages for twelve out of sixteen circuits while using only two settings of functional subsequence lengths. At the same time, the proposed method achieves higher proportion of at-speed test vectors for all the circuits used. The average results for all benchmark circuits, shown in the last row of Table 1, also show that the proposed method achieves the best fault coverage among the three methods and allows on the average 53.3% tests at-speed. In contrast, MTS typically uses more than two test sessions and allows 21.3% at-speed tests.

5. Conclusion

We developed a method to select the lengths of functional sequences in a BIST scheme for scan designs. The advantages of this algorithm are:

- (1) It increases the number of test vectors applied at-speed.
- (2) It reduces the number of settings of functional sequence lengths, which simplifies the BIST controller.
- (3) On an average, it achieves higher fault coverage than those achieved by a previous multiple test session scheme.

References

- [1] V.D.Agrawal, C.R.Kime and K.K.Saluja, "A Tutorial on Built-In Self-Test, Part 2: Applications," IEEE Design & Test of Computers, vol. 10, no. 22 pp.69-77, June 1993.
- [2] P. C. Maxwell, R. C. Aitken, V. Johansen, and I. Chiang, "The Effect of Different Test Sets on Quality Level Prediction: When is 80% Better Than 90%?" Proc. ITC., pp.358-364, Sept. 1991.
- [3] H.C.Tsai, K.T. Cheng and S.Bhawmik, "Improving The Test Quality for Scan-Based BIST Using A General Test Application Scheme," Proc. of DAC, pp.748-753, June 1999.
- [4] I. Pomeranz and S. M. Reddy, "On Full Reset as a Design-for-Testability Technique," VLSI Design, pp. 534-536, Jan. 1997; also in "On the Use of Fully Specified Initial States for Testing of Synchronous Sequential Circuits," IEEE Trans. On Computers, pp.175-182, Feb. 2000.
- [5] F. Brglez, P. Pownall, and P.Hum, "Application of Testability Analysis: From ATPG to Critical Path Tracing," Proc. IEEE Int. Test Conf., pp.705-712, Sept. 1984.

Table 1: Experimental Results for benchmark circuits:

Bench mark	Functional cycle lengths		Percentage of at-speed tests (%)		CPU Time (Sec.)		Average Fault Coverage For 10 different seeds		
	MTS	TTS	MTS	TTS	MTS	TTS	STS	MTS	TTS
S3330	1,2,3	1, 252	16.3	52.6	0.600	12.566	89.03	92.34	92.56
S3384	2,3,4	9, 252	22.8	71.8	0.617	11.050	96.62	96.91	97.44
S4863	1,4,5	9, 252	23.7	71.8	0.900	16.433	97.82	99.74	99.91
S5378	1,2,3	1,16	16.3	35.3	1.017	3.300	98.51	98.66	98.54
S9234.1	1,2,3,4	6, 7	19.4	39.3	2.700	7.200	94.16	95.95	95.89
S15850.1	1,2,3	1,32	16.3	42.6	5.283	23.067	92.67	92.95	92.39
S38417	1,2,3	3, 6	16.3	30.3	11.95	31.250	95.83	96.96	97.06
S38584	1,2,3	4,24	16.3	49.6	11.22	40.150	95.52	95.59	95.36
B14	2,3,5,6,7,8	17, 48	32.1	72.9	2.367	23.783	82.58	84.20	85.00
B15	2,3,4,5	6,12	25.4	46.0	4.233	11.283	90.64	96.24	97.21
B17	2,3,4	6, 10	22.8	43.8	13.93	39.200	91.16	96.02	96.69
B18	2,3,4	6,8	22.8	41.0	35.43	100.33	90.19	94.65	95.18
B19	2,3,4	6,8	22.8	41.0	77.85	217.66	89.92	94.46	95.05
B20	2,3,5,6	10,216	27.6	72.8	5.583	84.966	86.59	89.73	91.01
B21	2,3,5,6,7	14,256	30.4	77.0	5.917	96.467	82.07	85.77	87.50
B22	2,4,5,6	9,48	29.0	65.1	9.317	45.670	86.69	89.11	90.55
Avg.	————		21.3	53.3	11.81	47.77	91.25	93.71	94.21